# EXPERIENCE WITH A WIRELESS NETWORK TESTBED BASED ON SIGNAL PROPAGATION EMULATION

*Kevin Borries, Xiaohui Wang, Glenn Judd, Peter Steenkiste, Daniel Stancil*

Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15215, USA
phone: +1 412 268 3261, fax: +1 412 268 5576, email: prs@cs.cmu.edu
web: www.cs.cmu.edu/˜ prs

## ABSTRACT

The evaluation of wireless research is challenging because signals traveling through the ether are affected by the physical environment, including movement by people and objects. As a result, testbed experiments are hard to control and are non-repeatable. We have developed a wireless networking testbed based on digital signal propagation emulation that provides control over the signal propagation environment. The testbed has been in regular use for research and education since early 2007. In this paper we present measurements illustrating the properties of the emulator testbed. We also compare the results of various experiments on the emulator with simulation results to shed some light on when the increased accuracy of the emulator testbed is important. Finally, we use the experience gained on the emulator to identify classes of experiments for which the emulator is well suited, compared with other evaluation platforms.

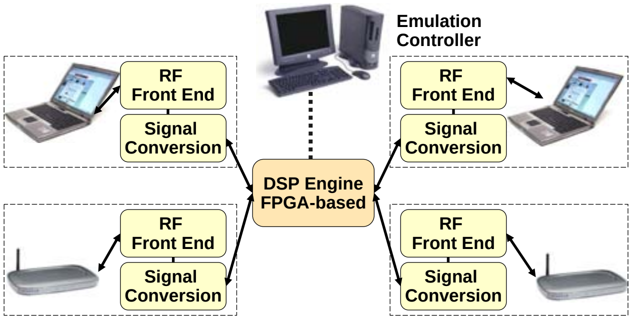


Figure 1: Emulator Architecture

## 1. INTRODUCTION

The evaluation of wireless networking protocols and systems is a challenging problem. The reason is that the performance of the wireless network is intimately tied to the physical environment, so changes in the environment, e.g. people moving around or objects being moved, affects the outcome of experiments. As a result, experiments are hard to isolate, are non-repeatable and hard to control, and side-by-side comparison of alternate solutions is challenging. Moreover, the complex interactions between the the physical environment and signal propagation, and the dynamic nature of wireless channels due to interference and fading can make it difficult to interpret and explain results. Finally, wireless deployments are very diverse, making it difficult to generalize results from one or even a handful of locations to the many environments that may be encountered in practice.

Not surprising, many researchers view simulation as an attractive alternative. Simulators such as ns-2 [34] and OPNET [38] are widely used but it has been observed that the lack of physical layer accuracy can lead to the incorrect conclusions [26]. This is especially true if the behavior and performance of the protocol or system feature under study is very sensitive to physical layer effects. To improve accuracy, researchers have developed a variety of evaluation platforms that combine simulation with real world platforms or measurements. Examples include hybrid solutions [50]. analog channel emulation [16], and partially controllable wireless testbeds [49, 44, 9]. These approaches are useful for some types of experiments, but they have limitations such as limited physical layer control or repeatability, limited flexibility, or high complexity.

We have developed a wireless networking testbed based on signal propagation emulation [18, 17]. It achieves a high level of realism since it uses real wireless devices but it also provides full control and repeatability at the physical layer. This makes it a very attractive platform for the evaluation and testing of various wireless networking technologies. The emulator testbed is accessible over the Internet and has been in regular use by researchers at CMU and elsewhere since 2007. This paper reports our experience in using the emulator testbed. We make contributions in three areas. First, we report the results of a set of low level measurements that characterize the link-level behavior of commercial WiFi cards. These results illustrate the capabilities of the emulator, and extend earlier work [21]. Second, we compare some emulator testbed results with simulation. Finally, we use the experience gained from some of projects that have used the emulator testbed to explore what types of experiments can most benefit from using the emulator testbed, as opposed to other platforms such as simulators or in-the-wild testbeds.

The remainder of this paper is organized as follows. In the next section we give an overview of the wireless network emulator testbed. We present related work in Section 3. We use the emulator to characterize wireless channels under diverse conditions and to compare emulator with simulation results in Section 4. Section 5 identifies several classes of emulator experiments and discusses how they relate to other wireless evaluation platforms. Finally, we summarize our results in Section 6.

## 2. EMULATOR OVERVIEW

The wireless network emulator testbed [18] is based on the idea of signal propagation emulation, as is illustrated in Figure 1. Wireless devices are connected to the emulator testbed using a cable attached to the antenna port of their wireless network card. The RF signal transmitted by each device is passed to a signal conversion module where it is mixed to an intermediate frequency, digitized, and forwarded to an FPGA. The FPGA emulates the effects of signal propagation (e.g. attenuation, fading, multi-path, and interference). Finally, for each device, the FPGA combines the processed signals from the other devices into an output signal and sends it to a signal conversion module, where it is converted into an RF signal and forwarded to the wireless card through its antenna port.

All analog components in the emulator testbed (wireless devices, signal conversion modules) are shielded from each other and from the environment (dotted boxes in Figure 1) so that no communication occurs over the air. As a result, devices only communicate through the emulator, so we have full control over the signal propagation environment. The wireless channels between the devices

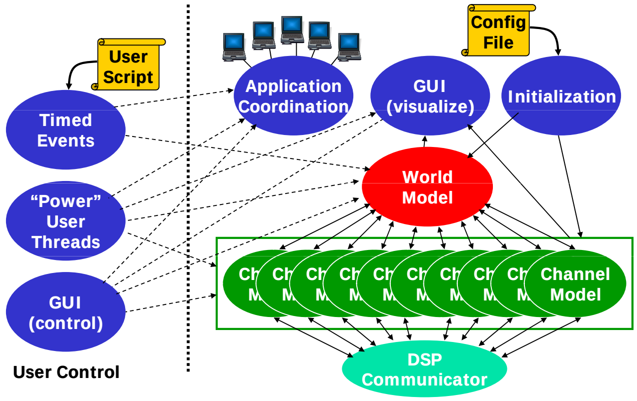


Figure 2: Emulator Control Software

are the only simulated component of the system and channels are modeled at the signal level. The wireless hardware, including signal generation and reception, and the software on the end hosts are all real, resulting in a very high degree of realism.

Wireless experiments can be controlled in a number of ways from the Emulation Controller, a Linux server. The most commonly use approach is to use an emulated physical environment that models the movement of the wireless devices (World Model in Figure 2) and controls the execution of applications on the end nodes. It coordinates the movement of devices with the modeling of the signal propagation in the FPGA by modifying the channel parameters in real time (Channel Models). For example, when nodes move in the emulated physical environment, the emulation controller adjusts the parameters for the large-scale attenuation and fine grain fading of the various channels based on the speed and location of the devices and it sends the new channel parameters to the FPGA. Alternatively, users can bypass the world model and directly control the wireless channels, e.g. to create very simple controlled experiments.

Users can control wireless experiments using three interfaces [19]. First, users can control key features of the experiment such as node placement in the emulated physical space and channel parameters using an interactive GUI that runs as an applet in a web browser. Second, the emulator offers a simple scripting interface. The scripting language is very simple, e.g. there is no control flow, and it is mostly useful for first-time users. Finally, users can build arbitrarily complex experiments using Java. This programmatic interface offers direct access to all emulator features and provides tight control over the timing of experiments.

A wireless emulator testbed has been in regular use by both CMU and external users since early 2007. It supports the full 2.4 GHz ISM band and has 15 nodes. The nodes are laptops with 802.11b interfaces based on the Atheros chipset. More recently, we have added support for 802.11g experiments, and we have also attached software-defined radios (USRP and GNU Radio) and Bluetooth devices to the emulator. Initially, we used custom software for resource allocation and management of the emulator devices. In 2008, we switched to using the Utah Emulab software, which offers a number of advantages, including space sharing of the emulator hardware and support for custom OS images. The emulator testbed is currently part of the GENI development effort as one of the projects using the ProtoGeni control framework [15].

## 3. RELATED WORK

"Testbeds in the wild" are the best way to perform realistic wireless networking experiments since both the wireless devices and signal propagation environment are real. Such testbeds offer a very high degree of realism, but they also have some disadvantages. First, the results of experiments are not repeatable. Movements by people and objects change the physical environment and other wireless devices such as production networks interfere with experiments, making it nearly impossible to reproduce results. Second, these same effects may make it difficult to interpret testbed results. Finally, control of experiments, especially those involving mobile devices, is cumbersome.

Because of the challenges associated with open testbeds, a number of research groups have developed testbeds that offer some control over the signal propagation environment. Examples include WHYNET [50], Emulab [49], Orbit [39], MiNT [10, 9] and others [14, 25]. The approaches are useful for certain types of experiments, but they do not address all challenges. For example, the range of modeled environments and the diversity of supported experiments tends to be limited and isolation is often a problem. The emulator testbed is more flexible (can be used for very diverse experiments) and provides a higher degree of isolation and realism, although its scale is more limited compared with some of the other approaches.

Simulators are an attractive alternative to testbeds. They offer a high degree of control and repeatability, and are relatively easy to use. However, wireless simulators must recreate all aspects of the system, including all layers of the protocol stack as well as the physical environment. In practice, this requires making many simplifications. For example, fundamental features such as how received frames are interpreted [47] and how the MAC layer is affected by the physical layer (e.g. packet capture) are often approximated, if they are modeled at all. ns-2 [37] is very widely used because it is open source, but studies have shown that problems with both the physical and 802.11 layer modeling can result in incorrect results [47, 26, 6, 4]. QualNet [43] and OpNet [38] have better physical and MAC layer accuracy, but they are more difficult to extend. Simulators can be useful in the early phases of research projects, since protocol implementations are not dependent on hardware features and it is easy to run large scale experiments. However, testbeds are needed to achieve higher degrees of realism, especially for experiments that are sensitive to physical layer effects.

The problems with simulators and testbeds have motivated researchers to combine simulation and real systems, giving users the the flexibility to optimize the type of modeling that is used for different parts of the system. However, it is still difficult to simultaneously achieve high levels of realism and control (e.g. to sample a large space of propagation environments) as we can do with the emulator. Another hybrid approach is to collect packet traces and to replay them either over wired networks [35, 36, 27, 13] or over a more-or-less errorless wireless network [11]. This improves the repeatability of wireless networking experiments, but it only captures the high-level error characteristics of the wireless environment, so these approaches are only appropriate for high-level protocol and application-level experiments.

A number of groups have explored hardware-based solutions. A first example is RAMON [16], which uses three programmable attenuators. While it is possible to change the attenuation, it is not possible to control pairwise interaction between wireless hosts or implement more complex functions such as multipath emulation. [24] describes a similar approach. Hardware-based fading simulators using analog or digital components [2, 5, 8, 45] can be used, but these have traditionally been point-to-point simulators. Commercial channel emulators also fall in this category [12]. While very sophisticated, this equipment is expensive and can only model a small number of channels, so it is not a *network* testbed. In contrast, the emulator can model hundreds of interfering channels.

## 4. CHARACTERIZATION

The main goal of the emulator is to accurately reproduce the behavior of a wireless network in a controlled environment. As described in Section 2 only the communication channel is simulated. Therefore, if the emulator can faithfully reproduce the wireless channel, the performance of an emulated wireless network will be accurate. Wireless channels are characterized by three main components: path loss, fading, and multipath. In this section we show that the emulator can generate a wide range of wireless channels and these channels can be used to characterize the behavior of network radios. We also compare the behavior of wireless radios in the emulator with that of radios in a network simulator, specifically ns-2. While more accurate simulators are available, we picked ns-2 because it has been the most popular simulator in academia, so there are numerous published results based on ns-2.

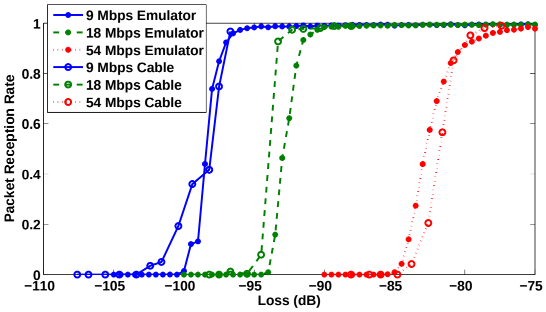


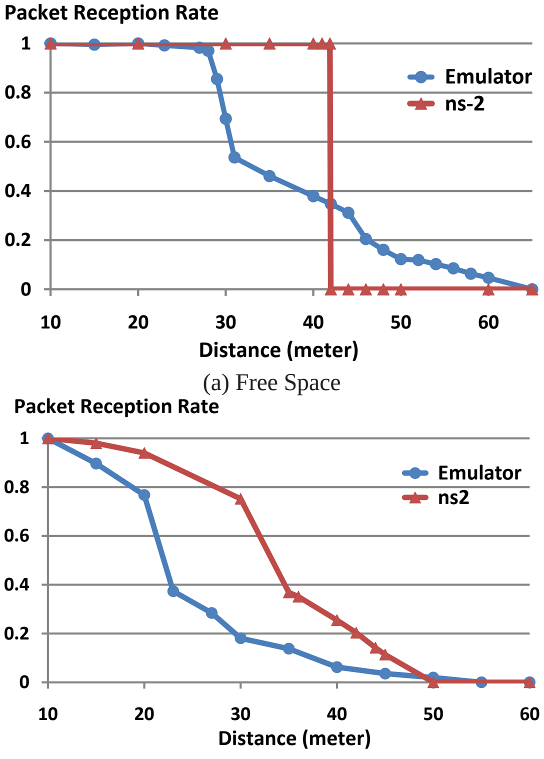
Figure 3: Performance of 802.11g with Different Rates in the Emulator and a Loss-Controlled Cable

### 4.1 Path Loss Performance

One of the most fundamental descriptions of a wireless channel is the log-distance model where the channel loss is proportional to the log of the distance between radios. Because radio distances can vary greatly, the emulator output must have a wide dynamic range. It was shown in [3] that the emulator can produce channel losses from 55 dB to 117 dB with 0.1 dB resolution. A loss of 55 dB corresponds to a distance of about 10 m in free space and is well within the range where two 802.11g radios should communicate reliably at the highest rate. A loss of 117 dB attenuates wireless signals transmitted at 20 dBm to below the noise floor where communication is not achievable even at the lowest rates.

Not only must the emulator simulate the channel losses accurately but it must also preserve the integrity of the transmitted signals. The process of mixing and sampling will add some distortion to the transmitted signals [20]. This distortion is only a concern if the performance of the radios in the emulator is not consistent with the performance of the same radios communicating through the air. The radios in the emulator system have the Atheros 5212 chipset with a patched version of the madwifi driver version 0.9.4. The madwifi driver has been modified similar to [48] to allow better reception of packets at low signal strengths. The performance of Atheros cards in the emulator was tested by transmitting 1000 UDP broadcast packets from one radio to another and recording how many packets were received. The test was performed over various losses and rates. The same tests were performed with the radios' antenna ports connected through a cable with a controlled attenuation. The results of these tests are shown in Figure 3. The packet reception rate versus path loss is about the same when the radios are connected to the emulator as when they are connected by the cable. Similar results for 802.11b rates are presented in [3]. Not only do the packet reception rates in the emulator match closely with the cable test but the RSSI and noise floor readings on the cards are consistent between the emulator and the cable test. The path loss experiment has shown that any distortion caused by the emulator is not severe enough to change the behavior of the radios connected to the emulator. This experiment also shows that the emulator can accurately reproduce the effect of path loss on a wireless network.

The emulator software can control the path loss between a pair of radios with adjustable path loss exponent *n*. The same log-distance path loss model is available in ns-2 [37]. However, the impact of path loss on radio performance is quite different between the emulator and the simulator. The TCP throughput between two 802.11b radios over a range of distances is shown in Figure 4(a). The path loss exponent *n* is set to 3 and we present results for both the emulator and ns-2. Since the packet reception success in the simulator is completely determined by signal strength, the simulation of path loss in ns-2 leads to an artificial 'brick wall' at a certain distance. In the emulator, the path loss is directly applied to the wireless signal in the emulator and it relies on real radios for transmitting and receiving packets. As a result, the packet reception rate has a gradual transition from a good region to a disconnected region, similar to Figure 3.



(a) Free Space

(b) Fading (K=3dB)

Figure 4: Throughput Tests in Emulator and Simulator

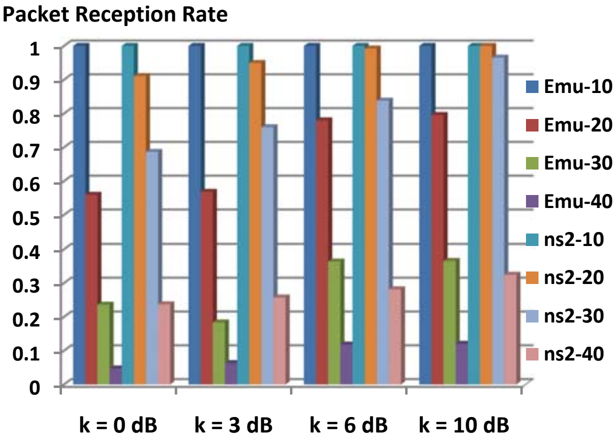


Figure 5: Throughput for variable K in Emulator and Simulator

### 4.2 Fading Performance

Fading is a wireless channel phenomenon where the received signal strength fluctuates rapidly with motion in a rich scattering environment. The fluctuations are commonly modeled statistically with defined distributions and autocorrelations. The Ricean fading model characterizes the propagation property when a received signal includes one dominant line-of-sight signal (usually carrying more power) and other scattered signals. The fading follows Ricean distribution with factor $K$, which is defined as the ratio of the power of dominant signal over the power of the scattered signals. Rayleigh fading is a special case when $K$ equals to 0, i.e. there is no line-of-sight or dominant signal.

To study the impact of fading on radio performance, the throughput versus distance measurement is performed using the Ricean fading model implemented in both the emulator and ns-2 [42] respectively. The same log-distance model ($n$=3) is used for path loss, and the $K$-factor is set to 3dB. As shown in Figure 4(b), the throughput decreases when Ricean fading is introduced. The emulator curve from Figure 4(a) shifts towards left in Figure 4(b). The ns-2 curve is now also more realistic, although it still overestimates the packet reception rate. It is clearly possible to improve the match between the simulator an emulator results by shifting the ns-2 curve by about 10 meters.

A low $K$-factor value implies that the random fluctuations from the scattered signals are more dominant than the line-of-sight component resulting in steeper fluctuations in channel loss. Figure 5 shows the throughput measurement results with various $K$-factors at distance 10m, 20m, 30m, and 40m respectively. The throughput value is normalized to the case with $K$ of 0 and a distance of 10 meters. Both the emulator tests and the ns-2 simulations show that Ricean fading with lower $K$-factor leads to lower throughput.

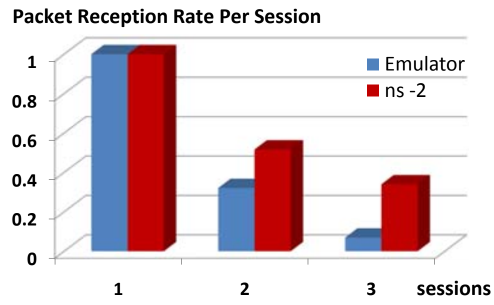


Figure 6: Static Throughput Test in Emulator and Simulator

### 4.3 Simple Network Scenarios

Wireless networking experiments typically include multiple radios in communication range, so it becomes important to accurately model carrier sense and the effects of interference. Figure 6 shows an average TCP throughput obtained in a simple network scenario. The number of active sessions increases from 1 (no other active radios) to 3. All radios are stationary, with RTS/CTS disabled, and the packet reception rate is normalized to the case when there are no other active radios in range. As expected, the results in Figure 6 show that the packet delivery rate decreases as the number of competing transmitters increases. Higher values from ns-2 again shows that the packet reception model in ns-2 tends to overestimate received signal strengths, or underestimate the effect of interference from other nodes. Since the emulator testbed uses real radios, carrier sense and the effects of real, not simulated, the results should be very accurate.

Many wireless experiments need to account for mobility which introduces additional dynamics in channel quality. A simple TCP throughput result for a mobile scenario is shown in Figure 7. A TCP connection is set up between a pair of radios, with one radio moving along a predefined route at the speed of 0.5 m/s. We use the same channel parameters from the fading test, and Auto Rate Fallback (ARF) is enabled for both the emulator and simulator experiments. Values are normalized to the highest one during the whole trip. While the results are fairly similar during the first part of the experiment, they diverge significantly during the second part. This could be caused by differences in the physical or datalink layer of the simulated radios.

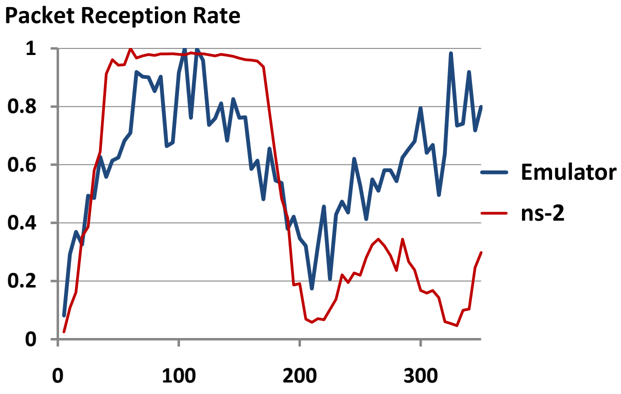


Figure 7: Mobile Throughput Test in Emulator and Simulator

### 4.4 Multipath Performance

Multipath causes a distinct problem for digital communications when the delay spread of the channel is larger than the symbol period of communication system. In this case delayed symbols interfere with each other at the receiver resulting in inter-symbol interference (ISI). A good equalizer can improve system performance in channels with ISI. However, equalizers will only correct ISI up to a maximum delay spread. It was shown in [21] that at 11Mbps, equalization failed with a delay spread greater than 400ns. The standard OFDM equalizer cannot remove interference with a delay longer than the guard interval which is 800 ns for 802.11a/g. Therefore, the performance of 802.11a/g should drop dramatically when the channel has more than 800 ns of delay spread.

To test how well the 802.11g radios work in multipath channels, a simple two path experiment was set up in the emulator. One path was held at a constant 70 dB loss while the other path had a delay varying from 0 to 1.38 us in 16.6 ns steps and a loss varying from 82 dB to 70 dB in 0.2 dB increments. The results are shown in Figure 8. The graphs show the packet reception rate (as a gray scale) as a function of the delay spread (x-axis) and the difference in path loss between the two paths (y-axis), for different transmit rates. As expected, the packet reception rate drops after 800 ns for all OFDM rates. Also, the packet reception rate in multipath is worse for higher rates than lower rates. There are a large number of dropped packets at higher rates with delays lower than 800 ns. In particular, there is a consistent drop in packet reception when the path has a delay of 230 ns. This drop was seen at rates above 18 Msps on different channels and different nodes. We believe that the drop at 230 ns is an artifact of the equalizers in the radios but more investigation is required to identify the problem.

In a normal office indoor environment a maximum delay spread around 60ns is expected. The radios under test perform admirably well in that range. However in large factories and stadiums the delay spread can reach above 200 ns. In these environments the high 802.11g data rates will have reduced performance due to multipath. In outdoor environments the delay spread can be over 1000 ns which can cause the performance of 802.11g to drop for all rates. These tests results show that even with a delay spread less than the guard interval, multipath may still degrade system performance.

This experiment demonstrated the capability of the emulator identify performance issues in wireless network cards. It is very difficult to control a multipath channel in a real-world wireless testbed. Only expensive channel simulators have the capability to simulate multipath channels to a fine resolution.

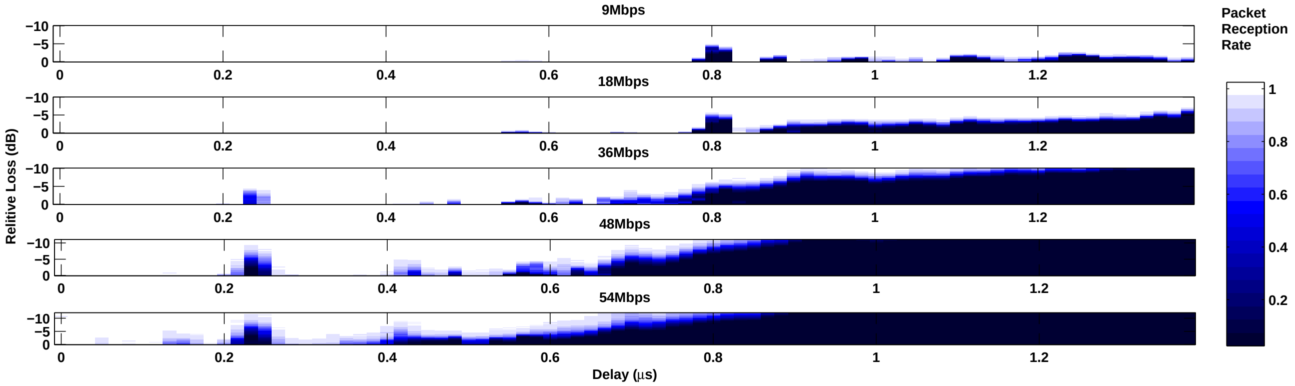


Figure 8: Packet Reception Rates with a Two Path Channel

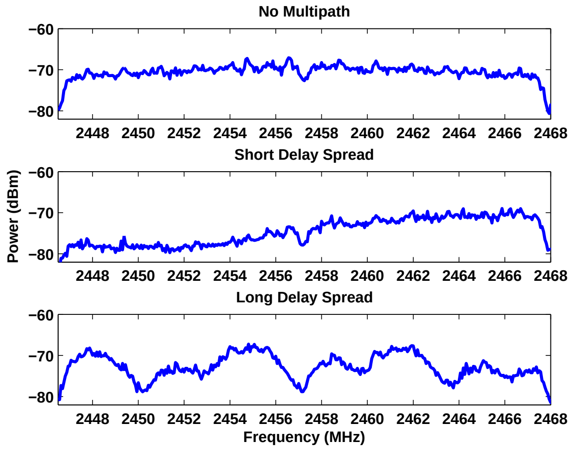


Figure 9: Spectrum of OFDM Signals in Channels with Varying Delay Spread

### 4.5 Multipath Fading

A channel with both multipath and fading distorts the spectrum of the transmitted signal. This effect was captured by monitoring the output of the emulator with a spectrum analyzer. Figure 9 shows the output of an OFDM waveform under three separate channel conditions: no multipath, short delay spread and long delay spread. The short delay spread has paths with delays at 0 ns, 28 ns, and 39 ns. The long delay spread has paths with delays at 0 ns, 183 ns, and 361 ns. In the last two cases, we are seeing frequency selective fading. Note that the long delay spread causes rapid fluctuations in the spectrum.

When there is motion in such a channel each subcarrier component will experience fading independently. Therefore, not only with the received spectrum change amplitude due to fading but the shape of the spectrum will also change over time. The emulator can generate multipath fading environments with independent fading tables for each path in a channel. The spectral fluctuations are shown in Figures 10, 11 and 12. The fluctuations needed to be slowed down so they could be recorded by the spectrum analyzer. Normally the fluctuations occur about 100x faster.

### 4.6 Future Expansions

The emulator can model path loss, multipath and fading accurately. However, the emulator does not precisely control the phase of the wireless signals. A new version of the emulator will downconvert the transmitted signals to baseband where the phase of the signal can be controlled. Channel phase control is important in radios with antenna arrays such as 802.11n. Secondly the new system will have better support for multipath channels. Currently only 8 channels in the system are equipped with multipath. A new channel model structure will make better use of the FPGA resources and expand multipath to all channels.

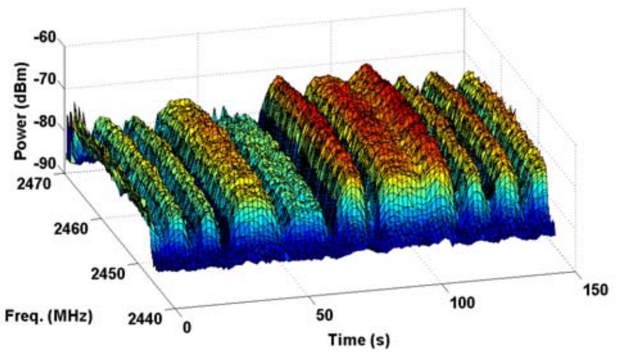


Figure 10: Spectrum of OFDM Signals in a Fading Channel with no Multipath

## 5. EXPERIENCE USING THE EMULATOR

Many research groups at CMU and elsewhere have used the emulator testbed and a growing number of publications include emulator measurements. These results suggest that there are several classes of experiments for which the emulator testbed is particularly well suited. We review these classes in this section.

### 5.1 Optimization and Evaluation of Wireless Protocols

The original motivation for the emulator testbed was to support controlled and repeatable experiments of new wireless protocols. This makes it possible to compare the performance of different protocols executing on real wireless devices under identical signal propagation conditions.

An example is a recent project that developed a protocol for opportunistic relaying of packets (PRO) [33, 30]. PRO is a link-layer protocol that allows nodes that overhear a failed packet transmission to retransmit the packet on behalf of the source, if they have better connectivity to the destination. PRO can improve channel efficiency by reducing the number of retransmissions and the use of higher transmit rates. PRO is a fully distributed protocol that relies on the existing 802.11 backoff mechanism to arbitrate channel access for

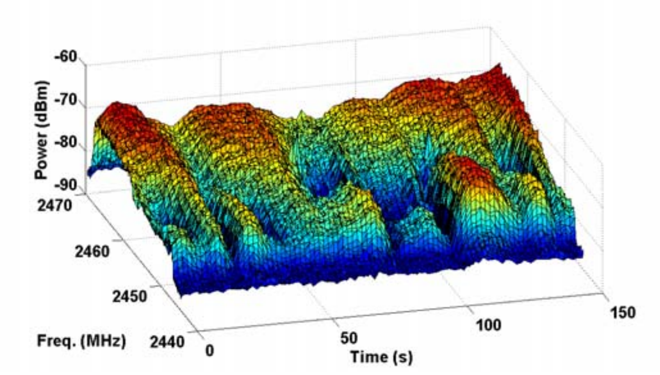


Figure 11: Spectrum of OFDM Signals in a Fading Channel with Short Delay Spread

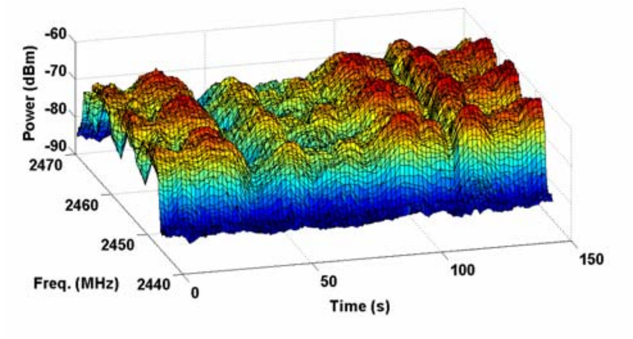


Figure 12: Spectrum of OFDM Signals in a Fading Channel with long Delay Spread

retransmissions. The emulator testbed was used to compare the performance of PRO with standard 802.11 and static 2-hop forwarding protocols under very diverse conditions, e.g. diverse node topology, types of fading, levels of competing traffic, and for mobile devices. We found that the performance benefits of PRO generally increase with increased levels of fading and competing traffic. Moreover, PRO can improve performance for mobile users, e.g. users using a mobile device while walking on campus.

PRO was also evaluated in testbeds deployed in two different campus buildings. What is interesting is that the results obtained on the emulator qualitatively match the results collected in real world environments. For example, the performance benefits of PRO were much higher in the busy student center compared with the relative static office building, and the benefits were also higher during the day than at night, when the wireless network is largely idle. Finally, both the emulator and real-world measurements showed that for very poor channels with near zero packet success rates, PRO can improve performance considerably, effectively extending the range of the wireless LAN.

The emulator testbed has similarly been used to evaluate rate adaptation algorithms [23], 802.11 back-off strategies [32], time-aware retransmission [31], and 802.11 handoff performance [22]. Figur 13 shows an example result taken from [22]. It compares three handoff protocols for a scenario involving a mobile users moving along a circuit that passes by seven access points.

### 5.2 Device and Link Characterization

The emulator is an ideal platform for characterizing the properties of wireless links, both with respect to different wireless hardware (e.g. from different vendors) and different channel conditions (e.g. multipath, fading, ..). Such studies typically include a small number of nodes placed in simple topologies so that results can be interpreted easily. The emulator testbed can be easily programmed to exhaustively measure a wide range of parameter values, e.g. levels of attenuation, fading parameters, etc. The results in Section 4 provide some examples, and a more detailed study for 802.11b can be found in [21]. Figure 14 shows some example results exploring off-channel reception for 802.11b. We see that off-channel reception works fairly well for offsets up to two channels at a 1 Mbs rate, but it does not work at all at 11 Mbp. The results of such studies can also be used to improve accuracy of simulators.

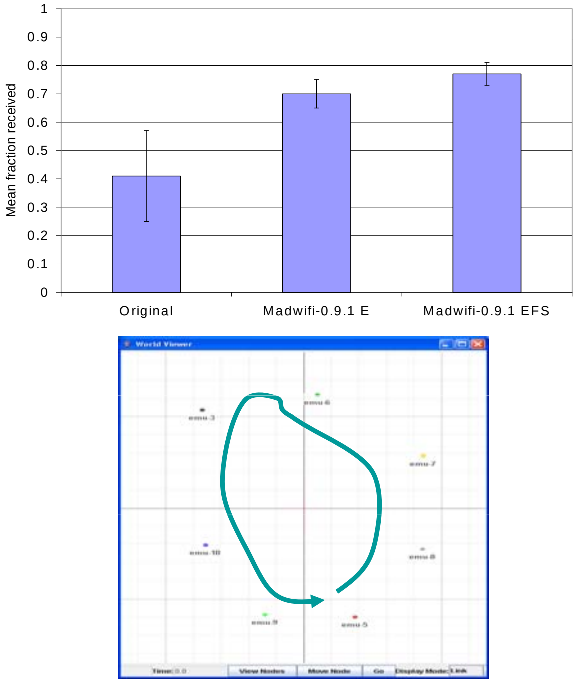


Figure 13: Handoff Comparison

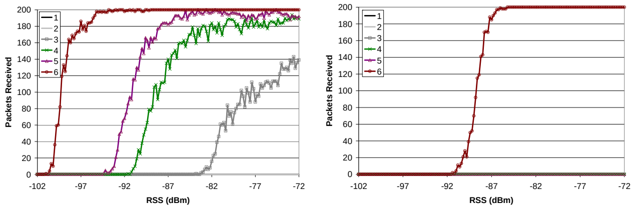


Figure 14: Off-channel Reception at 1 Mbs (left) and 11 Mbs (right)

Controlled studies can also be used in the early stages of a project to provide insight into how physical layer effects influence packet reception and protocols. The emulator was used this way by several research projects including interference-aware transmit power selection [29], interference-aware routing in ad hoc networks [41, 40], channel width adaptation [7], and identifying WiFi, Bluetooth and other devices for and diagnostic purposes [28].

### 5.3 Analyzing Real World Behavior

Understanding results obtained in in-the-wild testbeds can be difficult because it is often difficult to determine what factors are responsible for a particular observation. Controlled experiments can help. For example, an early version of the emulator was used to help explain the lack of correlation between RSSI and packet success rate observed in Roofnet [1]. After eliminating a number of candidate factors, it was determined that multi-path was the likely cause. This was done by systematically analyzing various multipath configurations on the emulator. The results showed that the wireless cards were designed for indoor use and can only recover from limited delay spreads. Discussion with the card manufacturer

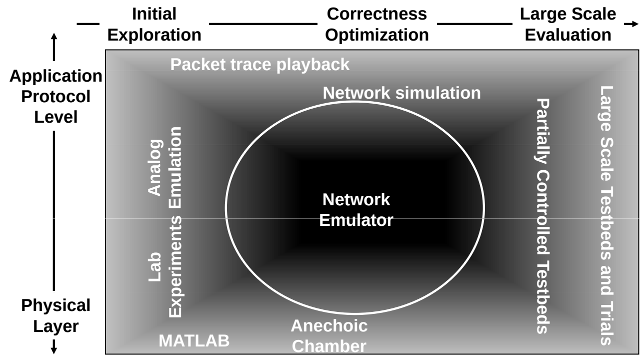


Figure 15: Spectrum of Evaluation Techniques

confirmed this [21]. Other groups propose to use the emulator help explain results in vehicular networks.

### 5.4 Use of Emulator in Education

The wireless network emulator has been used to support assignments and projects in several courses [46]. The assignments are structured exercises that show have various physical layer effects, such as attenuation, interference, and fading, impact wireless network performance. A second set of assignment evaluates the effectiveness of MAC layer features such as transmit rate adaptation, RTS/CTS, and channel selection. Projects are more open ended and can, for example, compare the performance of different handoff strategies or TCP-over-wireless approaches.

The emulator testbed is an attractive platform for course work. First, students find it more interesting to work with than simulation, and it is of course more realistic. Compared with open testbeds, the ability to control the channels and repeat experiments simplified the preparation of assignments and explaining the results. Moreover, multiple students can run experiments at the same time without interfering with each other. Last but not least, the emulator testbed is readily available, so the course staff does not need to put effort in deploying and maintaining the infrastructure.

### 5.5 Comparison of Evaluation Tools

The large number of wireless experimental options raises the question of when different types of tools are most appropriate. While there is no consensus on this issue within the community, we can certainly identify some rough guidelines. Figure 15 shows the various stages that research (or development) projects go through. Projects start (on the left) with the initial development and evaluation of ideas and end (on the right) with fully realistic testing at full scale. Intermediate stages may include verifying protocol correctness, prototyping, and iterative optimization. The vertical axis captures the nature of the effort. Projects at the bottom focus on the physical layer and are largely independent of what protocols or applications are used (e.g. MIMO link optimization). Projects at the top develop protocols or application-level mechanisms that are largely insensitive to the details of the physical layer (e.g. video streaming). In between, projects combine physical layer and protocol elements (e.g. rate selection, MAC protocols for MIMO). While Figure 15 clearly oversimplifies a very complex process, it does allow us to identify some areas where specific tools are most useful. We show some tools, several of which were discussed in Section 3, in the figure.

Our experience so far indicates that the emulator testbed is most useful in the center of Figure 15 (white elipse marking darker region). Experiments such as link characterization and course assignments fall in the left of the elipse, while performance evaluation of protocols and explaining real-world behavior fall towards the right. The emulator is also largely complementary to other technologies. On the far left, techniques such as simulation are easier to use while on the far right, only large-scale real-world testing will work. At the top or bottom, techniques that are optimized for protocol or physical layer testing will suffice. However, in the center region, the unique capabilities of the emulator, namely supporting experiments that are simultaneously realistic, fully controllable, repeatable and diverse, are critical.

## 6. CONCLUSION

This paper presented a wireless network testbed capable of producing accurate and repeatable experiments. The benefit of the emulator comes from its ability to connect real radio hardware through a controlled simulated channel. The channels produced by the emulator were shown to be accurate compared to an analog channel. Also experiments that used the emulator were shown to be more accurate than experiments based on a simulator. The emulator can also produce fine resolution multipath channels that are difficult to control in real-world testbeds. Since the emulator became available in 2007 many research groups have sought this system for its flexibility control and realism.

### Acknowledgements

This research was funded in part by NSF under award numbers CCR-0205266 and CNS-0434824. Additional support was provided by Intel and Xilinx.

## REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level Measurements from an 802.11b Mesh Network. In *Proceedings of the ACM SIGCOMM Conference on Network Architectures and Protocols*, Portland, August 2004. ACM.

[2] G. Arredondo, W. H. Chriss, and E. H. Walker. A multipath fading simulator for mobile radio. *IEEE Trans. On Vehicular Tech.*, VT-22(4):241–244, November 1973.

[3] K. Borries, G. Judd, D. Stancil, and P. Steenkiste. FPGA-Based Channel Simulator for a Wireless Network Emulator. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pages 1–5, April 2009.

[4] J.-Y. L. Boudec and M. Vojnovic. Perfect Simulation and Stationarity of a Class of Mobility Models. In *IEEE INFOCOM*, 2005.

[5] E. Casas and C. Leung. A Simple Digital Fading Simulator for Mobile Radio. *IEEE Transactions on Vehicular Technology*, 39(3):205–212, August 1990.

[6] D. Cavin, Y. Sasson, and A. Schiper. On the accuracy of MANET simulators. In *Workshop on Principles of Mobile Computing (POMC'02)*, pages 38–43. ACM, October 2002.

[7] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and V. Bahl. A Case for Adapting Width in Wireless Networks. In *Proceedings of the ACM SIGCOMM Conference on Network Architectures and Protocols (SIGCOMM08)*, August 2008.

[8] R. A. Comroe. All-digital Rayleigh fading simulator. In *Proceedings of National Electronics Conference*, pages 136–139, 1978.

[9] P. De, R. Krishnan, A. Raniwala, K. Tatavarthi, N. Syed, J. Modi, and T. Chiueh. MiNT-m: An Autonomous Mobile Wireless Experimentation Platform. In *Proc. of Mobisys 2006*, Uppsala, Sweden, June 2006.

[10] P. De, A. Raniwala, S. Sharma, and T. Chiueh. MiNT: A Miniaturized Network Testbed for Mobile Wireless Research. In *Proc. of Infocom 2005*, March 2005.

[11] D. Eckhardt and P. Steenkiste. A Trace-based Evaluation of Adaptive Error Correction for a Wireless Local Area Network. *Mobile Networks and Applications (MONET)*, 1998. Special Issue on Adaptive Mobile Networking and Computing.

[12] Elektrobit. http://www.elektrobit.com.

[13] K. Fall. Network Emulation in the Vint/NS Simulator. In *IEEE ISCC'99*. IEEE, 1999.

[14] J. Flynn, H. Tewari, and D. O'Mahony. Jemu: A Wireless Network Emulator for Mobile Ad Hoc Networks. In *First Joint IEI/IEE Symposium on Telecommunications Systems Research*, Dublin, Ireland, November 2001.

[15] GENI - Global Environment for Network Innovations. http://groups.geni.net/geni/wiki/CmuLab.

[16] E. Hernandez and S. Helal. RAMON: Rapid Mobility Network Emulator. In *Proc. of the 27th IEEE Conference on Local Computer Networks (LCN'02)*, Tampa, FL, November 2002. IEEE.

[17] G. Judd. Repeatable and realistic wireless experimentation through physical emulation, 2006. PhD thesis, Department of Computer Science, Carnegie Mellon University. Also availabel as Technical report CMU-CS-06-164.

[18] G. Judd and P. Steenkiste. Using Emulation to Understand and Improve Wireless Networks and Applications. In *Proceedings of NSDI 2005*, Boston, MA, May 2005.

[19] G. Judd and P. Steenkiste. A software architecture for physical layer wireless network emulation. In *The First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH 2006)*. ACM, September 2006.

[20] G. Judd and P. Steenkiste. Design and Implementation of an RF Front End for Physical Layer Wireless Network Emulation. In *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pages 974–979. IEEE, April 2007.

[21] G. Judd and P. Steenkiste. Characterizing 802.11 Wireless Link Behavior. *Wireless Networks (WINET) Journal*, 2008. to appear.

[22] G. Judd, X. Wang, M.-H. Lu, and P. Steenkiste. Using Physical Layer Emulation to Optimize and Evaluate Mobile and Wireless Systems. In *5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous 2008)*, July 2008.

[23] G. Judd, X. Wang, and P. Steenkiste. Efficient Channel-aware Rate Adaptation in Dynamic Environments. In *The Sixth International Conference on Mobile Systems, Applications, and Services (MobiSys 2008)*, June 2008.

[24] J. T. Kaba and D. R. Raichle. Testbed on a Desktop: Strategies and Techniques to Support Multi-hop MANET Routing Protocol Development. In *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'01)*. ACM, October 2001.

[25] Q. Ke, D. A. Maltz, and D. B. Johnson. Emulation of Multi-Hop Wireless Ad Hoc Networks. In *Seventh International Workshop on Mobile Multimedia Communications*, October 2000.

[26] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. In *Proceedings of MSWiM 2004*, Venice, Italy, October 2004.

[27] F. Kubinszky, Z. Lzr, A. Varga, B. Eged, and A. Telegdy. Emulation of Ad-hoc Networks on IEEE 802.11. In *MIKON 2000 Conference*, Wroclaw, Poland, May 2000.

[28] K. Lakshminarayanan, S. Sapra, S. Seshan, and P. Steenkiste. RFDump: An Architecture for Monitoring the Wireless Ether. In *CoNEXT '09: Proceedings of the 5th ACM international conference on emerging Networking EXperiments and Technologies*, New York, NY, USA, 2009. ACM.

[29] X. Liu, S. Seshan, and P. Steenkiste. Interference-Aware Power Management for Dense Wireless Networks. In *Proceedings of the Annual Conference of ITA*, September 2007.

[30] M.-H. Lu. Optimizing Wireless Transmission for Video Streaming, June 2009. PhD Thesis, Department of Electrical Engineering, CMU, Pittsburgh, PA.

[31] M.-H. Lu, P. Steenkiste, and T. Chen. On A Time-based Adaptive Retry Strategy for Video Streaming in 802.11 WLANs. *Wireless Communications and Mobile Computing*, 2007.

[32] M.-H. Lu, P. Steenkiste, and T. Chen. Using Commodity Hardware Platform to Develop and Evaluate CSMA Protocols. In *The Third ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH 2008)*, September 2008. held in conjunction with ACM MobiCom 2008.

[33] M.-H. Lu, P. Steenkiste, and T. Chen. Design, implementation and evaluation of an efficient opportunistic retransmission protocol. In *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 73–84, New York, NY, USA, 2009. ACM.

[34] S. McCanne and S. Floyd. UCB/LBNL/VINT Network Simulator - ns (version 2), April 1999.

[35] G. Nguyen, R. Katz, B. Noble, and M. Satyanarayanan. A trace-based approach for modelling wireless channel behavior. In *Proc. of the Winter Simulation Conf.*, San Diego, CA, December 1996. IEEE.

[36] B. Noble, M. Satyanarayanan, G. Nguyen, and R. Katz. Trace-based Mobile Network Emulation. In *SIGCOMM '97*, pages 51–62, Cannes, September 1997. ACM.

[37] Ns2. http://www.isi.edu/nsnam/ns.

[38] Opnet. http://www.opnet.com.

[39] ORBIT - Wireless Network Testbed. http://www.winlab.rutgers.edu/pub/docs/focus/ORBIT.html.

[40] G. Parissidis. Interference-aware Routing in Wireless Multi-Hop Networks, May 2008. PhD Thesis, Department of Electrical Engineering, ETH, Zurich, Switserland.

[41] G. Parissidis, M. Karaliopoulos, M. May, T. Spyropoulos, and B. Plattner. Interference in Ad Hoc Wireless Networks: A Model and its Experimental Evaluation. In *Mobile and Wireless Networks (WoWMoM 2008)*, June 2008.

[42] R. J. Punnoose, P. V. Nikitin, and D. D. Stancil. Efficient simulation of rician fading within a packet simulator. In *IEEE Conf. On Vehicular Technology*, Boston, September 2000. IEEE.

[43] Qualnet. http://www.scalable-networks.com.

[44] 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. of WCNC 2005.*, New Orleans, LA, March 2005.

[45] J. I. Smith. A Computer generated multipath fading simulation for mobile radio. *IEEE Transactions on Vehicular Technology*, 24(3):39–40, August 1975.

[46] P. Steenkiste. A hands-on introduction to wireless networking. Undergraduate course taught at CMU, http://www.cs.cmu.edu/ prs/wireless08/.

[47] M. Takai, J. Martin, and R. Bagrodia. Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks. In *MobiHoc 2001*. ACM, October 2001.

[48] I. Tinnirello, D. Giustiniano, L. Scalia, and G. Bianchi. On the side-effects of proprietary solutions for fading and interference mitigation in ieee 802.11b/g outdoor links. *Comput. Netw.*, 53(2):141–152, 2009.

[49] B. White, J. Lepreau, and S. Guruprasad. Lowering the Barrier to Wireless and Mobile Experimentation. In *First Workshop on Hot Topics in Networks (HotNets-I)*. ACM, October 2002.

[50] WHYNET: Scalable Testbed for Next Generation Mobile Wireless Networking Technologies. http://pcl.cs.ucla.edu/projects/whynet/.