The Use of a Controlled Wireless Testbed in Courses

Peter Steenkiste
Departments of CS and ECE
Carnegie Mellon University
Pittsburgh, PA, USA
prs@cs.cmu.edu

ABSTRACT

Wireless networking has become a popular topic in both undergraduate and graduate courses. However, putting together good assignments in wireless networking is difficult because the behavior of the wireless network depends strongly on the physical environment. We have used a wireless networking testbed based on signal propagation emulation in a number of wireless networking courses. The wireless emulator supports highly realistic experiments, while also offering a high degree of control and repeatability. This combination is very useful in a teaching context. In this paper we give an overview of the wireless emulator and we describe how it was used to support assignments and open-ended projects in several courses.

Categories and Subject Descriptors

K.3.1 [**Computers and Education**]: Computer Use in Education

General Terms

Experimentation

Keywords

Wireless networking, education

1. INTRODUCTION

The rapid growth in wireless networking is generating a lot of interest in the topic and as a result, many institutions now offer wireless networking courses at the undergraduate and graduate level. Developing a good wireless networking course involves two challenges. A first challenge is that the topic is inherently interdisciplinary, since the course needs to include material on communications (which is typically taught in Electrical Engineering departments), and telecommunications and computer networking (which are typically taught in Computer Engineering or Computer Science departments). While it is possible to change the degree of

coverage depending on the focus of the course, both areas must be covered. We focus on courses in wireless computer networks and our target audience is students with a computer science or computer engineering background.

A second challenge is how to offer hands-on assignments and projects. While lectures and paper assignments are a good way of teaching and testing the concepts of wireless networks, hands-on assignments are also needed, both to clarify and reinforce the concepts covered in class and to show how the course material operates in a real world setting. We are particularly interested in assignments that illustrate the differences between wired and wireless networks, and that show how physical layer properties and protocol features affect the performance of the wireless network. These concepts tends to be very abstract when presented in the classroom because they cannot be visualized easily.

The behavior of wireless networks depends however heavily on the physical environment. As a result, hands-on assignments suffer from lack of repeatability. It is also difficult to explain results, since they are influenced by potentially many external factors (production wireless networks, people moving around, microwaves) that are hard to identify. In this paper we report on our experience in using a wireless networking testbed that supports repeatable, easy-to-control experiments for assignments and projects in several wireless networking courses.

The remainder of this paper is organized as follows. In the next section, we elaborate on the challenges associated with offering hands-on assignments in wireless networking courses. In Section 3, we give an overview of the wireless emulator testbed, focusing on features that are relevant to education. We describe the course structure, assignments, and projects in Sections 4 through 6. We discuss our experience in Section 7 and summarize in Section 8.

2. HANDS-ON WIRELESS ASSIGNMENTS

The behavior of wireless networks is heavily influenced by the physical environment. For example, seemingly similar wireless networks can have very different performance properties depending on the nature of the walls and objects in the environment. Even worse, movement by people or objects can change the signal propagation environment and thus the network behavior very quickly. Similarly, other wireless devices (e.g. production wireless networks and cordless phones) or any devices using the unlicensed band (e.g. microwaves) can interfere with the experiment. This tight dependency on the physical environment adds considerable complexity to the preparation of wireless assignments:

- *Repeatability* - Wireless experiments are generally not repeatable, i.e. repeated execution of the same experiment will often yield different results. In some cases, outside effects (e.g. interference from production networks) will overwhelm the feature being studied (e.g. impact of distance on throughput). This makes it difficult to make sure that students take away the right lessons and it also creates a problem for grading.
- *Control* - Controlling experiments requires control over the physical world, e.g. placement of devices and objects, movement, etc. This is difficult to do precisely and contributes to the lack of repeatability.
- *Simultaneous experiments* - Practically speaking, multiple students must be able to run experiments at the same time. This is difficult since their experiments will interfere with each other.
- *Observability* - When experiments yield unexpected results, it is often difficult to explain why. The reason is that most factors that affect the wireless network are not visible and can only be detected using expensive specialized equipment. This lack of observability exacerbates the above problems and it is likely to be very confusing for inexperienced students.
- *Breadth of experiments* - It is likely to be only possible to run experiments in one location, e.g. a specific teaching lab. Ideally, it would be possible to compare performance in more diverse environments (indoor/outdoor, stationary/mobile, etc.).

The most common approach for assignments in wireless networking courses, and networking courses in general, is to use simulator such as ns-2 [8] and OPNET [9]. They are fairly easy to use and do not require special infrastructure. However, it has been observed that they often lack physical layer accuracy, which can lead to unrealistic network behavior [7]. From an educational perspective, simulation is also not as exciting as working with real wireless devices, and it is not likely to be as effective in reinforcing the link between course concepts and the real world. Some courses have offered assignments or projects using actual wireless devices. These assignments typically fall into one of two categories. First, assignments at a fairly high level in the protocol stack [2]. Examples include configuring an access point or experimenting with password sniffing. While useful, these experiments are at a higher level than what one would like to offer in a wireless course. Second, some more advanced wireless courses have open ended projects that can leverage equipment from research projects. Again, these projects are of a different nature than what we are looking for in an introductory wireless networking course.

In this paper, we report on our experience in using a new type of wireless networking testbed based on signal propagation emulation for both assignments and open-ended projects in several wireless networking courses.

3. WIRELESS NETWORK EMULATOR

Signal propagation emulation [3] makes it possible to conduct network experiments using real wireless devices running in real-time in a controlled environment. The operation of the emulator testbed is illustrated in Figure 1. A number of wireless devices are connected to the emulator through a cable attached to the antenna port of their wireless network cards. On transmit, the RF signal from a given device is
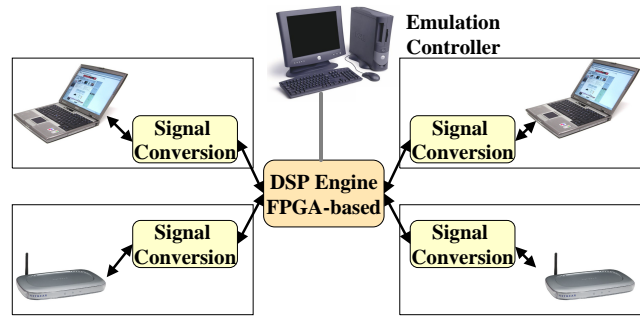

Figure 1: Emulator Implementation

passed into the signal conversion module where it is shifted to a lower frequency, digitized, and then forwarded in digital form into a central DSP Engine that is built around an FPGA. The DSP Engine models the effects of signal propagation (e.g. large-scale attenuation, multi-path, and small-scale fading) on each signal path. Finally, for each device, the DSP combines the processed input signals from all the other devices and sends it to the signal conversion module. It converts the digital signal back into an RF signal and sends it to the wireless card through the antenna port.

The emulator simultaneously offers a high degree of realism and control. The devices are shielded from each other (boxes in Figure 1) so that no communication occurs over the air. Since the devices only communicate through the emulator, we have full control over the signal propagation environment. The only simulated element is the propagation of signals between hosts. Channels are modeled at the signal level but the wireless hardware, signal generation, signal reception, and software on the end hosts are all real.

Emulation is controlled by an Emulation Controller executing on the Emulation Control PC. The Emulation Controller models the emulated physical environment including the movement of the wireless devices (World Model in Figure 2). It also coordinates the movement of devices with the modeling of the signal propagation in the FPGAs by modifying its parameters in real time (Channel Models). For example, as nodes move in the emulated physical environment, the emulation controller adjusts the parameters for the large-scale attenuation and fine grain fading in a way that is consistent with the speed and location of the devices and writes the new values to the FPGA. To use the system, user log into the Emulator Control PC over the Internet.

Users can specify and control wireless experiments in three different ways [4] (left side of Figure 2):

- *Interactive GUI* - Users can control key features of the experiments such as node placement and channel parameters using a GUI that runs as an applet in a web browser. When using the GUI, users often log into the laptops to manually control applications.
- *Scripting* - Users can write a script to control nodes, applications and wireless channels. The scripting language is very simple, e.g. there is no control flow.
- *Programmatic control* - Users can use Java code to access to all emulator features by directly calling the appropriate classes. Programmatic control offers very tight control over the timing of experiments.

The wireless emulator supports the full 2.4 GHz ISM band and has 15 nodes. The nodes are laptops with 802.11b inter-
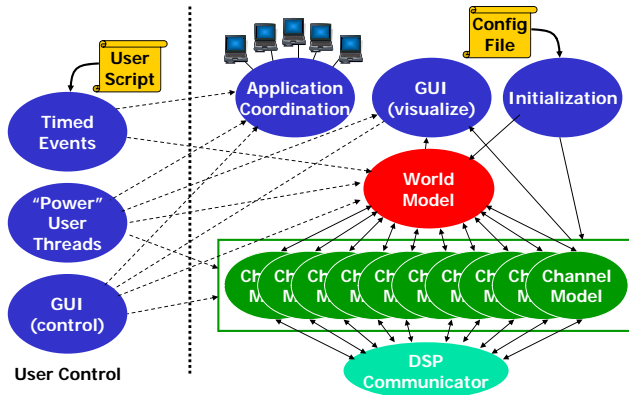
Figure 2: Emulator control software

Assignment	Problem
1	Impact of distance
	GUI vs scripts vs Java
2	Impact transmit rate
	Evaluation auto-rate
	Impact of interference
	Use of multiple channels
3	Impact of hidden terminals
	Evaluation RTS/CTS
	Characterize packet capture
	TCP over wireless
	Effect of MAC retransmission

Table 1: Overview Course Assignments

faces based on the Atheros chipset. In the course, all nodes ran Linux and we used the Wadwifi device driver for wireless communication. Some nodes also have software radios, but these were not used in the courses.

The original goal of the wireless emulator testbed was to allow researchers to compare the performance of different wireless protocols under identical circumstances, e.g. [5, 6]. However, the same properties that make the testbed an attractive platform for the research community, are also useful in an educational setting.

4. COURSE STRUCTURE

We developed an undergraduate course titled "Hands-on Introduction to Wireless Networking" that introduces computer science and computer engineering students to wireless networking. The prerequisite is an introductory course in computer networking, so students are familiar with the OSI layered network model and the general operation of MAC, transport and routing protocols. The course was offered at ETH Zurich in Spring 2007 and at CMU in Spring 2008. Both offerings were "mini courses" that have half the class time of a regular course. The course includes a set of lectures, three hands-on assignments, and a more open-ended project. The assignments and project are graded and students are also tested in a final exam and two short quizzes given during the semester. In Spring 2009, we are also offering a full scale version of the course for graduate students. The graduate course uses the emulator for projects but does not include the assignments.

The lectures are broken up into three segments. The first segment provides an introduction to wireless communications. It includes a discussion of wireless channel properties, antennas, and basic coding and modulation techniques. These topics are covered at a relatively high level so the material is accessible to students with a CS background. The goal of this course segment is to provide insights in the challenges associated with wireless communication and the degree to which different physical layer techniques can overcome these challenges. The second segment focuses on MAC protocols. We introduce various wireless MAC techniques and discuss how they are applied in widely used standards, including WiFi, personal area networks, WiMax, and cellular. The last segment covers a wide range of wireless networking topics, such as how wireless networks function

in the broader internet, multi-hop wireless networks, sensor networks, and management of wireless networks. The lecture time allocated to the three segments is roughly 30%, 40%, and 30% of the semester.

5. ASSIGNMENTS

The goal of the assignments is to reinforce the concepts introduced in class and to make the phenomena and techniques more concrete. There are three assignments, covering the physical layer, MAC protocol techniques, and interactions between TCP and the wireless network (see Table 1).

The first assignment is a very simply assignment that teaches the user how to use the emulator using the three usage modes (GUI, script, Java). The students study how throughput changes as a function of the distance between the sender and the receiver. They do this by collecting throughput results for different distances, using a log-distance model for attenuation. They also run experiments with and without fine-grain fading. They execute the same assignment using the three usage modes so they get a feel for the relative advantages and disadvantages of the three modes. This is important for both later assignments and the project.

The second assignment looks at the performance of wireless links under different conditions. First, students measure link throughput for different distances and different transmit rates. They see how the best transmit rate depends on the attenuation of the channel. The also run a set of experiments using an auto-rate selection algorithm, exploring under what conditions the auto-rate selection algorithm does a good job picking a good transmit rate and under what conditions it fails or performs poorly. Finally, students run a set of experiments exploring how interference between multiple wireless links affects performance. Contention for the ether can reduce network capacity as a result of collisions, retransmissions, and increased idle time due to backoff. Students study how the network capacity changes as a function of the number of wireless nodes and the traffic load. The final set of experiments explores how the use of multiple channels can improve throughput.

The third assignment looks at the performance of the MAC layer. In the first problem the students are asked to set up a hidden terminal situation, consisting of a transmit-receive pair and an additional transmitter that interferes with the receiver; the two transmitters cannot hear each other so the carrier-sense mechanism of 802.11 fails, resulting in collisions at the receiver. In the real world, it is dif-

ficult to set up a hidden terminal situation that behaves in a consistent way, but this is very easy to do on the wireless emulator testbed. The students can then study how performance drops in a hidden terminal situation and they also evaluate under what conditions the use of request-to-send/clear-to-send (RTS/CTS) can help. They can for example see the tradeoff between a reduction in packet loss due to collisions and the overhead associated with RTS/CTS. In the second problem they study the packet capture effect, in which a receiver is often able to decode packets despite significant interference from another transmitter due to the robustness of the coding at the physical layer. The students again set up a hidden terminal situation with two transmitters that cannot hear each other transmitting to the same receiver. They then quantify the capture effect for wireless channels with different attenuation and fading characteristics. They also study how the transmit rate affect packet capture: the packet capture effect is much stronger at the lower transmit rates since they use a more robust coding. In the final problem the students run TCP over lossy wireless links and observe how even relative low packet loss rate (of a few percent) can dramatically reduce the performance of TCP. They then evaluate how MAC layer retransmission can overcome this problem.

Each problem in the assignments typically has three components. First, students are asked to collect a set of measurements on the emulator. Second, they have to plot and interpret the results. Finally, they are asked to answer a couple of short homework-style questions on course material related to the experiment.

A number of other problems were considered for the assignments but they were not included because of time constraints. These problems would be appropriate for a full length version of the course. Examples include looking at effects such as multi-path, mobility, and node density on performance, the benefits and pitfalls associated with the use of directional antennas, and a study of exposed node scenarios. A final problem we considered would allow the students to explore a more complex scenario representing, for example, an indoor wireless network in which walls and objects create shadowing and fading effects. Such a scenario opens the door for many problems such as optimizing the placement of access points.

6. PROJECTS

In the second part of both the graduate and undergraduate versions of the course, students do a more open-ended project. Projects are done in teams of two or three students and each team can pick its own project. A list of possible projects is provided, although students can propose their own project as well. Projects fall in different categories.

The first and most challenging type of project involves installing new protocols on the emulator nodes and running experiments. The protocols are typically research prototypes that have been made available on the web. Examples include various routing protocols for ad hoc wireless networks and TCP variants optimized to run over wireless networks. This type of project is typically only practical for students with fairly extensive programming experience. The projects can also get complicated since the code available on the web was often developed for different or older versions of linux so some effort may be required to get the code to work. However, if successful, these projects are very educational.

Students can experiment with new protocols, trying to confirm results presented in the research papers. They can also compare different protocols (e.g. ad hoc routing protocols) and see how they compare under different conditions.

A second category of projects involves repeating and extending published results characterizing the behavior of wireless 802.11 links, e.g. [5]. This typically involves relatively little programming and it reinforces and expands on material covered in the first half of the course. Finally, it is possible to define smaller projects that can easily be implemented at user level. Examples including a protocol for using directional antennas or building model of a world that include objects that attenuate and reflect wireless signals.

The projects have generally worked out well. It is very important that students can pick their own project so they can select a project that matches their background. There is huge variability in the quality of projects. There are several reasons for that. First, there is simply a big difference in the amount of energy students put in the project: some teams enthusiastically put on a huge effort (and learn a lot) while others do a more minimal project. Second, the backgrounds of students are quite different. Finally, there is some luck involved, especially with the first category of projects. Since the projects are open ended, they sometimes turn out to be harder than expected so students end up with fewer results.

7. DISCUSSION

We discuss our experience with using a controlled wireless testbed in courses and we present possible improvements.

7.1 Experience

Overall, the feedback from the students on the course has been very positive. The students enjoy working with real wireless hardware for the assignments and they are able to do more interesting projects on the testbed than would otherwise be possible. While there were some glitches for the first offering of the course (e.g. availability of the hardware, minimal documentation), all students were able to complete the assignments and their project. Since then, the emulator testbed has matured and it is currently used on a daily basis by researchers both at CMU and elsewhere. The learning curve for using the emulator is a concern, but the first assignment seems to do a good job in bringing the students up to speed. An interesting point is that the graduate course that is using the emulator is offered in the CMU Electrical and Computer Engineering department. About a third of the students have a communications background and we found that they generally need more help in learning how to run experiments on the emulator.

The ease of control and repeatability of the assignments is very important. Ease of control allows us to do very diverse assignments on a single hardware platform, as described above. Repeatability is also important. It helps in preparing meaningful assignments and it helps identify and fix problems in the student's work.

Note that repeatability does not mean that repeated experiments give exactly the same results. Experiments will often give different results because of randomness present in different parts of the system. First, the signal propagation models that are used in the emulator include fast fading, an effect that is caused by random mobility in the environment. By design, the detailed behavior of fast fading is random, although it generally averages out on the long run. Second,

network protocols running on the laptops include randomness by design, e.g. the random backoff in 802.11 and TCP timers. Finally, operating systems can introduce randomness, e.g. as a result of masked interrupts that can delay the handling of incoming packets. The differences in the results are usually small, although they can be large for specific experiments, e.g. TCP over lossy channels. Of course, there is educational value to the students to see and quantify the impact of these random effects, since some level of randomness is unavoidable in any large scale system.

Finally, the wireless emulator sits on a private network and wireless communication is isolated from the physical environment. This means that experiments cannot harm production networks and we do not have to be concerned that failed experiments will affect other users, or that the emulator testbed will be used to launch DOS attacks. This also means that we can give users root privileges on the laptops, which is useful for some projects.

7.2 Challenges and possible improvements

The emulator testbed is fairly small (15 nodes) so it can only support a limited number of users. The first time we offered the course we had eight students and the emulator software only allowed one user to use the system at a time and testbed was also used by some research groups. As a result, it was at times difficult to gain access to the system. Shortly after that, the emulator group replaced the proprietary software that was used to manage the emulator hardware by the Emulab software [1] from the University of Utah. One of the advantages of this change was that users can now do space sharing of the testbed: multiple users can allocate different subsets of nodes and they can run simultaneous experiments while being fully isolated. This turned out to be very beneficial in the later instances of the course. Several students can use the emulator simultaneously since most problems in the assignments only require 2-4 nodes. This makes it possible to support larger class sizes. In Fall 2008, an undergraduate wireless networking course at the University of Wisconsin used the the wireless emulator testbed for an assignment. The course had about 30 students and our experience shows that supporting this class size is not a problem.

Doing the same style of projects in larger classes may be more problematic. First, projects typically require more nodes and often also require more debugging time. Second, projects can become quite time consuming for the teaching staff. A more practical approach is probably to define and test a small set of projects, from which students can then pick. This would make the projects more efficient, and may in fact result in a better learning experience for less well prepared students. It may still be possible to allow teams of more advanced students to take on more ambitious projects. This is the approach we are using the graduate offering of the course in Spring 2009, which has 35 students.

The benefits of the emulator testbed (repeatability and ease of control) also have some drawbacks. It may leave students with the false impression that wireless networks sometimes (or even often) have very nice predictable performance. In practice this rarely happens. One possible way to counteract this concern is to include some experiments in the assignments that create more complex and more challenging scenarios on the emulator. An alternative could be to include some experiments on a traditional uncontrolled testbed so they can see this the contrast. Of course, students use wireless on a daily basis and have experienced its sometimes unpredictable performance first hand, so this concern may be unfounded.

8. CONCLUSION

In this paper we described our experience in using a wireless networking testbed that supports repeatable and easy to control experiments for assignments and projects in courses on wireless networking. The testbed allows us to give assignments exploring both physical layer and MAC protocol behavior that would be very difficult, if not impossible, to do in a traditional wireless network testbed that is open to the environment. Feedback on the course has been very positive. The students enjoy working with real wireless hardware for the assignments and they are able to do more interesting projects on the testbed than would otherwise be possible. One of the challenges is scaling this approach to larger class sizes, but our experience so far suggests that supporting class sizes of up to 30 students is possible.

9. ACKNOWLEDGMENTS

10. REFERENCES

[1] J.-P. Hubaux. Emulab - Network Emulation Testbed, 2008. http://www.emulab.net/.

[2] J.-P. Hubaux. Mobile Networks - Hands-on Exercises, Spring 2008. Available at http://mobnet.epfl.ch/index.php?page=material.

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

[4] G. Judd and P. Steenkiste. Software architecture for physical layer wireless network emulation. In *The First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH 2006)*, Los Angeles, Sep 2006. ACM.

[5] G. Judd and P. Steenkiste. Understanding Link-level 802.11 Behavior: Replacing Convention with Measurement. In *Wireless Internet Conference 2007 (Wicon07)*, Austin, Texas, October 2007.

[6] 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)*, Dublin, Ireland, July 2008.

[7] 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.

[8] S. McCanne and S. Floyd. UCB/LBNL/VINT Network Simulator - ns (version 2). http://www.isi.edu/nsnam/ns/, April 1999.

[9] OPNET Tech. Opnet. http://www.opnet.com.