Demo Abstract: ASSERT - An Advanced Wireless Environment Research Testbed

Paul Johnson[†*], Ehsan Nourbakhsh[†], Ryan Burchfield[†], Jeff Dix[†], Ravi Prakash[†], S. Venkatesan[†], Neeraj Mittal[†], and Lee McFearin[‡]

[†]Department of Computer Science, The University of Texas at Dallas

[‡]Crane Wireless Monitoring Solutions

Abstract

Software simulation has often been used to evaluate proposed protocols for wireless devices. Simulation allows for rapid development and testing, but does not provide a realistic RF environment. To compensate for this, field experiments are performed. However, problems encountered during field experiments can be difficult to locate and correcting problems on-site can be time-consuming. Emulations attempt to provide the advantages of simulation and field experiments without the suffering the disadvantages of both. This demo provides an overview of the hardware, software, and emulation capabilities of ASSERT, an emulation testbed.

Categories and Subject Descriptors

C.2.1 [**Computer-Communications Networks**]: Wireless Communications; D.2.5 [**Testing and Debugging**]: Distributed Debugging; C.4 [**Performance of Systems**]: Measurement Techniques

General Terms

Measurement, Design, Experimentation

Keywords

ASSERT, Wireless, Sensor, Testbeds

1 Introduction

A required step for each innovation in wireless hardware and software design is field testing. Bugs found during field experiments can be difficult to detect and correct. Even once corrected, each device has to be reprogrammed and the experiments run again. The higher cost induced by field experiments has motivated the research community to use simulations to evaluate their solutions. See [4] for more information about the problems of outdoor experiments. While simulations are easier to perform and can be repeated multiple times, their results only weakly correlate with actual deployments. The major reason for this discrepancy is model complexity. Even sophisticated simulation tools such as NS-3 [2] make simplifying assumptions about physical properties. These assumptions can often have drastic effects on algorithm performance [5].

*Contact E-mail: paul.johnson@student.utdallas.edu

SenSys'09, November 4–6, 2009, Berkeley, CA, USA.
ACM 978-1-60558-748-6

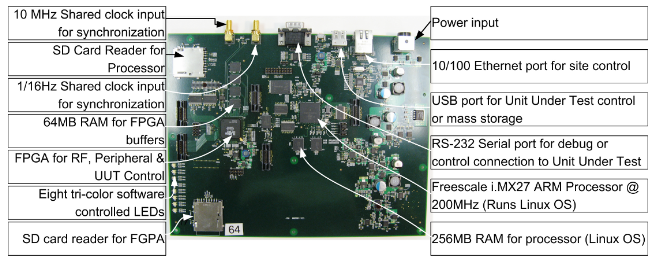

Figure 1. One Site Board. Each Site Board is paired with one RF board

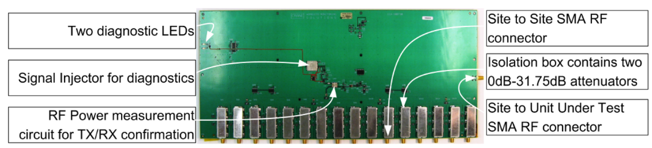

Figure 2. One RF Board

2 Motivation

Our motivation when creating ASSERT was to provide a balance between simulation and field experiments. We wanted the ability to run controlled, repeatable experiments that would have similar performance to that of field experiments without the associated deployment and travel costs. We also wanted to provide a platform where the user can control the RF environment. Much like in simulations where a user can specify node locations and link characteristics, we wanted to give the user a high level of control over the RF environment. We also wanted to provide detailed feedback to aid the user with design and debugging.

3 Design

ASSERT has been a work in progress for the past two years. The first year was largely spent designing the platform, hardware and software. We currently have completed and deployed a forty site testbed.

3.1 Hardware Design

The hardware for each site consists of two mated boards, the Site board and the RF board. Each Site board (Figure 1) consists of an ARM 200MHz processor, 256 MB of onboard RAM, a Xilinx FPGA, support for an external clock, and various peripheries such as Ethernet and RS-232. The RF board (Figure 2) consists of 17 SMA coaxial connectors,

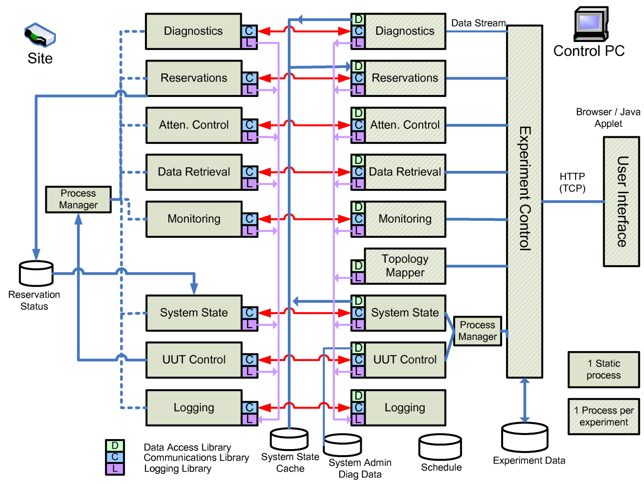

Figure 3. Overview of ASSERT Software Architecture

16 are used to connect to other Site-RF board pairs, and 1 is dedicated to the Unit Under Test(UUT). The testbed does not need to know the modulation or encoding scheme of the device because it emulates the over-the-air propagation using cables. This is attractive to industry as proprietary information is protected. Currently we have 29 MICA2 (900 MHz) Crossbow Motes [1] with gateways as UUTs. However, any device operating in the 700MHz to 1100MHz range is natively supported. The testbed is not limited to ISM bands. It can perform experiments in licensed bands at nominal power levels without interfering with commercial devices.

Each of the 16 SMA connectors have two programmable attenuators (0-31.75 dB each) that we can control using the FPGA, varying them as frequently as once per millisecond. By feeding the same 10MHz clock source to each of the Site-RF board pairs, we can synchronize all testbed activities, allowing us to reliably repeat experiments. The RF board also has a built-in signal generator and RSS device for diagnostic purposes. By re-wiring the testbed, we can emulate any topology (or subgraph thereof) where each node has a maximum degree of 16.

3.2 Software Architecture

The ASSERT software performs a variety of tasks, including monitoring for faults, efficiently allocating resources, and providing an easy to use interface. It is responsible for running experiments and collecting any results generated. The software is divided into *slices*, with each slice implementing a specific functionality. The *experiment control slice* is the bridge between different slices. As shown in Figure 3, most of the slices operate on both the control PC and sites.

An experiment is started once a remote user connects to the *user interface slice* and configures a new experiment. Initially the sites are reserved for the defined duration of this experiment through the *reservation slice*. This ensures that all selected sites are used exclusively for this experiment. After this step, the experiment control slice synchronizes internal clocks of all reserved sites and sends the details of the experiment to the *attenuator control slice*. This information includes the duration of the experiment, the links used in this experiment, and the distribution of attenuation values (static, normal, uniform, etc.) and their frequency of change.

Consistent with the producer-consumer model employed by operating systems, each site acts as a producer by generating a sequence of attenuation values. Acting as a consumer, the FPGA on the site hardware reads these values and sets the attenuators accordingly. During an experiment, *system state slice* reports to the control PC status of each site including the experiment it is running, remaining time of the current experiment along with health information of each site. This allows the control PC to detect site failures during an experiment, invalidating the results.

Once the experiment starts, the *data retrieval slice* starts recording UUT output. The *monitoring slice* records the Received Signal Strength (RSS) metric observed on each site. The RSS and UUT logs are returned to the user as the experiment results. The *logging slice* records all error and control messages generated by the testbed software. When the experiment completes, sites are reset to their initial states.

The *diagnostics slice* provides maintenance and topology information by periodically, and on demand, checking the integrity of RF links on the testbed when no experiment is running. A run of the diagnostics slice gives a topology snapshot of the testbed cabling. It also reports which links have changed since the previous run.

4 Demonstrations

To demonstrate the operation of the testbed we propose to use four sites and one control PC. Three of the UUTs used are Crossbow MICA2 Motes [1]. Fourth UUT is a Wi-Spy Spectrum Analyzer [3] which allows us to independently verify UUT transmissions. All these sites are cabled as a clique of size four (K_4). We will first demonstrate how the testbed is isolated from external RF activity. We run a simple neighbor discovery algorithm on the UUTs. This algorithm uses heartbeat messages to detect neighbors. By changing the attenuation of each link, we can emulate dynamic network topologies, and to some extent, mobility. We developed a user-level application that parses the UUT logs to reconstruct the connectivity between the motes. We also show how we can precisely correlate the transmissions and receptions between the UUTs using the RSS logs.

5 Acknowledgments

ASSERT was designed in collaboration with Crane Wireless Monitoring Solutions and funded by US Department of Defense, Defense Microelectronics Activity (DMEA).

6 References

[1] Crossbow MICA2 868, 916 MHz. http://www.xbow.com/Products/productdetails.aspx?sid=174.

[2] The Network Simulator - ns-3. http://www.nsnam.org/.

[3] Wi-Spy 900x: Spectrum Analyzer for 900MHz ISM Band. http://www.metageek.net/products/wi-spy-900x.

[4] R. Burchfield, E. Nourbakhsh, J. Dix, K. Sahu, S. Venkatesan, and R. Prakash. "RF in the Jungle": Effect of Environment Assumptions on Wireless Experiment Repeatability. In *IEEE ICC'09*, 2009.

[5] D. Cavin, Y. Sasson, and A. Schiper. On the Accuracy of MANET Simulators. In *POMC '02*, 2002.