ASSERT: A Wireless Networking Testbed

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

Distributed Systems Lab, Computer Science Department,
The University of Texas at Dallas 75080 USA
`{ehsaan,jxd028100,paj041000,ryanb,ravip,venky,neerajm}@utdallas.edu`

Abstract. As wireless networks become a critical part of home, business and industrial infrastructure, researchers will meet these demands by providing new networking technologies. However, these technologies must be tested before they can be released for mainstream use. We identify the key design considerations for a wireless networking testbed as *a*) accuracy *b*) controllability *c*) mobility *d*) repeatability *e*) cost effectiveness *f*) data collection *g*) resource sharing *h*) multi-nodal capability *i*) scalability. In this paper we portray how we have used coaxial cables and our custom hardware of RF switches and programmable attenuators to create Advanced wireleSS Enviroment Research Testbed (ASSERT), addressing the above requirements. The created network is immune to interference from other wireless networks, and can emulate mobility and link deterioration. ASSERT supports various types of wireless devices, providing researchers in academia as well as industry with the necessary experimentation tools to validate their designed protocols and devices.

Key words: ASSERT, Wireless, Sensor, Testbeds, Repeatability

1 Introduction

As wireless networking is becoming more pervasive, there has been a greater desire to develop communication hardware and protocol stacks that have a number of desirable properties like increased throughput, reduced latency, reduced energy consumption, quality of service, security, etc. Consequently, several academic and industrial research groups are actively working towards improving the performance of wireless networks. Due to their inherent complexity, accurate theoretical analysis of the performance of large wireless networks is quite challenging. Hence, several researchers have resorted to simulation experiments to evaluate the performance of large wireless networks. Most simulators make a set of simplifying assumptions about the communication medium and the communication protocols [1, 2, 3, 4]. This enables them to run experiments within a reasonable amount of time. However, sometimes these assumptions can bias

---

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

experiments in a significant way. It is no surprise that often the results of simulation experiments differ significantly from the actual performance of wireless networks.

Over the last few years several research groups have initiated the development and deployment of wireless networking testbeds such as Netbed [5], Kansei [6], Trio [7], ExScal [8] and other testbeds at U.C. Berkeley [9]. The underlying assumption of all these endeavors is that experiments conducted on a testbed composed of actual wireless devices communicating over the air will yield results representative of performance in field deployments. Several of these testbeds are deployed in general-purpose laboratories in academic buildings. As these laboratories are not shielded from external wireless interference, the experimenters have little or no control over the environment in which experiments are conducted. As wireless interference fluctuates in an unpredictable fashion it is also not possible to accurately compensate for the interference. As a result, it is almost impossible for experimenters to independently reproduce the results obtained by other research groups.

Building a Faraday cage large enough to house wireless networks of non-trivial diameter is prohibitively expensive. The other alternative of deploying the network outdoors, sufficiently far from any interferer, is also not very attractive. Outdoor deployments, unless sufficiently ruggedized, can deteriorate quickly due to variations in temperature and humidity. They are also prone to vandalism. Moreover, it may not be possible to conduct outdoor experiments during inclement weather. Also, innovative ideas need to be employed to conduct mobile networking experiments if one does not have a lot of manpower available.

Based on the above discussion, one can count some general requirements for an ideal testbed, similar to considerations that De et. al in [10] proposed for a multihop testbed. The ideal testbed shall: *a*) accurately reflect wireless network behavior (**accuracy**) *b*) give the experimenter enough control to configure topology and environment conditions, thus eliminating uncontrolled background noise sources (**controllability**) *c*) be able to emulate mobility of the nodes (**mobility**) *d*) conduct experiments that are reproducible and easily repeatable (**repeatability**) *e*) be cost effective in terms of hardware, manpower, space and time requirements to set up, run experiments on and maintain (**cost effectiveness**) *f*) provide necessary tools to the experimenter to collect and analyze data (**data collection**) *g*) be able to share the available resources to conduct multiple experiments without interfering with each other (**resource sharing**) *h*) have **multi-nodal capability** (i.e., it will support many types of nodes) *i*) have the ability to scale to a large number of nodes (**scalability**).

In the next sections, we will present some clear examples of challenges in designing and building our large-scale testbed called ASSERT(**A**dvanced wirele**SS** **E**nvironment **R**esearch **T**estbed). In Section 2 we show how other testbeds have worked towards some of the above requirements. Section 3 will present some of the main characteristics of our work and show how we are able to satisfy the above requirements. Following this overview, we will describe the architectures selected for both our hardware and software implementations of the ASSERTin Section 4. Some final conclusions and future work are discussed in Section 5.

2 Related Work

Creating an environment to test and validate new protocols and hardware designs has been a challenge for wireless researchers. The desired environment should be precisely controllable and the possibility of repeating the same experiment is vital. The first group of attempts to create such environment focused on using the antenna of unit under test (UUT), resulting in over-the-air transmissions. Efforts such as MoteLab [11] and ORBIT [12] are examples of this category. These testbeds do not enable the researcher to control the exposure of the UUT to background noise and interference from other nodes. The distance between nodes is limited to the physical placement of the devices, and **mobility** is not provided by design. Although the size of the network they are able to create is large, they are not able to partition them effectively, thus failing to address **resource sharing** requirement. Noise injection and MAC filtering can be used to create topologies, but as [13] pointed out **repeatability** and reproducibility of results from noise injection is reduced if nodes with marginal SNR are involved. MAC filtering will also fail to address **mobility** because in this method either a packet is able to go through or is completely dropped, unlike the way actual movement alters a signal.

Mint-m [14] and Mobile Emulab [15] address the **mobility** and **controllability** requirements by using small robots to move the UUTs around the test area, placing the nodes as the experimenter requests. Added attenuation between the UUT and antenna can change the virtual distance between nodes. Pharos [16] uses a similar approach of using robots, but the environment is set outdoors. These approaches still do not eliminate the problem of exposure to background noise or interference from other testbed devices. This would fail to address **controllability** requirement. Also, **mobility** is either limited to the speed of the robots or is not supported as a design feature.

To overcome lack of control over the environment in the aforementioned testbeds, a second category of efforts digitize the outgoing signal of the UUTs and use existing RF propagation models to emulate effects such as distance and multi-path on the signal. The resulting altered signal is fed to the destination devices. This approach provides the required **controllability** requirement, but its **accuracy** is limited to the precision of the applied RF propagation model. Furthermore, another limiting factor would be the available processing power. Signal alternation requires sophisticated calculations, hence higher number of nodes will make the processing power requirements harder to achieve, failing to address the **cost effectiveness** requirement. Work done in CMU [17] is one example of such testbeds.

To avoid the processing requirements and the reliance on theoretical propagation models, the third category of testbeds focus on simply using coaxial cables to connect different nodes. Attenuators and RF switches can be used to form topologies and emulate distance between nodes. MeshTest [18] is one example of such set up with a design close to our work. However, their method requires complex design for higher number of nodes. High cost of the switch matrices used might be another obstacle to building larger networks, failing to address **cost effectiveness** requirement. Our work focuses on creating a scalable testbed using attenuators and RF switches. We isolate the nodes from each other and ensure that multiple experiments can run without interfering with each other. Our design choices ensure we are able to scale to at least one thousand nodes without unreasonable processing power requirements. We also add a generic support for new UUTs, satisfying the **multi-nodal capability** requirement.

3 Design Overview

ASSERT relies on creating virtual distances using controlled attenuation levels. If two nodes are far from each other, the signal received on one node from the other is attenuated by a certain calculated value to emulate distance. This attenuation value can be calculated using simple formulae and validated through experimentation, addressing the **accuracy** requirement. We control the attenuation value between nodes dynamically and record our calculated values. This will allow us to repeat the experiment exactly the same way multiple times.

Each UUT is connected to a site. This site consists of a microprocessor and its peripherals, also a Field Programmable Gate Array (FPGA) as well as a set of 16 attenuators and corresponding RF connectors. The site has an expandable interface board that allows the processor to communicate with the UUT through an interface such as RS-232 or USB. The site processor is running Linux and is connected to an Ethernet network for network storage and communication. Each connector can be connected to another site using a coaxial RF cable. The main functionality of the FPGA is to set the attenuators on each RF connector as instructed by the microprocessor and poll the RSS meter on a millisecond basis. In order to control the environment and the way that the UUTs connect to each other, we replace the antenna of each UUT with a coaxial connector. This connector is connected to 16 other connectors via a set of attenuators and RF switches. This 1-to-16 connection means we are able to add more sites and enhance our testbed only by adding sites, addressing the **scalability** requirement. We synchronize the clock on sites through our clock distributors so that we are able to set the attenuations on these 16 connectors, called ports, almost instantly on all sites. This will allow us to control how the sites are “virtually” moving in reference to each other with a high precision, addressing the **mobility** requirement. A block diagram of a site is demonstrated in Figure 1. As shown in Figure 2, a front-end computer called Control PC is connected to all sites through Ethernet and is responsible for getting experiment definitiion from user, sending control information and gathering the results.

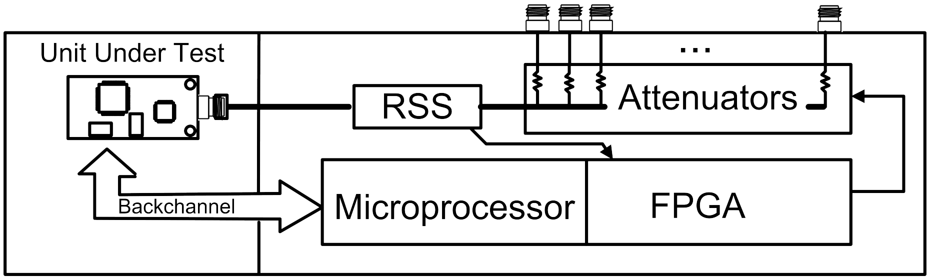

Fig. 1. Block diagram of one site and the related RF, data and command lines

3.1 Reproducibility and Repeatability

As demonstrated in [19], reproducibility of experimentation can be difficult due to inconsistency in environmental conditions. Many wireless networking testbeds operate in schools and laboratory environments, where none-testbed devices can interfere with an experiment. To the testbed, these devices are seen as noise. It is important to run tests in real environments, where you can study the effect of noise on your devices performance. But for a testbed to provide reproducible experiments, we need the ability to control the noise around the testbed, so that each run of an experiment will experience the same amount of noise allowing us to make comparison across testbed experiments. As we have stated above, our solution to this problem is to place all communication between devices on coaxial cables. By properly shielding a wireless device, and connecting it to our coax network, we have control over which devices can “see” each other in the network. We can also prevent outside leakage from other wireless networks operating on the same band.

Since we are using coaxial cable as the medium to transmit RF signal, we would like to emulate effects the environment has on signal reception such as multipath. To the network layer, the effects of RF propagation come down to whether bits/packets are “lost”. By changing our attenuators according to a fading pattern or distribution, we can emulate many real world affects on an RF signal. Researchers will have the ability to select what attenuation patterns they would like to use. Each site processor will calculate the attenuation for each link due to the particular attenuation pattern and set the link’s attenuators accordingly. The digital attenuators can be changed on a millisecond basis. All the parameters for the link’s attenuation patterns, including random number generator seeds are stored in a Control PC’s database, and can be retrieved again to rerun the experiment. Using the same concept, we can also emulate mobility. Here the key issue is a rapidly changing topology, changing neighbors, and of course packet loss. We can do this as above by changing each link’s attenuation dynamically to emulate the changing environment, or neighbors.

3.2 Multi-nodal Capability and Data Collection

As evident in the previous sections, there are only some general expectations we have from the unit under test. Apart from these general basic requirements, we consider the UUT a black box which is vital for the **multi-nodal capability** requirement. We expect the UUT to transmit in the frequency range our RF equipment is designed to work. We also expect it to have an antenna that can be replaced by a coaxial cable, so that we can connect the RF transmitter to one of our testbed sites. The other optional requirement is to have a RS-232 interface so that the UUT can receive commands, such as *reset* or *load image*, from the site. This interface can be also used for the data collection mechanisms we have provided. All logged data written by the UUT to the RS-232 serial interface is kept on file, with timestamps of each log message, and is returned by the testbed software as part of the experiment results.

The timestamps can be used by the experimenter to correlate data from different units under test. For example, since all the clocks are synchronized, the experimenter can see if a transmission by one node at a certain time has been received at another node or not. Furthermore, we also record the RSS meter reading of each site as well, and return the timestamped values to the user as part of the results. In the given example, the experimenter can ensure that the signal did go through the RF component of the sender and it did cause RSS changes on the receiver side. This combination of correlated data can be an important tool for the researcher, addressing the **data collection** requirement.

3.3 Experiment Setup and Execution

At the beginning of the experiment, the user selects if they want to create a new experiment or repeat an already existing one. If they choose to create a new one, they shall provide list of sites they need and the links between these sites. They also indicate when will each link or group of links change their attenuation pattern or value. The user is optionally prompted to upload the *image* to be written to the UUT, if the UUT supports such functionality. Once a generic image is created, specified sites are reserved for the setup stage. This will ensure that sites are available for the experiment and no other experiment is using them. Next step is to synchronize the clocks of the sites, and flashing the UUTs if an image was provided. Then an XML descriptor of the timeline of the experiment is generated. This XML file contains the ports on each site, their attenuation pattern and the start and end time each pattern will be active. Any port not described in this pattern is set to maximum attenuation. The XML descriptor for each site is sent to it. After all sites receive and parse the XML descriptors, they will start the experiment at the same time by resetting the UUTs, enabling logging and starting to set attenuation levels on ports. The reservation is renewed for the duration of the experiment.

Each site will report to the Control PC when the end time of the last attenuation indicated in the XML descriptor has passed. Once Control PC has got the successful termination signals from all the sites in an experiment, the Control

PC will disable logging and release the sites. The user is notified and the log files created during the experiment are made available to them. If any of the sites encounter a fatal error during an experiment, they will notify the Control PC. The Control PC will then terminate the experiment early, and notify the user that a problem has occurred.

One other method to run an experiment is to run multiple runs of the same experiment. The same procedure will be repeated for the number of times that user requested, and in the end the collection of all results are passed to the user. They will be also notified if any of the runs failed. This will be a powerful tool for experimenters, because each time the emulated environment will be replicated precisely as the previous run.

The previous methods give the necessary support for the user to easily and quickly set up one or multiple runs of an experiment, or repeat an existing one. This is to address the **controllability** requirement mentioned earlier. It also is addressing the **cost effectiveness** requirement. We are significantly reducing the amount of time that the experimenter has to spend to create one an experiment on the testbed. As we reserve the required sites during the experiment and also set the attenuation levels to maximum on unused links, we are able to partition the testbed and run multiple experiments in different parts of it. This will increase the utilization of the testbed, addressing the **resource sharing** requirement.

4 System Architecture

4.1 Hardware Architecture

It is best to think of ASSERT hardware as a graph(as in Figure 2), with nodes representing sites in the testbed, and edges representing RF links between sites. Each site consists of a custom digital board (Figure 3) and a custom RF board (Figure 4). The digital board has a processor, memory, FPGA and serial interfaces where the *unit under test (UUT)* can connect. The processes executing on the digital board can control the operations of the UUT, monitor the experiment, and gather results as described in Section 4.2. The RF board connects to the digital board and provides an interface through which the antenna port of the UUT can connect to the RF board. The UUT interface leads to a 1×16 power divider/combiner. Each output port of the power divider/combiner leads to a programmable attenuator (controllable by the digital board) which can provide signal attenuation between 0dB and 63.5dB, in steps of 0.25dB. The attenuators from two different RF boards can be connected via a coaxial cable forming an RF link between two sites. Thus the signal on this link can be attenuated in the range 0dB to 127dB: a maximum of 63.5dB attenuation provided by each of the two programmable attenuators on the path.

Continuing with the graph theoretic description, the digital and RF boards together correspond to a node with a maximum of sixteen incident edges. The

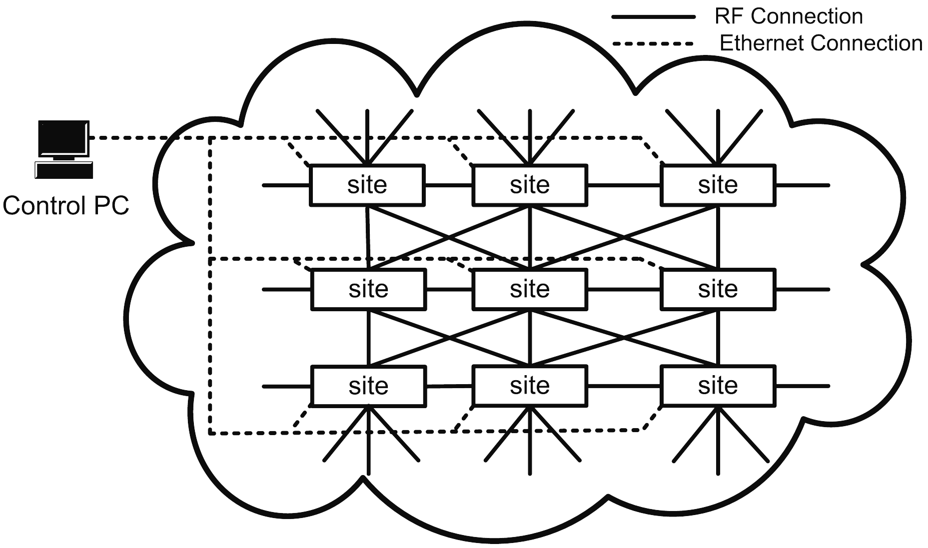

Fig. 2. ASSERT RF Grid and control plane. Lines are coaxial cables, dotted lines are Ethernet connections used for control and data collection.

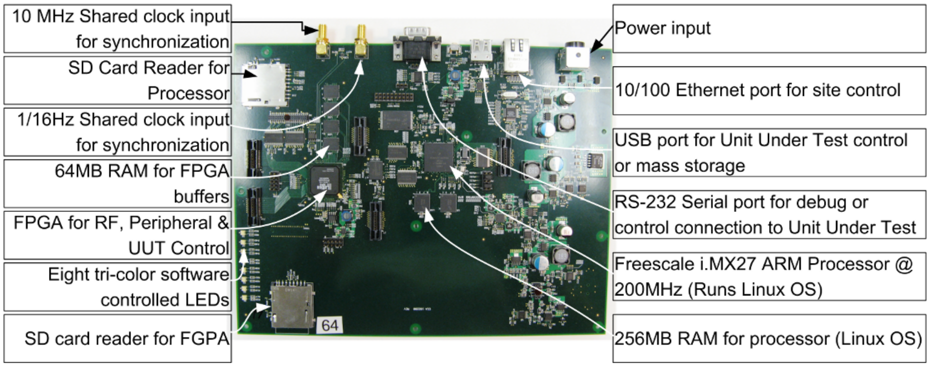

Fig. 3. One Site Board. Each Site Board is paired with one RF board

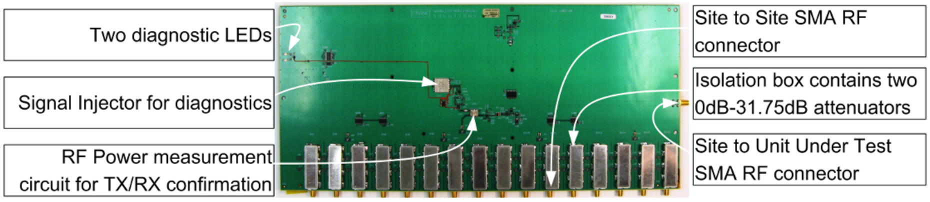

Fig. 4. One RF Board

components on the RF board can currently support all communication bands between 720 MHz and 1125 MHz. In the near future we plan to extend the support to the 2.4 GHz ISM band. The sites, each with a degree of sixteen, are connected to form a mesh. By selectively maximizing signal attenuation along some links between pairs of sites, the corresponding RF links can be removed. Similarly, the quality of links between pairs of neighboring nodes can be manipulated by changing the level of signal attenuation as rapidly as one db every millisecond.

4.2 Software Architecture

The ASSERT software performs a variety of tasks, including monitoring the testbed for faults, allocating its resources in an efficient manner, providing an easy to use interface to users of the testbed, running user experiments on the testbed, and gathering and reporting the results of the experiments. The software is divided into *slices*, with each slice implementing a specific functionality. We now describe the software architecture in the context of the creation and execution of a user's experiment.

The *diagnostics slice* runs both periodically and on demand to check the integrity of RF links on the testbed. This involves selecting sites sequentially, having them send signals along each incident link, and measuring the received signal strength for different values of attenuation along the link. This slice registers a link between two sites if both can hear the signal generated by each other. The quality of this link can be determined as the strength of the received signal compared with the strength of generated signal. A run of the diagnostics slice gives a snapshot of the topology of the testbed in terms of which links are functional and which links are broken and need to be repaired.

The *user interface slice*, running on the central controller provides a graphical user interface to the users. The network topology can be selected from a library of topologies (like mesh, star, ring, etc.) provided by the user interface. If the topology the user wishes to emulate is not present in the library, the user can specify it as a graph with vertices and links between vertices. The user can also specify the characteristics of the wireless links between vertices. Once again, link characteristics can be specified either by selecting from a library of fading patterns, or by providing the formulae to define the link characteristics.

Once the user has specified the desired topology, the user interface slice invokes the *experiment control slice*. The experiment control slice first gathers the current state of the testbed by querying the *system state slice*. The system state slice returns the state of all the sites as well as the set of reservations currently running on the testbed. Then the experiment control slice invokes the *topology mapper slice*. The topology mapper slice computes the topology based on user input as a subgraph of the portion of the testbed that is not running any experiment and is thus available at the moment. The result of the topology mapper slice's operation is conveyed to the experiment control slice. If the topology mapper slice is successful in finding a portion of the testbed to run the experiment, the experiment control slice invokes the *reservation slice*. The reservation slice reserves the corresponding testbed sites for the experiment.

These reservations are implemented as leases of finite duration. If the experiment needs to run longer than the lease duration, the lease must be renewed prior to its expiration.

Then the experiment control slice invokes the *attenuator control slice* at all the reserved sites. As part of this invocation, the experiment control slice informs each reserved site about the properties of the links incident on it. To implement the desired link characteristics the sites at either end of the emulated wireless link work cooperatively. Consistent with the producer-consumer model employed by operating systems, acting as a producer each site generates a sequence of attenuation values along with the time offset from the beginning of the experiment when these attenuation values are to be used on a link. Acting as a consumer, the FPGA on the site hardware reads these values and sets the attenuation values accordingly once it is informed that the experiment has started. Concurrently, the experiment control slice informs the unit under test to start executing the experiment through the *UUT control slice*. Thus, as the unit under test is running the experiment and sending and receiving messages along the emulated wireless links, the attenuator control slice is manipulating the characteristics of these links.

For the entire duration of an experiment running on ASSERT, the *system state slice* monitors the state of the sites. The *logging slice* records all error and control messages generated by all the participating sites. The *UUT logging slice* records information that the UUT writes to serial port of the site as it is running. This data can be used for debugging the UUT by the researcher. Finally, on the completion of the experiment the experiment control slice invokes the *data retrieval slice* to gather the results of the experiment from all the participating sites. These results, along with the UUT logs, are then conveyed to the user via the user interface. The results can also be stored in a database associated with the testbed.

After the experiment completion, the experiment control slice instructs the reservation slice to release all the sites reserved for the experiment. It also updates the system state database to reflect the availability of the newly released sites. As a security measure, and to block access to a user's code by other users, the sites can be re-flashed to original configuration, wiping all user data and settings from them. Similar measures are also designed in the Control PC stopping users from having access to data of others. Furthermore, all the data including experiment details and results can be stored on one single portable hard drive. Also physical and remote access to the facility can be restricted, effectively providing a secure environment for performing experiments that require extra security. In a nutshell, all the software slices together can be thought of as the operating system for ASSERT.

5 Conclusion

The Advanced wireleSS Enviroment Research Testbed (ASSERT) has a small footprint, and emulates mobility and link detoriation inside a room in a repeatable manner. All the experiments are controlled through front-end computers, and network topology can be modified through a sequence of keystrokes and mouse clicks. This takes significantly less time than physically changing the topology in existing over-the-air wireless networking testbeds like Trio, Kansei and ExScal. ASSERT is immune to interference from other devices in the laboratory and the environment. It will be possible for experimenters to inject noise or the desired interference into the system, and observe their impact on the system being studied. Communication between nodes in the testbed does not leak into the environment. We perform all the signal manipulation purely in the RF domain. This allows us to scale to higher number of nodes. Furthermore, while an RF emulator is centralized to allow processing on a single board, our solution is decentralized. With ASSERT it is possible to conduct experiments in licensed bands like the cellular service band without interfering with the services offered by the owners of these licensed bands. Currently, network equipment vendors are forced to test their hardwares in unpopulated areas to minimize interference. Through sophisticated custom hardware and easy-to-use control software ASSERT has many valuable features that allow it to reduce the cost of testing wireless networking protocols at scale.

Future Work: ASSERT currently consists of forty nodes, it is designed to scale to at least one thousand nodes without any design changes. We are converting our user interface (UI) from current small Java program to a web based application, so that the experiment set up and data gathering is done by the experimenter from their browser. The other goal is to add more preset topologies, so that the experimenter has an extended database of already existing topologies, while they are always able to define their own topology.

References

1. D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, 2004.
2. K. Pawlikowski, H. Jeong, and J. Lee, "On Credibility of Simulation Studies of Telecommunication Networks," *IEEE Communications Magazine*, vol. 40, no. 1, 2002.
3. D. Cavin, Y. Sasson, and A. Schiper, "On The Accuracy of MANET Simulators," in *Proceedings of the second ACM international workshop on Principles of Mobile Computing.* ACM New York, NY, USA, 2002.
4. R. Skehill, P. Scully, and S. McGrath, "Characteristics, Results and Findings of IEEE 802.11 in an RF Isolated Testbed," in *18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2007.
5. B. White, J. Lepreau, and S. Guruprasad, "Lowering the Barrier to Wireless and Mobile Experimentation," in *In Proceedings of HotnetsI*, 2002.
6. E. Ertin, A. Arora, R. Ramnath, V. Naik, S. Bapat, V. Kulathumani, M. Sridharan, H. Zhang, H. Cao, and M. Nesterenko, "Kansei: a testbed for sensing at scale," *Proceedings of the fifth international conference on Information processing in sensor networks*, 2006.

7. P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler, "Trio: enabling sustainable and scalable outdoor wireless sensor network deployments," in *Proceedings of IPSN '06*, 2006.
8. A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridharan, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, M. Nesterenko, R. Shah, S. Kulkami, M. Aramugam, L. Wang, M. Gouda, Y. ri Choi, D. Culler, P. Dutta, C. Sharp, G. Tolle, M. Grimmer, B. Ferriera, and K. Parker, "Exscal: elements of an extreme scale wireless sensor network," *Proceedings. 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, Aug. 2005.
9. "Soda Hall wireless and sensor network testbeds." [Online]. Available: http://www.millennium.berkeley.edu/sensornets/
10. P. De, A. Raniwala, S. Sharma, and T. cker Chiueh, "Design Considerations for a Multihop Wireless Network Testbed," *IEEE Communications Magazine*, vol. 43, no. 10, 2005.
11. G. Werner-Allen, P. Swieskowski, and M. Welsh, "MoteLab: a wireless sensor network testbed," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005.
12. 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," *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 3, 2005.
13. S. Kaul, M. Gruteser, and I. Seskar, "Creating wireless multi-hop topologies on space-constrained indoor testbeds through noise injection," in *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, 2006.
14. P. De, A. Raniwala, R. Krishnan, K. Tatavarthi, J. Modi, N. A. Syed, S. Sharma, and T. cker Chiueh, "MiNT-m: an autonomous mobile wireless experimentation platform," in *Proceedings of the 4th international conference on Mobile Systems, Applications and Services*, 2006.
15. D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau, "Mobile Emulab: A Robotic Wireless and Sensor Network Testbed," *Proceedings of INFOCOM '06*, April 2006.
16. D. Stovall, N. Paine, A. Petz, J. Enderle, C. Julien, and S. Vishwanath, "Pharos: An Application-Oriented Testbed for Heterogeneous Wireless Networking Environments," The University of Texas at Austin, Tech. Rep. TR-UTEDGE-2009-006, 2009.
17. G. Judd, "Using Physical Layer Emulation to Understand and Improve Wireless Networks," Ph.D. dissertation, Carnegie Mellon University, School of Computer Science (available as Tech Report CMU-CS-06-164), 2006.
18. T. Clancy and B. Walker, "MeshTest: Laboratory-Based Wireless Testbed for Large Topologies," in *3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities, (TridentCom)*, 2007.
19. 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 International Conference on Communications (ICC)*, 2009.