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,venky,neerajm,ravip}@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 Environment Research Testbed (ASSERT), addressing the above requirements. ASSERT supports various types of wireless devices, providing researchers in academia and industry with the necessary experimentation tools to validate their designed protocols and devices.

Keywords: 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 experiments in a significant way. It is no surprise that often the results of

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

T. Magedanz et al. (Eds.): TridentCom 2010, LNICST 46, pp. 209–218, 2011.

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 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. Outdoor deployments, unless sufficiently ruggedized, can deteriorate quickly due to variations in temperature and humidity. 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*) provide enough control to configure topology and environment conditions (**controllability**) *c*) 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 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 ASSERT in 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 category of testbeds, such as MoteLab [11] and ORBIT [12], attempted to create such environment by focusing on using the antenna of unit under test (UUT), resulting in over-the-air transmissions. Researcher is not able to control the exposure of the UUT to background noise and interference from other nodes, so **controllability** requirement is not addressed . 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 Signal-to-Noise Ratio (SNR) are involved. Using MAC filtering, as suggested by some researchers to emulate mobility, will also fail to address **mobility** because in this method either a packet is able to go through or is completely dropped, unlike actual movement. Mint-m [14] and Mobile Emulab [15] address the **mobility** and **controllability** requirements by using added attenuation between UUTs and antenna and small robots to move the UUTs around the test area. 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.

A second category of efforts such as work done in CMU [17] digitize the outgoing signal of the UUTs and use existing RF propagation models to emulate effects such as distance and multipath 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, signal alternation requires sophisticated calculations so higher number of nodes will make the processing power requirements harder to achieve, failing to address the **cost effectiveness** requirement.

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.

3 Design Overview

Our testbed currently consists of forty devices, known as "sites" in this context. Each 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 processor is running Linux and is connected

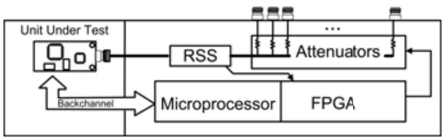

Fig. 1. Block diagram of one site and the related RF, data and command lines

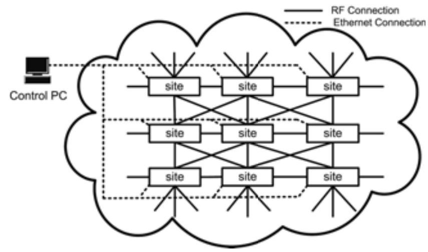

Fig. 2. ASSERT RF Grid and control plane

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. The attenuators are set to values either calculated based on a theoretical model, or based on real life recorded attenuation values such as work done by Lee et al. [19] addressing the **accuracy** requirement. This real life readings can be gathered once for some generic scenarios and then used multiple times to compose more complex terrains.

Each UUT is connected to a site, and the site connects it to up to 16 other sites through its internal switches and attenuators. 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 definition from user, sending control information and gathering the results. Our design choices ensure we are able to scale to at least one thousand nodes without unreasonable processing power requirements.

3.1 Reproducibility and Repeatability

As demonstrated in [20], reproducibility of experimentation can be difficult due to inconsistency in environmental conditions. Many wireless networking testbeds operate in schools and laboratory environments, where non-testbed devices can interfere with an experiment. But for a testbed to provide reproducible experiments, we need the ability to control the noise around the testbed, so that we can compare different 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.

We would like to emulate effects the environment has on signal reception, such as multipath. Each site processor will calculate the attenuation for each link due to the selected pattern by the user and set the link's attenuators accordingly on both sites 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 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. We consider the UUT a black box, a consideration 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 along with RSS meter readings, with timestamps of each log message, and is returned by the testbed software as part of the experiment results. This combination of correlated data can be an important tool for the researcher, addressing the **data collection** requirement.

3.3 Experiment Setup and Execution

We provide more details about creation and execution of an experiment through a common use case of our testbed. We currently have twenty five Crossbow MICA2 motes installed in our testbed. The first step a user has to take to create a new experiment is to create the corresponding topology script. This script specifies which sites are involved in this experiment, what are the attenuation values and how these values change over time. As an example, user might select ten motes and name them according to their own plan. Then use the attenuation values gathered by Lee et al. [19], recorded specifically for MICA2 motes, to place nodes realistically. The user might also decide that a node A slowly moves away from its neighbors at time T. This movement can be realized in the script by using a step function to increase attenuation of all its links at time T until it is disconnected. Same can be done to add new nodes or adding background noise to some links using a normal distribution attenuation pattern. The integrated signal generator of the sites can be used to act as a constant noise source. By changing quality of the links from these noise sources to each site we can control amount of noise the UUT inside that site experiences.

The other parameters that the testbed will ask from user are the number of times the experiment has to run, and the image to be flashed on all UUTs. It is worth mentioning that the user only compiles one generic image, and the testbed adjusts it for each UUT based on experiment setup. Once all sites are flashed and have received the experiment details formatted as an XML file, the experiment starts. 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 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, it will notify the Control PC. The Control PC will then terminate the experiment early, and notify the user that a problem has occurred.

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 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 UUT can be connected. 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.

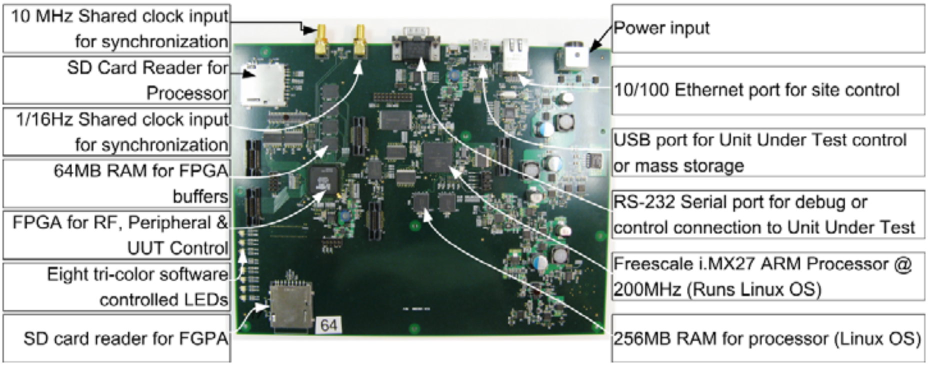

Fig. 3. One Site Board. Each Site Board is paired with one RF board

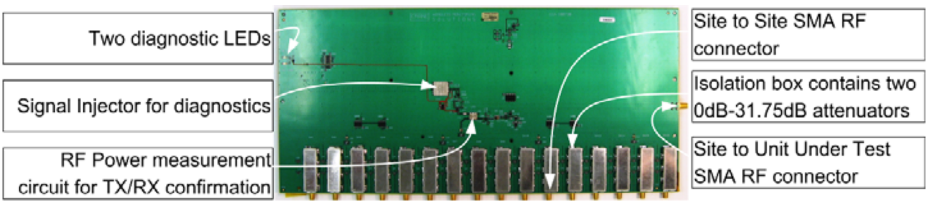

Fig. 4. One RF Board

Continuing with the graph theoretic description, the digital and RF boards together correspond to a node with a maximum of sixteen incident edges. The 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. The quality of links can be manipulated by changing the level of signal attenuation as rapidly as one db every millisecond.

4.2 Software Architecture

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 in the database 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.

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 formula 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. If the topology mapper slice is successful, 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. which are then conveyed to the user via the user interface.Experiment control slice instructs the reservation slice to release all the sites reserved for the experiment.

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 deterioration 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. 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 does all the calculations in a central location for all nodes, our solution is decentralized as we are able to break down the attenuation changes into tasks for each site. 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. 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. Kotz, D., Newport, C., Gray, R.S., Liu, J., Yuan, Y., Elliott, C.: Experimental Evaluation of Wireless Simulation Assumptions. In: Proceedings of MSWiM 2004 (2004)
2. Pawlikowski, K., Jeong, H., Lee, J.: On Credibility of Simulation Studies of Telecommunication Networks. IEEE Communications Magazine 40(1) (2002)

3. Cavin, D., Sasson, Y., Schiper, A.: On The Accuracy of MANET Simulators. In: Proceedings of the Second ACM International Workshop on Principles of Mobile Computing (2002)
4. Skehill, R., Scully, P., McGrath, S.: Characteristics, Results and Findings of IEEE 802.11 in an RF Isolated Testbed. In: Proceedings of PIMRC 2007 (2007)
5. White, B., Lepreau, J., Guruprasad, S.: Lowering the Barrier to Wireless and Mobile Experimentation. In: Proceedings of HotnetsI (2002)
6. Ertin, E., Arora, A., Ramnath, R., Naik, V., et al.: Kansei: A Testbed for Sensing at Scale. In: Proceedings of the Fifth International Conference on Information Processing in Sensor Networks (2006)
7. Dutta, P., Hui, J., Jeong, J., Kim, S., Sharp, C., Taneja, J., Tolle, G., Whitehouse, K., Culler, D.: Trio: enabling sustainable and scalable outdoor wireless sensor network deployments. In: Proceedings of IPSN 2006 (2006)
8. Arora, A., Ramnath, R., Ertin, E., et al.: Exscal: Elements of an extreme scale wireless sensor network. In: Proceedings of 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (August 2005)
9. Soda Hall wireless and sensor network testbeds, `http://www.millennium.berkeley.edu/sensornets/`
10. De, P., Raniwala, A., Sharma, S., cker Chiueh, T.: Design Considerations for a Multihop Wireless Network Testbed. IEEE Communications Magazine 43(10) (2005)
11. Werner-Allen, G., Swieskowski, P., Welsh, M.: MoteLab: A Wireless Sensor Network Testbed. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (2005)
12. Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., et al.: Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols. In: IEEE Wireless Communications and Networking Conference 2005, vol. 3 (2005)
13. Kaul, S., Gruteser, M., Seskar, I.: Creating Wireless Multi-hop Topologies on Space-Constrained Indoor Testbeds Through Noise Injection. In: TridentCom 2006 (2006)
14. De, P., Raniwala, A., Krishnan, R., Tatavarthi, K., et al.: MiNT-m: An Autonomous Mobile Wireless Experimentation Platform. In: Proceedings of the 4th International Conference on Mobile Systems, Applications and Services (2006)
15. Johnson, D., Stack, T., Fish, R., Flickinger, D.M., Stoller, L., Ricci, R., Lepreau, J.: Mobile Emulab: A Robotic Wireless and Sensor Network Testbed. In: Proceedings of INFOCOM 2006 (April 2006)
16. Stovall, D., Paine, N., Petz, A., Enderle, J., Julien, C., Vishwanath, S.: Pharos: An Application-Oriented Testbed for Heterogeneous Wireless Networking Environments. The University of Texas at Austin, Tech. Rep. TR-UTEDGE-2009-006 (2009)
17. Judd, G.: 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. Clancy, T., Walker, B.: MeshTest: Laboratory-Based Wireless Testbed for Large Topologies. In: TridentCom 2007 (2007)
19. Lee, H., Cerpa, A., Levis, P.: Improving Wireless Simulation Through Noise Modeling. In: Proceedings of IPSN 2007, p. 30 (2007)
20. Burchfield, R., Nourbakhsh, E., Dix, J., Sahu, K., Venkatesan, S., Prakash, R.: RF in the Jungle: Effect of Environment Assumptions on Wireless Experiment Repeatability. In: IEEE International Conference on Communications, ICC (2009)