Computer Communications 73 (2016) 99–107

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Wireless Networking Testbed and Emulator (WiNeTestEr)

J.D. Beshay[a,*], K.S. Subramani[a,*], N. Mahabeleshwar[a], E. Nourbakhsh[a], B. McMillin[a], B. Banerjee[a], R. Prakash[a], Y. Du[b], P. Huang[b], T. Xi[b], Y. You[b], J.D. Camp[b], P. Gui[b], D. Rajan[b], J. Chen[c]

[a] The University of Texas at Dallas, Richardson, TX, USA
[b] Southern Methodist University, Dallas, TX, USA
[c] The University of Arizona, Tucson, AZ, USA

ARTICLE INFO

Article history:
Available online 18 August 2015

Keywords:
Wireless
RF
Channel emulation
Testbed

ABSTRACT

Repeatability, isolation and accuracy are the most desired factors while testing wireless devices. However, they cannot be guaranteed by traditional drive tests. Channel emulators play a major role in filling these gaps in testing. In this paper we present an efficient channel emulator which is better than existing commercial products in terms of cost, remote access, support for complex network topologies and scalability. We present the hardware and software architecture of our channel emulator and describe the experiments we conducted to evaluate its performance against a commercial channel emulator.

1. Introduction

The ability to conduct repeatable experiments is crucial to the development of wireless devices and protocols. Most of the time, researchers make simplifying assumptions about the nature of their test environment and the experiment control procedures. However these assumptions do not always hold good [1] making it harder to isolate device/protocol performance from environmental effects.

The current wireless networking testbeds use a wide range of approaches, varying from fully software-simulated testbeds like ns-3 [2] to real hardware running in Faraday cages. However, the two extremes have their respective limitations. Simulators are easy to implement but they are limited by the models provided in software. Different simulators might yield different results depending on the assumptions and simulation techniques used [3]. On the other hand using hardware in Faraday cages is not always affordable and it is not flexible enough to test complicated scenarios involving mobility or signal reflections. The balance between the two extremes is provided by channel emulators.

The simplest form of channel emulation is achieved by replacing the antennae of two communicating transceivers with shielded RF cables and a programmable attenuator in between. An example of an emulator based on this design is ASSERT [4], developed by our group a few years back. By increasing (or decreasing) the attenuation it simulates the transceivers moving apart (or moving closer). The rate at which attenuation is varied corresponds to the relative speed of the transceivers. This scenario covers the fading effect of wireless channels.

However, wireless transmissions are not only affected by fading but also by multiple reflections of the same signal from obstacles in the surrounding environment (multi-path effects). Devices with multiple antennae (MIMO) exploit these signal reflections to achieve better throughput. Accurate emulation of multi-path effects requires creating multiple copies of the transmitted signal with different time delays (phases). This cannot be achieved by attenuators. Instead it is done by digitizing the signal and manipulating it using digital signal processing (DSP). The resulting digital signal is converted back to analog.

Commercial solutions exist for emulating environments with multi-path effects [5,6]. However they are prohibitively expensive and are limited to simulating environments with 2 pairs of devices or less. Commercial channel emulators are thus impractical for researchers who are usually cost-constrained and interested in experiments that involve the interaction (and interference) between multiple devices with a higher degree of connectivity. As a result, researchers sought to develop their own channel emulators that could achieve multipath effects for multiple devices while maintaining relatively low cost. An example is the work in [7] which uses a single field programmable gate array (FPGA) to simulate a 90 MHz-wide environment for up to 15 devices operating in the 2.4 GHz ISM band. The design in [7] cannot scale due to the FPGA resource constraints.

In this paper, we present our Wireless Networking Testbed and Emulator (WiNeTestEr) which is designed to emulate 100-MHz-wide

* Corresponding authors.
E-mail addresses: joseph.beshay@utdallas.edu (J.D. Beshay), kiruba.subramani@utdallas.edu (K.S. Subramani).

http://dx.doi.org/10.1016/j.comcom.2015.08.007

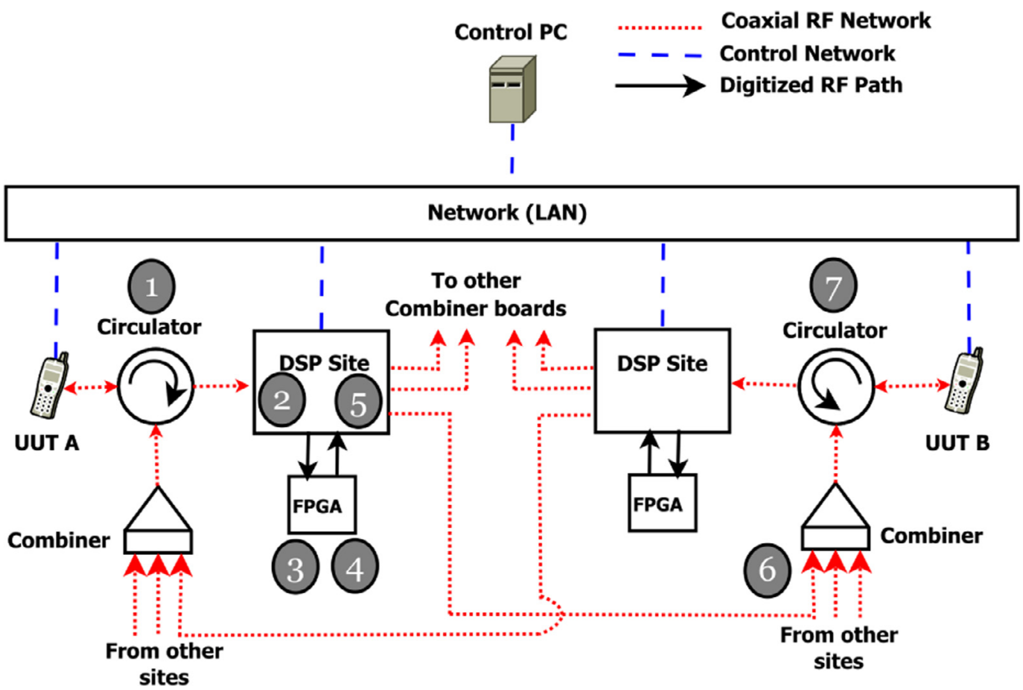

Fig. 1. A simple topology of two UUTs connected by a bidirectional link.

environments with multipath in the 2.4 GHz ISM band. The main features of WiNeTestEr are:

- Scalability: the system uses a distributed channel emulation architecture running on multiple FPGAs so it can potentially scale to hundreds of nodes,
- remote access: a device control protocol allows the system to run experiments without onsite-operator intervention,
- concurrent experiments: the modular design allows for multiple independent experiments to be run by different users at the same time,
- multi technology support: experiments can be performed on different technology devices operating in their native frequencies (Bluetooth, WiFi, Zigbee, etc.). The design is flexible to allow adding more frequency bands in the future with minimal changes,
- full duplex channels: the channel between two devices is full duplex with support for non-reciprocal channel conditions i.e. the signal can experience a certain environment in one direction and a different one in the other.

WiNeTestEr is our followup to ASSERT [4]. ASSERT performs channel emulation in the 900 MHz ISM band using attenuation. Attenuators are used to control the transmitted signal strength to emulate the required channel conditions (deep fade, slow fade) but it cannot emulate multi-path effects. WiNeTestEr was developed to bridge this gap.

2. System overview

WiNeTestEr is a DSP-based channel emulator. The system is built around the following core function: RF signals generated by a transmitter are digitized, processed in the digital domain, converted to analog and delivered to the receiver. The processing phase can apply attenuation and amplification to emulate fading, or create multiple copies of the signal with different signal levels and phases to emulate multipath effects.

WiNeTestEr can be logically broken into three parts: units under test (UUTs), DSP sites and the Control PC. UUTs are the devices with RF transceivers. Signals traveling between pairs of devices are manipulated to provide the desired channel conditions for the experiment being conducted. The device antennae are removed and replaced with coaxial cables connected to the DSP sites.

DSP sites, we call them sites for short, are the heart of channel emulation. A site takes the signal transmitted by a UUT, manipulates it in the digital domain using an FPGA to apply the specified channel conditions and delivers the resulting analog signal to the receiving UUT via a coaxial RF cable. This constitutes a uni-directional link between two UUTs. Bidirectional communication is achieved by using multiple sites. The interconnection of UUTs in the emulated environment is realized by coaxial cables between sites. This implies that a transciever needs to have at least two coaxial cables connected to its antenna port; one for outgoing and the other for incoming signals. The higher the degree of connectivity the more cables that need to be connected to the UUT. This is achieved by using circulators and combiner boards. A circulator has three terminals; signal from terminal #1 goes to #2, #2 to #3 and #3 to #1. A combiner board takes a number of input signals over different coaxial cables and combines them into one output signal over a single coaxial cable. The desired connectivity is achieved by connecting the UUT antenna port to terminal #1 of the circulator. The transmitted signal comes out on #2 which goes to the site. The combiner board takes all received signals and passes them to #3 which comes out on #1 and goes into the UUT. This setup is illustrated in Fig. 1.

The Control PC is what glues the whole system together. WiNeTestEr software is highly distributed with different subcomponents running on UUTs, sites and a remote GUI running on user machines. These subcomponents are all connected to the Control PC over an IP network. The Control PC is responsible for:

- Keeping track of the status of all UUTs and sites.
- Keeping track of the environment topology (cable interconnections between sites and UUTs).
- Accepting requests to run experiments from users.
- Coordinating sites and UUTs to run the desired experiments.
- Collecting results and delivering them to the user.

Another approach to understanding the design of WiNeTestEr is to consider the hardware and software aspects separately. Fig. 1 shows how the hardware is physically connected to emulate a bidirectional