Environment-Independent Virtual Wireless Testbed

Hiroshi Mano

Interdisciplinary Graduate School of Medicine and Engineering,
University of Yamanashi, 4-4-37 Takeda, Kofu, Yamanashi 400-8510, Japan
`hiroshi@manosan.org`

Abstract. Current digital packet-switching communications involves more cross-layer trade-offs than legacy wireless communications. Conventionally, each network layer is studied separately, and then the total system is evaluated in field tests. However, the results are always specific to the particular environmental conditions during the test and are not reproducible. Here we propose an architecture for an environment-independent virtual wireless testbed. We describe the implementation and validation results for a programmable wireless propagation emulator—the key component of the testbed. It was found that the system could emulate wireless propagation over at least 10 wireless nodes in real time using 10 field-programmable gate arrays with programmable parameters consisting of the channel model, path loss model, antenna model, and emulation scenario. Thus, the system is promising as a replacement for conventional field tests.

Keywords: Wireless testbed, Wireless emulator, Virtual networking.

1 Introduction

Today, digital broadband communication systems often rely on a combination of wireless and wired network technologies. Furthermore, systems are rapidly switching from single-purpose (such as voice specific) to general-purpose (IP) services by adopting packet switching. These packet-switching systems employ more cross-layer collaboration than conventional networks, such as TCP retransmission for quality of service (QoS) control and Mobile IP for transparency.

Conventionally, the wireless layer is evaluated without taking into consideration the upper-layer protocols, which are largely evaluated with wireless simulators or specific existing wireless technologies (i.e., LTE, Wi-Fi, or Wi-MAX). Therefore, a comprehensive evaluation of any new network technology requires a complete field test. However, the results of any field test are always specific to the environmental conditions during the test (geographic location, physical specifications, MAC, link-layer protocols, radio licensing terms, etc.) and are therefore not easy to reproduce.

Here we describe an environment-independent virtual wireless testbed that is fully programmable, from the physical layer to the application layer, and can interface with external network devices in real time while allowing easy reproducibility. In this section, we describe the current state of today's wireless testbed and the advantages of

T. Korakis, M. Zink, and M. Ott (Eds.): TridentCom 2012, LNICST 44, pp. 33–47, 2012.

an environment-independent system. Then, in Sections 2 and 3, we describe the proposed architecture and an implementation of the main component, the programmable wireless propagation emulator. Section 4 presents the results of an evaluation of the emulator, and Section 5 lists our conclusions.

1.1 Current State of the Art

In the field of network research, several wireless testbeds have been developed, but none allow complete environment-independent evaluation without a field test. GNURADIO [1], for example, is a software-defined radio (SDR) that provides programmable capability for the PHY and MAC layers. However, it has specific RF units. This necessitates the use of a propagation simulator or a field test for total evaluation. The ORBIT Radio Grid Testbed [2] provides an evaluation environment for upper-layer protocols such as for mesh networks comprising established IEEE802.11 wireless LANs and IEEE802.16 Wi-MAX stations on a grid. However, only a fixed environment is provided for path loss and delay spread, and the lower network layers (MAC and PHY) are not reconfigurable. In addition, the CMU wireless emulator [3] includes a digital signal processor (DSP) for user-defined propagation between specific wireless units in real time, but it does not include wireless radios.

Thus, the existing wireless testbeds are restricted to particular environments (propagation, channel, etc.), device characteristics (frequency, power, modulation/demodulation, antenna, etc.), and upper-layer protocols (multiplex, duplex, packet congestion control, etc.). The consequences of these restrictions are quite significant in practice. For example, we spent much time and money to test a high-speed hand-over protocol over Wi-Fi at an actual motor sports course with a professional driver and a racing car. However, this only demonstrated the performance under a particular set of conditions (2.4 GHz, DSS, single vehicle, and closed-circuit environment). If we needed to perform the same evaluation using different preconditions (OFDM, 5.8 GHz, multiple vehicles, etc.), we would have to spend the same amount of resources again.

1.2 Advantages of Virtual Wireless Testbed

The proposed environment-independent virtual wireless testbed is fully programmable from the physical layer to the application layer and can interface with external network devices in real time for easy reproducibility. Because of the wide adaptability of the proposed architecture, there are numerous prospective uses with no regulatory limitations. This will allow users to evaluate their own new concepts without requiring radio licenses and test fields. The use case shown in Fig. 1 is one possible example. The testbed is a typical use case of network virtualization.

Most new ideas on the future of the Internet are evaluated over the testbed that has been deployed over a layer provided by virtual technologies.

The adaptivity of the virtual network provides an effective evaluation environment. However, only the wireless part depends on particular technologies.

Therefore, we hope to use this virtual wireless testbed as part of a virtual network that provides network service functions such as flow control, mobility control, and authentication.

For example, with a heterogeneous network system with a different wireless network, a user can configure each wireless node to employ different wireless technologies. In addition, the propagation environment is also programmable; this makes it easy to evaluate systems in different environments such as urban, rural, and indoor settings.

Although this system does not dispense with all experimental tests, the user can reproduce the experimental environment by inputting the data gathered during the experiment. In addition, and most importantly, the user only needs to program or configure the target part to be evaluated—all other parts would be described by ideal parameters. For example, if a user wants to evaluate a new concept for an antenna, it is only necessary to program the antenna characteristics, while selectable predetermined values are provided for all of the other layers: PHY, MAC, LINK, and transport.

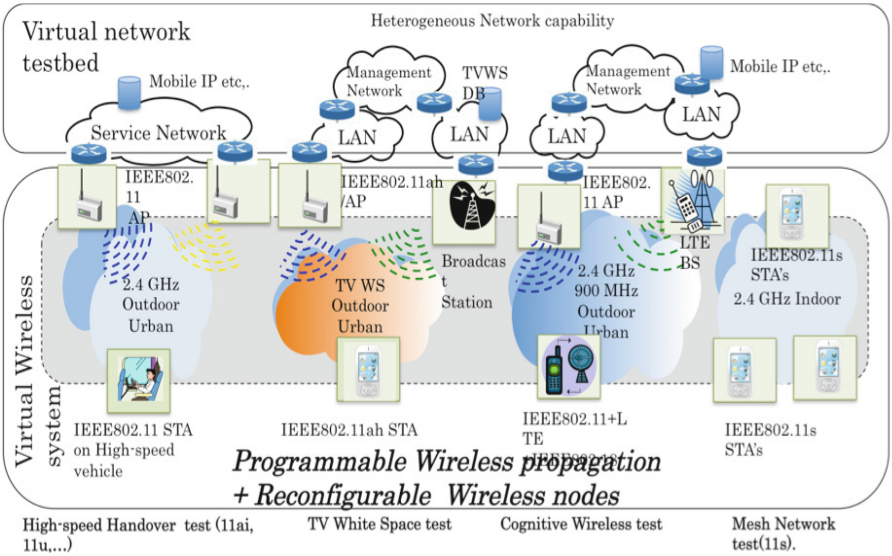


Fig. 1. Prospective use case for proposed virtual wireless testbed

2 Proposed Architecture

As shown in Fig. 2, the proposed testbed incorporates a scenario generator, several reconfigurable wireless nodes, and a programmable wireless propagation emulator.

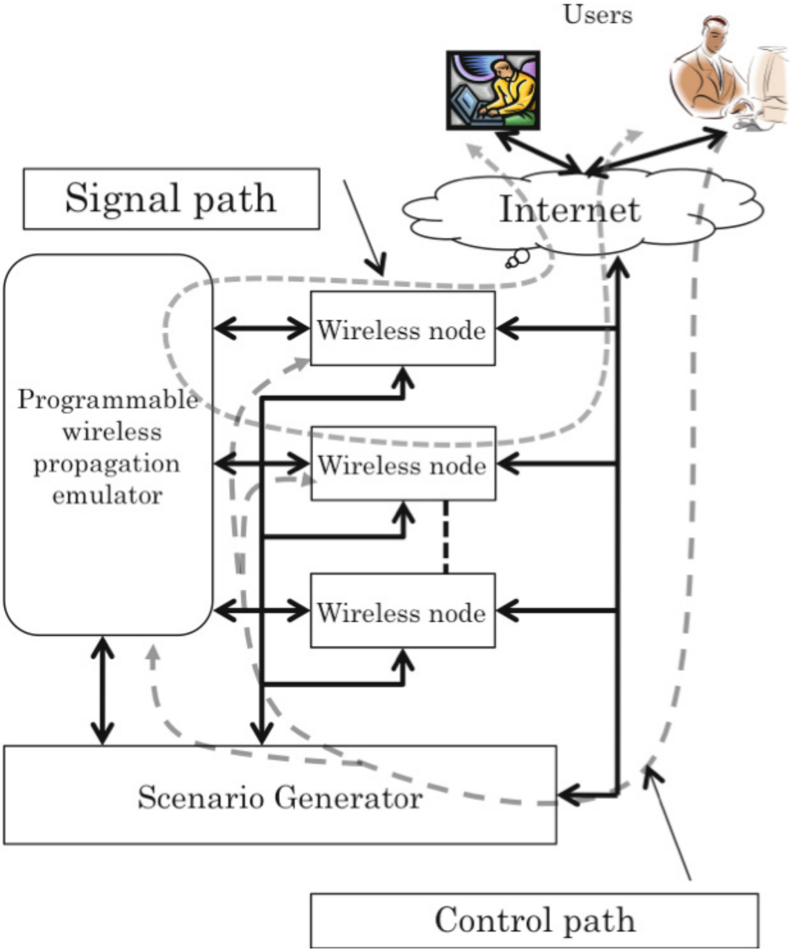


Fig. 2. Architecture of environment-independent virtual wireless testbed

2.1 Scenario Generator

The scenario generator provides the user interface, allows resource management (setting/selecting parameters and scenarios), and calculates the dynamic propagation scenario that would be executed by the emulator. A commercial off-the-shelf personal computer is used as the scenario generator.

2.2 Wireless Nodes

The wireless nodes, as shown in Fig. 3, are implemented on field-programmable gate arrays (FPGAs) and provide a fully reconfigurable PHY, MAC, and Frontend (roll-off

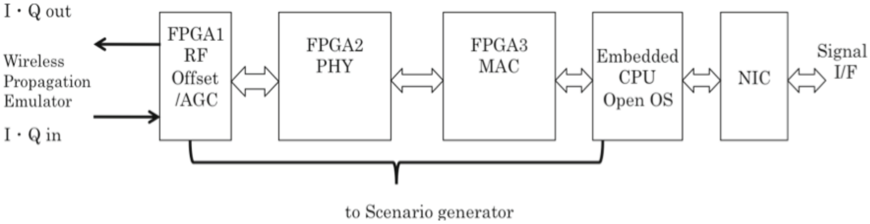


Fig. 3. Reconfigurable wireless node

filter, AGC, etc.). In addition, they incorporate an embedded computer, which is used for the Link or higher-layer control. For a virtual network, a user may also implement a virtual host on this embedded computer.

2.3 Programmable Wireless Propagation Emulator

As shown in Fig. 4, this system emulates the propagation path between wireless nodes. The channel and path loss models are implemented using a finite impulse response (FIR) filter on FPGA hardware. All wireless propagation behavior can be described as a combination of delay, amplitude, and phase. The channel model is implemented with a seven-tap FIR that also emulates frequency-selective fading. The multiplication value (*Kn* and *dn* in Fig. 4) for the signal is calculated beforehand using the scenario generator, and the values are updated based on the time and the user-defined scenario. All the wireless nodes are connected to each other through this propagation emulator using I/Q complex signals. Therefore, $\left(2^n - n\right)$ propagation paths are required for this propagation emulator.

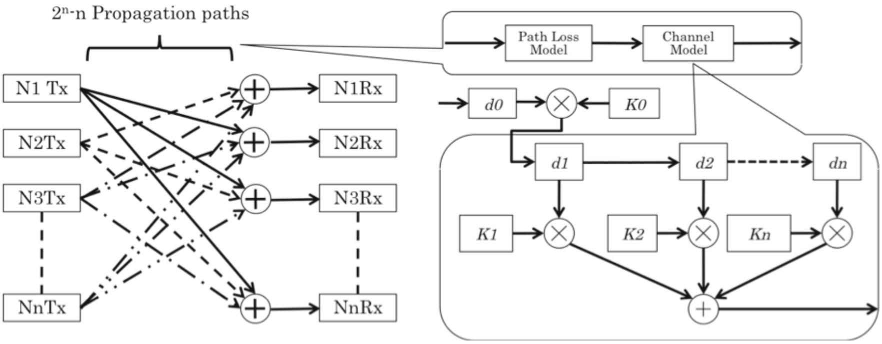


Fig. 4. Programmable wireless propagation emulator

3 Implementation of Programmable Wireless Propagation Emulator

We implemented the scenario generator and programmable wireless propagation emulator for 10 wireless nodes. The scenario generator was implemented in software on a Linux system, and the programmable wireless propagation emulator was implemented using a hardware board that integrates FPGAs. Gigabit Ethernet connects the scenario generator and programmable wireless propagation emulator, as shown in Fig.5.

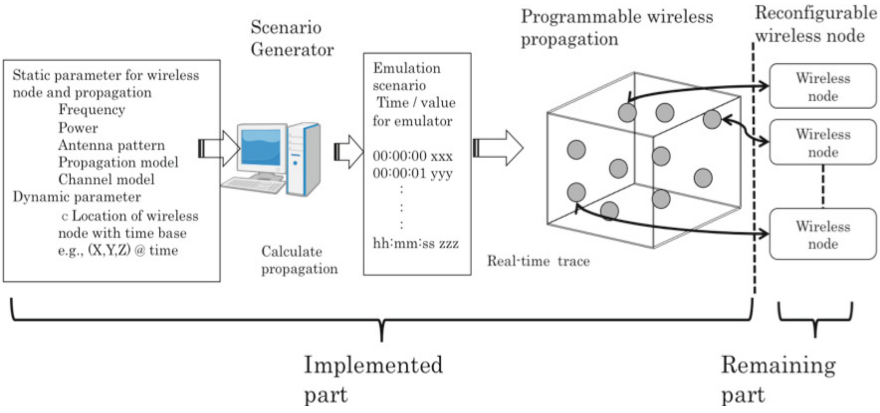


Fig. 5. System overview

3.1 Scenario Generator

The scenario generator software comprises the following components:

3.1.1 GUI

We implemented the entire user interface in Adobe Flash and PHP, and considered system sharing. This GUI provides a user-friendly interface that shows the scenario behavior and all the parameters of the wireless nodes.

3.1.2 Propagation Characteristics Generation

The scenario generator calculates the propagation fluctuation parameter using the conditions provided by the user through the GUI. The implemented prototype supports the configurable conditions for the propagation path shown in Table. 1.

Table 1. Configurable conditions

Path loss model	Channel ITU-R M.1225	Fading Model	Antenna
Two-way ray	Indoor A/B	Classic	Isotopic
COST-231Hata	Pedestrian A/B	Flat	Dipole
	Vehicular A/B	User defined	ITU-R M.2135
	PDSekm		ITU-R M.1245
	User defined		User defined

3.1.3 Path Loss Calculation

The scenario generator calculates the path loss between nodes using the conventional formulas described in the following. In the case of the two-ray model, the path loss is given by (1) for sufficiently short distances and by (2) otherwise.

$$L = 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) \tag{1}$$

$$L = 40\log_{10}(d) - 20\log_{10}(h_t) - 20\log_{10}(h_r) \tag{2}$$

For the COST-231 Hata model, the path loss is given by

$$\begin{aligned} &L = 46.3 + 33.9\log(f) - 13.82\log(h_t) - a(h_r) + \left[44.9 - 6.55\log(h_t)\right]\log(d) + 3 \\ &a(hr) = (1.1\log(f) - 0.7)h_r - (1.56\log(f) - 0.8) \end{aligned} \tag{3}$$

where L represents the path loss in decibels; d represents the distance, λ represents the wavelength, and h_t *and* h_r are the antenna heights all in meters; and f denotes the frequency in megahertz.

3.1.4 Fading Calculation

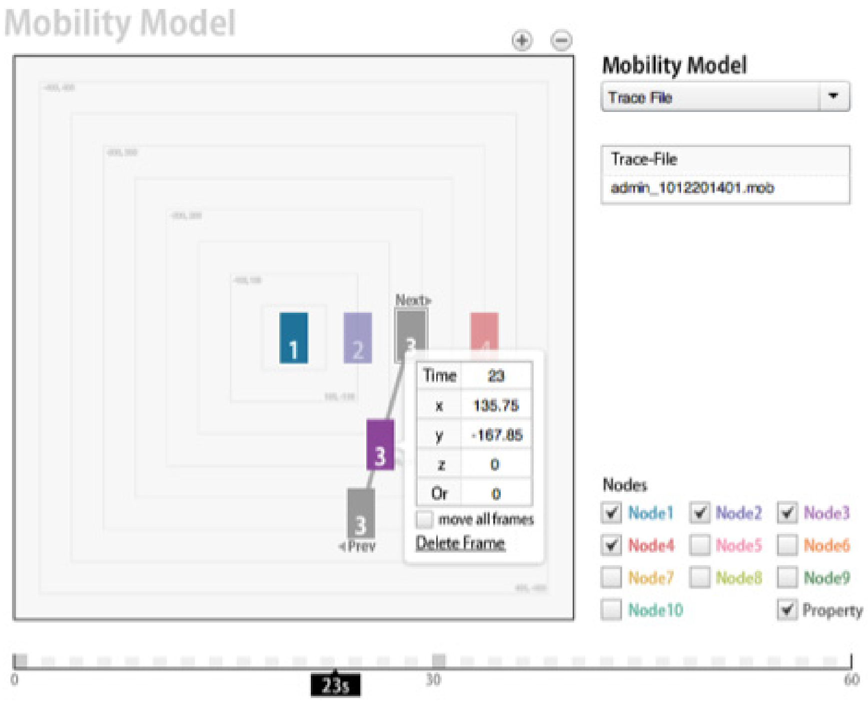
The scenario generator calculates the vector data for the Rayleigh fading wave $r(t)$ as

$$\begin{aligned} &r(t) = x(t) + j * y(t) \\ &= \left[\sqrt{\frac{2}{N_1+1}}\sum_{n=1}^{N_1}\sin\left(\frac{\pi n}{N_1}\right)cos\left\{2\pi f_d cos\left(\frac{2\pi n}{N_1}\right)t\right\} + \frac{1}{\sqrt{N_1+1}}cos\left(2\pi f_d t\right)\right] \\ &+ j\sqrt{\frac{2}{N_1}}\sum_{n=1}^{N_1}\sin\left(\frac{\pi n}{N_1}\right)cos\left\{2\pi f_d cos\left(\frac{2\pi n}{N_1}\right)t\right\} \end{aligned} \tag{4}$$

where d is the maximum fading frequency in hertz and N_1 is the number of FIR taps.

3.1.5 Emulation Scenario

The emulation scenario is defined by the time and location (*x*, *y*, *z*, and azimuth) of the wireless nodes using the GUI, as shown in Fig. 6. The scenario generator calculates the time-based fluctuation of the propagation using the process described above.



Mobility Model: Trace File

Trace-File: admin_1012201401.mob

Time	23
x	135.75
y	-167.85
z	0
Or	0

☐ move all frames

Nodes

☑ Node1 ☑ Node2 ☑ Node3
☑ Node4 ☐ Node5 ☐ Node6
☐ Node7 ☐ Node8 ☐ Node9
☐ Node10 ☑ Property

Fig. 6. Emulation scenario GUI

3.2 Programmable Wireless Propagation Emulator

The programmable wireless propagation emulator was implemented in hardware using 10 FPGAs (Vertex-6 XC6VLX130T, 1156 pin, 200 MHz). We implemented this 10-node capability in the virtual wireless space with a 49% slice, 74% DSP, and 60% RAM consumption.

After calculating the vector data of the path loss and channel model using the scenario generator, the data were transferred to the programmable wireless propagation emulator over Gigabit Ethernet. The programmable wireless propagation emulator operates in real time by combining the delay memory and complex multipliers, as shown in Fig. 7.

In this proposed architecture, 215 adders and 360 multiplexers emulate wireless propagation with 200-MHz clocks. In addition, the required number of I/O pins *P* between the wireless nodes and propagation emulator is given by

$$P = n \times N \times k \tag{5}$$

where n is the number of wireless nodes, N is the bit depth or signal resolution, and $k = 4$ (I,Q, +,-).

In other words, the scalability of the proposed architecture depends on the clock speed of the FPGA, signal resolution, and the number of I/O pins.

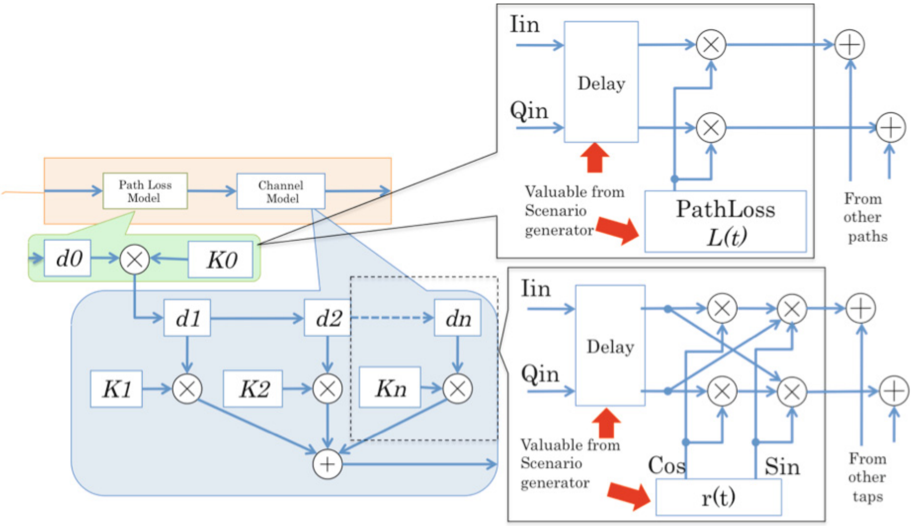


Fig. 7. Block diagram of programmable wireless propagation emulator

4 Evaluation of Programmable Wireless Propagation Emulator

We evaluated the accuracy, process delay, and behavior of the implemented programmable wireless propagation emulator. In this section, the evaluation methodology is explained first, and then the results of the evaluation are described.

4.1 Evaluation Methodology

Here we used existing legacy radio measurement equipment such as a spectrum analyzer, signal generator, and commercial Wi-Fi products. Because the programmable wireless emulator only had digital I/Q interfaces, it was not amenable to visible inspection. Therefore, we employed analog-digital (A/D) and digital-analog (D/A) converters for use between the measurement equipment and programmable wireless emulator. In addition, the throughput was measured with iPerf on a personal computer connected to the both ends of the wireless LAN device.

4.2 Path Loss Characteristics

We measured the throughput between 2.4-GHz wireless LAN access points (AP) and stations (STA) using the emulator and an RF attenuator, as shown in Fig. 8. The results of this measurement showed that differences exist between the emulator and RF attenuator, as pointed in Fig.9.

The difference in the low-attenuation range is caused by saturation of the analog circuit, and the difference in the high-attenuation range is caused by noise in the analog circuit. As a result, we confirmed that the implemented emulator can replace the RF circuit in adding attenuation to the signal path with the desired scenario but does not have the characteristics of analog circuits. Further, we confirmed that the accuracy of attenuation to a static signal was +/- 1.0dB in comparison to the RF attenuator. In addition, the signal delay of the emulator was confirmed to be only 1 μs by pulse reply.

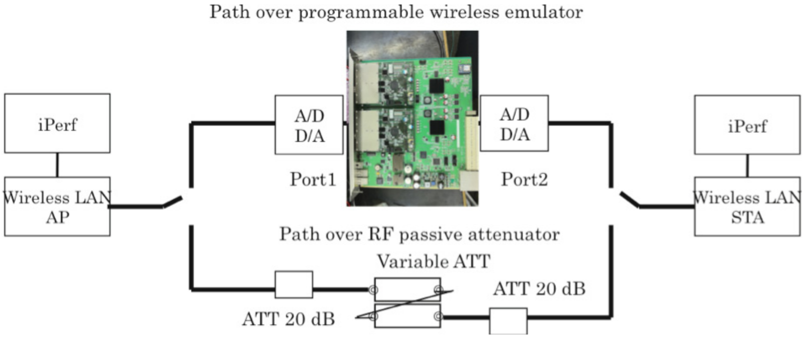


Fig. 8. Measurement scheme for path loss characteristics

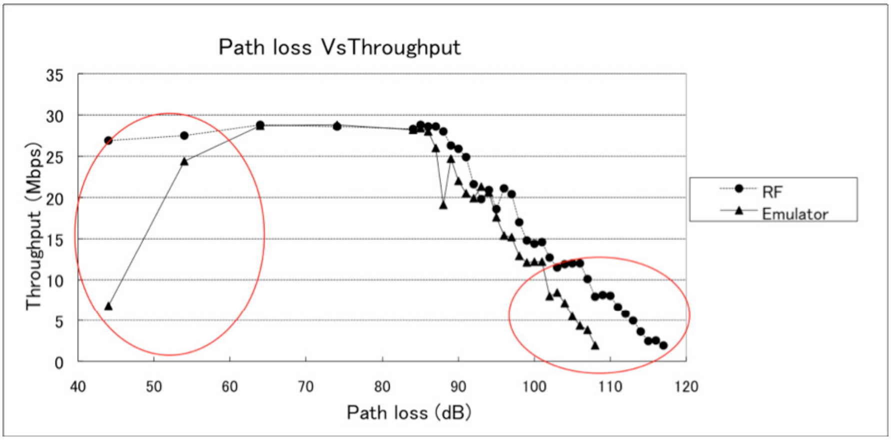


Fig. 9. Throughput for path loss

4.3 Desired/Undesired Signal Characteristics

We measured the desired/undesired (D/U) ratio at 10% outage throughput of a wireless LAN link with an added undesired CW or OFDM signal, as shown in Fig. 10. From the measurement results shown in Fig. 12, we see that the D/U characteristics over the emulator and RF attenuator are in close agreement. In addition, it was confirmed that the emulator added signals with different propagation paths.

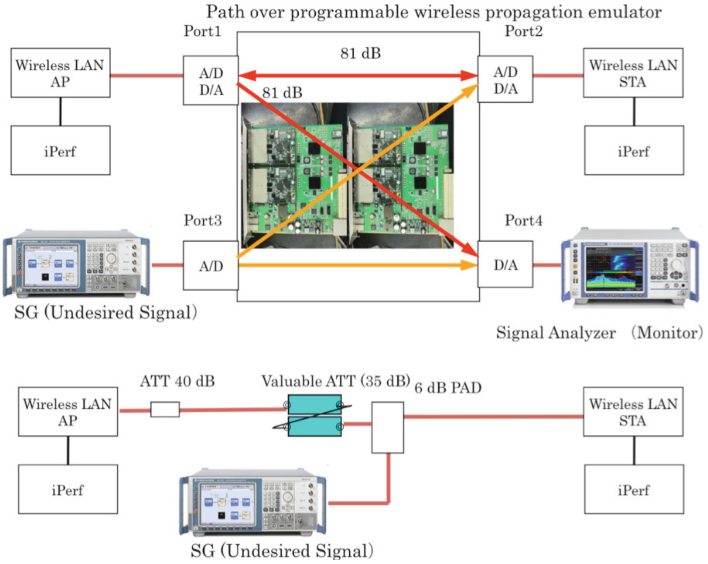


Fig. 10. Measurement scheme for D/U signal ratio

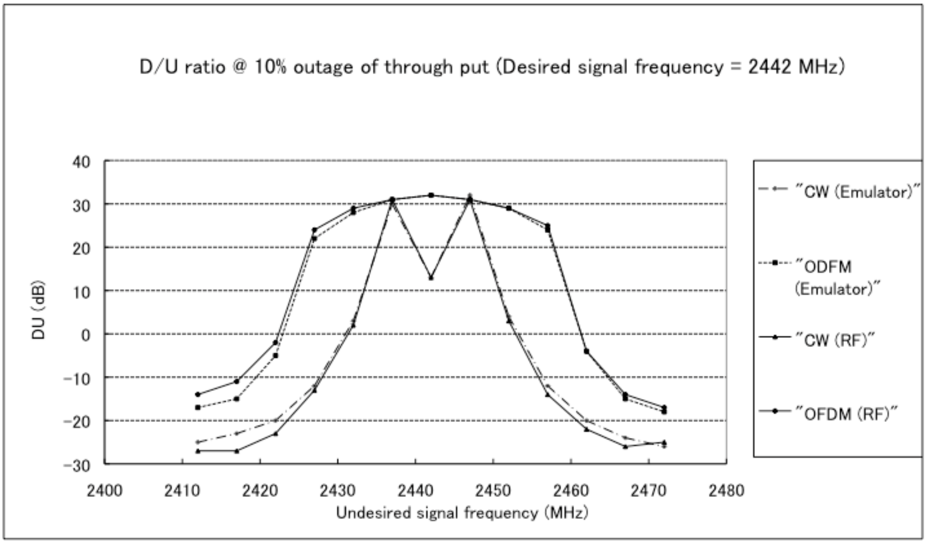


Fig. 11. D/U signal ratio over programmable wireless propagation emulator

4.4 Fading Signal Characteristics

Using the same measurement scheme as that shown in Fig. 10, we added a particular fading channel model to the wireless LAN link between the AP and STA. By monitoring the signal at port 4, we confirmed that a signal occurred over the propagation path, as shown in Fig. 13.

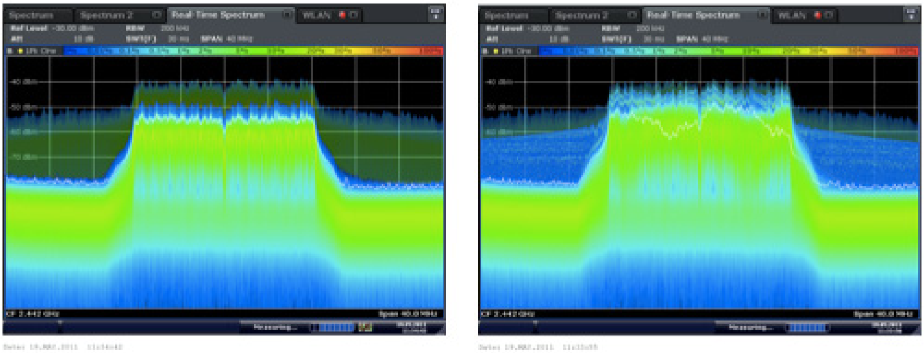
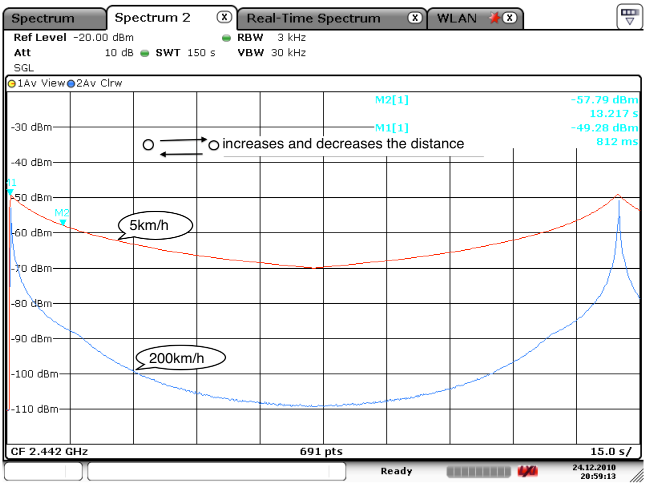


Fig. 12. Fading effect: flat/classic

4.5 Dynamic Emulation of Moving Station

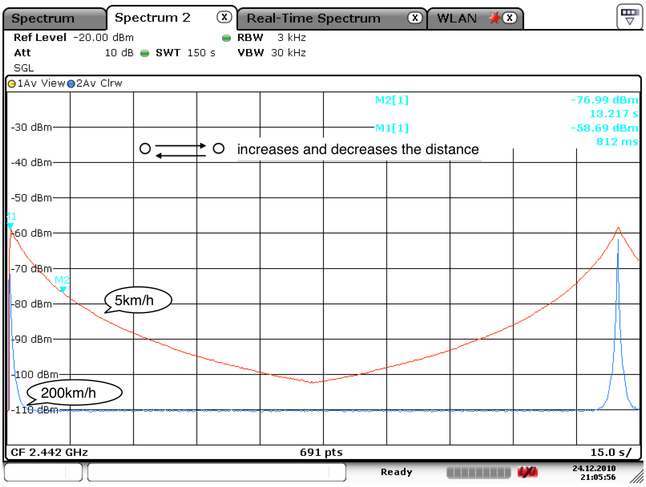
We programmed a vehicle scenario in which the distance between two wireless nodes increases and decreases with time. We then measured the attenuated signal level with the CW signal as shown in Fig. 14 and Fig. 15.



Ref Level -20.00 dBm
Att 10 dB
SWT 150 s
RBW 3 kHz
VBW 30 kHz
SGL
1Av View 2Av Clrw
M2[1] -57.79 dBm 13.217 s
M1[1] -49.28 dBm 812 ms
CF 2.442 GHz
691 pts
15.0 s/
Ready
24.12.2010 20:59:13

X: 15 Sec/Div Y: Received signal level 10dB/Div

Fig. 13. Signal output level for vehicle scenario with two-way ray model



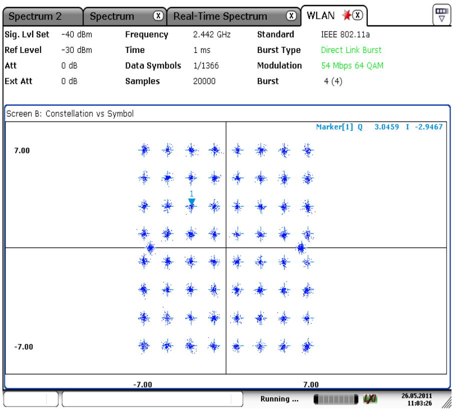
Ref Level -20.00 dBm
Att 10 dB
SWT 150 s
RBW 3 kHz
VBW 30 kHz
SGL
1Av View 2Av Clrw
M2[1] -76.99 dBm 13.217 s
M1[1] -58.69 dBm 812 ms
CF 2.442 GHz
691 pts
15.0 s/
Ready
24.12.2010 21:05:56

X: 15 Sec/Div Y:Received signal level 10dB/Div

Fig. 14. Signal output level for vehicle scenario with COST-231 Hata model

4.6 Error Vector Magnitude Measurement

We measured the internal effect on the error vector magnitude (EVM) using an ideal OFDM signal. Table 2 lists the measured EVM data for the original signal and the signal after passing through the emulator path shown in Fig. 16. As seen from these measured data, the EVM indeed degraded. However, it is reasonable to assume that these changes were caused by the characteristics of the analog conversion because the emulator performs only digital operations, and the typical digital operation error is not as small as that measured. In addition, the results showed that there was no significant quality change that would violate the IEEE802.11 standard.



Sig. Lvl Set -40 dBm
Ref Level -30 dBm
Att 0 dB
Ext Att 0 dB
Frequency 2.442 GHz
Time 1 ms
Data Symbols 1/1366
Samples 20000
Standard IEEE 802.11a
Burst Type Direct Link Burst
Modulation 54 Mbps 64 QAM
Burst 4 (4)
Screen B: Constellation vs Symbol
Marker[1] Q 3.0459 I -2.9467
Running ...
26.05.2011 11:03:26

Date: 26.MAY.2011 11:03:25

Fig. 15. EVM through programmable wireless propagation emulator

Table 2. EVM of programmable wireless propagation emulator

Item	Measurement point		Difference	Unit	IEEE802.11 Limit
	Input	Output			
EVM All Carr.	1.03	2.95	+1.92	%	5.62
	-39.76	-30.61	-9.15	dB	-25.00
EVM Data Carr.	1.04	2.98	+1.90	%	5.62
	-39.69	-30.53	-8.84	-25.00	-25.00
EMV Pilot Carr	0.91	2.57	-1.6	%	39.81
	-40.77	-31.80	-9.46	dB	-8.00
IQ Offset	-69.50	-58.08	-11.42	dB	-15.00
Gain Imbalance	-0.09	-0.07	-0.02	%	-
	-0.01	-0.01	0	dB	-
Quadrature Err.	0.00	-0.04	0.02		
Freq Err	-149.13	-152.60	-3.60	Hz	+/-48840
Symb Clock Err.	0.09	-0.03	0.12	ppm	+/-20

5 Conclusions

In this paper, we proposed an environment-independent virtual wireless testbed architecture and reported the implementation of a wireless propagation emulator.

The results of the actual implementation of the programmable wireless propagation emulator showed the following:

- The wireless propagation emulator emulates wireless propagation with sufficient accuracy (±1.0 dB).
- The emulation delay is small enough for this system to be used as a real-time air interface (1.0 μs).
- The wireless propagation emulator emulates the typical behavior of flat fading and frequency-selective fading.
- Complex vector signals with at least an 85-MHz bandwidth can be emulated simultaneously.
- A 10-node emulator can be realized using existing commercial FPGA products.

The results of this implementation demonstrated that the combination of a scenario generator and the propagation emulator is capable of performing propagation emulation with wide adaptability. Compared with existing testbeds, the proposed programmable wireless propagation emulator can be implemented without depending on wireless node characteristics (PHY, MAC, Link, and upper layers).

The number of I/O pins of the FPGA limits the scalability to a large number of nodes because we divided each propagation module into 10 FPGAs. However, a greater integration of FPGAs may help solve this scalability problem by reducing the number of discrete components.

In conclusion, the technical feasibility of the proposed architecture for an environment-independent virtual wireless testbed was demonstrated through an actual implementation. The performance of the programmable wireless propagation emulator was sufficient for real-time applications. The reconfigurable wireless node, the last remaining part of the proposed testbed, will be implemented in subsequent work. In addition, we intend to evaluate our concept for a fast initial link setup protocol, which will be proposed as IEEE802.11ai, using the virtual wireless testbed. For IEEE802.11ai, we must evaluate a large number of STA-to-AP connections using a vehicle model, which is the typical use case for a virtual wireless testbed.

Acknowledgments. The assistance received from Tetsuya Shigeyasu, Hiroshi Matsuno, Akihiro Nakao, Marc Emmelmann, Masanori Hanawa, and Eiko Torkai in the preparation of this manuscript is gratefully acknowledged.

The author is grateful to Toshihisa Yamada, Satoshi Funada, Masanori Uno, Mineo Takai, Shigeru Kaneda, and Fred Okuma for their assistance with the implementation of the programmable wireless propagation emulator.

References

1. Blossom, E.: GNU Radio: Tools for exploring the radio frequency spectrum. Linux Journal 122 (June 2004)
2. Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., Siracusa, R., Liu, H., Singh, M.: Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols. In: Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC 2005 (2005)
3. Judd, G.: Using Physical Layer Emulation to Understand and Improve Wireless Networks, PhD Thesis, Department of Computer Science, Carnegie Mellon University, Also published as technical report CMU-CS-06-164 (October 2006)
4. Harada, H., Prasad, R.: Simulation and Software Radio for Mobile Communications. Artech House, Norwood (2002)
5. Mano, H., Yamada, T., Funada, S., Uno, M., Takai, M., Kaneda, S.: Wireless propagation emulator for virtual wireless testbed. In: IPSJ DPS Workshop (2011)