Grid Software Defined Radio Network Testbed for Hybrid Measurement and Emulation

Kapil R. Dandekar*, Simon Begashaw*, Marko Jacovic*, Alex Lackpour*,
Ilhaan Rasheed*, Xaime Rivas Rey*, Cem Sahin*, Sharif Shaher‡, Geoffrey Mainland†

*Department of Electrical and Computer Engineering
†Department of Computer Science
Drexel University, Philadelphia, PA, USA
E-mail: {dandekar,sgb42,mj355,jal77,ir54,
xr39,cs486,mainland}@drexel.edu

‡Echo Ridge, LLC
Sterling, VA, USA
E-mail: sharif.shaher@echoridgenet.com

***Abstract*—Traditional approaches to experimental characterization of wireless communication systems typically involves highly specialized and small-scale experiments to examine narrow aspects of each of these applications. We present the Drexel Grid SDR Testbed, a unified experimental framework to rapidly prototype and evaluate these diverse systems using: (i) field measurements to evaluate real time transceiver and channel-specific effects and (ii) network emulation to evaluate systems at a large scale with controllable and repeatable channels. We present the hardware and software architecture for our testbed, and describe how it is being used for research and education. Specifically, we show experimental network layer metrics in different application domains, and discuss future opportunities using this unique experimental capability.**

***Index Terms*—wireless communications, testbeds, software defined radio**

I. Introduction

The desire for pervasive integration of information-based services into all aspects of modern life has led to tremendous proliferation of wireless communication technology. There has been an explosion of new wireless standards to address the insatiable consumer demand for data while adhering to constraints on the availability of radio spectrum. These standards each address different niche areas discriminated by physical scale (e.g., macro/micro/pico/femto cells, heterogeneous networks, body area networks, cellular networks) and target application (e.g., voice, data, sensor, control). Traditional approaches to experimental characterization of these systems involves highly specialized and small-scale experiments to examine narrow aspects of each of these applications.

To address these needs, we present the Drexel Grid SDR Testbed, a unified experimental framework to rapidly prototype and evaluate these diverse systems using: *i.)* field measurements to evaluate real-time transceiver and channel-specific effects and *ii.)* network emulation to evaluate systems at a large scale with controllable and repeatable propagation channels. The hardware of the Drexel Grid SDR Testbed consists of: *i.)* the Echo Ridge DYnamic Spectrum Environment Emulator (DYSE) 24 port network channel emulator for multi-link emulation as well as hybrid measurement and simulation [1], [2], *ii.)* a centralized server for experiment management hosting virtual machines for each of the nodes in the grid, *iii.)* 20 USRP N210 software defined radios (SDRs) deployed in a ceiling based network for reconfigurable over the air (OTA) testing or network and channel emulation, and *iv.)* a wired network for routing radio frequency (RF), timing, and control signals to nodes for automated experimentation with the testbed.

Complementing this hardware infrastructure is a software infrastructure composed of *i.)* an Application Programming Interface (API) with a custom built experimenter interface for reserving nodes and controlling the wireless channel emulator for both OTA and emulated experiments, and *ii.)* an LXC container based system for deploying SDR software supporting either GNU Radio or a custom full radio protocol stack that we have developed for cross layer prototyping and experimentation. An overview of the testbed is shown in Fig. 1. Some of the features of our testbed include:

- *Flexible, real-time prototyping* - Many SDR platforms in the industrial and academic research community are focused on relatively small variations of existing standards. The Drexel Grid SDR Testbed will build upon our work in developing a real-time, cross-layer, full radio stack software implementation, along with our experience in developing new antenna technologies for wireless communication systems. Ultimately, we envision that the timing and synchronization capabilities of the Drexel Grid SDR Testbed nodes will enable "exotic" physical (PHY) layers (e.g., distributed MIMO, interference alignment) to be tested for the first time in a large-scale network setting.
- *Repeatable and customizable channels* - Many existing wireless testbeds in fixed-grid deployments (e.g., ORBIT [3], SkyNet [4]) or ad hoc deployment across urban (e.g., RoofNet [5], CitySense [6]) or campus areas (e.g., emuLab [7], EWANT [8], MiNT [9], MeshTest [10], GENI wireless [11], POWDER [12], COSMOS [13]) do not have real-time customizable and repeatable RF propagation channels. We are using the EchoRidge DYSE channel emulator [1], [2] to create customizable and repeatable channels using industry-standard and custom channel models while also enabling OTA testing in an indoor office environment with novel antenna technologies that provide broadband and

The work described in this paper was supported by the National Science Foundation through grants CNS-1738070, DGE-1723606, CNS-1730140, CNS-1828236, and CCF-1717088.

directional radiation pattern capabilities. Furthermore, we have developed software to import and apply site-specific electromagnetic ray tracing simulation results to the real-time emulated RF propagation channels between radio nodes in the grid.
- *Scalability* - While testbed-based measurement campaigns often provide greater realism, they usually also suffer from a lack of scalability (i.e., less than a dozen nodes in any single experiment [14]) given the difficulty and expense in developing and maintaining custom hardware. We envision that the Drexel Grid SDR Testbed will addresses this limitation by providing a software API and hardware network emulation that allows physical nodes and virtual nodes to co-exist with one another from the perspective of the overlying network as well as the underlying RF propagation channels.

While there are elements of existing wireless testbeds in the Drexel Grid SDR Testbed, the closest analogous testbed is the Colosseum created by John Hopkins University Applied Physics Laboratory for the DARPA Spectrum Collaboration Challenge [15]. The Colosseum makes use of very powerful baseband processing nodes to provide a platform for diverse implementations of collaborative intelligent radio networks. However, the radios in the Colosseum do not have an OTA testing capability with which to evaluate new antenna technologies. Furthermore, while we leverage some ideas from the Colosseum for the experimenter interface in our testbed, notably in the use of Linux Containers, it is important to note that the Colosseum or its software implementation, are not currently open to the general academic and industrial research community. It is our intent to make our testbed available to the community.

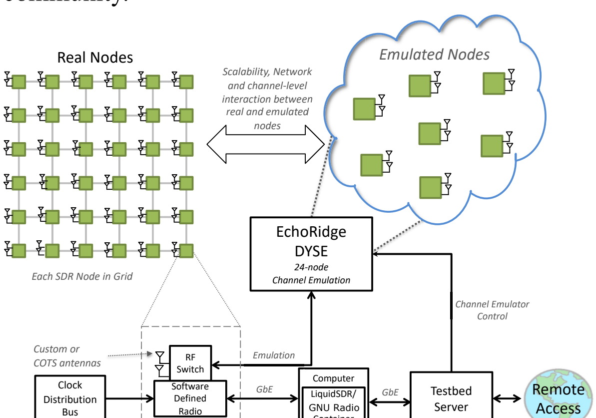

Fig. 1. Drexel Grid SDR Testbed allowing for customizable, repeatable, and scalable evaluation wireless network channels

One of the unique aspects of the testbed is that it can prototype actual transceivers to be deployed in the field with realistic RF propagation and network conditions provided by the emulator, prior to field deployment. This experimental capability and approach is unique, and will hopefully reduce the barriers to the development of future wireless technologies.

This paper is organized as follows: Section II contains an overview of the testbed and presents the hardware and software architecture of the system. Applications of the testbed for undergraduate and graduate education are described in Section III. Preliminary experiments with the testbed illustrating the diversity of applications that can be considered, are presented in Section IV. Conclusions and potential directions for future work are given in Section V.

II. Testbed Architecture

A. Hardware Architecture

1) Physical Layout: Our lab contains a ceiling based “grid” scaffolding network that is being used to mount software defined radios, RFID, and other sensor technologies. The grid is 33 ft by 36 ft with a cross-bar spacing of 3 ft. The grid has power and network connectivity at most of the junctions. The network connectivity is routed to a patch panel in a nearby server room, and can be modified to be local-only for internal measurements or accessible via the Internet for remote measurements and data collection.

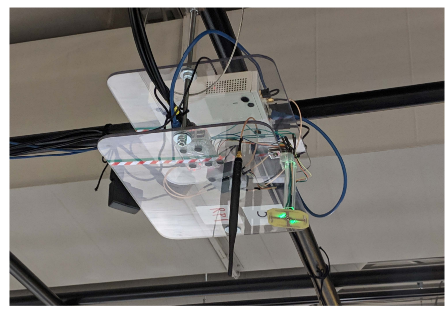

Fig. 2. Drexel Grid SDR Testbed Radio Platform

The radio platforms on the grid scaffolding are shown in Fig. 2. While we currently have Ettus N210 radios on these platforms, we plan to upgrade these nodes to Ettus X310 radios. These radios are on a dedicated Gigabit Ethernet network via the network panel to a rack of computers in our server room. Each radio platform has a custom designed circuit board that serves as a four-way RF switch. This switch allows for the radio signals from the radio to be transmitted through either: *i.)* commercial off-the-shelf antennas OTA, *ii.)* custom-built electrically reconfigurable antennas OTA, *iii.)* an RF cable connection to the DYSE channel emulator, or *iv.)* an RF cable connection to an RF switch matrix for easy connection to ground level equipment. These RF cables were characterized via a vector network analyzer to determine the relative amplitude and phase they induce in an input RF signal. This data can be used to develop calibration algorithms to ensure that signals reaching the channel emulator and associated measurement equipment are of comparable amplitudes and phases. In the future, we envision distributed clock and timing synchronization cables to all of the radios to enable optional RF carrier phase coherent operation. This level of synchronization will be needed to evaluate more intricate PHY layers requiring synchronous transmissions (e.g., distributed beam forming, interference alignment, multi-user MIMO).

2) Wireless Channel Emulation: Full mesh wireless channel emulation capability is provided by a 24x24 full mesh wireless channel emulator. The Echo Ridge DYSE [1], [2] system emulates realistic RF environments in real-time to support testing of a broad variety of wireless devices and systems. A block diagram of this equipment is shown in Fig. 3. RF devices/systems under test (SUT) are interconnected through emulated mobile fading RF propagation channels in the convenience of a controlled laboratory environment. Up to 24 transmit and 24 receive devices can be tested concurrently in any combination. Each of the (576) RF channels between devices can support 100 MHz instantaneous bandwidth (IBW), for a total capacity of over 57 GHz. Systems under test can consist of both physical and virtual (software-modeled) devices, and RF paths/channels can interconnect any combination of physical and/or virtual devices. The physical radios connected to the emulation system have a tuning range between 10 MHz to 6 GHz. The channel emulation system supports both RF and digital (I/Q) input/output connection. Radio paths can be made as complex as required by the overlying testbed, and can include RF propagation phenomena such as delay, Doppler shift, fading, multi-path, shadowing, absorption, and scintillation. Emulated RF environments can thus range from rural to urban core.

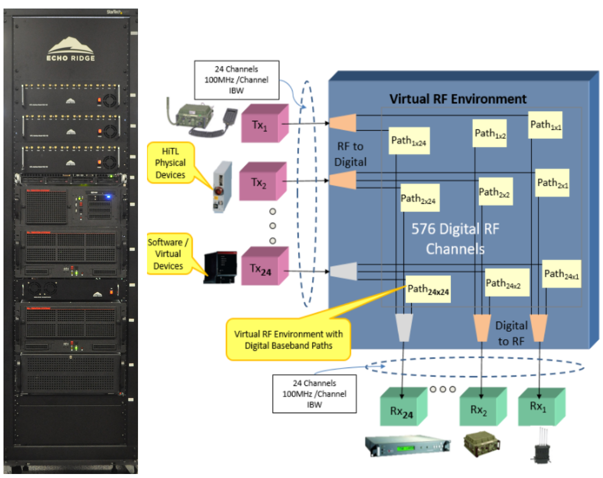

Fig. 3. Echo Ridge DYnamic Spectrum Environment Emulator (DYSE) picture and block diagram

B. Software Architecture

The testbed must provide experimenters low-level access to all hardware while maintaining security and isolation between users. This functionality is accomplished through the use of Linux Containers (LXC) containers to manually manage access to the existing SDR hardware. We are adding automation to this system, so that nodes can be reserved—and LXC containers automatically brought up and torn down—through either a web-based reservation system or a REST web API. As well as allocating testbed hardware, this system automatically manages the Echo Ridge DYSE channel emulator [1] and loads user-specified scenarios that define virtual RF environments. The DYSE also supports virtual emitters that can allow simulated RF nodes to interact with real radios.

1) Experimenter Interface: In order to execute software based radios on this testbed, users of the Drexel Grid SDR Testbed utilize LXC Containers. Containers allow users to decouple software from underlying hardware resources, enabling radio development on and off the testbed. Users can develop SDRs on their local machines and log into the testbed for testing and measurement activities. This simplifies radio development and makes the testbed compatible with other large scale SDR testbeds such as DARPA's SC2 Colosseum [15].

The use of containers also allows developers to use custom software stacks for radio implementations, enabling users to choose from various SDR development tools such as GNU Radio [16] and liquid-dsp [17]. Standards compliant SDR implementations such as srsLTE [18] can also be run on the testbed, providing a flexible platform for cutting edge wireless experimentation. Containers have the advantage of providing software isolation, making node management easier for testbed maintenance and enabling bare-metal performance to radio stacks without the overhead of Virtual Machines.

The current software architecture of this testbed consists of Open-source Linux-based tools. Contemporary back-end deployment tools were used to minimize costs and simplify software engineering and maintenance. Canonical Metal-as-a-Service (MaaS) [19] is being used to deploy operating systems on host computers. Host computers run on Ubuntu (version 16.04 at the time of writing) since this Linux distribution is widely used for SDR applications and supports popular libraries for the same. Ansible [20] is currently being used to configure and manage radio hosts and install required software. Ansible allows for simple management of the large number of nodes on this testbed and helps automate frequently performed maintenance and deployment tasks.

2) Far Field Propagation Models: Large scale propagation models can easily be imported into the testbed to apply channels to any combination of radio nodes according to an experimenter-driven schedule. These channels are computed offline using a variety of experimenter derived models that range from simple free-space path loss models or industry-standard cluster models (e.g., [21]) written in MATLAB to complex multipath scenarios implemented using electromagnetic ray tracing tools (e.g., [22]). The DYSE system has a mechanism that allows the maximum propagation delay per link to be traded for maximum number of multipath taps. A small scale RF fading simulator is currently being developed for deployment on the DYSE.

3) Dragon Radio: Users of the testbed have the ability to use well-established tools, such as GNU Radio, to program their radios. We also provide users with an LXC container holding the source code and compiled binaries for a full-featured software-defined radio of our own design, Dragon Radio. This radio utilizes a TUN/TAP interface to allow for seamless integration with standard Linux network traffic test applications and routing protocols. Using a purely software-based design, this radio is capable of transmission rates in excess of 3 bits/s/Hz with either USRP N210s or USRP X310s. Although we can obtain this data rate on a single 1 MHz channel with modest hardware, the number of channels that can be simultaneously decoded depends on the CPU used. We regularly test a 10-node configuration where each node consists of a Dell PowerEdge R730 with 2x Intel Xeon E5-

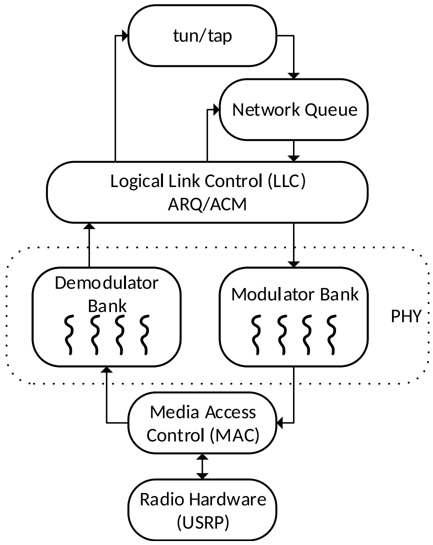

Fig. 4. Dragon Radio Architecture

2650 v4 cores running at 2.2 GHz attached to a USRP X310 via 10 Gbps Ethernet. When operated in a Frequency Division Duplex (FDD) mode with 10 FDMA (Frequency Division Multiple Access) channels in a 10 MHz band, each node transmits in its own 1 MHz channel and simultaneously receives and demodulates the other 9 FDMA channels in parallel - this setup is capable of attaining an aggregate throughput rate of more than 22 Mbps.

The PHY layer of Dragon Radio is based on the open-source liquid-dsp communications signal processing framework, which provides well-tested modem functions for Fourier-based multi-carrier modulations, Orthogonal Frequency Division Multiplexing (OFDM), as well as single-carrier Quadrature Amplitude Modulation (QAM) and Gaussian Minimum Shift Keying (GMSK). It also includes an interface to an open-source Forward Error Correction (FEC) library. We selected liquid-dsp over GNU Radio as the basis of Dragon Radio because we required full control over scheduling, data flow, and transmission timing. The PHY, Media Access Control (MAC), and datalink layers of the radio are written in C++. This functionality is exposed to Python via pybind11 [23], allowing control and spectrum sharing policies to be implemented in a high-level language. The combination of high-performance lower layers with high-level control allows testbed users with diverse research interests and backgrounds to immediately get up-and-running on the testbed.

Fig. 4 shows a high-level view of Dragon Radio's architecture. The radio makes extensive use of both parallelism and concurrency. For example, a bank of demodulator threads acts in parallel to demodulate multiple radio channels simultaneously, and C++ atomics are used to coordinate concurrent radio signal reception and demodulation. In the hybrid FDMA/Time Division Multiple Access (TDMA) MAC we have developed, parallelism enables frequency diversity, and concurrency decreases latency because demodulators do not need to wait for an entire TDMA slot's worth of data to have been received before demodulation can begin. We give a brief description of each element of the diagram and its capabilities here:

- **`tun/tap` Interface**: The radio creates a `tun/tap` interface on startup, which is used as both the ultimate source and destination for packets. Applications can communicate OTA by binding a socket to the interface's (configurable) IP address and using the standard socket interface to send and receive IP packets.
- **Network Queue**: The "network queue" is composed of one or more Click-style [24] network components, the simplest of which are standard FIFO and LIFO queues. We have implemented many other components, such as one that tags packets with QoS indicators. These tags can then be used by downstream network components to, e.g., prioritize particular traffic. The configuration parameters of all components are exposed to Python, which is used to connect the components together. This makes it easier to experiment with new packet processing configurations by avoiding writing any C++.
- **Logical Link Control (LLC)**: The LLC layer implements both Adaptive Coding and Modulation (ACM) and Selective-Repeat Automatic Repeat reQuest (SR-ARQ) control protocols. ACM's selection of the optimal Modulation and Coding Scheme (MCS) to use for a given channel is driven by recent estimates of the packet error rate (PER) for that channel based on the state of the SR-ARQ protocol. Each packet is tagged with an appropriate MCS selected by ACM, which is then used to modulate the packet in the PHY layer.
- **Physical Layer (PHY)**: The physical layer utilizes liquid-dsp's PHY implementations; currently OFDM and two single-carrier modulation schemes are available. A thread pool demodulates incoming signals, allowing multiple channels to be decoded simultaneously. A smaller thread pool modulates just enough packets to satisfy the MAC layer's needs. This ensures that the MAC layer always has enough modulated packets to fill a transmission slot while allowing ARQ to retransmit packets without incurring a large retransmission latency.
- **Hybrid FDMA/TDMA MAC**: Our MAC layer protocol allows scheduling of transmissions across a time/frequency "matrix" whose entries, set from Python, are which radio nodes can transmit in each resource block. We have used this functionality to implement a centralized controller—written in Python—that adjusts the MAC schedule as new nodes appear in the network and as individuals nodes' network loads change. When a node is assigned multiple adjacent TDMA slots, these slots are fused into a single "superslot" without any intervening guard intervals. This allows our MAC to operate as a pure FDMA MAC, a pure TDMA MAC, or anywhere in between.
- **Radio Hardware**: There is a thin shim between the MAC layer and the radio hardware that manages burst transmission and reception. This allows the MAC layer to send and receive data for specific TDMA slots without having to worry about hardware underruns/overruns.

Our architecture description does not include any of the high-level “cognitive” control we have implemented in Python in the process of building Dragon Radio [25], [26]. There are also some low-level, cross-layer components that do not appear in the diagram. Of particular significance is the fine-grained time synchronization implementation we have built that requires coordination between the MAC, PHY, and LLC layers. Time synchronization leverages a modified version of liquid-dsp that precisely tracks packet reception times and MAC layer support for sending packets at precise times. These tools allow the LLC to collect a history of timestamp packets, which Python code then uses to perform a linear regression to set clock offset and skew relative to a master clock node. We can synchronize clocks closely enough across the network that a TDMA slot guard time of $1\,\mathrm{ms}$ is enough to prevent collision.

By providing testbed users with a high-performance, configurable radio that can be controlled purely from Python, we are able to support users spanning from a novice programmer to a C++ signal processing expert. We have already built laboratory exercises around Dragon Radio, and plan to continue investing in developing materials that leverage it as a teaching tool.

III. Applications of the Testbed for Education

Preparing researchers to use a system like the Drexel Grid SDR Testbed is challenging because it may require antennas, communications, and signal processing knowledge typically taught in electrical engineering curricula with application layer, networking layer, and linux-based coding development typically taught to computer engineers and computer scientists. Thus, we have developed educational modules using the Drexel Grid SDR Testbed for undergraduate coursework teaching the fundamentals of analog and digital communication. We have also offered and delivered graduate and undergraduate laboratory coursework in “Wireless Network Security” [27]. In addition, we are building a “Radio Wars” graphical user interface to the testbed to allow students to see a graphical representation of the network while using gamification to stoke student interest. These educational experiences provide a potential platform to recruit researchers to later leverage the testbed for research purposes.

1) Analog and Digital Communication: The Drexel Grid SDR Testbed was used to develop a new laboratory for the undergraduate class, “Introduction to Analog and Digital Communication”, in which students learn communication fundamentals. The course previously used the EMONA TIMS hardware platform for demonstrating topics relating to the course. However, the issue with the EMONA TIMS hardware was that it did not provide a platform for students to pursue more advanced wireless and networking topics. Laboratory materials developed in GNU Radio Companion on our testbed include *i.)* amplitude modulation (DSB-AM), *ii.)* frequency modulation (FM), *iii.)* binary phase shift keying (BPSK), *iv.)* M-ary quadrature amplitude modulation (M-QAM).

2) Wireless Network Security: This developed laboratory course introduces students to security vulnerabilities present in wireless networks. Topics covered begin with an introduction

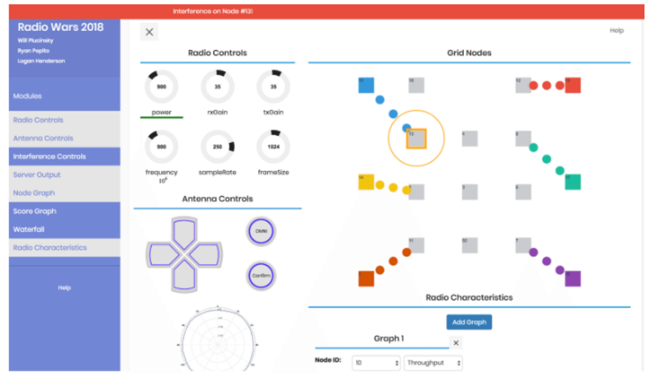

Fig. 5. Radio Wars Visualization System of Drexel Grid SDR Testbed in Interference Game

to wireless networks and the threat models observed due to the broadcast nature of the wireless medium. Current security issues in WiFi networks are then analyzed including link layer vulnerabilities, access point spoofing, eavesdropping, secrecy, confidentiality, and integrity of the data transmitted. WiFi security techniques such as Wired Equivalent Privacy (WEP) and WiFi Protected Access 2 (WPA2), as well as other wireless protocols including GSM and LTE are discussed. Additional topics covered include routing security, trust, user management, key management, and authentication servers. The course also goes into research-oriented wireless network security topics in PHY layer security (encryption and user authentication) and jamming. Lectures are integrated with hands-on laboratory experiments, where students gain experience in implementing security principles discussed during the lectures on the Drexel Grid SDR Testbed. Laboratories developed for the course include topics related to: software defined radio, attack models (eavesdropping, jamming/interference attacks, playback attacks), cross-layer vulnerabilities, wireless network security, encryption schemes, PHY layer security, user authentication, contemporary issues in RFID, IoT, vehicle-to-vehicle, Zigbee, WiFi and LTE security, and others.

3) Radio Wars: We are using the Drexel Grid SDR Testbed to emphasize the interdisciplinary nature of cybersecurity and information assurance. We are developing an academic year long competition available as a undergraduate senior capstone project sequence as well as to master of science students, around a “Radio Wars” competition in which teams leverage the various interdisciplinary degrees of freedom of a SDR (e.g., antennas, modulation, signal processing, networking, application-level software) to securely transmit information while preventing others from doing so. The Radio Wars educational suite is split up into four modules being: 1) Interference and Power Management 2) Encryption 3) Spoofing and Authentication and finally 4) Multi-Team Full Melee. We are developing a back-end database, user-interface, and programming API that allows students actions (e.g., changing transmit power, antenna direction, modulation and coding parameters, MAC design, routing algorithms) on the radios to be collected and logged. We have also developed a visualization system that allows for these actions to be displayed in a way to provide an intuitive display of how these design decisions lead to different network

Fig. 6. Wireless link used for OTA experiment

performance. A screenshot of the visualization system depicting the Drexel Grid SDR Testbed is shown in Fig. 5.

IV. Preliminary Experiments and Results

While the Drexel Grid SDR Testbed is under continuous development, it has already been used to perform experiments representative of its different modes of operation. In this section, we present results using the testbed and our Dragon Radio protocol stack for: *i.)* OTA testing with conventional and custom antennas, *ii.)* testing with emulated RF propagation channels derived from a MATLAB simulation, and *iii.)* testing with emulated RF propagation channels derived from site-specific electromagnetic ray tracing simulations. While presenting each of these case studies, we also highlight how the testbed can be used for a diverse set of application domains.

1) Over the Air Testing: Reconfigurable antennas (RA) are capable of dynamically re-shaping their radiation patterns in response to the needs of the overlying wireless link or network. For this experiment, we evaluated the performance of Dragon Radio in a point-to-point OTA link using two nodes in the Drexel Grid SDR Testbed. We demonstrate how using RAs increases the throughput rate and decreases the PER when compared to conventional omnidirectional antennas. While there are numerous studies on the electromagnetic characteristics of RAs and how they can be used to improve radio link performance, there has been limited work on controlling RAs based on a radio's network level performance metrics. Furthermore, the Drexel Grid SDR Testbed allows us to repeatably test these antennas in complex network topologies and laydowns.

The RA used in this paper [28] has 5 distinct operational modes, or configuration states, and is shown installed underneath the Drexel Grid SDR Testbed radio platform in Fig. 2. It has an omni-directional gain mode and four directional gain modes that are separated by 90° in the azimuth plane, with measured far field radiation patterns shown in [28]. The experiment consisted of two Dragon Radio nodes, one acting as a transmitter and one as a receiver. Regarding the physical placement of the radio nodes on our grid scaffolding system, the receiver was offset 9 ft vertically and 15 ft horizontally on the grid with respect to the transmitter (Tx-Rx distance of 17.5 ft). The receiving node was equipped with a pattern RA and the transmitter was equipped with a commercial omnidirectional folded-dipole antenna. As described in section Section II, Dragon Radio was configured for this experiment to use a TDMA protocol and its throughput performance was tested for 120 seconds for each antenna state at a center frequency of 2 GHz. The Received Signal Strength Indicator (RSSI) and Error Vector Magnitude (EVM) were extracted from the radio's log files and link throughput and PER were measured using iperf2 network test software. The performance results of this experiment is shown in Table I. State 0 of the RA corresponds to an omni-directional gain pattern while states 1-4 represent the directional gain patterns. While the RSSI varied significantly as a function of the RA's state, the radio link maintains a fixed throughput and EVM for all antenna states except state 3. This is explained by the overall high Signal-to-Noise Ratio (SNR) experienced on the link due to its short separation distance (17.5 ft) and the relatively wide beamwidth of the RA's directional gain patterns. In contrast, switching the RA to state 3 significantly degraded link performance because the low-gain back lobe of its radiation pattern points directly at the transmitter.

TABLE I
RA performance under no interference condition

Antenna State	RSSI [dBm]	EVM [dB]	Throughput [kbps]	PER [%]
0	-46.5	-24	1290	0
1	-56.1	-22	1290	0
2	-49.5	-24	1290	0
3	-66	-14	470	9.5
4	-55	-24	1290	0.5

TABLE II
RA performance under interference condition

Antenna State	RSSI [dBm]	EVM [dB]	Throughput [kbps]	PER [%]
0	-46.5	-18	817	0.87
1	-56	-17	699	5.4
2	-49	-19	703	0.33
3	N/A	N/A	0	100
4	-49	-22	1290	0.46

With these results as a performance baseline, the next step was to test if any of the receiving RA's five gain patterns could spatially null the interference generated from a nearby RF continuous wave (CW) jamming node while steering enough antenna gain towards the transmitter. Thus, the previous experiment was repeated after enabling a nearby RF CW jammer with an omnidirectional gain pattern and a mean Equivalent Isotropic Radiated Power (EIRP) that was equivalent to the Dragon Radio transmitter. The RF jamming node was located 15 ft from the receiver in the grid. From the measured data in Table II, it is evident that there is an optimal directional state (state 4) for maximizing throughput in the Dragon Radio link. For this configuration, the Dragon Radio's ACM protocol was able to quickly restore throughput to the best baseline performance. We found for this experimental scenario that the RA's omni-directional antenna mode (state 0), representative of a conventional commercial antennas used in mobile stations, experienced significant throughput degradation due to its inability to spatially null RF jamming interference. We can see from these tables that the use of a RA, in coordination with Dragon Radio's adaptive protocols, creates an interference tolerant cross-layer radio system. A natural extension of this research is to consider adaptive algorithmic approaches to automatically select optimal antenna state in tandem with cross-

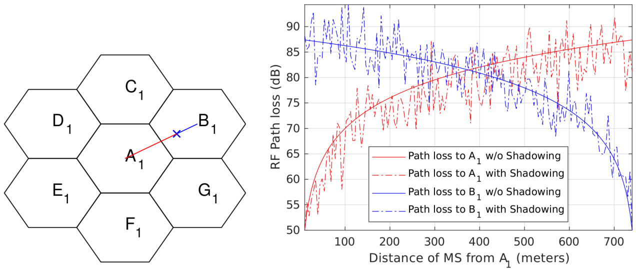

Fig. 7. Cellular scenario with a mobile user moving on a linear trajectory from the coverage area of one cell (A_1) to an adjacent cell (B_1) and RF Path loss from MS to two serving BSs with and without shadowing

layer control knobs in Dragon Radio. These results demonstrate how the Drexel Grid SDR Testbed can be used for OTA test and evaluation of a variety of antenna technologies.

2) Emulated MATLAB channels: A simulation was created in MATLAB to model the canonical scenario of soft handoff (make-before-break) of a Mobile Station (MS) moving between two cellular Base Stations (BSs). As shown in Fig. 7, a MS moves in a straight line between two serving BS (A_1 and B_1) where the red portion of the line indicates progress along the route, the blue portion indicates the remaining distance along the route, and the blue **x** is the location of the MS along the route when the simulation was paused. Fig. 7 shows the simulated RF path loss between the MS and the two BSs as the MS travels from BS A_1 towards B_1 at a RF carrier frequency of $900\,\mathrm{MHz}$ over a time period of $150\,\mathrm{s}$. Superimposed upon the free-space path loss is log-normally distributed RF fading that represents the effect of RF shadowing due to surface clutter. Once generated, all simulated RF path loss values are written into the data format compatible with the DYSE channel emulator. Mobile network emulation experiments can be conducted in the testbed under realistic and repeatable conditions, which is a large benefit of the system due to the difficulty of performing repeated testing under mobility.

For this experiment, three Dragon Radio nodes are configured to use a TDMA MAC protocol and assigned a single $50\,\mathrm{ms}$ slot for transmission (time slots *1, 2,* and *3*, respectively) while DYSE emulates the previously described RF propagation channels. As with all of the Dragon Radio experiments described in this paper, the PHY layer is configured as a 64 subcarrier OFDM waveform with a $1\,\mathrm{MHz}$ bandwidth. The ACM protocol uses a simple capacity maximizing algorithm to select among all of the available MCSs that are comprised of a Reed-Solomon (255,233) FEC and a range of PSK and QAM modulations. The network's throughput performance is measured throughout the experiment using the MGEN network testing software [29]. Furthermore, MGEN tags each data packet with per-flow sequence numbers and timestamps so that end-to-end throughput, latency, loss, and jitter metrics can be calculated from the packet traces written in MGEN's sending and receiving log files in a post-processing step. In this experiment, two $655\,\mathrm{kbps}$ UDP dataflows are sent by the A_1 and B_1 BSs to the MS in time slots *2* and *3*, respectively. As described in the overview of the Dragon Radio software in Section II, the Dragon Radio's Logical Link Controller uses

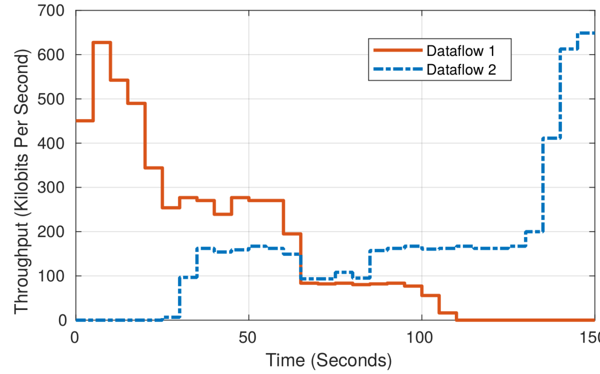

Fig. 8. Throughput for two downlink dataflows collected using Dragon Radio in TDMA mode in a cellular handoff test scenario

a cross-layer ACM protocol that optimizes its MCS selection based on the arrival of ARQ protocol acknowledgement (ACK) and negative-acknowledgement (NACK) messages that are periodically sent from the MS to each BS in time slot *1*.

Fig. 8 shows that the throughput for Dataflow 1 from A_1 to the MS (blue trace) and Dataflow 2 from B_1 to the MS (red trace) are roughly correlated with the path losses from each BS to the MS at each time instant of the experiment. This is an expected result, but there is a noticeable non-symmetry when comparing the two throughput traces where the rate of change for throughput for Dataflow 2 is relatively low. This is because the tested version of the Dragon Radio's ACM protocol is simple and is slow to adapt the MCS as RF channel conditions improve. The root cause of the performance issue with our ACM protocol is described in detail in the next section. One of the benefits of our testbed is that radio nodes can be exposed to repeatable and controllable dynamic RF channel conditions and the limitations of radio protocols can be identified, understood, and addressed.

3) Emulated Ray Traced Channels: Ray tracing software provides an accurate method of modeling the RF propagation characteristics of a unique environment. The tools use transceiver locations, waveform and antenna properties, building shape and material, and other environmental factors with physics driven models to analyze propagation paths [30]. The resulting complex impulse responses and propagation characteristics between each node in the study are used to configure the DYSE channel emulator described in Section II. Ray tracing is particularly useful for scenarios in which obtaining actual measurements may be difficult to acquire, due to frequency licensing, area permits, and cost.

Unmanned Aerial Vehicles (UAVs) have the potential to solve many upcoming challenges in civil, industrial, and military domains due to flight speed capabilities, high maneuverability, and low-cost implementation. Testing of UAV communication systems requires compliance from both the Federal Aviation Administration (FAA) and Federal Communications Commission (FCC), which can be time-consuming and difficult to obtain. Flight-paths in urban or suburban scenarios need street and sidewalk closures as well, requiring additional permits and potential public safety man-hours. In addition, it

Fig. 9. Google Earth Capture and Wireless InSite Model of Study Area with receiver locations and UAV flight-path shown

is inherently difficult to precisely repeat UAV trajectories and channel conditions. The successful characterization of proposed algorithms is contingent upon the ability to reproduce testing environments. Due to the aforementioned challenges, a large amount of analysis is based on simulation, in which practical hardware constraints and imperfections associated with real radios are ignored (e.g. [31]–[33]). The unique capabilities of the Drexel Grid SDR Testbed provide a method of analyzing UAV communication links in a repeatable, controlled, realistic site-specific environment with physical radios.

We developed a model in [34] using Wireless InSite [22], a ray tracing software tool developed by Remcom, Inc., to characterize the outdoor urban environment shown in Fig. 9. The buildings consist of concrete and metal material and have a maximum height of approximately $35\,\mathrm{m}$ to align with the real dimensions of the area. In Fig. 9, our study consists of a rotary-based UAV traveling at a velocity of $1\,\mathrm{m/s}$ from ground locations A to B with an average altitude fluctuating between $20\,\mathrm{m}$ and $25\,\mathrm{m}$. The UAV is providing service to the ground nodes n_1, n_2, and n_3 as positioned in the figure, each with a height of $2\,\mathrm{m}$. The distance along the path of the UAV has a $2\,\mathrm{m}$ resolution, with each point corresponding to unique channel realizations between the UAV and the ground nodes. The uplink signal power of the dominant path received by the UAV for a carrier frequency of $2.0\,\mathrm{GHz}$ is shown for each ground node as a function of time in Fig. 10. The received power accurately depicts the scenario from Fig. 9, with the UAV flight-path starting close to n_1 then traveling away from it with a non-line-of-sight (LOS) condition, n_2 being serviceable for a limited portion of the flight-path due to its alley location, and n_3 experiencing overall favored conditions with LOS. The resulting channel characteristics were provided as an input to the wireless channel emulation system described in Section II.

For this experiment, four Dragon Radio nodes form a hub-and-spoke TDMA radio network where the UAV's radio is assigned to transmit in time slot *1* and the three ground node radios are assigned to transmit in time slots *2*, *3*, and *4*, respectively. MGEN is used to concurrently send a $655\,\mathrm{kbps}$ UDP dataflow from each of the three ground nodes to the UAV while DYSE emulates $400\,\mathrm{s}$ of site-specific RF path loss coefficients shown in Fig. 10. The UAV periodically uses slot *1* to send a combination of ACK and NACK messages to the ground nodes as part of the Dragon Radio's adaptive link SR-ARQ and ACM protocols. The Dragon Radio's PHY and MAC layer configurations are the same as described in Section IV-2.

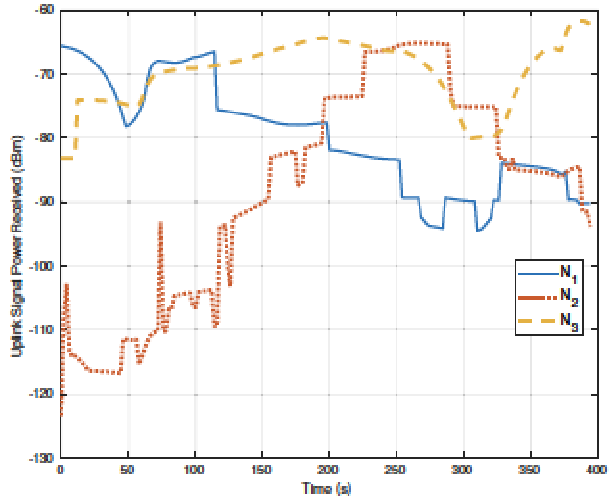

Fig. 10. Uplink power from ground nodes to UAV during flight-path based on ray tracing study

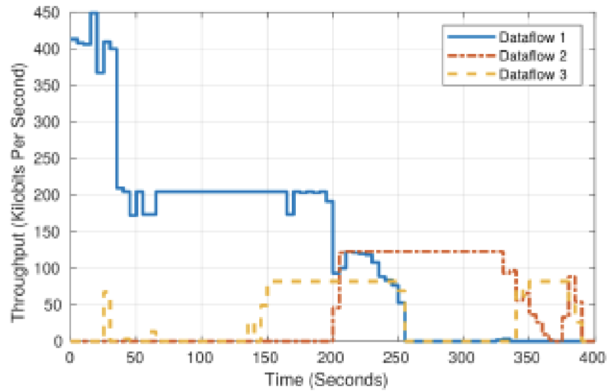

Fig. 11. Throughput for the three uplink dataflows collected using Dragon Radio in the UAV test scenario

Fig. 11 depicts the throughput for the three dataflows over the course of the emulated mobile networking scenario. As expected, the time-varying throughput for each dataflow is roughly correlated with the amount of RF path loss between each ground node and the UAV at that time instant. The tested version of Dragon Radio used a simple ACM algorithm that incrementally reduces the link's MCS until an acceptable PER threshold level is achieved. Once the measured PER is observed below a defined threshold, the ACM algorithm increases the MCS to the next highest rate with a probability based on whether the measured PER was above, or below, the PER threshold for previous attempts to increase the MCS. This algorithm design is adequate for rapidly adapting the MCS for links with overall decreasing channel quality, but introduces significant latency for adapting the MCS for links with increasing channel quality due to its probabilistic exploration of higher MCS based on prior outcomes. This is the reason why there is a relatively high initial throughput for dataflow 1 (decreasing channel quality over time) and low initial throughput for dataflows 2 and 3 (increasing channel quality over time) shown in Fig. 11. We are now improving the ACM algorithm so that it increases the probability of switching to a higher MCS based on knowledge that there is a higher MCS that achieves an acceptable PER for the observed EVM. This design improvement was motivated by conducting repeatable networking experiments with multiple radio nodes and realistic emulated mobile RF channels in the Drexel Grid SDR Testbed.

V. Conclusions and Future Work

In this paper, we have presented the Drexel Grid SDR Testbed, and demonstrated some preliminary results with this testbed. We described the hardware and software architecture for the testbed and how it will enable: *i.)* flexible and real-time prototyping using a custom full radio software defined radio protocol stack that we have developed, *ii.)* repeatable and customizable channels using a 24x24 full mesh wireless channel emulation system, and *iii.)* scalable testing allowing for the co-existence of real and simulated nodes. Our intent is to make this testbed, and our full stack radio, available for experimentation by the academic and industrial research community.

While this paper has demonstrated the potential use of this testbed for OTA experimentation, as well as for emulated cellular and drone based applications, future work will involve considering additional applications pertaining to the Internet of Things (IoT) and cybersecurity. We are currently expanding our testbed to support synchronized clocks to enable distributed array processing and interference alignment algorithms to be tested. We are also expanding the testbed to include OTA capabilities with millimeter wave communication links.

Acknowledgment

The authors would like to acknowledge Logan Henderson, Ryan Pepito, and William Plucinsky for their work on the Radio Wars project, and Oday Bshara, Yuqiao Liu, Md. Abu Saleh Tajin for radio frequency design support of the testbed.

References

[1] Echo Ridge, "DYSE - DYnamic Spectrum Environment Emulator." https://www.echoridgenet.com/products/dyse.

[2] J. Kennedy, J. Carlson, and V. Chakravarthy, "Performance evaluation of spectrum sharing technologies," in *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, pp. 27–33, April 2014.

[3] M. Ott, I. Seskar, R. Siraccusa, and M. Singh, "ORBIT testbed software architecture: supporting experiments as a service," in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*, pp. 136–145, Feb 2005.

[4] T. F. Collins and A. M. Wyglinski, "SkyNet: SDR-based physical simulation testbed," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pp. 1–2, Sep. 2015.

[5] B. A. Chambers, *The Grid Roofnet : a Rooftop Ad Hoc Wireless Network by*. PhD thesis, Massachusetts Institute of Technology, 2002.

[6] R. N. Murty, G. Mainland, I. Rose, A. R. Chowdhury, A. Gosain, J. Bers, and M. Welsh, "CitySense: An Urban-Scale Wireless Sensor Network and Testbed," in *2008 IEEE Conference on Technologies for Homeland Security*, pp. 583–588, IEEE, May 2008.

[7] C. Siaterlis, A. Garcia, and B. Genge, "On the use of emulab testbeds for scientifically rigorous experiments," *Communications Surveys Tutorials, IEEE*, vol. 15, pp. 929–942, Second 2013.

[8] S. Sanghani, T. X. Brown, S. Bhandare, and S. Doshi, "EWANT : The Emulated Wireless Ad Hoc Network Testbed," in *Proc. of IEEE WCNC '03*, vol. 00, pp. 1844–1849, 2003.

[9] P. D. Ashish, R. Srikant, and S. T.-c. Chiueh, "MiNT : A Miniaturized Network Testbed for Mobile Wireless Research," in *Proc. of IEEE INFOCOM*, pp. 2731–2742, 2005.

[10] T. C. Clancy and B. D. Walker, "MeshTest: Laboratory-Based Wireless Testbed for Large Topologies," in *Proc. of TridentCom '07*, pp. 1–6, IEEE, 2007.

[11] A. Gosain and I. Seskar, "GENI wireless testbed: An open edge ecosystem for ubiquitous computing applications," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 54–56, March 2017.

[12] POWDER, "Platform for Open Wireless Data-driven Experimental Research." https://powderwireless.net.

[13] COSMOS, "Cloud Enhanced Open Software Defined Mobile Wireless Testbed for City-Scale Deployment." http://cosmos-lab.org.

[14] K. J. Kwak, Y. Sagduyu, J. Yackoski, B. Azimi-Sadjadi, A. Namazi, J. Deng, and J. Li, "Airborne network evaluation: challenges and high fidelity emulation solution," *IEEE Communications Magazine*, vol. 52, pp. 30–36, October 2014.

[15] DARPA, "DARPA Spectrum Collaboration Challenge Colosseum Overview." https://spectrumcollaborationchallenge.com/wp-content/uploads/SC2-Architecture-Info-Day_p2.pdf, Jan. 2017.

[16] "GNURadio - The Free & Open Software Radio Ecosystem." https://www.gnuradio.org/.

[17] J. D. Gaeddert, "Liquid DSP - Software-Defined Radio Digital Signal Processing Library." http://liquidsdr.org/.

[18] Software Radio Systems Limited, "Open source SDR LTE software suite." http://www.softwareradiosystems.com/products/#srslte.

[19] Canonical, "Metal-As-A-Service (MAAS)." https://maas.io.

[20] Ansible, "Ansible is Simple IT Automation." https://www.ansible.com/.

[21] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz (Release 14)," Technical Specification (TS) 38.901, 3rd Generation Partnership Project (3GPP), 07 2017. Version 14.1.1.

[22] REMCOM, "Wireless InSite: 3D Wireless Prediction Software." https://www.remcom.com/wireless-insite-em-propagation-software/.

[23] W. Jakob, J. Rhinelander, and D. Moldovan, "pybind11 – seamless operability between C++11 and Python," 2017. https://github.com/pybind/pybind11.

[24] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click modular router," *ACM Transactions on Computer Systems*, vol. 18, pp. 263–297, Aug. 2000.

[25] A. Lackpour et al., "Design and implementation of the Secondary User-Enhanced Spectrum Sharing (SUESS) radio," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–2, March 2017.

[26] A. Lackpour et al., "Enhanced 5G spectrum sharing using a new adaptive NC-OFDM waveform with reconfigurable antennas," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–2, March 2017.

[27] C. Sahin, D. Nguyen, J. Chacko, and K. R. Dandekar, "Wireless cybersecurity education via a software defined radio laboratory," in *2015 IEEE Frontiers in Education Conference (FIE)*, pp. 1–8, Oct 2015.

[28] D. Patron, K. R. Dandekar, and D. Piazza, "A wideband planar antenna with reconfigurable omnidirectional and directional radiation patterns," *Electronics Letters*, 2013.

[29] Naval Research Lab, "Networks and Communication Systems Branch Multi-Generator (MGEN)." https://www.nrl.navy.mil/itd/ncs/products/mgen, 2019.

[30] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015.

[31] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," *IEEE Communications Letters*, vol. 20, pp. 1647–1650, Aug 2016.

[32] M. M. Azari, F. Rosas, K. Chen, and S. Pollin, "Ultra reliable UAV communication using altitude and cooperation diversity," *IEEE Transactions on Communications*, vol. 66, pp. 330–344, Jan 2018.

[33] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Transactions on Communications*, vol. 64, pp. 4983–4996, Dec 2016.

[34] M. Jacovic, O. Bshara, and K. R. Dandekar, "Waveform Design of UAV Data Links in Urban Environments for Interference Mitigation," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, Aug 2018.