Virtualization for Wireless Testbeds

Nicholas Kaminski*, Johann M. Marquez-Barja* and Luiz A DaSilva*†
*CTVR / the telecommunications research centre
Trinity College Dublin, Ireland.
{kaminskn,marquejm,dasilval}@tcd.ie
†Virginia Tech, USA
ldasilva@vt.edu

I. Introduction

Supporting the rapid evolution of wireless communications technologies presents a challenge to any wireless testbed facility. In order to maintain high paced innovation, Trinity College Dublin (TCD), a member of the CREW cognitive network testbed federation, has deployed a wireless testbed-as-a-service platform. This deployment aims at reducing the overhead associated with implementation-based experiments by eliminating the need for users to provision their own equipment, configure connections, and deploy computational environments. Moreover, our facility does not reduce user control of their experimental system by forcing them to conform to a pre-determined set of libraries or to fit their work within restrictive confines to avoid inferring with the configurations of other users. Rather, we have applied the methods of cloud-computing to provide users with fully customizable computational environments automatically connected to dedicated radio hardware. These environments initially provide the software required to realize a wide range of radio systems, and users are free to add other software as needed. By offering an individualized experience to every user, we provide a facility that removes the impediments to implementation-based research without slowing the pace of advanced wireless research.

II. Testbed Description

TCD runs a wireless testbed for exploration of Software Defined Radio (SDR) systems, focusing on providing users the ability to customize the platform to their needs. In pursuit of providing this custom testbed experience, TCD has implemented a testbed-as-a-service platform based on offering the user individual experimentation units, supported by virtual computational platforms. Experimentation units are a logical organization of resources consisting of a computational element, a hardware radio front-end component, and the Iris software defined radio package. Each experimentation unit represents the minimum set of resources required by a user to construct a radio element in TCD's facility. Due to the capabilities of Iris, these experimental units effectively represent the potential to realize any arbitrary radio system that a user may require. Packaging resources into experimental units quantizes TCD's testbed facility in a manner that allows users to deploy a customized radio system as needed.

User customization is underpinned through the application of a cloud-computing based management system. This paradigm combines virtualized computational resources, loaded with a highly configurable (and reconfigurable) SDR package, and widely flexible radio hardware to support experimentation based research. Virtualization of computational elements provides each user complete control over an isolated computational environment. Since each user controls their own virtualized environment, users are free to completely individualize their computational platform, including configuration of global libraries or loading of additional software. Users may save their configurations, including all details of the computational system, for later use, at any point. The management system streamlines coordination of virtual computers, handling the deployment of these environments and their connection to radio hardware at the request of the user. The result of cloud-computing based management is that each user gets the experience of a personal testbed, yet the resources are shared among many separate users and projects.

Figure 1 displays the architecture of TCD's virtualized testbed. Each experimentation unit initially provides users with an virtual computer running Ubuntu 14.04 and loaded with Iris. An array of servers, referred to as host servers, provide the computational power to run these experimentation units. Each virtual machine is connected to a Universal Software Radio Peripheral (USRP) mounted on our ceiling grid within our dedicated testing space. These connections are made through dedicated network ports; i.e., each experimentation unit is provided with a full network interface controller (NIC) to connect to a USRP. The frontend server coordinates and controls virtual machines, handling the deployment of computational environments and their connection to USRPs. The frontend server also provides users access to experimentation units, currently through SSH. Centralized control also allows the frontend to handle the scheduling of user access.

Figure 2 displays the support structure for the virtualization management system. A gateway machine provides a central point for remote testbed access, data access, and documentation. This gateway machine also supports an HTML based user interface system. The server dedicated to hosting documentation provides users with a wealth of information on how to use the testbed. The data server holds user data separately from virtual machines, providing a more permanent storage location and allowing the collection of information from several virtual machines. This support system allows users to take full advantage of the cloud base management structure.

TCD uses this facility to support internal research on a range of topics. Through involvement in the CREW European project, this facility is currently being used to study small-cell deployment in radar bands. Specifically, we have implemented cognitive uses with the capability to identify features of radar signal and a model radar system on the testbed. This system will serve as the foundation for integration with libLTE, TCD's software LTE package. Through involvement in the FORGE European project, this facility is being leverage for educational purposes. Here, the facility provides students with the opportunity to experience hands-on experimentation. Currently, this facility is being integrated into the Fed4Fire federation to extend this functionality to external users through standard FIRE connection.

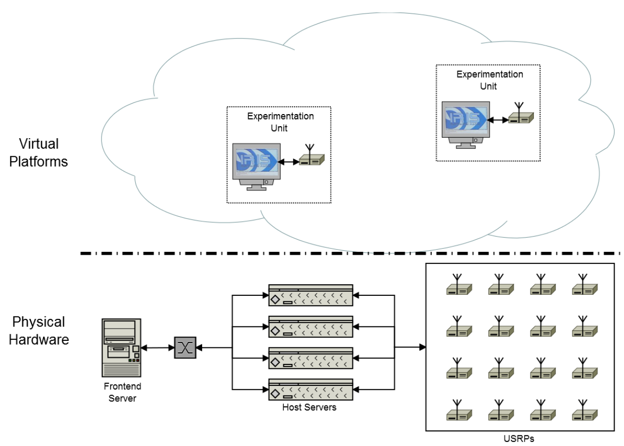

Fig. 1. Testbed Architecture

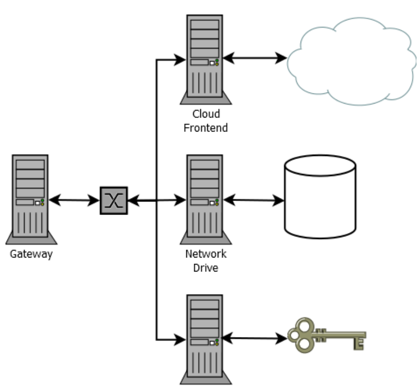

Fig. 2. Support Architecture

III. Demo Overview

Since the fundamental purpose of a testbed is to conduct wireless experimentation, testbed facilities must avoid undue amounts of overhead to preserve useful operation. Additional services undermine the core purpose of a testbed if they load the resources in a manner that impairs user experience. Thus in the construction of the testbed discussed above, we took care to design a system that provides users with an on-demand experience without large of amounts of overhead.

We provide a demonstration of this streamlined testbed-as-a-service to show its utility and lack of crippling overhead. We will display a side-by-side comparison of the cloud-computing based testbed-as-a-service described above and a more traditional statically implemented testbed realization, accessing both remotely. The more traditional testbed will consist of a natively installed computational platform, statically configured to run the desired experiment and accessed through a standard SSH connection with X-forwarding. Each of these platforms will be used to perform a multicarrier-modulation experiment. This experiment will illustrate the reconfigurability of the Iris SDR package on each platform to demonstrate the suitability of our approach to constructing a testbed-as-a-service facility.

The experiment displayed will consist of sending packets from transmitter units to receiver units using an orthogonal frequency-division multiplexing (OFDM) signal. This signal is generated in Iris and transmitted/received through the USRPs. During operation, users may change the centre frequency, bandwidth, and gain dynamically, observing the impact of these parameters on the signal in real-time. The user will be provided with several graphical representations of the performance of the radio system. A waterfall plot will display the time-frequency representation of channel use, as exemplified in Figure 3.

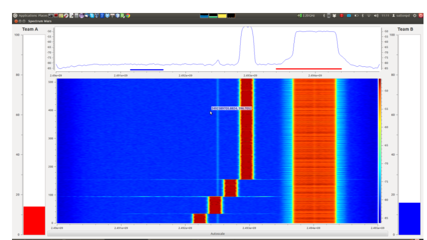

Fig. 3. Waterfall Plot Display

Providing this side-by-side comparison will demonstrate the reduced overhead in the testbed-as-a-service platform in a visceral manner. Thus, the demonstration serves as initial benchmarking results in showing the suitability of our approach to providing a testbed. In this way, we will show the feasibility of offering customizable experimental units.

Acronyms

NIC network interface controller
OFDM orthogonal frequency-division multiplexing
SDR Software Defined Radio
TCD Trinity College Dublin
USRP Universal Software Radio Peripheral