10

Experimental Systems for Next-Generation Wireless Networking

Sachin Ganu, Max Ott, and Ivan Seskar

10.1 Introduction

With the evolution of wireless technologies that continue to offer higher data rates using both licensed and unlicensed spectrum, the number of portable, handheld computing devices using wireless connectivity to the Internet has increased dramatically. Another major category for growth in wireless devices is that of embedded wireless devices or sensors that help monitor and control objects and events in the physical world via the Internet. Vehicular networking is an emerging application for wireless networking with a focus on increased road safety.

The broad architectural challenge facing the wireless and network research communities is that of evolving the Internet architecture to efficiently incorporate emerging wired and wireless network elements such as mobile terminals, ad hoc routers, and embedded sensors and to provide end-to-end service abstractions that facilitate application development. A top-down approach to the problem starts by identifying canonical wireless scenarios that cover a broad range of environments such as cellular data services, WiFi hot spots, mobile peer-to-peer (P2P), ad hoc mesh networks for broadband access, vehicular networks, sensor networks, and pervasive systems. These wireless application scenarios lead to a rich diversity of networking requirements for the future Internet that need to be analyzed and validated experimentally. One of the key challenges faced in characterization and evaluation of these complex wireless scenarios is the lack of generally available tools for modeling, emulation, or rapid prototyping of a complete wireless network. It has been observed that much of this work relies on a formal separation between the radio and networking layers. As a result, most of the contemporary research in wireless networks is primarily based on simulations or in-house small-scale emulators.

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

Pure simulation-based approaches rely on discrete event network simulation providing support for simulation of transport protocols, network layer protocols, and multicasting, as well as routing protocols over both wired and wireless links. The link layer is characterized based on various parameters such as loss, latency, and bit rate. The simulation tools allow the highest flexibility and programmability by abstracting these various physical layer effects, thus facilitating quick development and evaluation of novel layer-2 and networking protocols. Simulation-based techniques provide a cost-effective and repeatable method for large-scale system evaluation. Because all the processing is software-based, even large-scale topologies can be simulated in reasonable time using parallel computation, if necessary. The main drawback of simulation-based evaluation is simplistic modeling of certain key parameters that impact the accuracy of the evaluation. For example, as described in Kotz et al.[1] and Pawlikowski et al.,[2] most of the simulation models assume that the radio's transmission area is perfectly circular or that the wireless links between two communicating entities are symmetric in terms of link losses. Also, some simulation models incorrectly assume fixed-link bandwidths over fixed distances. In addition, some of the traffic models may be too simplistic to capture the characteristics of actual usage patterns, thereby resulting in simulation results not reflecting real-world constraints and conditions.

One of the common network simulators is *ns*,[3] which is popular because of variety of contributed models based on its open-source distribution. Wireless extensions to this simulator were developed[4] to enable simulation of mobile nodes connected by wireless links, including the ability to simulate multihop wireless ad hoc networks. To improve the simulation efficiency of discrete event simulation, GlomoSim[5] offers scalable simulation environment for wireless and wired network systems based on parallelizing execution of discrete-event simulations on different computing infrastructure. Commercially available network simulators such as Opnet Modeler[6] and Qualnet[7] are software tools for network modeling and simulation, and have been extended to support application performance management and network planning tools.

Emulation techniques combine software-based simulation and hardware in order to introduce realistic physical link layer under more controllable environments. This approach introduces some degree of realism by using devices for actual communication but still simulating some network behavior in software. This approach can be used to test protocols under certain realistic network conditions. For example, network topologies can be emulated by setting up packet filters in software to selectively drop packets or via artificial attenuation of the transmit power levels. Although this approach is more realistic than simulations, due to the use of actual hardware, it is limited to small-scale, controlled experiments that still lack the full degree of realism in terms of actual user mobility patterns and real physical layer link conditions such as attenuation, multipath, and fading effects.

Small-scale experimental setups such as the Ad-Hoc Protocol Evaluation Testbed (APE)[8] have attempted to address the deficiencies of the simulation-based approaches. The APE testbed provides software that can be installed on laptops with wireless NICs, as well as script-based mobility patterns that can be used to conduct repeatable mobility ad hoc wireless networking experiments. Software tools have been provided to create arbitrary topologies using MAC layer filtering of packets, allowing multihop links to be created in a controlled manner. Similarly, MobiEmu[9] emulates mobile ad hoc network environment with a fixed network of machines in a lab environment and supports mobility scenarios without actually moving nodes physically by dynamically installing or removing packet filters, thereby emulating network dynamics. Kiess and Mauve[10] provide a good summary of various testbeds.

Real-world experimental setups provide a completely realistic environment with complex interactions between different protocol components and enable evaluation of protocols in practical environments that consider software and hardware limitations. Examples include the Roofnet[11] network at MIT, which is an outdoor deployment of 802.11b-based wireless nodes on rooftops to create multihop wireless links to characterize wireless link behavior useful for study of novel multihop routing protocols. However, these setups are still limited in size and lack the ability to perform controlled repeatable experimentation.

Even though there are drawbacks to each approach, they are complementary to each other, and a general evaluation methodology can encompass all three approaches. To support this, there is a need for a framework that allows portability of protocol implementations from simulation, to emulation, to real-world testbeds. This way, some of the earlier studies can be done in more controlled simulation environments and the final protocol can be evaluated on emulators and testbeds. Results from actual experiments can help further improve the protocol design as well as simulation models, and is critical in supplementing simulation results with real-world experimental data that captures the complexities of various components, cross-layer interactions, and dependencies between various layers of the protocol.

This chapter describes the requirements, design considerations, and challenges for large-scale, open-access networking research testbeds. Key components including programmability at different layers of the networking stack and control-and-management software to enable multiple simultaneous experimenters to access the resources using virtualization are also discussed. Several existing testbeds including the ORBIT testbed at Rutgers University, CitySense project at University of Massachusetts, Amherst, Kansei sensor network testbed at Ohio State University, and others are described in terms of capability and deployment scale, as well as experiments that they support. This leads into a discussion of ongoing efforts to federate several such different substrates and existing wired experimental infrastructure to create a globally distributed experimental network resource as a part of NSF-sponsored Global Environment for Network Innovations (GENI).

10.2 Future Wireless Networking Testbeds: Requirements and Challenges

10.2.1 Design Requirements

The experimental infrastructure described earlier should be designed to incorporate a wide range of wireless networking capabilities in order to provide experimenters with access to emerging radio technologies that are becoming increasingly important at the Internet edge. Such an infrastructure should support various types of wireless technologies including 802.11 (WiFi), emerging WiMAX for metro area wireless access, sensor networking, and the more recent software radio platforms providing programmable means to enable a new generation of "adaptive wireless networks."

To conduct insightful experiments on these different kinds of networks, it is extremely important to provide the ability to program various radio nodes and network elements at all levels to run protocols and software supplied by the user. The user should also be able to collect real-time measurements at different granularities, dynamically change parameters at run-time, and configure the network topology in a flexible manner.

In addition, wireless communication is highly influenced by the environment in which it takes place. The same hardware configuration inside an office building, on a factory floor, or outdoors may result in very different behavior. This leads to a potential requirement for multiple similar testbeds in various representative environments.

Another requirement of the system is efficient resource utilization that can be achieved by simultaneously supporting more than one "virtual network" or protocol implementations. One approach to accomplish this has been demonstrated in the PlanetLab testbed[12,13] and involves the concept of partitioning of node and link resources ("slices") in a nonoverlapping manner across multiple virtual networks. Similar concepts can be applied to wireless networks taking care to account for shared-media interactions between wireless links and nodes. The degree of virtualization that can be practically achieved depends on the hardware platforms used and the specific wireless networking scenario under consideration.

A final requirement of the experimental infrastructure is supporting repeatability, which is defined as the ability of the infrastructure to help predict within

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

certain error boundaries the performance of the system under test under certain parameters.

10.2.2 Design Challenges

Supporting the previously described requirements in a multiuser wireless experimental facility presents some interesting challenges. These include routine ones related to management and user interface for experimentation (user account maintenance, access control, user portal for experimenters) as well as complex ones related to optimizing the usage of the testbed by accommodating as many users as possible in a given time duration.

In wired experimentation, resources for individual users can be segregated either at the MAC layer using VLANs, or IP layer using firewalls, or a combination of both. Wireless experimentation poses an interesting challenge due to the inherent broadcast nature of the medium, thereby affecting the other nodes in the vicinity. This results in complex interference and interactions between different "independent" allocated resources, making it far more difficult to set up a reproducible wireless networking experiment. Other factors unique to wireless environments include:

- Radio channel properties depend on specific wireless node locations and surroundings.
- Physical layer bit-rates and error-rates are time-varying.
- Shared medium layer-2 protocols on the radio link have a strong impact on network performance.
- There are complex interactions between different layers of the wireless protocol stack and currently their mutual interaction cannot be studied easily.
- User's exhibit random mobility and their location also plays a role.

In addition, visibility and programmability of different radio layer parameters are challenging due to limited software ability to change existing protocol behavior.

10.2.3 Key Components for Flexible Wireless Experimentation

We describe several key components that address the requirements and meet the design challenges outlined earlier to provide multiuser open-access wireless networking infrastructure.

10.2.3.1 Open WiFi/WiMAX Hardware and Software

Most of the commercially available wireless devices provide limited support to change the behavior of the radio as well as the link layer. However, it is important to provide programmable access at various layers of the network stack. With

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

the advent of WiMAX as a wide-area wireless alternative to cellular networks, it is important to have support for outdoor deployments with dual mode devices that can switch from indoor WiFi to outdoor WiMAX mode based on their location.

10.2.3.2 Virtualization of Wireless MAC

Virtualization is about slicing a resource into smaller portions with minimal interactions or interference between them. This is one way to efficiently utilize the experimental system resources. Testbeds, such as PlanetLab, virtualize their existing resources to simultaneously support multiple applications or simulated networks. Unlike traditional wired networks, it is difficult to virtualize wireless environments because the performance and characteristics of wireless networks are greatly influenced by their MAC and PHY layers. Some level of virtualization can be achieved by allocating different frequency spectrum resources to different users. Virtualization of the medium access layer is even more challenging. Recent research[14] indicates that it is feasible to build inexpensive, virtualized wireless interface that can also be used for efficient overlay networks for experiments at the MAC layer using available software and off-the-shelf wireless network interface cards. SoftMAC[15] is a software library that provides software control over a specific wireless network hardware interface. The basic SoftMAC layer can be extended to create MultiMAC.[15] Packets are received using a common PHY layer implemented by the underlying radio and are then presented to the different MAC layers. The resulting MultiMAC implementation can be used to build both virtualized and overlay wireless networks at the MAC layer, and to do so using commodity hardware, although on a limited set of computing platforms.

10.2.3.3 Cognitive Radio Platforms

Software radios are quickly emerging as the enabling platform for future wireless communications systems, from commercial,[16] to open source,[17] to military.[18] In a software radio, many signal-processing functions such as modulation, coding, and spreading are performed in software. This enables agile radios – a software radio that uses its flexibility to dynamically change waveform characteristics and behavior in response to instruction. A cognitive radio extends these concepts to a radio that senses and reacts to its operating environment. In addition to the ability to virtualize the medium access layer and providing software control to tweak the different parameters in commercially available WiFi platforms, programmability of physical layer will be very important in future wireless networks. Cognitive radio platforms for research are currently being developed

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

at several organizations including University of Utah,[17] University of Kansas,[19] Rutgers University,[20] and Rice University.[21]

10.2.3.4 Wireless Network Monitoring and Measurement

A common requirement for many routing algorithms, congestion avoidance, admission control, and other protocols or applications is the need to estimate certain performance metrics such as bottleneck bandwidth, available bandwidth, and bulk transfer capacity over wireless links. Many of these require measurements, but measurements are also instrumental in evaluating and benchmarking system performance. Thus, wireless measurement plays an important role in the design, development, and operation of wireless networking protocols and architecture.

For this reason, measurements should be viewed as an important core systems service. Often, obtaining the raw measurements is hardware and system dependent, requiring specific additional processing as well as initial configuration. A service approach will also avoid duplication when multiple components require the same measurements. From an experimentation point of view, it needs to be designed so that measurement collection and its delivery causes minimal disruption to the actual system being evaluated. One approach to address this is separating the experimental plane and control/management plane or temporarily storing the measurement data locally and collating at the conclusion of the experiment.

10.2.3.5 Mobility Support

Mobility of wireless devices, and the dynamic component it adds, is one of the most important aspects in many experiments. Mobility in the context of testbeds falls roughly into two categories: making devices move and measuring their movements. Accordingly, there are two types of testbeds supporting mobility: controlled and observed. In controlled mobility environments, the device is usually mounted on a mobile platform, such as a robot that can be directly controlled by the experimenter. In other environments, where devices are, for instance, carried by buses[34] or people, mobility can either be directly observed by measuring a device's location, or the statistical properties of the underlying mobility patterns are known, or the environment is categorized as being representative of a specific environment.

10.2.3.6 Wireless Network User Models and Measurements Repository

Many research publications report experimental evaluations of various wireless protocols under specific user mobility models and traffic modeling parameters. It is important to efficiently collect and preserve large amounts of user and systems

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

Table 10.1. *Summary of Sample Wireless Experimental Testbeds*

Testbed	Classification	Scale	Features
ORBIT	Indoor wireless grid	400 nodes with dual radio 802.11 a/b/g radios, Zigbee radios and GNU radio platforms (USRP and USRP2)	High-density indoor wireless experimentation Multihop topology emulation using noise injection Limited mobility emulation using software Dedicated wired network for experiment control, resource management, and data collection.
Kansei	Indoor sensor grid	210 sensor nodes with 802.11b radios and sensing radio	Support for sensor data generation, real-time event creation, data storage Limited mobility support using robot-mounted nodes
CitySense	Outdoor distributed sensor	100 pole-mounted nodes with modular sensing radio and 802.11b/g radio	802.11b/g radio and OLSR routing protocol for remote monitoring and management
DieselNet	Outdoor mobile networking	40 buses with wireless nodes	Supports wireless access on one radio and opportunistic peer to-peer-or dedicated relay-based communication on other radios
Emulab	Indoor office environment	40 nodes with 802.11 a/b/g and GNU Radio platform (USRP and USRP2) scattered around the two floors of a large building	Multiplexed virtual node implementation allowing emulation of larger networks
VT-CORNET	Indoor office environment	48 USRP2-based nodes with custom RF daughterboard	Use of Software Communications Architecture (SCA) framework for waveform generation.

observations in a consistent manner. This will enable the development of new inference and data-mining algorithms to extract new insights from previously conducted experiments.

The Crawdad project[22] at Dartmouth University is an example of an archive with the capacity to store wireless trace data from many contributing locations and tools for collecting, data anonymity, and analysis. Sample data sets include association and mobility trails of users at conferences. It is also important to take into account the privacy concerns for long-term experimentation involving real users.

10.3 Existing Wireless Testbeds

In this section, several experimental testbed facilities that have been built as part of the NSF-sponsored initiative for open wireless multiuser experimental facility (MXF) testbeds are described. Specifically, we include testbeds purpose-built for indoor wireless ad hoc and mesh networking research (ORBIT), sensor networking (Kansei), citywide outdoor sensor deployment for environment monitoring (CitySense), vehicular networking testbed (DieselNet), network emulation testbed (EMULAB), and cognitive radio testbed (VT-CORNET). Table 10.1 broadly summarizes the capabilities, scales of deployments, and other features of each testbed.

10.3.1 ORBIT: Indoor Wireless Testbed for Ad hoc and Mesh Networking

10.3.1.1 ORBIT System

The ORBIT large-scale radio grid emulator[23] consists of an array of 20×20 programmable nodes, each with multiple 802.11a,b,g,n or additionally other radios cards such as Bluetooth, Zigbee, USRP, and USRP2. The radio nodes are connected to an array of back-end servers over a switched Ethernet network, with separate physical interfaces for data, management, and control. Interference sources and spectrum monitoring equipment are also integrated into the radio grid, as shown in Figure 10.1. Users of the grid use an *Experiment Controller* to conduct experiments involving network topologies and protocol software specified using an ns-2 like scripting language. Experiments are described in a domain-specific language, called OEDL.[28] A *radio mapping algorithm*[27] that uses controllable noise sources spaced across the grid to emulate the effect of physical distance is used to map real-world wireless network scenarios to specific nodes in the grid.

10.3.1.2 Hardware Components

The ORBIT nodes in this setup are suspended from the ceiling with grid spacing of 1 m, as shown in Figure 10.2; each node is equipped with two 802.11a/b/g radio cards and optionally with Bluetooth, Zigbee, and GNU radios. The custom-built radio node is a programmable platform including a 1 GHz VIA C3 processor with 512MB of RAM and 20 GB of hard disk memory. This configuration is necessary to support heavy-duty protocol processing, computation, and storage

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

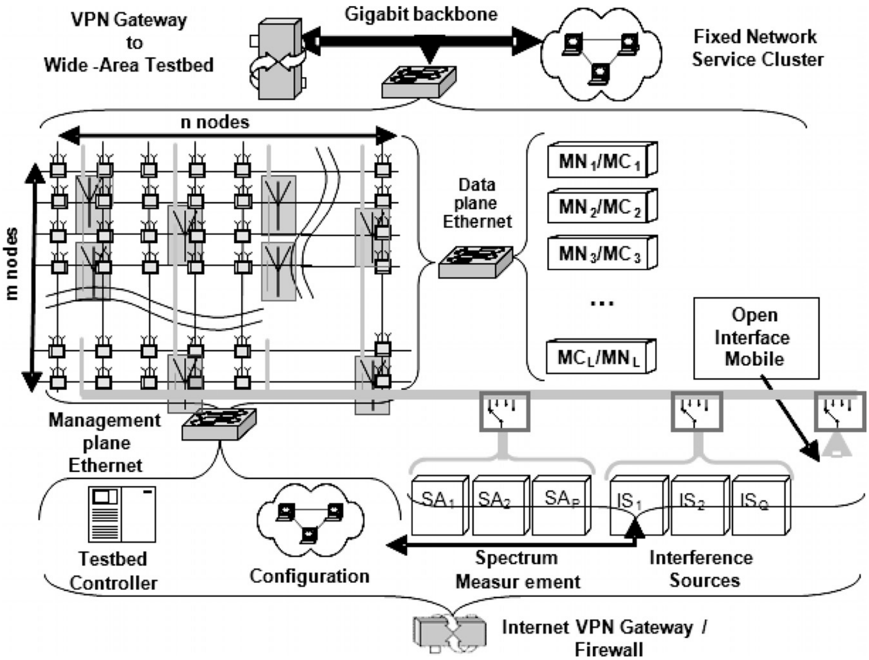


Figure 10.1. ORBIT radio grid architecture.

at high throughputs. Each platform also includes two wired 1,000 Base-T Ethernet interfaces for experimental data and control. In addition, each node contains a custom-designed chassis manager module with Ethernet connectivity to allow for remote monitoring and rebooting of nodes.

RF instrumentation: The ORBIT grid includes equipment for measurement of radio signal levels and to create various types of artificial RF interference. This can be used to create different topologies in the

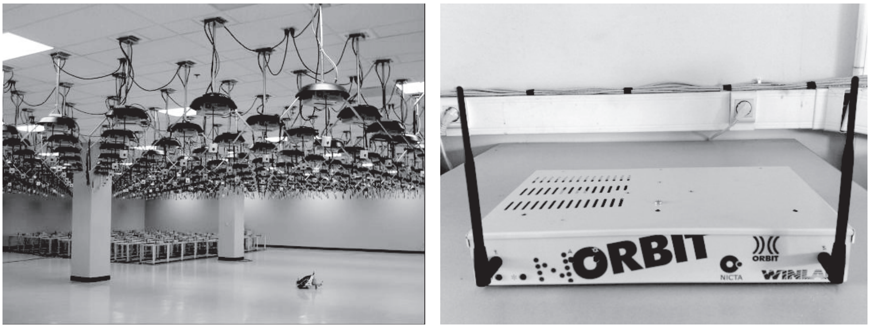


Figure 10.2. ORBIT radio grid, ORBIT nodes.

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

static grid (e.g., multihop mesh) with the same physical positioning of the nodes.

WLAN Monitoring: An independent WLAN monitoring system provides a MAC/network layer view of the radio grid's components using a number of WLAN "observers" spread across the system.

Support Servers: The testbed's back-end equipment includes several front-end servers for Web services, experiment support, and data storage. The database servers support multiterabyte storage capacity. There is also an Ethernet switching array with approximately 1,400 ports necessary to switch traffic from 3 × 400 grid node interfaces and the servers.

ORBIT Sandboxes: The ORBIT system includes several "sandboxes," each with two radio nodes connected by RF cable, to permit users to debug their experimental code before using the large grid for experiments at scale. This helps improve testbed utilization by off-loading most-early-stage experiments and software development to sandboxes rather than the main radio grid.

10.3.1.3 Software Architecture

The ORBIT system uses a time-slot-based reservation mechanism to schedule experiments (Figure 10.3). Users log into a portal to reserve future time slots

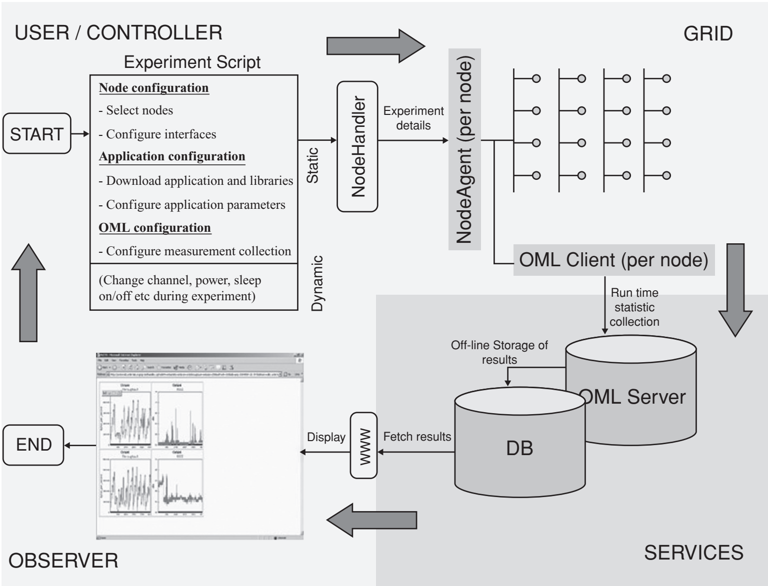


Figure 10.3. Experiment lifecycle on ORBIT testbed.

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

for experimentation and are granted access to their slots only at the allotted time. Users log into the system during their allocated slot and can launch the experiment via scripts. The entire experiment is usually captured in a script that is descriptive: It defines the wireless nodes, their roles, various network parameters such as traffic pattern, protocol, and packet sizes, as well as wireless settings such as channel, power levels, and the like. ORBIT uses open-source drivers MadWiFi and ath5k for Atheros and IPW2200 for Intel 802.11a/b/g interfaces. Additionally, users can specify the statistics (at different layers of the stack) to be collected at configurable intervals (time- or sample-based). The script is interpreted by the experiment controller software that powers up the wireless nodes, installs custom images if necessary, configures network parameters on these nodes, and starts the experiment work flow. It also initialized databases necessary for collecting the results of the experiment based on user-selected statistics and granularity of collection. The ORBIT Measurement Framework Library (OML)[26] is responsible for data collection in a nonintrusive manner over dedicated Ethernet interface and network. In addition to time- and sample-based filters, the OML framework allows run-time filters to be applied to either of these measurement techniques to report minimum, maximum, average, or sum of time-based or sample-based measurements.[29] The readers are referred to Raychaudhuri et al.,[23] Ganu et al.,[24] Ott et al.,[25] Singh et al.,[26] Lei et al.,[27] Rakotoarivelo et al.,[28] and White et al.[29] for further details on the ORBIT testbed and Management Framework (OMF).

10.3.2 Kansei Sensor Networking Testbed

The Kansei testbed[30] is hosted at the Ohio State University and supports a two-dimensional grid of experimental sensor hardware with support for sensor data generation, real-time event creation, data storage, and the related experiment management software.

10.3.2.1 Hardware Infrastructure

The Kansei testbed consists of three types of sensor arrays: stationary, portable, and mobile. The stationary array (as shown in Figure 10.4) has 210 sensor nodes placed in a 15×14 meter rectangular grid with 1 meter spacing. Each node consists of two hardware platforms: the Extreme Scale Motes (XSMs) has a 7.3 MHz CPU and 4KB of RAM and uses a 433 MHz single-channel radio supporting a data rate of up to 38.4 KBps. All this is supported by the TinyOS operating system. The other platform is a Stargate[31] node with a 400 MHz CPU, PCMCIA-based 802.11b wireless card, and direct interface with the XSM sensor platform, whereas Linux is used for the operating system. The stationary array infrastructure can be coupled with one or more portable arrays for recording sensor data

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

(a) (b)

Figure 10.4. Kansei Sensor testbed and hardware. (Picture source: http://ceti.cse.ohio-state.edu/kansei).

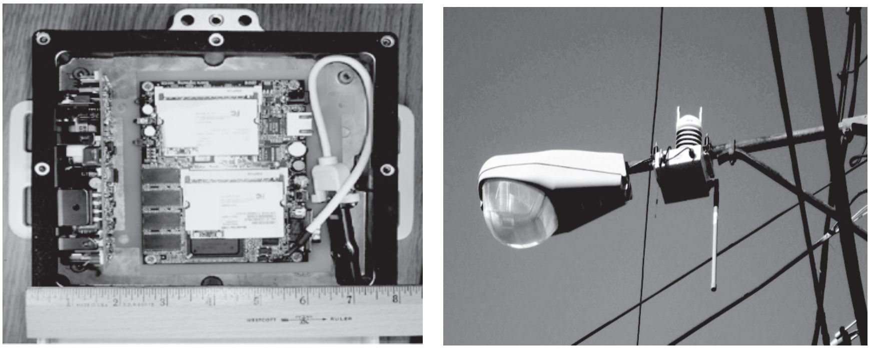
in situ and for field-testing sensor-network applications. The mobile platform consists of robot-mounted mobile nodes that can move on a Plexiglas base.

10.3.2.2 Software Infrastructure

The testbed uses the Kansei Director software to manage experiments. This provides support for scheduling of experiments, script-based execution, monitoring, data collection, and storage. For larger-scale experiments, the software also supports a hybrid simulation model that runs large-scale simulations on a host machine, interfaces with the actual testbed for physical events such as sensing or traffic delivery (while pausing the simulation state) until the event is completed on the real testbed, and reinserts the results from the physical testbed into the simulation engine. The testbed has a Web-based interface on which experiments can be scheduled and the results retrieved. Further details on the experimental framework and usage model can be found on the Kansei Web site.[30]

10.3.2.3 CitySense: Urban-Scale Wireless Network Testbed

CitySense testbed[32] is hosted in the city of Cambridge, Massachusetts, and provides an urban-scale wireless network testbed for experimental purposes (Figure 10.5). The deployment consists of about 100 radio nodes mounted at different locations in the city on street lights. These nodes have the ability to monitor physical events via sensors and are powered by street-light mountings. In contrast to traditional sensor platforms that are battery-operated with severe power constraints, this platform enables newer applications that can make use of the increased power and processing capability of the hardware.



(a) (b)

Figure 10.5. CitySense Sensor Node hardware; mounted on light pole. (Picture source: http://www.citysense.net/).

10.3.2.4 Hardware Infrastructure

The experimental platform used for the wireless nodes is based on Soekris net4826 motherboard running Linux OS with mini-PCI-based 802.11a/b/g radio cards. The nodes interface with a modular sensing hardware that can serve multiple modes (weather sensing using Vaisala WXT510 unit, or CO_2 sensing using Vaisala GMP343 units). These sensing units connect to the Soekris motherboard via its serial port.

10.3.2.5 Software Infrastructure

Unlike the previous testbeds that use wired network for the control plane activities including experiment deployment, management, monitoring, and data collection, CitySense architecture uses multihop wireless mesh techniques running optimized OLSR routing protocol on a dedicated 802.11 radio. Nodes periodically monitor their connectivity to the Web-based management software and reboot the connection with the system in case of missing network *heartbeats*. The other 802.11 radio is available for experimental purposes. A Web-based user interface provides real-time access to network statistics.

10.3.2.6 DieselNet: Outdoor Mobile Networking Testbed

The DieselNet testbed[33] in the city of Amherst, Massachusetts, is an outdoor mobile testbed with a focus on disruption tolerant networking (Figure 10.6). The testbed comprises wireless nodes installed in city buses that ply different routes. The nodes communicate to establish ad hoc connectivity with each other when in physical proximity, as well as through the use of dedicated relay nodes (stationary) deployed at strategic locations.

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

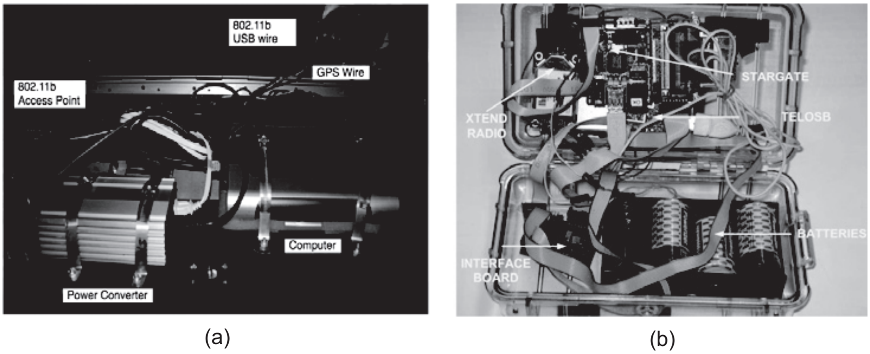


Figure 10.6. DieselNet Mobile Node in a bus and stationary relays. (Picture source: http://prisms.cs.umass.edu/dome/umassdieselnet).

10.3.2.7 Hardware Infrastructure

The wireless nodes installed in buses are HaCom Open Brick nodes with 577 MHz CPU, 256MB RAM, 40GB hard drive running Linux OS. The node has three radios: an 802.11b Access Point (AP) to wireless access to bus passengers, a second USB-based 802.11b interface that constantly scans the RF neighborhood for peer nodes in nearby buses, and a longer-range MaxStream XTend 900MHz radio to connect to the stationary relay nodes. The nodes also have GPS receivers to keep track of location. The relay nodes use a combination of COTS low-power platforms (Stargate PXA255 platform) and low-power microcontrollers. They are equipped with MaxStream long-distance radios to communicate with the wireless nodes in the buses.

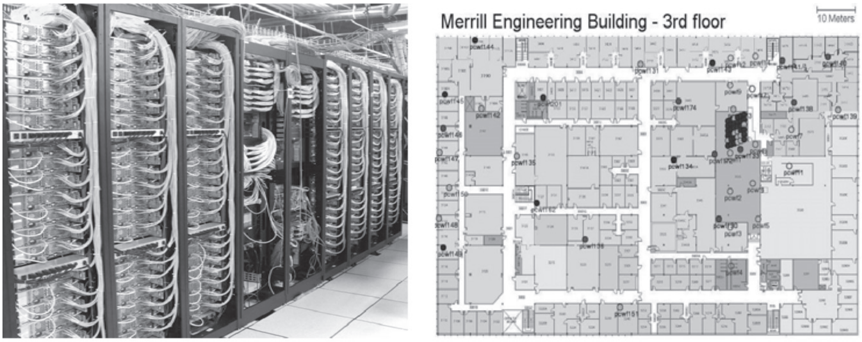
10.3.2.8 Software Infrastructure

The developed testbed control software allows the deployment of applications and tracking of node mobility, connectivity, and throughput. Further details on the testbed and common usage models can be found in Banerjee et al.[33] and University of Massachusetts DieselNet Web site.[34]

10.3.2.9 Emulab: Network Emulation Testbed

Emulab is an open-access testbed[35] hosted at the University of Utah in Salt Lake City (Figure 10.7). Originally designed as a wired network emulation testbed that offered controlled and deterministic network topologies that were configurable via scripts, this has now been expanded to include wireless devices including 802.11 nodes, USRP (Universal Software Radio Peripheral)[36]-based software radios, as well as sensor networks. A common user interface has been created to control access to the resources.

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press



(a) (b)

Figure 10.7. Emulab and wireless node deployment. (Picture source: http://emulab.net)

10.3.2.10 Hardware Infrastructure

The wireless nodes scattered throughout the building have two basic configurations: a) 3 GHz Pentium 4 based machines with 1 GB of DDR-400 RAM, 120 GB hard drive, and 2 802.11 a/b/g cards; and b) 600 MHz Pentium III machines with 256MB of PC100 RAM, 13GB hard drive, and 1 802.11 Atheros chipset-based a/b/g wireless device; 35 nodes have Zigbee and USRP/USRP2 devices attached. The 25 USRP devices have 900 MHz daughterboard whereas 10 USRP2 devices have 2.4 and 5 GHz (ISM band) daughterboard.

10.3.2.11 Software Infrastructure

To be able to use the wireless resources in parallel with minimal impact on other concurrent experiments, the software enables partitioning of the wireless medium by using separate frequency bands that do not overlap. Configurability is restricted to the options permitted via the Atheros chipset such as link rate selection, mode of operation, transmit power levels, and so forth.

In addition, users have access to USRP-based radios that have been deployed as shown in Figure 10.7, and the infrastructure allows users to utilize the bare cognitive radio hardware and download custom software onto the USRP nodes or as a managed wireless interface. Further details can be found at EMULAB Web site.[35,37,38]

10.3.3 VT-CORNET

VT-CORNET is a wireless communication network testbed based on 48 radio nodes scattered on four floors in a building at Virginia Tech. The primary focus of this testbed is to provide a tool for developing methods that allow efficient

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

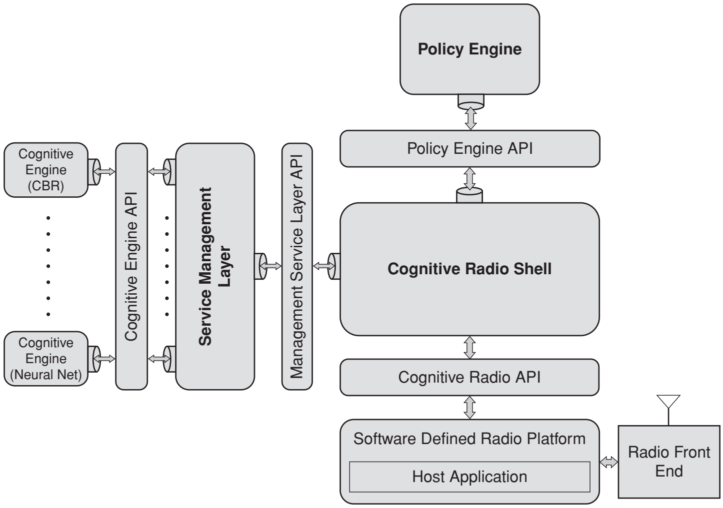


Figure 10.8. VT-CROSS architecture. (Picture source: http://wireless.vt.edu/coreareas/cognitive.html)

sharing of spectrum resources among heterogeneous devices and to maximize the speed and reliability of communications.

10.3.3.1 Hardware Infrastructure

The hardware infrastructure is based on USRP2 with custom daughterboard covering frequency range of 100 MHz to 4 GHz.

10.3.3.2 Software Infrastructure

The testbed uses modular open-source software framework[39] shown in Figure 10.8, which is based on a set of modules that use socket as intercomponent communication method. This allows for individual modules to be implemented in any programming language that supports TCP/IP socket abstraction. Currently, the system has the following components:

- Cognitive Radio Shell (CRS)
- Cognitive Engine (CE)
- Policy Engine (PE)
- Service Management Layer (MSL)
- Software-Defined Radio Host Platform.

The software design is based on Software Communications Architecture (SCA),[40] which is an architectural framework for standardizing the deployment, management, interconnection, and intercommunication of software application

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

components in embedded, distributed-computing communication systems. This enables programmable radios to load waveforms, run applications, and be networked into an integrated system, and provides a standard operating environment across all hardware, leading to interoperability by enabling the same waveform to be ported to all radios.

10.4 Global Environment for Network Innovations (GENI)

The testbeds described in the earlier sections provide an important research tool to study indoor wireless ad hoc and mesh networks, sensor networks, cognitive radios, and outdoor mobile scenarios in isolation. However, many of the new Internet application concepts involve a mixture of wireless end-users, physical sensing, and user mobility, thereby requiring integrated research infrastructure.

A recent NSF-sponsored multiphase initiative known as The Global Environment for Network Innovations (GENI)[41] addresses the need for building such an experimental network research infrastructure. The objective is to understand complex interactions between several different network technologies and protocols based on realistic usage models and identify next steps toward a novel architecture to meet the requirements of the evolving networking trends. The first phase of the project was a "Planning phase" (some time between 2005 and 2007) that resulted in initial design documents, draft funding proposals, and establishment of the GPO (GENI Project Office) at BBN.

The GENI vision that came out of the planning phase is to support a wide range of experimental protocols over heterogeneous substrates including fiber optics, high-speed routers, citywide experimental urban radio networks, computing clouds, and sensor networks. This will be in a form of a shared global facility available to experimenters with support for experiment deployment, measurements, and data analysis and sharing of information. To achieve this goal, GENI proposes the following core components:

- **Programmability** to enable experimenters to deploy custom software onto the experimental infrastructure and provide ability to control and modify various parameters and behavior of underlying network elements
- **Resource Sharing** to allow the simultaneous use of the infrastructure by multiple users and experiments while ensuring maximum isolation between these experiments. Thus, network bandwidth and CPU resources will be proportionally shared among various instances of experiments. This can be achieved using virtualization techniques where every unique experiment can get a "slice" of distributed network and computing resources proportional to the requirements of the experiment.
- **Federation** will enable an integration of heterogeneous resources that may be owned and operated by various organizations to create a global distributed

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

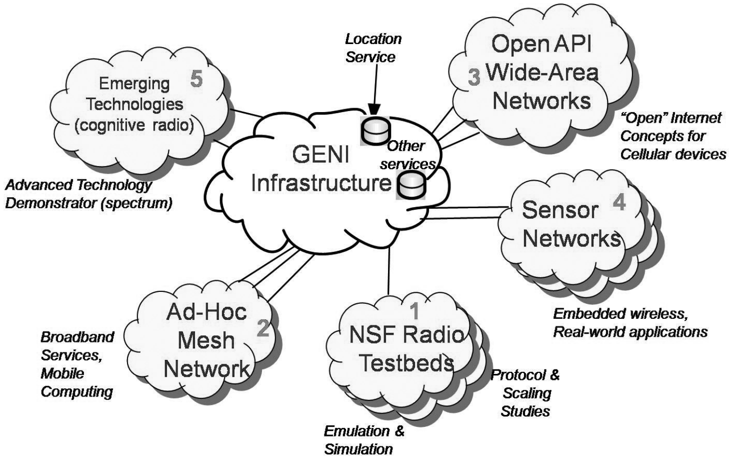


Figure 10.9. Experimental networks for future Internet research – Overview.

network of resources. This will require wireless testbeds to be integrated with the wide-area Internet infrastructure in order to facilitate end-to-end experimentation with new protocols (shown in Figure 10.9).

- **Integrated control and management framework** will allow discovery of resources for experimentation, creating a slice based on user request, initializing the experiment state, providing mechanisms for users to deploy and launch their experiments, define measurement points, and collect measurements at run-time from the experiment. Such an integrated control-and-management framework will require the ability to coordinate and control various heterogeneous network resources that may be governed by different usage policies and present them as a virtual slice to the experimenter upon request.

The strategy of the GENI GPO is based on spiral development, flexible funding model with large community involvement and organic network growth. This development plan is currently being executed with the goal of a large-scale network deployment over approximately two to three years. GENI phases (or spirals) last a year and are based on open solicitation process and peer review. In general, insights and experience from earlier phases are used to set specific goals for the later ones. To date, GENI phases consist of three sets of projects:

- **Planning** projects (approximately 2005–2007) focused on identifying research challenges and developing preliminary designs for programmable network deployments.
- **Spiral 1** projects[42] (approximately from October 2008 to September 2009) focused on proof-of-concept prototypes for technology risk reduction and to develop, integrate, and attempt to operate very rudimentary, end-to-end working prototypes.

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

- **Spiral 2** projects[43] (approximately from October 2009 to September 2010) with increased focus on deployment and experimental network operations.

We next describe ongoing efforts to meet the various requirements and design goals described earlier, as well as wireless-related projects from the two spirals.

10.4.1.1 Phase 1: Integration of Wireless and Wired Experimental Networks

As a first step toward achieving the global network of experimental resources, we describe ongoing efforts to integrate two different experimental network testbeds operated through two different control frameworks – PlanetLab[12] and ORBIT[23] – and support the execution of experiments that span both wired and wireless network elements. Each of the existing testbeds provides support for experimental flexibility and some level of network virtualization, and they need to be harmonized to work within the proposed future GENI model.

Key technical requirements for such an integrated testbed are as follows:

(1) virtualization of network resources (wired and wireless) to provide capabilities for support of multiple concurrent experiments ("slices") on the same set of nodes;
(2) integration of control and management across wired and wireless networks, providing research users with a single programming interface and experimental methodology.

10.4.1.2 Network Virtualization

Virtualization techniques are intended to share a set of computing and communication resources (CPUs, routers, links, networks) among multiple users with the appearance of dedicated, noninterfering allocation of resources. While VMWare[44] provides full virtualization where multiple OSs can be run on the same machine, its memory requirements restrict the number of simultaneous experiments per node. Paravirtualization techniques such as Xen[45] similarly use a "hypervisor" layer that can host multiple guest operating systems by controlling access to hardware and scheduling across physical CPU's access to controls, but suffers from similar memory usage constraints. A third approach, adopted by PlanetLab[12,13] and User Mode Linux,[46] is to virtualize at the level of system calls and provide reduced isolation in favor of supporting a much larger number of users per node. With regards to the virtualization of network resources, VLANs enable a network of computers to communicate even though they may be connected to different physical segments of a LAN. More recently, PlanetLab's virtual network access module isolates the traffic of multiple users from one another.

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

Virtualization of a wireless network is inherently different from the wired counterpart because the wireless medium is a broadcast medium, and isolating the wireless physical layer per user involves eliminating co-channel as well as adjacent channel interference from other collocated wireless devices. Several methods to extend the concept of virtualization into the wireless domain are described next, with the objective of supporting certain classes of service software and protocol experiments over a common set of experimental wireless networking resources.

10.4.1.3 Wireless Virtualization Approaches

Wireless network virtualization can be approached using a combination of methods that range from simple "virtual MAC" (VMAC) techniques (for restricted 802.11-based access point topologies) to more general space division multiple access (SDMA), frequency division multiple access (FDMA), and time division multiple access (TDMA) methods for separating "slices" in the network.

- **Virtual MAC (VMAC):** The VMAC technique is based on logical portioning of the radio channel based on individually assigned slice BSSIDs. Current 802.11 radio drivers provide support for up to 16 BSS per radio; however, topologies are restricted to fixed-star scenarios where an AP is configured in the normal infrastructure WLAN mode, and devices with WLAN cards connect to the AP with a single-hop wireless connection. Virtualized use of such a star topology wireless network may be appropriate for certain classes of long-term service or protocol experiments, but may not be particularly useful for wireless mesh or other experiments involving more complex or dynamically changing topologies.
- **Spatial diversity + VMAC:** To increase the number of slices that can be supported by an experimental wireless network, spatial separation can be combined with the previously described VMAC method. In this approach, multiple AP-based star networks coexist on the radio grid in which the AP-based clusters are separated by a distance greater than the radio interference range allowing them to coexist (on the same channel) without interference.
- **Frequency diversity + VMAC:** VMAC can also be combined with frequency diversity to support a larger number of simultaneous slices for a given set of radio nodes. One way of implementing this technique would be to use wireless nodes with multiple ($n > 1$) radio cards where the cards operate on orthogonal frequencies.
- **Time multiplexing:** The idea in TDMA-based approach is to share a given wireless node among multiple experiments using different time slots. For example, a wireless node can be logically partitioned into three virtual nodes by allocating three nonoverlapping time slots to three experiments. Initial

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

work from University of Wisconsin[49] presents a virtual TDMA-like multiplexing built on top of existing 802.11 MAC.

A combination of these techniques can also be used to increase the number of supported users per set of devices. The virtualization described earlier will allow multiple users to share the wireless grid resources with minimal interference. This model can be extended further so as to support wired networking experiments with large numbers of emulated wireless nodes at the edge.

10.4.1.4 Integrated Control and Management Plane

Currently operational wired testbeds such as PlanetLab employ "service oriented" network architecture and provide the users with the ability to run long-term experiments. This experimental testbed runs on top of the Internet as an overlay, thereby giving researchers access to (1) a large set of geographically distributed machines; (2) a realistic network substrate that experiences congestion, failures, and diverse link behaviors; and (3) the potential for a realistic client workload. Experimenters get access to a long-running slice of these distributed resources via a central slice creation and management utility.

In contrast, wireless experimental testbeds such as ORBIT employ an "exclusive access for limited time" model, via the ORBIT Radio Resource Management and Scheduling facility, where a user can get exclusive access to the entire resources for a limited duration of time. The ORBIT system also provides tools to define and describe experiments, including their required resources, deployment options, and measurement collection settings. This promotes a repeatable mode of running experiments and systematic data archiving.

To facilitate experimentation over these globally distributed wired and wireless resources, it is important to integrate these two different experimental models to provide a common and consistent abstraction for the experimenter. This requires the various control-and-management frameworks to coordinate access to the network resources, as well as a simple interface to enable users to run different experiments. To integrate these two different experimental models, the following approaches[48] are considered:

- wireless experiments requiring access to wired nodes;
- wired experimenters requiring access to wireless edge as a part of their experiment.

In viewing the wired network as an extension of the wireless testbed, the first approach (Figure 10.10) involves the use of a long-term "slice" in the wired infrastructure and interfacing the slice to the wireless testbed using a gateway. The ORBIT experimental infrastructure (described in Section 10.1) has been extended to communicate with nodes on the local subnet (wireless nodes) as well as remote PlanetLab nodes. Specifically, the naming/addressing scheme

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

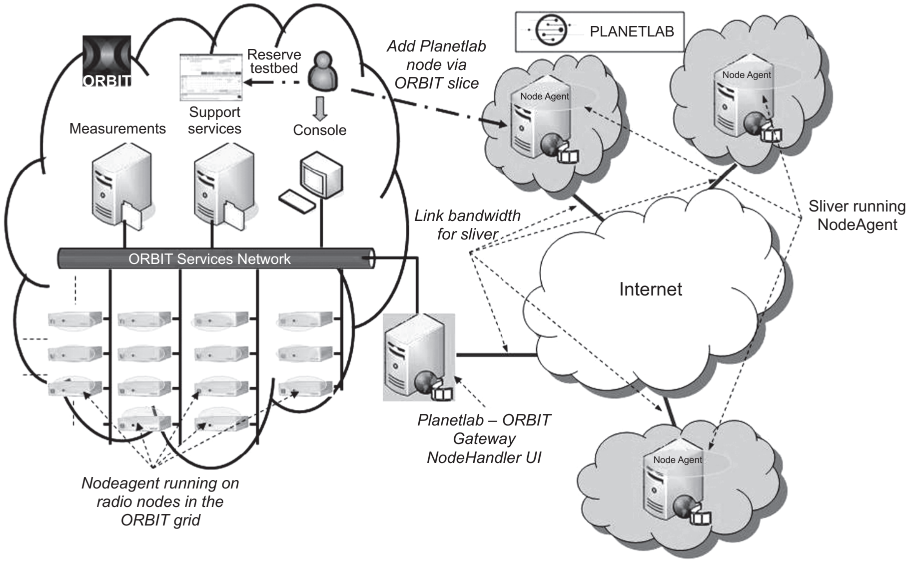


Figure 10.10. Wired extension to wireless experiment.

and communication protocol for the ORBIT experimental framework has been extended to allow access to geographically diverse nodes. A proxy service handles communication with remote wired nodes using GRE tunnels on behalf of the experiment controller, thereby enabling distant nodes to be transparently included as a part of the wireless experiment.

In the second approach, we consider wired users who require an abstraction of a wireless edge to be included in the experiment. The second model would provide a PlanetLab-ORBIT gateway as a node that users of a PL slice can access whenever they want an emulated wireless edge network. This gateway will provide abstractions for setup, control, and measurement on a specified wireless topology.

Proof of concept experiments conducted over this integrated network testbed is described in Mahindra et al.[48] This unified design can serve as a practical foundation for wired/wireless integration in heterogeneous testbeds in the bigger GENI concept.

10.4.1.5 Spiral 2 "Meso-Scale" Deployments

As part of Spiral 2, the GENI GPO launched two related campus build-outs based on OpenFlow[50] and WiMAX technologies, with significant commonality in the technologies employed and overlaps in the campuses chosen. In an effort to create a large-scale testbed, the GPO solicited resources from both Internet2 (I2) and National LambdaRail (NLR) to provide both bandwidth and operational support to connect the campus deployments into a larger, national research

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

networks. The main objectives of the current “meso-scale” deployments are as follows:

- create a large-scale experimentation infrastructure that will enable new forms of network science and engineering;
- support broad community participation and enable “opt in” by early users across 13 major campuses;
- ensure participation of equipment manufacturers, such as HP, Juniper, NEC, Arista, Cisco, and Nicira, to GENI-enable commercial equipment.

10.4.1.6 OpenFlow and Enterprise GENI

The hardware platform of choice for the network build-out is based on the OpenFlow architecture[50] enabling researchers to run experimental protocols in the networks they already use every day. OpenFlow is based on a standardized interface to control the forwarding table of an Ethernet switch. More specifically, OpenFlow provides control over individual flows at the individual device level. This provides the basis for multilayer network slicing, creation of virtual network slices that can be defined across a combination of physical layer, link layer, network layer, and/or transport layer flow rules.

As shown in Figure 10.11, the integration of OpenFlow into the GENI architecture consists of two components: a FlowVisor and an Aggregate Manager. The

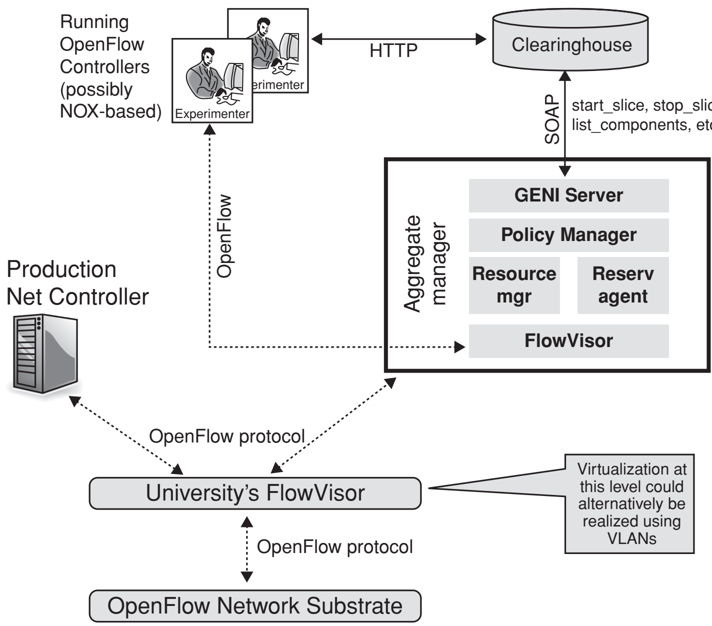


Figure 10.11. Enterprise GENI architecture. (Picture source: http://www.openflowswitch.org/wk/index.php/E-GENI)

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

FlowVisor partitions a physical OpenFlow switch into multiple logical (slice) switches and acts as a proxy for multiple OpenFlow controllers. FlowVisor is also responsible for guaranteeing isolation between multiple experiments running on the same switching fabric. The Aggregate Manage is a GENI-compliant OpenFlow controller that can control a subset of switch (network) resources allocated for experimentation. It enables GENI experimenters' access to OpenFlow environments along with campus/enterprise access to GENI experimental network infrastructure.

10.4.1.7 WiMAX

Extending the work on WiMAX virtualization undertaken Spiral 1,[42] the WiMAX meso-scale project in Spiral 2 will create an open, programmable, GENI-enabled "cellular-like" infrastructure across eight major research university campuses as shown in Figure 10.12. This project leverages a commercial 802.16e base station from NEC, replacing the standard WiMAX controller with an open GENI software implementation that supports virtualization and layer 2/3 programmability.

Open WiMAX base stations provide network researchers with wide-area coverage and the ability to support both mobile and fixed end-users. This will open up a path for direct "opt in" of student users in these campuses into GENI research experiments, via WiMAX modems and, as they become available, WiMAX handsets.

The core of this Spiral 2 project is the "GENI WiMAX base station kit" that consists of a NEC 802.16e base station (BS) with indoor and outdoor unit

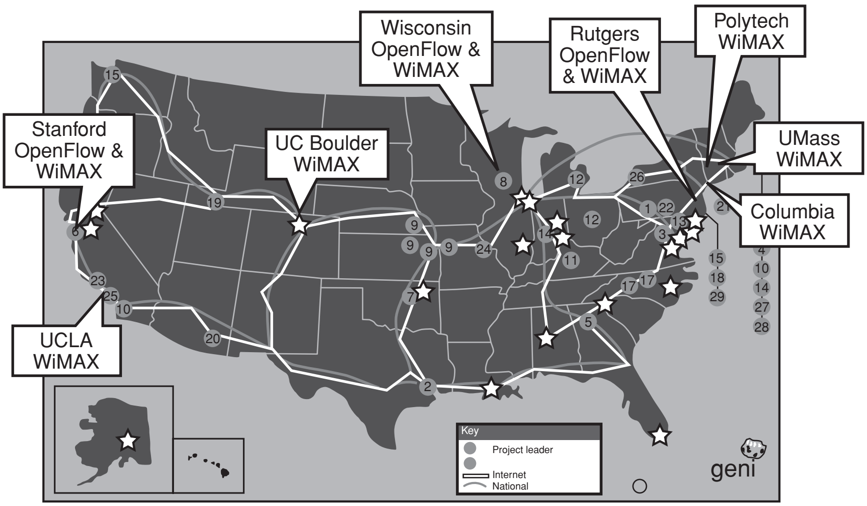
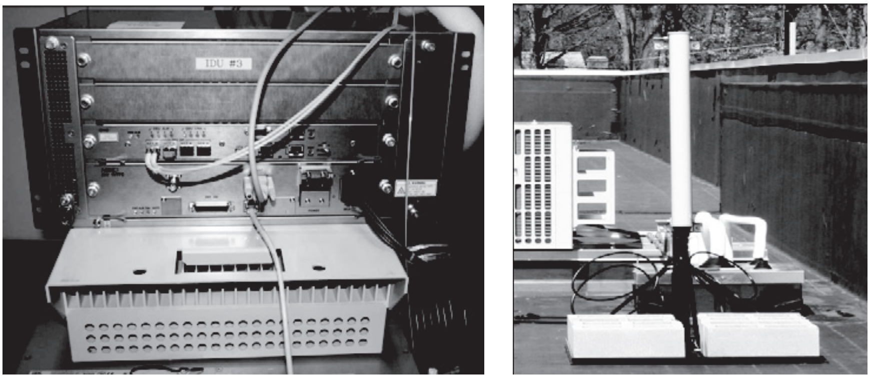


Figure 10.12. Proposed WiMAX deployments for GENI. (With permission from: http://groups.geni.net)

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press



(a) (b)

Figure 10.13. WiMAX base station.

(Figure 10.13a), a BS controller, antenna (Figure 10.13b), and a number of WiMAX clients. The BS is a 5U rack-based system that can be populated with up to three channel cards (sectors). It operates in the 2.5 GHz or the 3.5 GHz bands and can be tuned to use 5, 7, or 10 MHz wide channels. The WiMAX base station has an external Linux-based PC controller that runs the open API GENI control software offering programmable wireless networking capabilities and provides support for multiple GENI slices (virtualization). Each controller PC provides each experimenter's slice with its own virtual machine and the ability to control the respective radio resources allocated to each associated client. In the GENI context, each slice can be viewed as a dedicated BS (vBS) supporting multiple clients and multiple traffic types per client.

The WiMAX base stations will be integrated into OMF (ORBIT management framework), which is one of the supported GENI control frameworks. An experimenter will be able to access and create their own virtual WiMAX network through the portal and use available OMF tools for experiment control, management, and measurement to conduct their experiments. Further details can be found at www.Geni.net.[51]

10.5 Concluding Remarks

The large-scale wide area networking testbed with heterogeneous networking elements including wired and wireless devices can thus create infrastructure for entirely new forms of experimentation at a much larger scale than has previously been available. The new generation of a networking testbed described in this chapter is capable of supporting multiple future Internet prototypes in parallel, and can thus serve as an important tool not only for research-stage validations but also for early service deployments with real-world applications and end-users.

Worldwide deployment and eventual federation of these testbeds should lower the barrier for large-scale release of new protocols and services, thus improving the prospects for clean-slate networking ideas discussed in other chapters of this book.

References

[1] Kotz, D., Newport, C., Gray, R. S., Liu, J., Yuan, Y., and Elliott, C. 2004. Experimental Evaluation of Wireless Simulation Assumptions. *MSWiM 2004*.

[2] Pawlikowski, K., Jeong, H.-D. J., and Lee, J.-S. R. 2002. On Credibility of Simulation Studies of Telecommunication Networks. *IEEE Communications Magazine*, 40(1): 132–139.

[3] NS Simulator, http://www.isi.edu/nsnam/ns/

[4] Wireless and Mobility extensions to ns, http://www.monarch.cs.rice.edu/cmu-ns.html

[5] Zeng, X., Bagrodia, R., and Gerla, M. 1998. GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks. *Proceedings of the 12th Workshop on Parallel and Distributed Simulations – PADS '98*.

[6] OPNET, http:/www.opnet.com

[7] Qualnet, http://www.scalable-networks.com/

[8] Lundgren, H., Lundberg, D., Nielsen, J., Nordström, E., and Tschudin, C. 2002. A Large-scale Testbed for Reproducible Ad hoc Protocol Evaluations. *Proceedings of IEEE Wireless Communications and Networking Conference*.

[9] Zhang, Y., and Li, W. 2002. An Integrated Environment for Testing Mobile Ad-Hoc Networks. *Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*.

[10] Kiess, W., and Mauve, M. 2007. A Survey on Real-World Implementations of Mobile Ad-hoc Networks. *Ad Hoc Networks*, 5(3), 324–339.

[11] MIT Roofnet. http://www.pdos.lcs.mit.edu/roofnet/

[12] Peterson, L., Shenker, S., and Turner, J. 2004. Overcoming the Internet Impasse Through Virtualization. *Third Workshop on Hot Topics in Networking (HotNets-III)*.

[13] Huang, M. *VNET: Planetlab Virtualized Network Access*. http://www.planet-lab.org/PDN/PDN-05–029

[14] Rao, A., and Stoica, I. 2005. An Overlay MAC Layer for 802.11 networks. *Proceedings of ACM Mobisys*.

[15] Neufeld, M., Fifield, J., Doerr, C., Sheth, A., and Grunwald, D. 2005. SoftMAC – Flexible Wireless Research Platform. *Proceedings of the Fourth Workshop on Hot Topics in Networks (HotNets)*.

[16] VANU. http://www.vanu.com

[17] GNURadio. http://www.gnu.org/software/gnuradio

[18] Joint Tactical Radio System. http://jtrs.army.mil

[19] Minden, G. *KU Agile Radio Technology*. http://www.ittc.ku.edu/techreview2005/presentations/ Minden_Agile Radios.ppt

[20] Ackland, B., Bushnell, M., Rose, C., Seskar, I., Raychaudhuri, D., Tentzeris, M., Papapolymerou, J., Laskar, J., and Sizer, T. 2004. NeTS-ProWiN: High Performance Cognitive Radio Platform with Integrated Physical and Network Layer Capabilities. *National Science Foundation*.

[21] Wireless Open Access Research Platform, http://warp.rice.edu/trac

[22] Crawdad, A. *Community Resource for Archiving Wireless Data at Dartmouth*. http://crawdad.cs.dartmouth.edu/index.php

[23] Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., Siracusa, R., Liu, H., and Singh, M. 2005. Overview of the ORBIT Radio Grid

Testbed for Evaluation of Next-Generation Wireless Network Protocols. *Proceedings of the IEEE Wireless Communications and Networking Conference*.

[24] Ganu, S., Seskar, I., Ott, M, Raychaudhuri, D., and Paul, S. 2006. Architecture and Framework for Supporting Open-Access Multi-User Wireless Experimentation. *First International Conference on Communication Systems Software and Middleware*.

[25] Ott, M., Seskar, I., Siracusa, R., and Singh, M. 2005. ORBIT Testbed Software Architecture: Supporting Experiments as a Service. *Proceedings of IEEE Tridentcom 2005*.

[26] Singh, M., Ott, M., Seskar, I., and Kamat, P. 2005. ORBIT Measurement Framework and Library (OML): Motivations, Design, Implementation and Features. *Proceedings of IEEE Tridentcom* 2005.

[27] Lei, J., Yates, R., Greenstein, L., and Liu, H. 2005. Wireless Link SNR Mapping onto an Indoor Testbed. *IEEE International Conference of Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM)*.

[28] Rakotoarivelo, T., Ott. M., Jourjon, G., and Seskar, I. 2009. OMF: A Control and Management Framework for Networking Testbeds. *SOSP Workshop On Real Overlays and Distributed Systems*.

[29] White, J., Jourjon, G., Rakatoarivelo, T., and Ott, M. 2010. Measurement Architectures for Network Experiments with Disconnected Mobile Nodes. *Proceedings of IEEE Tridentcom* 2010.

[30] The Kansei Sensor Testbed. http://ceti.cse.ohio-state.edu/kansei/

[31] Stargate platform, http://platformx.sourceforge.net/Links/resource.html

[32] Murty, R., Mainland, G., Rose, I., Chowdhury, A. R., Gosain, A., Bers, J., and Welsh, M. 2008. CitySense: A Vision for an Urban-Scale Wireless Networking Testbed,. *Proceedings of the 2008 IEEE International Conference on Technologies for Homeland Security*.

[33] Banerjee, N., Corner, M. D., Towsley, D., and Levine, B. 2008. Relays, Meshes, Base Stations: Enhancing Mobile Networks with Infrastructure. *Proceedings of ACM Mobicom*.

[34] UMass DieselNet, http://prisms.cs.umass.edu/dome/umassdieselnet

[35] EMULAB, http://www.emulab.net/

[36] Universal Software Radio Peripheral, http://www.ettus.com

[37] http://users.emulab.net/trac/emulab/wiki/wireless

[38] http://users.emulab.net/trac/emulab/wiki/gnuradio

[39] VT-CORNET, http://wireless.vt.edu/coreareas/cognitive.html

[40] Software Communications Architecture, http://sca.jpeojtrs.mil

[41] Global Environment for Network Innovations (GENI) System Overview Document ID: GENI-SE-SY-SO-02.0 September 29, 2008.

[42] GENI Spiral One: http://groups.geni.net/geni/wiki/SpiralOne

[43] GENI Spiral Two: http://groups.geni.net/geni/wiki/SpiralTwo

[44] Devine, S., Bugnion, E., and Rosenblum, M. 1998. Virtualization System Including a Virtual Machine Monitor for a Computer with a Segmented Architecture. *US Patent 6397242*.

[45] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. 2003. Xen and the Art of Virtualization. *SOSP*.

[46] Dike, J. 2001. User-mode Linux. *Proceedings of the 5th Annual Linux Showcase and Conference*.

[47] Paul, S., Ganu, S., Kamat, P., and Royer, E. B. *Requirements for Wireless GENI Experiment Control and Management, GDD-07–43*. http://www.geni.net/GDD/GDD-07–43.pdf

[48] Mahindra, R., Bhanage, G., Hadjichristofi, G., Ganu, S., Seskar, I., and Raychaudhuri, D. 2008. Integration of Heterogeneous Networking Testbeds. *Proceedings of 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*.

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press

[49] Smith, G., Chaturvedi, A., Arunesh Mishra, A., and Suman Banerjee, S. 2007. Wireless Virtualization on Commodity 802.11 Hardware. *ACM WinTECH Workshop (co-located with ACM Mobicom)*.
[50] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. 2008. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, 38(2).
[51] http://groups.geni.net/geni/attachment/wiki/WiMAX/GENI_WiMax_System_Engg_v0.1.pdf

https://doi.org/10.1017/CBO9780511921117.011 Published online by Cambridge University Press