A Service Oriented Experimentation Framework for Virtualized WiMAX Systems*

Gautam Bhanage, Ivan Seskar, and Dipankar Raychaudhuri

WINLAB, Rutgers University, North Brunswick 08902, USA
{gautamb,seskar,ray}@winlab.rutgers.edu
http://www.winlab.rutgers.edu

Abstract. Testbeds for networking research allow experimenters to validate performance of algorithms and architectures under realistic conditions. Further, virtualizing such testbeds allows the deployer to improve utilization of the testbed while preserving reproducibility of the results and originality of the control environment. This paper describes an essential set of services for deploying a virtualized wireless testbed. It proposes (1) environment control, (2) virtual radio control, (3) slice feedback, and (4) a virtual radio isolation service as four fundamental services required for deploying these virtualized wireless testbeds. Using a virtualized WiMAX basestation as an example, this paper describes the design of the WiMAX-VM and the WiMAX-RF grid services which encompass the four fundamental services.

1 Introduction

Recent focus on experimental evaluation of research ideas as an intermediate step before deployment on actual networks has given rise to a large number of testbeds [1,2,3,4,5,6]. Such platforms allow the experimenter to evaluate the performance of their research ideas under realistic but controlled network conditions. This is also the motivation for the design and implementation of the GENI [7] federation, which consists of a large group of diverse, and geographically spread testbed components that are stitched together through a controlled framework.

In the context of most of these testbeds, it is observed that eventually the system designers of these testbeds have worked towards virtualizing them [8,1,3,9]. Virtualization of networking testbeds allows for better utilization of the resources. In such cases, the focus of most virtualized testbeds has been on wired network components, mainly because most of the network virtualization research has been focussed on the wired side. However, with the recent efforts towards using wireless virtualization for testbeds [1,9], we focus on addressing the issues involved in the design of services for streamlining the usage of such virtualized wireless testbeds.

* Research supported in part by the National Science Foundation grant#CNS-072505.

T. Korakis et al. (Eds.): TridentCom 2011, LNICST 90, pp. 152–161, 2012.

Specifically, the contributions of this paper are as described below:

1. Essential services: We describe a set of services that are essential for supporting a virtualized wireless testbeds.
2. WiMAX-VM and WiMAX-RF services: We present the design and implementation of the two services we implemented for our virtualized WiMAX testbed.
3. Use cases: We also present a set of experimental use cases describing how the proposed WiMAX-VM and WiMAX-RF services can be used by the experimenters.

Paper Organization: The rest of the paper is organized as follows. Section 2 provides a discussion on related work for testbed virtualization. Section 3 describes the different aspects of the virtualized wireless testbed that need to be provided through services. Sections 4 and 5 discuss design and implementation details of the WiMAX-VM and WiMAX-RF services implemented for the virtualized WiMAX testbed. Finally, Section 7 discusses conclusions.

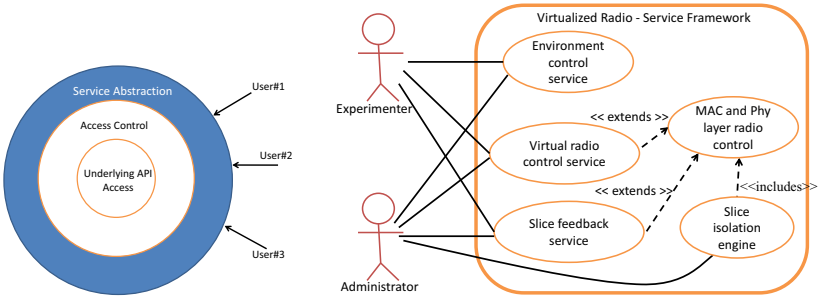
2 Related Work

Previous studies like Planetlab [5] and VINI [3], have focussed on designing virtualized testbeds for the wired world. Initial studies [9,10] have been done to determine the feasibility and performance that could be achieved if large wireless testbeds like ORBIT [2] were operated in a virtualized mode. The criterion for selection of specific virtualization platforms have been discussed here [11]. Another group of studies have focussed on discussion of the required architecture for virtualizing a WiMAX testbed [1,12], WiFi based virtual networks [13,14] and wireless cards [15]. We will now discuss previous studies that have proposed mechanisms for controlling virtualized resources in general.

In terms of control frameworks for testbeds, Emulab [4] provides an experimentation environment that supports simulation, emulation, and large scale evaluation of wide area networks. The OMF [16] platform to which our services are proposed as an extension, was developed in parallel for initially supporting ORBIT testbed specific equipment, but now spans across a wide range of wired and wireless equipment. The open resource control framework (ORCA) [6] uses a similar approach and can be used specifically to control certain shared computing environments. Similarly the planetlab and the Emulab control framework use software suites that can be used to control resources for short and long lived experiments across virtualized wired experiments and certain wireless experiments for the Emulab testbed. The WiMAX-VM and the WiMAX-RF services proposed in this paper are extensions to the OMF framework that will allow the experimenter to use a virtualized wireless testbed.

3 Service Oriented Experimentation Architecture

Before we begin a discussions on the services specific to the virtualized WiMAX testbed, we will lay down guidelines for the design of services for a virtualized wireless testbeds.



(a) A service.

(b) UML use case diagram for the four fundamental service on a virtualized wireless testbed.

Fig. 1. A description of a testbed service, and the interactions among the proposed four fundamental services for a virtualized wireless testbed

3.1 Generic Virtualized Radio Testbed Services

In our context, we define a *service* as a common, automated, and arbitrated mechanism built in the testbed to perform functions that would usually require administrator intervention. This *service*, as shown in the Figure 1(a) also helps the administrator to implicitly extend programmability into the slices[1], while simplifying interaction with the users of the virtualized testbed. Though some of the services described below (e.g. environment control) are not unique to virtualized wireless testbeds, we define a minimum set of services required to support the same:

- Environment Control: A service is needed for controlling the virtualized environment itself. Such a service is useful to the system administrator, and the users. This service will be responsible for abstracting the underlying virtualization technology. E.g. Consider two testbed sites which have similar capabilities. Site A may prefer to use OpenVZ [17] instead of KVM [18] preferred by site B as a platform virtualization. However, with the use of this environment control service a single user should be able to use both the sites with similar API calls while being agnostic to the underlying virtualization technology. This environment control service is also responsible for maintaining both control and performance isolation across different environments.
- Virtual Radio Control: This service will work within a slice and will be responsible for exposing control of the radio to the slice. Though this service will be radio type specific (E.g. WiFi, WiMAX), the API exposed by the service allows the experimenter to be agnostic to the underlying hardware.
- Slice Feedback Service: A feedback service is needed for generating and providing experiment related feedback to the slice. This service will provide the necessary measurement related feedback to the slice.

[1] From here on, we will use the term slices, users, and experimenters interchangeably.

– Virtual Radio isolation service: A service is required for isolating the radio resources used by each slice due to the inherent nature of the wireless medium. Such a mechanism is responsible for enforcing slice airtime quotas so that experiments performed with different virtual radios are repeatable[2]. Such services are required for all radios irrespective of whether they support QoS differentiation or not.

The interactions among these four fundamental services are as shown in the use case diagram in Figure 1(b). It can be seen that the slice feedback, virtual radio control, and the slice isolation services are largely connected to the underlying physical radio, while the environment control service operates independently. Also, an experimenter is only able to access certain features of each of these services as allowed by the credentials and testbed administration policy. All the fundamental functions may either be incorporated into a single service or could be designed as independent services. We will now provide a brief background on the virtual basestation based WiMAX testbed design, followed by a discussion on how we can incorporate the functions discussed above as services in the testbed.

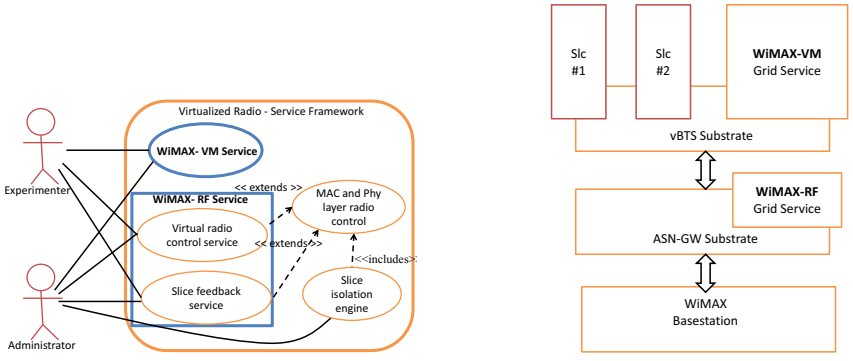
3.2 Virtual Basestation Design

The virtual basestation framework [1] is a platform for supporting experimentation on virtualized WiMAX networks. The virtual basestations are emulated by virtual machines in the virtual basestation (vBTS) substrate. This substrate is responsible for containing the slices and dynamically connecting to the physical basestation through the application service network (ASN) gateway. The ASN gateway in turn is responsible for forwarding packets to and from the physical basestation. The virtualized basestation architecture supports the following enhancements over the raw physical basestation. (1) An end to end layer-2 datapath is provided for forwarding MAC frames directly between the virtual basestation slices and the physical basestation. (2) A slice isolation engine described here [19] is provided for isolating the radio consumption across slices. This engine uses SNMP feedback from the basestation radio to determine radio usage per slice, and appropriately limit traffic from every slice to provide isolation.

3.3 WiMAX Testbed Specific Services

As per the guidelines described above, we build the WiMAX-VM (virtual machine control) service and the WiMAX-RF(radio control) service. The WiMAX-VM service is the environment control service needed to support the abstraction of the virtual basestation itself, while the WiMAX-RF service is built for controlling and monitoring the virtual radio related features from within every slice. The generic slice feedback service discussed above is included as a part of the WiMAX-RF service, and hence is not discussed separately. The generic virtual radio isolation service is already implemented for the WiMAX framework and is discussed

[2] It is important to note that the performance with this service will be repeatable only when the experimenter has some control over the RF environment in which these experiments are performed.



(a) Services interactions. (b) Services architecture

Fig. 2. The UML use case diagrams and the architectural layout diagram for describing interactions within the WiMAX-VM and the WiMAX-RF service

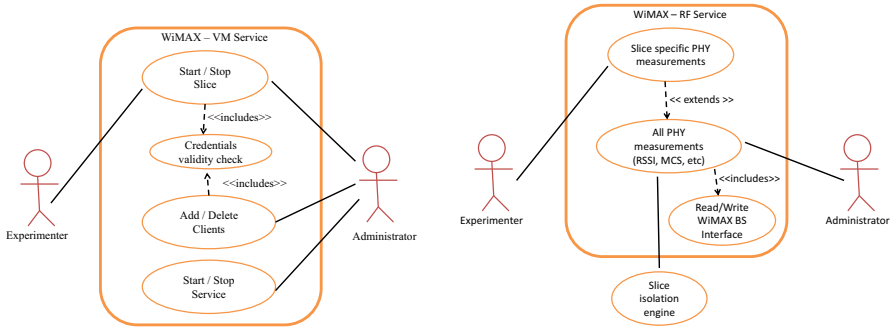
here [1,19]. We will now discuss the detailed design and implementation for the WiMAX-VM and the WiMAX-RF services.

The functional interaction between these services are based on the interaction between the fundamental services and are as described in the Figure 2(a). The WiMAX-VM and the WiMAX-RF services will be the user facing services and will be available independent of each other. These services will interact with internal functional components like the datapath control unit. As shown in Figure 2(b), we house the WiMAX-VM service on the vBTS substrate and the WiMAX-RF on the ASN-GW. More details on the implementation and the working of each of these services will be explained in later sections.

3.4 Desirable Features for Service Implementation

Before we begin a description of the services themselves, we lay down the guidelines for building testbed oriented service. Specific features which are desirable are:

1. Platform Independence: The platform providing services should be independent from the environment from where they are invoked. E.g. The service should support invocation from any environment, such as a Windows based or Unix based environment.
2. Pub - Sub Support: The service should support the publish - subscribe model for information exchange i.e. the experimenter should be able to express interest in some of the experiment parameters such as throughput, physical rate, or the occurrence of certain network events.
3. Measurement Latency: The services architecture should be able to provide measurements within certain worst case latency requirements. This requirement relates to the worst case response time of the system. Such features are essential for supporting experiments that require real time feedback from the system.



(a) UML use case diagram for the WiMAX-VM service.

(b) UML use case diagram for the WiMAX-RF service.

Fig. 3. The UML use case diagrams for describing interactions within the WiMAX-VM and the WiMAX-RF service

4. Incremental Design: The framework supporting the services themselves should support incremental design i.e the administrator should be able to deploy new services which will co-exist with the ones already deployed.

The OMF [16] platform allows us to provide web services while fulfilling all of the requirements discussed above. Even though our testbed users will possibly use a non-IP layer-3 for a datapath within experiments, the control path is still a conventional IP based network within which the *http* based web services will work properly. In the following sections we will discuss how the OMF framework is used for implementing the WiMAX-VM and the WiMAX-RF grid services.

4 WiMAX-VM Grid Service

4.1 Design Goals

The WiMAX VM service should allow the experimenter to control basic slice related functions like initializing the virtual machines in the slice, setting up a datapath from the vBTS substrate (the substrate running the virtual machines) to the physical basestations and adding/deleting wireless clients to the slice.

4.2 WiMAX-VM Service Architecture

The UML [20] use case diagram for this service is as shown in the Figure 3(a). The main actors in the system are the experimenter and the administrator. As shown, the experimenter will be able to access only the slice specific components

of the service, while the administrator will be able to access all the components of the service, including mechanisms to restart the service itself. The experimenter will be only able to access functions like start slice, stop slice, add and remove clients. The administrator on the other hand can access additional functions like start or stop service, and list slices, in addition to all the components available to the experimenter.

The service is deployed as an extension of the OMF [16] framework on the vBTS substrate as previously shown in Figure 2(b). This service can be run on the host or as a part of a virtual machine on the vBTS substrate and is invoked by an administrator. Apart from providing an encapsulation over the complexity and specifics of controlling one type of virtual machine technology, this service is also responsible for making sure that: (1) The requests from the clients are compatible with the policies specified by the administrators, and (2) Setting up the partial layer-2 datapath from the vBTS substrate, connecting it to the ASN-GW. This datapath setup allows wireless clients to transparently send and receive traffic to and from the slice through the physical basestation, and the ASN - GW. A set of example function invocations on the service are as described in the following section.

4.3 Selected Service Specifics

Each of the sample service requests described below return an XML response, which indicate the success/failure of the call, and additional status or information if requested. However, for the sake of brevity, we will exclude the responses.

1. $wget\ \ http://wm-asngw-02:5012/wimaxvm/initvms$
   - This function is invoked by the administrator, and is used to initialize the VM grid service. Checks for running VMs and initializes the layer-2 datapath on the machine. It also communicates with an intermediate datapath controller on the ASN-GW machine for setting up an end - to - end path to the WiMAX BTS from the vBTS substrate.

2. $wget\ \ http://wm-asngw-02:5012/wimaxvm/vmlist$
   - Allows the administrator to have a detailed view of the running VMs.
   - Shows VM statistics like up time, owner, slice ID, and interface details.

3. $wget\ \ http://wm-asngw-02:5012/wimaxvm/start?vmname=vm1$
   $wget\ \ http://wm-asngw-02:5012/wimaxvm/stop?vmname=vm1$
   - Starts/Stops VM instance
   - Configures VLANs on VM substrate

4. $wget\ \ http://wm-asngw-02:5012/wimaxvm/addclient?vmname=vm1\&clientmac=84:22:10.14.2b.9a$
   - Registers a client with MAC address "$84:22:10.14.2b.9a$" to the slice and adds the default service flow settings for the client.

5 WiMAX-RF Grid Service

5.1 Design Goals

As discussed in the design principles for building virtualized wireless testbed services, the RF service is responsible for making the higher layers in the experimentation framework radio hardware agnostic.

5.2 Architecture

The physical location of the service is on the ASN-GW as previously described in the Figure 2(b). The WiMAX-RF service is responsible for communicating with the basestation indoor unit for fetching the radio related parameter. The UML use case diagram for the service is as shown in the Figure 3(b). As before the actors in the system are the experimenter, and the administrator of the services. As shown in the diagram, every function invocation on the service by the experimenter is validated for proper permissions before being responded to. This ensures that the experimenter is only able to access RF performance measurements for clients owned by her own slice only. The administrator on the other hand will be able to access all the functions supported in the WiMAX-RF service.

As with the WiMAX-VM service, the WiMAX-RF service is implemented using the OMF framework. Specific functions such as getting radio conditions of the clients belonging to specific slices are obtained partially through an SNMP interface to the basestation. The rest of the queries are satisfied through a command line application that is able to query and set basestation specific parameters[3]. We will now discuss a couple of commands that are supported by the service.

5.3 Selected Service Specifics

The WiMAX-RF service supports two primary groups of functions. The *get* functions are useful for obtaining information form the framework, while the group of *set* functions can be used to reset parameters. Some calls are as discussed below:

1. $http://cons-wm-01:5052/wimaxrf/get?arq$
   - This call queries for the overall *arq* settings on the basestation.
   - A sample response will include the type of ARQ enabled on the basestation and the retry limits and their timeout settings.
   - Since this is a read query, all slices and administrators can issue this query.
2. $http://cons-wm-01:5052/wimaxrf/set?bsid$
   - This is an example of a *set* function on the basestation.
   - This function can be invoked only by the administrator since it changes the overall basestation settings and affects all the slices.

[3] Though the exact set of parameters that can be controlled with this service are dependent on the basestation hardware, we design for a broad subset of radio control parameters like throughput, downlink profile control, and basic RF measurements that will be supported by most basestations.

Table 1. Comparison of wireless experiments for the virtual basestation framework

Feature/Experiments	Our Services	Additional Logic	Not Supported
Handoff Emulation	√	√	-
Security Experiments	√	√	-
Network Coding	√	√	-
Mobility and Routing	√*	√	-
Rate And Power Control	-	-	√
Wireless Applications	√	√	-
PHY Measurements	√	√	-
MAC Parameter Control	√	-	-
Transport layer Modification	-	√	-

* Our RF service supports design of cross layer routing and the VM service supports transparent route setups.

6 Usage

Experimental Use Cases: A discussion on how a broad set of wireless experiments may be emulated on a virtualized testbed is as shown in the table 1. We see that most of the wireless experiments which can be emulated on a virtualized testbed will benefit from our services framework. A few selected experiments like rate and power are not supported in a virtualized wireless testbed since they will affect all slices. Modification and evaluation of transport layer protocols on a per slice basis is supported by the virtual basestation design. However, in this case the services are not very useful, except possibly for using cross layer feedback for transport layer adaptation. In other cases like the ones with handoff emulation, or mechanisms for implementing physical layer security, the WiMAX-RF service will be very useful for measuring physical layer parameters.

Mapping Algorithms: The services provided in our framework can also be coupled to the outputs of slice resource allocation algorithms. These algorithms are responsible for checking current resource utilization which can be determined using our WiMAX-RF service, and the allocation can be done using both the WiMAX-VM and the WiMAX-RF service

7 Conclusions and Future Directions

A service oriented approach for the design of virtualized wireless testbeds is presented. Four fundamental services are required for environment control, virtual radio control, slice feedback, and virtual radio isolation. Using a WiMAX basestation as a proof of concept, we show how these four fundamental functionalities when incorporated in our WiMAX-VM, and WiMAX-RF services are useful for setting up the testbed. We also discuss how different wireless experiments can be supported on the virtualized WiMAX testbed. In future, we plan to integrate the proposed services as fully functional parts in the GENI experimental framework.

References

1. Bhanage, G., Seskar, I., Mahindra, R., Raychaudhuri, D.: Virtual Basestation: architecture for an open shared wimax framework. In: ACM Sigcomm Conference, VISA Workshop, New Delhi, India (September 2010)
2. Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., Siracusa, R., Liu, H., Singh, M.: Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In: WCNC (March 2005)
3. VINI, a virtual network infrastructure, `http://www.vini-veritas.net/`
4. White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C., Joglekar, A.: An integrated experimental environment for distributed systems and networks. In: Proceedings of OSDI, Boston (December 2002)
5. Peterson, L., Muir, S., Roscoe, T., Klingaman, A.: PlanetLab Architecture: An Overview. Technical Report PDN–06–031, PlanetLab Consortium (May 2006)
6. The geni orca control framework, `http://www.nicl.cs.duke.edu/orca/`
7. GENI design principles, `http://www.geni.net/`
8. Hibler, M., Ricci, R., Stoller, L., Duerig, J., Guruprasad, S., Stacky, T., Webby, K., Lepreau, J.: Large-scale Virtualization in the Emulab Network Testbed. In: Proceedings of USENIX (2008)
9. Mahindra, R., Bhanage, G., Hadjichristofi, G., Seskar, I., Raychaudhuri, D., Zhang, Y.: Space Versus Time Separation for wireless virtualization On an Indoor Grid. In: Next Generation Internet (NGI) testbeds (March 2008)
10. Mahindra, R., Bhanage, G., Hadjichristofi, G., Ganu, S., Kamat, P., Seskar, I., Raychaudhuri, D.: Integration of heterogeneous networking testbeds. In: Proceedings of TridentCom 2008, pp. 1–6 (2008)
11. Bhanage, G., Seskar, I., Zhang, Y., Raychaudhuri, D., Jain, S.: Experimental evaluation of openvz from a testbed deployment perspective. In: 6th International Conference of Testbeds and Research Infrastructure (ICST Tridentcom), Berlin, Germany (May 2010)
12. Kokku, R., Mahindra, R., Zhang, H., Rangarajan, S.: Nvs: a virtualization substrate for wimax networks. In: Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking, MobiCom 2010, pp. 233–244. ACM, New York (2010)
13. Bhanage, G., Vete, D., Seskar, I., Raychaudhuri, D.: SplitAP: leveraging wireless network virtualization for flexible sharing of WLANs. In: IEEE Globecom 2010 - Next Generation Networking Symposium (GC10 - NGN), Miami, Florida, USA (December 2010)
14. Smith, G., Chaturvedi, A., Mishra, A., Banerjee, S.: Wireless virtualization on commodity 802.11 hardware. In: Proceedings of Wintech, pp. 75–82. ACM, New York (2007)
15. Chandra, R.: Multinet: Connecting to multiple ieee 802.11 networks using a single wireless card. In: IEEE INFOCOM, Hong Kong (2004)
16. Rakotoarivelo, T., Ott, M., Jourjon, G., Seskar, I.: Omf: a control and management framework for networking testbeds. SIGOPS Oper. Syst. Rev. 43, 54–59 (2010)
17. OpenVZ instruction manual, `http://wiki.openvz.org/`
18. Kernel virtual machines, `http://www.linux-kvm.org/page/Main_Page`
19. Bhanage, G., Daya, R., Seskar, I., Raychaudhuri, D.: VNTS: a virtual network traffic shaper for air time fairness in 802:16e slices. In: IEEE ICC - Wireless and Mobile Networking Symposium, South Africa (May 2010)
20. Unified modeling language 2.2, `http://www.omg.org/spec/UML/2.2/`