Software Services for Experimentation on an Open WiMAX Base Station

BY MANASI JAGANNATHA

A thesis submitted to the

Graduate School - New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Professor D. Raychaudhuri

and approved by

____

____

____

New Brunswick, New Jersey

October, 2010

ABSTRACT OF THE THESIS

Software Services for Experimentation on an Open WiMAX Base Station

By Manasi Jagannatha

Thesis Director:
Professor Dipankar Raychaudhuri

This thesis addresses the software services required for programming the Open WiMAX Basestation. Specifically, it will discuss the software design methodology used to provide extended control of the virtual basestation abstractions through a centralized control mechanism. It also provides means by which an end to end layer-2 datapath can be setup to reach the clients from the virtual basestations while keeping the design decoupled from the underlying WiMAX hardware itself.

Using an OMF(cOntrol Management Framework) based grid service, the architecture supports two major sets of functions for providing programmability to the experimenter: slice management and radio control, we classify these by running separate services. The *wimaxvm* service which is the main contribution of this thesis, allows for automated slice management and IP independent data-path control. Creation of an IP independent datapath is critical for the design, since this helps the architecture to simultaneously support multiple stacks with different layer-3 and above protocols. By implementing components hosted across two different physical substrates, the prototype service shows that it is feasible to decouple the slice management, and radio control - monitoring functionalities.

The control and management framework (OMF) is used as an underlying platform, along with other system components such as the click modular router and independent control components like the datapath generator, the service prototype is implemented on a Linux platform and can be hosted on the same machine running the VM substrate.

Sample experiment scenarios are emulated using the prototype service for showing ease of API usage. One experiment shows how handoff mechanisms can be implemented using the API provided by the framework. Using a standard ruby based script around our language independent API an independent handoff mechanism is implemented in every virtual basestation. Finally, the thesis demonstrates how an independent video on the edge service can be setup using the API to reduce rate matching costs by avoiding independent rate matching for clients. Results show that the PSNR value of the video steam at the client is maintained high and hence improves the quality of the video.

Acknowledgements

First and foremost, I would like to express my sincere Thanks to my advisor, Prof. Dipankar Raychaudhuri for his constant support and guidance. I would also like to thank Ivan Seskar for being patient with my endless questions.

Next I would like to thank my friend Gautam, for all his support and guidance. I would like to thank my parents for believing and supporting me at each phase of my life.

I would like to thank my friends who have been my extended family. A special thanks to Chandru.

Dedication

My parents, friends and Bill Watterson

Abbreviations

Mb : Mega Bites

IP : Internet Protocol

MAC : Medium Access Control

Mbps : Mega bits per second

RTT : Round Trip Time

TCP : Transmission Control Protocol

UDP : User Datagram Protocol

ORBIT : Open Access Research Testbed for Next-Generation Wireless Networks

WINLAB : Wireless Information Networks Laboratory

OMF : cOntrol and Management Framework

GRE : Generic routing Encapsulation

WiMAX : Worldwide Interoperability for Microwave Access

GENI : Global Energy Network Institute

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Networking research, until recently, has to a large extent relied on short range Wi-Fi connectivity primarily because of its low cost as well as the wide availability of

software support. However, the recent surge in open wide area wireless network connectivity is fuelling novel and innovative experimentation with the creative goal of the cellular- internet convergence paradigm. While the convergence between mobile and internet worlds is potentially very lucrative, there is a lot of research involved in expanding networks to cope with the growth in traffic. This kind of research would require a WiMAX wireless test bed for researchers.

ORBIT (open access research test bed for next generation wireless networks) is a network of 400 nodes arranged in a square 2-D grid. The project conceived and developed at Winlab, Rutgers University affords researchers the ability to use an online scheduler to reserve time slices on the grid to access any of its 400 nodes. During the access time slot the researcher can image all the radio nodes ,with the goal of performing experiments on a variety of simulated scenarios.

GENI (Global Environment for Network Innovations) is a virtual laboratory at the frontiers of network science and engineering for exploring future internets at scale.[1] The GENI project aims to leverage a state of the art WiMAX base station product from NEC to develop an open source, programmable, and virtualizable cellular base station node. One of the primary motivations behind the GENI base station node is to support flexibility in experimentation independent of the complexities in the underlying base station architecture. With the precipitous growth in cellular services the architectures enabling cellular-internet convergence are quintessential for the future of any internet research agenda.

1.1:Project background and motivation:

To facilitate WiMAX-based experimentation, a WiMAX base station built by NEC is used. A key requirement for this project is "network virtualization", which makes it possible for many experimenters to use the test bed resources, such as the WiMAX base station, simultaneously. Accordingly, this project is aimed at the design and evaluation of virtualization methods for the WiMAX base station deployed in the ORBIT/GENI outdoor test bed.

Virtualization of the WiMAX base station provides a convenient approach to providing separate environments for independent experimental users. There are several problems associated with virtualization of a network device such as a router, or a base station.

Ensuring fairness of policies across virtual networks is tricky for a cellular base station since the channel changes continuously for a mobile device, thereby consuming varying amounts of resources. Allocation of resources to virtual networks is further complicated by the flow and packet scheduler algorithms which are implemented in a commercial base station.

Another important aspect of the design is support for multiple service classes. Each experimenter should be able to include a wide range of service classes as part of integrated ORBIT test bed experiments.

1.2: Contributions of the thesis:

One of the main contributions of this thesis is an implementation of software services that enables experimenters to use the Open Wimax base station seamlessly. We have used the existing Orbit Management Framework (OMF) to host the services required to setup and control experiments.

The Slice Manager Grid Service is a user based API that is used to create slices and adds clients to the slice. The Radio Resource Management API is used to get useful base station and client-based measurements. This thesis proposes a novel design and implementation of the Slice Manager Grid Service and the Data Path Generator (discussed in Chapter 2). Fair resource sharing among independent co-existing experiments is implemented using the above mentioned software services. A thorough performance evaluation of the software services and its possible uses are also illustrated.

1.3: Thesis outline:

The thesis is organized as follows. Chapter 2 provides an introduction to the problem and a high level overview along with the motivation behind this thesis. Chapter 3 introduces the design and implementation of the proposed architecture. Chapter 4 provides a performance evaluation of the proposed open Wimax base station framework and design. The thesis concludes in chapter 5 while providing possible avenues for future research and potential extensions to the proposed framework.

CHAPTER 2: PROJECT BACKGROUND AND MOTIVATION

2.1: Orbit Test bed:

For a long time the wireless network research community has relied on simulation results from platforms like NS2, which may not always be able to emulate real world scenarios. The ORBIT project provides a flexible wireless network test bed that is open to the experimental research community. Repeatability of wireless experiments is clearly a deciding factor in evaluating the performance of the test bed.

The steps involved in setting up an experiment on one of the nodes in the ORBIT framework is as shown in figure 2.1[1]. The experimenter writes a script which provides a description of the experiment, its static and dynamic parameters, and the set of measurements that need to be collected.

The experiment description is used to configure and assign various resources to the experiment. The experiment description defines what parameters need to be measured.

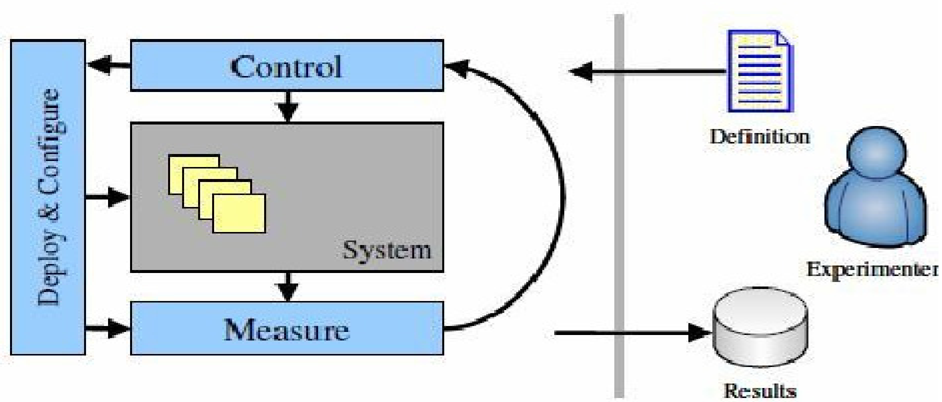


Figure 2.1 Model of experiment

Hence the ORBIT framework facilitates the easy implementation of protocols and experiments that need to be evaluated.

[1] Please see [26]

2.2: OMF framework

OMF (Orbit Management framework) is implemented to control, manage and get measurements from the nodes used in the experimental setup. The OMF architecture consists of 3 logical planes:

1. Control

2. Measurement, and

3. Management.

The Control plane in OMF is dedicated to the control of experimental executions, while the Management plane is responsible for managing the infrastructure. In other words, the Control plane includes the OMF tools that a researcher uses to describe his/her experiments, and the OMF entities responsible for orchestrating it.

The Measurement plane includes the OMF tools to execute an experiment, and the corresponding OMF entities to collect and store measured data. Finally, the Management plane includes the OMF functions and entities to provision and configure the resources, which are provided by the test bed facilities and used by the experiments.

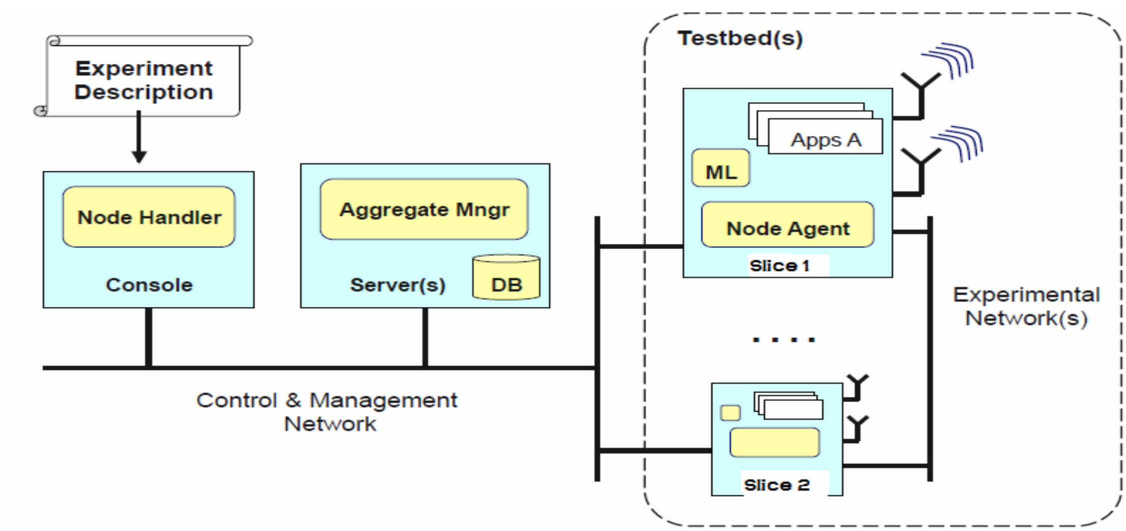


Figure 2.2 OMF framework schematic representation[2]

[2] Please see [25]

The OMF service is used to implement the user based Slice Control API and the Radio Resource Management API. OMF provides a platform to implement a grid service which can be accessed independent of the location in which the actual service is implemented. The software services for the Open WiMAX testbed are hosted using the OMF platform. This is a conscious design choice made to make the integration of ORBIT and the WiMAX testbeds easy in the future.

2.3: Virtualization of the WiMAX base station

The aim of virtualization of the WiMAX base station is to help multiple experimenters time share the resources of the base station. Each experimenter is given a slice which represents a controllable virtual base station whose resources can be used independent of the co existing slices. Abstraction is a feature that aims to provide each experimenter an illusion of owning the entire Base station and its resources at least in a limited way. Programmability has a goal of providing sufficient degrees of freedom to experimenters. Each experimenter has the ability to control some of the broad BTS parameters. Additionally, each experimenter can add a set of clients and custom service flows. An IP independent protocol stack is implemented to provide experimenters using IP independent protocols, the freedom to implement their own protocol(s). A grid service is provided for creation, destruction, and maintenance of slices. Clients can be added and slice allocation control can be done using the grid service.

2.4: WiMAX base station and its interfaces

Figure 2.3 depicts a schematic representation of the WiMAX base station and its connection to the rest of the network. The WiMAX base station is connected to an access network with Layer2 switched connectivity using Ethernet. The figure indicates three distinct interfaces associated with the WiMAX base station. The first is the control interface for experimenter access to virtual networks (slices) supported by the external base station controller. This is the primary interface relevant to an experimenter, and is based on the cOntrol and Management Framework (OMF)[3]. The second interface is the R6+ interface by which the base station controller communicates with the base station hardware (which includes an internal controller running a proprietary NEC operating system and control/management software). The R6+ interface exposes the hardware features such as assignment of MAC/PHY resources (*i.e.* OFDMA time-frequency slots, power levels, service classification, *etc.*) to each flow, as well as management interfaces for initial configuration, scheduler policy selection, and queue management.

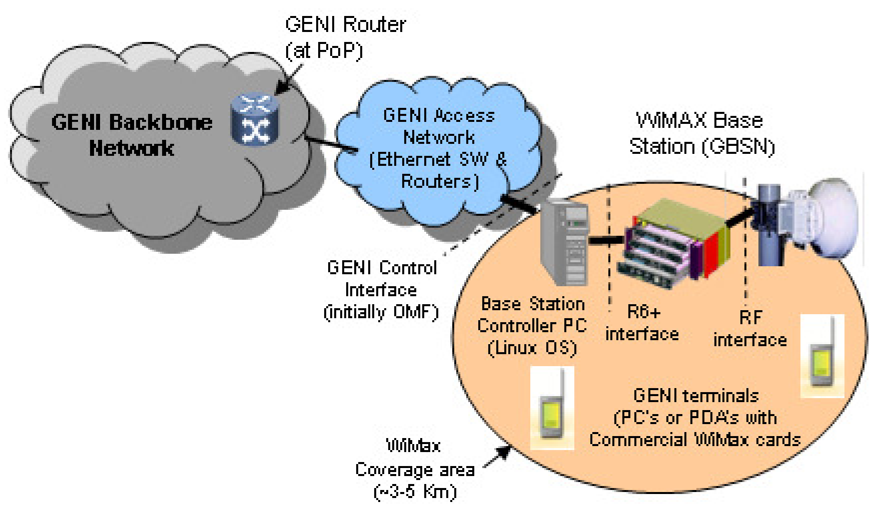


Figure 2.3 High level description of WiMAX base station and its interfaces to the GENI network

The NEC WiMAX base-station hardware is a 5U rack-based system which consists of multiple Channel Cards (CHC) and a Network Interface Card. The shelf can be populated with up to three channel cards, each supporting one sector for a maximum of three sectors [2]. The BS operates in the 2.5 Ghz or the 3.5 Ghz bands and can be tuned to use either 5, 7, or 10 Mhz channels. At the MAC frame level, 5 msec frames are supported as per the 802.16e standard. The TDD standard for multiplexing is supported, where the sub-channels for Downlink (DL) and Uplink (UL) can be partitioned in multiple time-frequency configurations. The base station supports standard adaptive modulation schemes based on QPSK, 16QAM, and 64QAM. The interface card provides one Ethernet

Interface (10/100/1000) which will be used to connect to the high performance PC that is used to control the device. The base station has been tested for radio coverage and performance in realistic urban environments and is being used in early WiMAX deployments – typical coverage radius is ~3-5Km, and peak service bit-rates achievable range from 15-30 Mbps depending on operating mode and terrain [2].

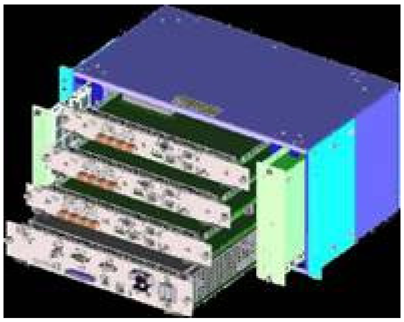


Figure 2.4 NEC's Release 1 802.16eBS

2.5: External Base Station Controller Architecture

The controller provides support for multiple slices assigned to the WiMAX node. Each slice runs within its own virtual machine. Each VM is capable of providing multiple virtual interfaces, so that programs loaded on a slice that run within a virtual machine can emulate a router of its own, and perform routing. Virtual interfaces are mapped to physical interfaces based on the next hop for a virtual interface. The controller will receive IP packets from the base station on the R6+ interface mentioned earlier. When a packet is received, it will be forwarded to the appropriate slice for further processing.

A Layer 2 routing method is used to avoid restricting the protocol stack to the IP stack only. The outgoing IP packets from a slice will be placed on queues specific to a virtual interface. Outgoing packets on virtual interfaces mapped to the Layer 2 interface of the WiMAX base station will be tagged so that they can be assigned traffic class and bandwidth parameters (BE, ertPS, rtPS *etc.*) as determined by the flow CID (Connection ID).

Experimenters are able to access the WiMAX network through the ORBIT portal, which provides scripting, experiment control, management, and measurement tools necessary to run an experiment.

2.6: WiMAX Overview

2.6.1: Introduction

The past decade has seen a rise in user mobility and the need to stay connected at all times. This has resulted in increased interest in the IEEE 802.16e mobile Worldwide Interoperability for Microwave Access (WiMAX) systems. The IEEE 802.16 standard was set up with a view to develop an air interface standard for wireless broadband [5]. This standard aims to provide Broadband Wireless Access (BWA) and is hence considered an attractive replacement for wired broadband services. It has the advantage of being easily deployable and hence can act as a last-mile broadband wireless access technique in high population cities, and also in areas where there is no prevailing infrastructure for wired connections.

The IEEE 802.16e standard forms the basis for the WiMAX solution for mobile applications and is referred to as mobile WiMAX. Some of the salient features of Mobile WiMAX are :

1. High Data Rates:

The inclusion of MIMO antenna techniques along with flexible sub-channelization schemes, and advanced coding and modulation, all enable the Mobile WiMAX technology to achieve peak PHY data rates of around 25 Mbps and 6.7 Mbps for the downlink and the uplink, respectively. These peak theoretical PHY data rates are achieved when using 64 QAM modulations with rate 5/6 error-correction coding, with a specified DL/UL ratio and bandwidth.

2. Quality of Service (QoS):

One of the main premises of IEEE 802.16 MAC architecture is Quality of Service (QoS). The MAC layer has a connection oriented architecture which can support a variety of applications including multimedia services. The system offers specific support for constant bit rate, variable bit rate, real-time, and non-real-time traffic flows, in addition to best-effort data traffic. The MAC can support a large number of multiple connections, each with its own QoS requirement.

3. Scalability bandwidth and data rate support:

WiMAX allows scaling of data rates depending on current channel conditions. OFDM supports this scalability through changing the size of the Fast Fourier Transform based on channel conditions. Different networks have different bandwidth allocations, and scaling is essential to support roaming.

4. Security:

WiMAX has a robust privacy and key management protocol and also supports encryption using Advanced Encryption Standard (AES). Provision for authentication architecture based on Extensible Authentication Protocol is also provided. This allows for various user credentials like digital certificates, user name / password, smart cards, *etc*.

5. Mobility:

Mobile WiMAX needs to ensure that real time applications like VoIP can handle mobility, without significant degradation in service. This is supported using optimized handover schemes that have latencies that are less than 50 milliseconds.

2.6.2: WIMAX Physical Layer

The WiMAX physical layer is based on Orthogonal Frequency Division Multiplexing (OFDM). OFDM is ideal for Non Line Of Sight (NLOS), high data rate transmissions. OFDM is used in a number of commercial broadband systems like DSL, Wi-Fi, *etc*. In OFDM, a high data rate bit stream is divided into multiple lower data rate bit streams, which are modulated on separate carriers, which are referred to as tones or subcarriers [6]. OFDM provides a number of advantages in comparison to other solutions used for transmitting at high speeds. One of the main factors is the graceful degradation of performance as the delay spreads exceed the values that it was originally designed for. It is also robust against narrow band interference and is suitable for coherent demodulation.

The subcarriers can be divided into groups of subcarriers that are referred to as sub-channels. Sub-channels can be made from subcarriers that are either pseudo-randomly generated or are contiguous. Different sub-channels can be allocated to different users to support a multi-access scheme. This scheme is used in Mobile WiMAX and is referred to as OFDMA. Both uplink and downlink sub channelization is supported in Mobile WiMAX [6].

Allocating sub-channels to users based on their frequency response is termed as Adaptive Modulation and Coding (AMC). This can be used to enhance the channel capacity by varying the sub-channel according to the Signal to Noise Ratio (SNR) [5]. The Mobile WiMAX air interface supports both TDD and FDD modes [6]. In the TDD mode, the downlink frame is followed by a small guard interval and then the uplink frame is transmitted; whereas in the FDD mode, the uplink and downlink frame are transmitted simultaneously.

The TDD mode is usually preferred because here it is possible to dynamically allocate DL/UL resources to support asymmetric UL/DL ratio, and also only one channel, as compared to two, in FDD. Multiple users are allocated data regions within the frame and these allocations are specified using UL/DL MAP messages. The MAP messages include the burst profile which determines the Modulation and Coding scheme that is used in that link. The uplink sub-frame comprises of several uplink bursts from different users and also has a Channel Quality Indicator Channel (CQICH) through which the client can send feedback to the BS.

WiMAX supports a number of modulation and coding schemes like QPSK, 16 QAM and 64 QAM. It also provides the ability to adaptively vary the modulation scheme

based on channel conditions. The BS can vary the modulation scheme based on channel quality feedback and received signal strength for downlink and uplink respectively. The data rate performance of the physical layer varies based on a number of operating parameters. The channel bandwidth and modulation along with the coding scheme used play a major rule in influencing the data rate.

2.6.3: WiMAX MAC Layer

The WiMAX MAC layer acts as an interface between the higher transport layer and the lower physical layer. The MAC layer is responsible for arranging outgoing packets from the upper layer, called Service Data Units (SDUs), and organizes them into Protocol Data Units (PDUs) [5]. WiMAX is designed to provide very high data rates and this is done by providing the flexibility to combine multiple SDUs into a single PDU, to save on MAC overhead when the SDU size is small; or by partitioning a large SDU into multiple PDUs. Multiple PDUs can also be transmitted simultaneously in a single burst to avoid PHY overhead.

The MAC protocol is connection oriented, where each SS that enters the network needs to register with the BS and set up connections which are used for data transfer with the BS. The BS in turn assigns a unique 16 bit Connection Identifier (CID) for each connection. WiMAX also defines the concept of a service flow. This is identified using a Service Flow Identifier (SFID) and is a unidirectional flow of packets with a specified QoS. Every CID is mapped to a service flow and a QoS level. The MAC layer is responsible for assigning air link resources and providing QoS. The BST allocates resources to all the users for both uplink and downlink. The downlink bandwidth is allocated based on incoming traffic and the uplink bandwidth is allocated based on the request that is made by the MS via polling by the BST.

rtPS	real-time polling service
ertPS	enhanced real-time polling service

nrtPS	non real-time polling service
UGS	unsolicited grant service
BE	best effort

Figure 2.5 Scheduling services

The QoS parameters can be identified as scheduling type, traffic priority, maximum and minimum rates, transmission PDU format, SDU type, and size, *etc*. The service flows can be provisioned through the management system or created dynamically via signaling. WiMAX defines five scheduling services, which are:

1. Unsolicited grant service (UGS)

This is designed for fixed packet sizes at a constant bit rate (CBR). An example of an application that can use this service is VoIP without slice suppression.

2. Real time polling services (rtPS)

This service is designed to support real time service flows that generate variable size data packets on a regular basis. MPEG video is an example for this.

3. Non-real-time polling service (nrtPS)

This service is designed to support delay tolerant data streams, such as FTP, that require variable-size data grants at a minimum guaranteed rate.

4. Extended real-time variable rate (ERT-VR) service

This service is designed to support real time applications like VoIp with silence suppression, that need guaranteed data rate and delay, even though they have variable data rates.

5. Best-effort (BE) service

This service is designed to support data streams that do not require minimum service level guarantee, like web browsing.

2.6.4: Introduction to Virtualization

In recent years virtualization has attracted enormous attention. As a result of this trend there are many commercially, as well as publicly available virtualization systems. Dividing the resources of a computer into multiple execution environments is termed as “virtualization”.

These virtual execution environments or *virtual machines* are isolated from each other: it is not possible for the execution of one virtual machine to adversely affect the performance of another .The term virtualization is not well-defined since it refers to the abstraction of resources across many aspects of computing.

Virtualization offers researchers a tool to evaluate communication protocols. With the aid of virtualization, it is possible to create several virtual machines on a single host system. Each virtual machine can run a separate operating system, and hence represent an entire computer system. By coupling several virtual machines over the network, it is possible to create a whole virtual network of virtual machines.

Furthermore, by extending the virtual network via emulation techniques it is possible to emulate a whole wireless network. The most important advantage of virtual environments is that the development of the software can be done on the real machine, tested on the virtual network of virtual machines, and later be installed without any modifications on the real test bed.

KVM is used to create virtual machines on an external machine. This technique allows us to create containers that can be used to access the WiMAX BST. Resources can be allocated to these slices, to allow independent experiments to run simultaneously on the BST. The main advantage of implementing this technique of virtualization is, it supports latest kernel updates.

2.6.5: KVM

KVM is a virtualization technique that uses a kernel module to turn the Linux kernel into a hypervisor [21]. This was the first virtualization technique that is a part of the main Linux kernel. The guest operating systems can run in the user-space of the host Linux kernel with the help of this module. Each guest operating system is a single process of the host operating system (or hypervisor).

A new execution mode, guest mode, is introduced into the kernel by the KVM Module. Plain kernel supports just kernel and user modes. The guest mode of the kernel is supported by the kernel module exporting a device called /dev/kvm. This device provides a VM with its own address space, that is separate from the kernel or any other VM. The devices in the device tree are common to all user space applications but isolation is provided by providing a different map for /dev/kvm, for each process that opens it. The physical memory that is mapped for the guest operating system is actually virtual memory mapped into the process [21]. Translation from guest physical address to host physical address is provided by maintaining a set of shadow pages.

KVM is a part of the virtualization solution. The ability to virtualize the processor for multiple OSs is provided by the processor itself. As discussed earlier, the memory is virtualized by KVM. Modifying and executing a QEMU process on each operating system process enables I/O virtualization. QEMU is a platform virtualization solution that allows virtualization of an entire PC environment. Figure 2.6 provides a view of virtualization with KVM. The bottom-most layer is the hardware that is capable of virtualization. The next layer is provided by the Linux kernel with the KVM module, which acts as a hypervisor. The Linux kernel supports the guest operating system, which is loaded through the kvm utility; while also providing support to run applications just like any other Linux kernel.

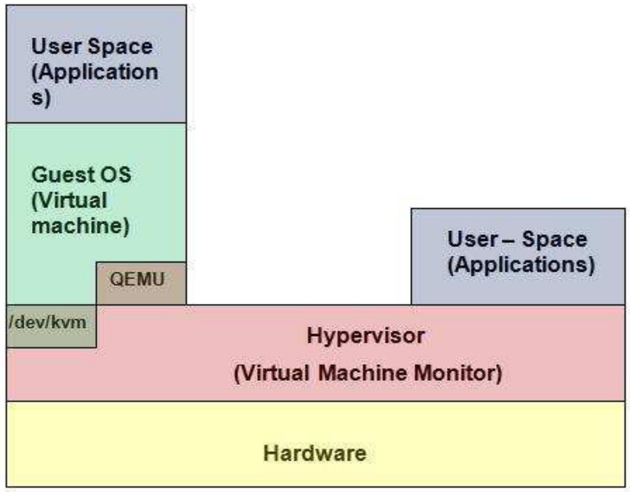


Figure: 2.6 The virtualization components with Kernel Virtual Machine (KVM)

The guest operating system that is running on the top can run any application just like the host operating system. The main advantage of KVM is that it uses the Linux Kernel as the hypervisor and hence any changes to the standard Kernel benefit both the host (hypervisor) and the guest operating system. The drawback of this system involves the need for a user space QEMU process, that can provide I/O virtualization, and also the need for a virtualization capable processor.

CHAPTER 3: SYSTEM ARCHITECTURE AND DESIGN

3.1: Design goals of the API

The 3 principle design goals implemented to virtualize a WiMAX base station are:

Isolation: Virtual networks can have significant coupling, affecting their throughput performance. Furthermore, this unfairness is amplified by auto-rate and scheduling algorithms in the WiMAX base station. This unfairness leads to the effect of moving the base station resources to a mobile client with a low signal to noise ratio. In other words isolation between slices needs to be achieved by decoupling the effects due to each of the slices. VNTS(Virtual network traffic shaping) is an algorithm developed using feedback from the base station's control interface to assess the current channel utilization for each slice.[13] A control parameter in the form of downlink data rate is defined and this value is regulated dynamically for each slice, using an implementation of the Click Modular Router [11] in order to maintain fairness across slices. The VNTS auto-reconfigures itself and imposes fairness at pre-set time intervals.

Abstraction: Aims at providing each slice an illusion of owing the base station and its resources. Abstraction is provided by using virtual machines where each virtual machine is considered to be a slice that can be used by an experimenter. KVM(Kernel virtual machine), discussed in chapter 2 is an efficient virtualization technique which provides an independent linux kernel environment for each slice. Each virtual machine has its own virtual wireless interface. In our architecture, traffic from each virtual machine is sent to a common bridge.

Programmability: To enable easy usage of the virtualized base station it is important to provide the experimenter sufficient degrees of freedom. Programmability aims at providing an API for setting up and controlling experiments. It is also important to have a framework to collect measurements from the experiments. This thesis concentrates on implementing a novel design and implementation of this aspect of design.OMF (cOntrol Management Framework) is a control and measurement framework that is used for the orbit test bed. We have used this as a platform for the development of our API's.

3.2: Design overview

Web Services are used for applications with clients and servers. Grid Services are basically web services with improved characteristics and services that are specific to the application.OMF provides a basic platform to implement a grid service. Since OMF is already being used successfully in the orbit test bed this was a natural design choice. The advantage of using OMF is that it facilitates access of the service hosted from the control network.

Furthermore, the grid service implemented runs independent of the location where the service is run. It can be accessed through a simple web request.

3.3: Grid service

To implement programmability, as a feature to the virtualized base station we would require two major types of functionalities namely, controlling the virtual machines and controlling the radio service of the base station. Controlling the virtual machines essentially entails providing a slice and the resources involved in setting up an experiment. Radio service grid service is required to record the radio resources like

RSSI,MCS and throughput that could be important measurement parameters in an experiment.

The services mentioned above are implemented as separate services namely *wimaxvm* and *wimaxrf* service .These two services are independent of each and hence are implemented as independent services.

3.4: API classification and design of* wimaxvm *service

The APIs are primarily classified as:

User exposed APIs:

- Slice management service
- Radio management service

Internal APIs:

- System Administration API
- SM-Datapath API
- RM-Datapath API
- BSFB-Datapath API

User exposed APIs are those that will be available to slice users for allowing experiment configuration and permitting smooth automated experimentation. These APIs will allow the users of a virtualized system to have a standard interface for common activities while being independent of the technology used underneath. An example could be that irrespective of the virtualization technology used to create virtual machines (KVM, OpenVZ or Xen), all standard API calls will allow the user to work, oblivious of the underlying details. Slice management

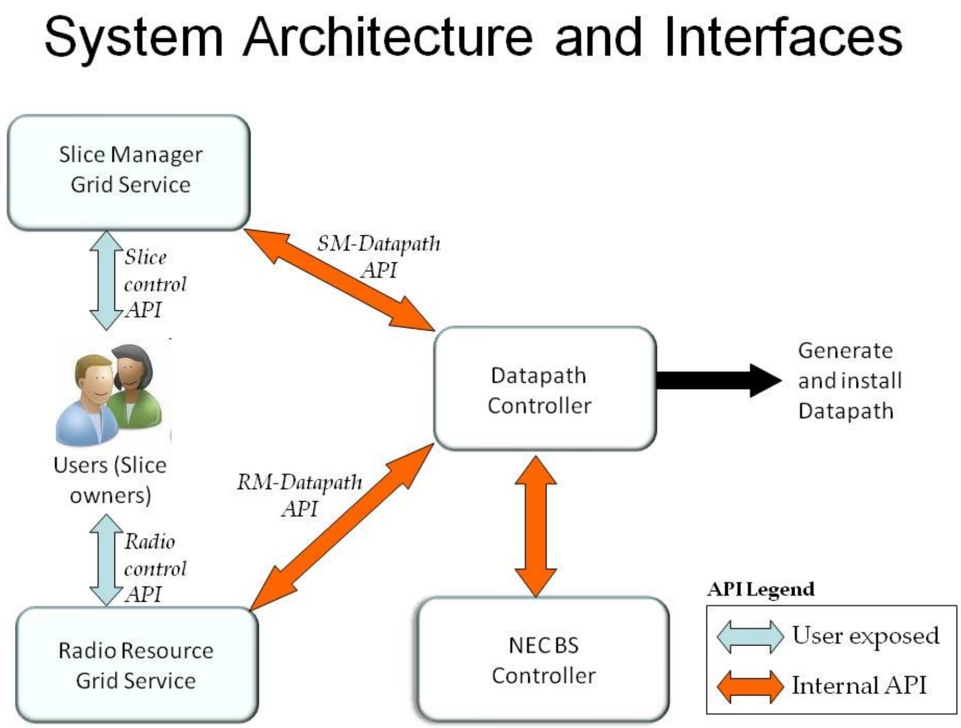


Figure 3.1: System Architecture and interfaces

APIs allow the user to configure slice parameters such as settings of the VM. The Slice manager grid service is further classified into virtual machine control and data path generation components.

Radio Resource Management (RRM) parameters allow the user to monitor radio level information such as per client MCS.

The other set of APIs are classified as internal API. These APIs are useful for understanding the interaction between different components of the system and provide means for appending or changing functionality of the system. The SM-Datapath API describes the API used for communication between the slice manager grid service and the datapath controller. The RM-Datapath controller describes the API used for interaction between Radio resource grid service and the datapath generator. The BSFB-Datapath API is the API used for interaction between the BS Feedback mechanism, which is a part of the customized NEC R6 controller and the datapath manager.

3.5: API design and Implementation

Section 3.4 talks about the API classification and their functionality. We will now discuss the design and implementation aspects of these APIs.

3.6: Slice Manager Grid Service

The slice manager grid service aims at controlling the virtual machines .Some of the functions implemented as a part of this service are described below.

CreateSlice (SliceID):

This is a function used to add a slice (virtual machine). The request is sent to the Slice Manager grid service through a web request. The slice manager in turn creates a slice with the corresponding slice ID and creates a corresponding VLAN .A VLAN is a logical entity - its creation and configuration is done completely in software.

Each slice (vm) is an independent *ubuntu* machine which can be used to host various software services. It can also be used to get various base station parameters like physical rate and RSSI value.

StopSlice (SliceID):

This function is used to stop a Slice once an experimenter is done with his experiment.

This helps in reusing slices once the experiments are complete.

Addclient (SliceID, clientID) :

When a client registers with the base station, it is added to the default slice. After an experimenter creates a slice, clients can be chosen and added to a slice. The Addclient function facilitates the above task. The sliceID and the clientID (which is either client MAC) are the arguments of this function.

Once the Slice is created and the client associates with the Base station, the slice manager service will inform the Slice that the datapath setup is complete. Completion of the datapath setup indicates that there is a L2 link from the slice virtual machine to the mobile client.

The addclient() API has an optional service class parameter that the user can specify. If the user does not specify the service class id, a default service class is allocated that has a single best effort type of service flow. After this sequence is complete, the user can direct traffic to the local virtual interface in the VM with the appropriate MAC address, and traffic will be automatically sent to the correct wireless client.

The Slice Manager service is hosted as a Grid Web Service with the aid of the OMF framework. Web requests can be made to this service to execute each of the functions described above.

A set of web requests and are listed below:

1. Initialize the Slice Manager Web service and create the data structures required:

   wget http://wm-asngw-02:5012/wimaxvm/initvms

2. If an experimenter needs to set up an experiment he starts by adding a slice.This request takes Slice ID as a parameter:

wget http://wm-asngw-02:5012/wimaxvm/start?vmname=vm1

3. List the Slices that are being used.

wget http://wm-asngw-02:5012/wimaxvm/vmlist

4. Stop a slice.

wget http://wm-asngw-02:5012/wimaxvm/stop?vmname=vm1

5. Add a Client to the Slice

wget http://wm-asngw-02:5012/wimaxvm/addclient?vmname=vm1\&clientmac=00:1b:8b:54:02:9a

3.7: Radio Resource Management Grid Service

This API defines the calls that can be made by the system user for controlling radio-related features of the WiMAX system. This API is further classified based on the functionality it provides.

This API will allow the user to create user-defined service flow types apart from the 5 standard service flows supported as part of the current model. An experimenter can use the Radio resource manager grid service to get base station parameters like RSSI values, Physical layer rate, and throughput to each client.

3.8: Basic data path setup

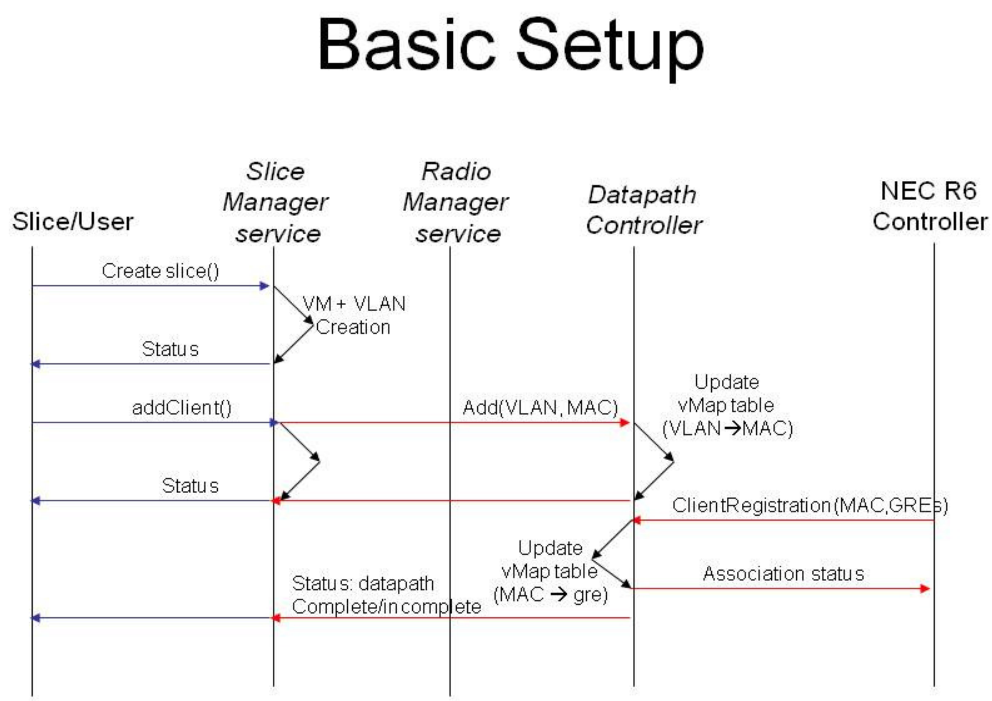


Figure 3.2: Steps to set up an experiment

To set up the basic end to end data path between the slice and its associated clients there are 3 entities that have to synchronize and communicate with each other. When the client associates with the base station the NEC BS controller assigns uplink and downlink GRE tunnel numbers to the client. The GRE tunnel number associated with the clients is used to route the corresponding uplink and downlink traffic to the client. The NEC BS controller communicates this information to the Data path generator entity in the form of a tuple (client MAC address, uplink GRE tunnel ,Downlink GRE tunnel) using TCP sockets.TCP sockets are used as opposed to UDP sockets because an end to end connection is established each time a client associates and is assigned a tunnel number.

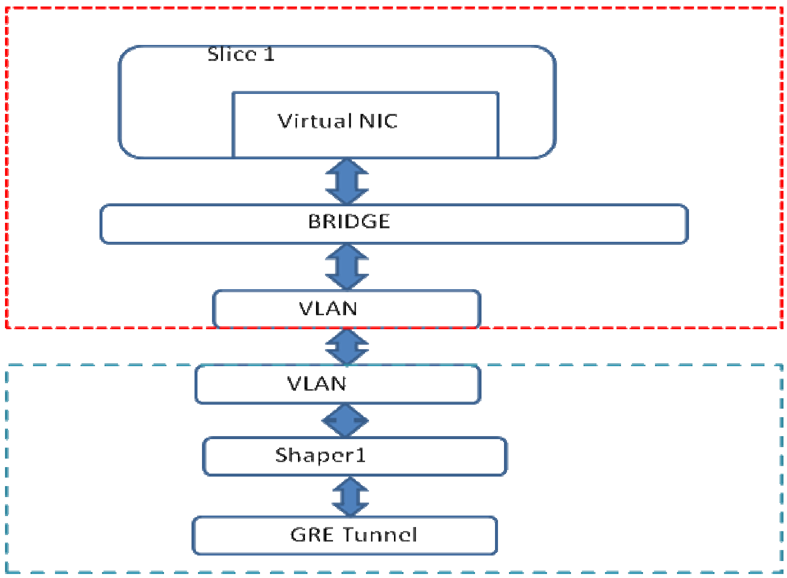


Figure 3.3 : End to end Datapath

The data path generator is implemented using C. It assimilates information from the Slice manager grid service and the NEC BS controller and generates a Click script that creates an end to end route between the client and its respective slice. The steps involved in setting up a data path are listed below.

1. The client sends a request to the NEC Base station controller to associates with the BTS.

Depending on the policy implemented at the BTS the client is either allowed or denied permission to associate.

2. The NEC BTS controller assigns uplink and downlink GRE tunnel numbers to each client.(Client MAC,gre_ul_001,gre_dl_002)
3. The experimenter uses the Slice Manager API and creates a slice to setup an experiment.

CreateSlice(SliceID)

4. Then the experimenter assigns the clients that need to associate with the slice using the function addClient(SliceID,clientMAC). This request sends a message to the datapath generator which creates a VLAN corresponding to each client that is added on the slice. The GRE tunnel numbers corresponding to each client are matched with the VLAN creates and an end to end

data path is setup. Figure 3.4 shows the end to end data connection that is established between the slice and the client.

VLANs are created to provide the segmentation services traditionally provided by routers in LAN configurations. VLANs address issues such as scalability, security, and network management. In our architecture we use VLANS to enable layer 2 routing. Routing based on MAC addresses and their corresponding VLAN id's enables us to implement IP independent protocols on each Slice.

3.9: Automation of Click Script generation:

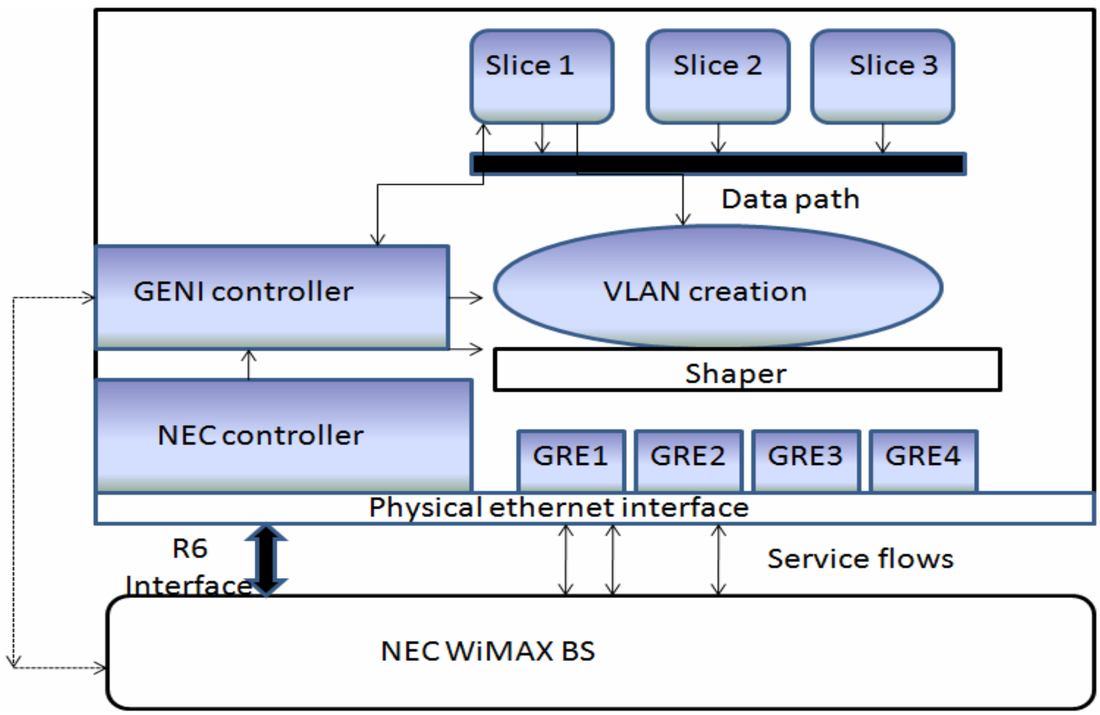


Figure 3.4: Schematic representation of the virtual base stations and its interfaces

One of the key contributions of this thesis is in the design and implementation of the software service that aids in the seamless automation of data path creation. The click scripts that implement the end to end data path connection are generated and executed on the fly, once the web service is launched.

The Datapath generator component communicates with the NEC controller and gets client GRE tunnel information. The GENI controller which contains the grid service provides slice information.

This information is synchronized and VLANs are created and a click script is automatically generated.

In the next section we illustrate some example applications that could use our virtualized WiMAX base station.

CHAPTER 4: SOFTWARE SERVICES AND CASE STUDY RESULTS

4.1: Experiments and their requirement

In this chapter individual experiments which exemplify how our software framework could be used for implementing various experimental scenarios are discussed.

4.2: Case studies

4.2.1: Case A: Limited edge - video service

Experiment Motivation:

Streaming media applications has become an integral part of the internet world. Implementing various congestion control methods is essential to improve the quality of the video stream in case of congestion. Choosing the compression scheme without requiring video-servers to re encode the data, and fitting the resulting stream into the rapidly varying available bandwidth could be one such solution to the problem. Although, rapid fluctuations in video quality should be avoided as it can get annoying for the viewer.

In our evaluation we present a mechanism for improving the video quality of a mobile client by adaptively improving the video quality as the client channel condition improves.

In a typical video stream server and client setup the video server streams the video to an intermediate server where the video is encoded to an appropriate constant bit rate video and then streamed to the client. The encoding bit rate depends on the clients channel quality. We have implemented a few adaptive bit rate matching schemes to illustrate the usage of our architecture.

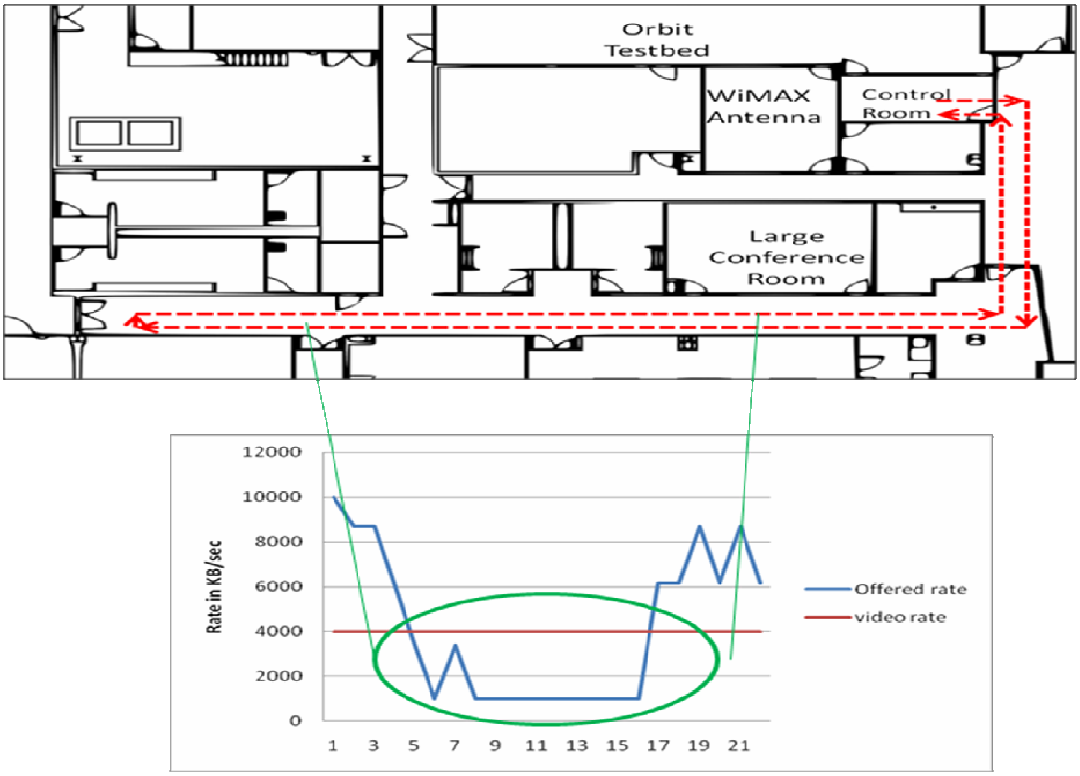


Figure 4.1 : Offered rate and Video rate comparision

To setup a premise for our experiment we plotted the offered rate at the client against time for a mobile client. As shown in figure 4.1, the experimenter walked along a pre defined path in WINLAB. The green region in the figure signifies the region in which channel quality drops drastically. If we stream a constant bit rate video even though the offered rate is much higher as shown in figure 4.1 we do not make use of the bandwidth available. Hence an adaptive rate matching scheme would do well to improve the video quality.

Goal of the Experiment: To illustrate the ease with which an adaptive video rate matching scheme can be implemented using the Software services implemented.

Encoding is a commonly used method, used to match the downlink rate at the client with the constant bit rate of the video that is streamed to the client .As the number of clients in the system increase re encoding the video for each client becomes computationally intensive.

We propose an adaptive video rate matching scheme that clubs clients with similar downlink rates into one Slice and using the addClient() function which facilitates adding clients to different slices , clients are switched between slices .

Experiment setup:

We use the Slice manager grid service to initiate 3 slices and add a client to one of the Slices. Using evalvid[23] ,a tool meant for video evaluation and encoding, a high definition video is encoded to three different bit rates 9Mb/s,4Mb/s and 1Mb/s. PSNR(Peak Signal to noise ratio) is used as an evaluation metric for video evaluation. It is shown that as long as the video content and the codec type are not changed, PSNR is a valid quality measure.[16]

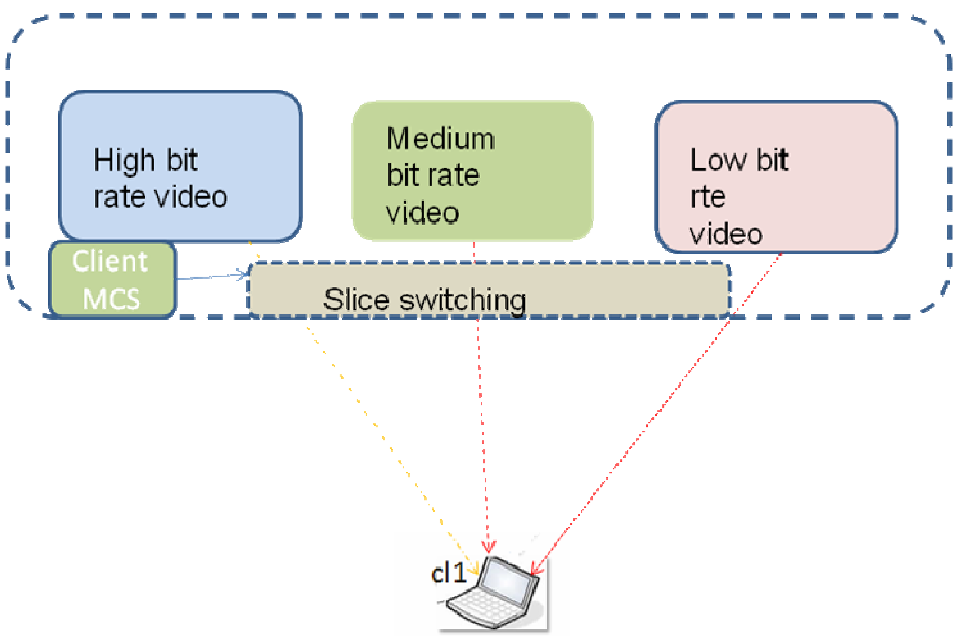


Figure 4.2 : Video Service experiment

Using the Grid service we setup the following Slices:

1. CreateSlice(Slice1),CreateSlice(Slice2) and CreateSlice(Slice3).

2. StartSlice(Slice1), StartSlice(Slice2) and StartSlice(Slice3).

3. Addclient(Slice1,MACaddress).

An assumption is made that all three slices receive multicast packets of the media stream from the stream server. Slice1, Slice2 and Slice3 encode the video for a constant bit rate of 9Mb/s,4Mb/s and 1Mb/s respectively. The bit rate at which the video is encoded is an indicator of the video quality, as the bit rate increases the quality of the video also increases proportionally. If we have a 9Mb/s video being streamed to a mobile client associated with the slice, We notice in figure 4.3, PSNR drops when the client has an offered physical rate of less than 9Mb and the PSNR is high when the physical rate matches or is higher than 9Mb.

We have proposed and implemented an adaptive rate matching scheme which helps maintain the PSNR and in turn the video quality(by maintaining the PSNR range between 30-35 in figure 4.3). We switch the client to the appropriate video slice using the function Addclient(SliceID,Client MAC). We have implemented this switching logic in RUBY. It is observed that the traffic switch happens almost instantly and does not affect the quality of the video.

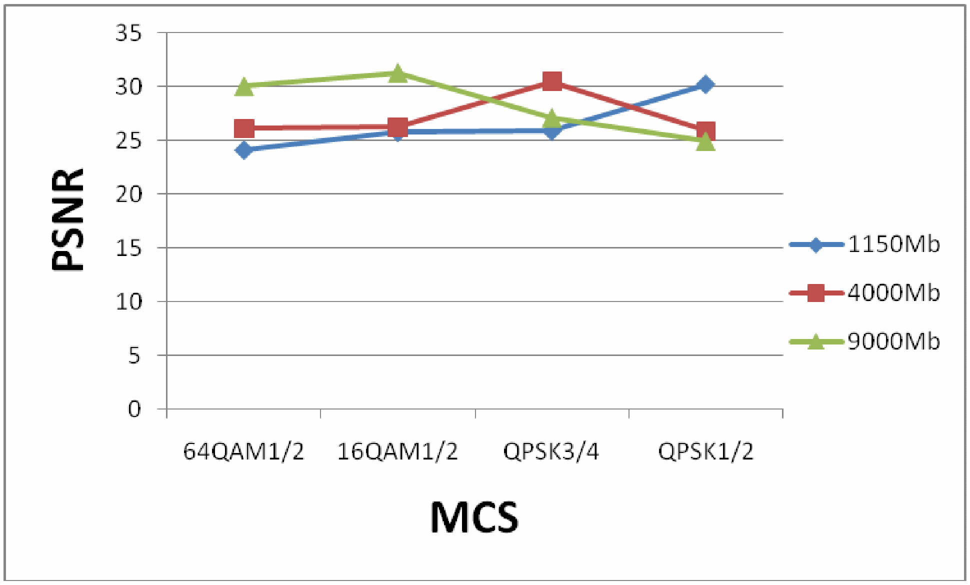


Figure 4.3: PSNR v/s MCS

In figure 4.3 the PSNR values corresponding to the different Modulation and coding schemes are plotted . The general trend is that the PSNR, and in turns the video quality drops as the MCS drops due to deteriorating channel quality.

We notice that videos with different coding schemes have different PSNR peaks. For example the 4000Mb/s video has a peak as between 16QAM1/2 and QPSK3/4. Based on the results of the graph above we can adaptively match the downlink rate at the client with the video rate that we steam.

Adaptive average rate matching scheme:

In this scheme we poll the downlink rate at the client every second and average it over 4 seconds and get the average downlink rate at the client .Depending on the offered rate we can decide which video to stream. If the offered rate is greater than 9Mb/s we stream the video encoded at 9Mb/s. If it drops lower than this then we pick the video encoded at 4Mb/s. We use the Radio resource management grid service to get the rate at each of the clients associated.

The figure 4.4   shows the Average downlink rate against the average rate calculated by the adaptive average coding scheme.

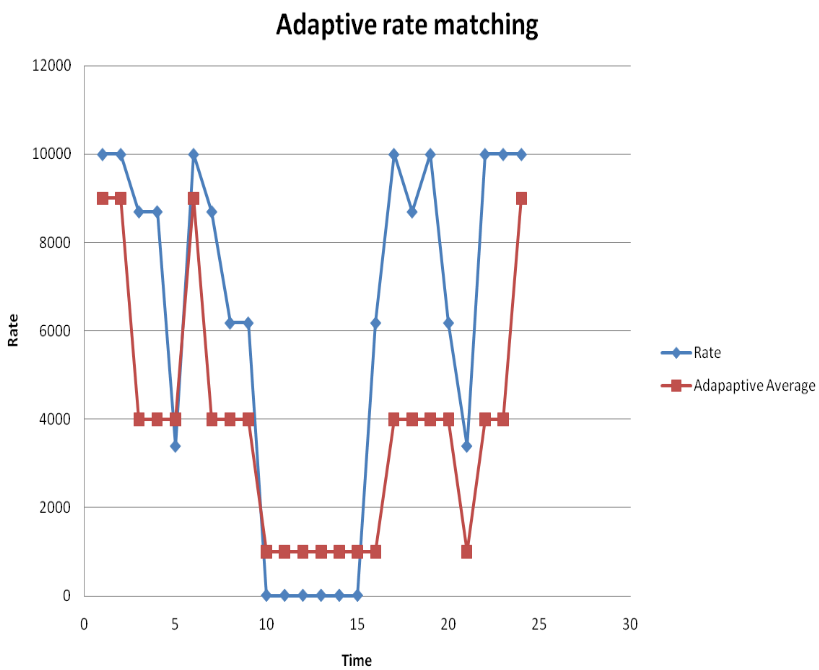


Figure 4.4 Rate v/s time for Adaptive average Scheme

In the figure 4.5 we can see that the PSNR of the 4Mb video drops when the RSSI value is very low. The adaptive matching curve shows that

by matching the appropriate video coding scheme the video quality can be maintained and does not drop below a threshold value.

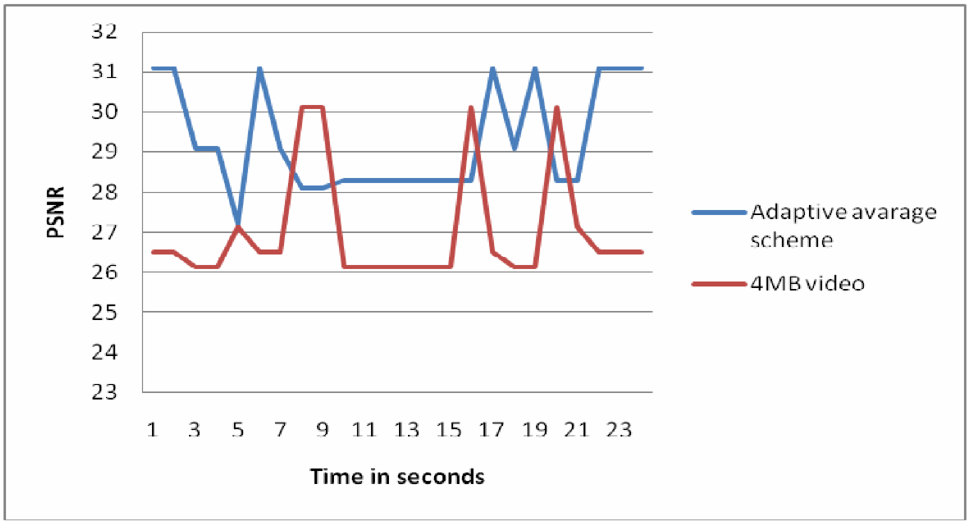


Figure 4.5: PSNR v/s time for Average rate matching scheme

Adaptive Polling rate matching scheme:

Another adaptive video rate matching scheme that was tested was the Adaptive Polling scheme where the base station is queried every 3 seconds and the downlink rate is matched.

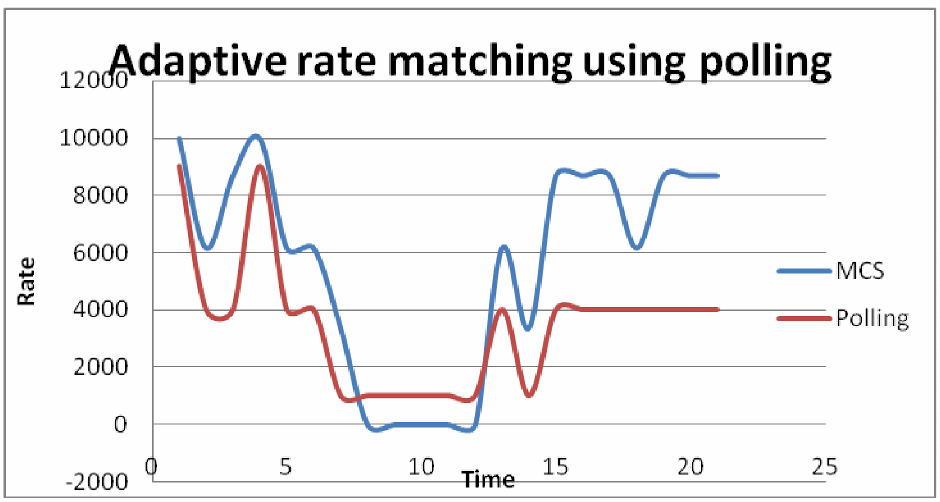


Figure 4.6: Rate v/s time for Adaptive polling scheme

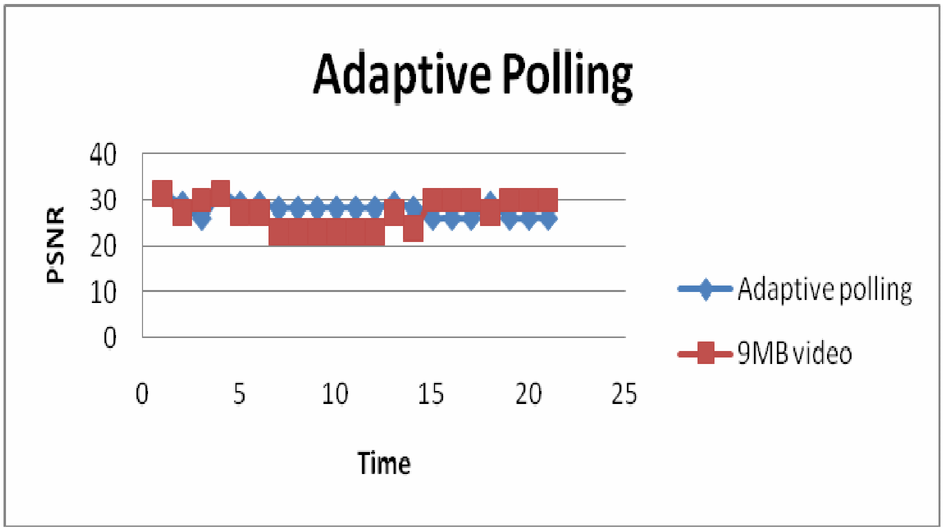
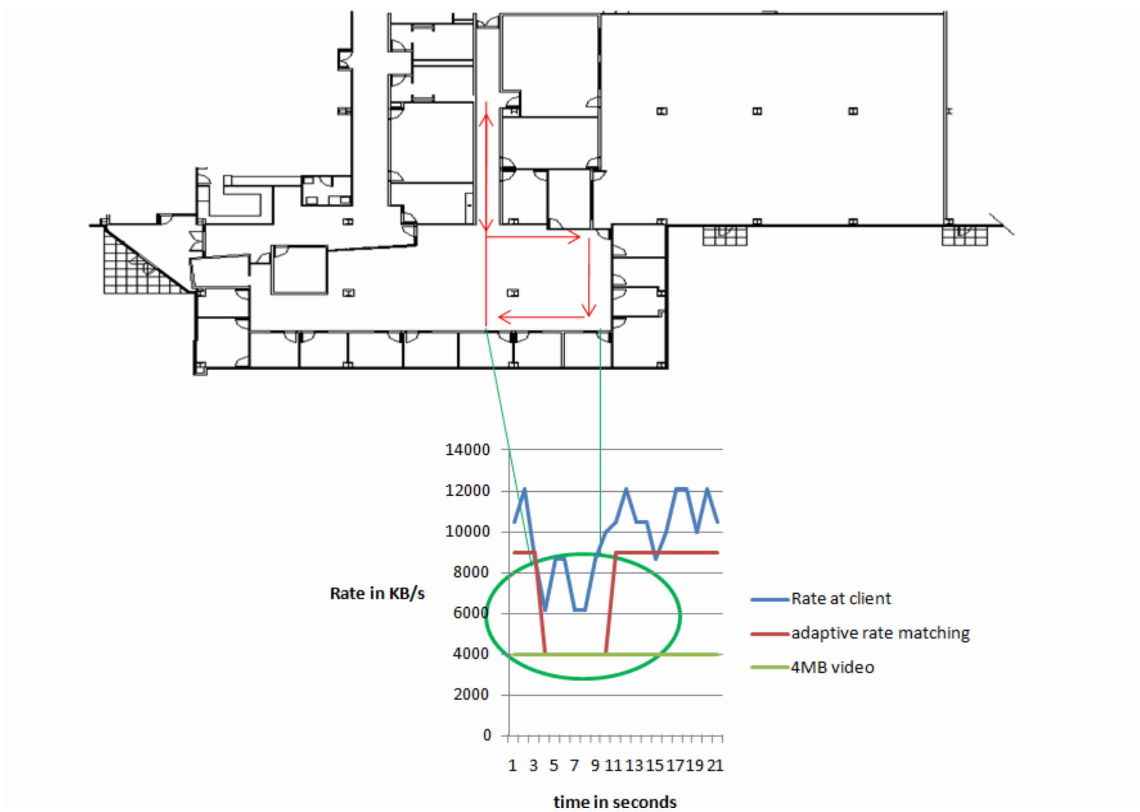
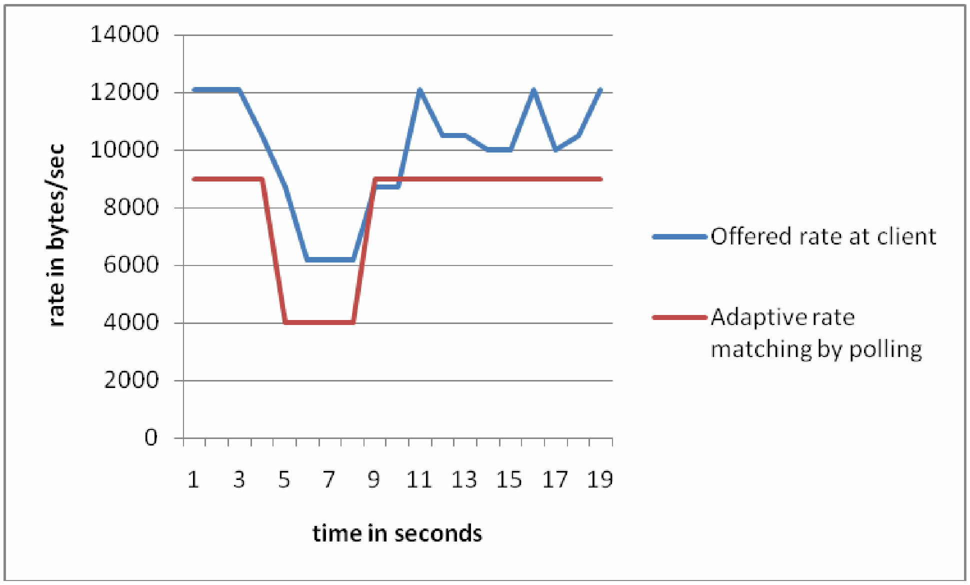


Figure 4.7 :PSNR v/s time for Adaptive Polling scheme

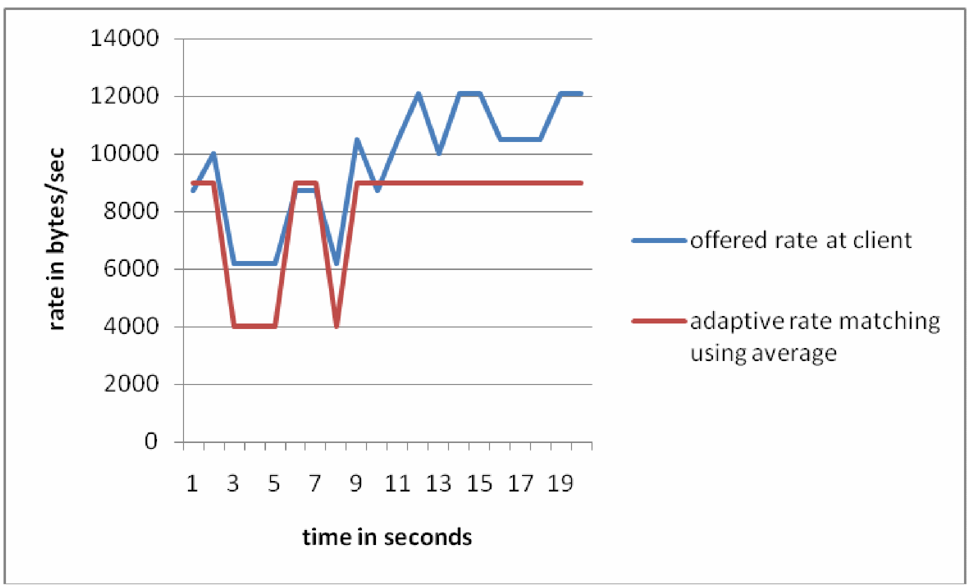
We repeated the set of experiments explained above for a different path to observe if the video rate matching scheme fairs well for a different path.



4.8: WINLAB floor plan path along which the mobile client is moving



4.9: Rate v/s time for Adaptive polling scheme



4.10: Rate v/s time for Adaptive average scheme

4.2: Case B : Base station Handoff Emulation

4.2.1: Motivation

All mobile networks have a handoff mechanism that maintains uninterrupted user communication session during client movement from one location to another. Handoff mechanism handles subscriber station (SS) switching from one Base Station (BS) to another. Different handoff techniques have been developed. In general, they can be broadly divided into soft handoff and hard handoff.

Goal of the experiment: To emulate a mobile handoff mechanism using the Grid service implemented.

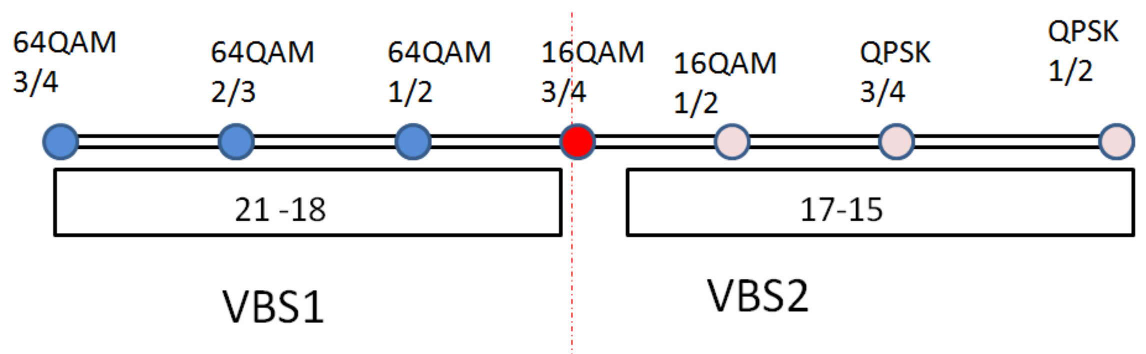


Figure 4.10 .b Range of the WiMAX Base station

Figure 4.10 .b shows the channel indicator and the Modulation and coding scheme that the NEC WiMAX base station uses. To emulate base station handoff we divide the range into two halves. 64QAM ¾ - 16 QAM ½ is considered to be the range of one virtual base station and 16 QAM ½ - QPSK ½ is considered to be the range of another virtual base station. The numbers ranging from 21-15

is an indicator of the channel quality where 21 is the best and 15 signify poor channel quality. The table below gives the rates corresponding to the MCS schemes mentioned above.

Modulation	*Coding*	Rate (kbps)
$64QAM$	5/6	16100
$64QAM$	3/4	14500
$64QAM$	2/3	13800
$64QAM$	1/2	10700
$16QAM$	3/4	10700
$16QAM$	1/2	7180
$QPSK$	3/4	5390
$QPSK$	1/2	3580

Figure 4.10.c: Modulation and coding schemes and its corresponding Rates

The next section explains how the slice manager grid service is used to emulate virtual base stations.

4.2.2: Emulated Mobile client handoff

A mobile handoff scenario is emulated using the Slice manager grid service. We initiate 2 slices VBS1(Virtual Base station) and VBS2 using the createSlice() function. The addClient() function is used to associate a mobile client to VBS1.A switching and signaling logic is implemented in each of the slices. The Virtual Base stations(slices) communicate with each other and implement a handoff mechanism depending on the offered rate at the client.

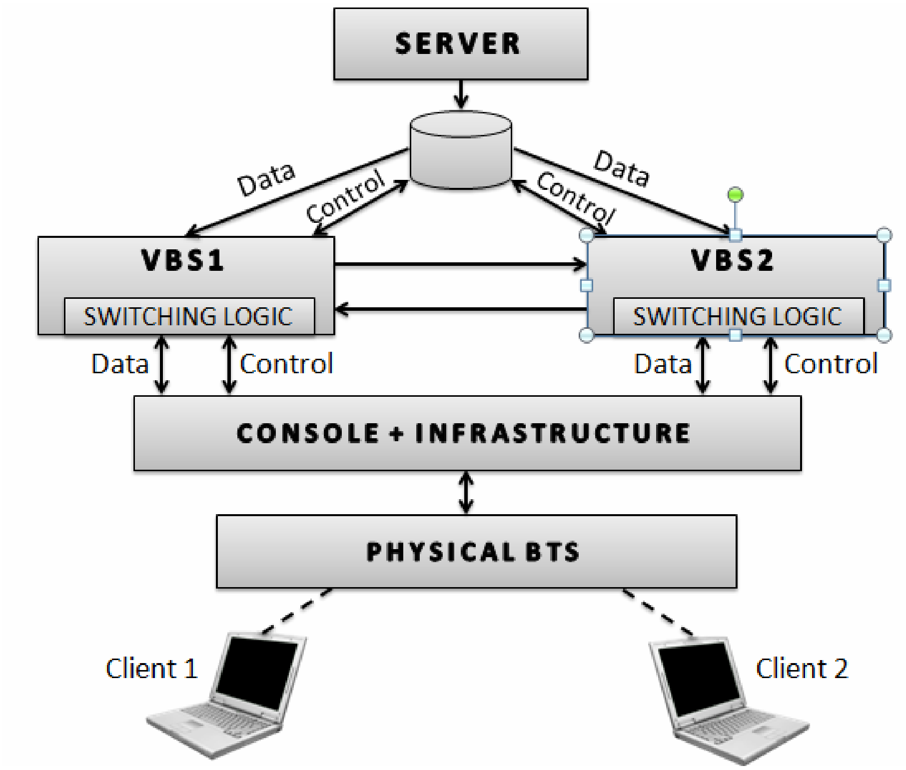


Figure 4.11: Base station handoff schematic representation

In a typical Base station handoff setup we would have a Data server that routes packets to a switch/router which in turn sends control messages to the base stations and routes the packets to the VBS to which the mobile client is currently associated. In our setup we assume that the 2 virtual base stations get multicast packets of the traffic from the server. The virtual base stations send control messages to each other in order to facilitate client handoff. For example when the physical rate to the client on VBS1 drops below a threshold value the base station sends a request to switch (Request1) and once it receives an acknowledgement from VBS2 the client is switched to VBS2.

The switching logic is implemented as a RUBY script.

Experiment setup: We create 2 slices using the createSlice() web request, each representing virtual base stations VBS1 and VBS2. Add a client to BS1 using the Addclient() service. As mentioned earlier 15-21 is a channel indicator where 15 signifies poor quality .When the recorded rate drops below 18.VBS1 hands off to VBS2 when the rate increases to 18 the client is switched back to VBS1.

The formula used to implement the switching logic is

At VBS1	MCS – 18 (If the value drops to 0 switch to VBS2)
At VBS2	18 –MCS (If the value drops to 0 switch to VBS1)

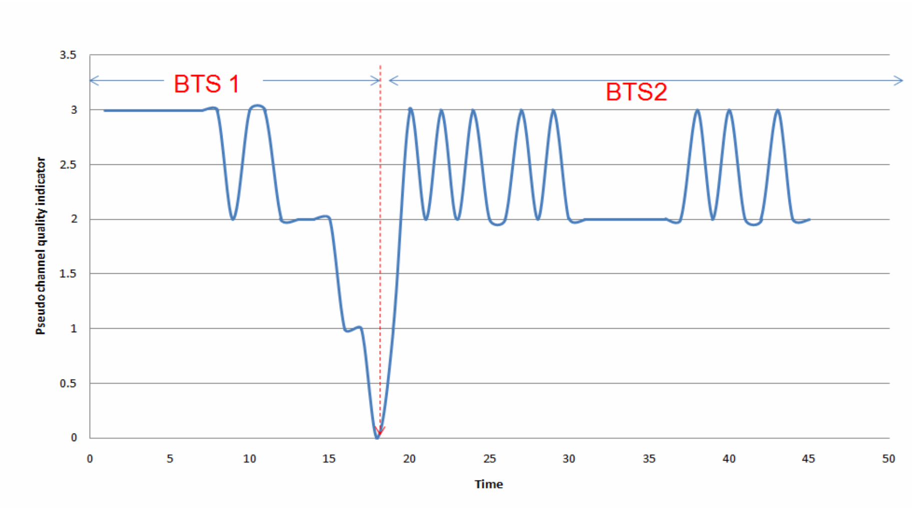


Figure 4.13 :Base station Handoff emulation

Figure 4.13 is a channel quality indicator. The rate recorded at the client is scaled down to range between 0-3, using the formula mentioned above. We see that when the channel indicator drops to 0 the client is handed over to VBS2.

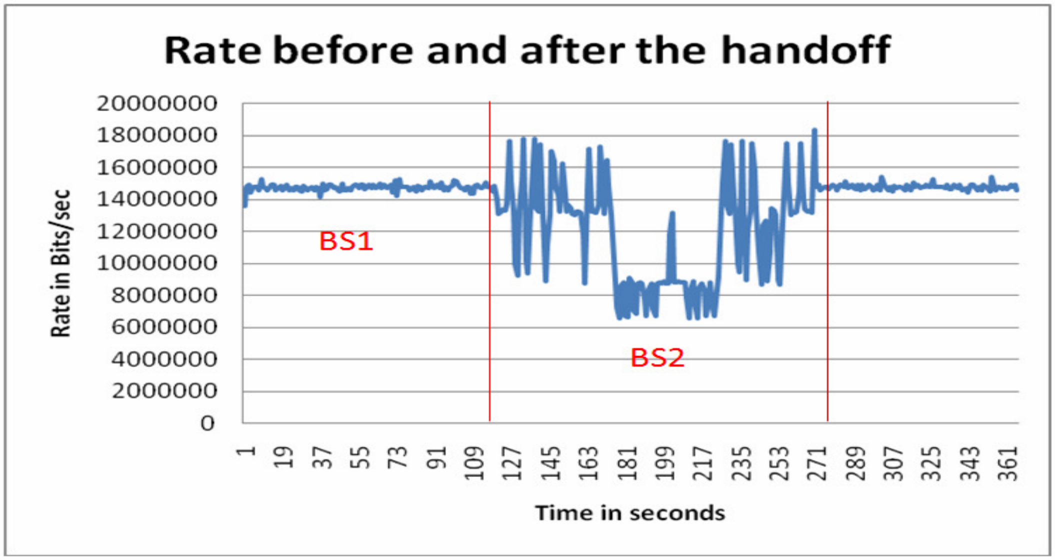


Figure 4.14 : Client throughput before and after handoff

Figure 4.14 shows the throughput at the client before and after handoff from VBS1- VBS2.We see that there is no real improvement in the client throughput since this is a handoff emulation mechanism. Although, the Base station handoff code can be ported to a real base station handoff mechanism without too many changes.

4.2.3: Case C: Isolation of different service types

In this Case study we evaluate the performance of the Isolation mechanism employed in our architecture.

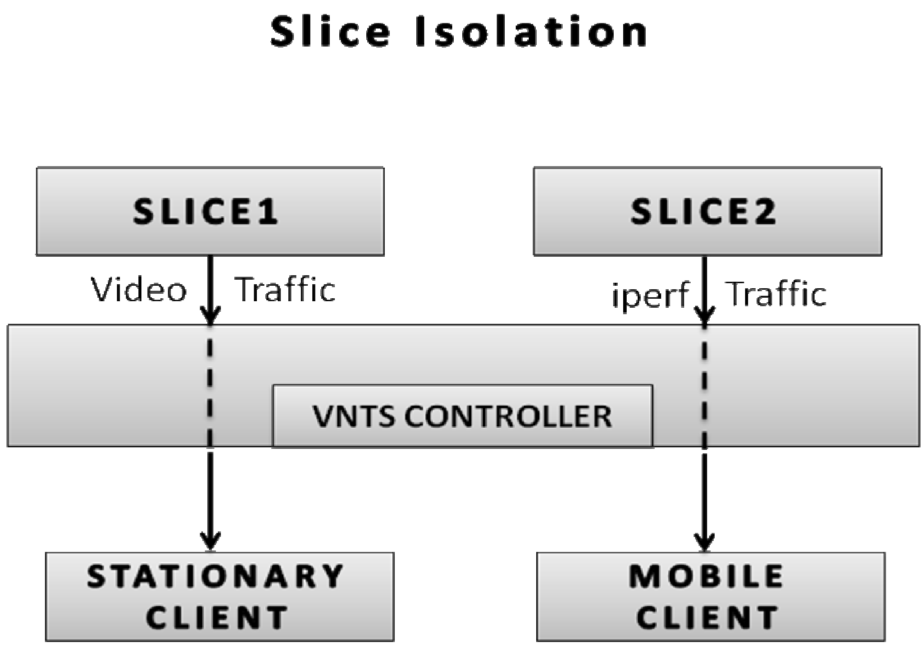


Figure 4.16: Slice Isolation

Results:

Experiment setup: We have 2 slices, one slice has a stationary client associated to it and another has a mobile client associated. When we measure the throughput at the clients we observe that when the mobile client moves to a bad signal region the stationary client's throughput drops drastically due to the coupling effect between the slices.

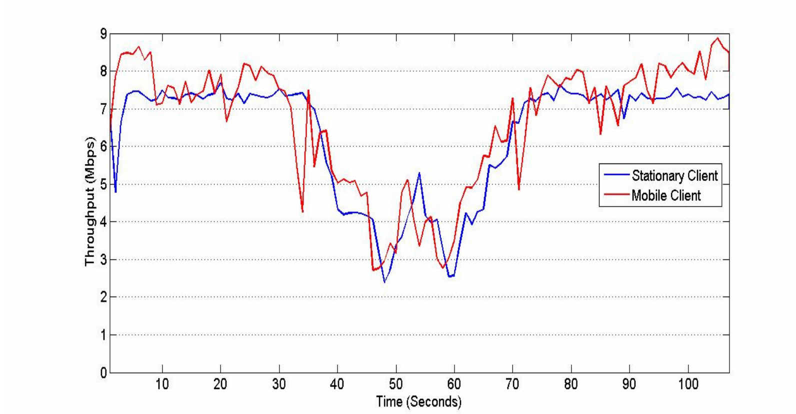


Figure 4.17 Throughput at clients without isolation

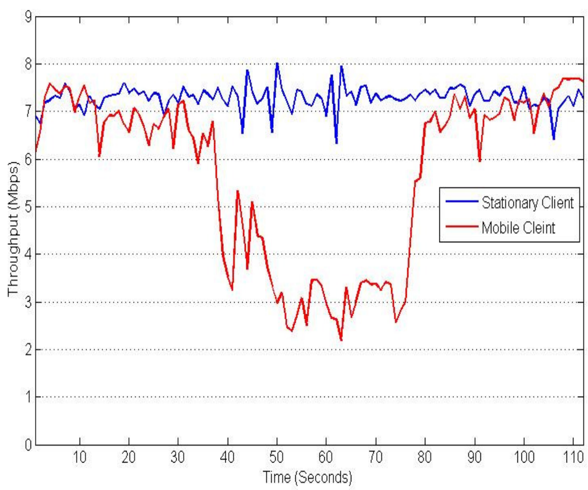


Figure 4.18 : Throughput v/s time with Shaping

With the VNTS(Virtual Network Traffic shaping)[13] isolation mechanics which uses adaptive shaping we observe that the stationary client's throughput remains constant even as the mobile client's throughput drops.

Experiment Goal : To evaluate the Isolation mechanism (VNTS) implemented in the Open WiMAX base station.

Experiment setup: We have two Slices one running an iperf UDP flow to a mobile client and the other slice is streaming a video to a stationary client. We evaluate the current isolation mechanism used in the architecture. VNTS(Virtual network traffic shaping)[13] is an adaptive shaping technique used to ensure fairness in the virtual network setup we use in our architecture. The X axis shows the offered load at the mobile client. The results below show that even as the channel for the mobile client deteriorates, the adaptive scheme is able to appropriately limit the base station utilization by the mobile node's slice, thereby preventing any performance coupling with the static client's slice.

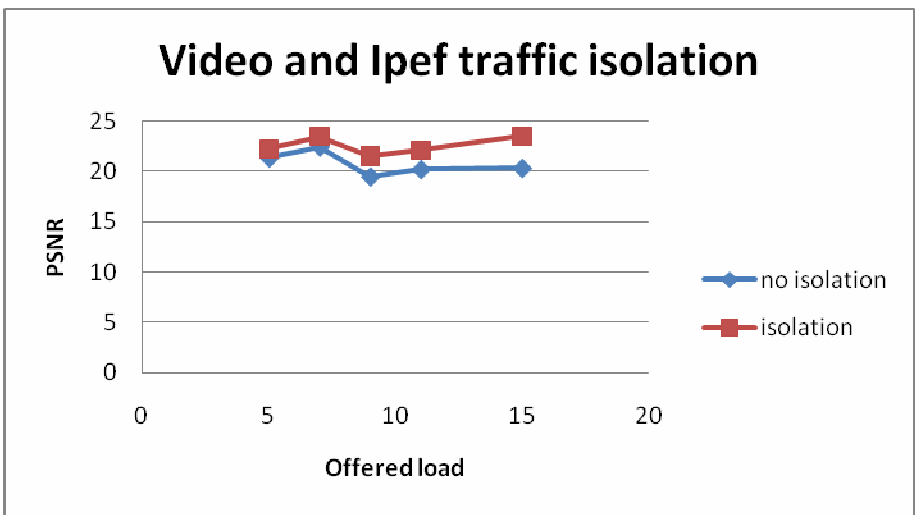


Figure 4.19: Video quality with and without Isolation

Chapter 5: CONCLUSION AND FUTURE WORK

The Open WiMAX Base station is made accessible to the experimenter community by designing and implementing the Slice Manager Grid service and the Radio Resource manager grid service.

The Slice manager service is responsible for Slice creation and addition of clients to each of the slice. The Radio Resource Management grid service is implemented to get and set the virtual base station parameters. A dynamic datapath generator that creates an end to end data path from the slice to its respective clients is implemented using the *Click* software router. Routing is implemented independent of the IP stack to enable operators to implement clean slate architectures.

A video rate adaptation scheme is implemented to illustrate the ease with which an experimenter can implement and evaluate video experiments. A base station handoff emulation experiment uses the Slice manager Grid service and can be used to measure handoff latencies and implement various handoff mechanisms.

Future work would involve improving the current Radio Resource manager service. Inclusion of infrastructure to measure client packet error rate would be useful to the experimenter community. The current Slice Manager Service design has not considered security in its design. Experiments should not be able to set or reset each other's virtual base station parameters. The current implementation of the *Click* router is re installed each time a client associates or dis-associates from the base station. A more efficient way of resetting the *click* router needs to be implementation.

References

[1] GENI. http://www.geni.net/.

[2] Meilong Jang. NEC WiMax Base Station Document.

[3] D. Raychaudhuri, M. Ott, and I. Secker. Orbit radio grid tested for evaluation of next-generation wireless network protocols. In *TRIDENTCOM '05: Proceedings of the First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities*, pages 308–309, Washington, DC, USA, 2005. IEEE Computer Society.

[4] VINI. http://www.vini-veritas.net/.

[5] Ieee standard for local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems amendment 2: Physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1. Technical report, 2006.

[6] Dr. Sassan Ahmadi. Introduction to mobile WiMAX Radio Access Technology: PHY and MAC Architecture. http://www.mat.ucsb.edu/~ggroup/ahmadiUCSB_slides_Dec7.pdf.

[7] Yan Zhang and Hsiao-Hwa Chen. *MOBILE WiMAX - Toward Broadband Wireless Metropolitan Area Networks*. Auerbach Publications, 2007.

[8] Fan Wang, A. Ghosh, C. Sankaran, P. Fleming, F. Hsieh, and S. Benes. Mobile wimax systems: performance and evolution. *Communications Magazine, IEEE*, 46(10):41–49, October 2008.

[9] M. Tim Jones. Discover the Linux Kernel Virtual Machine- Learn the KVM architecture and advantages . http://www.ibm.com/developerworks/linux/library/l-linux-kvm/resources.

[10] Kvm: Kernel-based virtualization driver. White Paper.

[11] Eddie Kohler. *The Click Modular Router*. PhD thesis, Massachusetts Institute of Technology, 2001.

[12] G. Bhanage, R. Mahindra, I. Seskar, D. Raychaudhuri, and Y. Zhang, "Testbed Design For Facilitating Simultaneous WiMAX Experiments", extended abstract in the ACM wireless security symposium (WiSEC), Hoboken, NJ, March, 2010

[13] G. Bhanage, R. Daya, I. Seskar, and D. Raychaudhuri, "VNTS: a virtual network traffic shaper for air time fairness in 802:16e slices," In proceedings of IEEE ICC - Wireless and Mobile Networking Symposium, South Africa, May, 2010

[14] Bo Li, Yang Qin and Chor Ping Low and Choon Lim Gwee, "A Survey on Mobile WiMAX" ,*Communications Magazine, IEEE* In Communications Magazine, IEEE, Vol. 45, No. 12. (2007), pp. 70-75

[15] Reza Rejaie ,Mark Handley and Deborah Estrin," Layered Quality Adaptation for Internet Video Streaming", ACM *SIGCOMM'99*, Cambridge,Massachusetts, USA.

[16] Q. Huynh-Thu and M. Ghanbari," Scope of validity of PSNR in image/video quality assessment", IEEE 19th June 2008 Vol. 44 No. 13

[17] Kostas Pentikousis, Jarno Pinola, Esa Piri, and Frerk Fitzek," An Experimental Investigation of VoIP and Video Streaming over Fixed WiMAX"

[18] Z. Daia, R. Fracchiaa, J. Gosteaub, P. Pellatia and G. Viviera" Vertical handover criteria and algorithm in IEEE 802.11 and 802.16 hybrid networks", Communications, 2008. ICC '08. IEEE International Conference on 19-23 May 2008 .

[19]N.M. Mosharaf Kabir Chowdhury and Raouf Boutaba," A Survey of Network Virtualization", Technical Report

[20] MesutGünes,AlexanderZimmermann,MartinWenig,JanRitzerfeld and UlrichMeis, "From Simulations to Testbeds — Architecture of the Hybrid MCG-Mesh Testbed**",** The publications of the Department of Computer Science of RWTH Aachen

[21] Kvm: Kernel-based virtualization driver. White Paper.

[22] S Hanks, T Li, D Farinacci, P Traina - 2000 - javvin.com

[23]Jirka Klaue, Berthold Rathke, and Adam Wolisz," EvalVid - A Framework for Video Transmission and Quality Evaluation",Adaptability in Heterogeneous Communication Networks with Wireless Access (AKOM)

[24] George C. Hadjichristofi, Avi Brender, Marco Gruteser, Rajesh Mahindra and Ivan Seskar." A Wired-Wireless Testbed Architecture for Network Layer Experimentation Based on ORBIT and VINI", WiNTECH'07, Montreal.

[25] Thierry Rakotoarivelo, Maximilian Ott, Guillaume Jourjon and Ivan Seskar , "OMF: A Control and Management Framework for Networking Test beds",Roads 2009.

[26] Maximilian Ott, Ivan Seskar, Robert Siraccusa, Manpreet Singh,"ORBIT Testbed Software Architecture: Supporting Experiments as a Service",