An SDN-based Framework for QoS-aware Mobile Cloud Computing

Wijaya Dheeshakthi Ekanayake Mudiyanselage

A Thesis submitted to the Faculty of Graduate and Postdoctoral Studies in partial fulfilment of the requirements for the degree of

MASTER OF APPLIED SCIENCE

in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
University of Ottawa
Ottawa, Canada

May 2016

Table of Contents

Table of Contents ii
List of Figures iv
List of Tables v
Abstract vi
Acknowledgement vii
Acronyms viii
1. Introduction 1
1.1 Motivation 1
1.2 Thesis Objectives 4
1.3 Thesis Contribution 6
1.4 Thesis Organization 8
2 Background and Related Work 10
2.1 Overview 10
2.2 Mobile Cloud Computing 10
2.2.1 Proposed Mobile Cloud Computing Approaches 14
2.2.2 Quality of Service (QoS) in Mobile Cloud Computing 26
2.2.3 Support for Mobile Cloud Services at Radio Access Networks 27
2.2.4 Service Mobility in Mobile Clouds 30
2.3 Cloud Computing 31
2.4 Software Defined Networking 38
2.4.1 Traditional Networks and Challenges 38
2.4.2 Software Defined Network Architecture 41
2.4.3 OpenFlow Protocol 43
2.5 Summary 46
3 QoS-aware Regional IaaS Mobile Cloud with Software Defined Networking 47
3.1 Objective 47
3.2 The Importance of the Proximity for Mobile Clouds 47
3.3 Mobile Cloud Services with Regional Datacenters 50
3.4 Mobility Management with SDN 53

3.5 User Mobility across Regions 57
3.6 System Architecture 58
3.7 User Mobility and Datacenter Management OpenFlow Network Controllers 63
3.8 Decision Making Process for Optimum VM Placement 68
3.9 Infrastructure Acquisition Procedure of Regional Mobile Cloud Service 72
3.10 Summary 73
4 Experimental analysis of the Regional Cloud 75
4.1 Objective 75
4.2 Test Environment for Regional Cloud Service 75
4.2.1 Mininet Test Environment Implementation 76
4.2.2 Physical Test Environment Implementation 79
4.3 Designing an OpenFlow Based Wireless Network 80
4.4 Centralized Control Algorithms for the framework 83
4.5 Virtual Machines for Regional Cloud Service 85
4.5.1 Live Migration of Virtual Machines 86
4.5.2 Seamless Service Migration among Datacenters 88
4.5.3 Seamless VM migration Process in Regional Datacenters 89
4.6 Summary 91
5 Performance Evaluation 92
5.1 Objective 92
5.2 Regional Cloud Service Framework Evaluation Test Case 92
5.3 Evaluation Criteria and Results 93
5.4 Summary 100
6 Conclusion and Future Work 102
6.1 Conclusion 102
6.2 Future Work 104
6.2.1 Extending the Cloud Service Optimization Procedure 104
6.2.2 Distributed Network Controller Platform for Mobility Management 105
6.2.3 Integration with Other Mobile Cloud Approaches 106
References 108

List of Figures

Figure 2.1 Two Tier Mobile Cloud Computing Architecture....13
Figure 2.2 A Simplified Illustration of Cloudlets in a Mobile Wireless Network....18
Figure 2.3 Different Possibilities of Mobile Edge Computing Server Placement [17]....22
Figure 2.4 Functions of Edge Cloud Paradigm [40]....24
Figure 2.5 Follow-Me-Cloud Scenario with Distributed Datacenters [51]....28
Figure 2.6 Three Layers of the SDN Architecture....42
Figure 2.7 OpenFlow Switch Architecture and Internals....44
Figure 2.8 Packet Header Fields that are matched against the Switch Flow Table [61]....45
Figure 3.1 Regional Datacenter Serves Mobile Users within a Region....50
Figure 3.2 Regional Datacenter for a Distributed Mobile Cloud Services....52
Figure 3.3 Regional Mobile Cloud Service Architecture....59
Figure 3.4 Regional Mobile Cloud Service Framework....61
Figure 3.5 User Access And Mobility Management Controller....64
Figure 3.6 Datacenter Network Controller Modules....67
Figure 3.7 Cloud Resource Relocation Considerations....70
Figure 4.1 Mininet Emulation Environment with Four Regions....77
Figure 4.2 Flow Path Configuration during User and Service Mobility....84
Figure 4.3 Stages of VM Live Migration Process....87
Figure 5.1 VM Creation Time of Physical and Mininet Test Environments....95
Figure 5.2 Ping Variation of Geographical User Mobility in Mininet....95
Figure 5.3 Seamless User Mobility on Physical and Mininet Environments....96
Figure 5.4 Ping Latency Variation During User Mobility in the Physical Testbed....97
Figure 5.5 Ping Latency Variation (A) Mininet and (B) Physical Test Environment with A Management....98
Figure 5.6 VM Migration Completion Time Variation on Different Application Scenarios....100

List of Tables

Table 2.1 Classification of different cloud providers 35

Table 5.1 VM migration completion time in Physical testbed 99

Abstract

In mobile cloud computing (MCC), rich mobile application data is processed at the cloud infrastructure by reliving resource limited mobile devices from computationally complex tasks. However, due to the ubiquitous and mobility nature, providing time critical rich applications over remote cloud infrastructure is a challenging task for mobile application service providers. Therefore, according to the literature, close proximity placement of cloud services has been identified as a way to achieve lower end-to-end access delay and thereby provide a higher quality of experience (QoE) for rich mobile application users.

However, providing a higher Quality of Service (QoS) with mobility is still a challenge within close proximity clouds. Access delay to a closely placed cloud tends to be increased over time when users move away from the cloud. However, reactive resource relocation mechanism proposed in literature does not provide a comprehensive mechanism to guarantee the QoS and as well as to minimize service provisioning cost for mobile cloud service providers.

As a result, in this thesis' research work, using the benefits of SDN and the data plane programmability with logically centralized controllers, a resource allocation framework was proposed for IaaS mobile clouds with regional datacenters. The user mobility problem was analyzed within SDN-enabled wireless networks and addressed the possible service level agreement violations that could occur with inter-regional mobility. The proposed framework is composed of an optimization algorithm to provide seamless cloud service during user mobility. Further a service provisioning cost minimization criteria was considered during an event of resource allocation and inter-regional user mobility.

Acknowledgement

First of all, I express my gratitude to my supervisor, Professor Ahmed Karmouch, for potentially providing me with guidance, support and motivation throughout my studies to complete this thesis. His valuable technical help and encouragement were a source of inspiration all along the way.

Secondly, I am grateful to Mr. Heli Amarasinghe, who helped with valuable suggestions to strengthen this research topic. The implementation of the various test environments was less challenging due to his technical support during the course of the experimental evaluation.

Finally, I want to dedicate this thesis to all of my family members who have helped me over the past two years, providing love and support during my studies. In particular, I would like to thank my parents who have supported me in numerous ways throughout my educational life to make me a successful person. I am also very appreciative and in debted to my spouse, Thushari, who has provided immense support during my master's program journey.

Acronyms

ACL	Access Control List	MAC	Media Access Control
API	Application-Programming Interface	MEC	Mobile-Edge Computing
ARP	Address Resolution Protocol	OPEX	Operational Expenditure
CPU	Central Processing Unit	OVS	Open Virtual-Switch
DHCP	Dynamic Host Configuration Protocol	PaaS	Platform-as-a-Service
DPID	Data-path ID	P2P	Peer-to-peer
EPS	Evolved Packet Core	QoE	Quality of Experience
FTP	File Transfer Protocol	QoS	Quality of Service
IaaS	Infrastructure-as-a-Service	RAN	Radio Access Network
IP	Internet Protocol	RARP	Reverse Address Resolution Protocol
ISP	Internet Service Provider	SLA	Service-Level Agreement
ICMP	Internet Control Message Protocol	TCP	Transmission Control Protocol
KVM	Kernel-based Virtual Machine	VPN	Virtual Private Network
LAN	Local Area Network	VM	Virtual Machine
L2VPN	Layer 2 Virtual Private Network	WAN	Wide Area Network
LTE	Long-Term Evolution		

1. Introduction

1.1 Motivation

Today, due to advances in mobile computing and communication technologies, network traffic generated by mobile devices occupies a large fraction of the total global Internet traffic [1-2]. Future trends of mobile Internet traffic also show a significant growth in the usage of the number of mobile devices, and advancement in wireless access technologies resulting in an immense growth of the data traffic [3]. It also implies a potential market in the production of more customized mobile services using sophisticated hardware and software features available in smart mobile devices. Therefore, a future aspect of mobile computing will be to enable these devices to run rich mobile applications that in the past were traditionally limited only to static powerful computing devices.

However, the traditional belief related to mobile devices is that they are resource constrained in terms of processing capability, network performance, energy, and storage. Based on the aforementioned predictions, even though next-generation smart mobile devices have been proven to be more powerful, a sufficient energy storage mechanism has not been evolved that will withstand the execution of large computational tasks [4-5]. Hence, it is challenging for mobile application developers to introduce resource consuming application services for mobile users. With regards to general constraints of mobile devices, another challenge is that the heterogeneity of devices, in terms of hardware and software features. This heterogeneity also can become a challenge for application developers to build rich mobile applications.

The devices which are still capable of running such applications suffer from aforementioned energy limitations, resulting in the application becoming deadweight for the battery lifetime.

As a solution, researchers have introduced approaches to fulfill applications required resources outside mobile devices such that, computationally intensive tasks are offloaded and executed in an external infrastructure [5]. The concept of cloud computing is an approach to providing on demand, flexible scalable computing and storage resources for cloud users. Specifically, in mobile computing, the above intrinsic resource limitations are addressed by providing applications required resources at the mobile cloud. Therefore, for devices running cloud-based applications, their processing and/or storage tasks are carried out in the mobile cloud completely or partially to minimize the workload on mobile devices.

When providing cloud based applications for mobile users, the performance depends on several factors, such as QoS support, performance of the cloud infrastructure, scalability, and the cost [6]. However, guaranteeing such application requirement for better performance is a challenging task mainly due to its unpredictable mobility nature. One of the approaches to providing cloud resources for mobile applications is to use remote clouds. However, remote clouds tend to suffer from a highly variable access delay with a risk of packet loss. As a result, hosting time-critical mobile applications data at a remote cloud is ineffective. When the cloud access is done through a multi-hop wide area network, such as over the Internet, cloud service providers have a minimal freedom to optimize the entire end-to-end path between end nodes. As a result, placing cloud resources in close proximity with mobile users is accepted to be more efficient over the remote clouds in mobile cloud computing [7-9]. Other advantages of these close proximity cloud approaches are: location awareness,

geographical distribution, localization of user traffic, along with minimal network cost on both cloud and mobile network service providers.

However, closely placed distributed mobile clouds contain a limited computational resources compared to remote clouds. On the other hand, the relative distance between user locations is a major factor affecting the cloud service operation, and hence requires frequent optimization to maintain closeness in the event of user mobility. One research direction in the context of mobile clouds named edge clouds proposes small-scale datacenters at the edge of the user locations such as cities, suburbs, etc. Placing cloud resources at the edge addresses the required latency, location information, bandwidth requirements and support for mobility [10].

Recently, software defined network (SDN) has become a widely accepted solution in cloud datacenter for network management. Literature has shown that centralization of the entire network control logic as proposed by SDN architecture provides more flexibility and dynamic controllability over traditional networks [11-12]. Similarly, the SDN concept has been used to manage backhaul network traffic to optimize mobile application QoS in mobile access networks using network programmability given by SDN [13-14]. Furthermore, the literature shows the possibilities for cloud resources migration across datacenters using SDN more conveniently over traditional approaches [14-15]. As a result, communication solutions such as L2VPN for inter-datacenter layer-2 information distribution will not be required.

Motivated by the benefits of edge clouds, in this thesis a QoS-aware regional IaaS cloud service for mobile users has been proposed. The dynamic network configurability given by

SDN is used to build the regional datacenter concept to support IaaS cloud operation and user mobility. Furthermore, a control mechanism is introduced to allocate resources for mobile users and monitor user mobility to optimize cloud service based on the user location. Therefore, the QoS-aware regional cloud service framework with SDN addresses the challenges of mobile cloud services and provides service providers a cost efficient solution to manage mobile cloud services.

1.2 Thesis Objectives

The goal of this thesis is to develop a QoS-aware IaaS cloud framework for mobile applications. It includes seamless mobility management with SDN and cloud service provider cost minimization with optimum resource placement. According to the literature, mobile clouds differ from traditional cloud services in many ways. Mobile clouds include different approaches in handling user mobility, maintaining lower access delay and service costs for cloud service and mobile network providers. In this research, the challenges with the cloud and datacenter network management were identified with existing mobile cloud solutions. This approach included addressing key limitations with current approaches and proposed a new IaaS mobile cloud service architecture. It included the following key characteristics:

Close Proximity: In mobile cloud-based services, the proximity is an essential requirement to minimize end-to-end delay between mobile user device and the user service hosted cloud instance [16]. Such proximity requirements introduce new research directions in mobile cloud computing. A well-known placement is addressed in mobile edge computing (MEC) where a small scale computing server is placed at the edge of the mobile network such as in a base station site to host application services [10] [17]. With mobile edge computing,

mobile network providers are able to minimize transmission network load by localizing mobile user traffic within a small region. Therefore, using the benefits of edge computing, the objective was to provide SLA requirements by localizing mobile application traffic within a local region and thereby providing a QoS-aware IaaS cloud service for regional mobile users.

Mobile Cloud IaaS with QoS: During cloud resource provisioning for mobile cloud applications, user SLA requirements must be considered based on the user locations. However, in addition to SLA, service provisioning cost on the cloud service provider must also be considered. As a result, in order to optimally place cloud resources among regional datacenters based on above requirements, an efficient cloud control mechanism is required. For the resource allocation mechanism, [18] was used to allocate cloud resource in datacenters. With the benefits of hardware virtualization in regional datacenters, dedicated cloud resources can be allocated or released for mobile users on demand. Additionally, the geographical mobility was identified as a bottleneck for mobile cloud service which affects the user QoE [19]. As a result, the regional cloud service's optimization algorithm must find a suitable datacenter based on the current location of the mobile user. Therefore, the objective of the proposed regional cloud service controller is to optimally allocate and orchestrate resources during user mobility.

SDN-based Network Management: Complexities in centralized network management and dynamic configurations with legacy networks has led to the SDN playing a major role in datacenter networks [20]. Similarly, mobility management, load balancing, and QoS optimization are few advantages of SDN being utilized in mobile networks. With SDN and network virtualization, automation of cloud services has become much more convenient.

Therefore, in this thesis, the objective is to use the strength of SDN in regional clouds to enable seamless resource relocation based on the optimization. In the case of mobility, the possibilities of handover within and across heterogeneous mobile networks were identified. Literature has introduced several approaches to handle user mobility across both legacy and SDN-based wireless networks [21]. As a result, in this thesis, the user mobility behavior was observed and maintained seamless connectivity with the proposed regional cloud service.

1.3 Thesis Contribution

Based on the characteristics discussed above in section 1.2, a mobile cloud service has to maintain QoS with mobility across mobile networks. According to the literature, mobile cloud approaches with close proximity provide the ability to access local cloud services with higher bandwidth and minimal delay required by cloud applications. In addition, placing cloud services at the mobile network provider edge has become a highly emerging topic among the research community. Considering the benefits of edge clouds, in this thesis's, a regional cloud service framework was introduced for mobile applications. The region concept was initially derived based on the application traffic offload by local gateways at mobile networks. With distributed regional datacenters, a single regional datacenter is capable of serving mobile users from a number of mobile networks within a region. The main benefit of the proposed regional cloud is that it provides on-demand cloud resources for wireless mobile users within a region regardless of their access technologies.

The strength of SDN in mobile wireless networks was investigated for user mobility management. The extension for mobile networks with SDN opens the door for many network-based applications to provide consistent service with user mobility across heterogeneous wireless networks. The focus of this thesis's research was to handle seamless

handover at the IP network of the SDN-enabled wireless networks and make the mobility possible among subnets.

The proposed IaaS mobile cloud service is proposed to be based on the IaaS model such that, computing resource for any application processing can be acquired at the regional cloud datacenter. The proposed QoS-aware IaaS regional cloud extends mobile device capabilities by providing resources in the close proximity cloud.

The geographical user mobility has been identified as a challenge for mobile cloud service providers to maintain cloud service SLA. In near proximity clouds, a relative distance change affects the performance and consistency of the service. As a result, in this research, a cloud service relocation mechanism was proposed based on inter-regional user mobility. The proposed framework in this thesis considers mobile user region and service provider cost to find an optimum regional datacenter for the seamless cloud service relocation. The relocation mechanism leverages the control plane programmability with SDN to update traffic flow packet headers to maintain end-to-end transparency between end-nodes.

A logically centralized control mechanism was introduced to conveniently handle user application requests for IaaS and orchestrate cloud resources with user mobility among regional datacenters. The regional cloud leverages the benefits of physical resource virtualization and network virtualization provided by SDN. Therefore, the cloud control modules are used for on-demand resource creation, network flow level updates and resource migrations across datacenters. As a result, the proposed mobile cloud controller optimizes cloud service reactively without degrading the application performance.

1.4 Thesis Organization

The remainder of this thesis is structured as follows:

In chapter 2, the background and related work information are presented, including an exposure to existing mobile cloud solutions, architectures and challenges while introducing a novel SDN-based mobile cloud. Analysis is given of the cloud computing model in general and focus is on the benefits of decentralizing the cloud minimizing challenges when using the centralized cloud approach. Then the concept of mobile cloud computing is introduced together with the existing mobile cloud approaches and supporting mobile network characteristics. Then, several optimization solutions in mobile clouds are presented with the goal of maintaining QoS with seamless cloud relocation. Finally, an analysis is presented of the challenges in legacy networks regarding cloud computing. In addition, the novel solution called software defined networking and its widely used implementation, OpenFlow protocol are addressed.

In chapter 3, the benefits of close proximity placement for mobile clouds to minimize end to end latency are identified. Then the regional cloud concept for IaaS mobile clouds is introduced. As well, attention is given to the cloud service framework with its physically distributed regional datacenters and a logically centralized cloud controller. The latter is used to facilitate the resource provisioning and cloud service optimization.

In chapter 4, the experimental analysis of the proposed cloud service framework is presented including local and regional mobility scenarios. As well, the functionalities of datacenter and wireless network OpenFlow controllers are given. The theoretical view of the live VM migration process is analyzed, along with an investigation of the issues of geographical VM

migration for distributed clouds. Finally, the preparation procedure for the virtual test environment is stated along with the challenges regarding mobility emulation.

In chapter 5, initially, several scenarios of seamless cloud resource relocation with user mobility are presented. Then, the proposed cloud service framework's performance is evaluated using an emulated testbed and a physical testbed for the above scenarios. The outcomes of the evaluated test scenarios are given based on the simulation results. Also, a comparison is made of the results from the non-optimization scenario to the cloud service optimization during user mobility. Finally, results from both the testbeds are compared in order to predict the scalability of the proposed architecture.

In chapter 6, the conclusion of the proposed cloud service framework is presented with the emphasis on the potential enhancement of the proposed mobile cloud service framework.

2 Background and Related Work

2.1 Overview

In this chapter, the concept of mobile cloud computing is presented and attention is given to how it addresses the intrinsic mobile device limitations and how it supports rich mobile application execution in mobile devices. A general study on the cloud computing concept is given along with the motivation for the distributed cloud architecture. In addition, a description is given of the mobile cloud approaches with different cloud placement methods with the goal of achieving minimal latency in mobile cloud services. A number of bottlenecks with the traditional network architecture are elaborated and, as well as the concept of software-defined networks and the well-known OpenFlow protocol are addressed. Furthermore, several techniques that are mentioned in literature for maintaining QoS with user mobility are described.

2.2 Mobile Cloud Computing

Today mobile devices such as tablets, mobile phones, etc. have become an essential part of our lives. The mobility nature, ability to access rich mobile application services, high performance hardware, software and networks, as well as affordability are some factors that have aided the growth in usage of mobile devices. While providing such rich services to the users, the mobile devices are facing some challenges with their limited physical resources. These challenges are considered to be mobile device inherited, and can be briefly described as below [22].

- Mobile devices have poor resources compared to static devices: Factors such as processor speed, memory size, and disk capacity can be affected by the limited energy, size, weight and other ergonomic features of mobile devices.
- Mobility is inherently hazardous: Portable devices are more vulnerable to losses and damages
- Mobile connectivity is highly variable in performance and reliability: Bandwidth limitations, environmental factors, service providers, coverage and many other factors affect the connectivity
- Mobile devices contain a finite energy source: Although the battery technology is improving, the power consumption is also increasing because of high-performance processors, GPU etc. Therefore, the limited and available energy is a crucial factor to consider.

The concept of mobile computing is a combination of three main entities including hardware, software, and communication [23]. The hardware represents a mobile device such as a smartphone; the software concept is the number of rich applications and services that are running on mobile devices, such as games, photo editor software, augmented reality, wearable computing etc. The communication factor considers the mobile network, its infrastructure, protocols and delivery of data. The applications mentioned above require extensive processing power from the device. However, on the other hand due to the fact that a limited amount of resources is available in the devices, application developers are bound to implement these applications below the device capability threshold. When consideration is given to the trends in mobile devices and the growth of physical hardware specifications, it is likely these limitations will not be solved in near future [24]. At this time, even the

processing capabilities are being increased. However, guaranteeing a sufficient amount of energy to support processing intensive tasks is still a difficult task.

Therefore, the concept of Cloud Computing (CC) provides possibilities to address the limitations in mobile devices by providing required computing resources outside the mobile devices. It allows the users to obtain computing services over the Internet by dynamically providing computing resources as required. The major advantage to cloud computing is that it eliminates the user requirement of future planning of resources, and instead, cloud enterprise can provide service based on the demand with a minimum management effort. Such availability of resources for a mobile computing environment can be called mobile cloud computing [25]. Therefore, Mobile Cloud Computing (MCC) is a replacement technology for heavy mobile computing resources. In MCC, computing offloading operation is carried out in order to migrate (offload) computationally intensive tasks to a resource rich cloud infrastructure. As a result, MCC is able to reduce the amount of processing power, battery life, and storage required for large resource consuming applications, and also it can avoid the challenges that may arise due to the lack of hardware resources within heterogeneous mobile devices.

Figure 2.1 illustrates two-tier architecture for mobile clouds. Tier 1 shows the near proximity clouds provided by local cloud providers. The tier 2 clouds are the distance immobile clouds. The synchronization of data between the tier 1 entities and the tier 2 entities are transparent to the end-user device and being carried out over the Internet. These services are optimized for better performance by placing as close as possible to the users and enabling access through 3G/4G radio access networks or Wi-Fi networks.

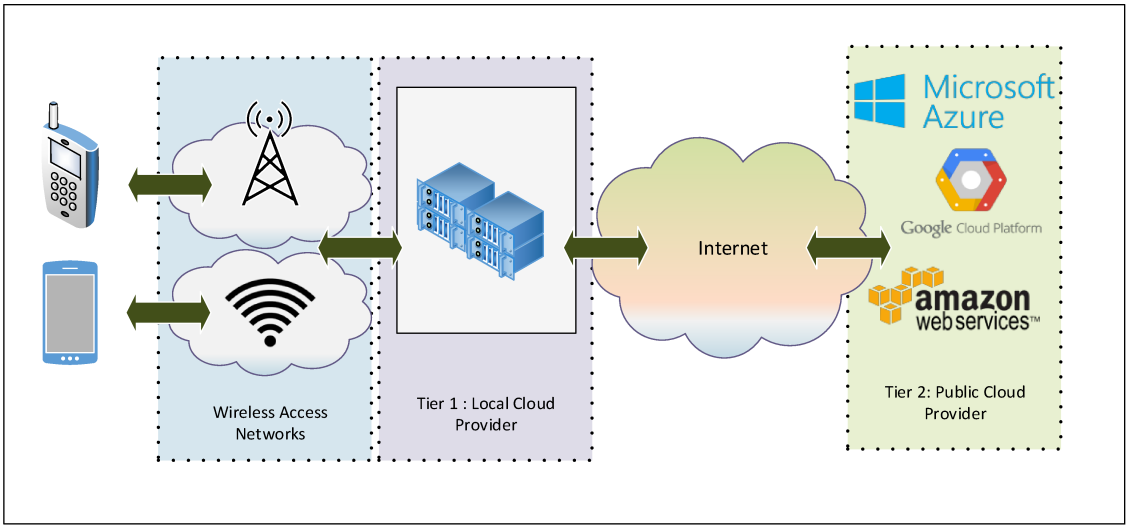


Figure 2.1 Two Tier Mobile Cloud Computing Architecture

When offloading application data into the cloud, several factors are required to be considered, such as: support for mobility, enabling technologies and placement of the execution environment. Successful delivery of these application requirements is also reflected by the user perception or quality of experience (QoE). Therefore, in this thesis, user SLA is referred to the requirements defined by application service provider in order to deliver higher QoE to the end-user. These requirements include a low end-to-end delay from the device to task host and sufficient amount of bandwidth for the application data transfer.

The mobile application experienced latency is highly dependent on the communication path from a device to the cloud (end-to-end delay). Therefore, the minimum length in terms of physical distance and a minimum number of hops in the communication path guarantees the low latency when other QoS parameters in the mobile network such as congestion and queuing delay are assumed to be minimal. However, unlike static users, mobile users are ubiquitous and move unpredictably. The effect of such mobility is the variation of the end-to-end communication path length resulting in the variation of QoE of the end user.

Furthermore, if the cloud is placed in a centralized location, the more the user moves away from it, the more latency is increased, finally leading to QoE degradation at the user device. In addition, the more traffic forwarded to the core of the mobile network, the more costly for the mobile network providers. Therefore, specifically for mobile cloud computing, a comprehensive cloud service placement strategy is required to be identified, in order to meet SLA requirements with respect to mobility, and also for the minimization of mobile core network traffic.

2.2.1 Proposed Mobile Cloud Computing Approaches

In mobile cloud computing, several cloud resource placement methods have been proposed in literature for the optimum delivery of services. Primarily, these approaches are expected to meet latency and bandwidth requirements of the mobile user application while respecting the consistency of the cloud service with user mobility. Many of the recently proposed research solutions support end user mobility with consistent QoE either by distributing or hierarchically placing cloud services from remote clouds to near proximity clouds. Therefore, in mobile cloud computing, several approaches can be considered based on the application requirement, proximity and processing capability.

Remote Public Clouds

Large datacenters located in remote locations hosting cloud services such as IaaS, PaaS to SaaS, contain massive and powerful computing and storage devices. The benefits of the remote cloud services for the cloud service provider are the economies of scale [26] and resource availability. Most of the mobile applications including e-mail and social network services are remotely hosted services and are analogous to remote cloud services.

However, placing cloud in a central location initiates the issue of poor connectivity for the remote mobile users and also a failure along the communication path could disrupt the operation of the entire cloud service. Generally, user applications accessing the cloud service from locations far away from the centralized cloud experience communication path delay and hence are more likely to undergo QoE violations. This is in comparison to the users located near the cloud resource-hosted datacenters. Furthermore, it is highly challenging to reduce the latency in the wide area network [27]; therefore, remote clouds for mobile cloud computing do not provide adequate support for cloud-based rich mobile applications.

Placing any cloud closer to the user also generates a set of questions, including what is the definition of 'closer' in this context. If the closer location is the place where the cloud service can be hosted such that at all times user SLA is met, then the cloud service must be mobile, with the user minimizing the distance change. However, in order to achieve this, there should be a cloud service provisioning architecture that handles consistency with mobility. In several research approaches, an effort is made to bring the cloud resources from a centralized place to the close proximity of the user. This approach minimizes the end-to-end latency and also localizes network traffic within the network. Such approaches can be described as follows:

Mobile Ad-Hoc Clouds

With the rapid growth of computational and storage capabilities of mobile devices, a novel peer-to-peer communication methods were proposed to form distributed computing environment [28] using intermittently available, unreliable infrastructure. The latter was

motivated due to the fact that, most of the organizational and personal IT infrastructure is not fully utilized in 100% of the time [29], and as a result, the idle devices collectively create a large pool of computing resources that can be utilized for mobile computing. Furthermore, short–range wireless communication technologies such as Bluetooth, Wi-Fi direct or device-to-device (D2D) in mobile cellular were identified as a key enabler of ad-hoc computing. And also using aforementioned communication methods, a device is able to become a computational service provider or a client resource user. As a result, a client device can directly offload application data to service provider mobile devices.

In contrast to centralized cloud approaches, ad-hoc cloud computing paradigm contains key characteristics including: volunteering resources, lack of trust, lower interference and diverse workload [30]. According to the ad-hoc cloud architecture proposed by [30], a volunteering set of ad-hoc devices are participated for processing of computational tasks submitted by cloud users over an ad-hoc cloud server. This approach benefits reliable ad-hoc devices available within an organization to form application-specific virtual ad hoc cloudlets. The approach provides a higher utilization of ad-hoc devices, and as well as the general benefits of mobile cloud computing.

In ad-hoc clouds, tasks are partitioned to smaller subtasks and offloaded to other ad-hoc devices at runtime. Then the subtasks are distributed among ad-hoc clients over peer-to-peer connections in a timely fashion based on a scheduling mechanism. However, the challenge with ad-hoc clouds is the lack of mobility support. If the communication session is disrupted, or an offloadee ad-hoc device fails, the subtask being executed in the particular device must be rescheduled and assigned to another ad-hoc client.

As a result, authors of [28] proposed an opportunistic ad-hoc cloudlet architecture integrating the remote cloud for computational task augmentation. The approach uses 3G/4G, Wi-Fi and D2D communication technologies opportunistically, in order to improve the reliability of the cloud offload. Furthermore, a major benefit of this approach is that, users are not required to be in contact during the task execution. When, an offloadee has completed a subtask, the result can be opportunistically uploaded back to the offloader allowing a higher mobility support for the offloader.

However, despite of the advantages including closer proximity, ad-hoc clouds still suffer due to resource limitations. This in turn, hinders the ability of complex mobile application execution within mobile devices. In addition, several challenge of the ad-hoc cloud service can be categorized as follows [30]:

- Heterogeneous computing nodes
- Risk of node availability
- Volatility of ad-hoc nodes

Cloudlets

Cloudlet concept was initially proposed by the authors of [31] with the goal of placing a small scale computing server as closer as possible to the user. The cloudlet was proposed to be located as a middle tier entity between the mobile device and the remote cloud. The goal of placing a proximal cloudlet is to minimize end-to-end latency, and also to increase the cloud service availability. Additionally, mobile device to cloud connectivity over a WAN has been considered as a major communication bottleneck for the provisioning of crisp responsive applications [16] [32]. As a result, when the remote cloud link is not available,

the nearby cloudlets can be easily accessed with the higher availability. The possibility of using virtual machines at the cloudlets for mobile computing is well analyzed in the literature [16] [31], where the usage of virtual machines (VMs) shows the benefits of individual user service isolation. These VMs are created dynamically at the time of a resource request and will be destroyed once the operation is finished. This gives the flexibility of on-demand provisioning of cloud services.

In cloudlets, the use of VMs allows decoupling of service instances from the hardware used by the cloud service providers [33]. This helps cloud providers run user applications regardless the underlying hardware or operating systems. In addition, VMs can be created, cloned, migrated and deleted dynamically; hence services can be moved among cloudlets with minimum disruption to users. Another advantage of using VMs in cloud datacenters is the ability to tailor service instances based on the customer's needs [34], with the required CPU, memory and disk space, etc. As well, many VMs can be placed on a single physical server, which minimizes the cost for the cloud providers.

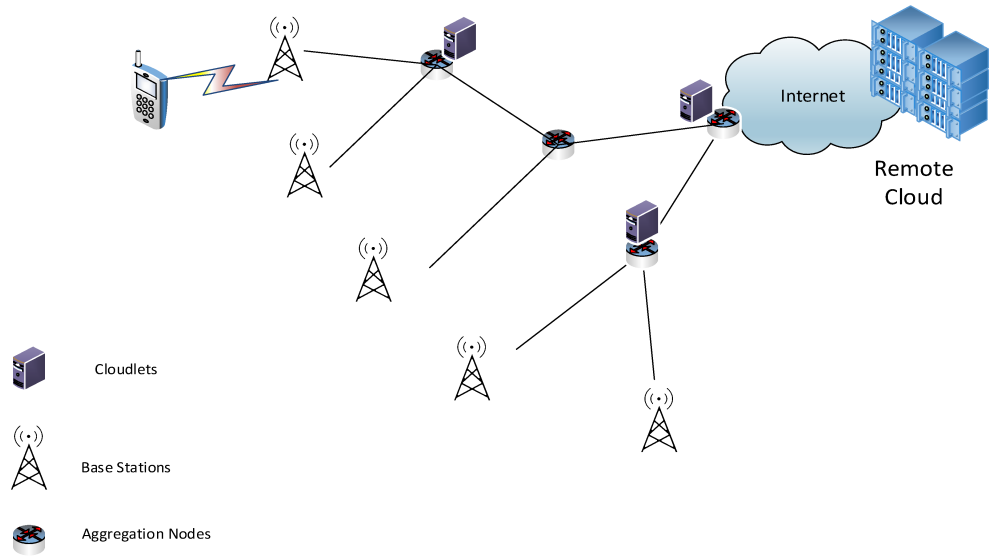


Figure 2.2 A Simplified Illustration of Cloudlets in a Mobile Wireless Network

An approach of cloudlet planning and designing was analyzed by the authors of [33]. It provides an architectural view of the cloudlets in a mobile network as shown in Figure 2.2. According to the authors of [33], cloudlets can be placed in wireless access points or base station sites, aggregation nodes, and in the core of mobile networks. The problem of cloudlet placement arises, when deciding where to place the cloudlet and how to assign users into cloudlets. This problem adds extra complexity with the user mobility and VM orchestration with respect to user mobility. In cloudlets planning, there are three options [33]: VM bulk migration, VM live migration, or a static planning method without any VM migration across cloudlets. Therefore, the size of cloudlets is characterized as small-scale computing resources placed closer to the user.

In the distributed cloudlet architecture, the question arises of why the cloud is still used with cloudlets when considering largely distributed clouds. However, the main advantage to a cloudlet-based architecture is that it provides the ability to push heavy tasks to powerful servers in the central cloud and to download only necessary files to the cloudlet when needed [35]. In the cloudlet architecture, cloudlets require to calculate resources before user request is passed into the cloud to help user mobility [36]. Therefore, the cloudlet which is in closest proximity to a hierarchy of clouds can be used for rich interactive mobile services with augmentation.

Due to closer proximity and lower connection delay cloudlets can be used to provide multimedia applications with higher user experience [37]. Additionally, when comparing central clouds to cloudlets, some research work shows that the cloudlet based approach outperforms the cloud-based approach if the maximum number of cloudlet hops is two or less [38]. If the hops are more than two, it is not always guaranteed that the cloudlet

provides better performance compared to the cloud. However, as a conclusion, the cloudlets only work if the number of hops is less or equal to two in order to outperform cloud-based services.

According to the authors of [9] [10], cloudlets can be accessed only through Wi-Fi networks, and hence, cloudlets are able to cover a small region. Furthermore, authors of [10], mentions that, cloudlets can be connected over a local area network, and are owned and managed by end users. As a result, it does not share the mobile network operator's knowledge. Providing a comparison between edge clouds and cloudlets, it is mentioned that, cloudlets are not committed for ubiquitous computing, and also do not contain sufficient computing resources. Therefore, according to the authors, in order to address the above limitations in the cloudlets, Mobile Edge Computing (MEC) concept was proposed.

However, authors of [39], describes that, user experience can be improved by placing computing resources closer to mobile users with low end-to-end latency. In their research, authors refer several closely placed mobile cloud approaches such as: cloudlets, micro datacenters, fog, and mobile edge clouds. Subsequently, the term 'cloudlet' was used to represent above scenarios in Wi-Fi or mobile cellular networks. As a result, it can also be concluded that, the cloudlets is not yet a standardized architecture, and hence, can be applied to a closely placed small scale cloud for mobile computing.

Mobile Edge Clouds (MEC)

Considering the benefits of proximity and distribution of remote computing services, it was proposed that, mobile network edge can be used for mobile computing. As a result, Mobile Edge Computing [9-10] [17] [40-41] introduces a small scale computing server to be placed at the radio access networks (RAN). According to literature, in most of the MEC-based

approaches, the MEC servers are directly attached to the cellular base station. These MEC servers provide flexibility for application developers and content providers to deliver latency critical rich mobile applications to the end users. Furthermore, the content delivery can be done more efficiently when the MEC servers are caching content from a remote server, and also providing users within the proximity.

On the other hand, the growth of Internet-based mobile applications appeared to be stressing the mobile network operator core due to accessing remotely hosted data [9]. As a result, network operators are required to upgrade and enhance the capacity of the network core in order to meet the continuously increasing demands. Therefore, another aspect of MEC is to move services to the network edge and thereby reduce network traffic at the mobile network core.

ETSI (European Telecommunication Standards Institute) has standardized the requirements and possible architectural placements of MEC servers within mobile cellular networks. According to the ETSI's white paper, the benefits of MEC can be characterized in the following list [17]:

- Locally accessible computing resources
- Close proximity to users
- Lower latency giving shorter response time thus enhancing user experience
- Knowledge about user location allowing many applications to customize location-based services
- Real-time network information

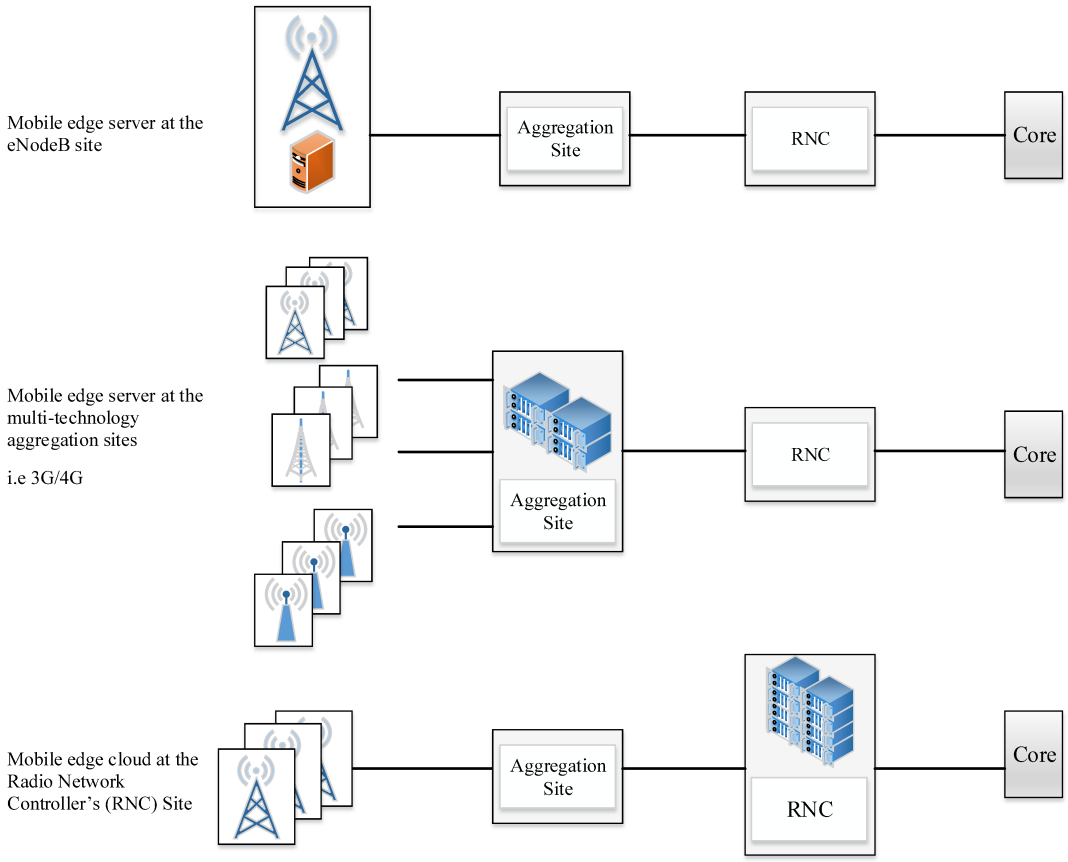


Figure 2.3 Different Possibilities of Mobile Edge Computing Server Placement [17]

Figure 2.3 shows the different possibilities of placement of MEC server in the mobile network edge, such as in the eNodeB site of the LTE based mobile cellular network. However, it appears to be important to find the best approach for placement of the cloud server due to the user mobility behavior and the type of application. Considering the research works regarding mobile edge clouds, it can conclude that any cloud-based resource placed in MEC server can be migrated dynamically with the geographical user mobility within a mobile cellular network [53].

According to the authors in [9], the MEC ecosystem is composed of four major stakeholders, namely mobile end users, network operators, Internet infrastructure providers, and application service providers. Mobile network operators own and manage base stations,

cellular network links and MEC servers while Internet infrastructure providers manage Internet access and routing. The application service providers can host application services on MEC servers for the fast and optimum delivery of services. Additionally, MEC servers located in base stations has the direct access user traffic, as well as radio network information. As a result, MEC servers are capable of processing specific user requests at the edge of the network without any involvement of the core.

However, authors of [9] mention the two application types that can be implemented on MEC servers based on the traffic flow: transparent/pass-through and terminating application traffic. The transparent/pass-through application traffic from mobile devices can be monitored and rerouted to the core network as required. However, in the case of terminating application traffic, they are encapsulated with an IP header and are forwarded to the VMs hosted in the MEC server. If the current MEC server does not host an application-specific VM, then the packet flow is forwarded to a MEC server which hosts such a service VM. Consequently, the mobile applications can be transparently served with optimal performance by placing application services at the MEC.

In addition to the MEC approach, authors of [45] proposed a mobile edge cloud platform with small scale datacenter closer to user equipement. The approach is to connect or to co-locate a small scale datacenter at an eNodeB site. The latter is named eNodeB cloud (eNBC). Placing an eNBC at the mobile network edge reduces the amount of network traffic at mobile anchor points. Similarly, the approach avoids processing of the packet flow resulted by a number of network elements, and hence it assures the optimum delivery of the service with minimal delay. The user mobility scenario is addressed, referring to seamless

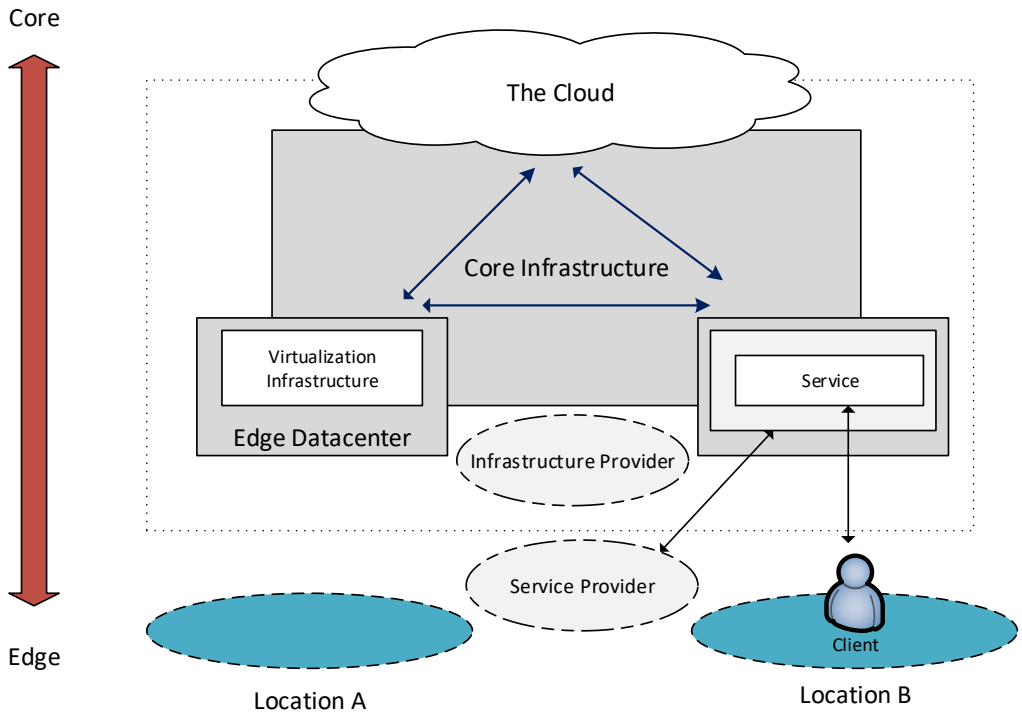


Figure 2.4 Functions of Edge Cloud Paradigm [40]

handover across eNodeB sites along with transparent user equipment context migration among eNBCs.

According to the authors of [42], generally, the edge computing paradigm originated from the P2P concept and expanded, resulting in decentralization no longer being the core objective. These edge datacenters, because of their placement, come with several characteristics that include: the proximity, intelligence, trust, control and the human users at the edge. There are two cloud approaches involving the edge clouds namely edge-only computing and hybrid computing (with an edge and centralized datacenters) [43]. According to the literature, when comparing outcomes of placing processing functions at edge datacenters for interactive gaming applications, the edge-only architecture with distributed clouds performs better than the remote cloud service. The negative factor with edge cloud is

that it demands higher maintenance and initial deployment cost. Therefore, hybrid clouds provide higher availability for on-demand latency sensitive applications for many users.

In one of the edge cloud-based architecture's named "Spaceify" [46], client applications are processed at the edge node that acts as a resource-rich Internet gateway for the devices. As a result, in this client-edge-server model, users directly connect to the edge clouds covering the user location. In the Spaceify concept, the authors partition physical space such as a university, bus station, etc. into blocks and introduce localized web services based on the space. In this way, authors expect to enhance user experience and interactive-ness of mobile users within the partitioned physical space.

Figure 2.4 illustrates the virtualized infrastructure at the edge datacenter which is operated by the infrastructure provider for a multi-tenant environment as proposed by [40]. The advantage of this edge cloud platform is that any third party, end user or IaaS provider itself can operate virtual resources and also can work autonomously or collaboratively with other edge clouds. In this architecture, there is a possibility of hierarchical placement of cloud resources from edge clouds to remote clouds. Therefore, multiple IaaS providers can create an open system where users can be observed regardless of the mobile operator network.

Consequently, the mobile edge cloud concept enables application developers to provide rich mobile services using close proximity computing resources at the network edge. At the same time, mobile network operators can benefit from the services placement at the edge such that it can balance the network traffic at the backhaul network and reduce overall traffic at the network core. Similarly, cloud providers are able to reduce workload at the remote cloud infrastructure and minimize network utilization costs.

2.2.2 Quality of Service (QoS) in Mobile Cloud Computing

In early, 1990s, the QoS concept was emerged for IP networks with three technological approaches, namely: traffic shaping, differentiated services (DiffServ), and reservation of resources using signalling mechanisms [47]. These approaches were proposed to classify different traffic flows within a network, and also to manage based on specific requirements or policies. At present, Quality of Service (QoS) has become an important element of day-to-day computer networking applications. Due to the effect of today's growth of multimedia applications and data, the network capabilities have been dreadfully stretched to serve high-quality video and delay-sensitive applications. However, this effect is complemented with organizational business outcomes and hence, network services must be provided with security, predictability, measurability and also guaranteed availability. As a result, higher business values can be achieved by network parameters such as: communication delay, bandwidth, delay variation (jitter), also and packet loss [48]. Thus, QoS is an important element of a network service that involves techniques to manage the aforementioned parameters on network resources.

Proximal placement of cloud services also a QoS improvement strategy for mobile applications and thereby increase user QoE. In QoE measurements, mean opinion score [44] value models can be used to compare the absolute benefit of close proximity placement. In the above scenario, QoS can be improved by minimizing end-to-end delay [6-7]. As result, the end user QoE can be improved [9] [14]. Current proposals on MCC are mainly focused on the improvement of QoS, and thereby provide rich application services on resource-limited mobile devices. Execution of rich applications requires computing offloading solutions with a tight latency and/or higher bandwidth requirements.

The QoS metrics for mobile cloud computing can be grouped in to several major aspects, such as: connectivity, reliability, capacity, and delay [49]. According to the authors of [45], deploying cloud services within the edge reduces the number of network elements that the user traffic is required to traverse. As a result, optimal delivery of services is ensured, as well as improvement in QoS. Therefore, the distance between mobile users to cloud is an imperative parameters required to be considered in mobile cloud computing.

2.2.3 Support for Mobile Cloud Services at Radio Access Networks

Closer placement of centralized cloud resources is considered as the distribution of computing resources. Each distributed cloud resource element is placed in a closer proximity to meet user and network provider requirements. Such distribution of cloud resources has become a highly emerging topic among the research community these days. In addition, with traditional networks that handle the entire mobile network traffic hinders the performance in several ways [19]. This includes overloading the central location due to traffic backhauling, unnecessary processing in the middle boxes of the network path between the mobile device and the serving server, and the long communication path that adds unnecessary latency to remote mobile user. This delay due to communication path length degrades user QoE in high-quality mobile application services. The concept of distribution of network gateways with selected IP traffic offload (SIPTO) [50] provides an approach to offload selected network traffic at the network edge, resulting in minimizing the backhauling in the mobile operator network. This approach gives the flexibility of providing latency critical cloud services at the edge of the LTE-based radio access networks.

The idea of optimum cloud service provisioning comes with the service provider requirement while meeting user QoE. The primary requirement of the user is reflected by the

SLA criteria. The physical distance between the client and the server that defines the latency due to propagation delay and processing time at intermediate hops, are implicitly unavoidable factors. When further QoS control mechanisms are not defined, the only way to improve user QoE is by placing the server as close as possible to the client [14].

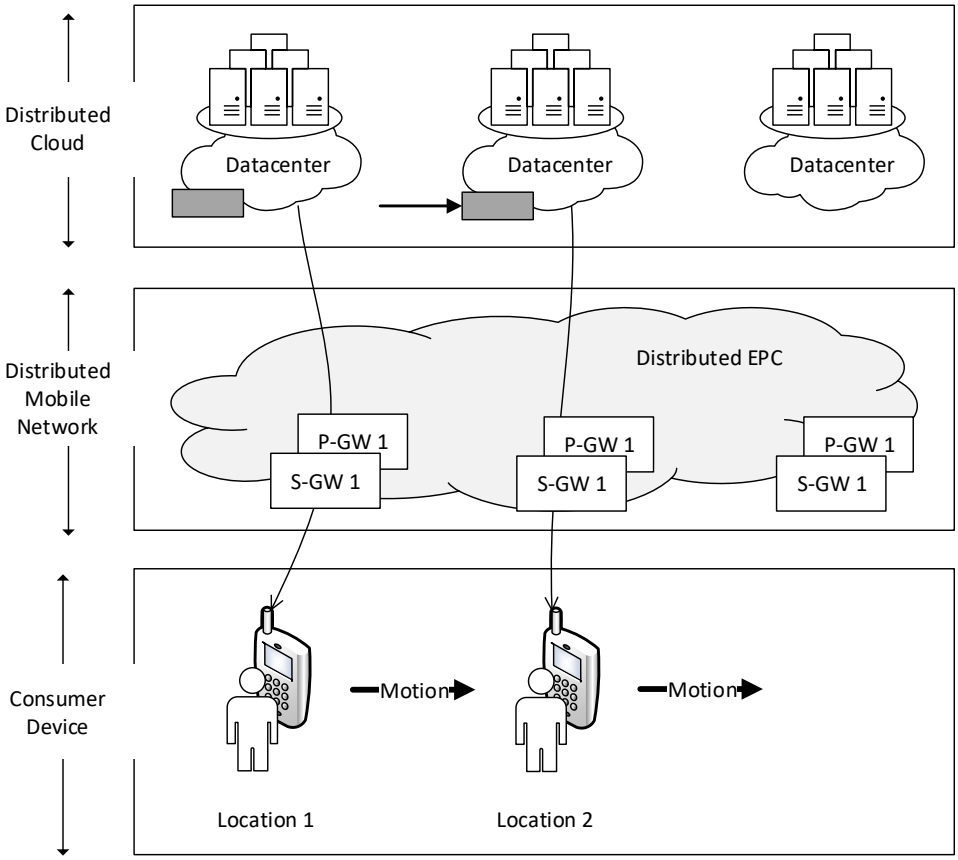


Figure 2.5 Follow-Me-Cloud Scenario with Distributed Datacenters [51]

This challenge is unavoidable in centralized mobile networks with user mobility. As a result of the variable communication delay, a dynamic resource orchestration strategy called 'Follow-Me-Cloud' (FMC) [51] is proposed for EPS where the moving client is expected to be followed by the cloud service. This idea can be realized by the distributed gateways with mobile networks [19] and also distributed datacenters near the edge of the mobile network.

The goal of this approach is to minimize the physical distance from the client to the server and also maximize the bandwidth. As shown in Figure 2.5, the mobility of the user across a physically distributed mobile network incurs the cloud service to be followed across distributed clouds to maintain the user QoE within an EPS.

In order to minimize network traffic at mobile anchor points of a mobile cellular network, authors of [45] porpose a mobility supported edge computing apprach. In this approach, mobile services are hosted in a small scale datacentre that is proposed to be collocated or attached to an eNodeB site. The latter is named eNodeB cloud (eNBC). Placing cloud services at the eNodeB site reduces the network elements a user traffic has to traverse, and hence, improves QoS. Furthermore, attention is given to user mobility scenario, and as a result, authors propose a seamless user handover approach across eNodeB sites together with transparent user context migration among eNBCs.

In edge cloud services, during a handover, continuous connectivity between end nodes is an important aspect that requires to be considered in order to maintain QoS [45]. The main goal of IEEE 802.1 standard is to allow handover between heterogeneous wireless access network technologies, with or without disrupting ongoing communication sessions. In [52], referring to the IEEE 802.1 standards, a seamless mobility framework is introduced in order to manage connections during mobility within a single convergent network. In addition, Mobile IP (MIP), Mobile IPv6, and Proxy Mobile IPv6 are few of IETF standards that were proposed to support seamless mobility across different wireless access networks. In these approaches, the mobile user traffic is encapsulated and transferred through network tunnels between a network middle point and mobile user device.

The other approach with SDN is the FMC based handover [21]. In this method, mobile devices are capable of moving across subnets without network level reconfigurations. The FMC based handover procedure is composed of several features such as end point transparency, identifier, and locator splitting, local mobility optimization and use of tunneling to avoid encapsulation [21]. Compared to other mobile IP-based approaches, the SDN-based FMC handover avoids triangular traffic routing required in MIP-based approaches, and also minimizes network latency caused by encapsulation.

2.2.4 Service Mobility in Mobile Clouds

In mobile clouds services, user mobility must be considered as an important parameter that affects QoS. As a result, in literature, cloud services are migrated along with the user, in order to minimize end-to-end latency, and as well as to maximize the bandwidth.

In the FMC scenario, the basic idea is to localize cloud service instance with the user locations in mobile networks [14] [21] [51]. The initial proposal introduces cloud service architecture to provide higher QoE for mobile users benefitting the distributed cloud [51]. User dedicated cloud instances are migrated to the nearest distributed cloud such that, local network gateways located in the mobile cellular network forward selected application traffic with a minimum communication latency. On the other hand, the architecture provides a number of benefits for mobile network providers by selected local traffic to be locally offloaded and balance the load in the mobile core network. The research work by the authors of [14] presents an implementation of the FMC scenario with software defined networks at their edge networks. The advantage of this approach is that with SDN, endpoint mobility across subnets can be transparently managed by OpenFlow switches at their network edge by separating the identifier and locator addresses.

Cloud service migration mechanism with user mobility was analyzed in literature with mobile-edge clouds, using a model and a decision-making process as to when and where to migrate cloud resources [53]. The user mobility is modeled on one and two-dimensional plane with a random walking scenario and the decision making is carried out using the Markov decision process. The cloud services are provided using mobile-edge clouds which are placed very closer to the user, i.e. base stations, and are connected to each other via the backhaul network. When the mobile user moves from one area to the other, user can connect back to the cloud service via the backhaul network of the decision maker can migrate the cloud service to the nearest mobile edge cloud closer to user's current location. Further, considering clouds at the network edge provide a cloud services orchestration and decision-making process in a mobile edge-cloud environment considering both one and two-dimensional random user mobility [53]. The goal of these approaches is to find the optimum place where the migration cost compensates the traffic backhauling cost. This approach is more cost effective compared to the solution provided by FMC, avoiding unnecessary migrations.

2.3 Cloud Computing

With the rapid growth of computing technologies, powerful computing resources have become available at a lower cost. Meanwhile, the affordability of high-speed communication services has enabled the provisioning of a large numeber of new services over the Internet. As a result, the concept of cloud computing has become highly popular as a new paradigm where computing is considered to be a utility such that it can be rented and released on demand. In cloud computing, resources such as processing, memory, applications, platforms and networking functions are commoditized and delivered to cloud

users in a manner similar to traditional utilities such as water, gas, and electricity. As a result, the cloud computing has become a new concept with ample of business opportunities for enterprises.

The cloud computing concept is backed up by a strong definition given from the National Institute of Standards and Technology (NIST) [54]. According to the NIST definition of cloud, general characteristics of the cloud computing concept can be described as below.

- Pay per Use: In cloud computing users are charged based on the usage of resources and workload carried out on the cloud resources. It allows users to minimize costs for dedicated resources and their management costs by acquiring and resources on demand.
- Network Accessibility: Cloud resources can be accessed via standard networks connections. Therefore, users do not require additional methods to access cloud resources in the provider's location.
- Resource pooling: Cloud resources can be pooled and provided to many customers in the form of physical or virtual resources. These resources can be dynamically allocated to users on demand and the user does not require knowledge of the exact location of the resources.
- Elasticity: Companies can scale up as computing requirements increase and then scale down again as demands decrease. This way the cloud user can assume having a pool of an unlimited number of resources only paying for the currently acquired resources.

- Measured Service: The utilization of cloud resources can be monitored, measured and reported transparently for both the cloud service provider and user.

The concept of grid computing which was started in the early 1990s was an initial approach for providing computing resources as a utility under the model of utility computing. General characteristics of grid computing are similar to the cloud computing concept [55], such as geographical distribution, heterogeneity (host software or hardware), resource sharing, multiple administrations etc. However, cloud computing differs from grids in several ways [26], such as scalability, on demand configurability with virtualization techniques, support the economies of scale. These benefits have given motivation for new business aspects of the pay-as-you-go computing model and hence resulted in the cloud computing being widely accepted over grid computing.

Based on the cloud deployment model, a service provider can be considered as one of the three categories, namely private, public or a hybrid cloud provider. A private cloud service provider delivers cloud services to internal users of an organization. Internal users may or may not be charged for the use of cloud resources. This model offers versatility and convenience while preserving management, control and security. In contrast to private clouds, a public cloud provides compute resources for any public user. Public cloud services are usually sold an on-demand basis and charged based on the customer's used minutes or the hours. Customers only pay for the CPU cycles, storage or bandwidth they consumed. However, public clouds do not provide fine-grained control of the cloud resources for users, which in turn hinder the ability to assure the security and data privacy of cloud users.

Private cloud providers may also acquire additional resources from the public cloud when the demand within its organization increases. A few of such well-known public cloud providers are: Amazon Web Services (AWS), Microsoft Azure, Google APP Engine and Sun's Sun Grid. Apart from the private and public clouds, there is a combination of private clouds and public clouds, which are called Hybrid clouds. The goal of hybrid cloud providers is to create a scalable and automated environment which takes advantage of both what a public cloud infrastructure can provide and control over mission-critical data, as characteristics of the private cloud.

There are three well-known cloud service modes based on the type of services provided. They are namely SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service) and can be considered as three layers from top to bottom. From the conceptual point, each service can be a service to the other layer on the top and can be used to build any cloud service business model. The organization and functionalities of each model show specific characteristics for each layer to form various business models and can be described as below:

SaaS: Software-as-a-Service model allows hosting software applications and web services in cloud infrastructure and deliver over the Internet for SaaS users. Even using a simple resource poor device such as a thin client, SaaS can be accessed with a web browser or a specific web-based application. The important factor here is that the SaaS user has no capability to configure or to identify the underlying cloud hardware such as servers, racks, networks or operating systems within the cloud. However, the users have a limited customizability within the application.

PaaS: Platform-as-a-Service clouds provide an environment to host applications, development tools, and libraries in their cloud infrastructure. Cloud users can subscribe to a PaaS cloud service and receive an API to access the platform for application development, usage of tools and run applications. Similar to SaaS, the users have no capability to configure cloud network, servers, and operating systems. However, the PaaS user has the ability to configure the applications running on the platform.

Table 2.1 Classification of different cloud providers

Features	**Cloud Provider**			
	Amazon EC2	Google App Engine	Microsoft Azure	Sun's Sun Grid
Service Model	IaaS	PaaS	PaaS	IaaS
Service Type	Computing & Storage Services	Web Applications	Web and non-web application	Computing Services
Virtualization Technology	Xen hypervisor	Application container	Via the fabric controller	Job management system
User access interface	Amazon EC2 command line	Web-based administration	Azure portal	Sun Grid web portal
Programming framework	Linux based Amazon Machine Image (AMI)	Python Scripting	Microsoft.NET framework	Java, C, C++, Fortran on Sun Solaris
API over web	Provided	Provided	Provided	Provided

IaaS: IaaS provides fundamental computing resources for its customers. It includes processing, storage, networks and other computing resources that can be provided according to the cloud service characteristics. IaaS providers such as Amazon AWS provide virtual server instances and storage with an API that let users migrate workload to a virtual machine (VM) within the Amazon datacenter. The latter provides users the opportunity to choose the type of VMs and their operating systems. In IaaS clouds, the user obtains a portion of

control over the cloud resources; therefore, users can customize the operating system, networks, storage and other parameters on demand.

In the initial cloud computing architecture, the cloud was considered to be a centralized entity serving from thousands to millions of users [56]. Therefore, large and consolidated centralized datacenters were comprised of several thousands of computing servers and networking devices. Such a centralized cloud hosting method was a well know example of the economies of scale [57]. Large cloud providers such as Amazon, Microsoft, Google, etc., benefit from the economies-of-scale due to the large client base. Centralized clouds are mainly appropriate for large stateless services with a limited data transmission requirement such as the web, batch data analysis, data mining etc. [58]. Key advantages of a centralized cloud are the ability to optimally use resources, and the convenience of managing in a homogenous environment.

However, along with such advantages, centralized clouds due to their architecture inherit several challenges, in particular, the distance from its users. In one hand, it avoids the capability to introduce delay critical applications. On the other hand, the cost of accessing the cloud via the network greatly increases with the distance from the users. Another challenge with the central clouds is the over-provisioning. Cloud providers have to plan for the peak usage and thus in most of the non-peak times, the resources are held idle without any effective utilization. Additionally, in the case of a power or a network failure, the centralized clouds are highly susceptible to outages. A solution for that is to keep redundant power storage/generators and network links which in turn adds to the operational cost of the cloud services.

In current cloud computing architecture, there are several approaches to address the limitations with the centralized cloud concept. One suggestion is to have decentralized clouds that distribute computing functions hierarchically to the nodes at edge networks. Another approach is to use distributed clouds. One of the motivations that led to distributed clouds becoming a reality was task parallelization. In some applications, tasks can be parallelizable such that communication traffic among parallel tasks can be maintained at a minimum level. Similarly, functions carried out by a central cloud can be divided and distributed among different geographical locations where each individual datacenter operates with minimum interaction with others. The initial approach for decentralized clouds introduced the application services eligible for distributed clouds as "embarrassingly distributed cloud services" [56] with small-scale clouds instead of a single centralized cloud.

Distributed clouds are appropriate for applications involving massively distributed data processing and interactive users. In general, distributed clouds can be classified into three categories based on the architecture from tightly coupled to highly dispersed [58]: Multi-datacenter clouds with several tightly coupled datacenter owned by a single cloud provider, loosely coupled multi-service clouds, composed of datacenters with different cloud providers and decentralized edge clouds, composed of edge compute resources to serve dispersed cloud users. With distributed clouds, it is possible to provide cloud services for users from the closest cloud located near to the user location. This approach provides the benefits of lower latency and a better distribution of the workload among distributed datacenters.

2.4 Software Defined Networking

Software Defined Networking (SDN) is novel network architecture introduced recently with the main objective of decoupling the forwarding plane and the control plane of network elements, and thereby makes the network completely programmable. This, in turn, differentiates SDN from traditional networks such that, with SDN, the entire network system becomes a more flexibly manageable application based virtual entity. The openness and programmability of the network control plane given by SDN enable network operators and network administrators to manage their network functions avoiding the limitations in legacy networks which can be described as below.

2.4.1 Traditional Networks and Challenges

Traditionally, networks are mainly composed of proprietary networking devices such as switches and routers that direct network traffic based on the limited knowledge within each device about the entire network. The Internet, which is matured well enough to provide a global connectivity today are composed of such networks. As a result, the network architecture has evolved immensely to support the traditional organizational requirements.

However, most of the current requirements imposed by organizations are challenging to fulfill with this traditional network architecture. Also, meeting the market demands had become virtually impossible due to financial, device-level and management challenges which involve manual configurations to get the best out of proprietary networking devices [20]. As a result, datacenters providers face a number of challenges to support the current demand for the cloud. Several limitations in the current network architecture can be described as below [20]:

- **Dependency on Vendors**: The biggest challenge in today's network industry is choosing appropriate devices to design a network. Different providers manufacture different types of network devices, some of them run their proprietary protocols and services. This hinders the capability of mixed use of multi-vendor devices. Apart from that, the product service life cycle ends within few years because of the added capabilities of newer devices, hence, the buyers are automatically imposed to go for the new devices and services. Lack of a widely accepted standard and open interface among them are two factors which limit the ability of network operators to tailor the network to needs.
- **Complexity**: Today networking technology consists of discreet sets of protocols designed to interconnect hosts reliably based on link speeds, topology, physical distance, etc. Due to business and technical requirements for the delivery of reliable, high-performance, secure and broad network connection, these networking protocols have evolved rapidly. However, networks have become more sophisticated because of a large number of protocols involved to fulfill various business requirements. For example, each time a new device (server or a VM) is added or removed from the system/datacenter, the network administrators have to configure network routers, switches, firewalls, authentication and access control lists, virtual local area networks (VLAN), QoS parameters and other protocol based configurations. However, these configurations are highly dependent on the network topology, device capabilities, vendors, software versions, etc. On the other hand, today's servers support hardware virtualization unlike before, where servers can host a number of VMs that run distributed applications and exchange data among each other. VMs are migrated among servers to optimize and rebalance server workloads within datacenters. As a result, a pattern of a traffic flow generated by a physical endpoint due to a VM changes

after the live migration procedure. Such a scenario can make serious challenges to traditional networks such as changing addressing schemes, routing decisions. Apart from the virtualization, today many enterprises use QoS based services for different applications. However, in traditional networks, these QoS services are followed by a careful manual configuration. Therefore, network level configurations become a major challenge for the IT personnel when each device is required to configure QoS parameters per session, per-application basis.

- **Difficult to implement a new network policy**: In current network architecture lots of protocols were introduced to define the behavior of flow paths within a network based on QoS, ACLs, and other policies. As a result, if a new device is added to an existing large network composed of thousands of middleboxes and network elements, a rigorous configuration is required covering thousands of devices and mechanisms. It could take several hours/days for reconfigurations across the entire network when each time a new device or a virtual machine is added to the network. This makes an extra cost for the firm, and also adds immense complexities regarding the configurations to provide a consistent set of access, security, QoS, and other policies dynamically to the network. The failure to provide these services in time would leave many negative consequences such as security breaches in the enterprise.
- **Lack of Scalability**: Usually datacenters does storing, processing and provide services to its users. Also it can be seen that, a large number of users are accessing datacenter based resources. As the number of users grows, the datacenter needs prompt expansions to meet the increasing user demand; including expansions of infrastructure and the network. However, the problem is when each time a new device is added to a datacenter

based on user requests, entire or a part of the network needs to be reconfigured and managed manually. This procedure involves various users who are requesting services dynamically. It is obvious that such network scaling cannot be done with manual configuration.

In a datacenter, there may be different classes of users who require different quality of service and different performance needs. This drives the datacenter IT to provide customized and on-demand delivery of services to different users. Such individual requirement based policies cannot be easily implemented with existing networks. These requirements can be made feasible with specialized expensive devices at the edge of the networks, this one the other hand causes increasing capital and operational expenditure for enterprises. Additionally, if the current mobile network architecture is considered, the networks are difficult to scale, difficult to manage, inflexible and costly to manage with traditional network technologies [59].

These limitations have made a mismatch between the user/industrial requirements and network capabilities. As a result, new standard with an open network architecture named Software Defined Networking was introduced.

2.4.2 Software Defined Network Architecture

In Software Defined Networking, the control logic is separated from the data-plane and logically centralized to manage the network by applications [20]. Figure 2.6 shows the general SDN architecture as proposed by Open Networking organization. It depicts that the system is divided into three main layers: application layer, a control layer, and network infrastructure layer. The core networking decisions are taken by upper layers. The control layer is mainly responsible for enabling communication between network infrastructure and applications.

In SDN, since entire network intelligence is logically centralized, the control layer has a global view of the entire network. This gives the applications to handle the network resources and applies policies considering the entire network as a single logical switch. In one hand, network operators can control the whole network from a single logical point through the controller simplifying both network design and operations. Also, on the other hand, SDN makes network devices simply controllable by hiding their internal processes and protocols, by merely letting the applications to access the network devices through network controllers.

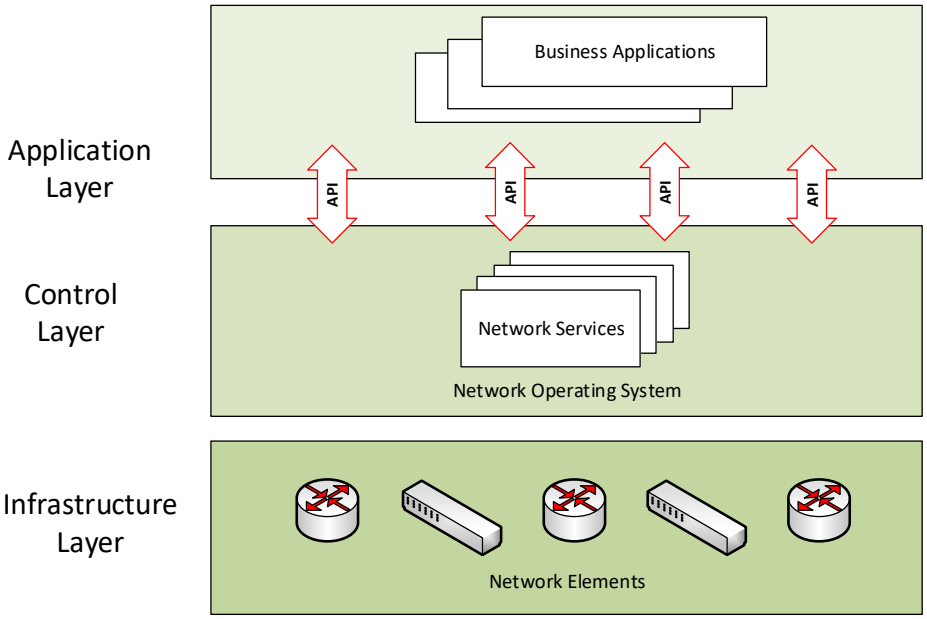


Figure 2.6 Three Layers of the SDN Architecture

A major advantage of SDN for the large scale network operators is that, the entire network can be configured from a single logical point rather than manually configure thousands of network devices. The reason for this flexibility is that, the control logic of all the devices is managed by SDN controllers. Also, because of its centralized intelligence, network operators can control the network in real-time and deploy new applications to serve

dynamically within a short period of time. With SDN, it is not required to wait until the network policies or features are embedded in vendors' proprietary devices. Instead, the IT can develop the required feature as an application and run it on the controller. For this purpose, SDN provides a standard API such that a control algorithm can provide the capability to implement typical network functions such as: routing, access-control, load balancing, QoS optimization, energy optimization along with many user specific functions and policies. As a result, SDN makes the entire network management easily feasible through intelligent orchestration and provisioning systems.

Open Networking Foundation has introduced SDN and provided guidelines for future SDN-based implementations. The open API is used to manage networks with multi-vendor devices, which in turn provide services such as: on-demand resource allocation, virtualized network management, and secures cloud services etc. [20]. There are several implementations based on SDN specifications. One such implementation is OpenFlow [60].

2.4.3 OpenFlow Protocol

OpenFlow is the de facto standard developed by Open Networking Foundation based on SDN specifications. OpenFlow is a control protocol that is being used to define a secure communication between the centralized control logic and the underlying data plane. With OpenFlow, the data plane network elements are identified as OpenFlow switches. The OpenFlow protocol enables the update of OpenFlow switch flow table based on the application running on the control. The architecture of the OpenFlow switch based on the OpenFlow version 1.0 [61] is depicted in Figure 2.7. It shows the secure channel inside the OpenFlow switch which uses OpenFlow protocol to communicate with the external controller. Inside the switch hardware, a flow table is maintained. The flow table contains

three major fields namely: header field, counters, and a set of actions. The header field of the flow table is defined by the controller to classify flows of packets. Then the switch checks the header field against incoming packet to obtain the corresponding action. The action is also configured by the controller. In addition to that, there is a counter module which keeps dataflow statistics such as the packet count.

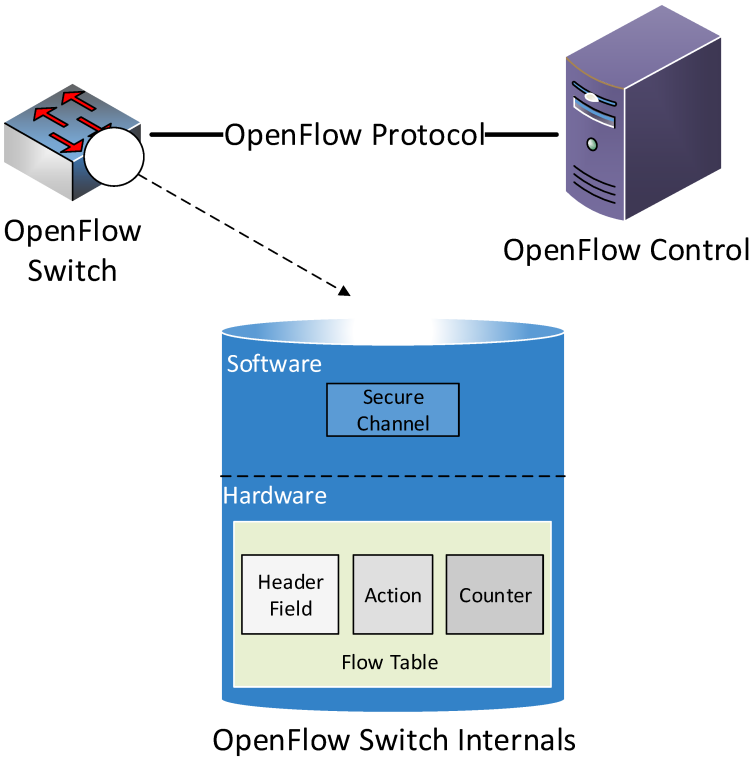


Figure 2.7 OpenFlow Switch Architecture and Internals

The OpenFlow switches can be configured in different ways such that each flow reached to the switch can be treated uniquely based on flow table entries. Most commonly the switches are configured such that, any flow that is not matched to a flow table entry will either be forwarded to the controller or dropped. If the flow rule update request is forwarded to the switch, it is done by sending an event including packet header information using OpenFlow protocol. The application running on top of the controller can define an action for the flow

by sending an action back to the switch. The action message is sent by encapsulating using OpenFlow protocol. With OpenFlow protocol, flow processing can be done either proactively or reactively. In proactive processing, flows are installed into the switch flow table in advance, before the arrival of any matching flow. In contrast to proactive processing, the reactive flow processing method will forward all the unmatched packets to the controller for processing. Both of these approaches contain its own advantages and disadvantages based on the usage. However, due to the extreme flexibility with OpenFlow, a distinct action for each flow can be uniquely defined.

Figure 2.8 shows the fields of an incoming packet that is used to match against header field of the flow table in OpenFlow-v1.0. If multiple match entries are found, a priority value will define the corresponding action for a particular packet. The flows are processed based on the physical layer parameters (ingress/egress ports) to the transport layer header portion of the incoming packet.

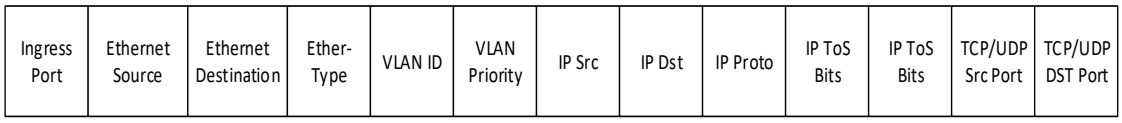


Figure 2.8 Packet Header Fields that are matched against the Switch Flow Table [61]

With further research in OpenFlow, the flow table characteristics were evolved supporting pipeline processing involving multiple flow tables. With pipelining, a matching flow can be forwarded for further processing in subsequent flow tables [62]. Then, the flow table is extended with additional parameters such as priority, timeout, cookies, and instructions.

Due to the immense flexibility and openness provided by OpenFlow protocol, it is being continuously improved to support a larger number of network applications. Furthermore, OpenFlow gives an opportunity to use the standard API to develop network controllers or control applications by avoiding technical barriers, and enabling innovations in communication networks.

2.5 Summary

In this chapter, an analysis has been given of both the concept of cloud computing and the distribution of the cloud. In addition, attention has been given to how clouds can support rich mobile service by addressing the intrinsic limitations of mobile devices. Also, focus has been on different cloud placement approaches, from centralized clouds to distributed close proximity clouds in order to provide a better cloud service to end users. Based on interest in distributed edge clouds, the advantages of distributed and near proximity clouds over the centralized remote clouds are addressed. With the benefits of edge clouds, it is concluded that the distribution of clouds will improve QoS of mobile cloud services. Then, several of the considerable limitations in existing network architecture, especially for cloud providers were presented. Subsequently, an explanation is given to describe how the SDN-based solutions address the legacy network architecture based challenges. Finally, by using the benefits of SDN, attention is directed to proposed approaches for mobile cloud service optimization with user mobility in order to meet user SLA.

3 QoS-aware Regional IaaS Mobile Cloud with Software Defined Networking

3.1 Objective

The objective of this chapter is to provide a detailed description of the QoS-aware IaaS clouds service framework for mobile users with SDN. Initially, the benefits of using distributed regional datacenters closer to mobile users to achieve higher QoS for IaaS is addressed. The role of SDN in network management in both datacenters and wireless networks for regional cloud service is also highlighted in this chapter. Also, a description is given of the proposed cloud controller used to manage IaaS requests for mobile users in the regional cloud service. Additionally, attention is given to how the current solutions with SDN-enabled wireless networks manage user mobility and how they are leveraged to address possible SLA violations during seamless mobility across subnets.

3.2 The Importance of the Proximity for Mobile Clouds

In mobile cloud computing, an application required computing resources are provided using external infrastructure. When cloud resources are being accessed, in the absence of a mobility management mechanism, cloud service access will be disrupted during an event of user handover [45]. This result will adversely affect the QoS. As a result, in order to provide a QoS-aware mobile cloud service, service accessibility with the application required QoS is needed to be maintained irrespective of the user access location.

Furthermore, allocating dedicated communication paths for a set of mobile users to access the cloud is not cost effective. Furthermore, it is highly concerned that, mobile network infrastructure is overly stressed due to the rapid growth of rich mobile services, such as

multimedia applications. As a result, placing cloud resources with close proximity to mobile users has been widely addressed in the research community. The main goal of this approach is to minimizing end-to-end delay from the mobile device to the cloud enabling rich mobile services at the network edge. In addition, another goal is to minimize network traffic at mobile network core.

In most of the close proximity mobile cloud concepts, such as ad-hoc clouds and cloudlets, the major benefit is the low communication delay. Due to the availability of cloud resources in the nearest vicinity, mobile devices can acquire/lease infrastructure on demand for offloading of mobile applications data. In ad-hoc clouds, the cloud resources are provided by nearby mobile devices, forming a mobile device cloud using direct peer-to-peer communication methods. In the cloudlet model, cloudlets are placed in base station or aggregation nodes of a wireless network. As a result, cloudlets can be accessible using Wi-Fi or cellular radio links over a minimum number of hops along the communication path.

However, with the aforementioned small-scale clouds, it is difficult to process complex computational tasks due to resource limitations. Furthermore, in ad-hoc cloud services, addressing the user mobility has become a challenging task as a result of the change of the relative distance between end-points. Similarly, when the mobile users move across networks, cloudlet accessing adds additional cost on the mobile network provider. This cost includes: traffic synchronization, tunneling, and frequent resource relocation. As a result, handling mobility is a costly complex task when using close proximity mobile cloud solutions.

Among proximal mobile cloud services, another research focus is mobile edge clouds. The concept of mobile edge clouds uses small scale datacenter at the last mile closer to user locations. Following the edge computing paradigm, in mobile edge computing, a computing server is installed at the edge of the mobile network such as in base station sites by the network operator. The goal of this approach is to host mobile application services as close as possible to end users. The benefits of the edge cloud include a minimal number of hops that the network traffic needs to flow in order to reach the cloud. Another benefit is the existence of wide range of possibilities for introducing location-based business application services. As a result, the cloud service provider has the ability of maintaining application specific QoS for mobile users within a given network.

By providing application services over external resources such as: remote centralized datacenters, distributed edge datacenters, hierarchically placed cloudlets or directly connected ad-hoc mobile devices, etc., application service providers expect to address the resource limitations in mobile devices to enable rich mobile application. As a result, in this thesis, with the objectives of minimizing cloud access delay for higher QoE, supporting end points mobility, and minimizing service provisioning cost on the service provider, it is proposed that efficient seamless user and service mobility be provided in the following way: QoS-aware IaaS mobile cloud service be designed with physical distributed regional datacenters and SDN-based networks.

3.3 Mobile Cloud Services with Regional Datacenters

The close proximity cloud resource placement methods have shown a number of benefits for introducing rich applications for mobile users. As a result, motivated by the benefits of the edge cloud placement, a distributed regional IaaS cloud architecture is proposed in the thesis. In this cloud architecture, as shown in Figure 3.1, a regional datacenter (RDC) serves mobile users within a given geographical region. The region has been defined as a city, urban area, etc., where it is assumed that, by placing the datacenter, user SLA requirements are always met within the region.

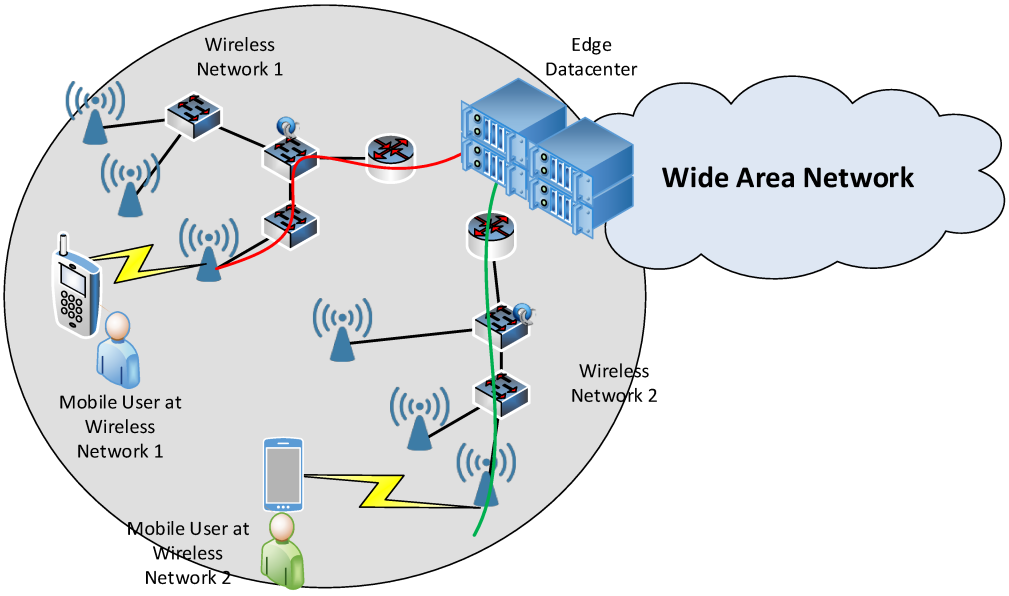


Figure 3.1 Regional Datacenter Serves Mobile Users within a Region

According to the RDC architecture, cloud application traffic is directly forwarded to the nearest regional datacenter using the local network gateways. SLA requirements for mobile user applications are always considered at the time of RDC placement within a region.

Therefore, it was assumed that, the mobility within the region does not violate user SLA for a RDC-based mobile cloud service.

Unlike remote centralized clouds, regional cloud service is composed of geographically distributed RDCs. Each RDC serves a set of local regional mobile users, while a cloud service controller manages the user access to the cloud service. Mobile user applications can request cloud resources on demand by communicating with the local regional datacenter. With an initial resource request, local RDC responds to the mobile user application by creating a dedicated resource(s) application in the optimum RDC. The optimum RDC is considered as a datacenter such that, it provides the user SLA and a minimal service provisioning cost for the cloud service provider. As a result, lower latency requirements of a mobile application can be satisfied by localizing user traffic within a region. Interconnection of the distributed RDCs is done using dedicated network connections over the wide area network (WAN), as shown in Figure 3.2. In contrast to hierarchical and remote cloud approaches, the distributed regional cloud functions as a logically centralized cloud. As a result, synchronization of data from a central cloud to edge cloud will not be required unlike the cloudlets.

The proposed regional cloud service is managed by a logically centralized cloud controller. Few of the functionalities of the cloud controller are distributed in each RDC. Therefore, each RDC contains a local cloud control module which manages the local cloud access and user request processing functions. The centralized controller component maintains the location information of client mobile users and their cloud resource information. In addition, it runs an optimization algorithm to find an optimum location to place cloud resources based on a given user region.

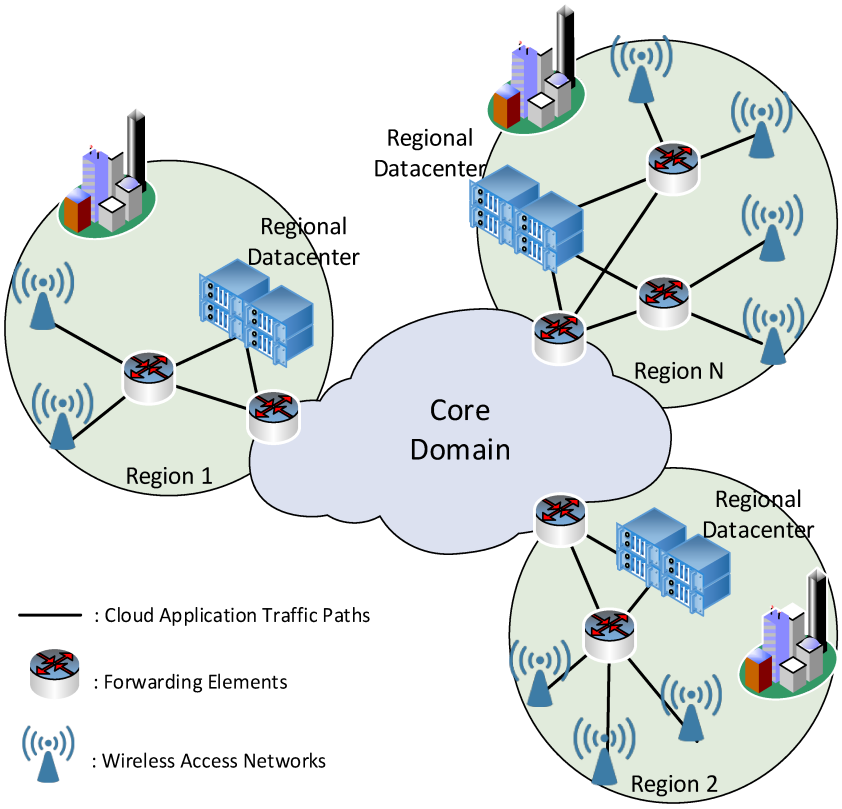


Figure 3.2 Regional Datacenter for a Distributed Mobile Cloud Services

Providing mobile cloud services from regional datacenters introduces a number of challenges in an event of user mobility. Such user mobility can be analyzed in two different ways: mobility within a region or mobility across regions. As per the initial assumption, the user SLA is not violated during the mobility within a region. However, in a case of inter-regional mobility, user traffic is required to be transferred remotely over WAN in order to maintain uninterrupted access to the user dedicated cloud resources. Remotely connecting to the cloud has several challenges such as: (a) exceeding the maximum tolerable latency for application performance, (b) introducing a risk of packet loss and (c) imposing a higher bandwidth usage cost on the cloud service provider. As a result, resource provisioning with

user mobility is a challenging task in the distributed cloud architecture. Thus, the regional cloud service framework is required to address possible SLA violations as a result of user mobility.

In an event of user mobility, the user relocation can be detected in several ways, such as: (a) observing the change of the mobile device's network identity, (b) obtaining user location information from the cloud application. However, instead of the implicit observations, an alternative approach is to obtain user mobility information directly from mobile network providers. Furthermore, if the mobility information can be obtained externally, in a case of mobile handover, the cloud application access can be managed transparently to the end nodes. The benefit of this approach is that it helps to maintain seamless cloud connectivity without requiring additional configurations in either of the device or the datacenter. Therefore, in this thesis, the mobility problem was investigated, and an SDN-based solution was proposed to maintain seamless end-to-end connectivity.

3.4 Mobility Management with SDN

For the regional cloud service optimization with user mobility, it was assumed that, the user mobility happens in one of the two ways: (a). seamless mobile handover while accessing the cloud service, (b). cloud service access with a complete connection re-establishment. During seamless mobile device handover, ongoing communication session is required to be maintained without interruption. Furthermore, mobile device handover can be done within the same wireless access network domain or across access network domains.

In the case of handover between two access points within the same network, user data traffic flow can be maintained by updating traffic paths in local network elements of the wireless network. As a result, ongoing communication is not affected, as well as network

identity is unchanged. However, when a device moves from one wireless network to the other, device network address must be changed based on the destination network subnet. This procedure imposes an ongoing TCP/IP-based session to be re-established causing an interruption to application cloud access. This scenario greatly affects the QoS. As a result, the network identity of a mobile device is required to be continuously kept within its configurations in order to preserve transparency during mobility.

In literature, several research approaches have proposed inter-network mobile handover solutions to overcome the discontinuity during mobile handover. Using Mobile IP, Mobile IPv6, and Hierarchical Mobile IP are some of the approaches being used in LTE mobile cellular networks. In these approaches, the mobile device is able to keep the user device network identity unchanged. The home agent and the foreign agents within the network create tunnels for the moving user to forward packets to the device. This procedure requires the packets to go through a triangular path before reaching the end device. In the locator identifier separation concept, a mobility endpoint can move across subnets without reconfigurations of network settings. This is done by using a locator network address to represent a mobile device's during mobility. The moving mobile device is configured with an IP address which is called the identifier address, and this address is kept during the movement to another network. At the destination network, a locator address is generated to represent the device's current network, and the addresses are mapped to keep the mobile device unnoticeable about the location change.

In Follow-Me-Cloud based handover procedure with SDN follows a similar approach by breaking the locator and identifier address at the edge of the networks. This allows the end points to move among any SDN-enabled network and still transparently managing the end to

end connections by configuring necessary flow rules at the network switches to map between locator address to the identifier address. This procedure also helps the cloud endpoints such as VMs to move across subnets with lesser network overhead. Therefore, analyzing the above approach with SDN, the benefits of having SDN-based wireless network are identified for the provisioning of seamless cloud service with user mobility across wireless networks.

To enable wireless mobility across networks, a wireless access network controller is used which manages the access, address generation and mobility handling in a wireless network. This OpenFlow-based controller is named as the wireless access and mobility management controller (WAMM). Since the OpenFlow controller manages wired network functionalities within the wireless network by excluding the physical characteristics of the wireless channel, it can identify each device in several ways, such as: the incoming port, device physical and network addresses. Additionally, the controller can identify the cloud service traffic flow by looking at the destination IP addresses and transport layer headers. Since handling mobility is out of the scope of the cloud service provider, it does not consider how the user is moving within a region. But, during mobility, if the user identity is changed while accessing the cloud service, such information is required to update the network state of the RDC. Therefore, the WAMM controller in the mobile wireless network is required to update the local RDC with inter-network user handover.

In order to uniquely identify each user device within the wireless network, the mobile device's MAC address is used. Therefore, a given wireless network provider's WAMM controller manages a physical address-based registry, including a list of registered users within its network. Each wireless network maintains a list of registered users, and therefore,

when a device is trying to access a different network, the controller can identify the device as as a foreign device. Therefore, when a user moves, wireless access controller of the destination network allows access to the network by searching the registry and communicating with the remote network which owns it. It is also assumed that, by default, all the users are given the ability to move among wireless networks without any restriction. Furthermore, the physical layer handover procedure including channel selection is avoided since it is out of the scope of this research. Therefore, during an event of user mobility, two states are considered: the user is in the home network and ready to handover and the state soon after the user has handed over to a foreign network.

When the newly accessing device is from a foreign network, initially home network is updated of the device's current location. When the seamless user handover happens, communication continuation is maintained by generating pseudo locator IP (PIP). Therefore, this PIP address information must be updated at the cloud service hosted datacenter network to remap the PIP to an original IP address. In the proposed architecture, the WAMM controller updates the cloud controller about a currently moving device at the time of handover.

Generation of PIP is required only for devices that are handed over from foreign networks. Therefore, when a user device accessing the wireless network without any handover, it is assumed that, the initial data connection request arrives as a DHCP request at the destination WAMM controller. When the DHCP request is detected, the WAMM checks the ownership of the device by looking at the own registry. If the device is not a part of the own network, then a lookup process needs to be carried out to find the source of the device. Afterward the WAMM updates the source wireless network WAMM controller with the

new location details of the user device. When the location is updated, the WAMM runs its DHCP module, that in turn generates a new IP address and send it to the device.

When a mobile device, changes its address at the time of reconnection, the user request may arrive at the RDC as an access request to an existing cloud resource. Then the cloud controller updates the user address information in its database and forwards the cloud access information to the user device.

3.5 User Mobility across Regions

In the proposed regional cloud service framework, the users that are consistently located within the region are assumed as best served and least costly. However, it is expected that, the user is able to move across regions, in cases such as: weekend trips, conference etc. At a destination region, if the user requests for new cloud resources, the controller finds an optimum RDC based on the new location of the mobile user, and it allocates cloud resources at the selected RDC. However, if the user accesses existing cloud resource, the inter-regional user mobility problem needs to be analyzed in order to meet user SLA. In the literature, there are different approaches to address mobility. One way is to predict mobility patterns and to performing mobility-aware cloud resource allocation in advance [63]. However, prediction based approaches also add a risk of accuracy. Cloud resource allocation at a new region where the user is about to move, may not be always successful. With careful predictions considering historical values such as social network data and also as a form of user input would increase the accuracy of the decision-making process in prediction based approaches.

In the proposed approach with RDCs, a reactive cloud service optimization framework is introduced to maintain user SLA, and minimize service provisioning cost in an event of a

user movement. When a user moves from one region to another, and connects back to the cloud service from a destination regional wireless network, the user application request for cloud resource access is forwarded to the new regional datacenter. If the request is for the access of an existing resource, the local RDC controller identifies the user's cloud resources, and builds a traffic path to the service hosted datacenter. Accessing the cloud service in this way adds latency mainly due to traffic tunneling to a remote datacenter through the WAN. The latency may affect the performance of the required services by reducing user QoE. As a result, the centralized cloud controller generates a cloud service optimization process across RDCs based on the user location. The goal of this procedure is to relocate cloud resources such that it will achieve better application performance, and at the same time it will minimize the service provisioning cost.

3.6 System Architecture

The system architecture of IaaS mobile cloud with RDCs is composed of SDN-based wireless access networks, local regional datacenters and a logically centralized cloud controller. The logically centralized controller operates collaboratively with the management modules located in each RDC, and also it maintains a database of allocated resources along with clients' information, as shown in Figure 3.3. The local management module controls the local RDC functions, and it also forwards necessary user requests to the global controller named mobile-cloud service controller (MCSC).

According to Figure 3.3, each region contains SDN-based wireless access networks and a regional datacenter. The figure depicts a single wireless network controller named WAMM, however, it is proposed that, each wireless network may contain its own SDN controller within a region. These wireless networks contain wireless access points and OpenFlow

switches at their IP networks. In a case of an inter-network handover, the generated PIP for a mobile device is informed to the global MCSC controller via the WAMM.

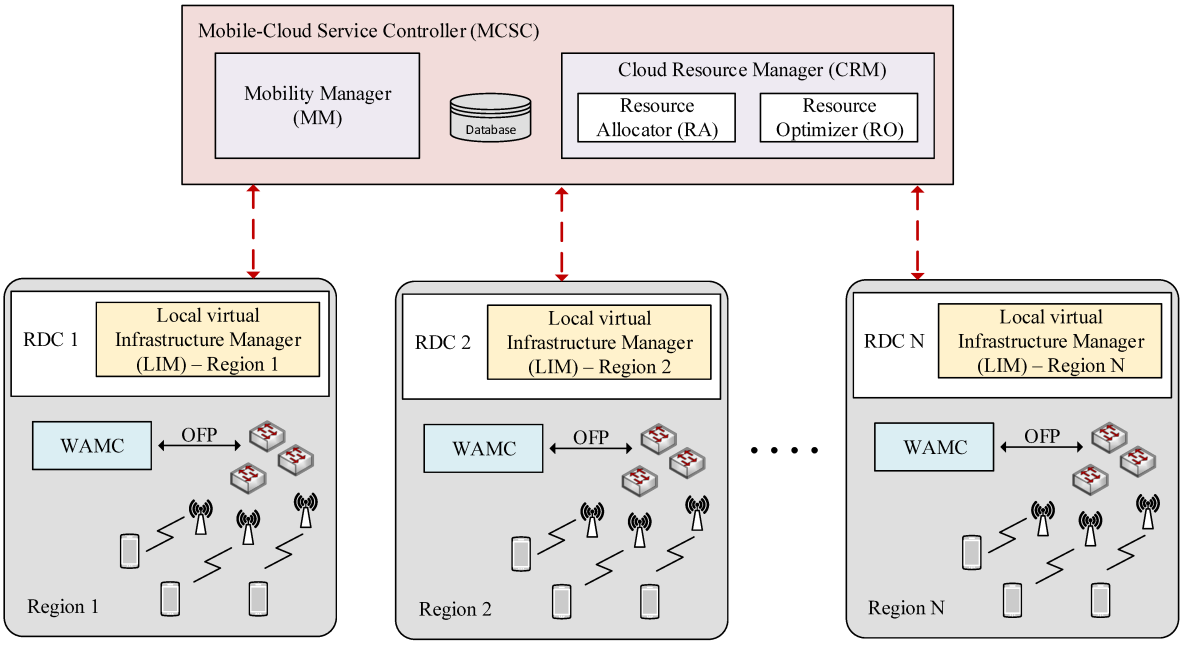


Figure 3.3 Regional Mobile Cloud Service Architecture

In each RDC, the local control module named local virtual infrastructure manager (LIM) handles user access to the cloud service as shown in Figure 3.4. All the mobile users will be connected to the cloud via the LIM of the local RDC. The monitoring module of LIM monitors local resource usage, and also it monitors the inter-RDC WAN link parameters, such as the available bandwidth. The local resource controller located in LIM manages the infrastructure in the local RDC. It contains a network resource controller (NRC), which is an OpenFlow-based controller to manage flows within the RDC. Additionally, the IT resource controller (ITRC) enables the control of virtual cloud resources to the LIM. The cloud service access interface is a logical interface which includes the handling of user requests,

interconnecting local RDC with other RDCs and providing access information to the existing cloud users.

MCSC handles user resource allocation requests and cloud service optimization process once requested by a local RDC. The optimizer located in the MCSC module minimizes service provisioning cost when a user mobility has detected. By communicating other modules in both wireless networks and monitoring modules located in RDCs, it can obtain required information dynamically in order to process an optimization event. The resource allocator (RA) is a function in the cloud resource manager (CRM) generates an output of the resource allocation procedure. LIM uses the ITRC and NRC modules to create virtual resources and update flow rules in the RDC.

The mobile user request for infrastructure in a particular region is accepted by the cloud service access interface as shown in Figure 3.4. If the request is for a new resource, LIM forwards the request to the MCSC. It will in turn run the optimizer to find the optimum location of the cloud resource such that it meets user SLA within the region and minimize service provisioning cost. When the requested resources are created, the user application is provided with the corresponding connectivity information such as the network address and TCP port by the cloud service access interface. If the user request is for the access of an existing cloud resource in local RDC, the user request is directly processed and connectivity information is provided back to the mobile application.

By making the MCSC a centralized module, it has the information of all RDCs and cloud user access information. The actual placement of the MCSC can be within one of the RDCs or in a separate premise which has the minimum access delay to the farthest RDC. However

due to the knowledge of mobile users and their resources, the MCSC can address inter-region user mobility or optimum resource allocation in the minimum cost RDC with SLA.

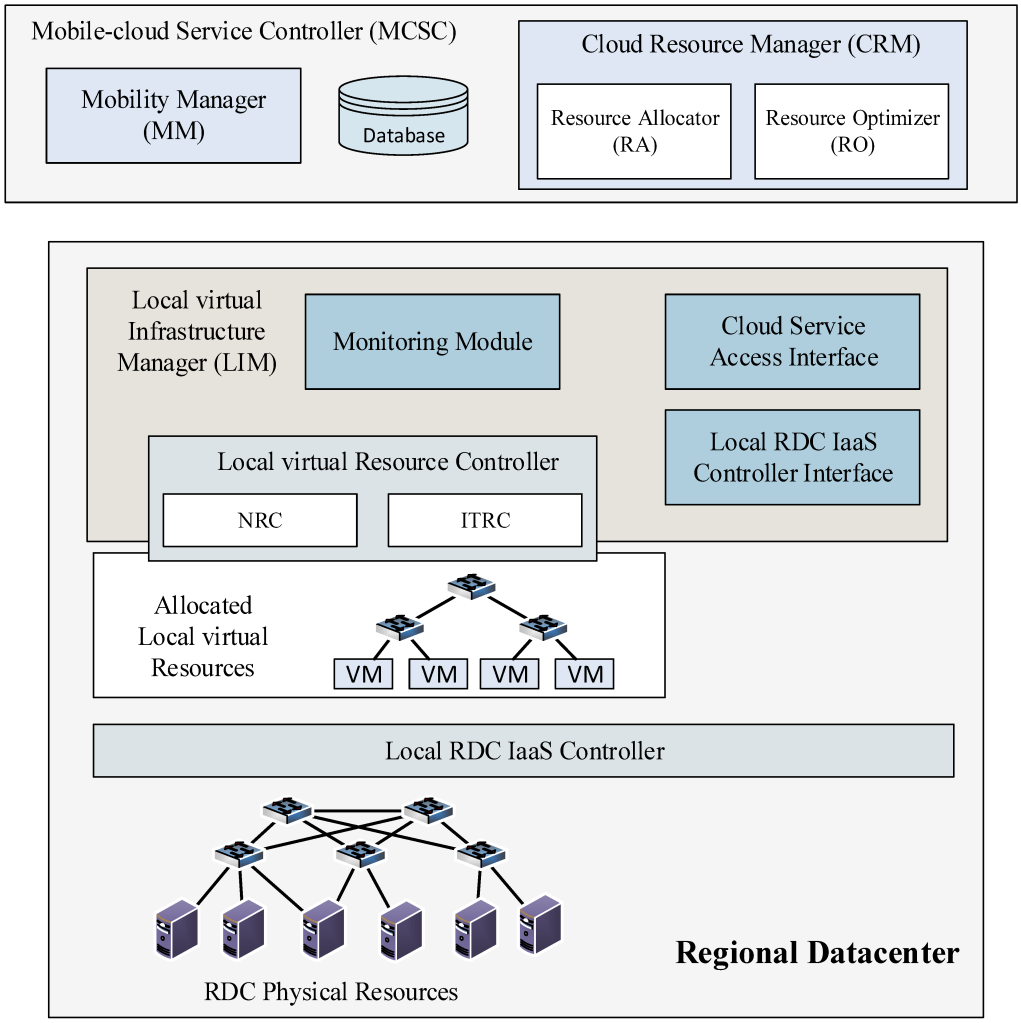


Figure 3.4 Regional Mobile Cloud Service Framework

The WAMM control module contains the functionality of giving access to the wireless network, issue DHCP-based IP addresses and manages network layer connectivity during mobile handover. Additionally, it monitors mobile user IP traffic, and in turn selectively forward them to the local gateway. When managing mobile network handover, the WAMM

module is required to communicate with the corresponding destination wireless network WAMM controller, in order to provide seamless IP network session without breaking ongoing TCP/IP sessions.

When a user handover occurs between networks, the network identity of the device can be maintained by separating IP address into two parts, based on the identifier and the locator concept. When the mobile device is handed over to another wireless network, the destination WAMM controller generates a locator IP address for the device so that the traffic destined to it can be routed to the new wireless network. This address is then forwarded to the source wireless network WAMM controller, which will update its local user information database. At the same time, it updates local network state to change any incoming flow for the device to destination wireless network. The destination network wireless controller updates flow rules in its network switches to map between locator address and identifier address to continue any session transparently.

However, when inter-wireless network handover is carried out, the other end point must be aware of the address change of the client mobile device. In order to distribute locator IP generated for the mobile device with the source wireless network and the respective RDC, WAMM of the destination wireless networks communicate with the MM of the MCSC. When the update is received, the MSCS forwards a network state change request to LIM. Then LIM updates flow rules in its RDC using the NRC controller.

When a user moves geographically and connects from a different region, the local LIM and the new regional RDC processes the request. Then the LIM search if the client resources are located within the new RDC, and if not forwards the request to the global MCSC. The

MCSC finds the resource hosted RDC for the user and forward RDC information back the new regional RDC LIM. As a result, LIM establishes an inter-RDC connection to the source RDC to provide connectivity to the user. Once such remote resource access is identified, the MCSC module generates an optimization procedure using resource optimizer (RO) in CRM to verify both user SLA is met and the operation cost can be minimized.

Then the RO module finds optimization parameters required for the assessment of the cloud resource usage for the given user using the MCSC database and monitoring module located in the new regional RDC. Once the optimization algorithm reaches a decision, it communicates with LIMs at respective RDCs and initiates a seamless resource migration. It also uses the network programmability in the RDCs to route user traffic as soon after the VM is migrated to the new RDC.

3.7 User Mobility and Datacenter Management OpenFlow Network Controllers

With the existence of SDN-based network, a network controller is designed to customize traffic flows within the networks dynamically with pieces of software running on the controller. This software is called an application that specifies a set of actions for a given set of flows. Therefore, in the proposed architecture, the WAMM and NRC module in local resource controller (LRC) are OpenFlow-based controllers, and therefore use OpenFlow protocol to configure network elements in both wireless and datacenter networks.

As shown in Figure 3.5, the WAMM application in wireless access networks shows the functionalities regarding the cloud service. In addition to that, the wireless service providers are allowed to extend the capabilities based on the other requirements. Therefore, within the proposed mobile cloud service with RDC context, each function with the blocks is

abstracted as shown in Figure 3.5. Operations of each block in WAMM module can be described as follow,

Mobility Information Distribution: This interface is used to provide user location and PIP information to the MCSC when the user is handed over across subnets.

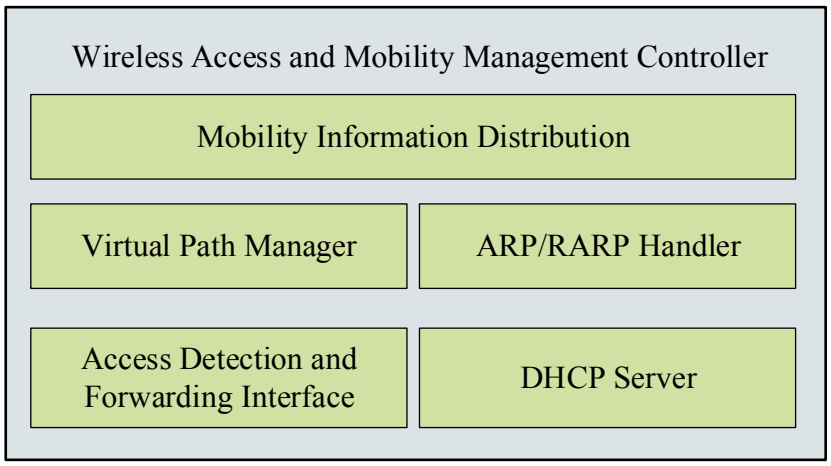


Figure 3.5 User Access And Mobility Management Controller

ARP/RARP Handler: When a device is connected to a wireless network, it sends ARP packet to identify the other network elements within the network. Especially when traffic is destined to a remote network, it sends an ARP packet requesting gateway address. This module successfully maps the local gateway MAC address to the device configured gateway IP address and sends an ARP reply. This way when a device is moved to another network, the transparency can be maintained. Similarly, when a device with PIP communicates at the new wireless network, its gateway requests the MAC address of the device corresponds to PIP. In such situations, the ARP Handler intercept such ARP requests and prepares an ARP reply for the gateway putting device MAC address and PIP.

DHCP Server: DHCP server module is used to process DHCP requests from a device accessing the network. It maintains the DHCP address scheme and allocates network

addresses for devices connecting to the network. Additionally, it generates PIPs for new devices.

Virtual Path Manager: When the device is handed over from a different wireless network, this module created a tunnel to that network. Then the same module in the source wireless network forwards the incoming packets for the device to destination wireless network using the PIP. Therefore, after seamless user handover, rest of the incoming packets can be forwarded to the device with minimum packet drops.

User Location and Address Database: The user location and address database contains a list of registered mobile users and their identities. It uses device MAC address to uniquely identify each device within the network. When a remote device is connected, both the local and source network database is updated with its current wireless network and IP or PIP information.

Access Detection and Forwarding Interface: Access Detection and Forwarding module contain the major functionalities of the network controller. It communicates with the MCSC using User Mobility Monitoring Interface. Addition to that, this module intercepts events from the network elements and passes for processing based on the type of the packet. Additionally, it prepares flow rules and sends them to corresponding OpenFlow switches to update flow rules. When a new device is connected to the network, it searches the device ownership through User Location and Address Database. If it is a remote device, then update these databases and generate an IP using DHCP. For devices coming to the network with the handover, it does the database update procedure and also uses DHCP module to generate a PIP for the device. Then it uses the PIP to map network traffic within the network.

The NRC is also an OpenFlow-based controller composes of several modules based on their functionalities as shown in Figure 3.6 which takes care of processing network events within an RDC. Having individual network resource controllers at each RDC improves the performance of the entire architecture by minimizing event propagation and flow rule installation delay. The individual operations of each module of the NMC can be described as follows,

Network Management Interface: LIM uses this bidirectional interface to access datacenter network and process network level events. When a network address including PIP is generated or a VM, it is passed to the LIM through this interface.

ARP/RARP Handler: ARP/RARP Handler is used to handle ARP and RARP request for VMs within the RDC. After a VM migration from a different RDC, the VM first sends an RARP request asking the network address corresponding to the device. This module prepares an RARP reply from the details gathered at the time of VM live migration. For any VM that sends ARP request asking the gateway or network identity, it forwards such traffic to the corresponding end host. For migrated VMs from different RDCs, the ARP/RARP Handler intercepts such ARP requests, and then it creates ARP replies using VM configured gateway network address and new RDC gateway physical address. Similarly, when the gateway sends an ARP request asking the physical address of a device corresponding to a given PIP, the module generates a reply on behalf of the VM.

DHCP Server: DHCP Server module is used by to generate network addresses for the VMs that are added to the RDC. When a VM is live migrated from a different RDC, this module is used to generate PIP for the VM.

Virtual Path Manager: When a VM is live migrated from a different DC, this module is used to create a virtual tunnel from the source RDC network together with the Virtual Path Manager in that network. Therefore, after VM live migration, the source RDC can forward any incoming packets to the new RDC using cloud service access interface with the PIP as the destination address. Therefore, the packet drops after VM migration can be minimized.

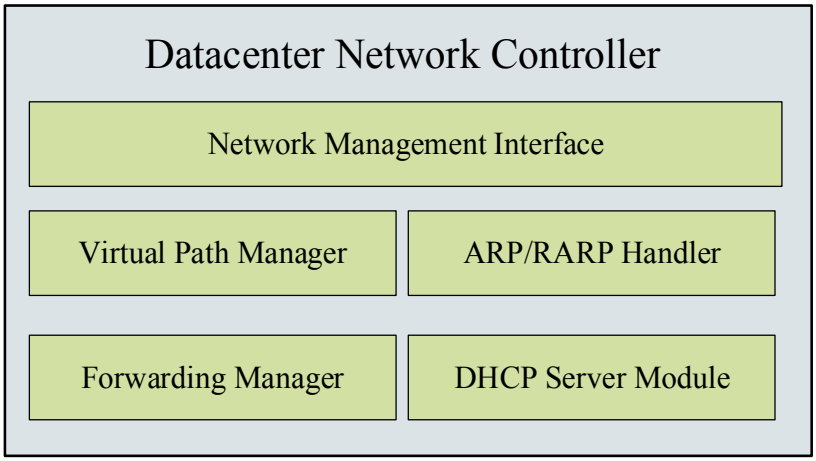


Figure 3.6 Datacenter Network Controller Modules

Forwarding Manager: The Forwarding Manager module listens to events generated at the data plane of the RDC network. Also, it is used OpenFlow protocol to install/delete flow rules into the OpenFlow switches. The Forwarding Manager detects packets generated from the VMs, especially the RARP/ARP which needs special processing are passed to the ARP/RARP Handler. The other packets are similarly passed to the corresponding module within the controller by this module. When a VM is live migrated to the local RDC, this module updates the database about the location changes with the VMs identity. It then requests a network address from the DHCP Server module for a PIP. This information is updated into the local database and also passed to the MCSC to propagate to source RDC.

3.8 Decision Making Process for Optimum VM Placement

The requirement of optimum VM placement is generated when a user has moved from a region to the other. The user mobility is detected by the LIM of the destination RDC, during the cloud resource access. As a result, LIM passes the request to the MCSC, which in turn finds the resource for the mobile user application. Subsequently, MCSC controller runs the optimization algorithm to find the best decision to make at a time of inter-regional user mobility. The optimization algorithm running at the RO module outputs an optimization decision corresponding to a givenn user and a cloud resource (VM) pair.

The optimization algorithm must consider two factors, namely user SLA requirements and minimum cloud service operation cost. It must be considered at the time of the cloud service optimization. Initially, the user SLA requirements are needed to be considered before any consideration on minimizing operational costs. Therefore, when a user moves to a wireless network in a different region, the cloud service optimization algorithm considers the SLA criteria of the mobile user. Parameters that are required for optimization procedure such as end-to-end network latency and minimum bandwidth are continuously monitored by each RDC individually. Since the latency increases after a user region changes, the algorithm considers the latency with higher priority.

A parameter vector containing inter-RDC latency and bandwidth usage is proposed, and it is maintained by the monitoring module located in LIM of each RDC. Additionally, for a VM corresponds to a moved user, RDC calculates the current latency from the mobile device. By aggregating device and inter-RDC latencies among nearby RDCs, the algorithm generates a set of RDCs to place the VM. These RDCs are selected for the optimization procedure only if the SLA requirements can be met by placing the VM at these RDCs. By meeting the SLA

criteria does not guarantee the proper cloud service operation due to several factors. This includes the duration of the stay, cost of migrating and operating a VM to a given RDC and resources availability to place a VM. Maintaining the user SLA and minimizing service provider cost are required to optimize at the same time, hence, the cloud service optimization algorithm can be described as follows:

Consider a scenario in which a given mobile user u, has moved from access network a_i to access network a_j . The variable d_i is defined as a datacenter in region i , where the cloud service (or VM) is originally located, and d_j as the closest datacenter to the user's current location in region j . Access network a_j will forward user application traffic to d_j as it is the regional datacenter that serves a_j.

If $D_{j,u}^{i}$ is the set of RDCs that meets the SLA requirements of the user application, $D_{j,u}^{i}$ can be written as,

$$D_{j,u}^{i} = \{d_1, d_2, d_3, \dots, d_n\} \qquad n \in \mathbb{Z}^{+}$$

If $\left|D_{j,u}^{i}\right| \neq 0$, it is assumed that the regional datacenter $d_j \in D_{j,u}^{i}$ has the minimum latency to the user's device. Also, $D_{j,u}^{i}$ may contain d_i if the SLA is already met at the current VM location.

Let d_k be any datacenter such that $d_k \in D_{j,u}^{i}$ where $\delta^{l_{u,j}}$ and $\delta^{l_{k,j}}$ denotes individual link latency values from user device u to d_j and d_k to d_j respectively. If $\propto$ is the maximum tolerable latency that guarantees SLA;

$$\delta^{l_{k,j}} \leq \propto - \delta^{l_{u,j}} \tag{1}$$

$\delta^{l_{u,j}} = \delta^{l_{k,j}} = 0$ when $k = j$

As shown in Figure 3.7 the algorithm considers three VM placement options based on the parameters gathered. The objective of the algorithm is to find the most suitable datacenter d_k, with minimum service provisioning cost while maintaining user SLA. For the cloud resource access, placing the user VM to the nearest RDC d_j is preferred [47] as it can then provide service with minimum latency and highest bandwidth. However, in the proposed approach with RDCs, moving cloud resources to the nearest datacenter may not always be possible. This is due to the cost-efficiency of resources and energy price variations among datacenters, also due to the lack of available resources at the RDC.

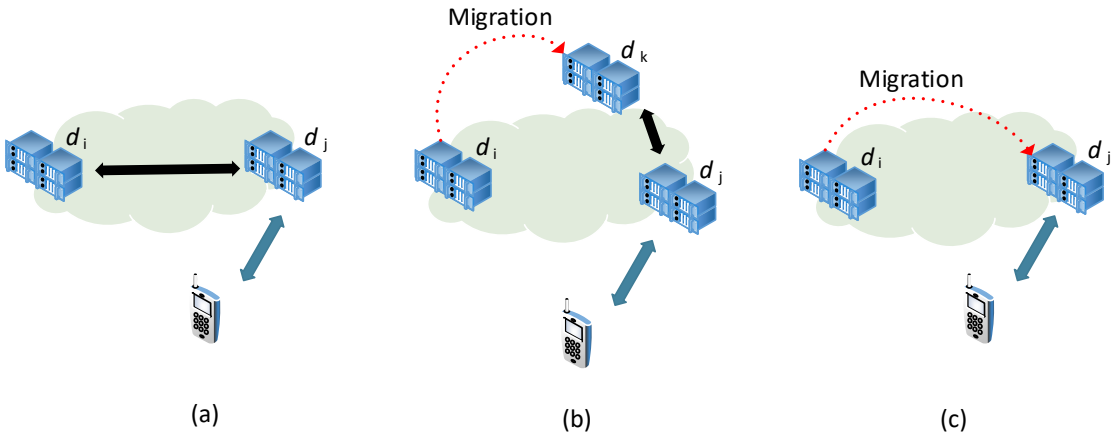


Figure 3.7 Cloud Resource Relocation Considerations

(a). Cloud resource is located at d*i* synchronize traffic through WAN, (b). Cloud resource is migrated to d*k* which in turn synchronize traffic to the d*j*, (c). Cloud resource is migrated to d*j* which in turn locally serve mobile user.

In order to find the optimum resource placement, at first, the case of a single user is considered, that accesses a VM located at d_i, and then moves to region a_j . Once the optimization procedure is triggered after detecting this change in location, it evaluates operational costs to decide whether to keep serving the user from d_i or to move the VM to

a different datacenter d_k . Before the migration process starts, the user accesses his VM located in d_i through WAN links $l_{i,j}$ and $l_{j,u}$ which are between d_i and u . Let $b^{l_{i,u}}$ be the sum of link bandwidth costs on links $l_{i,j}$ and $l_{j,u}$, when the user has moved to a_j and received cloud service access from d_i. If the computing cost per unit time for the normal operation of VM at d_i is represented by c^{d_i}, the cost can be calculated ($N_i^{u_j}$) for running a VM at d_i by:

$$N_i^{u_j}(t) = (b^{l_{i,u}} + c^{d_i})t \tag{2}$$

The parameter c^{d_i} contains the other costs of running the VM, including CPU and memory use. Bandwidth cost $b^{l_{i,u}}$ is the cost involved with WAN usage when the user is accessing the VM. Since the access network always forwards application traffic to the nearest datacenter d_j , the user also accesses the VM through d_j . Thus, link bandwidth cost $b_M^{i,k}$ is the sum of link costs per unit time between d_i and $d_j$during migration. Let $c_M^{i,k}$ be the sum of additional computing costs imposed on both source and destination datacenters by the migration process in a unit time. If the total VM migration time is $T_{i,k}^u$, the VM operation cost in d_i during the period T is $NP_i^{u_j}(T)$, the total cost at time t for both VM migration and operation at datacenter d_i can be written as:

$$C_{i,k}^u(t) = \begin{cases} (b_M^{i,k} + c_M^{i,k})t + NP_i^{u_j}(t) & ; 0 < t < T_{i,k}^u \\ (b_M^{i,k} + c_M^{i,k})T_{i,k}^u + NP_i^{u_j}(T_{i,k}^u) + NP_k^{u_j}(\mathrm{t} - T_{i,k}^u) & ; \mathrm{t} \geq T_{i,k}^u \end{cases} \tag{3}$$

H ere the migration time $T_{i,k}^u$ depends on several factors, as shown in [64]. If the VM's total memory is given by V_{mem}, and the migration throughput (data-rate) is given by R, and the VM page dirty rate is given by D , $T_{i,k}^u$ can be calculated by;

$$T_{i,k}^{u} = \frac{V_{mem}\,(1-\lambda^{n+1})}{R(1-\lambda)} \tag{4}$$

Where, $\lambda = \frac{D}{R}$

The page dirtying rate mainly depends on the type of application executed in the VM at the time of migration. For a given VM executing a specific type of application (file processing, database, streaming etc.), it can be assumed that, a constant page dirtying rate during migration interval. If T_{u_j} is the estimated duration that the user will spend in the region j, time T_u can be used to calculate the benefit of VM migration compared to no migration even if the SLA is already met. The value for T_u can be predicted from historical user mobility patterns, through social network data-mining or by direct application-level interaction with the mobile user [63] [65].

3.9 Infrastructure Acquisition Procedure of Regional Mobile Cloud Service

IaaS regional datacenters for mobile computing can be implemented with IaaS datacenters located in each region. However, in order to acquire/release IaaS on-demand, and to control the datacenter network based on mobile user requests, an efficient resource provisioning mechanism is required at the IaaS datacenter. The SDN-based IaaS framework proposed by the authors of [18], employs a mechanism to accept IaaS requests and allocate virtual resources providing network and IT resource control over the allocated virtual resources. Benefiting this resource provisioning mechanism with SDN, the regional datacenter concept is implemented with an SDN-based resource provisioning framework.

In [18] resource provisioning framework, virtual network slices created by FlowVisor [66] acting as an intermediate transparent proxy between the OpenFlow-based data plane and the SDN controller. The latter was highly motivated by novel SDN-based network slicing approaches such as [67]. Additionally, it uses Libvirt API [68] on KVM hypervisor [69] to create VMs on host server hardware based on available physical resources. By allowing the VM controllability over an IT resource controller IT resources controller, datacenter resources can be dynamically monitored and migrated across other RDCs. Resources monitoring is provided directly by the hypervisor.

Initally, when a resource allocation request is received from the MCSC, the LIM module in the RDC passes a resource request to the resource provisioning module located in the datacenter. Then, it creates the resources and add into the existing slice of resources. The network communication of the resources can be managed by the SDN-based network controller given by the resource provisioning framework. As a result, the RDC cloud application specific resources and network traffic can be managed with LIM. The benefit of this approach is to the convenience of managing virtual resources and network traffic within the datacenters and also allows other applications to co-exist within the same datacenter.

3.10 Summary

Closer placement of the cloud resources for mobile cloud users enhances end-user application service experience due to a number of factors such as: availability of high bandwidth links with low cost and low latency. For IaaS mobile clouds, these requirements have been identified to be more critical for QoS and service provider cost improvement. As a result, in this chapter the concept of regional clouds with regional datacenters is presented. With the proposed regional cloud, mobile device required computing resources are provided

on-demand based on the mobile application request. The regional datacenters architecture is presented with SDN-based network control and IT resource control for the purpose of managing the mobile cloud service respecting SLA. Additionally, a novel regional cloud service framework is proposed with a logically centralized cloud controller that manages regional datacenters. The proposed logically centralized cloud controller enables seamless user mobility and addresses possible SLA violations in the event of inter-regional user mobility.

4 Experimental analysis of the Regional Cloud

4.1 Objective

In this chapter, the research's test environment implementation procedure for the evaluation of RDC concept with SDN-based networks is presented. The implementation in two different ways was prepared by creating a virtual test environment and a physical environment. Therefore, in this chapter, an explanation is given of the characteristics of both the implementations and tools that were used to elaborate functionalities of the test environment. Also, this chapter addresses how to control traffic flow of each individual user at access points of a wireless network using an OpenFlow-based network controller. The VM migration process is explained and focus is on this research's approach to conducting cross-subnet live VM migration progress. Finally, attention is given to the centralized cloud controller approach used to experimentally evaluate the regional cloud service.

4.2 Test Environment for Regional Cloud Service

The performance of the proposed IaaS mobile cloud services framework is measured with an implementation of RDCs together with wireless access networks. The implementation was carried out in two different ways: software based emulation and an implementation of a physical testbed. For the purpose of software-based emulation, a network emulation tool called Mininet [70] was used. Mininet can be used to create real-world-like networks that contain software based network links and switches. Since the focus is on SDN-based networks at datacenters and wireless networks, the OpenFlow-supporting Open Virtual (OVS) switches [71] were used in both of the implementations.

4.2.1 Mininet Test Environment Implementation

In the Mininet emulated environment is composed of four regions covered by a regional datacenter at each region. Each regional datacenter was locally accessible from wireless networks in the respective region. In order to emulate the regional locality in the Mininet emulated topology, lower link latency and higher bandwidth values were used. Similarly, in the WAN links that connect different regional RDCs, the latency is incremented and bandwidth is decremented compared to local values to emulate geographical distance. Figure 4.1 shows the topology that is implemented in Mininet using locality and geographical distance information.

In order to create a WAN in the Mininet, four virtual routers were required as shown in Figure 4.1. WAN links were interconnected each other with higher latency links than that of local. These four routers were supposed to work as network gateways of each local network within a region. At the time of the implementation, a convenient way of implementing a router in Mininet emulation environment was not found. The reason for this is, by default, Mininet does not contain software-based routers, and instead an SDN-based router application on top of the OVS module could be developed. However, the approach was to use Mininet hosts with multiple network interfaces, and then, configure these interfaces with corresponding IP addresses and static routes. Furthermore, by enabling port forwarding among router hosts' Ethernet ports a set of Mininet based virtual routers were created. When automating topology creation with Mininet, the WAN routing tables in each Mininet host were created as a static routing table. Compared to the SDN-based router, the aforementioned approach avoids interruptions and flow rule update delay due to the involvement of an SDN controller for router configuration. Additionally, features such as

delay and bandwidth parameters could be conveniently added to network interfaces of Mininet hosts as necessary unlike a virtual switch working as a router.

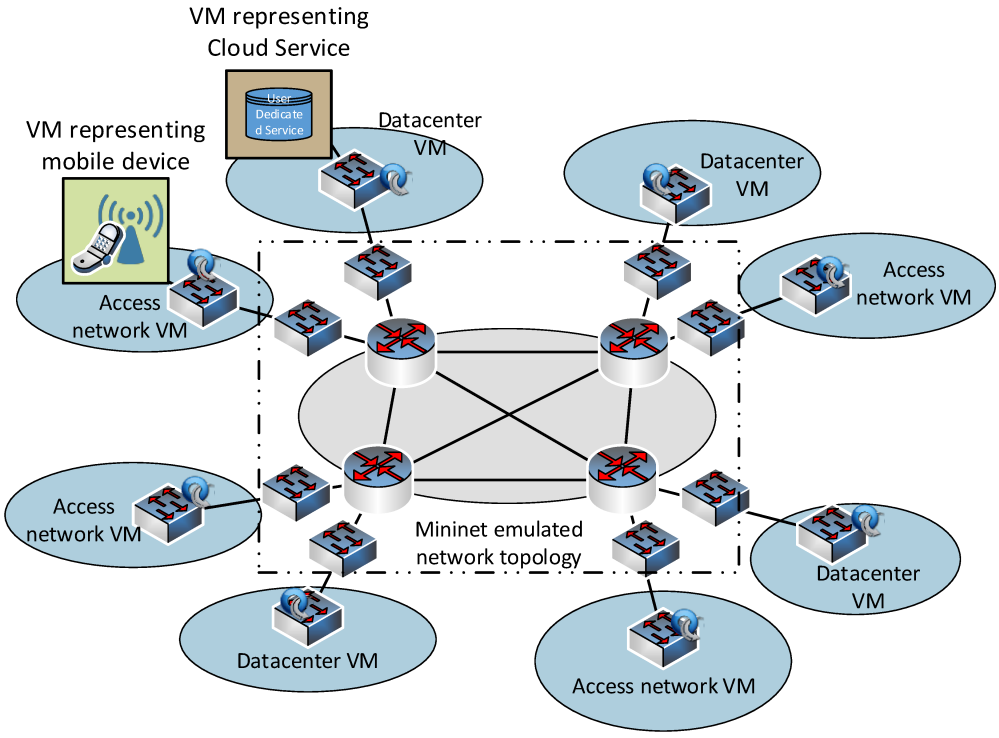


Figure 4.1 Mininet Emulation Environment with Four Regions

When creating the RDC based cloud services architecture, one of the challenges faced with Mininet environment was the complexity of emulating an RDC. A set of Mininet hosts could have been created, and considered as virtual datacenters. However, Mininet hosts do not behave as isolated entities; because they share the same physical machine files system and applications. Since such complete isolation cannot be achieved for regional cloud service testing and VM orchestrations, a set of large KVM based virtual machines (VMs) were created and connected to corresponding Mininet switches. These VMs communicated each other over the Mininet network, and hence required isolation among different DCs was obtained.

The other challenge faced was the simulation of user mobility in a Mininet network. One solution was to remove the Mininet host from a switch and reconnect it to a different Mininet switch. This procedure consumes unnecessary time to emulate user mobility or handover procedure in a virtual environment. As a solution, in order to prove the concept of user relocation based service migration, another set of KVM virtual machines were used as access networks in each region. Since KVM supports nested virtualization with the support of Intel processor hardware, smaller VMs representing mobile devices were created inside larger VMs. The large VMs represented wireless access networks. Inside the wireless access network VM, smaller VMs were connected to the Mininet network via OVS switches, and hence, the network flow was managed by OpenFlow controllers. Therefore, a live migration of a VM that represents a mobile device were performed between host VMs. When the VM is migrated, it is automatically connected to the OVS switch at the destination VM. This way user mobility was represented in virtual environment with minimum interruption for cloud service optimization purpose.

After a VM representing mobile device moved to another network, the user location change was detected by User Mobility Management controllers at the destination wireless network. From the cloud service provider's point of view, there are two ways to detect user mobility, either implicitly or explicitly. In implicit user mobility detection, the cloud service can obtain information about the user from the wireless access network at the time of initial network access or handover. In explicit detection, the cloud resource can observe the traffic it receives for a given user resource and identify the user location based on the network layer information. Else, user location information can be obtained in advance as a user input or from an application layer service. However, in order to avoid complexity, Mobility detection

after user moved to a new region was considered. After a user has moved, the wireless access network controller updated the cloud service framework controller about the user's new location.

The respective KVM host VMs for datacenters and access networks is shown in Figure 4.1. The dotted box contains the Topology created using Mininet. Other elements outside the dotted box are KVM VMs. All the links from VMs to Mininet switches are created by the KVM hypervisor, and it does not require any configuration. However, in addition to the Mininet topology, a physical testbed was implemented using real wireless hosts to measure the performance of the proposed framework.

4.2.2 Physical Test Environment Implementation

In the physical testbed implementation, six Linux-based workstations running Ubuntu 14.04 Server edition were created. These workstations were equipped with Intel Core 2 Duo processors and 4GB RAM. Two workstations were used to create two distributed datacenters, and the other two workstations were used to create wireless networks. Additionally, another two workstations were used to operate as physical routers in order to create four distinct network domains to emulate a WAN. It was assumed that, one datacenter and a wireless network are collocated in the same region and the other combination was similarly assumed to be located in a different region.

For the virtualization in the physical implementation, QEMU-KVM hypervisor at Linux workstations was used. These workstations were considered as regional datacenters, and hosted cloud resources along with network elements. The computing functions were carried out by the virtual machines created in the Linux workstation. During the time of VMs creation, VMs were bridged to OVS switches at a particular workstation. These OVS

switches were also connected to the Ethernet interface of the respective server machine and hence the Ethernet interface acted as the gateway for the RDC. These OVS switches at datacenters were configured to be controlled by the OpenFlow controllers.

In order to emulate the physical distance Linux Advance Routing and Traffic Control (TC) service [72] was used to add latency values in the router Ethernet ports. In the server machines used as wireless networks, OVS Switches were bridged to both wireless interface card and Ethernet interface card to obtain flow level controllability of wireless devices. Similarly, virtual machines in datacenters were also bridged to Ethernet port via OVS Switches. In both cases OVS switches worked as OpenFlow switches and their flow control was done by OpenFlow controllers. As mentioned above all the server machines had an extra network interface and its called as the management network. A management network was used to distribute control information and carry out VM live migrations.

In the physical testbed, any bandwidth restriction was not placed when connecting Linux workstations. The link bandwidths were fixed to the maximum bandwidth that the networks cards in each computer were capable of. In contrast to real-world networks, the local link bandwidth was set to 100MPBS each. This is mainly due to the limitations of physical network interface cards. Instead, link latency values were set to different values using Linux TC in order to emulate physical distance in the testbed.

4.3 Designing an OpenFlow Based Wireless Network

In order obtain dynamic flow-level controllability of mobile users, an OpenFlow-based wireless network in Linux environment were created. However, the controllability of flows was done at the fixed IP network of the wireless network. In order to create a wireless network, a regular PCI wireless network adaptor was used, along with Hostapd daemon [74]

and Open VSwitch (OVS). In a Linux workstation, an OVS switch was created to use as a bridge between the wireless network and Ethernet network. This OVS switch was given remote control using OpenFlow version 1.0 and worked as an OpenFlow switch. In order to create a wireless network on the wireless network adaptor, Hostapd daemon was used. It allowed IEEE 802.11g access point management and authentication capabilities on wireless adaptors. By creating a wireless network using Hostapd allowed wireless clients to connect to the OpenFlow-based network. Communication flows through the wireless network was visible at the network controller on OVS switch and the OpenFlow-based application could handle those flows accordingly.

When implementing the SDN-based mobile handover, the FMC-based seamless handover process was used. Authors of [21] analyze the FMC based handover procedure and compared it with other approaches, such as Mobile IP and HMIP to address end-points' mobility. FMC enables mobility across subnets in TCP/IP networks. The goal of FMC is the splitting of locator and identifier separation at the network edge. With OpenFlow, original addresses of the end points can be used after mobility. However, OpenFlow switches at the edge networks required to map the locator addresses back to identifier addresses in order to maintain TCP/IP sessions. In [14] [21], authors created FMC with multiple distributed controllers, which were called home controllers, foreign controllers and correspondent controllers, respect to the mobile entity. After the detection of user mobility, home controller sends device IP and ID information to the foreign controller. Then foreign controller generates a locator IP for the device and updates the home controller and correspondent controller. When the correspondent controller receives the address update, it installs flow rules within the switches to map identifier/locator information. Following the seamless

handover process in FMC, the SDN-based wireless control logic was implemented to provide cross subnet user mobility which was also called as the wireless access and mobility management controller.

In the proposed architecture, the OpenFlow controllers of wireless access networks are given the visibility of application traffic that are generated from mobile devices. Therefore, wireless networks are directed to forwards application traffic towards the nearest regional datacenter. In the proposed framework, the mobile device MAC address was used to uniquely identify each user in the wireless network. As a general scenario, by creating separate layer 3 subnets, one access network was kept independent from the other. Moreover, the WAMM controllers keep a database of registered mobile devices within their network, and also they manage network handover and mobility across wireless networks. The WAMM controllers are also responsible for managing the traffic between mobile device application and cloud service instance during an event of a handover. By using a pseudo network address as a locator address of the mobile device, a seamless connection throughout user mobility was maintained by wireless and datacenter network controllers.

In both mobile handover and cloud service migration, after a change in the network status, OpenFlow controllers immediately deleted invalid matching flow rules in respective switches. Then, they installed new flow rules instead of reactive flow processing to forward traffic by the mobile device to cloud service instance and vice versa, until an optimization decision was taken. This way the controller load created by reactive flow processing was minimized until a decision is made. As a result, if user desired cloud service instance relocation was not required, the controller left existing flow rules within the switches. On

the other hand, if service relocation was required across RDC, it may change the location and use a different location address at the new RDC.

4.4 Centralized Control Algorithms for the framework

In both of the physical implementation and Mininet testbed, the regional cloud concept was evaluated using a centralized cloud controller. The centralized controller was combined with the functionalities of: cloud service optimization, orchestration and network control at datacenters and wireless networks. Network control was done by accessing OpenFlow switches at each network using the OpenFlow protocol. Also, the controller accessed hypervisors at source and destination datacenters via Libvirt API in order to obtain VM information, and also to initiate a VM migration and monitor the virtual machine migration process.

The network controlling component of the centralized controller identified each network switch by their data path ID (DPID). Using the DPID, and maintaining a list of connection objects for each switch, the network controller could access the switches and add/remove flow rules individually. After detecting events of the data plane, the network controller generated flow rules per each switch, based on the type of the event. For the testing purposes, a scenario of a mobile user application accessing a single VM was considered. As shown in figure 6.1, initially the mobile user is allocated a cloud service instance at the nearest edge datacenter.

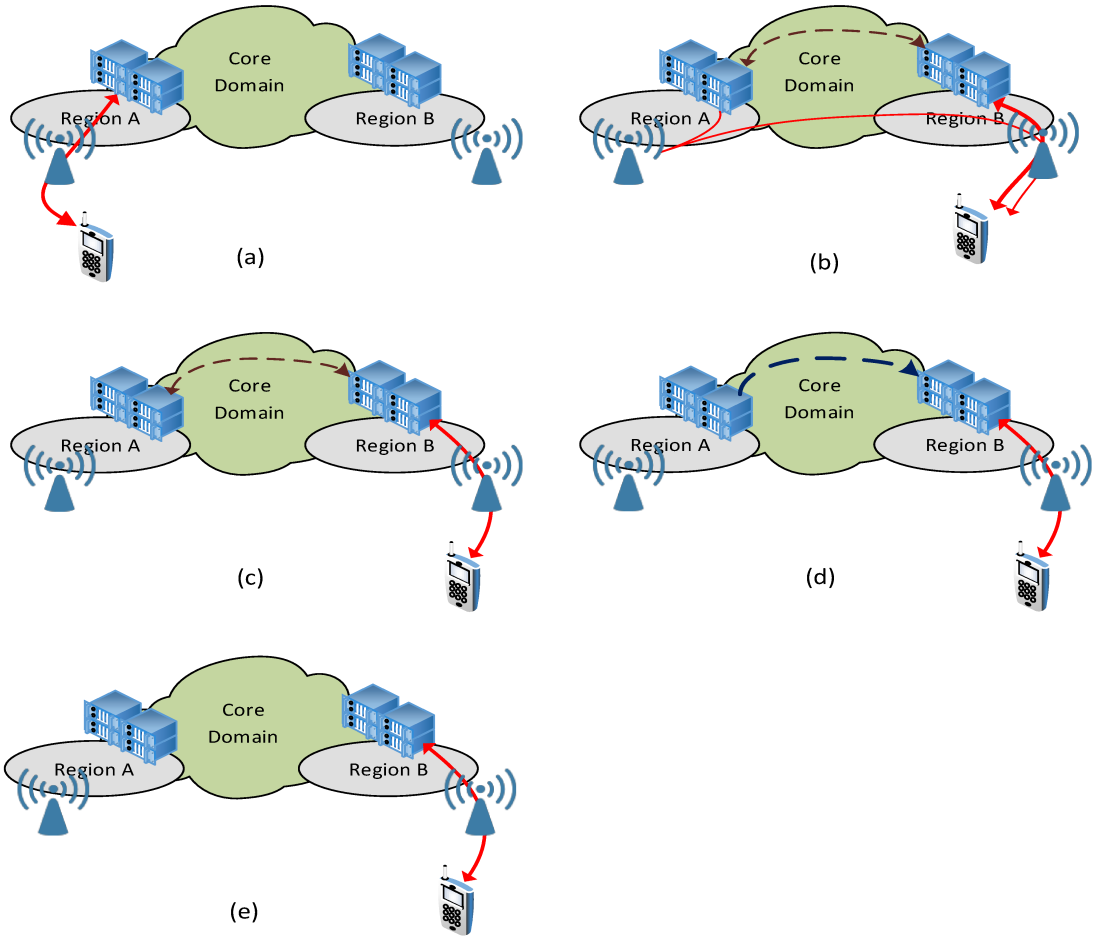


Figure 4.2 Flow Path Configuration during User and Service Mobility

Mobile user is accessing cloud resources locally, (b). Just after mobile device handover to a separate ion, (c). Synchronization of source RDC traffic, (d). Cloud service instance is being migrated to imum datacenter, (e). User is accessing optimized cloud service instance at new location

The number of flow rules required for seamless connectivity during mobility or migration depends on the network elements in the flow path. Assuming a single network element in each network, the minimum number of flow rules required for different scenarios can be analyzed carefully by observing the direction of the flow. Local cloud service access required minimum four flow rules, two per each switch in both wireless network and the datacenter network. User handover with OpenFlow requires additional flow rules. At the

new wireless network, two flow rules are required to map between pseudo IP address assigned to the device and original IP address.

Similarly, two rules are required to remap the pseudo-IP address of the device to the original IP address. Additionally, another rule is required at the previous wireless network to reroute all the incoming packets towards new user location until the flow rule at the datacenter network is updated. Otherwise, the stream of packets kept sending to the device source network will be lost as soon after the device is moved to the destination network.

4.5 Virtual Machines for Regional Cloud Service

In order to model cloud infrastructure for mobile users, possible solutions were investigated with their advantages. In literature, use of a VM, as the user dedicated cloud resource has been considered as the well-known solution due to its considerable benefits. One of such advantage is the VM migration process. VM migration either offline or live has given immense flexibility for cloud service providers to manage their infrastructure and minimize OPEX [15] [64]. Offline VM migration provides ways to backup applications running in one datacenter in a different datacenter when VM live migration provides the service continuity while migration process. In enterprises, VM live migration provides disaster recovery capability, server consolidation, fault tolerance, high availability and load balancing capability.

In the implementation, VMs were used as dedicated cloud instances for each mobile user. However, with the user mobility the cloud controller was required to orchestrate cloud resources among regional datacenters. As a result, the VM migration techniques were carried out in order to relocate cloud resources while maintaining user services consistently.

The following description is an analysis about the VM migration process that is beneficial to further optimize the cloud relocation process.

4.5.1 Live Migration of Virtual Machines

In literature, many researchers have investigated the VM live migration process and its optimizations to provide undisrupted cloud services. These optimizations can be analyzed in two different ways, network-based and host-based [75]. In host-based analysis, VM migration is carried out with two different hosts containing compatible hypervisors. In the process of VM migration, the main factors affecting the live migration process is the size of VM memory, network traffic, memory page dirtying rate, network transmission rate. During VM live migration, the essential requirement is the communication link between the hosts from VM source to the destination. During the process, the VM memory is copied to the destination host and it is called the pre-copy stage. As shown in Figure 4.3, the memory is copied iteratively during live migration. However, at the same time, it gets dirtied due to the running applications in the VM. After several iterations of the VM pre-copy cycles, VM migration process is terminated due to the following factors [64],

- VM memory dirtying rate is more than the transmission rate
- Remaining memory threshold has reached to complete migration
- Number of iterations exceeds the pre-set value
- Network traffic is larger compared to the VM memory

Once the copying phase is completed, the VM is stopped for a shorter time and continued from the destination host [15]. This stage is called the stop-and-copy phase. Since the stop-and-copy accounts a smaller time there is a negligible downtime for the VM, and hence, the user does not observe a considerable disruption.

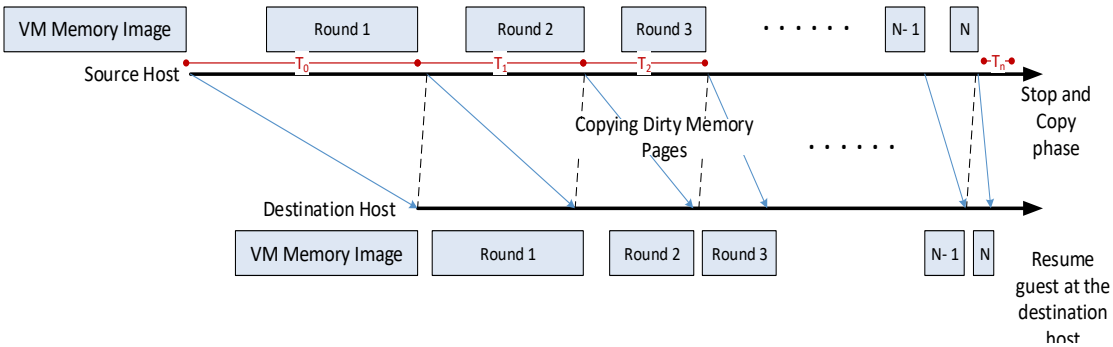


Figure 4.3 Stages of VM Live Migration Process

Migrating VMs within LANs is a simple process due to the availability intra-DC links with higher bandwidth and lower latency [76]. Also, VM migration within LAN does not require additional configurations in the network. Therefore, any communication session such as TCP can be transparently maintained during VM relocation. Compared to LAN VM migration, WAN migration is considerably challenging due to bandwidth constraints and accounted latency along the inter-DC links. Even with the availability of high-speed links, it is not possible to dedicate a large bandwidth link for a single VM migration that possibly does not happen often. For an example, [76] mentions that VMWare live migration required at least 622MBPS dedicated link for VM migration with maximum 5ms latency. Further, authors of [77], as a challenge of VM migration, shows during VM migration, a shared storage must to be available that is accessible at both source and destination hosts. The second issue the authors points out is the network level configurations. In [15] and [75], authors propose live migration techniques with SDN across datacenters. For the seamless operation, the migrated VM requires maintaining the MAC and IP address information at the destination datacenter to maintain the consistency.

4.5.2 Seamless Service Migration among Datacenters

When a VM is live migrated to a destination network, it first broadcast a reverse ARP (RARP) request (to know its IP address) or a gratuitous ARP (GARP) request (to initialize its IP stack). Therefore, in VM live migration scenario, such layer 2 requests must be handled by the network to ensure continuous network connectivity of the VM. For the inter-datacenter VM live migration, L2VPN can be used to distribute ARP requests in layer 2. However, this approach is not efficient due to the requirement of pre-established L2-VPN among all the distributed datacenters to broadcast ARP requests. Instead, they use SDN to update the flow paths in the network to forward traffic destined to migrated VM to the destination datacenter. Therefore, in these papers, authors use the SDN controllers to map layer-2 and layer-3 network information and forward traffic to optimize the live migration process. In [15] authors show the limitations of the non-SDN based approaches to maintaining network consistency after VM live migration across subnets, such as Mobile IP and Dynamic DNS.

In Mobile IP-based solutions, the home agent involves forwarding traffic to the migrated VM. Therefore, such a triangular routing path increases the latency in the communication channel and hence imposes possible risks of SLA violations in time-critical mobile cloud services. Further, in both Dynamic DNS requires changes in the network settings, which is not desirable in live migration processes. In the paper, they propose CrossRoads architecture which shows the seamless VM migration across datacenter can be achieved with SDN with no disruption to the end user. In CrossRoads architecture, they propose the use of pseudo IP addresses and MAC addresses for migrated VMs at the destination datacenter which are called the locator identities. Keeping the network configurations within the VM, which are

considered as the identifiers of the VM, the OpenFlow-based controller application generates and maps network information to maintain the transparency of the communication process. This research work gives a strong insight into network state management after cross subnet VM live migration process and can be used for many cloud-based scenarios.

Implementation of the Follow-Me-Cloud (FMC) with OpenFlow protocol based NOX controller is described in [73] which enables end points and service mobility in cloud datacenters. The author's mentions that the detection of the movement of a user endpoint is challenging and can be measured directly by looking at the access points attached or indirectly by the network later address. Regarding the end points mobility, authors of [21] mentions that the mobility detection entity is done by an external entity which is outside the scope of FMC and can be a layer 2 entity. In [73] authors provide an architecture for the OpenFlow controller with specific modules that are used to mobility detection, VM management, DHCP handling, ARP request handling, location mapping, location management and a database to manage mobility information about the end nodes.

4.5.3 Seamless VM migration Process in Regional Datacenters

In the RDC architecture, when the optimization procedure generates a migration request, the cloud service instance (VM) migration process was employed. Since the proposed approach was to provide seamless transparent cloud service to the end user, live migration of virtual machines was considered among distributed datacenters. This approach is highly motivated by the CrossRoads architecture for seamless VM migration across subnets.

If the optimizer makes a decision to migrate a VM, then, in order to maintain connectivity with the user, 5 flow rules are required to be updated on OpenFlow switches. Two flow rules are required to be installed at the datacenter network in order to map between: pseudo IP of

the VM and pseudo IP of the user device to the original IP of the VM and original IP of the device. Similarly, two flow rules are required at the wireless network to map between pseudo IP addresses of both VM and mobile device to their original IP addresses. Since, at the time of migration, all the traffic from user device are being forwarded to the source datacenter an additional flow rule must be installed at the source datacenter. When the migration is completed, until a new rule installed at the wireless network to forward new traffic to the new datacenter, the packet stream traveling to source datacenter network could be discarded. Therefore, another rule at the source datacenter network must be installed to reroute any incoming packet towards VM to the new datacenter. When new rules are updated at the wireless network, the existing flow rules are no longer needed, and hence when the user is in a new region, and the corresponding VM is at the optimum RDC, four flow rules are required.

Finally, for the purpose of user for cloud service optimization, minimum 8 flow rules are required. Similarly, for layer-3 packets corresponds to cloud service with M number of mobile users, the total rules installed at any time instance $\boldsymbol{\delta}$:

$$4M \leq \boldsymbol{\delta} \leq 5M$$

This excludes communications occurs within a datacenter except cloud application traffic. Additionally, in VM migration and user handover requires layer-2 frame handling such as RARP and ARP.

If the RDC cloud for mobile users contains a centralized controller, a large number of network state update requests will be received to the network controller. At each event, the network controller is required to generate a set of appropriate flow rules by consulting other modules within the controller. As a result, in order to avoid frequent access of the network

controller for a particular flow, flow rules are installed with a large timeout, instead of reactive flow processing. Each flow rule expiration accounts for a new flow generation time of the controller. Therefore, a long timeout minimizes the controller involvement. If an event generated from the User Mobility Management controller regarding a user handover, an installed flow path from mobile device to VM is deleted by accessing respective switches.

4.6 Summary

In this chapter, the implementation procedure for the RDC cloud architecture was described. The tools and methodologies used for both of the physical and software emulated test environment is also presented. It is identified that, in order to emulate user mobility, a comprehensive framework is required that measures physical layer characteristics such as the signal level, in order to carry out a mobile handover procedure. It is also noted that, the Mininet emulation environment is highly efficient for the development of proof-of-concept test environment. In addition to that, several software and hardware limitations for physical emulation testbed implementation were identified. As a result, software bugs were listed and, alternative approaches were followed. Finally, the physical implementation and the Mininet emulation environment were successfully developed to evaluate SDN-based RDC concept for mobile computing.

5 Performance Evaluation

5.1 Objective

In this chapter, an evaluation of the performance and behavior of the experimental testbed for regional cloud service is presented. Initially, the resource allocation and user mobility scenario are described, along with an explanation how the proposed controller handles each event individually. From the experimental results, it is observed that the seamless cloud service orchestration among regional datacenters can be achieved with SDN in their local networks. Then an analysis is given of the outcomes for several applications running in the user dedicated VM as a cloud service during the optimization and migration process.

5.2 Regional Cloud Service Framework Evaluation Test Case

To test and simulate a scenario of inter-region user mobility; Wi-Fi networks with different subnets were implemented. Additionally, the physical layer wireless handover procedure was omitted due to the implementation of handover in physical layer is not within the scope of this research. Instead, wireless network connectivity of mobile devices was switched manually in the case of physical implementation or live-migrate the mobile device VM to the destination wireless network in the Mininet environment. Similarly, in mobile device handover scenario, during service migration with the user movement, the concept of locater and identifier separation procedure was employed for seamless VM migration across datacenter networks.

For the purpose of cloud service relocation simulation, VMs were live-migrated from a source workstation to the other. Here, a small scale virtual machine was used on KVM

hypervisor with 128MB of RAM with 10GB storage running Ubuntu 10.4 Server edition as the user cloud instance. In VM movement, there are two cases possible, the first one with the VM live migration with the running state and second scenario is with the complete migration with the storage. The VM migration scenario is addressed in the [33] where different VM migration technique can be employed for different application scenarios. However, in this thesis' experiments, VM live migration was conducted, and observed the results for different user applications.

For evaluation purpose, two architectural variations were used. At first, RDC architecture was implemented using a separate network with very low latency and 100MBPS bandwidth in order to carry the control traffic and conduct VM live migration. This network was created parallel to the user data network, and also considered as the management network. Secondly, the same data network was used for both VM migration and control message transfer.

5.3 Evaluation Criteria and Results

Test results were taken by considering following scenarios:

a) Mobile application requests for cloud resources via local regional datacenter. Then resource allocator finds the optimum regional datacenter and creates a virtual machine. Once VM is created, the mobile application receives the IP address of the VM. Meanwhile, controller updates the database and mobile application connects to VM. Subsequently, the entire time taken to receive VM IP information back to the mobile device was measured.

b) The mobile user moves among SDN-based Wi-Fi networks within a local region. In this scenario, the time taken to update network state was measured and showed the effectiveness of seamless mobility using ICMP ping results.

c) The mobile user moves across region and request to access the existing user resource. As a result, a new connection is established to the source regional datacenter, and then, the user can remotely access the VM. Subsequently, the controller generates a cloud resource optimization process. The time taken to collect information from remote RDCs in order to find the optimum datacenter was measured. Afterwards, live VM migration is conducted to the corresponding datacenter. Here, the seamless connectivity during service migration across datacenters was investigated and plotted.

d) Different mobile applications are used to measure the cloud resource relocation performance. A media file server and an FTP based file server applications are employed in the VM. In these scenarios, the VM migration time for different applications was measured.

In scenario (a), the SDN-based framework for IaaS clouds proposed by [18] was used. The Figure 5.1 shows the VM creation time after receiving a resource request from a mobile application. According to the graph, the VM creation time in Mininet testbed is appeared to be constant. In the physical testbed, a fluctuation in the time of VM creation time was observed. It was assumed that the fluctuation is caused by the physical testbed, and is mainly due to the additional overheads added by the physical components, such as: different types of Ethernet network interface cards, background application workload, and background network traffic. However, according to the experimental results on VM creation, it can be

concluded that, the total time is taken to find an optimum RDC, and create a VM and send the VM IP to the mobile application is constant for the given dataset.

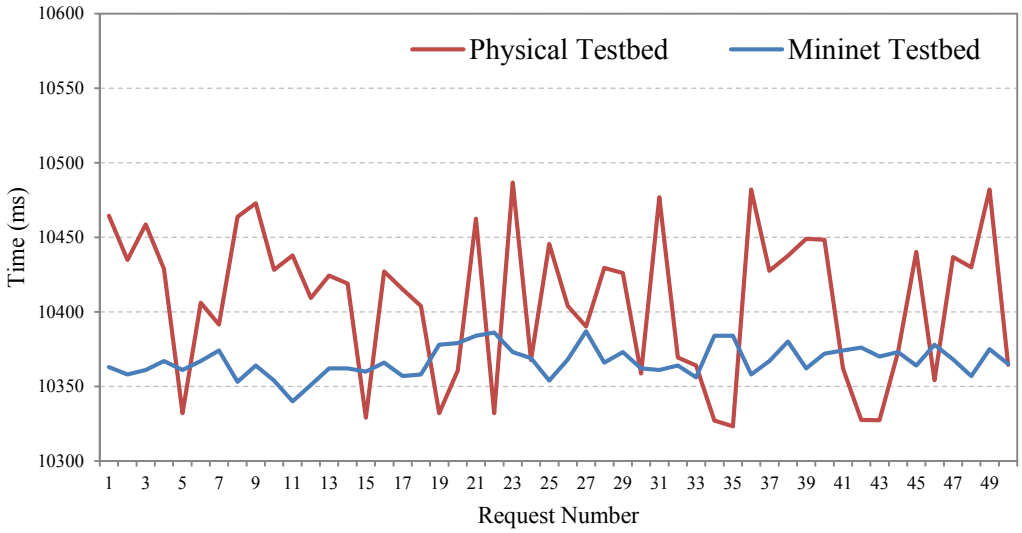


Figure 5.1 VM Creation Time of Physical and Mininet Test Environments

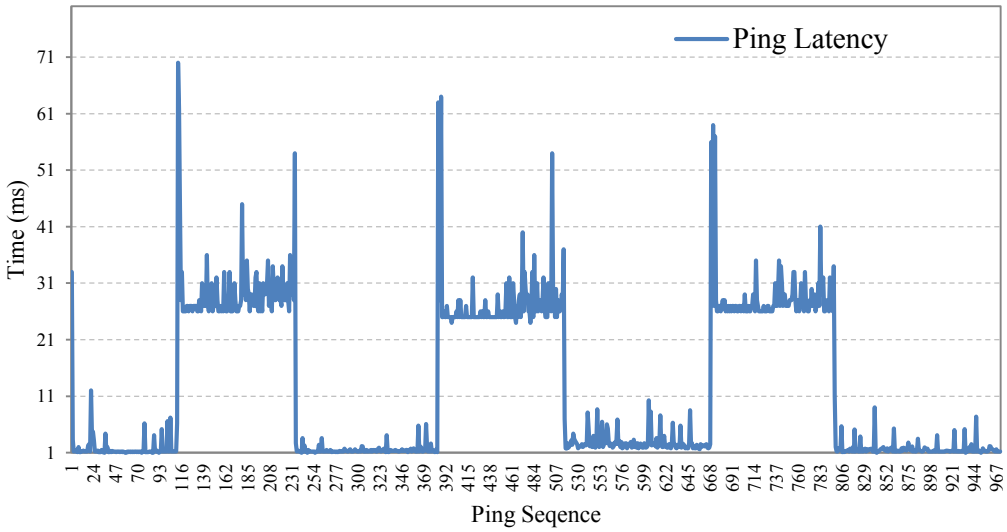
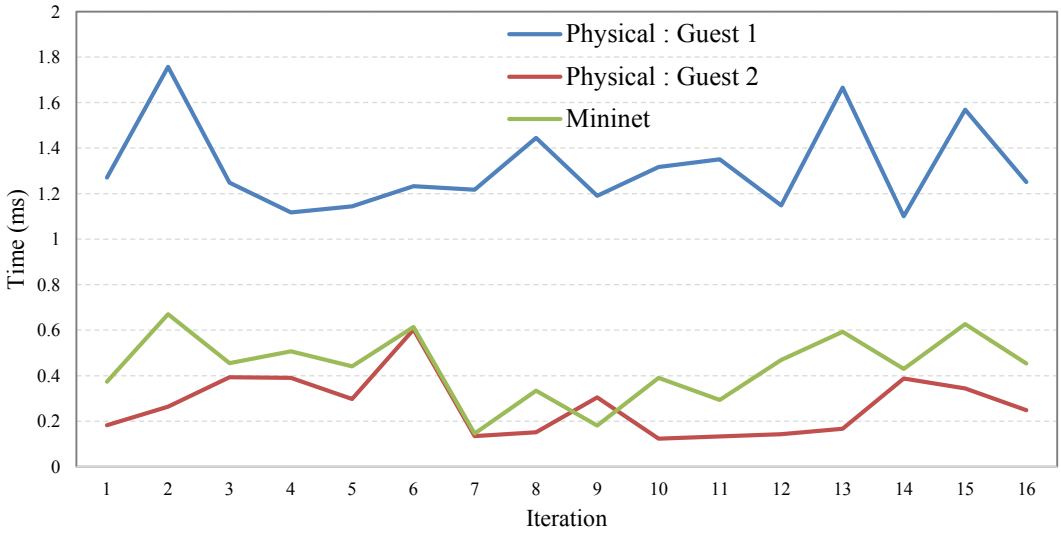


Figure 5.2 Ping Variation of Geographical User Mobility in Mininet

For scenario (c), the user mobility behavior within a region was observed as shown in Figure 5.3. In this test, two physical wireless guests were created, each running Ubuntu 14.04 (physical guest 1) and Windows 7 (Physical guest 2) operating systems. In this scenario, latency values within the local region were ignored, and hence time represent any delay added during for controller processing and other implementation environment based delays. It is clearly visible that, the physical guest 1 accounts higher delay for handover process. The reason for this result is, according to the assumption, the time taken to switch wireless access points manually. However, overall, the handover time is slightly consistent with the user mobility.



In scenario (c), considering user mobility across geographical distributed wireless, the ping latency variation as perceived by the mobile device was measured. In each case, VM is migrated to the nearest datacenter without the involvement of the optimizer. Figure 5.2 shows ICMP ping results in Mininet emulation environment when WAN link latency are set to 40ms and local network latency is set to 1ms. Similarly, Figure 5.4, shows the user

mobility between two regions. Here, a scenario of having a lower latency management network that interconnects wireless networks and datacenter networks was considered to carry control information with VM migration traffic. According to the figure, instantaneous peaks of latency correspond to a controller flow rule generation and update time.

During the ping latency measurements, a time interval of 200ms was set between two consecutive ping requests. Therefore, based on the above experimental results, it can be concluded that, seamless mobility among regions is possible during mobile handover and VM migration is conducted across subnets.

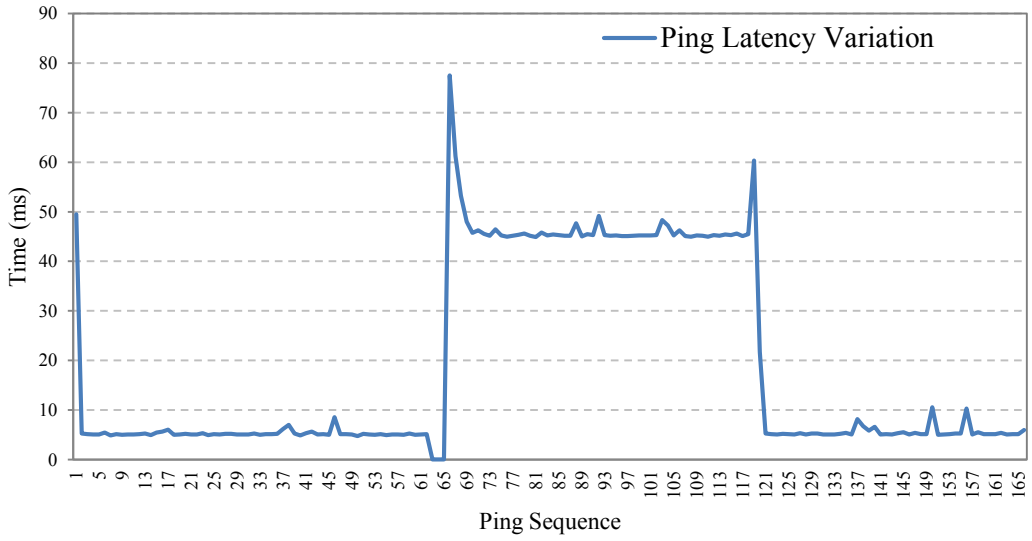
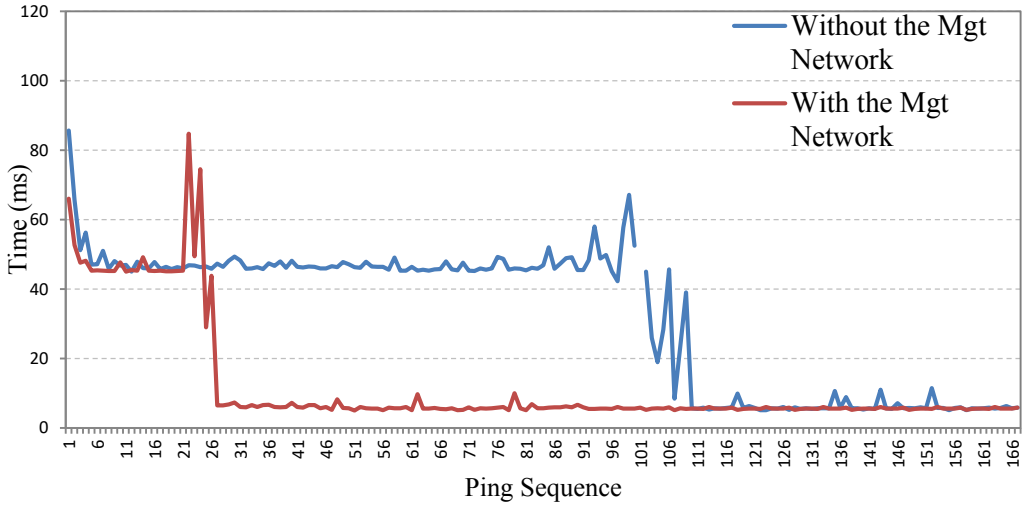
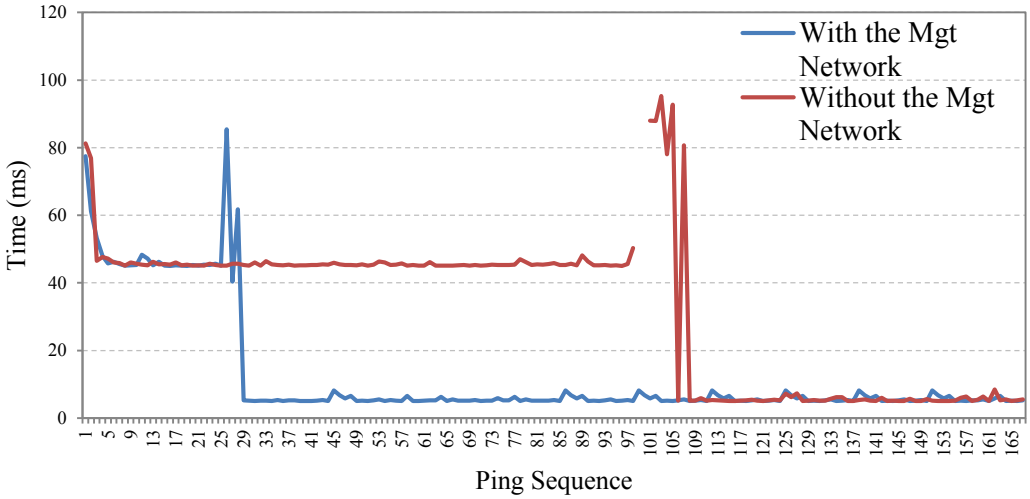


Figure 5.4 Ping Latency Variation During User Mobility in the Physical Testbed

Additionally, when the management network is used with a lower latency, it was observed that, VM migration takes a short time compared to the scenario with there is no management network. One of the observations of the results is a loss of ping packets after VM migration completion. The reason for this behaviour is the VM migration downtime, and as well as migration completion detection delay times of the controller.



(a)



(b)

Figure 5.5 Ping Latency Variation (A) Mininet and (B) Physical Test Environment with A Management

As a result, migration completion detection and new flow rule installation add a considerable delay from switches to the centralized controller due the wide area network latency. Within this time interval, existing flow rules forward ping packets to previous locations of the VM. The smaller the time stamp between consequent ping requests larger the observed packet drops. To avoid such packet losses as a result of network layer

connectivity failures, it is required to find the exact time of VM migration completion in order to install flow rules on time to forward packets to new location.

The VM migration time mainly depends on the workload carried out by the VM at the time of live migration [14]. This defines the page dirty rate of the VM. Therefore, the VM migration completion time was analyzed based on the test scenarios as shown in table 1.

A time overhead was not added on the data network during the VM migration completion time when the management network was used. Also, it provides higher end-to-end bandwidth and lower latency for the migration purpose. It yields a smaller migration time compared to scenario when the data network was used. However, the more delay was added in the wide area network links, the more time it was taken to complete VM migration. Therefore, Figure 5.6 shows time taken for 6 experimental test results for VM migration time on different load conditions on the VM.

Table 5.1 VM migration completion time in Physical testbed

Test Case for VM live migration	Average Migration Completion Time (s)	
	With the management network	Without the management network
ICMP Ping Test to VM	10.7	36.7
Video Streaming VM with MediaTomb Server [78]	11.5	41.2
Downloading a file from FTP server	23.0	74.2

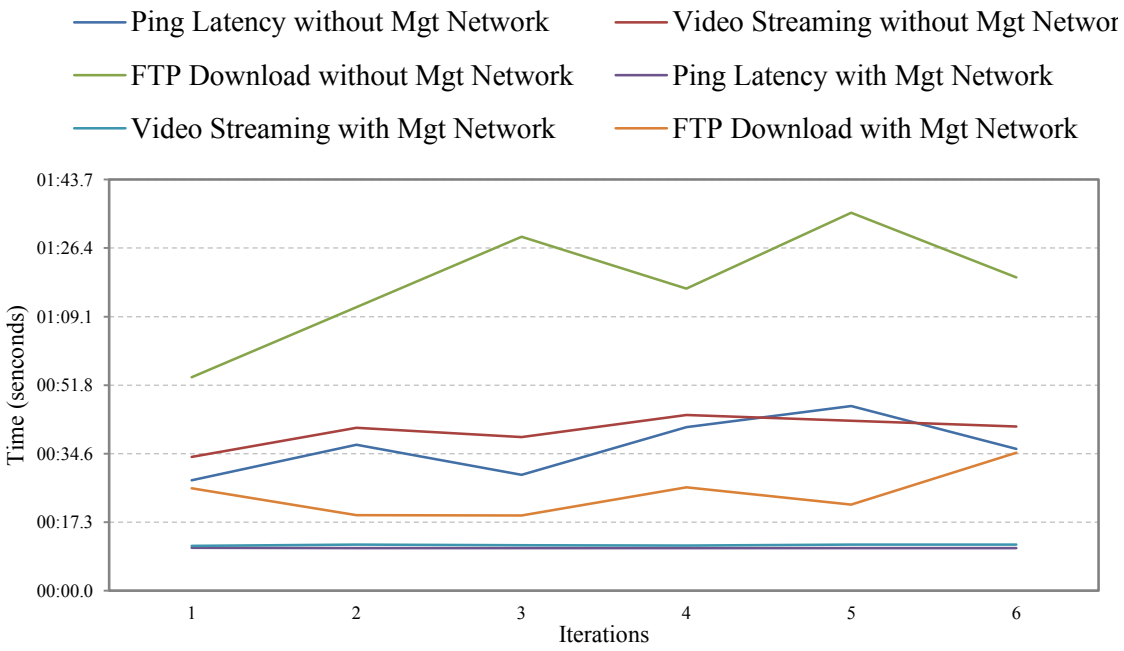


Figure 5.6 VM Migration Completion Time Variation on Different Application Scenarios

5.4 Summary

In this chapter, prepared physical and emulated test environments were evaluated. The outcome of the proposed algorithm was observed, and also the test results were plotted in comparison to the optimal and non-optimal resource relocation techniques. It is also noted that, inter-subnet seamless user migration can be successfully employed with SDN-based wireless networks. By detecting the VM migration completion time, the packet drop was minimized using the API provided by Libvirt. As a result, inter-regional cloud service relocation was carried out with minimal disruption. However, by using a higher latency network to transfer control information, a short period of service inconsistency was observed. It was also noted that, when the user data network is used to migrate a VM, it requires a longer time due to the latency value. As a result, during inter-regional user

mobility, the optimization algorithm is required to add the WAN-delay that affects the live migration procedure.

Finally, the VM migration performance was measure for three different application scenarios. A video streaming service with MediaTomb server was placed in the VM, in order to measure VM migration performance while video streaming is carried out as an application. Similarly, an FTP server was placed in the VM, and subsequently the migration time was measure during a file transfer from the VM to the user device. As a result, it was observed that, the SDN-based regional cloud service provides a seamless end-to-end connectivity during user mobility and a VM live migration events.

6 Conclusion and Future Work

6.1 Conclusion

Mobile cloud computing has become a widely popular research focus within the research community. It is viewed as a proven technology to provide computing resources for mobile devices in the cloud. Although, there are a large number of application level services provided over the cloud, a promising SDN-based QoS-aware approach for allocating computing infrastructure for mobile devices has not yet been extensively explored.

IaaS for mobile computing provides computing resources in closer proximity clouds with a lower access delay and a higher bandwidth. When resources are allocated in the cloud, it is required to maintain the user SLA disregarding the user location and mobility pattern. As a result, a mobility-aware cloud service control mechanism is required to optimize the cloud service with the goal of maintaining SLA and providing minimal service provisioning cost based on current user location. In this research work, a distributed cloud service has been proposed for IaaS mobile clouds and SDN-based networks were used for user and cloud resource mobility management based on the optimization decision.

Initially, the general definition of cloud computing together was provided with an introduction to existing clouds technologies. Then, a comprehensive survey of mobile cloud computing approaches was presented by examining their placement, proximity and also cross subnet cloud resource relocation. Also, the challenges with traditional network architecture are summarized along with a motivation for software defined networks. The benefits of software defined networks for datacenter network management are presented along with an introduction to OpenFlow protocol. In addition, attention is given to existing

approaches for handling mobility across SDN-based wireless networks with the goal of seamlessly managing end-to-end connectivity with the cloud.

Focus is directed to the distributed regional datacenter concept for IaaS mobile clouds which respect user SLA for mobile users. The benefits of SDN in datacenter network management have been leveraged. Also, a seamless resource relocation mechanism to the regional cloud was introduced to orchestrate cloud resources among regional datacenters based on an optimum decision made by the controller. The cloud service controller was developed as a logically centralized controller managing distributed regional datacenters. The cloud service optimization algorithm considers the user location and allocates resources matching user SLA. When user mobility is observed across regions, the optimizer generates an optimum cloud resource relocation decision. Most importantly, the mobile cloud service framework was introduced to support seamless user mobility across wireless networks without changing their network identity and thereby maintaining a transparent cloud access.

By observing the resource relocation procedure, it was noted in this research that the user SLA is maintained with user mobility across regions. Additionally, it is evident that, the optimizer avoids irrelevant migration decision compared to the always-follow scenario, and thereby reduces the service operational cost for the cloud service provider. Based on the experimental results, from this research, it can be concluded that the control message distribution delay over the WAN highly affects the optimal cloud service relocation process and requires a proactive network state update mechanism to avoid packet drops. Finally, it has been shown that, mobility-aware QoS and service cost optimization for mobile computing can be achieved with the proposed RDC-based IaaS clouds.

6.2 Future Work

This research work on mobile cloud computing has shown a number of possible directions for extension. These include the followings:

6.2.1 Extending the Cloud Service Optimization Procedure

When a single user moves from one region to the other, the mobility management entity within the controller gets updated with regard to the user location change. This, in turn, generates an optimization procedure to verify the user SLA has been met at the current location. When all the parameters that are required for optimization are collected, the resource optimization module at the controller finds the best location to place user VM respecting user SLA, as well as optimum operational cost. Subsequently, the controller processes individual user relocation-based QoS optimization requests, and individually generates outputs individually. Then the output passes to the local virtual infrastructure manager module at source and destination RDCs. Since the resource optimizer module is located in the mobile cloud service controller, which is a globally centralized controller, all the optimization requests received at this module must be processed individually. Due to the centralized nature of this module, it causes the optimizers to become overloaded with many requests. To avoid this nature, several variations in terms of architecture and the optimization logic can be proposed.

One of the approaches is to consider a batch of user request optimization. When considering a batch of mobile users, there are multiple possibilities for forming a batch that is being considered for processing together. The followings are a few examples that can be considered:

1. Set of mobile users moved to the same destination region
2. Set of users based on their originating region
3. Set of users belonging to a particular mobile cellular network

Among many other possibilities, consideration is mainly given to the above scenarios to form a batch of users. In case 1, by looking at each individual region, the optimizer can find the best placement considering nearby RDCs closer to the corresponding region. In case 2, a set of users originated from a region can be considered for the optimization. Here, there is a possibility that the corresponding cloud resources of these users are located in their optimum DCs rather than the own RDC. Therefore, these scattered cloud resources must be considered by accessing their originating datacenters. However, generally, the mobile user's destinations are also scattered, and the optimizer needs to consider a large number of source and destination RDCs for calculation. In the case 3 scenario, consideration is given to the network domain that users belong to. Since this kind of grouping is also similar to the case 2 scenario other than that they are on the same mobile network, the optimizer must talk to the scattered RDCs and collect the information regarding optimization. Therefore, the optimization algorithm can be extended to avoid bottlenecks within the preliminary resource optimizer.

6.2.2 Distributed Network Controller Platform for Mobility Management

When further analyzing the bottlenecks in the basic RDC based mobile cloud service, it is observed that the distribution of control messages among distributed controllers creates consistency maintenance issues. This can be considered as a major challenge in the distributed environment intuitively and hence cannot be completely avoided. In the proposed

architecture, it was noted that, after endpoint mobility, if the inter-controller link delay is higher, these controller messages will not reach the other controllers on time. This creates a window of packet loss, where OpenFlow switches at the data plane forward any incoming traffic flow to the earlier locations of the end points. Therefore, cloud architecture's performance must be investigated using centralization of distributed controllers that can minimize the consistency issue.

6.2.3 Integration with Other Mobile Cloud Approaches

In this thesis's research, it was observed that when planning a regional cloud to meet the lower network latency requirements, a physically geographical region must be identified. However, there is a possibility to introducing IaaS services with far lesser latency requirements, which in turn leads to shrinking the region in order to minimize a possible number of hops reaching the cloud. From a practical perspective, a regional cloud with a larger number of distributed datacenters triggers frequent optimizations requests with the mobility of users and resulting in higher CAPEX and OPEX. However, there are closer proximity clouds available to provide resources to mobile devices. Therefore, mobile cloud solutions such as cloudlets can be integrated to the regional cloud service.

A future extension to the regional cloud service is to integrate cloudlets within a local region with the goal of serving mobile users with very lower network access latency. A regional datacenter for a set of cloudlets can be planned within the region and as a result, load balancing among cloudlets will be handled by the regional datacenter. When user mobility across cloudlets occurs, the regional datacenter handles the local mobility. Furthermore, the tasks which are not able to be carried out in cloudlets will be forwarded to the regional

datacenter. Therefore, it was hoped the regional cloud can be extended by integrating cloudlets as a hierarchical mobile cloud service.

References

[1] Statista, Share of mobile Internet traffic in global regions Statistic. [Online] Available: www.statista.com/statistics/306528/share-of-mobile-Internet-tra_c-in-global-regions/

[2] Chaffey, Dave. Insights from KPCB US and global Internet trends 2015 report. [Online] Available: www.smartinsights.com/Internet-marketing-statistics/insights-from-kpcb-us-and-global-Internet-trends-2015 -report

[3] Cisco, Fixed and Mobile IP Network Traffic Forecast – VNI. [Online] Available: www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni

[4] Vallina-Rodriguez, Narseo, and Jon Crowcroft. "Energy management techniques in modern mobile handsets." Communications Surveys & Tutorials, IEEE 15, no. 1 (2013): 179-198.

[5] Silva, Francisco Airton, Paulo Maciel, Eder Quesado, Germano Zaicaner, Matheus Dornelas, and Bruno Silva. "Benchmark Applications Used in Mobil

e Cloud Computing: A Systematic Mapping Study."

[6] Ahmed, Ejaz, Abdullah Gani, Mehdi Sookhak, Siti Hafizah Ab Hamid, and Feng Xia. "Application optimization in mobile cloud computing: Motivation, taxonomies, and open challenges." Journal of Network and Computer Applications 52 (2015): 52-68.

[7] Wang, Yating, Ray Chen, and Ding-Chau Wang. "A survey of mobile cloud computing applications: Perspectives and challenges." Wireless Personal Communications 80, no. 4 (2015): 1607-1623.

[8] Satyanarayanan, Mahadev, Paramvir Bahl, Ramón Caceres, and Nigel Davies. "The case for vm-based cloudlets in mobile computing." Pervasive Computing, IEEE 8, no. 4 (2009): 14-23.

[9] Beck, Michael Till, Martin Werner, Sebastian Feld, and S. Schimper. "Mobile edge computing: A taxonomy." In Proc. of the Sixth International Conference on Advances in Future Internet. 2014.

[10] Ahmed, Arif, and Ejaz Ahmed. "A Survey on Mobile Edge Computing.", Accepted by 10th IEEE International Conference of Intelligent Systems and Control (ISCO 2016)

[11] Yen, Tseng-Chang, and Chi-Sheng Su. "An SDN-based cloud computing architecture and its mathematical model." In Information Science, Electronics and Electrical Engineering (ISEEE), 2014 International Conference on, vol. 3, pp. 1728-1731. IEEE, 2014.

[12] Jain, Raj, and Sudipta Paul. "Network virtualization and software defined networking for cloud computing: a survey." Communications Magazine, IEEE 51, no. 11 (2013): 24-31.

[13] Pentikousis, Kostas, Yan Wang, and Weihua Hu. "Mobileflow: Toward software-defined mobile networks." Communications Magazine, IEEE 51, no. 7 (2013): 44-53.

[14] Bifulco, Roberto, Marcus Brunner, Roberto Canonico, Peer Hasselmeyer, and Faisal Mir. "Scalability of a mobile cloud management system." In Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 17-22. ACM, 2012.

[15] Mann, Vijay, Anilkumar Vishnoi, Kalapriya Kannan, and Shivkumar Kalyanaraman. "CrossRoads: Seamless VM mobility across data centers through software defined

networking." In Network Operations and Management Symposium (NOMS), 2012 IEEE, pp. 88-96. IEEE, 2012

[16] Satyanarayanan, Mahadev, Paramvir Bahl, Ramón Caceres, and Nigel Davies. "The case for vm-based cloudlets in mobile computing." Pervasive Computing, IEEE 8, no. 4 (2009): 14-23.

[17] ETSI, MEC. "Mobile-Edge Computing." Introductory Technical White Paper, September (2014).

[18] Amarasinghe H. , Karmouch A., "SDN-based Framework for Infrastructure as a Service Clouds", Accepted by IEEE Cloud 2016, June 27 - July 2, 2016, San Francisco, USA

[19] Taleb, Tarik, Konstantinos Samdanis, and Fethi Filali. "Towards supporting highly mobile nodes in decentralized mobile operator networks." In Communications (ICC), 2012 IEEE International Conference on, pp. 5398-5402. IEEE, 2012.

[20] Open Networking Fundation. "Software-defined networking: The new norm for networks." ONF White Paper (2012).

[21] Bifulco, Roberto, and Roberto Canonico. "Analysis of the handover procedure in Follow-Me Cloud." In CLOUDNET, pp. 185-187. 2012.

[22] Satyanarayanan, Mahadev. "Fundamental challenges in mobile computing." In Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing, pp. 1-7. ACM, 1996.

[23] Qi, Han, and Abdullah Gani. "Research on mobile cloud computing: Review, trend and perspectives." In Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on, pp. 195-202. ieee, 2012.

[24] Fernando, Niroshinie, Seng W. Loke, and Wenny Rahayu. "Mobile cloud computing: A survey." Future Generation Computer Systems 29, no. 1 (2013): 84-106.

[25] Sharma, Amit K., and Priyanka Soni. "Mobile Cloud Computing (MCC): Open Research Issues." International Journal of Innovations in Engineering and Technology (IJIET) 2, no. 1 (2013).

[26] Cloud computing versus grid computing.

[Online] Available: www.ibm.com/developerworks/library/wa-cloudgrid

[27] Chen, Xu, Lei Jiao, Wenzhong Li, and Xiaoming Fu. "Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing." (2015).

[28] Chen, Min, Yixue Hao, Yong Li, Chin-Feng Lai, and Di Wu. "On the computation offloading at ad hoc cloudlet: architecture and service modes." Communications Magazine, IEEE 53, no. 6 (2015): 18-24.

[29] Hamdaqa, Mohammad, Mohamed M. Sabri, Akshay Singh, and Ladan Tahvildari. "Hadoop: MapReduce for ad-hoc cloud computing." In Proceedings of the 25th Annual International Conference on Computer Science and Software Engineering, pp. 26-34. IBM Corp., 2015.

[30] Gary Andrew McGilvary, Adam Barker, Malcolm Atkinson, “Ad hoc Cloud Computing: From Concept to Realization”, IEEE Cloud 2015 "Ad hoc Cloud Computing"

[31] Satyanarayanan, Mahadev, Grace Lewis, Edwin Morris, Soumya Simanta, Jeff Boleng, and Kiryong Ha. "The role of cloudlets in hostile environments." Pervasive Computing, IEEE 12, no. 4 (2013): 40-49.

[32] Ha, Kiryong, Padmanabhan Pillai, Grace Lewis, Soumya Simanta, Sarah Clinch, Nikki Davies, and Mahadev Satyanarayanan. "The impact of mobile multimedia applications on data center consolidation." In Cloud Engineering (IC2E), 2013 IEEE International Conference on, pp. 166-176. IEEE, 2013.

[33] Ceselli, Alberto, Marco Premoli, and Stefano Secci. "Cloudlet network design optimization." In IFIP Networking Conference (IFIP Networking), 2015, pp. 1-9. IEEE, 2015.

[34] Jo, Changyeon, Erik Gustafsson, Jeongseok Son, and Bernhard Egger. "Efficient live migration of virtual machines using shared storage." In ACM Sigplan Notices, vol. 48, no. 7, pp.41-50. ACM, 2013.

[35] Jararweh, Yaser, Loai Tawalbeh, Fadi Ababneh, and Fahd Dosari. "Resource efficient mobile computing using cloudlet infrastructure." In Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on, pp. 373-377. IEEE, 2013.

[36] Gai, Keke, Meikang Qiu, Hui Zhao, Lixin Tao, and Ziliang Zong. "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing." Journal of Network and Computer Applications 59 (2016): 46-54.

[37] Clinch, Sarah, Jan Harkes, Adrian Friday, Nigel Davies, and Mahadev Satyanarayanan. "How close is close enough? Understanding the role of cloudlets in supporting display

appropriation by mobile users." In Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on, pp. 122-127. IEEE, 2012.

[38] esehaye, Debessay, Yunlong Gao, Klara Nahrstedt, and Guijun Wang. "Impact of cloudlets on interactive mobile cloud applications." In Enterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International, pp. 123-132. IEEE, 2012.

[39] Gao, Ying, Wenlu Hu, Kiryong Ha, Brandon Amos, Padmanabhan Pillai, and Mahadev Satyanarayanan. "Are Cloudlets Necessary?" (2015).

[40] Roman, Rodrigo, Javier Lopez, and Masahiro Mambo. "Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges." arXiv preprint arXiv:1602.00484 (2016).

[41] Smarter wireless networks.

[Online] Available: www-935.ibm.com/services/multimedia/Smarter_wireless_networks.pdf

[42] Garcia Lopez, Pedro, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. "Edge-centric Computing: Vision and Challenges." ACM SIGCOMM Computer Communication Review 45, no. 5 (2015): 37-42.

[43] Choy, Sharon, Bernard Wong, Gwendal Simon, and Catherine Rosenberg. "A hybrid edge-cloud architecture for reducing on-demand gaming latency." Multimedia Systems 20, no. 5 (2014): 503-519.

[44] MOS - Mean Opinion Score for VoIP Testing

Available: http://www.voipmechanic.com/mos-mean-opinion-score.htm

[45] Assasa, Hany, Srinivasa Vinay Yadhav, and Lars Westberg. "Service Mobility in Mobile Networks." In Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on, pp. 397-404. IEEE, 2015.

[46] Savolainen, Petri, Sumi Helal, Jukka Reitmaa, Kai Kuikkaniemi, Giulio Jacucci, Mikko Rinne, Marko Turpeinen, and Sasu Tarkoma. "Spaceify: A client-edge-server ecosystem for mobile computing in smart spaces." In Proceedings of the 19th annual international conference on Mobile computing & networking, pp. 211-214. ACM, 2013.

[47] Dinesh C. Verma. Policy-Based Networking. New Rider Publishing Indianapolis, IN 2000

[48] Cisco, Quality of Service (QoS). [Online] Available: www.cisco.com/c/en/us/products/ios-nx-os-software/quality-of-service-qos/index.html

[49] Yi, Shanhe, Cheng Li, and Qun Li. "A survey of fog computing: concepts, applications and issues." In Proceedings of the 2015 Workshop on Mobile Big Data, pp. 37-42. ACM, 2015.

[50] 3GPP TR 23.829, "TS Group Services and System Aspects; Local IP Access and Selected IP Traffic offload", May 2010

[51] Taleb, Tarik, and Adlen Ksentini. "Follow me cloud: interworking federated clouds and distributed mobile networks." Network, IEEE 27, no. 5 (2013): 12-19.

[52] Chaparro-Marroquín, Juan C., and Roberto Bustamante-Miller. "Connection Management System based on MIH and MIP over NS-3."

[53] Wang, Shiqiang, Rahul Urgaonkar, Murtaza Zafer, Ting He, Kevin Chan, and Kin K. Leung. "Dynamic service migration in mobile edge-clouds." In IFIP Networking Conference (IFIP Networking), 2015, pp. 1-9. IEEE, 2015.

[54] Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011): 20-23.

[55] Zhang, Qi, Lu Cheng, and Raouf Boutaba. "Cloud computing: state-of-the-art and research challenges." Journal of Internet services and applications 1, no. 1 (2010): 7-18.

[56] Church, Kenneth, Albert G. Greenberg, and James R. Hamilton. "On Delivering Embarrassingly Distributed Cloud Services." In HotNets, pp. 55-60. 2008.

[57] Valancius, Vytautas, Nikolaos Laoutaris, Laurent Massoulié, Christophe Diot, and Pablo Rodriguez. "Greening the Internet with nano data centers." In Proceedings of the 5th international conference on Emerging networking experiments and technologies, pp. 37-48. ACM, 2009.

[58] Chandra, Aniruddha, Jon Weissman, and Benjamin Heintz. "Decentralized edge clouds." Internet Computing, IEEE 17, no. 5 (2013): 70-73.

[59] ONF Solution Brief. "OpenFlow™-Enabled Mobile and Wireless Networks." white paper (2013).

[60] McKeown, Nick, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. "OpenFlow: enabling innovation in campus networks." ACM SIGCOMM Computer Communication Review 38, no. 2 (2008): 69-74.

[61] OpenFlow Switch Specification of OpenFlow protocol 1.0. [Online] Available: www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf

[62] OpenFlow Switch Specification of OpenFlow protocol 1.3.2.

[Online] Available: www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.2.pdf

[63] Duong, Thuy-Van T., and Dinh Que Tran. "Mobility prediction based on collective movement behaviors in public WLANs." In Science and Information Conference (SAI), 2015, pp. 1003-1010. IEEE, 2015.

[64] Liu, Haikun, Hai Jin, Cheng-Zhong Xu, and Xiaofei Liao. "Performance and energy modeling for live migration of virtual machines." Cluster computing 16, no. 2 (2013): 249-264.

[65] Rallapalli, Swati, Wei Dong, Gene Moo Lee, Yi-Chao Chen, and Lili Qiu. "Analysis and applications of smartphone user mobility." In Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on, pp. 235-240. IEEE, 2013.

[66] R. Sherwood et al. "Flowvisor: A network virtualization layer." OpenFlow Switch Consortium, Tech. Rep (2009).

[67] Drutskoy, D., Keller, E., and Rexford, J. (2013). Scalable network virtualization in software-defined networks. Internet Computing, IEEE, 17(2), 20-27.

[68] Libvirt virtualization API. [Online] Available: www.libvirt.org

[69] KVM Hypervisor. [Online] Available: www.linux-kvm.org

[70] Mininet: An Instant Virtual Network on your Laptop (or other PC).

[Online] Available: mininet.org

[71] Production Quality, Multilayer Open Virtual Switch. [Online] Available: openvswitch.org

[72] The Linux Documentation Project - Introduction to Linux Traffic Control

[Online] Available: www.tldp.org/HOWTO/Traffic-Control-HOWTO/intro.html

[73] Taleb, Tarik, Peer Hasselmeyer, and Faisal Ghias Mir. "Follow-Me Cloud: An OpenFlow-based Implementation." In Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, pp. 240-245. IEEE, 2013.

[74] hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator. [Online] Available: www. w1.fi/hostapd/

[75] Liu, Jiaqiang, Yong Li, and Depeng Jin. "SDN-based live VM migration across datacenters." In ACM SIGCOMM Computer Communication Review, vol. 44, no. 4, pp. 583-584. ACM, 2014.

[76] Wood, Timothy, K. K. Ramakrishnan, Prashant Shenoy, and Jacobus Van der Merwe. "CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines." In ACM Sigplan Notices, vol. 46, no. 7, pp. 121-132. ACM, 2011

[77] Kondo, Toshiaki, Reiji Aibara, Kazufumi Suga, and Kumiko Maeda. "A mobility management system for the global live migration of virtual machine across multiple sites." In Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International, pp. 73-77. IEEE, 2014

[78] Mediatomb- Open source (GPL) UPnP MediaServer

[Online] Available: www.mediatomb.cc