Cognitive-Aware Network Virtualization Hypervisor for Efficient Resource Provisioning in Software Defined Cloud Networks

A thesis submitted in partial fulfilment of the requirements for the degree

of Doctor of Philosophy

by

Andrews Offei Nyanteh

Department of Electronic and Computer Engineering

College of Engineering, Design and Physical Sciences

Brunel University London

United Kingdom

December 2019

Abstract

Integration of different technologies forms an integral part of modern network engineering and 5G technology deployment. Although Software Defined Networking (SDN) and Network Functions Virtualization (NFV) function well independently, integrating these two technologies present the cooperate advantages to service providers and service users.

Operations of cloud computing technologies have been enhanced with the advent of SDN and NFV for efficient solutions deployment and infrastructure management in Software Defined Cloud Datacentre Networks (SDCDCN) where dynamic controllability is indispensable for elastic service provision. The provisioning of joint compute and network resources enabled by SDCN is essential to enforce reasonable Service Level Agreements (SLAs) stating the Quality of Service (QoS) while saving energy consumption and resource wastage. This thesis presents a Cognitive- Aware Network virtualization Hypervisor which was developed from merging the programmable dynamic network control attributes of SDN and the network slicing attributes of NFV to provision joint compute and network resources in SDCDCN for QoS fulfilment and energy efficiency. It focuses on the techniques for allocating Virtual Network Requests on physical hosts and switches considering SLA, QoS, and energy efficiency aspects.

The thesis advances the state-of the-art with the following key contributions: A modelling and simulation environment for Software Defined Cloud Datacentre Networks abstracting functionalities and behaviours of virtual and physical network resources. The second is a novel dynamic overbooking algorithm for energy efficiency and SLA enforcement with the migration of virtual machines and network flows. Finally, a performance-aware intelligent overbooking for predicting network resource usage and performance for the next defined time interval considering multiple performance indexes.

Publications Based on this Research

[1] Nyanteh, O.A., Li, M., Abbod, M., and Al-Raweshidy. H., (2019) 'CloudSimHypervisor: Modelling and Simulation of Software-Defined Cloud Networks', IEEE Access Journal, (Reviewer Feedback and Resubmitted)

[2] Nyanteh, O. A., Li, M., Abbod, M., (2019) 'SLA-Aware and Energy Efficient Intelligent Network Resource Overbooking', IEEE Access Journal, (SUBMITTED)

[3] Nyanteh, O. A., Li, M., Abbod, M., and Buyya, R., (2019) 'Intelligent Analysis and Prediction of User Requirements to Optimize Resource Utilization in Software-Define Cloud Networks', IEEE Access Journal, (SUBMITTED)

Declaration

It is hereby declared that the thesis in focus is the author's own work and is submitted for the first time to the Post Graduate Research Office. The study was originated, composed and reviewed by the mentioned author in the Department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences, Brunel University London, UK. All the information derived from other works has been properly referenced and acknowledged.

Andrews Offei Nyanteh
December, 2019
London, UK

Acknowledgements

The pursuit of a PhD is an incredible sweet and sour journey, mostly sour though. I am genuinely grateful and full of thanks to God I am, for granting life, strength, wisdom and the passion to overcome all the challenges that I encountered during my learning experience and on this journey. I couldn't have come this far without the support of some key people whose help I cannot repay but can only show my endless appreciation. First of these people is my supervisor, Professor Maozhen Li who stepped in and offered me a lifeline at a time when I needed one the most but a time which was unsuitable for any academic to accept a student. Professor Li offered the insightful guidance, continuous support and invaluable advice which I very much needed for a successful complete of my research work. The research advisory team should have comprised of more than one person but for lack of hands, Dr Maysam Abbod to whom I offer my deepest gratitude played the role of the many for the entire duration of my program. He was always present to listen to me and to support with developing my research skills and valuable guide on writing my papers.

I would also like to thank all the past and current members of the Wireless Networking and Communication Centre (WNCC) of Brunel University London for their friendship and support during my PhD. I acknowledge the Ghana Education Trust Fund (GETFUND) and the Ghana High Commission for Great Britain and Northern Ireland, for granting scholarships to pursue my PhD.

Finally, I would like to give my heartfelt appreciation to my family; my mother and my late father who planted the seed for the pursuit of success and greatness in me, my siblings, Yvonne, Nana Ankobeahene (Ebenezer Gyawu-Mante) of the Ahwerase Traditional Area and the Mante Family, Dr. Sampson O. Amoafo and my niece Michelle for their support and encouragement all the way through the time of my studies. And to my friends, Eric Owusu Ankomah, Micheal Sarfo Frimpong, Richard Akalayabah and Alaa Alisawi and his family for their support and earnest prayers from the very beginning.

Thesis Contents

Chapter 1 Introduction 1

1.1 Introduction 1

1.2 Background 2

1.2.1 Cloud Computing 2

1.2.2 Software Define Networks 4

1.2.3 Southbound Interface 4

1.2.4 Northbound Interface 4

1.2.5 Intermediate Controller Plane Interface (I-CPI) 5

1.2.6 The Concept of Virtualization 7

1.2.7 Virtualization in Networking 7

1.3 Motivation 8

1.4 Research Problems and Challenges 9

1.4.1 Research Challenges 10

1.5. Evaluating the Methodology for this Research 10

1.6 Thesis Contributions 11

1.7 Organization of the Thesis 12

Chapter 2 Literature Review 14

2.1 Introduction 14

2.2 Definition of Terms and Concepts 15

2.2.1 From Virtual Machines to Virtual Networks 15

2.2.2 Techniques for Virtualizing Networks 16

2.2.3 Combining Network Virtualization and Software Defined Networking 16

2.2.4 Managing a Physical SDN Network with a Hypervisor 18

2.2.5 Virtual Network Embedding 19

2.2.6 Non- virtualized SDN Vrs Virtualized SDN 19

2.2.7 Network Virtualization Hypervisor 20

2.2.8 Type of Network Virtualization Hypervisor Architecture 21

2.3 Taxonomy of usage of SDN and NFV in Software Defined Cloud Networks 22

2.3.1 Objective 23
2.3.2 Method Scope 25
2.3.3 Target Architecture 26
2.3.4 Resource Configuration 27
2.3.5 Evaluation Method 28
2.4 Current Research Studies on Software Defined Cloud Networks 29
2.4.1 Energy Efficiency in Software Defined Cloud Networks 29
2.5 Performance 35
2.6 Virtualization 42
2.7 Summary 47
Chapter 3 Modelling and Simulating Software-Defined Cloud Networks 48
3.1 Introduction 48
3.2 Related Work 50
3.3 Event-driven Simulation 52
3.4 Software-Defined Cloud Networks Simulation 53
3.5 CloudSimHypervisor 54
3.6 Framework Design 55
3.6.1 The Core Logic of CloudSimHypervisor 57
3.6.2 Network Modules 59
3.6.3 Packet Scheduler 60
3.6.4 Calculating Packet Transmission Time 61
3.6.5 Abstracting User Requests 62
3.7 Validation 62
3.7.1 CloudsimSDN-NFV Setup 63
3.7.2 Testbed Configuration 64
3.7.3 Validation Results 65
3.8 Use Case Evaluation 68
3.8.1 Joint Compute and Network Resource Utilization 68
3.9 Traffic Prioritization 72
3.10 Summary 75

Chapter 4 SLA-Aware and Energy Efficient Intelligent Network Resource Overbooking 76

4.1 Introduction 76
4.2 Related Work 78
4.3 Power Formulation 80
4.3.1 Power Models 80
4.4. Resource Allocation Architecture 84
4.5. Network Resource Overbooking Algorithm 85
4.6. Joint VNR Consolidation and Migration Algorithm with Dynamic Overbooking 87
4.6.1. Consolidation Algorithms 91
4.6.2. Baseline Algorithms 92
The proposed algorithm was compared with the following baseline algorithms 92
4.7. Performance Evaluation 93
4.8. Testbed Configuration 93
4.9. Workload 94
4.10. Migration Policy 96
4.11. Investigating the Impact of Migration Procedures 96
4.12. Investigating the Impact of Intelligent Overbooking Ratio. 99
4.13 Summary 104

Chapter 5 Performance-Aware Intelligent Overbooking Strategy to Enhance Resource Usage in Cloud Datacentre Networks. 105

5.1 Introduction 105
5.2. Related Work 108
5.3. Training of Artificial Neural Networks 110
5.4. Intelligent Autonomous Overbooking System 111
5.5. Performance Evaluation 116
5.6. Testbed Configuration 117
5.7. Workload 118
5.8. Investigating the Impact of Autonomous Overbooking on Datacentre Resource Utilisation 119
5.9. Summary 125

Chapter 6 Conclusion and Future Research 126

6.1 Conclusions and Discussion 126

6.2. Future Directions 128

6.2.1 Supporting Big Data Applications 128

6.2.2 Enhancing Security and Reliability in SDN and Cloud Network Unification 129

References **131**

List of Figures

Figure 1.1. A description of the classification of cloud computing
Figure 1.2 A general Architectural Framework for Software Defined Networking
Figure 1.3. Comparing of SDN and traditional networking.
Figure 1.4 A representation of Network Virtualization
Figure 1.5. Organization of the Thesis
Figure 2.1 Combining Network Virtualization and Software Defined Networking
Figure 2.2. Virtualizing the SDN Network (Perspective of the hypervisor): The inserted hypervisor directly interacts with the physical SDN Networks as well as two virtual SDN controllers.
Figure 2.3 Taxonomy of usage of SDN and NFV in Software Defined Cloud Networks
Figure 2.4: Sub-classification of the literature review.
Figure 3.1. A Software Defined Cloud Networking Architecture
Figure 3.2 CloudSimHypervisor Architecture.
Figure 3.3 Class diagram of CloudSim Hypervisor
Figure 3.4. Experimental setup for validating CloudSimHypervisor.
Figure 3.5. Comparing CloudSimHypervisor with CloudSimSDN-NFV with regards to processing speed and average transmission speed.
Figure 3.6. Results of comparison between CloudSimHypervisor and CloudSimSDN-NFV with regards to average transmission speed.
Figure 3.7. Diagrams showing efficient utilization of compute and network resources with the implementation of the Network Virtualization Hypervisor.
Figure 3.8. Effect of Network Traffic Prioritization.
Figure 4.1 Network Resource Allocation System
Figure 4.2 Demonstration of network events consolidation and Resource Overbooking
Figure .4.3. A Software Defined Cloud Networking Architecture
Figure 4.4. CPU request and usage distribution in a datacentre
Figure 4.5. Bandwidth request and usage distribution in a datacentre
Figure 4.6. Energy Consumption for different migration strategies in SDCDCN

Figure 4.7. Energy consumption for different overbooking strategies in SDCDCN
Figure 4.8. Energy consumption analysis in the entire datacentre network
Figure 4.9. SLA Violation rates for different overbooking strategies
Figure 5.1. Intelligent Autonomous Overbooking System Architecture
Figure.5.2 Topology of the Resource Usage Prediction Artificial Neural Network
Figure 5.3. A Software Defined Cloud Networking Architecture
Figure 5.4 Actual and predicted network resource usage
Figure 5.5. Actual and Predicted IPC Values
Figure 5.6 Google's host compute and network actual and Intelligent Overbooking Ratios for different Performance indexes
Figure 5.7. Google's host compute and network actual and Intelligent Predicted Performance rates for different Performance indexes

List of Tables

Table 2.1 Outlining summary of studies on energy efficiency of Cloud Networks.

Table 2.2. Outlining summary of studies on performance of Cloud Networks.

Table 2.3: Summary of current research for Software Defined Cloud Networks and Functions Virtualization.

Table 2.4: Descriptive Details of Software Defined Cloud Network Projects.

Table 3.1. Notations for class diagram of CloudSimHypervisor.

Table 3.2. Link configuration for the validation experiments.

Table 3.3. VM specification for joint compute and network resource utilization use case.

Table 3.4. Characteristics of network requests based on the model proposed by (Ersoz et al., 2007).

List of Abbreviations

5G	Fifth Generation
ANN	Artificial Neural Networks
A-CPI	Application Controller Plane Interface
AO-M	Application-Oriented Module
API	Application Programming Interfaces
C-SDN	Cloud Software Defined Networking
CP	Control Plane
CPI	Cycle -Per-Instruction
CPU	Central Processing Unit
CSP	Cloud Service Provider
CSV	Comma-Separated Values
D-CPI	Data-Controller Plane Interface
DCN	Datacentre Network
DDoS	Distributed Denial of Service
DFS	Depth First Search
DRAR	Dynamic Resource Allocation Ratio
DVFS	Dynamic Voltage and Frequency Scaling
GETFUND	Ghana Education Trust Fund
HTB	Hierarchical Token Bucket
I-CPI	Intermediate Controller Plane Interface
IaaS	Infrastructure-as-a-Service
IPC	Instruction -Per- Cycle
LB	Load Balancer
LTE	Long Term Evolution
MANO	Management and Orchestration
MPLS	Multiple Protocol Label Switching
MSE	Mean Squared Error
Naas	Network – as-a- Service
NAT	Network Address Translation

NE	Network Element
NFV	Network Functions Virtualization
NFVI	Network Functions Virtualization Infrastructure
NP	Non-deterministic Polynomial
NV	Network virtualization
NVP	Network Virtualization Platform
NVS	Network Virtualization Substrate
OF	OpenFlow
OS	Operating system
PaaS	Platform-as-a-Service
PC	Personal Computer
QoS	Quality of service
RAM	Random Access Memory
RAR	Resource Allocation Ratio
SaaS	Software-as-a-Service
SDCDCN	Software Defined Cloud Datacentre Network
SDCN	Software Defined Cloud Network
SDCN	Software Defined Cloud Networks
SDN	Software Define Networking
SFC	Service Function Chaining
SDCDCN	Software Defined Cloud Datacentre Network
SFQ	Stochastic Fairness Queuing
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SOA	Service Oriented Architecture
T-SDN	Transport Software Defined Networking
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMM	Virtual Machine Monitor
VN	Virtual Network
VNE	Virtual Network Embedding

VNF	Virtualized Network Functions
VNR	Virtual Network Request
VPN	Virtual Private Networking
vSDN	Virtual Software Defined Networking
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WNCC	Wireless Networking and Communication Centre
XMPP	Extensible Messaging and Presence Protocol

Chapter 1 Introduction

1.1 Introduction

The expeditious development of communication technologies and computer networking in the past decade has significantly influenced the services provided by the Internet; a network infrastructure which delivers sectional range of data communication services to a wide spread service delivery medium which supports a wide spectrum of extensive heterogeneous computing applications (Duan, Yan, & Vasilakos, 2012b) . To overcome the challenge of end-to-end quality of service (QoS) provisioning over the current internet and the shortcomings of traditional networks, the Network – as-a- Service (Naas) in software defined networking paradigm approach has been widely adopted in designing and implementing modern networks. NaaS introduces notable strategies with changes in modelling both network architecture and service delivery models (Duan et al., 2012b). These changes can be categorised into two major concepts: (i) network virtualization which decouples network functions that are related to service provisioning from their network infrastructures responsible for transporting and processing data (ii) service-oriented networking which encapsulates network functionalities and resources into entities that can be displayed and used through abstract interfaces.

The computing and networking community have been revolutionized with the emergence of Software-Defined Networking (SDN). Abstracting the network control logic from the data forwarding plane is an innovation which makes provision for network programmability. With a centralized view of the entire network, the SDN controller which is reconfigurable in runtime is able to administer to all segments of the network with a centralized programmable control logic for each independent scenario. This gives SDN the capability to overcome different Quality of Service (QoS) demands in runtime (Son, 2018). Cloud providers utilize virtualization technologies to separate resources from computers, storage and networking to facilitate elastic service delivery to cloud users (tenants). Cloud providers often operate tens of thousands of compute servers which are connected through thousands of switches and as result it is essential for them to have efficient and advanced procedures for provisioning compute and network resource. This objective cannot be achieved with the

use of traditional networks because each of its network switches has an independent control logic and the behaviour of the network is not uniformly controlled. This approach raises challenges with network manageability and administration which has an adverse impact on network performance. As a result, traditional network systems are not suitable for use in cloud datacentres which require an elastic setup to be able to dynamically provision large scale networking and computing resources which are capable of adapting to variation of workloads and requests from various users or tenants, an innovation that SDN has introduced in cloud networks.

To guarantee Service Level Agreement (SLA) for cloud users, it is essential for cloud service providers to have efficient resource provisioning for compute and network resources to ensure multiple reliability and QoS requirements. A case in point is IoT applications for real-time disaster management, medical software for cyber surgery and deadline constrained scientific applications which are QoS-critical applications require stringent and thorough policies in cloud resource management compared to other applications which do not fall in the same category (Son, 2018).

Having to manage workloads and requests from cloud users with different QoS requirements implies cloud service providers be able to effectively provision resources to adequately meet the demands of the various applications request as well as reducing energy consumption which contributes to minimizing operational expenditure. Other factors which could contribute to maximising profits, while reducing operational cost by cloud service providers in their bid to fulfilling QoS requirements include SLA violations and reliability of service delivery. Hence efficient compute and network resource provisioning which guarantees less user SLA violations, QoS fulfilment, low energy consumption and low operational cost is required by the computing and networking community.

1.2 Background

1.2.1 Cloud Computing

The emergence of cloud computing enabled on demand provisioning of networking and computing resources and present their services in a pay- as-you-go model (Son & Buyya,

2018b). Cloud computing, provides elastic and scalable provisioning of multiple computing and network resources without the need for physical infrastructure (Son, 2013).

Cloud providers deploy several heterogeneous datacentres in multiple geographical domains to enable provision of elastic and subscription-based services to their clients. Cloud computing is implemented in three main service models, which are Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) (Buyya, Calheiros, Son, Dastjerdi, & Yoon, 2014). Figure 1.1 shows the classification of cloud computing. SaaS presents cloud users with applications (software) which are ready for use. These applications include cloud-based scheduler services, social network services, email services. Cloud users are not required to install any of the SaaS applications on their local infrastructures or acquire licensing for their use since such expenses are included in the subscription charges which could be hourly, monthly or annually. The PaaS model provides a layered platform where users can execute their applications with ease. IaaS provides cloud users with virtual machines, servers and other cloud related infrastructure for deploying customized solutions which could SaaS or PaaS related. IaaS is the most fundamental service implementation in cloud computing (Son, 2018). Network- as- a- Service is the model adopted by cloud computing which employs Service Oriented Architecture (SOA) to enable network infrastructure to be implemented and utilized as a network service (Duan, Yan, & Vasilakos, 2012a).

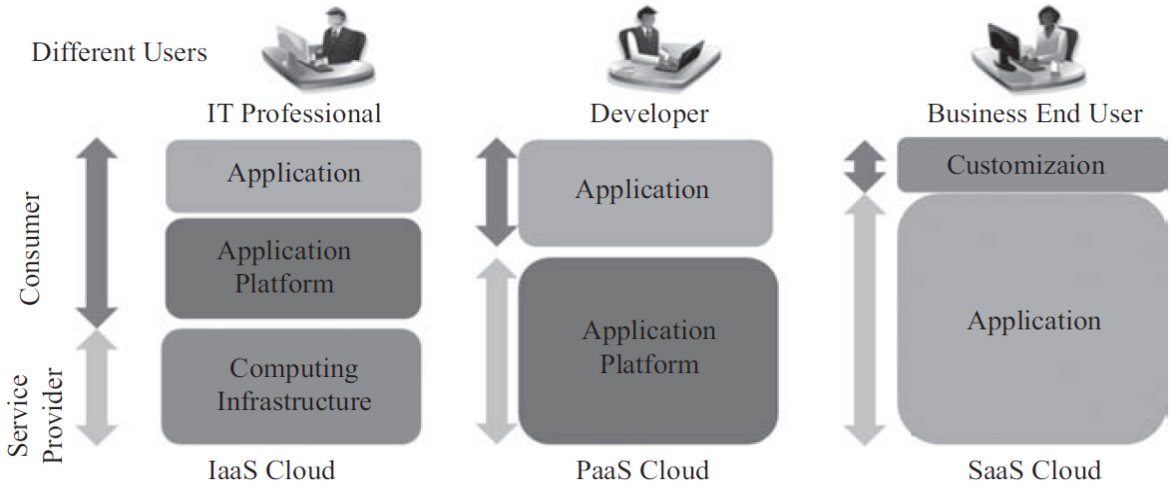


Figure. 1.1. A description of the classification of cloud computing according to (Duan, Yan, & Vasilakos, 2012a)

1.2.2 Software Defined Networks

Software defined networking (SDN) has emerged as potent networking technology which makes control of communications network flexible. SDN has the capability to separate the control function from the forwarding function of the networking devices in the physical infrastructure and defines open and programmable interface with the use of OpenFlow (OF) protocol (Kreutz et al., 2015). SDN presents a programmable centralized network control unit known as controller which gives a global view of the entire network (Son & Buyya, 2018b). This enables dynamic configuration of the network control logic and management, allowing components in the infrastructure plane (physical network) to adjust quickly to change in requests and workload requirements on-demand (Buyya et al., 2014), (Kreutz et al., 2015). The centralized network control of SDN operates in collaboration with some application programming interfaces (APIs) to enable communication in the entire network.

1.2.3 Southbound Interface

This is the application programming interface which enables communication and a logical connection between the controller and the infrastructure plane (Data Forwarding Plane). OpenFlow and other protocols such as extensible messaging and presence protocol (XMPP) and network configuration protocols which make this communication possible. This interface is also referred to as Data-controller plane interface (D-CPI) or the OF control channel or the control plane channel

1.2.4 Northbound Interface

This is the application programming interface which enables communication between the centralized controller and the application plane. This interface enables workloads and applications to access functions in the control plane irrespective of the specification of the underlying network components. This interface, also known as the application controller plane interface (A-CPI), is a medium where novel network applications which manages traffic steering and executes network decisions are sometimes developed by combing with functions which are executed in the controllers of software defined networks. These functions include firewall, load balancers and network monitors.

1.2.5 Intermediate Controller Plane Interface (I-CPI)

This is the application programming interface enables the SDN controller to inter-connect with controllers in other network domains as shown in Figure 1.2. This interface works together with the east-bound API which combines with the network control of network domains which do not execute SDN for instance it interfaces with the control plane of Multiprotocol Label Switch (MPLS) in a non-SDN network domain and the west-bound API which connects with SDN controllers in different networking domains.

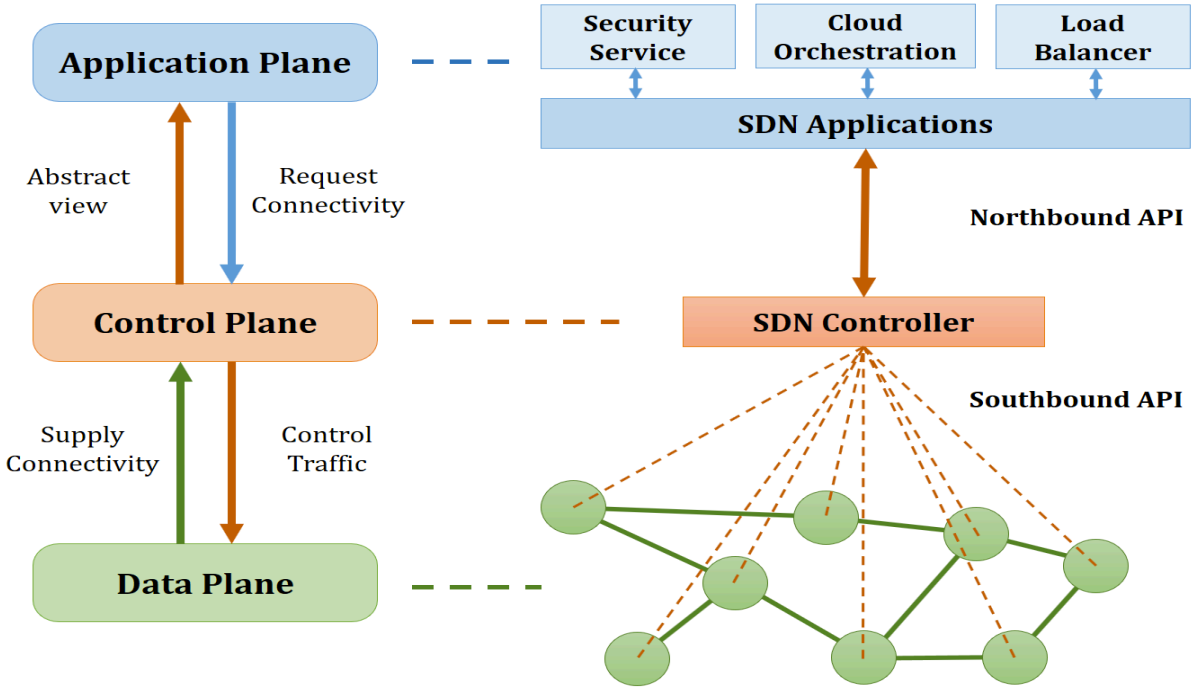


Figure. 1.2 A general Architectural Framework for Software Defined Networking

In packet networks (classical networks), network control functions are coupled with the data plane. Each network device has a network control logic and forwarding decisions are based on destination IP addresses via hop-by-hop forwarding of data packets and the behaviour of the network largely depends on packets flow from neighbouring network components (Son & Buyya, 2018a), (Heller, Sherwood, & McKeown, 2012).

Forwarding decisions in SDN is flow based which is a sequence of data packets carrying identical service policies from a source to a destination (Kreutz et al., 2015). The control entity is responsible for creating abstraction of the network forwarding technologies (e.g.

packet/flow switching) through an application programming interface (API). The control plane (Controller) receives workload requests or instructions from the applications layer and executes them by applying specific configurations to the infrastructure layer (Buyya et al., 2014). The controller makes the network control logic programmable such that it can pass control packets with SDN protocols to switches and other forwarding hardware components in the physical infrastructure and hence making the behaviour of the network thoroughly customizable and purpose oriented.

The open, programmable interfaces enable the networking applications to flexibly interconnect with the underlying physical network infrastructure (i.e., the data plane) which delivers networking services to the applications (Blenk, Basta, Reisslein, & Kellerer, 2016). The OpenFlow (OF) protocol (Mckeown et al., 2008) presents this standardized interface linking the physical network which forwards the payload data and the control plane by implementing an abstraction mechanism (Blenk, Basta, Reisslein, et al., 2016). This standardized data-to-control plane abstraction mechanism which is enable by the OpenFlow protocol makes Software Defined Networking a revolutionary technology for network virtualization.

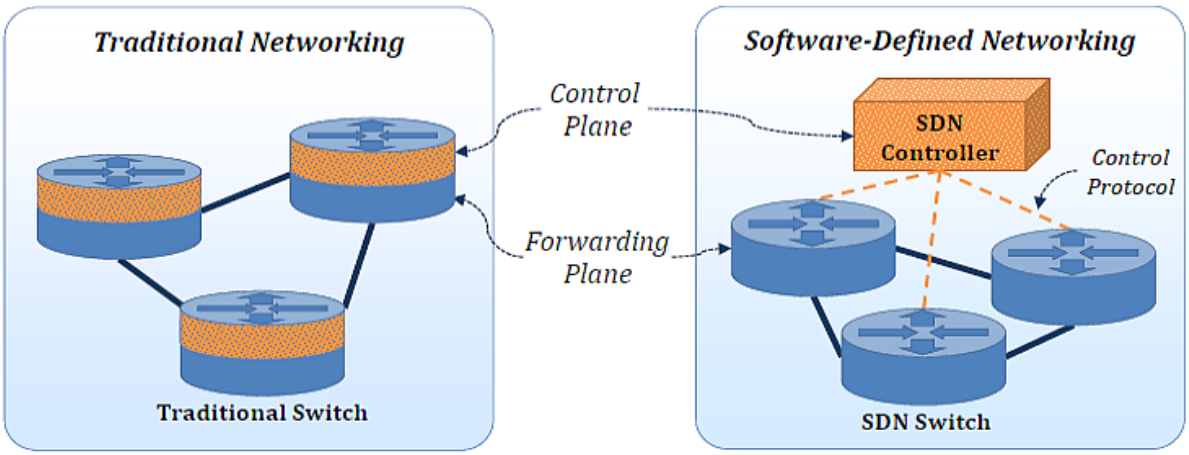


Figure 1.3. Comparing of SDN and traditional networking.

1.2.6 The Concept of Virtualization

Virtualization is a technique of simulating hardware device(s) into software using a firmware or a software. In computing and networking technologies, it involves separating a service or an application from the underlying physical delivery system. Essentially, virtualization simulates physical (e.g. computer hardware, storage capacity) and network resources (e.g. links, switching /routing devices) as software or virtual instances and it allows for multiple tenant access to resources (Duan, 2017). This separation layer is known as virtual machine monitor (VMM) primarily eclipse the physical resources of the computing system from the operating system (OS) as well as enabling sharing of the system resources.

Virtualization has turned out to be a prevailing concept in a variety of areas, mainly, virtual memory (Belbekkouche, Hasan, & Karmouch, 2012), virtual machines (Blenk & Kellerer, 2013), virtual storage access network (Fischer, Botero, Beck, De Meer, & Hesselbach, 2013) and virtual data centres (Khan, Zugenmaier, Jurca, & Kellerer, 2012). With virtualization, the total expenditure of equipment and administration could be exceptionally minimized due to the maximized hardware utilization, abstraction of functionalities from physical infrastructural setups, easier movements to current services and products, and flexible management (Belbekkouche et al., 2012), (Blenk & Kellerer, 2013),(Fischer et al., 2013),(Khan et al., 2012).

1.2.7 Virtualization in Networking

Network Virtualization (NV) is the phenomenon where a given physical network infrastructure and its resources are abstracted to create multiple logical virtual network slices of the underlying substrate (Blenk, Basta, Reisslein, et al., 2016). NV enables independent virtual networks to co-exist on one or more shared physical network infrastructure (Buyya et al., 2014) as shown in Figure 1.4. Each of these virtual networks has specific tailored network service demands or end-user applications requirements from the underlying physical network substrate (Mijumbi et al., 2016). Network service providers deliver services to different service users via sharing of these independent virtual networks deployed over a shared physical substrate. Network Virtualization is considered an enabling networking technology for the next generation internet. It presents the potential to overcome the current network ossification and limitations in the current internet and communication

networks by enhancing network resource efficiency and resource sharing (Blenk, Basta, & Kellerer, 2015).

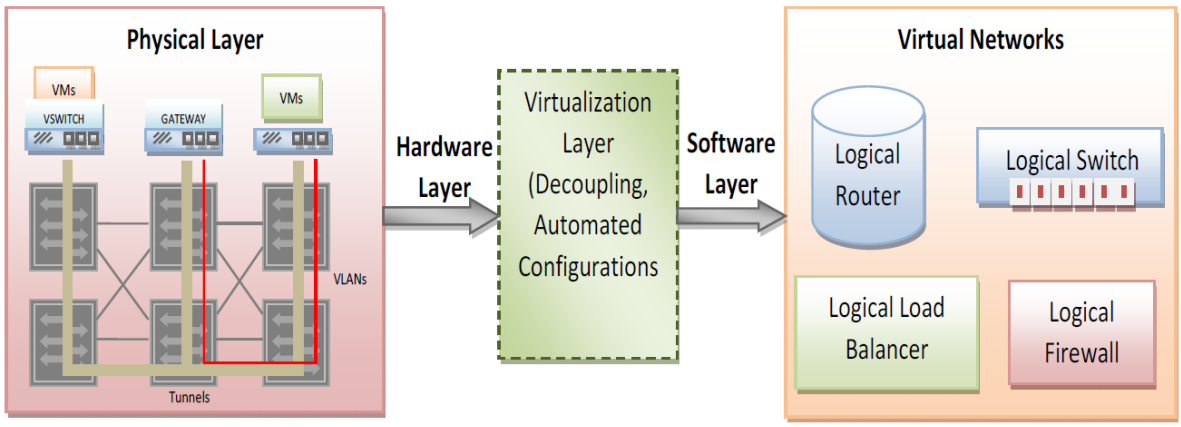


Figure.1.4 A representation of Network Virtualization.

1.3 Motivation

The adoption of cloud computing, emergence of software defined networking and network virtualization have rapidly transformed end-to-end service provisioning over the past decade. On -demand cloud user requests with well-defined service level agreements (SLA) often have different quality of service (QoS) requirements for processing and delivery. Although cloud computing enables the deployment of large-scale cloud network infrastructures and datacentres in different geographical locations, end-to-end service provisioning for user demands within these heterogeneous network domains has proven to be very difficult to achieve. This is because there's no uniform inter-domain service delivery platform with standardized operational policies for service providers to effectively provision end-to-end network services. Similarly, network slicing which is the bedrock of most 5G networks which are designed to support different types of services from different industries at different scales enables sharing of physical network infrastructures but on many different virtual network layers. This implies that, end-to-end communication paths would have to traverse multiple autonomous systems and layers operated by different organizations each with different network policies and hence incurring high network traffic latency, high network overhead and a higher probability of SLA violation. Having multiple cloud service providers does not guarantee availability of resources to meet user demands at the time of

requests. Also, these independent cloud network services do not automatically assist users to decide the type of service or network routes which are available or cost effective at the time of requests. Therefore, network service provisioning in such heterogeneous network environments require a higher-level network resource sharing for flexible interaction between users and service providers as well as a loose-coupling coordination between the autonomous systems involved. This calls for a cognitive aware service delivery platform which is agile and supports efficient user-to-network interaction and inter-domain coordination. Integrating software defined networks (SDN) and network virtualization (NV) in a cloud computing environment, we develop an intelligent network virtualization hypervisor which is able to integrate network resources of heterogeneous networks on a single platform or distributed and provision these network resources with different quality of service (QoS) requirements and parameters. The network virtualization hypervisor will be able to;

1. Prioritize network traffic and resources allocation considering different QoS parameters
2. Systematically determine dynamic overbooking ratios while reducing user SLA violations
3. Analyse and predict datacentre resource utilization through experimental validation using a proposed overbooking algorithm

1.4 Research Problems and Challenges

This research work aims to address research challenges on efficient provisioning of compute and network resources in a software defined cloud datacentre using a uniform cross-domain service delivery platform with standardized operational policies. To be precise, it can be stipulated as:

How to efficiently provision network resources using a self-configuring and self-optimizing network virtualization hypervisor.

1.4.1 Research Challenges

Cloud computing enables the deployment of large-scale cloud network infrastructures and datacentres in different geographical locations. However, efficient end-to-end provisioning of both compute and network resources between these heterogeneous network domains is very difficult to attain.

Similarly, network slicing which is the bedrock of 5G networks which are designed to support different types of services from different industries at different scales enables sharing of physical network infrastructures but on many different virtual network layers. As a result, provisioning of resources have to traverse multiple autonomous layers each with different network policies and hence incurring high network traffic latency and a higher probability of violating user SLA.

Network service provisioning in such heterogeneous network environments require a higher-level network resource sharing for flexible interaction between users and service providers as well as a loose-coupling coordination between the autonomous systems involved.

1.5. Evaluating the Methodology for this Research

The proposed methods used in this research work was evaluated using a discrete-event simulation and empirical operational data from the production environment of Google. The use of simulations makes it possible to implement and evaluate large-scale and complex systems such as virtualized software defined networks in multiple cloud datacentres during experiments; a task which would be almost to impossible achieve in the real world due to financial and management constraints. For this thesis, we developed and adopted CloudSimHypervisor simulation framework. CloudSimHypervisor is an extension of CloudsimSDN-NFV (Son and Buyya, 2019) (a discrete event simulation application which was developed by extending CloudSimSDN (Son et al., 2015) to model SDN virtualization hypervisor features which includes bandwidth allocation and management with regards to user SLA, dynamic network traffic monitoring, dynamic network and network policy reconfiguration for multiple cloud datacentres. This cloud environment simulation

framework is useful for reproducible experiments of large-scale cloud network infrastructure with varying configurations in brief time periods.

1.6 Thesis Contributions

The main contributions of this thesis have been put in three chapters; Modelling the network virtualization hypervisor for software defined cloud datacentre networks in a discrete event simulation framework, novel algorithms for joint compute and network resource provisioning and an intelligent overbooking algorithm to enhance network resource usage and performance in SDCDCN.

1. **Formalised a model and simulation framework for software defined cloud networks:**

Formalised a model for simulating Software Defined Cloud Datacentre Networks in a CloudSimSDN-NFV based discrete-event simulation framework which supports a number of new features that unify computing hosts, software defined networks and cloud datacentre networks. These features include SDN-enabled physical infrastructure components and their configuration, definition and requirements of NFV, SFC, Edge computing and cloud computing architecture. This simulation framework was equipped with features to support a network virtualization hypervisor with well-defined characteristics such ability to stitch together network resources to form isolated virtual SDN networks (vSDNs) and a representation of dynamic virtual networks requests, cloud workloads and their properties. Experimental validation and evaluation of the proposed model and simulation application was done by comparing the simulation results with the observed performances from other simulation applications.

2. **Novel joint computing and networking resource allocation algorithms for software defined cloud datacentre networks**

Formulated joint network resource allocation problem with regards to energy efficiency and SLA compliance. Energy and SLA aware resource overbooking algorithms that jointly allocate and consolidate network traffic with the use of the network virtualization hypervisor. A network bandwidth allocation procedure to allocate the bandwidth requested

to network service users. Implementation of a power model in a simulation framework to evaluate the power consumption in a software defined cloud datacentre network.

3. **Novel intelligent overbooking algorithm for improving resource utilization in software defined cloud datacentre networks:**

Formulated an intelligent performance aware and QoS overbooking algorithm. Developed a network resource usage and performance predictors to predict the usage of CPU and bandwidth performance of host SDCN components using Artificial Neural Networks (ANN). The Levenberg-Marquardt back-propagation strategy was adopted for optimization. Experimental validation using overbooking algorithm from the production environment of Google and multiple overbooking indexes. Network resource overbooking is a strategy which is adopted and implemented in large-scale datacenter networks to allow more user requests to be allocated to network resources than the usual resource capacity allocated for providing the service. It is useful for increasing network service providers' profit as well as enhancing customers' satisfaction (Moreno & Xu, 2012). The training of ANN was done with Google cluster-usage traces.

1.7 Organization of the Thesis

The structure of the chapters in this thesis chapters is shown in Figure 1.5. The remainder of the thesis is organized as follows:

Chapter 2 presents a taxonomy and literature review of usage of Software Defined Networking, SDN and Network Functions Virtualization, NFV in Software Defined Cloud Networks, SDCN.

Chapter 3 presents a modelling and simulation environment of Software Defined Cloud Networks and Datacentre Networks with the implementation of the Network Virtualization Hypervisor.

Chapter 4 suggests a novel dynamic overbooking algorithm for energy efficiency and reducing SLA violation based on a correlation distribution with the use of the network virtualization hypervisor in Software Defined Cloud Networks.

Chapter 5 discusses an analysis of resource utilization in a heterogeneous Software Defined Cloud Datacentre Network and predicting the volume of resources which would be required for effective datacentre network performance.

Chapter 6 summarises the thesis with a discussion on future directions.

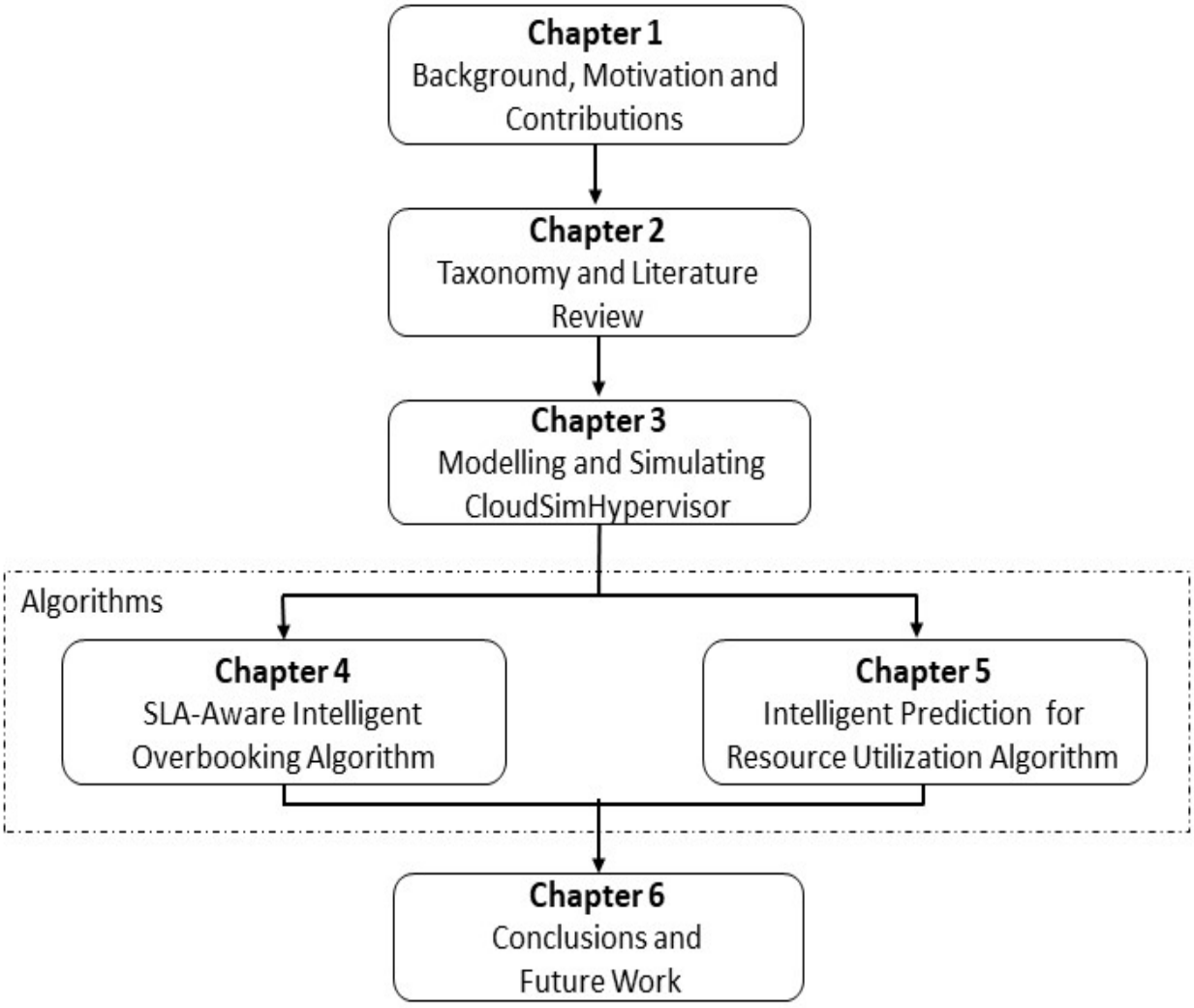


Figure 1.5. Organization of the Thesis

Chapter 2 Literature Review

2.1 Introduction

Network–as-a-Service and cloud computing both adopt the service-orientation concept which is the bedrock of network cloudification and network-cloud unification. This principle enables modern network architectures to leverage cloud technologies to deliver network services in models which are same as cloud service provisioning.

Although NaaS and Cloud Computing are research areas which have attracted extensive interest from both academia and industry, most of the surveys and taxonomies in previous studies concentrated on some salient implementation. For instance (Duan et al., 2014) developed an integrated framework of NaaS in SDN which can support inter-domain end-to-end QoS requirements by leveraging the network service orchestration interface. This account also considered network service capabilities model and thoroughly explained techniques for provisioning guaranteed required bandwidth for heterogeneous autonomous networking systems. The article further discussed the challenge in delivering QoS to support multiple applications, which is a significant challenge internet service provisioning.

(Duan et al., 2012) presented a comprehensive survey on NaaS from the perspective of networking and cloud convergence. The authors introduced a network-Cloud convergence framework based on the concepts of service-oriented architecture (SOA) and network virtualization with specific focus on state-of-the-art network service discovery, description and composition. They further reviewed challenges in deploying network cloud convergence technologies and research opportunities relevant in this area of study. (Baroncelli, 20101) contribution, also proposed a novel classification of NaaS with the perception of orchestrating it with other cloud services. This account introduced a network virtualization platform (NVP) which is a mediation layer capable of exploiting the properties and functions of the network resources of control plane (CP) - enabled networks. The proposed network virtualization layer coordinates resource sharing among the VM and it has the capability to map end-point addresses with well-defined QoS parameters of cloud network user requests to the network infrastructures which possess the required network resource to execute the demands of these user requests.

In this chapter, a thorough account of the Network –as-a-Service (NaaS) concept and Software Defined Networking (SDN) is considered. The study would also include a survey of technologies which are key to the composition and implementation of NaaS. The state-of-the-art both of NaaS and SDN is considered with regards to cloud computing, distributed computing and computer networks. The chapter further emphasize on the significant impact and challenges of NaaS in cloudification of computer networks and network-cloud unification context.

The rest of the chapter is organized as follows: we clarify terms and definitions to be used in this chapter and in this thesis in Section 2.1. In Section 2.2, the researchers present different a taxonomy of usage of SDN and NFV in Software Defined Cloud Networks and reviewed literature, this is followed by Section 2.3 that illustrates current research studies on software defined cloud networks, following which extensive surveys regarding the successes and the flaws of using SDN and NFV in clouds computing environments were conducted in the context of energy efficiency, performance and virtualization. Section 2.4. presents a comprehensive survey of network performance, as Section 2.5, recounts literature and survey on virtualization and summarizes the chapter in Section 2.6.

2.2 Definition of Terms and Concepts

The fundamental approach applied for provisioning resource in cloud networks for the future internet lies in the service- orientation concept in networking. This principle involves integrating key enabling computing and networking technologies in developing infrastructure capable of multiple on demand service delivery. We present the definitions and concepts of key technologies which constitute software defined cloud networks for different purposes based on a wide scope of collective literature survey in this section.

2.2.1 From Virtual Machines to Virtual Networks

Hypervisor also called (virtual machine monitors VMM) was originally developed to facilitate virtualization in the area of computers to aid in creating and managing multiple virtual machines (Blenk, Basta, Reisslein, et al., 2016). With full virtualization, several

virtual machines, each having a different operating system are executed on a common computing platform (Server). The hypervisor also allocates resources such as compute cycles on central processing units (CPU) to each of the virtual machines which are being executed on the physical computing platforms. Customarily, the hypervisor hangs on a set of requests for interfacing with physical computing platforms (an abstraction) (Blenk, Basta, Reisslein, et al., 2016).Computer virtualization enables virtual machines to execute different programs irrespective of the specifications of the underlying computing platforms.

The success of computer virtualization has inspired network virtualization. Comparably, multiple virtual networks can be executed on the same physical network infrastructure irrespective of its specifications through virtualization. A network virtualization hypervisor manages the flow of activities and allocation of resource in the virtual networking context. Network virtualization enables several service providers to flexibly provide current and enhanced services to using existing physical networking infrastructure (Blenk, Basta, Reisslein, et al., 2016).

2.2.2 Techniques for Virtualizing Networks

There are many comparable techniques that can be used to create network "slices" in the sphere of networking. Wavelength Division Multiplexing (WDM) is a technique which create network slices at the physical (photonic) layer (Blenk, Basta, Reisslein, et al., 2016). Virtual Local Area Networks (VLANs) is also employed create network slices at the link layer. Slices of the forwarding table can also be created using Multiple Protocol Label Switching (MPLS). However, network virtualization endeavours to create network slices which with properties across all network protocol layers of the entire network. The properties possessed by these network slices include slice-specific link bandwidth, switch CPU, forwarding tables and network topology view. A given virtual network (slice) provides a setting for examining novel networking paradigms, independent of the constraints imposed by the presently dominant Internet structures and protocols.

2.2.3 Combining Network Virtualization and Software Defined Networking

Software Defined Networking (SDN) provides dynamic programmable network control at run time. Network virtualization (NV) enables network resource sharing among multiple tenant virtual networks (Blenk, Basta, Zerwas, & Kellerer, 2016). These two technologies

though independent can complement each other to leverage the combined advantage of their concepts to enhance the operational efficiency of network and communications systems. With the use of OpenFlow protocol multiple virtual software defined networks (vSDNs) also known as tenant networks of a given physical SDN infrastructure can be executed on a network virtualization hypervisor. Each of these tenant networks is a representation of a "block" or "slice" of the shared physical network (Blenk, Basta, Reisslein, et al., 2016) and execute an isolated independent network operating system, Figure 2.1. Network resources in each of these tenant network slices is a part of the entire resources of the shared physical network infrastructure. Hence, virtualising the physical infrastructure of software defined networks enable multiple tenant networks (vSDNs) to be created. Each of these tenants networks can have individual tenant controllers which is customised for the demands of the vSDN (Blenk, Basta, Zerwas, et al., 2016). These tenant vSDNs could be independent service providers or organizations which utilise the resources of a single shared physical network infrastructure (Blenk, Basta, Reisslein, et al., 2016) . Flexible inter-connection of virtual software defined networks and programming of network slices at runtime form an integral part of next generation networks. This is the reason virtualization of software defined networks is considered a key enabler of the fifth generation (5G) wireless network technology.

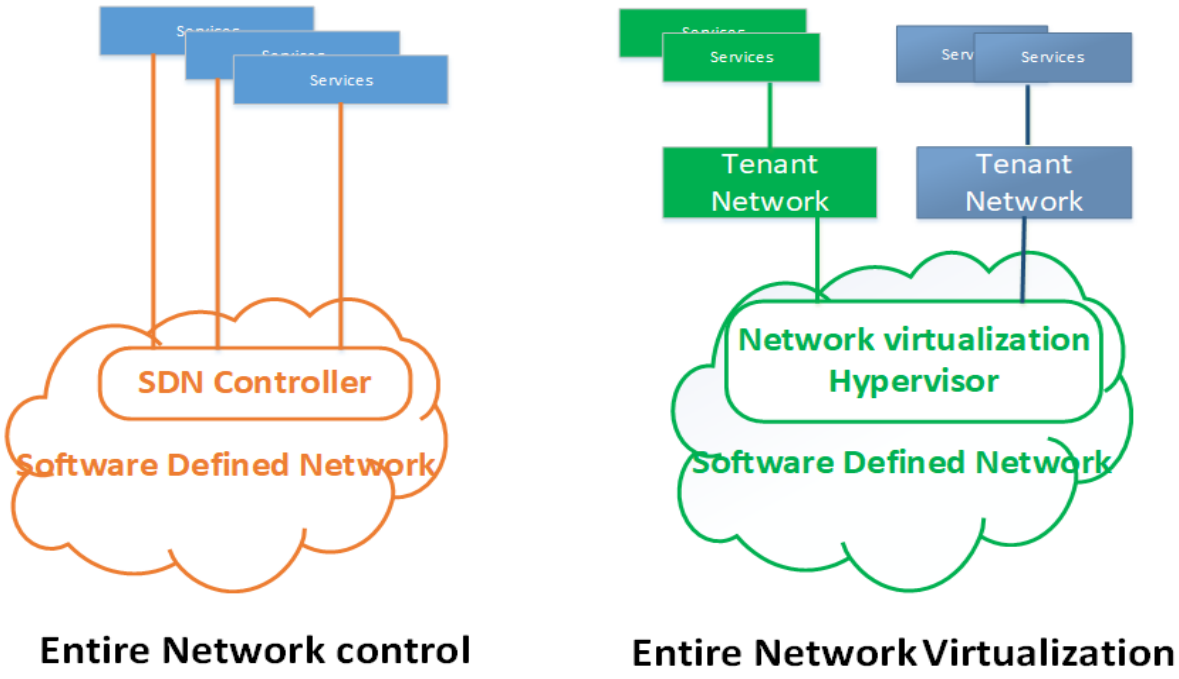


Figure 2.1 Combining Network Virtualization and Software Defined Networking

2.2.4 Managing a Physical SDN Network with a Hypervisor

The emergence and adaption of software defined networking means networks are no longer a consolidation of independent network devices which demand individualized configuration (Blenk, Basta, Reisslein, et al., 2016). Instead, networks can be executed as a single programmable entity. The operation of the centralized controller of SDN in collaboration with the application programming interfaces (APIs) presents network operators with a programmable network control which gives a global view of the entire network and network configurations can be flexibly altered at runtime. This programmable feature of the SDN setup can be utilized in executing virtual SDN networks (vSDNs). To be specific, the framework for virtualizing software defined networks, implements the network virtualization hypervisor between the physical network infrastructure of SDN and the control plane, figure 2.1. illustrates. The data control plane interface D-CPI is the interface through which the hypervisor interacts with and has a global view of the physical network. The network virtualization hypervisor also interacts with heterogeneous virtual software defined network controllers through several D-CPIs, Figure 2.2. These virtual SDN controllers could be traditional legacy SDN controllers. This phenomenon where the hypervisor interacts with heterogeneous vSDNs controllers via a data controller plane interface, D-CPI is considered a key feature of network virtualization hypervisors (Blenk, Basta, Zerwas, et al., 2016). It should be noted that OpenDaylight and some controllers of non-virtualized software defined networks are also used to virtualize networks. However, it is important to note that the virtual network slices produced with this kind of virtualization are administered only at the Application controller plane interface, A-CPI. This does not allow for efficient communication between data plane of software defined networks and their virtual network slices. As a result, OpenDaylight and comparable controllers are not considered as hypervisors. A Hypervisor has the capability to create independent tenant vSDNs with their respective controllers by simulating (virtualizing) the physical network infrastructure of SDN.

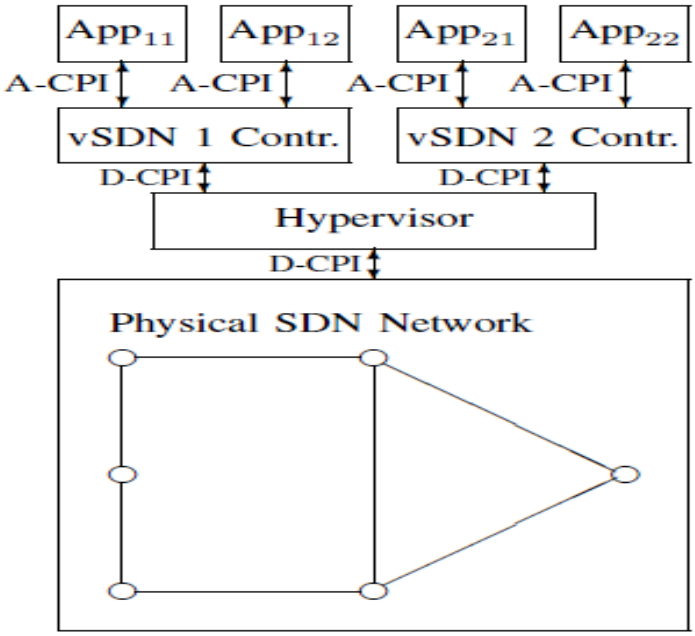


Figure 2.2. Virtualizing the SDN Network (Perspective of the hypervisor): The inserted hypervisor directly interacts with the physical SDN Networks as well as two virtual SDN controllers.

2.2.5 Virtual Network Embedding

The Virtual network embedding (VNE) is the phenomenon of mapping the demands of virtual network components to substrate network components and resources (Beck, Fischer, Botero, Linnhoff-Popien, & De Meer, 2015), Fischer et al. in (Fischer et al., 2013) presented a review and classification of different VNE algorithms which includes several applicable metrics and taxonomies.

2.2.6 Non- virtualized SDN Vrs Virtualized SDN

The design of the control plane and performance of the controller of non-virtualize software defined networks have a significant effect on the output of the network as it contributes to a low flow setup latency (Smith & Nair, 2005) (Waldspurger, 2002b). High control plane latencies contribute discrepancies in operational output of SDN components. Placement of controllers also has a significant effect on the decision latency for non-virtualized SDN networks and hence its performance (Sherwood et al., 2009).

As multiple virtual SDN networks are co-existing in a virtual SDN environment, the demands and needs of each tenant for SDN network performance can directly be applied to the network hypervisor placement problem. As already mentioned, in case of virtualized

SDN networks, the hypervisor processes all network control traffic exchanged between tenant controllers and their virtual SDN networks. In particular for long-propagation-delay WANs, the hypervisor instances have to be distributed for scalability reasons. As for non-virtualized SDN networks, an unplanned hypervisor placement may lead state inconsistencies of the applications of the tenants [8].

2.2.7 Network Virtualization Hypervisor

Software defined networking provides a platform to dynamically develop and deploy network protocols to ensure operational efficiency through the use of its programmable network control logic. However, the APIs currently available for discovering, accessing and provisioning network resources differ with respect to service providers as do the network policies and authentication schemes (Huang & Griffioen, 2013). Consequently, setting up an end-to-end SDN network demands OpenFlow applications which are capable of interconnecting the non-identical APIs provided by the various network services providers (Huang & Griffioen, 2013).

An abstraction layer which enables SDN applications to be seamlessly executed across different SDN providers allowing end-to-end internet-scale service provisioning is the ***network virtualization hypervisor***. This layer has the capability to conceal the underlying details of different SDN providers from the SDN applications and hence enable a heterogeneous group of SDN providers to execute an end-to-end applications and network service provisioning, a process which would otherwise be difficult or impossible to achieve. As a result, the hypervisor is able to 'stitch together' a vSDN slice from different physical software defined network infrastructures. This attribute of the network virtualization hypervisor addresses the challenges presented by;

virtualizing software defined network infrastructures which have different levels of abstraction and a variety of APIs which is also known as network slicing (Huang & Griffioen, 2013).

Hence, the motivation for the network virtualization hypervisor proposed in the research is the difficulty of virtualizing the current heterogeneous software defined network infrastructures.

The Network Hypervisor acts as a controller to the network applications of the vSDN tenants. Network applications can use a higher-level API to interact with the Network Hypervisor, while the Network Hypervisor interacts with the different SDN infrastructures and complies with their respective API attributes. This Network Hypervisor design provides vSDN network applications with a transparent operation of multi-domain SDN infrastructures.

In a perfect virtualized SDN environment, the network hypervisor has full control over the entire physical network infrastructure. (We do not consider the operation of a hypervisor in parallel with legacy SDN networks, since available hypervisors do not support such a parallel operation scenario).

2.2.8 Type of Network Virtualization Hypervisor Architecture

2.2.8.1 Centralized Network Virtualization Hypervisor

A hypervisor has a centralized architecture if it consists of a single central entity. This single central entity controls multiple network elements (NEs), i.e., Open Flow switches, in the physical network infrastructure. Also, the single central entity serves potentially multiple tenant controllers. Throughout our classification, hypervisors have been classified by the authors that do not require the distribution of their hypervisor functions as centralized. We also classify hypervisors as centralized, when no detailed distribution mechanisms for the hypervisor functions have been provided. We sub-classify the centralized hypervisors into hypervisors for general networks, hypervisors for special network types (e.g., optical or wireless networks), and policy-based hypervisors. Hypervisors are commonly implemented through software programs (and sometimes employ specialized NEs) (Blenk, Basta, Zerwas, et al., 2016). The existing centralized hypervisors are implemented through software programs that run on general purpose compute platforms, e.g., commodity compute servers and personal computers (PCs), henceforth referred to as “compute platforms” for brevity (Blenk, Basta, Zerwas, et al., 2016).

2.2.8.2 Distributed Network Virtualization Hypervisor

We classify an SDN hypervisor as a distributed hypervisor if the virtualization functions can run logically separated from each other. A distributed hypervisor appears logically as consisting of a single entity (similar to a centralized hypervisor); however, a distributed hypervisor consists of several distributed functions. A distributed hypervisor may decouple

management functions from translation functions or isolation functions. However, the hypervisor functions may depend on each other (Blenk, Basta, Zerwas, et al., 2016). Distributed hypervisors may employ compute platforms in conjunction with general-purpose NEs or/and special-purpose NEs. We define a general-purpose NE to be a commodity off-the-shelf switch without any specialized hypervisor extensions. We define a special-purpose NE to be a customized switch that has been augmented with specialized hypervisor functionalities or extensions to the OF specification, such as the capability to match on labels outside the OF specification (Blenk, Basta, Zerwas, et al., 2016).

2.3 Taxonomy of usage of SDN and NFV in Software Defined Cloud Networks

Software defined networking, network functions virtualization in the cloud computing environment have several use cases. As result, in this research we carried out a thorough study of these technologies considering different approaches for implementing and integrated them with other useful networking technologies and proposed a taxonomy. The taxonomy is explored in the context of objective, method scope, target architecture, application model, resource configuration, and evaluation method. Figure 2.3 illustrates the taxonomy designed for the rest of the literature review.

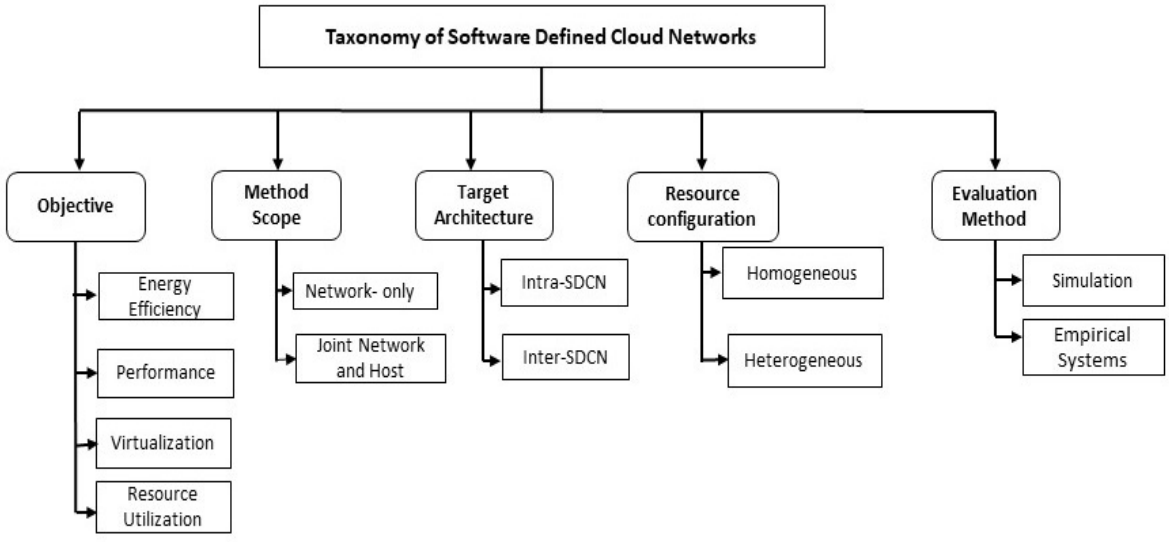


Figure. 2.3 Taxonomy of usage of SDN and NFV in Software Defined Cloud Networks

2.3.1 Objective

Communication service providers and Cloud providers leverage modern networking technologies such SDN, NFV and edge computing components in their datacentres to enhance operational efficiency. The research community also consider several different objectives in their research studies. In this thesis, we classify the objectives into four main blocks in our research survey and review of literature as shown in this section.

2.3.1.1. Energy Efficiency

This is a well-studied area in cloud computing and cloud-enabled technologies such as software defined networking, cloud network unification and network virtualization. Over the past two decades, energy efficiency management in cloud datacentres is an area that has received a lot of research attention because, these datacentres consume huge amount of energy globally. Host servers, networking devices, and cooling system are the consumers of the greater proportion of energy in cloud datacentres (Pakbaznia & Pedram, 2009). Servers, networking devices and cooling systems in datacentres which can be turned off when they are not in use to conserve some energy in the datacentre. The use of Software Defined Cloud Technologies in cloud datacentres enhances the possibility of network and computing elasticity. Consolidation of network traffic into a reduced number of network hosts and switches in runtime with the use of the dynamic network management resources in software defined cloud resources contributes to joint host and network energy optimization.

2.3.1.2 Performance

Performance of end-to-end cloud networks can be enhanced with the provision fine-grained centralized or distributed network control. The controller of Software Defined Networks provided a network a dynamic network control system which enhanced the monitoring and performance of computer networks better than traditional networks. Integrating other networking technologies such NFV, Edge computing components, Service Function Chaining (SFC), etc. extends the scope of the operation of network control and optimizes network operation such as micro-segmentation congestion of network traffic and load balancing. Deploying software defined cloud Datacentres with the implementation of a network virtualization hypervisor is one efficient method to reducing the response and processing time of network user requests and workloads. The delay in VM deployment can be decreased by reducing the amount of data transferred in end-to-end Software Defined

Cloud Networks (Samant & Bellur, 2016). Dynamic bandwidth and network resources allocation is another optimization technique in software cloud networks which does not just help to reduce throughput and latency but also enhances network resource availability and quality of service delivery to network users.

Although datacentre networks are expected to have network resources available during all of their operating hours, some traditional datacentres are not able to do so due to sudden breakdown or malfunction of network hardware components. In Software Defined Cloud datacentres networks, the effects of such breakdowns are mitigated with the use of the network virtualization hypervisor. An intelligent autonomous network hypervisor can identify these breakdowns and system malfunctions and micro-segment, consolidate or reroute network traffic flows by re- programming network components and Datacenter Network (DCN) topology configuration.

Quality of service delivery in computer network operations and management requires a guarantee to satisfy the demands of network users having different SLAs without having a negative impact on the performance of the entire network. This was almost impossible to attain in traditional networks and although it was enhanced with the introduction of SDN, software defined cloud network implementations with the network virtualization hypervisor, has endowed cloud datacentre networks with the capability to dynamically allocate and reallocate dedicated network resources to priority user without affecting the performance of the entire network (Son, 2018). Furthermore, the QoS of software defined cloud networks can be enhanced with mechanisms possessed by the hypervisor which allows for high workload acceptance rate, optimal orchestration of network and cloud resources, detection of network link congestion and dynamically changing the paths of network flows. (Gharbaoui, Martini, Adami, Giordano, & Castoldi, 2016) assessed network performance with regards to workload acceptance rate and the probability of blocking networks to evaluate network resource schemes.

2.3.1.3 Virtualization

As a concept, virtualization has been extended and redefined with advances in technology particularly with Software Defined Cloud Networks and datacentre implementations. Cloud computing and most of its implementations use virtualization to deploy virtual machines and other cloud resource for the IaaS, SaaS, PaaS (Buyya et al., 2014). The trending wave of

virtualization which involves an integration of SDN and NFV deploys Network-As- A-Service, NaaS, which emanated from the concept of computer resources virtualization. NFV enables the technique to abstract and assign control of network functions from hardware to software with the introduction of a hypervisor (Networks, 2014) . Previously, Hypervisors were deployed by running or installing them on physical hosts and were used to virtualize the host resources such as CPU, memory, and storage. This concept was employed in developing network virtualization hypervisors which are implemented in software, firmware and hardware with the introduction of NFV. If integrated with SDN which is capable of abstracting network control form the physical infrastructure the corporate benefits of these two technologies can be leveraged to create the network virtualization hypervisor which is can enable multiple isolated virtual networks to share a common physical network infrastructure (Braun & Menth, 2014). Virtual Network embedding problem, is phenomenon of mapping virtual network requests to physical network resources for execution (Fischer, Botero, Beck, De Meer, & Hesselbach, 2012).

Network functions virtualization traditionally presented in middleboxes with dedicated hardware boxes such as load balancers, firewalls and intrusion detection. These hardware boxes are difficult to scale and manage and are extremely expensive.

2.3.1.4 Resource Utilization:

Cloud computing and network virtualization and network slicing technologies have introduced a number of new methods of network resources and functions sharing and utilization. One of the methods which seem common to all these technologies is multi-tenancy of virtualized networks and network infrastructure sharing with or without SLAs. (Sciancalepore et al., 2017) introduced the network virtualization substrate (NVS) which enables different network operators to share resources under different conditions.

2.3.2 Method Scope

In this section we suggest a classification with regards to the scope of methodology. Researchers usually make use of a variation of classification in their study of networking and cloud computing (Amarasinghe, Jarray, & Karmouch, 2017) (Egilmez, Dane, Bagci, & Tekalp, 2012). Some consider just networking in cloud computing environment while others use an approach which consider both host and cloud networks. Those who consider the network-only methods usually make use data which is collected after the network

forwarding rules are altered by the controller in the cloud and hence making use of SDN to find a solution to the research problems. For instance, most of the research approaches which are aimed at network performance optimization usually consider cloud networking resources (Ishimori, Farias, Cerqueira, & Abelem, 2013) (Jain et al., 2013).

Joint research approaches take into account the optimization and orchestration of both the host and virtual network resources simultaneously. Optimization of network resources of traditional Datacentre networks mostly considered the host only approach (Beloglazov, Abawajy, & Buyya, 2012). With the introduction of software defined networks, research mostly find joint optimization more appealing (Zheng, Zheng, Li, & Wang, 2017) (Gharbaoui et al., 2016)(Cziva, Jouët, Stapleton, Tso, & Pezaros, 2016).

2.3.3 Target Architecture

Software defined cloud networks have introduced a new dimension of flexibility and control in modern networks integrating the attributes of SDN and NFV (Cui, Yu, & Yan, 2016) . It has also contributed immensely to the architectures of cloud datacentres. SDN has been used in many research studies to establish the relationship between host and controllers in local datacentre networks (Intra-DCN) where traditional switches were replaced with SDN-enabled switches and hosts. When integrated with NFV, datacentre networks are optimized with regards to utilization of resources, network traffic load balancing and bandwidth allocation.

Integrating it with Edge computer nodes, extends the network optimization and QoS delivery to cloud networks and datacentre networks in other geographical locations (Inter-DCN) (Jain et al., 2013) (Mechtri, Houidi, Louati, & Zeghlache, 2013) (Petri et al., 2015a). This mechanism has been beneficial to cloud service providers who operate multiple datacentres as it gives them leverage to develop network control systems to manage the various WANs between the datacentres. Other researchers such as (Licciardello et al., 2017)(Fichera, Gharbaoui, Castoldi, Martini, & Manzalini, 2017)(Bonafiglia, Castellano, Cerrato, & Risso, 2017) suggested various applications of network orchestration in network architecture development stating the use of the management and orchestration (MANO) feature of network functions virtualization to efficiently manage network traffic and balancing of network load in both inter and intra datacentre networks considering the use of edge

computing nodes in unifying multiple cloud networks where necessary. In these instances, the SDN controller for SDN enabled networks and the network virtualization hypervisor for software defined cloud networks logically managed the network control for geographically distributed datacentres. They further proposed methods which utilized the network functions virtualization infrastructure (NFVI) feature of NFV to providing the mechanism for enabling virtualization of network functions.

2.3.4 Resource Configuration

The different types and vendors of the hardware resources used in setting up cloud datacentre networks depicts the resource configuration of the datacentre. Resource configuration is a factor which affects the efficiency of the designing of the network topology in cloud a datacentre. For instance, a network resources consolidation technique which is used for energy conservation in a homogeneous setup would not be as effective in a heterogeneous setup if the specifications of the individual physical host network components are considered.

With *homogeneous* configurations, it is assumed that the specifications of the host networking components (physical machines, switches, network links and racks, etc.) are all the same in magnitude or are same in number. Homogeneous network configuration has proven to be useful for research simulations and experiments. This is because, it makes it easy to model complex network topologies and reduces the time researchers use in conducting experiments. However, homogeneous network setups are not applicable in the real-world datacentres as there are periodic upgrades of network resources and capacities.

Heterogeneous network configuration is the kind of network topology setup for a datacentre which is made up of physical network components having different specifications in magnitude and numbers. Although homogeneous network configuration is useful for research study, it is implemented only in limited testing environments as it cannot be used for complex software defined network topologies and evaluation processes. As a result, a lot of research studies consider homogeneous configuration of host network resources for preliminary experiments and implement heterogeneous network configuration which emulate real world datacentre network setups for extended research. Research studies on network optimization, energy efficiency etc. for inter- datacentre networks (inter-DCN)

usually consider heterogeneous network setup as the networking components for multiple datacentres would comprise hardware from different operators and vendors (Heller, Seetharaman, Mahadevan, & Y, 2009).

2.3.5 Evaluation Method

Accessing the performance and efficiency of Cloud datacentres can be challenging due to the huge number of components, resources and solutions and the complexity of the network topologies which are adopted with regards to the expected network traffic demand and other internal and external factors. As a result, researchers adopt strategies which produce the best output in the best time considering their resource constraint to test and evaluate datacentre efficiency and new methods to be implemented. Two strategies which researchers mostly adopt are simulations and empirical evaluation methods.

A. Simulation

Simulation is considered the most economically strategy to experiment new network topologies, algorithms and methods to rolled out with acceptable accuracy. Software defined cloud datacentre networks can be modelled with random configuration values at extremely low cost. Although the output of simulation experiments may sometimes be inconsistent with the realities of real-world network setups when the margin of error of the configurations used for the experiments statistically high, it is an easier solution for implementing proposed concepts which also produces quicker outputs. Most simulation toolkits such as iCanCloud (Núñez et al., 2012) are used as stand-alone applications to experiment new software defined cloud network concepts. They are sometimes implemented with empirical systems such as SDN controllers and multi-access edge computing nodes which could be adopted for real-world cloud datacentre networks for some research studies.

B. Empirical

Empirical method of evaluation produces more pragmatic outputs compared to the simulations. This is because the experiments are conducted with real-world network systems. However, it is almost impossible to evaluate large scale Software defined cloud networks with dynamic and multiplex topologies with this method as it seems impractical and the network components which are required are extremely expensive. Research studies which applied the empirical evaluation include CometCloud (Petri et al., 2015b), ElasticTree (Heller et al., 2009) and CARPO (X. Wang, Yao, Wang, Lu, & Cao, 2012)

2.4 Current Research Studies on Software Defined Cloud Networks

This section presents relevant research studies which have been conducted on software defined cloud networks as outlined in the suggested taxonomy above. Classification of the state-of-the-art was done with regards to the main objectives of this thesis. Although a lot of research studies consider multiple objectives, the classification in this section takes into account only the primary objectives of the research studies under consideration.

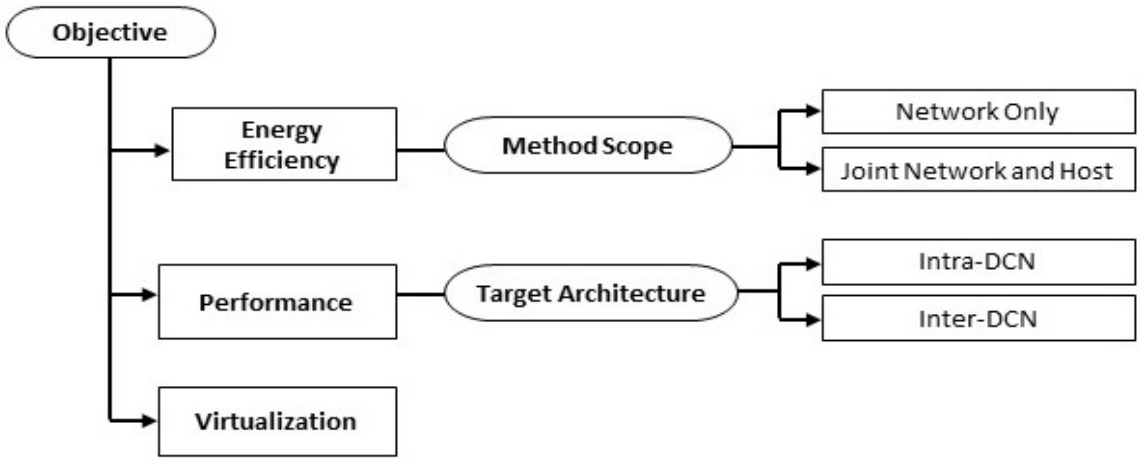


Figure 2.4: Sub-classification of the literature review.

Figure 2.4 displays a sub-category of the above taxonomy for further classification to be used for the surveyed research studies in this section. A comprehensive literature in each category is extended below.

2.4.1 Energy Efficiency in Software Defined Cloud Networks

Optimizing energy efficiency is one of the most popular and relevant research areas in cloud computing and cloud network unification such as SDN-enabled cloud computing and software defined cloud networking. I thorough study of previous researches revealed that enhancing the conservation of energy in cloud datacentre networks are usually considered in two broad categories which are network optimization and joint host-network optimization.

Network Optimization involves enhancing network components such as switches, routers and the adapters in the host networks of cloud datacentre networks. Similar to Dynamic Voltage and Frequency Scaling (DVFS) (Abts, Marty, Wells, Klausler, & Liu, 2010), where the link rate of network devices is reduced in the event of under- utilization, network topology in cloud datacentre networks dynamically changes with regards to the traffic from network user requests (Heller et al., 2009).

Alteration of the network topologies is done with the consolidation of network flows in under-utilized network connections or links. Another procedure for optimizing cloud datacentre networks is correlation-aware network optimization where, network traffic consolidation is compared with the total network flows in the datacentre network (X. Wang et al., 2012). The above-mentioned procedures are among the most applicable research procedures for optimizing network energy efficiency in software defined cloud networks.

Research studies on ***Joint host-network optimization*** takes into account the simultaneous impact of both the host and network resources in reducing energy consumption allocation of network resources, network traffic micro-segmentation and consolidation and overbooking. The network virtualization hypervisor receives user request and allocates these requests to physical host resources which have adequate available resources to process the requests. This process allows datacentre hosts to be free from hosting VMs since the network hypervisor receives and disseminates the user request and network traffic in order to reduce the energy consumption in the datacentre. The network virtualization hypervisor is also able to consolidate and micro-segment priority network traffic from ordinary and standard ones and hence turning off physical host resource which are not required for specific network activities. It also overbooks required host resources when necessary in order to maximize resource utilization. The use of the network virtualization hypervisor in software defined cloud networks, enable efficient implementation joint host-network techniques in reducing total energy consumption in cloud datacentres.

Project	Description	Author	Organization
ElaticTree	Network Traffic consolidation	Heller et al. [44]	Stanford University, USA
CARPO	Correlation analysis, traffic consolidation	Wang et al. [125]	The Ohio State University, USA

DISCO	Distributed traffic flow consolidation	Zheng et al. [134]	The Ohio State University, USA
FCTcon	Dynamic control of flow completion time in DCN	Zheng and Wang [133]	The Ohio State University, USA
GETB	Energy-aware traffic engineering	Assuncao et al. [29]	University of Lyon, France
VMPlanner	Grouping, and consolidation of VM	Fang et al. [35]	Beijing Jiaotong University, China
VMRouting	VM placement problem to a routing problem, shortest path	Jin et al. [60]	Florida International University, USA
PowerNetS	Correlation analysis, VM placement, migration	Zheng et al. [98, 100]	The Ohio State University, USA
S-CORE	VM management considering host-network	Cziva et al. [29]	University of Glasgow, UK
QRVE	Energy efficient VM placement and routing	Habibi et al. [43]	Amirkabir University of Technology, Iran
ODM-BD	Big data workload slicing in edge-cloud environment	Aujla et al. [7]	Tharpar University, India

Table 2.1 Outlining summary of studies on energy efficiency of Cloud Networks.

A. Network Optimization

Cloud datacentre optimization to enable efficient energy consumption has attracted massive interest from the research community. ElasticTree (Heller et al., 2009) proposed modifying links and switches to dynamically change the network topologies of datacentre networks in order to conserve energy. The fat-tree topology by design is able to consolidate traffic flows to a lesser number of host resources such as links and switches. The unused host network resources are turned- off to conserve energy in the datacentre network. The authors assessed the viability of their proposed procedure for trade-offs between robustness, performance and energy consumption using real-world data from production datacentres. A prototype of this method was implemented with a monitor for inputting various traffic patterns and their latencies for evaluation.

The correlation between traffic flow consolidation and traffic flows in network connections is considered in CARPO (Wang et al., 2012). Traffic flows which are less correlated are used in same networks. This enables the network links to accommodate more network flows with limited resource capacity and hence maximizing resource utilization. The remaining host network resources are the turned off to conserve energy. The authors of this proposed method observed Wikipedia traces for correlation analysis. They deduced from their observation that off-peak utilization of data traffic is less than that of peak since data traffic do not get to their peak at the same time. As a result, resource utilization during off peak is used for consolidation in order to optimize energy efficiency and reduce performance degradation. Furthermore, CARPO made use of link adaptation which alters the speed of every port in relation to the rate of utilization of the links. This demonstrated that the implementing the suggested correlation-aware traffic consolidation procedure with web application workload outperforms the ElasticTree method.

Zheng & Wang, (2017) presented a distributed traffic management framework which is capable of evaluating the correlation between network traffic flows and consolidating the flows with regards to energy savings, scalability and performance. This technique concentrated on the consolidation of correlation-aware flow scalability and reduced the calculation for evaluating decision making and correlation. The research team presented a switch based traffic consolidation algorithm that enhances the traffic flows through individual switches instead of considering the traffic flows in the entire datacentre network. This research team also proposed a systematic approach to flow control completion time (Zheng & Wang, 2017) in a study which considered optimizing energy efficiency flow transmission time.

Dias de Assunção et al., (2017) recently suggested a technique for engineering network traffic while saving energy with regards to SDN services. The team of researchers modelled a system to manage switches and network traffic with use of SDN services. The rational for their model is to forward all network traffic to connections and paths which have adequate network resources to execute and balance the load while idle switches are turned off. The model was evaluated with both simulations and empirical techniques. Large scale evaluation was done with the use of simulations while a real-world system was developed as a proof-of-concept and for small evaluation. These assessments demonstrated that the proposed architecture could maintain QoS and optimize energy efficiency by consolidating network

into a lesser number of network link connections. They also deduced that QoS may degrade with the use of network traffic consolidation in the event where the quantity of network packets are more than the link capacity, a phenomenon is known as ***Network Burst.*** As a result, the research team suggested frequent and uninterruptible observations of the network in order to increase the number links in the event of over utilization of existing links.

B Joint Host-Network Optimization

The research community has in recent times suggested optimization methods which leverage the simultaneous effects of joint host and network resources in datacentre network optimization. Software defined cloud networks which basically comprise SDN, NFV, SFCs, and edge computing components in a cloud computing environment make use of virtual networking resource provisioning, allocation, migration, sharing, placement and optimization to facilitate efficient QoS delivery.

Medina and Garcia suggested some efficient techniques for migrating virtual machines in cloud datacentres (Medina & GARCíA, 2014). They categorised migration techniques into three classes; process migration, memory migration and resume/suspend migration.

Being an effective means of user requests dissemination, (Kapil, Pilli, & Joshi, 2013) provided the research community with a comprehensive survey of live virtual network migration and the issues associated with it. They assessed the effectiveness of live migration procedures with a preamble which took into account network parameters such as network downtime, migration and rate of data transfer per data size. They classified this approach into pre-copy and post-copy methods.

A strategy to optimise placement of virtual machines and network routing using approximation algorithms was proposed by (Fang, Liang, Li, Chiaraviglio, & Xiong, 2013). The authors focused on proving that reducing energy consumption with regards to VM placement and network routing is an NP-complete problem. They implemented three NP-complete problems: traffic-aware VM grouping, distance-aware VM group to server-rack mapping and power-aware inter-VM traffic flow routing to formulate the VM placement and routing problem. The solution to the proposed NP-complete problem included VM grouping which consolidates VMs with joint high traffic, VM group placement which

allocates VMs to the rack of specified VM group in order to minimize inter-rack traffic through traffic flow consolidation.

Jin et al., (2013) applied integer-linear programming to adjust a joint host-network optimization problem. The authors combined network resource placement problem and routing problem and then formulate an integer linear algorithm to solve the problem. They then employed the depth-first search (DFS) criteria from graph theorem to determine the best host to be allocated virtual network resources. The derived algorithm was executed on an OpenFlow-bases physical system which was designed with fat-tree topology to compute the number of workloads the prototype can support compared with that in simulations.

Zheng, Wang, Li, & Wang, (2014) and Zheng et al., (2017) recently proposed PowerNets which enhances the efficiency of network traffic management and allocation of virtual resources with the help of correlation analysis. The authors, like (Wang et al., 2012) adapted the joint host-network optimization technique to enhance energy efficiency the in cloud datacentres. They leveraged correlation of virtual network resource consolidation and applied CARPO's network traffic consolidation to deduce a power model evaluate datacentre energy consumption which included the power consumed by each port on a switch, switch chassis, power consumed by idle servers and the maximum energy consumed by servers. They then measured the correlation coefficients between traffic flows from Wikipedia and Yahoo traces and designed the PowerNets framework with regards to the estimated correlations using the physical prototype and simulations.

Cziva et al., (2016) proposed a live virtual network resource migration pertaining to a network-wide communication cost using an SDN-enabled virtual network management platform. A cloud datacentre testbed was developed to implement the prototype. The research team suggested hierarchical architecture system which supports OpenFlow network resources and Libvirt virtual machine manager. The system also supports Ryu SDN controllers which were used for discovering network topologies, L2 switching management, calculating the weights on links and gathering statistics. They also deduced cloud network orchestration algorithms which reduce communication cost of VM-to-VM during VM migration. These algorithms were set for periodic recurrence and follow a hierarchical pattern in order to lower the cost of network traffic.

Habibi, Mokhtari, & Sabaei, (2017) presented an energy efficiency and QoS strategy using virtual network resource placement and network routing. This strategy was implemented on a system which consist of QoS-aware routing protocols, monitoring and flow detection mechanism, network topology storage. The algorithm utilized dynamic routing of traffic flow to attain QoS as well network requests and traffic consolidation to maintain energy efficiency in software defined cloud networks. The researchers deduced a combined approach and exploited necessary trade-offs between energy efficiency and QoS in their approach. They calculated network throughput, energy consumption and network resource utilization of software defined cloud datacentres using Mininet simulation application.

Aujla, Kumar, Zomaya, & Ranjan, (2018) suggested a decision-making scheme for slicing up workloads for data-intensive applications in a multiple edge software defined cloud network environment. Networking parameters which were considered in developing their system model which was used for making efficient decisions on inter-cloud data migration include bandwidth optimization, SLA, energy consumption, revenue and communication cost. The network virtualization hypervisor allocates network traffic flows to paths in multiple datacentres and edge nodes based on SLAs of the user requests. Google workload traces was used to evaluate the proposed decision-making scheme.

2.5 Performance

Optimization of network performance is a major benefit of adapting cloud computing. However, the emergence of software defined networking and network virtualization technologies have provided mechanisms for cloud and network unification which allow network service providers the capability to further automate and expand usage of network resource with advance and dynamic network architecture. In this thesis, we classify the various software defined cloud datacentre network architectures used into intra-Software Defined Cloud Networks (inter-SDCN) and inter-Software Defined Cloud Networks (intra-SDCN).

Intra-SDCN strategy is adopted to optimize network performance and resource utilization within a software defined cloud datacentre networks. ***Inter-SDCN*** is implemented to enhance QoS between multiple tenants of software defined cloud datacentre in a wide area network (WAN) setup. A number of research studies (Egilmez et al., 2012) (Jain et al.,

2013) (Antequera et al., 2018) (Wang, Butnariu, & Rexford, 2011) (Wu et al., 2017) which have been conducted in this area of studies relate performance efficiency to the dynamic network traffic flow routing using the network control system of SDN which is able to redirect traffic with regards to network activity and state, for instance locating an alternate route in the event of congestion or network burst to maintain required QoS. Although these techniques enhance the performance efficiency of datacentres locally and in multi-geo locations, it also presents some infrastructural challenges as the SDN controller is centralized and hence presenting a single source of failure which affects all other network components which depend on it. The SDN controller requires a huge amount of memory and compute resources as it manages all other network components regardless of the size of the network. These challenges could be overcome by integrating network virtualization and edge computing technologies with SDN in order to enhance the performance efficiency. We reviewed some research studies which concentrate on enhancing the performance software defined cloud networks and summarized them in the table below.

Project	Description	Author	Organization
OF-SLB	Server load balancing	Wang et al. [94]	Princeton University, USA
QoSFlow	QoS management system for traffic engineering	Ishimori et al. [47]	Federal University of Para, Brazil
AQSDN	Autonomic QoS management system	Wendong et al. [107]	Beijing University of Posts and Telecom, China
SDN-Orch	SDN-based orchestration for host-network resources	Martini et al. [61]	CNIT and Sant'Anna School, Italy
C-N-Orch	Cloud and network orchestration in a datacentre	Gharbaoui et al. [41]	Sant'Anna School and University of Pisa, Italy
Orch-Opti	SDN-based virtual resource orchestration system for optical DCN	Spadaro et al. [88]	UPC, Spain
OpenQoS	QoS-aware network traffic controller	Egilmez et al. [33]	Koc University, Turkey
B4	SDN-WAN for geographically distributed datacenters	Jain et al. [108]	Google Inc.
CNG	Enhanced networking of distributed VMs	Mechtri et al. [63]	Telecom SudParis, France
ADON	Network management system for scientific workload	Seetharam et al. [109]	University of Missouri, USA
CometCloud	SDN-enabled cloud federation for smart buildings	Petri et al., [72]	Rutgers University, USA
SD-IC	Inter-connection for federated SDN-clouds	Risdianto et al. [110]	GIST, Korea

Orch-IC	Network resource orchestration for inter-clouds	Kim et al. [54]	ETRI, Korea
VIAS	SDN overlay bypass for intercloud VPN	Jeong and Figueiredo [49]	University of Florida, USA
CL-Orch	Cross-layer network orchestration signaling framework	Cerroni etal. [26]	University of Bologna and Sant'Anna School, Italy
SVC	SDN-based vehicular cloud for software updates distribution	Azizian etal. [111]	Univeristy of Sherbrooke, Canada
BDT	Optimization model for bulk data transfers	Wu et al. [96]	The University of Hong Kong, Hong Kong
SDN-TE	Fault-tolerant cloud resource management	Amarasinghe et al. [5]	University of Ottawa, Canada

Table 2.2. Outlining summary of studies on performance of Cloud Networks.

A. Intra-SDCN Performance Improvement

Load balancers (LB) are usually used in traditional datacentres to re-channel requests from clients to servers in order evenly allocate network workload across heterogeneous servers. In place of using these dedicated expensive physical boxes, (Wang et al., 2011) suggested an OpenFlow-based load balancing procedure for cloud datacentre networks. The researchers suggested OpenFlow enabled switches with the capability to alter and redirect the routes of all incoming network traffic to allocate them across heterogeneous datacentre networks. The challenge that this method presents is the huge volume of traffic rules which has to be installed in the switches of the heterogeneous servers in the datacentres. The algorithms which were implemented by the authors discovered wildcard protocols and grouped network traffic flows to specifically defined servers in order to enhance network traffic scalability. The load balancing method that the researcher suggested was deployed on NOX OpenFlow controller.

Ishimori et al., (2013) focused on providing an adaptable QoS control logic by proposing a system for managing the QoS flows using multiple schedulers in a Linux kernel. Some Linux kernels traffic schedulers such Hierarchical Token Bucket (HTB) and Stochastic Fairness Queuing (SFQ) were integrated into OpenFlow networks in order to enable SDN controller to assign Linux traffic schedulers their appropriate QoS parameters. Standard OpenFlow switches were extended to develop QoS Flows to plan the data flows through the switches. When implemented and experimented in physical commodity switches, the researchers demonstrated that Linux schedulers do work on OpenFlow enabled switches and the control

overhead of the OpenFlow packets a reduction in the maximum bandwidth. This method provides an alternative framework for QoS-aware software defined cloud networks.

Another approach to improving the QoS of Datacentre Networks can be derived with the integration of SDN and Autonomous Networking Technologies as proposed by (Wang, Qi, Gong, Hu, & Que, 2014). With this approach, QoS requirements are automatically applied in configuring SDN controllers. Network queue management procedures, packet schedulers and their OpenFlow-based parameters are configured with the use of the autonomic QoS control module in order to manage the data flows by changing the forwarding rules to match packet headers. The research team initiated an information set which contains the characteristics of packets and was carried in the IP header. This packet information set which is used to mark packets is called the Packet Context. With the use of the mark, packets are organized in a queue based on their priority in order to meet different QoS demands. This procedure allowed customization of IP packet headers which limited its implementation in general-purpose networks but is applicable in intra-SDCN with a high possibility of customising network traffic.

Martini et al., (2016) proposed an architecture for managing both compute and network resources in a virtualized cloud datacentre. This system incorporates a VM-host management system which is able to orchestrate compute and network resources to enhance network request acceptance rate and datacentre resource utilization with SDN. It also contributes to maintaining the quality of network user experience. The system is composed of a coordinated configuration and provisioning, monitoring and registration functions and resource selection and composition functions. The authors develop a model which combines instantaneous and historical values compute network traffic and predict network load. Furthermore, the system uses a network resource allocation algorithm which discovers a network or server first for a virtual machine request. The efficiency of this system and the suggested algorithms were evaluated using both simulation and empirical setups.

Spadaro, Pagès, Agraz, Montero, & Perelló, (2016) proposed a virtual network orchestration model for optical datacentre network which integrates with OpenStack and is able to regulate OpenDaylight controller through the North-bound API. This system comprises of software defined networking resources and virtual networks and an algorithm for enhancing the acceptance ratio of network requests through the use of jointly mapping the virtual channels or links and the virtual machines in

a virtual network request to a datacentre network. The research team accessed performance efficiency of the proposed system and the algorithm in a cloud intra-datacentre network by estimating the acceptance demands and the probability of rejection rate.

B. Inter-SDCN Traffic Engineering

Software defined cloud networks provides platforms and resources which have optimized end-to-end network traffic transmission through heterogeneous networks. An OpenFlow-enabled software defined networking controller which was designed to dynamically re-route and allow end-to-end multimedia traffic transmission between end-users and streaming servers to meet different QoS requirements is OpenQoS (Egilmez et al., 2012). With OpenQoS incoming network traffic are categorized with regards to specified QoS requirements into two groups: multimedia traffic flows and other data flows. As the multimedia streaming flows usually have high QoS requirements, they are transferred onto dedicated routes with high resource allocation which are different from other flows routes that use the shortest path algorithms. The authors utilized the dynamic routing and flow separation mechanisms of software defined networks to enhance QoS delivery and compared with the resource reservation approach applied in traditional networks. OpenQoS use packet header filtering to join multiple flows together in groups of fields. However, packet filtering in switches should be carefully defined and aggregated due to the cost of processing it. Implementation and testing of network traffic prioritization in intra-SDCN and inter-SDCN of OpenQoS was done with the aid of a small-scale physical testbed.

Google embraced the concept of SDN and NFV with the deployment of B4 to decouple network control plane from the data forwarding plane to make their networks programmable (Jain et al., 2013). B4 which is a private wide area network which connects geographically distributed heterogeneous datacentre networks was developed with the exact specification and attributes of Google's datacentre networks. The attributes that were considered include, huge bandwidth demand, adjustable traffic requirements and complete control of edge servers and network. OpenFlow and applications of SDN which have the capability to balance network traffic congestion and use multi-path forwarding to dynamically allocate bandwidth in order to control network capacity was adapted to support traffic engineering and management of routing protocols. Google's assessment of network optimization after deploying B4 showed that, network resource utilization in their heterogeneous datacentres (inter-DCN) was optimized by 100% as an average of 70% utilization rate was recorded for

all the network connections and links. This performance result is an enhancement of twice or 3 times the efficiency as compared to traditional datacentre networks which have a resource utilization rate even after overbooking of 30%-40%

Mechtri et al., (2013) proposed an all-purposed SDN controller which can be configured and run by authorized users for connecting inter-cloud networking gateway. This controller allows interconnection of virtual machines of geographically distributed heterogeneous datacentre networks with the use of the SDN-enabled gateway system. It also allows these networks to be customised for resource allocation and management with regards to the requirements of the virtual machines. A cloud broker was initiated to control VM allocation between the distributed datacentres and the network infrastructures involved with regards to the network configurations.

ADON is an SDN and NFV enabled network control system which was proposed for a hybrid cloud architecture to execute strenuous science applications (Antequera et al., 2018). Although these science applications usually run on private clouds of universities, there are some workloads which require additional resources for execution which are available in public clouds. In such instances, where multiple science applications contend for remote resources, network bottlenecks could be a major challenge. As a result, the research team developed and installed a software defined networking controller to monitor the characteristics of all the scientific applications and prioritize network requests with regards to their QoS demands with the use of the dynamic resource allocation features in OpenFlow for bandwidth allocation and per-flow routing. The testbeds for these experiments were implemented with emulations from Mininet in two campus locations.

Risdianto, Shin, & Kim, (2016) suggested the use of SDN and NFV in federated cloud datacentre network interconnection. Their approach utilized L2 tunnel-based overlay virtual networking and L3 BGP leveraged routing exchange to interconnect multiple datacentre networks. Using an integrated system of SDN and NFV the system which was developed could flexibly configure and orchestrate resources in chosen cloud datacentre networks and their forwarding paths to optimize redundancy and load balancing. Using these two networking techniques enabled the flexible preselection of L2 and L3 for interconnection of cloud datacentre networks.

Bonafiglia et al., (2017) and Gray & Nadeau, (2016) SDN controller for ONOS and Open-Daylight were used for deploying L3 routing exchange and L2 monitoring respectively in physical prototypes.

Petri et al., (2015b) discussed integrating inter-cloud federations with software defined networks to estimate the volume of sensor data for smart buildings. Their studies focused on real-time data processing of applications which made use of huge volumes of data. They presented an architecture which applied concepts of federated clouds and software defined networks to in a multi-cloud interconnection to allow network programmability and scalability of network resources. Optimizing processing time and cost energy utilization was evaluated for different scenarios using smart building setups. A prototype of the architectures used for this study have been implemented on three building sites in the UK and the US and ratified with real sensor workloads. It was deduced from the implementations that the overall task execution for time-constraint workloads was drastically reduced due to reduction of the in-transit time due to SDN usage.

Kim, Kang, Cho, & Pahk, (2016) discussed an orchestration architecture which involved an integration of SDN-enabled clouds, NFV and transport networks. The research investigates how to use SDN and NFV features to control clouds and inter-cloud networks in order to enhance inter-DCN transmission. Cloud SDN (C-SDN) and Transport SDN (T-SDN) were implemented to manage intra-DCN and inter-DCN respectively. The authors setup Virtual Network Coordinators and Transport Network Coordinators to create all the virtual networks which would be used in the heterogeneous datacentre networks. This proposed orchestration architecture was tested on a physical testbed using OpenDaylight, ONOS, and OpenStack.

Jeong & Figueiredo, (2016) presented an NFV and SDN-enabled intercloud VPN technique which eliminates network overhead due to tunnelling protocols and increase overlay flows flexibility. In this research, virtual networks were created in heterogeneous datacentre networks and tunnelling packets were selectively bypassed using the setup which was created by integrating SDN into overlay controllers. This technique was tested with the use of containers which were running on virtual machines in configuring VPN for several VMs executed by different operators and datacentres. The physical prototype was deployed with the use of a Ryu controller which manages the Open Vswitches executed by the VMs.

Throughput on transmission through the containers and across cloud service providers was also computed.

Cerroni et al., (2015) suggested an SDN and NFV enabled signalling framework system for multi-domain data transport network orchestration which is able to deliver QoS requirements for multiple applications. The application -Oriented Module (AO-M) which executes services such as SIP proxy server, network module which manages the SDN controller layer also manages configuration of the network resources and description parser. Validation of this framework was done with empirical testbed in several locations with multiple commercial equipment.

Azizian, Cherkaoui, & Hafid, (2017) suggested the use of SDN, NFV and cloud computing to disburse software updates for vehicles. Using the management and orchestration (MANO) of NFV and the SDN controller, data flows of multiple inter-datacentre networks and networking devices in various base stations which manage the software update information is transferred through cloud datacentre networks to vehicles.

Amarasinghe et al., (2017) discussed a fault tolerant control framework for heterogeneous clouds using software defined networking. The research team proposed an adaptive traffic engineering mechanism to enable restoration of network failure using the dynamic routing and network monitoring features of software defined networking. The proposed fault tolerant framework discovers unexpected link failures and recovers them by resetting the forwarding rules of the switches in the cloud datacentre networks. The prototype of this proposed system was executed on a POX controller and the output was emulated with the use of Mininet simulation application.

2.6 Virtualization

As defined in Section 2.3, Software Defined Networking (SDN) provides dynamic programmable network control at run time. Network virtualization (NV) enables network resource sharing among multiple tenant virtual networks (Blenk, Basta, Zerwas, et al., 2016). The network virtualization technique slices physical network resources in cloud datacentre networks into well-defined segments for tenant virtual networks. Making use of generic computing resources, NFV transfigures network applications and functions to run on arbitrary hardware which are independent of custom networking hardware and which

requires extremely high computing power. As a result, alternative to procuring dedicated off the shelf hardware for resource intensive network functions like firewalls, NAT, load balancers, etc. network functions virtualization provides the technology which enables resources of generic hardware extra useful. NFV presents a paradigm shift of networking and deployment of network functions with a new wave of virtualization technologies (Callegati et al., 2016)

Popa et al., (2012) suggested the FairCloud as an energy saving technique which is applied by virtualizing cloud datacentre networks. The research team categorize the advantages of network resource sharing in cloud computing environment into three, which are; network proportionality, efficient resource utilization and guarantee of minimum bandwidth. They defined network proportionality as the activity of sharing network resources fairly among cloud tenants such that all the tenants have equal proportion of the resources. However, it is necessary to consider the fundamental trade-offs between the three categories as they are dependent on each other. For instance, change in network proportionality impacts minimum bandwidth guarantee and vice versa. Similarly, change in the network utilization trade-off would also impact network proportionality or minimum bandwidth guarantee and vice versa. To secure network proportionality in cloud datacentre networks, network bandwidth should be shared equally among cloud customers with same network plan although their bandwidth usage policies may vary. Hence firm network proportionality techniques in cloud datacentre networks reduces the general bandwidth utilization in the datacentre. The network sharing techniques which were suggested by the authors were evaluated with the use of simulation applications.

Souza, Miers, Fiorese, & Koslovski, (2017) investigated a QoS-aware virtual infrastructure (VMs and their network connections) allocation problem on SDN-clouds as a mixed-integer program. The researchers deduced a virtual infrastructure allocation algorithm in runtime.

To find the best solution to the mixed integer programming problem, which is an NP-hard heuristic problem, the research team applied a relaxed linear program rounding techniques and heuristic approaches. They initiated a new virtual machine selection method which considers end-to-end latency requirements of the virtual machines and the geographic location of the vacant zone. The algorithm also considered the following constraints; size of forwarding table, latency, and link and server capacity. The research team estimated five

different metrics in their evaluation of the proposed algorithm in a simulation environment. These metrics are the following; mean latency of the allocated virtual infrastructure, datacentre fragmentation, revenue-cost ratio, acceptance ratio, runtime of allocation.

Mijumbi et al., (2016) investigated the relationship between software defined networks and network functions virtualization. They presented a comprehensive explanation of the concept of NFV and related its functions with SDN and cloud computing. A business model and a detailed architecture of NFV and its modules were also developed. The study also presented key collaborative projects involving industry and academia and a broad scope of NFV standardization.

Esposito, Matta, & Ishakian, (2013) proposed a slice embedding problem during network virtualization as a challenge derived from network resource allocation. The problem has three components; resource discovery, virtual network mapping and allocation. Network resource allocation method, type of dynamics and constraint type characterized the literature survey of the study.

Leivadeas, Falkner, Lambadaris, & Kesidis, (2017) proposed an optimum allocation procedure for virtualized network functions for software defined networks in a cloud computing environment. Since VNF can be hosted on any hardware, the VNF allocation problem has gained attention with the advent of NFV. The research team conducted intensive research regarding services provided by single and multiple tenant VNF models. MANO part of NFV manages the control logic of SDN and cloud controllers to choose the most suitable location for VNFs. They deduced a VNF allocation problem to minimise the cost of operations to cloud service providers considering cloud-based servers and switches. An integer linear program with four heuristics was deduced for the most optimal solution.

Project	Description	Author	Organization
FairCloud	Trade-offs sharing networks in cloud computing	Popa et al. [74]	UC Berkeley, USA
QVIA-SDN	Virtual infrastructure allocation in SDN-clouds	Souza et al. [87]	Santa Catarina State University, Brazil
Opti-VNF	Optimal VNF allocation in a SDN-cloud	Leivadeas et al. [58]	Carleton University, Canada

Dyn-NFV	Dynamic NFV deployment with SDN	Callegati etal. [25]	University of Bologna, Italy
E2E-SO	End-to-end NFV orchestration for edge and cloud datacentres	Bonafiglia et al. [19]	Politecnico di Torino, Italy

Table 2.3: Summary of current research for Software Defined Cloud Networks and Functions Virtualization.

Callegati et al., (2016) suggested a proof-of-concept framework of a dynamic NFV implementation in the cloud. The proposed system is able to administer network and network functions orchestration and management to homogeneous and heterogeneous datacentre networks for telecommunication and cloud service providers with the use of cloud management services and network control. The research team applied VM hosting procedures for both single and multiple VNFs which easily adapt to network conditions. The proof-of-concept framework was executed on Ericsson Cloud Lab environment running Ericson Cloud Manager on top of OpenStack.

Project	Objective	Scope	Architecture		Resource Configuration		Evaluation Method	
			Intra-DCN	Inter-DCN	Homogeneous	Heterogeneous	Simulation	Empirical
ElaticTree	Energy	Network	√		√			√
CARPO	Energy	Network	√		√			√
DISCO	Energy	Network	√		√		√	√
FCTcon	Energy	Network	√		√		√	
GETB	Energy	Network	√		√		√	√
VMPlanner	Energy	Joint	√		√		√	
VM-Routing	Energy	Joint	√		√		√	√
PowerNetS	Energy	Joint	√		√			√
S-CORE	Energy	Joint	√		√			√
QRVE	Energy	Joint	√		√		√	
ODM-BD	Energy	Joint		√		√	√	
OF-SLB	Performance	Network	√		√		√	
QoSFlow	Performance	Network	√		√			√
AQSDN	Performance	Network	√		√			√
SDN-Orch	Performance	Joint	√		√		√	√
C-N-Orch	Performance	Joint	√		√		√	
Orch-Opti	Performance	Joint	√			√		√
OpenQoS	Performance	Network		√	√			√
B4	Performance	Network		√		√		√
CNG	Performance	Joint		√		√	√	
ADON	Performance	Network		√		√		√
CometCloud	Performance	Network		√		√		√
SD-IC	Performance	Network		√		√	√	√
Orch-IC	Performance	Network		√		√		√
VIAS	Performance	Joint		√		√		√
CL-Orch	Performance	Network		√		√		√
SVC	Performance	Joint		√		√	√	
BDT	Performance	Network		√	√			√
SDN-TE	Performance	Network		√	√		√	
FairCloud	Virtualization	Network	√		√		√	
QVIA-SDN	Virtualization	Joint	√		√		√	√
Opti-VNF	Virtualization	Joint	√		√		√	√
Dyn-NFV	Virtualization	Joint	√	√		√		√
E2E-SO	Virtualization	Joint	√	√		√		√

Table 2.4: Descriptive Details of Software Defined Cloud Network Projects.

2.7 Summary

All of the research studies and projects used in our survey have been summarized and compared in Table 2.4 with regards to the various components derived in our taxonomy as shown in Figure 2.3. A lot of research studies along various scopes are being conducted on energy efficiency and performance optimization with the use of modern networking technologies; SDN, NFV, Edge computing, cloud computing and IoT as the world prepares to embrace the 5th generation network technologies. Since research objectives and outcome vary due to various reasons, some researchers consider network-only methodologies while others focus on joint computing and networking resource optimization. Most of the studies on energy efficiency considered an intra-DCN architecture setup as a model since the focus was on energy conservation within the datacentre networks. It can be deduced from this observation that the volume of electricity being utilized in modern day datacentre networks has increased drastically and a major challenge for network and communication service providers in the industry (Delforge, 2014).

Inter-DCN architectural setup was the most preferred for most studies on enhancing the performance optimization of various networking technologies in the cloud computing environment although some few considered network performance with datacentre networks. The researchers implemented simple and complex topologies to optimize QoS and network bandwidth local and wide area networks. Software defined cloud networks make use of the dynamic adaptable network configuration, network orchestration and transport mechanisms and scalability of SDN, NFV, Edge computing, SFCs, multiple clouds in the cloud computing environment to acquire more availability and reliability in datacentre network resulting in better QoS.

Chapter 3 Modelling and Simulating Software-Defined Cloud Networks

3.1 Introduction

Modelling and simulation of software defined cloud networks can be a complex task, due to features such as scalability and availability of its on-demand resources. It is essential therefore to adapt tools and toolkits which have the appropriate functions for cloud environment simulations and testbed experiments. Developing strategies for implementing and comparing novel cloud networks to maximize network resource utilization require simulators which are able to facilitate designs which involve OpenFlow, software-defined virtual networks and inter-cloud datacentres. Network setups and topology of Open Flow switches (hosts) can be emulated using Mininet (Lantz, Heller, & McKeown, 2010). Primarily, Mininet provides a development environment and a virtual testbed to create prototypes of a variety of networks which includes software defined networks. It also has the capability to perform experiments involving discrete load balancing and network traffic management policies in the controller of software defined networks. However, Mininet does not have the capability to perform cloud network resource management procedures such as virtualizing network resource, network resource consolidation, network slicing and VM placement. Effective management of resources and policies to ensure quality of service (QoS), reduction in the SLA violation, energy efficiency and resource sharing and slicing form an integral part of network cloud unification. Like many other applications available for cloud environment simulations, virtualization is considered with specific regards to creating virtual machines from compute servers and its resources such as storage, RAM and CPU or virtual network embedding problem for the network virtualization with no provision for the network virtualization hypervisor and its properties.

To address this challenge, we developed CloudSimHypervisor, a cloud computing based, software defined technologies simulation framework which has the capability to simulate policies for collective allocation of compute and network resources. The adoption of cloud computing, emergence of software defined networking and network virtualization have rapidly transformed end-to-end service provisioning over the past decade. On -demand cloud user requests with well-defined service level agreements (SLA) often have different

quality of service (QoS) requirements for processing and delivery. Although cloud computing enables the deployment of large-scale cloud network infrastructures and datacentres in different geographical locations, end-to-end service provisioning for user demands within these heterogeneous network domains has proven to be very difficult to achieve. This is because there's no uniform inter-domain service delivery platform with standardized operational policies for service providers to effectively provision end-to-end network services. Similarly, network slicing which is the bedrock of most 5G networks which are designed to support different types of services from different industries at different scales enables sharing of physical network infrastructures but on many different virtual network layers.

This implies that, end-to-end communication paths would have to traverse multiple autonomous systems and layers operated by different organizations each with different network policies and hence incurring high network traffic latency, high network overhead and a higher probability of SLA violence. To successfully implement network infrastructure to mitigate these challenges demand integrating cross vendor heterogeneous network resource with the right specifications and right topologies. To achieve this objective of using and accelerating the deployment of Software Defined Cloud Networks, Figure.3.1 there is a need to check, cross-check and re-check their viability, efficiency and their ability to integrate with existing technologies. To accomplish this task in the real world can be very expensive and challenging. Therefore, there is a need for a cloud environment simulator, which has a user friendly and easy-to-learn testbed environment and is able to measure the performance and assess the behaviours of modern networking technologies in a controlled environment.

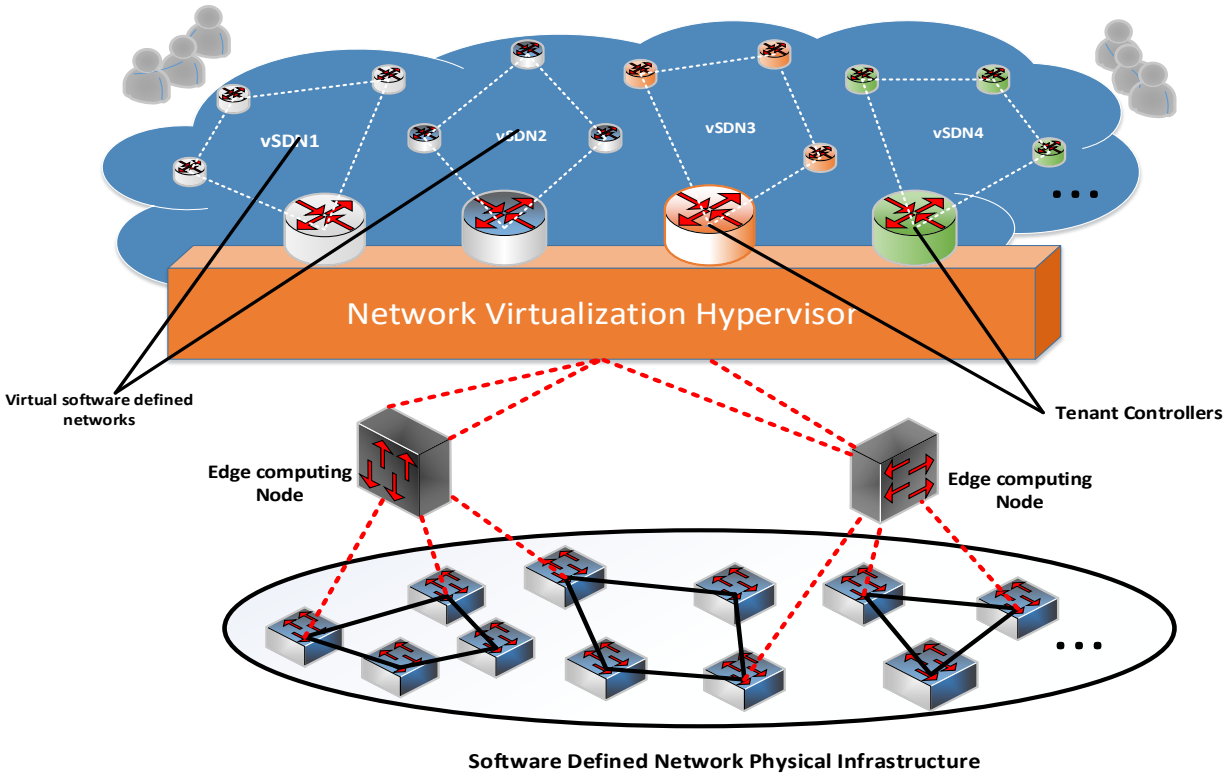


Figure. 3.1. A Software Defined Cloud Networking Architecture

3.2 Related Work on Solutions to verify SDN and NFV Concepts

As mentioned earlier, an appropriate simulation environment facilitates the process of research by making provision for repetitive experiments in controlled environment (Stojmenovic, 2008). These simulation applications provide the environment to study the impact of a variable or parameter on an objective function whiles other variable(s) relating to the research are controlled. An activity which could be extremely difficult or probably impossible to achieve in the real world. Several cloud environment simulation tools and frameworks with the capacity to support reproducible testing and evaluating novel cloud resource management policies, strategies and algorithms have been proposed in recent times.

One of these simulation tools is the iCanCloud simulator (Núñez et al., 2012).The primary objective of this solution is to simulate of large scale cloud experiments with specific regards to enabling a cost-performance analysis of workloads which are executed in a cloud datacentre. iCanCloud is equipped with the INET framework Network which allows simulation of network infrastructures which includes network devices e.g. (routers and switches) and protocols such as User Datagram Protocol (UDP) and Transmission Control

Protocol (TCP) (Núñez et al., 2012). This application does not support the modelling and simulation of SDN controllers and related features.

CloudSim (Calheiros, Ranjan, Beloglazov, De Rose, & Buyya, 2011) is a discrete event-based cloud environment simulator which is implemented in Java. It enables simulation of datacentres with many hosts. Virtual machines, VMs could be created and be placed in a host according to VM placement policy of the host. After creating the virtual machines, workloads or applications can be submitted and executed in the virtual machines. CloudSim, makes provision for additional elements can be developed and added to work with existing entities in the simulator by sending and receiving events. However, the cloud simulator does not support analysis of network performance evaluation in details.

NetworkCloudSim (Garg & Buyya, 2011), is extension of CloudSim which makes provision for simulating generalised application models and scalable networks with communication tasks such as message passing application and workloads in a datacentre. Network elements such as switches and links which were not originally available in CloudSim were implemented in CloudSim. These elements were used to analyse and estimate network transmission time.

Although NetworkCloudSim simulates scalable networks in a datacentre, the features that it implements does not include features for SDN and its dynamic configurable network setup as well features for virtualizing the network resources in the infrastructure plane of SDN. Our research highlights maximising utilization of network resource of software defined networks through the deployment of a self–configuring network virtualization hypervisor.

Another cloud environment simulator which is able to estimate and analyse energy efficiency of both computing and network resources which includes SDN features is GreenCloud (Kliazovich, Bouvry, & Khan, 2012). The GreenCloud framework was developed as an extension of NS2 simulator.

A simulation framework for testing and experimenting cloud management software and its functionalities is RC2Sim (Citron & Zlotnick, 2011). Experiments using this tool are performed by emulating the cloud management software in a single host. Developed purposely for analysing control commands to cloud infrastructure for instance, that request

for creating VMs, RC2Sim simulates networks through a module which considers the cloud network topology allocated to a user to calculate the expected data transfer times. This application does not analyse cloud applications' performance considering different cloud environment policies.

A widely used SDN emulation application which can be used to conduct experiments with hundreds of nodes for different network topologies in Linux is Mininet (Lantz et al., 2010).Mininet uses virtual test beds and virtualization techniques available in Linux OS to monitor and analyse network traffic in the SDN controller. Unlike some simulators previously discussed, results obtained from experiments using Mininet are mostly accurate with respect to delays and congestions at the operating system level. Mininet also makes provision for external OpenFlow controllers to be connected and tested. However, similar to NS-3, Mininet does not have the capability to test cloud-specific network attributes such as network user workload schedulers, virtual network allocation policies, etc.

A simulation framework for testing SDN cloud datacentre controllers using POX and Mininet, python controller for OpenFlow SDN standard was pioneered by (Teixeira et al., 2013).This application used POX to implement SDN controllers while Mininet was used to manage data traffic and network topologies. Results obtained from this application are applicable to practical scenarios and can be used for real life implementations. However, it does not allow simulation of cloud-specific features such as different configuration of VM types and application execution.

3.3 Event-driven Simulation

Simulations in CloudSim follow a well-defined process of sending and receiving events between entities. For instance, to create a virtual machine in on a host in a datacentre, a VM request event is sent to the datacentre entity. The datacentre entity receives the event and allocates resources to execute the requested VM through the use of VM allocation policies assigned to it. Likewise, the VM, through its processing scheduler can receive a CPU workload event through a datacentre entity. The processing scheduler in this activity receives the CPU workload when it arrives, calculates its processing end time and returns the completed event with the end time. This procedure is applied in CloudSimHypervisor as well to simulate network transmissions, creating and deleting VNF and inter-cloud events.

Simulation events are sent and received among entities while the policies and schedulers calculate event delays. These policies and schedulers can be customised and reused to implement various scenarios and algorithms (Son and Buyya, 2019).

3.4 Software-Defined Cloud Networks Simulation

As mentioned earlier, simulation environment quickens the processes and stages involved in theoretical research as it enables reproducible experiments in controlled environments (Beloglazov & Buyya, 2010). Furthermore, it enables strategies to be evaluated in different scenarios and to compare with existing procedures to discover enhanced novel concepts and methodology. There are simulation tools purposely for evaluating cloud infrastructure and their policies which do not support experiments for software defined networks and network virtualization. And there are other simulation applications which also focus on simulating SDN, virtual network embedding, VNE other network virtualization procedures but do not support cloud computing experiments (Son, 2018).

Software defined network clouds, cloud environment networks, network functions virtualization and their service functions leverage the properties of all the enabling technologies which complement the efficient deployment of such cloud infrastructures. As a result, a resource which has the capability to support the design and experimentation involving these technologies to evaluate different quality of service, QoS metrics under different experimental conditions is required. To achieve the stated objectives in this thesis, the experiments performed had the following requirements;

- The ability to simulate SDN physical infrastructure, components and their configuration
- Support to simulate flows and different physical and virtual network allocation policies which are implemented per flow in an integrated cloud network solution
- The ability to represent dynamic virtual networks requests and cloud workloads and their properties
- The capacity to support a network virtualization hypervisor with well-defined characteristics such as ability to stitch together network resources to form isolated virtual SDN networks (vSDNs)
- Support for network specific tenant controllers for each virtual software defined network
- Support for inter-cloud networking design and configuration

- Support to model network functions virtualization, NFV and Service Functions chaining, SFC
- Support for resource provisioning for network functions virtualization (NFV) in the edge computing environment
- The capacity to simulate NFV framework in edge and cloud computing (inter cloud data centres)
- Provision for policies which support network link selection, VM allocation, Virtual Network Function (VNF) placement, and SFC auto-scaling algorithms
- Capacity to evaluate performance of the above-mentioned frameworks with use case scenarios

The above-mentioned requirements and others motivated the design and development of the CloudSimHypervisor simulation framework. The details of this framework are presented in the next section.

3.5 CloudSimHypervisor

CloudSimHypervisor was developed by extending CloudSimSDN-NFV (Son and Buyya, 2019) which was developed as an extension of CloudSimSDN simulation toolkit (Son et al., 2015) which also extended CloudSim simulation toolkit (Calheiros et al., 2011).This simulation framework provides the environment to implement and evaluate SDN -enabled network resource management policies for single and multiple datacenters as well virtualizing network resources of heterogeneous physical software defined network infrastructure. It is able to simulate components in the physical infrastructure of SDN such as switches (core, aggregate and edge), SDNhosts, physical network links, switching components which are required to create a backbone connection to SDN network resources in multiple cloud datacenters such as gateway switches and intercloud switches, virtual network topologies, a variety of distributed network control mechanisms (Network operating systems) capable of emulating the network virtualization hypervisor and NFV enabled service functions to compute varying predefined quality of service (QoS) attributes. The simulator is also able to measure the performance metrics of the network hypervisor with regards to energy savings, cost efficiency and environmental conservation. Although Network-as-a-Service (NaaS) which involves a merger of software-defined networks (SDN)

and network functions virtualization (NFV) as well service oriented architectures (SOA) is one of key strategies for the fifth generation (5G) internet little consideration has been given to simulating integrated network architectures involving all of its key networking technologies. CloudSimHypervisor enables researchers to implement, test and evaluate techniques for maximizing network resource utilization by means of network virtualization and network slicing through the deployment of centralized and distributed network hypervisors, high performance edge nodes for simulating networks in different geographical domains, service functions chaining (SFC) and virtualized network functions (NFV).

3.6 Framework Design

The CloudSimHypervisor simulation framework was developed by extending functions, classes and packages of CloudSimSDN-NFV (Son and Buyya, 2019). CloudSimSDN-NFV (Son and Buyya, 2019) was developed by extending CloudSimSDN toolkit [8] which is also an extension of CloudSim toolkit (Calheiros et al., 2011). CloudSimHypervisor employs the capacity of CloudSimSDN-NFV to model the physical and virtual network infrastructure of SDN, their components and dynamic configuration, setting up virtual network functions (NFV), edge computer nodes and service functions chaining, SFC using its simulation engine. CloudSimSDN-NFV has well defined components to setup and configure different scenarios of software defined networks (SDN) and to simulate behaviours to demonstrate centralized network control and network traffic management with the use of the network operating system (SDN Controller). It also has components to model network functions virtualization (NFV) and service function chaining (SFC).

In CloudSimHypervisor simulation framework, Datacentre setup represents the physical software defined network infrastructure. Since this simulation application extends CloudSimSDN-NFV the physical network and virtual network topologies are configured and deployed in simulations using JSON files or by extending the topology generators class which is an object oriented programming Java class (Son & Buyya, 2018b). The SDN Broker can also be configured and programmed to simulate datacentre and end-user characteristics. End-user requests which is used as input workload for the simulation, is provided in CSV file format. The arrival times of different users, requesting for network resources and network traffic behaviour are modelled based on the model proposed in (Ersoz, Yousif, &

Das, 2007). Every workload details a start time, a source and a destination, an SLA and a data packet size of the request.

CloudSimHypervisor deploys policies and algorithms for software defined cloud networks with the implementation of the network virtualization hypervisor. Policies for links and host selection, resource overbooking algorithms, scheduling policies and different VM allocation policies are executed in simulations and experiments by the hypervisor. The built-in policies in this simulation application are re-useable. However, the abstract classes of these policies could be modified to suit a developer's requirements (Son & Buyya, 2018b). Virtual network topology which is implemented in the upper layer of CloudSimHypervisor framework is an input of user requests and workload. Computation of packet transmission time and application execution between the VMs and networks is done by the VM management services of the simulator (Son et al., 2015) by virtue of the distributions of workloads and user requests from the hypervisor. Furthermore, the network virtualization hypervisor layer is where all the network resources are provisioned. In this layer network resources are provisioned to satisfy the demands of network users and their requests which are delivered from the upper layer according their SLAs.

Network request and allocation policies which are required to provision resources for different QoS scenarios are deployed by this layer. The network virtualization hypervisor possesses capabilities to execute some unique network virtualization attributes such as abstraction of the entire physical network resources as single set and reusing them in multiple independent virtual network (vSDN) setups. It employs virtual network isolation attribute for setting up these independent virtual networks (VNs). This implies, irrespective of the number of virtual networks which are hosted by the hypervisor, they would be no collision between them. The separation of the network control logic from the physical network infrastructure in software defined networks makes it possible for the hypervisor to assign a tenant controller to each of the independent virtual networks. Each of these tenant controllers of the virtual software defined networks (vSDNs) has a unique operating system and network policies relevant to its virtual network.

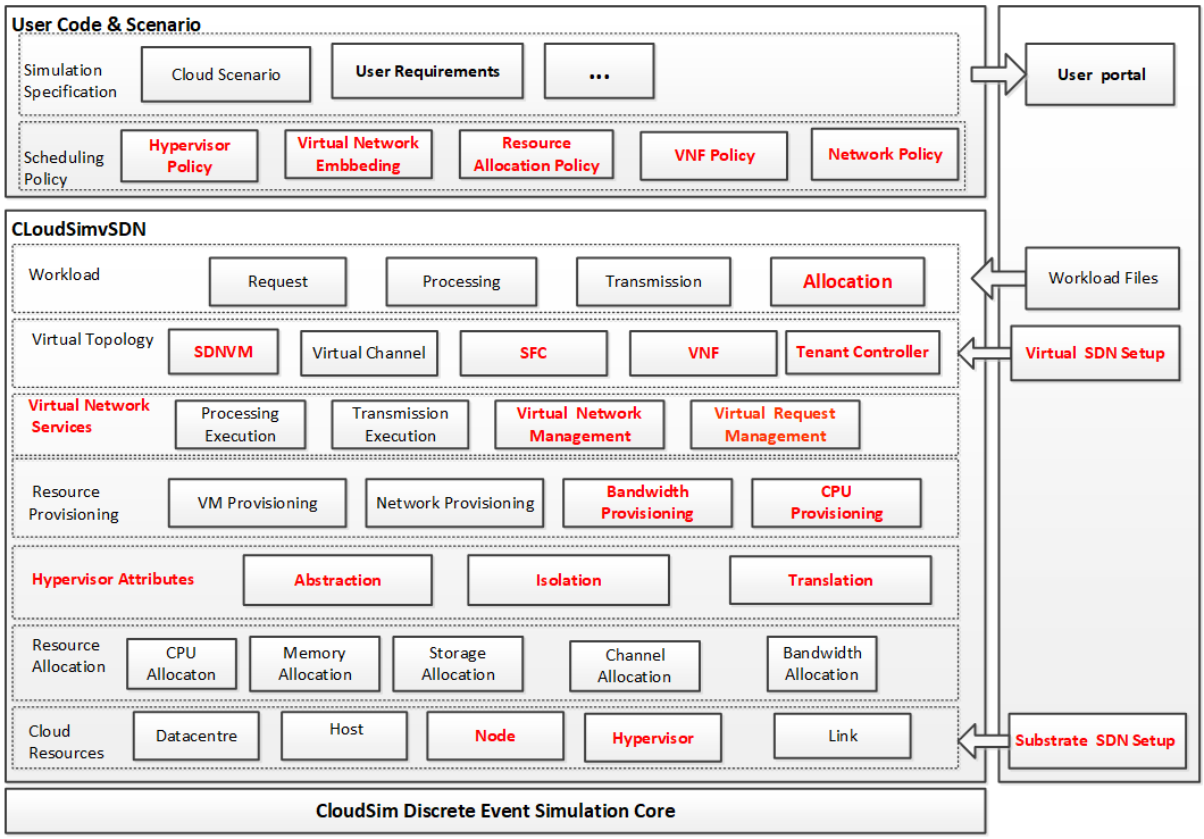


Figure.3.2 CloudSimHypervisor Architecture.

Note: The fields in red font are new fields added in CloudSimHypervisor.

The bottom most layer where the physical topology configuration of the network architecture been considered is the cloud resource layer, Figure.3.2. This layer contains modules physical network infrastructure configurations and resource allocation specifications.

3.6.1 The Core Logic of CloudSimHypervisor

The core logic of CloudSim simulates the fundamental compute elements of the cloud infrastructure. Physical hosts in CloudSim are defined with specific settings. The VMs which are hosted by these physical hosts must meet well defined requirements of CPU power, memory and storage size (Son et al., 2015). CloudSimSDN utilized other cloud computing components which CloudSim models which include datacentre, physical host, VM, VM scheduler, workload scheduler and also deployed software defined networks components. CloudsimSDN-NFV introduced network functions virtualization and service functions chaining and edge computing resources (Son and Buyya, 2019) which makes it

possible to simulate heterogeneous datacenters in different geographic domains. However, CloudsimSDN-NFV does not make provision for simulating and deploying a network virtualization hypervisor as a virtualization layer which supports abstraction of hosts and link resources a composite unit to reuse in setting up independent vSDNs. Figure.3.3 illustrates the Class diagram of CloudSimHypervisor.

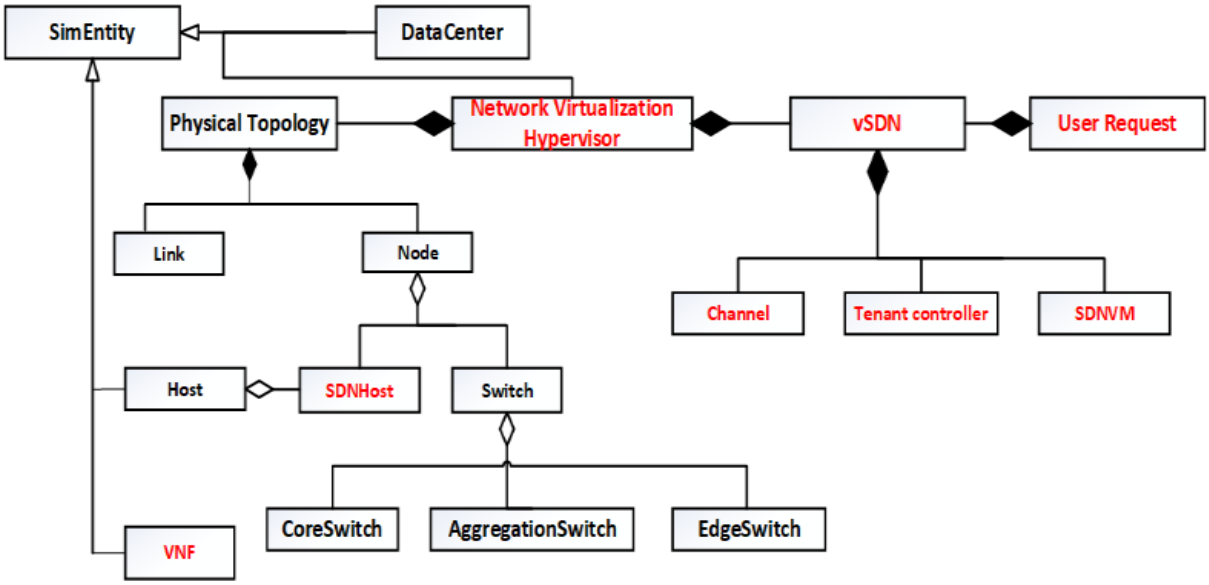


Figure.3.3 Class diagram of CloudSimHypervisor

Note: The classes in red font are new classes added in CloudSimHypervisor.

These classes compose and are contained by ◆——	this class
These classes compose without belonging to ◇——	this class
This class inherits from ——▷	this class

Table 3.1. Notations for class diagram of CloudSimHypervisor.

Although CloudsimSDN-NFV, makes provision for simulating SDN, NFV and SFC and edge computing resources, it considers VMs as applications hosted on the physical hosts of SDN and usually for mainly placement and migration. VMs and channels can be

haphazardly connected with no cognizance of being independent, multi-tenants and executing cloud solutions as isolated independent networks. With CloudSimHypervisor, we implemented modules for creating a network virtualization hypervisor as a platform which slices network resources of physical software defined networks and can be shared by these networks. It is a layer which supports co-existence and multi-tenancy of independent isolated virtual SDNs. It also supports abstraction of physical resource and stitching them into as independent virtual networks because it leverages the corporate advantages of software defined networking, SDN and network functions virtualization.

3.6.2 Network Modules

A node class was developed to facilitate simulation of packets transfer between VMs. This node which is under the control of the hypervisor performs SDN enabled functions. We developed the network virtualization hypervisor to perform the functions of the main network control. Forwarding rules are installed by the network virtualization hypervisor and can be dynamically changed depending on the network traffic. The virtual links connecting switches and/or VMs are represented with a Channel class that defines the physical path capacity of such channels. The class holds a list of physical network elements, such as switches and hosts, along with physical links between those elements. Since different virtual channels could share the same physical link, each physical link also maintains the list of channels passing through the link itself. If a new channel is created and added to the link, the link updates the shared bandwidth of all channels which passes through the link. Using the Network Hypervisor, it is also able to create dedicated channels for a specific traffic flow. As SDN allows the controller to differentiate network flows depending on the type of traffic, our framework also can create a channel for a specific flow with dedicated bandwidth allocation. In this case, an extra channel is created in addition to the default channel, and the packets with specific flow id are forwarded using the new channel.

Network virtualization hypervisor class represents the network controller managing the overall network behaviour of the simulation. It monitors all the network's channels and packets and decides how to reconfigure each network element. User-defined network policies can be developed by extending this class. It also calculates the estimated arrival time for each packet based on the allocated bandwidth for each channel and the number of packets sharing the same channel. If there is more than one channel sharing a link, each

channel size is also included in the bandwidth calculation. Functions and behaviours supported by SDN and NFV are implemented in the network hypervisor class. For example, if dynamic bandwidth allocation is necessary to be simulated, policies specifying how to allocate bandwidth to each flow are implemented in this class.

3.6.3 Packet Scheduler

We model the packet scheduler similar to the Cloudlet scheduling in the CloudSim and CloudsimSDN-NFV. In CloudSim, Cloudlet which has the length of the processing workload models computing workload for processing CPU. CloudletScheduler simulates scheduling of the processing workload in each VM based on the simulation scenario. In CloudletSchedulerTimeShared, the processor capacity is evenly shared by all Cloudlets submitted and currently processed at the VM. For example, if five Cloudlets are submitted to a VM, the CPU capacity of the VM is shared among them so that each Cloudlet can be assigned 20% of the total CPU capacity (Son and Buyya, 2019). On the other hand, CloudletSchedulerSpaceShared processes only one Cloudlet at each time so that 100% of the CPU capacity will be assigned to the first Cloudlet submitted to the VM. The other Cloudlets are in the waiting list and processed in the queue once the earlier Cloudlet completes the processing (Son and Buyya, 2019).

In CloudSimHypervisor, the network packet scheduler is designed with Packet and PacketScheduler Classes similar to Cloudlet and CloudletScheduler. Packet Class represents the network transmission workload which has the size of the network packet. PacketScheduler distributes the available network bandwidth among currently transferring Packets with the same source and destination VMs. If multiple flows share the same physical link, the bandwidth of the physical link is distributed among these flows and then PacketScheduler can allocate the distributed bandwidth onto Packets in the scheduler. Similar to CloudletScheduler, we implement PacketScheduler with two models, timesharing and space-sharing. In time-sharing, the available bandwidth is equally shared among Packets from the same source VM to the same destination VM. In SpaceShared, the entire bandwidth of the virtual network is allocated to the first Packet submitted to the network, and the rest are waiting in the queue until the transmission of the first Packet is completed.

3.6.4 Calculating Packet Transmission Time

Simulation of network requires that the transmission time for data transferred between hosts is calculated. Calculation is straightforward if the data is transmitted for one hop that is not shared with other hosts. However, it is more complicated to estimate travel time when the packet needs to be transferred to the host via multiple hops where some are shared by other hosts. In fact, data is fragmented into several packets involved in multiple fragmentation process on each network layer depending on protocols. The fragmentation processes are complicated and varied on different protocols. Therefore, since the transmission process model is simplified the estimation of transmission time is reduced. We introduce the class Channel, an end-to-end edge from sender to receiver consisting of multiple links. It is a path for data packets that are going through a series of queues of ports in different switches. The class Link is a physical medium between ports of switches or hosts. The class Transmission refers the transferring data between two hosts which travels through the channel. In each link, bandwidth is first allocated to the priority channel if SDN is configured to allocate a specific amount of bandwidth to the channel. Afterwards, the remaining bandwidth is equally shared by the channels passing through the link. Thus, allocated bandwidth $BW_{c,l}$ for a channel c in the link l is defined as

$$BW_{c,l} = \frac{BW_l}{N_l} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ldots. \ (3.1)$$

where the link (l) has available bandwidth (BW_l) shared by the number of channels (N_l). As a channel is composed of multiple links, the transmission speed of the channel basically depends on the least bandwidth among the links. Even if some links have higher bandwidth, there would be a bottleneck when packets pass through a link with lower bandwidth. Thus, for the time period Δt , when no channel has been added or removed, the amount of data Dc transferred from sender to receiver on a channel c can be calculated with the Equation

$$Dc = \Delta t \times \text{Min}\ (BW_{c,l}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (3.2)$$

When a new channel is added, Network Operating System informs all links where the new channel passes through, and existing channels are updated with a new lower bandwidth value. Channels and links are also updated when a data transmission is finished, and the allocated channel is deleted. In this case, the remaining channels will have more bandwidth

as there is one less channel using the link. Updated bandwidth values are used to calculate the size of data transferred for the next time period.

3.6.5 Abstracting User Requests

In real world network events, jobs associated with network transport can be abstracted as a combination of computation processing interspersed with transfers of packets. Considering a web service model for instance, when a request is received at the front-end server, e.g. web server, the front-end server computes the request and creates another request to the mid-tier server, e.g. application server. Similarly, the mid-tier servers process the received requests and transfer them to the back-end server, e.g. database server. Hence, to model a request which contains both workloads and network transmissions, three classes are implemented: Request, Processing and Transmission where Processing and Transmission classes are implemented with respect to the Activity interface (Son & Buyya, 2018b). Every Request is made up of a list of multiple Activity objects, which are implemented as Process computation or Transmission. Process computations comprise workload (Cloudlet), and Transmission has a network transmission requirement (Package). Network request is made up of Processing and Transmission objects which usually appear in a well-defined order. For ease of use, list of requests can be generated in a CSV format which has multiple pairs of Processing and Transmission. To logically estimate network transfer time for each packet, we made provision for Queue in nodes for each flow. For example, if a flow is set up between two hosts, the queue should be set up in the sender's host as well as in all switches that packets go through.

3.7 Validation

Validation of CloudSimHypervisor is a focal point when it comes to its relevance with regards to simulation. In order to validate CloudSimHypervisor, we conducted a series of experiments that compared CloudSimHypervisor with CloudsimSDN-NFV with the same workloads. As it was indicated earlier, CloudsimSDN-NFV is a simulation application which makes provision for simulating SDN, NFV, SFC and edge computing resources, which considers the execution of cloud applications and resources mainly on the basis of resource migration and placement of VMs in a cloud computing environment. It does not make provision for a virtualization layer which has the capability of decoupling the virtual

resource as a unit of sets and stitch them together to developing independent virtual networks capable of deploying different cloud solutions through network specific tenant controllers. Another relevant attribute of the hypervisor is its ability to isolate these independent virtual networks, a feature which prevents cyber security breaches. Our objective was first to model different scenarios of software defined cloud networks (integrated SDN & NFV) with different data sizes, testbed environment configurations and different shortest paths between the hosts and other network elements. We then analysed how close the data transfer rate between the network elements (host, links and channels) in CloudSimSDN- NFV and CloudSimHypervisor as test of the accuracy of the CloudSimHypervisor.

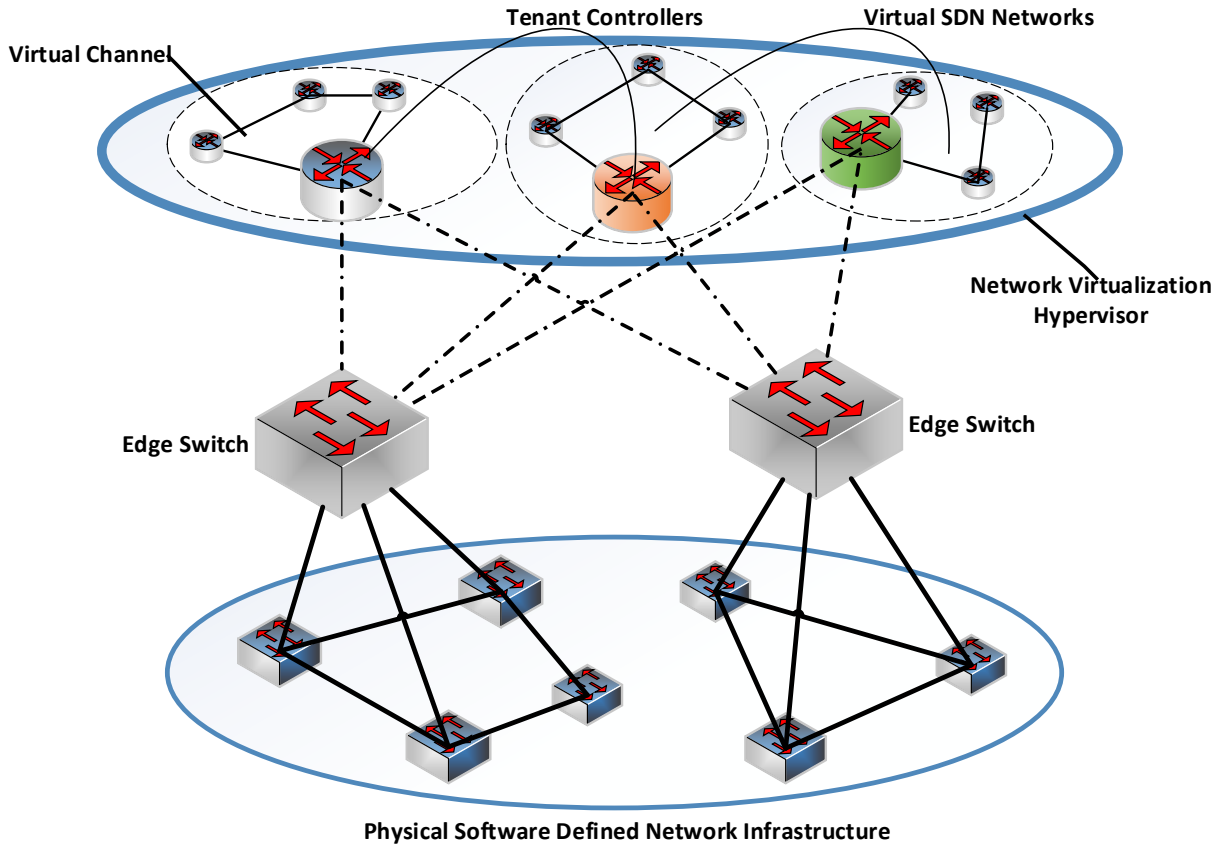


Figure 3.4. Experimental setup for validating CloudSimHypervisor.

3.7.1 CloudsimSDN-NFV Setup

Experimental setup for CloudSimSDN-NFV was done in a Java IDE using the CloudSimSDN-NFV framework (Son and Buyya, 2019). The physical network topology is setup in CloudSimSDN-NFV by creating and adding the physical host, switches (Core,

Aggregate, Edge) and links which form the SDN-enabled cloud datacentre in a JSON file. The virtual network topology which is the resource deployment request when network users send VM formation request to the cloud network resources present the topology of the virtual network with QoS requirements and Service Level Agreements, SLA. This was input as a JSON file. Workloads of network transmission and compute processing are passed from network users to VMs when the VMs are created in the SDN-enabled cloud datacentres.
In this experiment, VM is placed in each physical host in CloudSimSDN-NFV. Hence, each VM represents a physical machine. The configured link speed between core and edge switches, and between edge switches and hosts

3.7.2 Testbed Configuration

We setup a three-tier network topology which comprise two physical software defined networks with one of them having four hosts and the other three hosts as shown in Figure.3.4. The setup also has two edge switches which connect the physical software defined network to the network virtualization hypervisor. The hypervisor supports three isolate virtual software defined networks (vSDNs). Each of the vSDN comprise three virtual nodes and a network specific tenant controller. The bandwidth allocated to the links from the edge switches to the hypervisor and between the edge switch and the hosts in the physical networks are same as used for the setup in CloudSimSDN-NFV. This topology can support a number of scenarios, for instance, transferring data through dedicated routes and transferring data across random links or network elements.

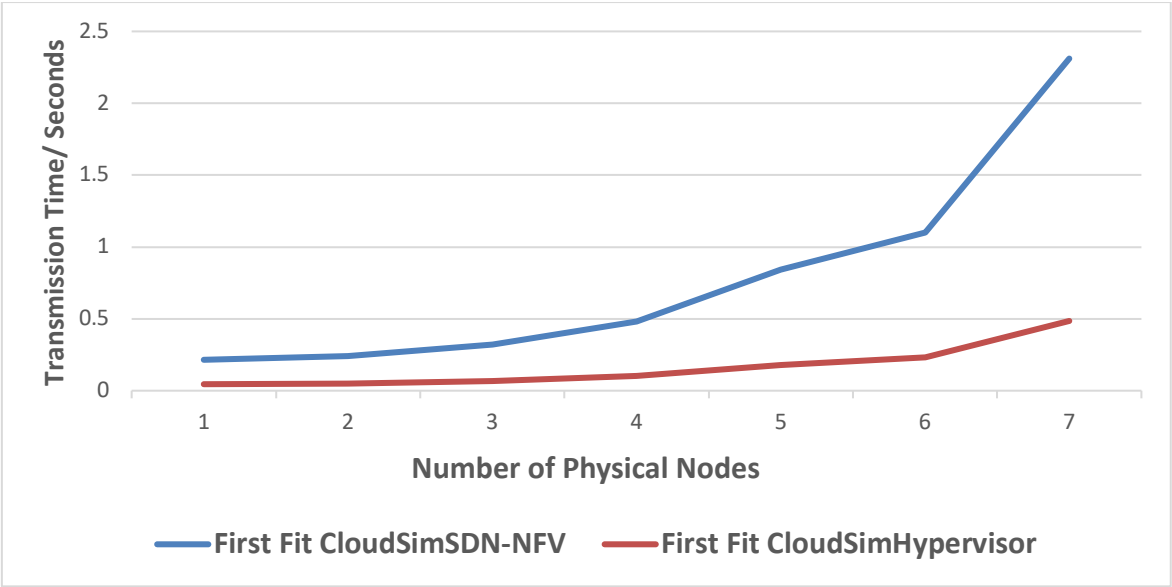
Link	**Bandwidth**
Hypervisor / Core ↔ Edge Switches	50 Mbps
Edge Switches ↔ Host	50 Mbps

Table 3.2. Link configuration for the validation experiments.

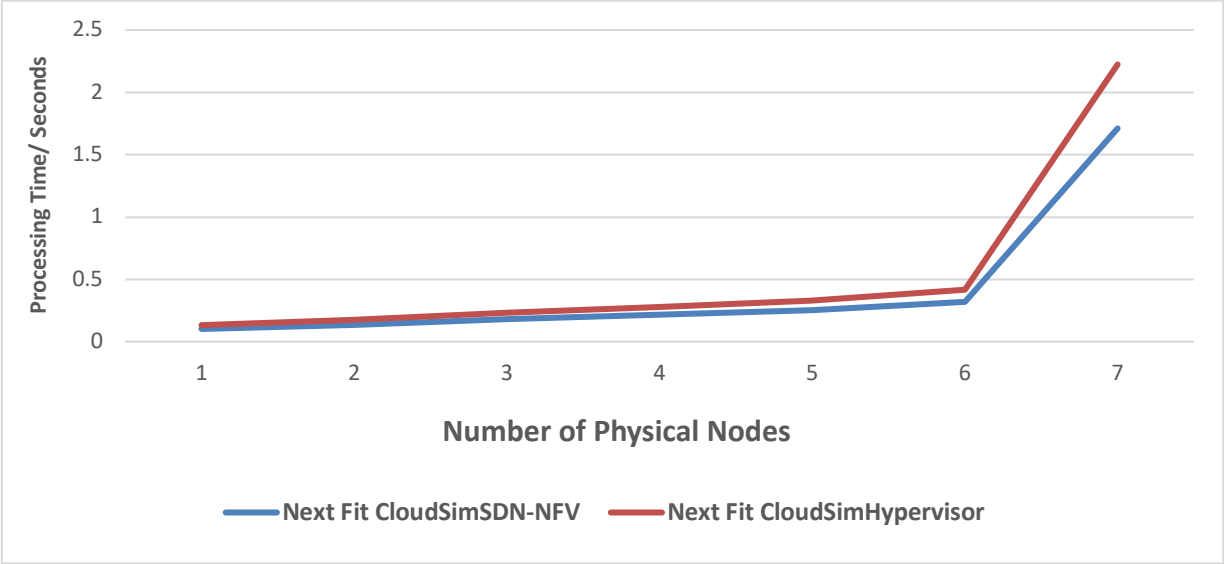
Each host physical network is configured to receive request for compute and network resources and send data with different sizes randomly to the network virtualization hypervisor through the edge nodes/ switches which enable links to be shared among multiple connections.

3.7.3 Validation Results

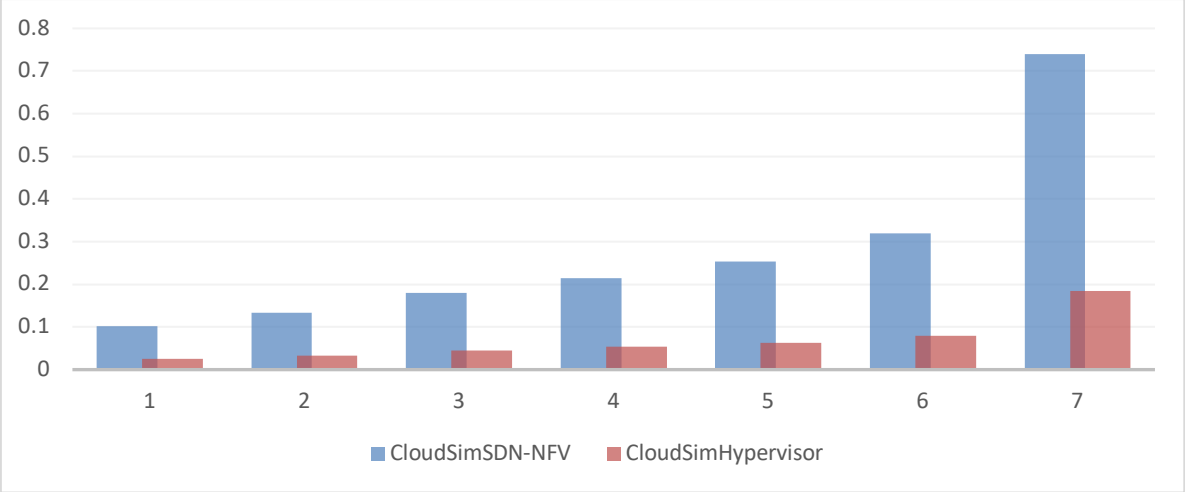
The graphs in Figure 3.5 display the measured processing time and transmission time for three experimental scenarios comparing CloudSimHypervisor and CloudSimSDN-NFV. In Scenario 1, the First fit VM placement algorithm was used in the experiment to determine the transmission time of network packets in CloudSimHypervisor and CloudSimSDN. First fit placement algorithm uses VM demand forecasting technique to place VMs on the first available physical hosts with adequate resources to support network requests. With the First Fit algorithm, the magnitude of the network request does not exceed the total capacity of the host resources and it also used to conduct experiments aimed at reducing SLA violations. The difference in the transmission speed between CloudSimHypervisor and CloudSimSDN-NFV was observed to have improved by about 21.7% when using CloudSimHypervisor to CloudSimSDN-NFV. In Scenario 2, the simulation applications were used to compute processing time in an experiment using the next fit algorithm. With the next fit algorithm, the network virtualization hypervisor receives user requests and searches for an initial physical host. If the host satisfies the resource requirements to execute the request, it is selected to replace the next host. The results for this scenario also show that the CloudSimHypervisor computed the transmission time of the processes by a wider margin, averagely about 38% compared to the CloudSimSDN-NFV. The third scenario considers random distribution of virtual network requests to the host in the physical software defined network based on the dynamic decisions of the network virtualization hypervisor. We call this method the Hypervisor fit algorithm. In this method, the hypervisor has a global or distributed view of the entire network setup and as a result is able to dynamically allocate resource to network which possess resources to specific magnitudes and micro-segments network traffic when necessary in order to balance network load and manage flow of traffic.



(a) Scenario 1



(b) Scenario



(c) Scenario 3

Figure 3.5. Comparing CloudSimHypervisor with CloudSimSDN-NFV with regards to processing speed and average transmission speed.

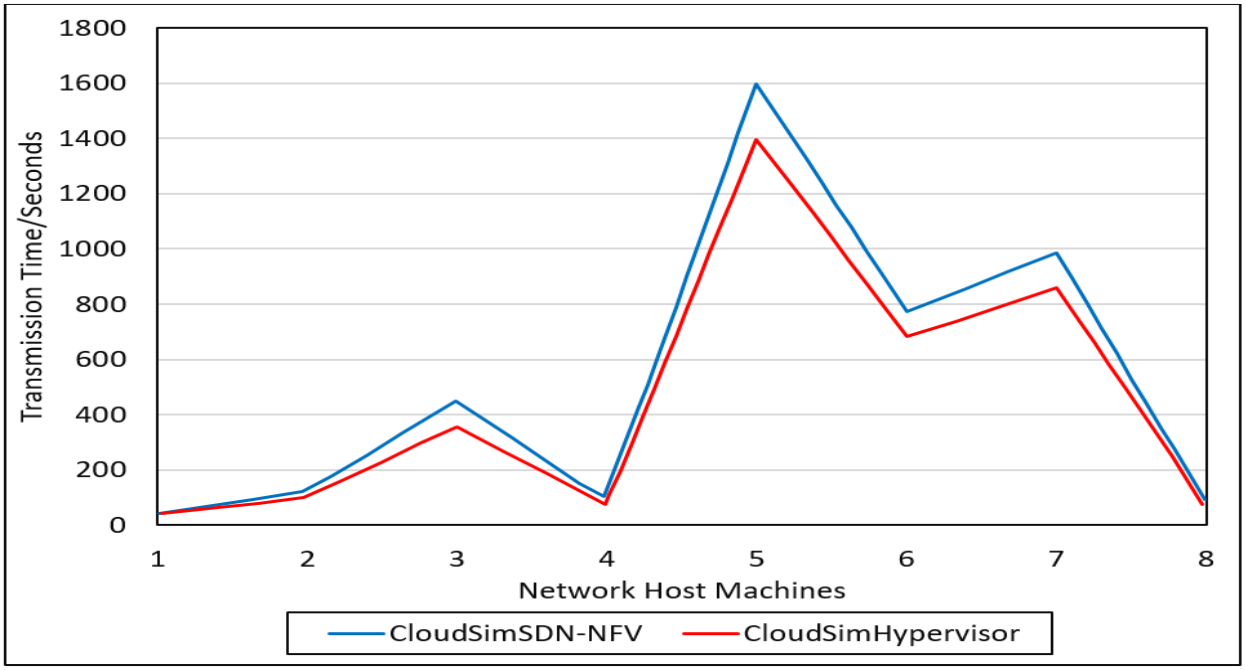


Figure 3.6. Results of comparison between CloudSimHypervisor and CloudSimSDN-NFV with regards to average transmission speed.

Figure 3.6 displays results of a validation experiment which compared how fast network packets are transmitted in a software defined cloud network when using CloudSimHypervisor and CloudSimSDN-NFV. The experiment used the details of the network setup in figure 3.4. The results showed that transmission speed of network packets is faster when using CloudSimHypervisor. This can be attributed to processing phases involved when using CloudSimSDN-NFV (Physical Host layer, edge switch layer, Aggregation Switch layer, Core Switch and then to the controller) which increase the delay in processing network user request when using CloudSimSDN-NFV.

3.8 Use Case Evaluation

This Thesis focuses on two use cases (built in the context of multi-tier web applications) to demonstrate the capabilities of using CloudSimHypervisor and to highlight the strengths of adopting the network virtualization hypervisor in software defined cloud datacentre networks. The joint computer and network resource utilization is considered first. Then the simulation application is further used to access how prioritizing network traffic would affect usage of network bandwidth and link resources considering different QoS parameters.

3.8.1 Joint Compute and Network Resource Utilization

The first use case evaluates the impact of using the network virtualization hypervisor on compute and network resources utilization in a software defined cloud network. SDN enabled cloud datacenters enhance network resource utilization and energy savings via VM consolidation procedure. Due to the consolidation of resources, switches and host which are unused can be turned off by the SDN controller. However, for software defined cloud networks (SDCN) which deploys a network hypervisor which is able to support several independent isolated virtual networks from multiple user requests and efficiently allocate these requests to the appropriate host compute and network resources through virtual network embedding enhances efficient use of these resources. The network virtualization hypervisor through the network specific tenant controllers of the independent virtual networks, accurately map or allocate network requests with specific requirements to the appropriate compute and network components which have adequate available resources to process these requests. The network hypervisor at any point of execution is updated with available resources of the entire network and the resources required for executing all pending tasks. The allocation of request to available resources is done through allocations policies

executed by the hypervisor with regards to certain performance parameters such as response time, transmission time and queuing time using network traffic consolidation technique. Due to how efficient the network virtualization hypervisor allocates resources in SDCNs, there is efficient usage of hosts and switches resources with accurate measure of workload allocation and are turned out when necessary hence maximizing energy efficiency in cloud datacentre networks.

i. Setup for Experiment 1

The first use case evaluates the impact of using the network virtualization hypervisor on compute and network resources utilization in a software defined cloud network. In this scenario, we setup a large-scale cloud datacentre with 100 physical machines connected through 20 edge switches. The physical machines are configured with 16, 24 and 32 cores, and each core has 10,000 MIPS capacity. The network bandwidths of all links between switches and physical machines were equally set to 2Gbps. The network virtualization hypervisor is setup to receive workloads from network user requests which consist of CPU processing and network transmission requirements and then pass it to the physical networks for execution through the edge switches.

For the virtual topology, 500 VM creation requests of which 100 are tenant controllers were randomly generated based on selected VM types specified in Table 2. The tenant controllers in the simulation are configured to mediate between the set of VMs forming a particular virtual network and the hypervisor. Each of these requests has a different start time and lifetime following exponential distribution and Pareto distribution respectively (Ersoz et al., 2007) which we adopted from (Son et al., 2015). To ensure that switches are working throughout the VM lifetime, network workload was also created for the execution time of VMs (Son et al., 2015).

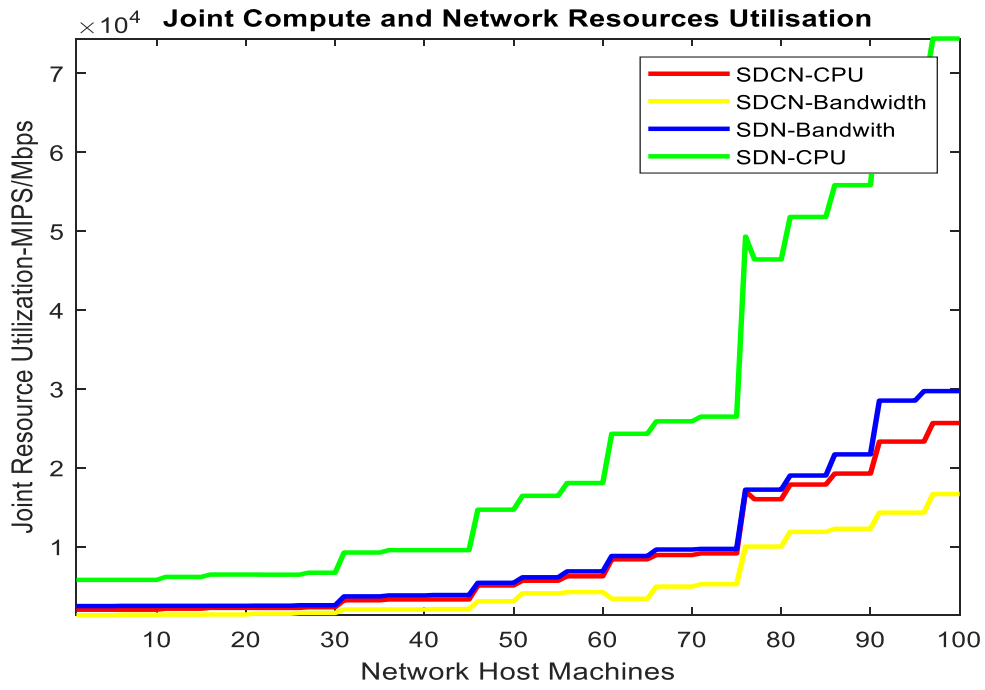
VM Type	Cores	MIPs	Bandwidth
Tenant controller	16	9000	500 Mbps
Web Server	4	2000	100 Mbps
App Server	8	1500	100 Mbps
DB Server	12	2400	100 Mbps
Proxy	8	2000	500 Mbps

Firewall	8	3000	500 Mbps

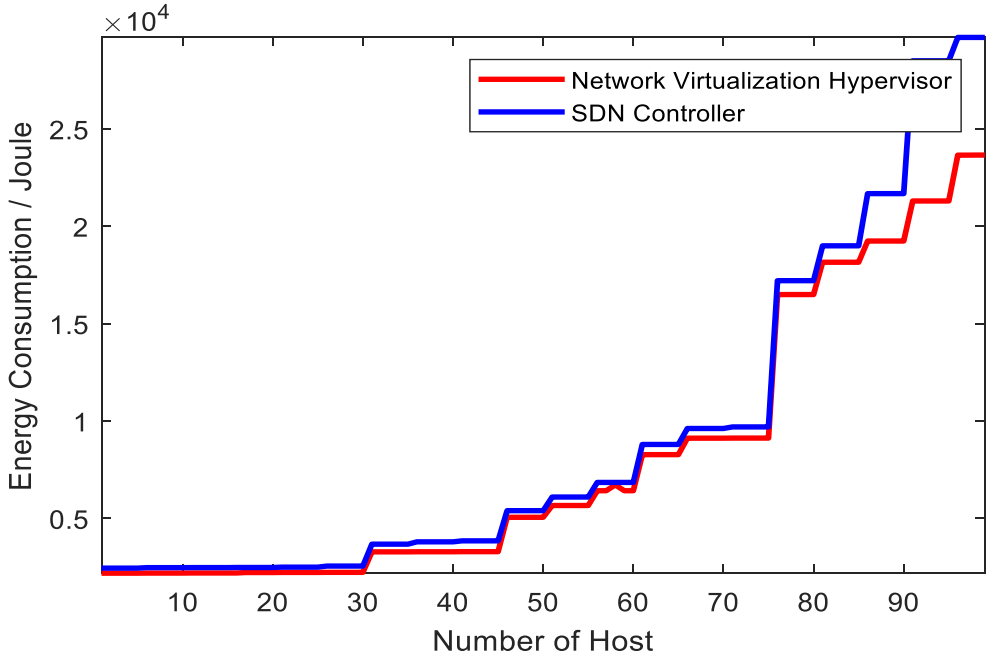
Table 3.3. VM specification for joint compute and network resource utilization use case.

	Distribution	
Request inter-interval times	Log-normal Distribution	μ=1.5627, α=1.5458
Packet sizes	Log-normal Distribution	μ =5.6129, α =0.1343 (Ch1) μ =4.6455, α =0.8013 (Ch2) μ =3.6839, α =0.8261 (Ch3) μ =7.0104, α=0.8481 (Ch4)
Workload sizes	Pareto Distribution	Location=12.3486, Shape=0.9713

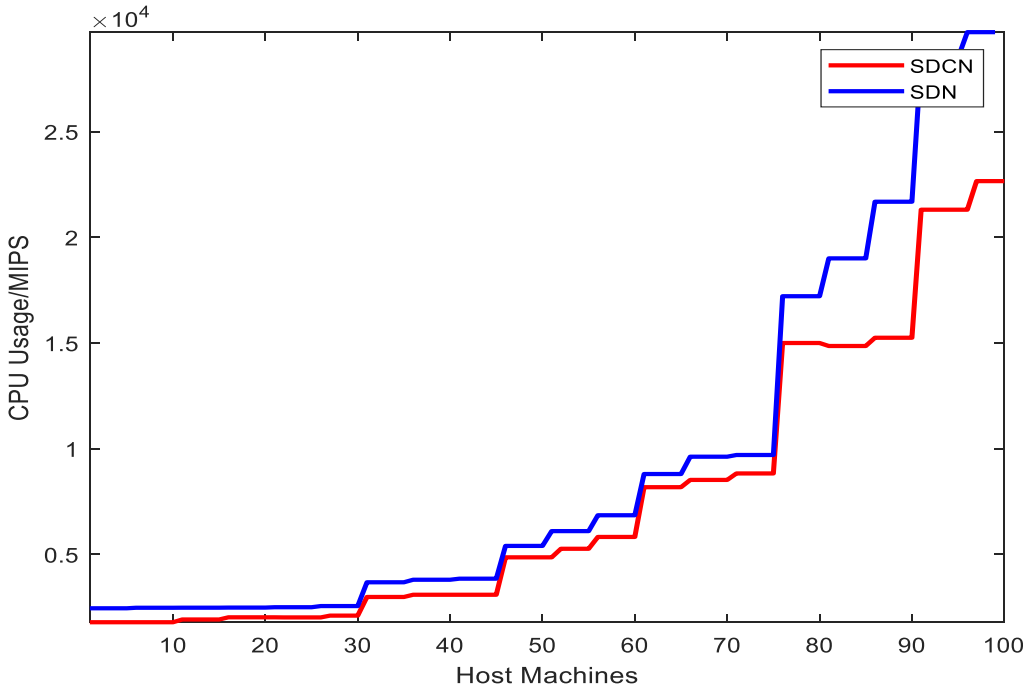
Table 3.4. Characteristics of network requests based on the model proposed by (Ersoz et al., 2007).



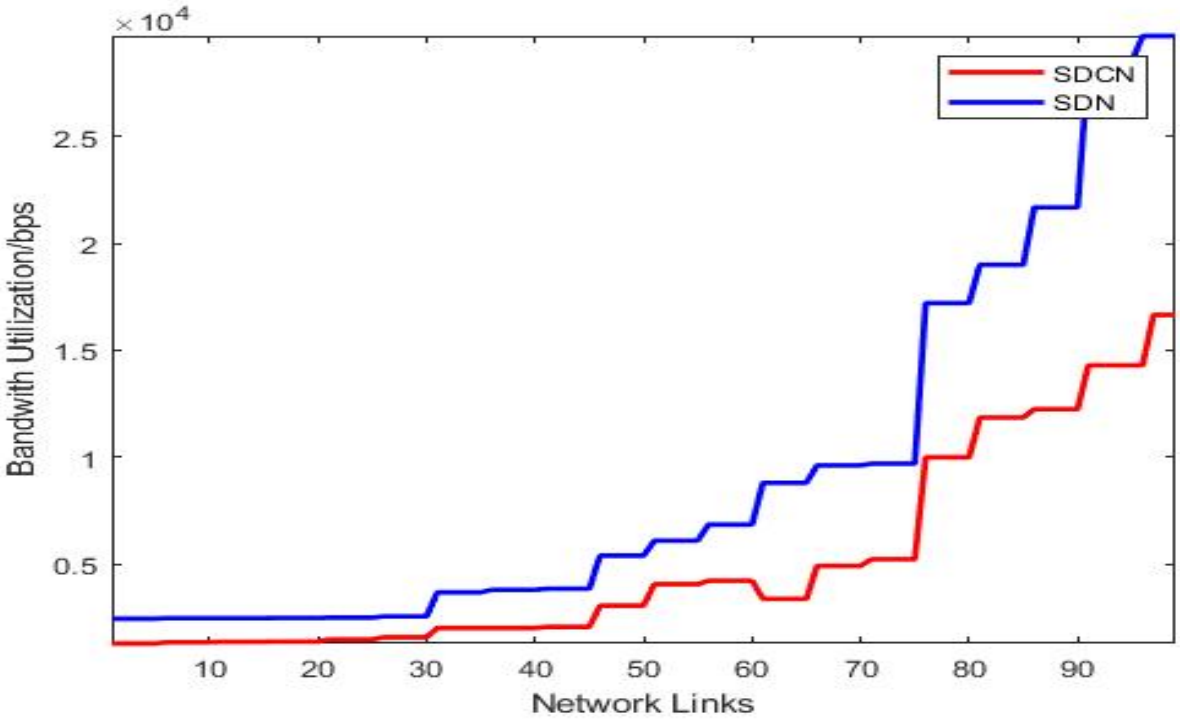
(a)


×10^4
Network Virtualization Hypervisor
SDN Controller
Energy Consumption / Joule
2.5
2
1.5
1
0.5
10
20
30
40
50
60
70
80
90
Number of Host

(b)


×10^4
SDCN
SDN
CPU Usage/MIPS
2.5
2
1.5
1
0.5
10
20
30
40
50
60
70
80
90
100
Host Machines

(c)



(d)

Figure 3.7. Diagrams showing efficient utilization of compute and network resources with the implementation of the Network Virtualization Hypervisor.

The performance of the network virtualization hypervisor is assessed based on CloudSimHypervisor simulation framework by using it to setup and conduct first an experiment of joint compute and network resource utilization. In Figure 3.7a, we compare usage of CPU and bandwidth of a software defined cloud network (SDCN) setup which has 100 host machines implements a hypervisor and compare it with a software defined network (SDN) which implements a controller with same number of host computers. Results show that the network virtualization hypervisor utilized less amount of both CPU and bandwidth to execute the same workloads as the software defined networking controller.

3.9 Traffic Prioritization

Prioritizing network traffic based on the user type was difficult due to complexity and overhead of configuring network elements in traditional cloud datacentre networks. Software defined networking provided a dynamic cloud environment which enhanced this challenge through network traffic consolidation and VM placement techniques (Son et al., 2015). However, the emergence of network virtualization in software defined cloud

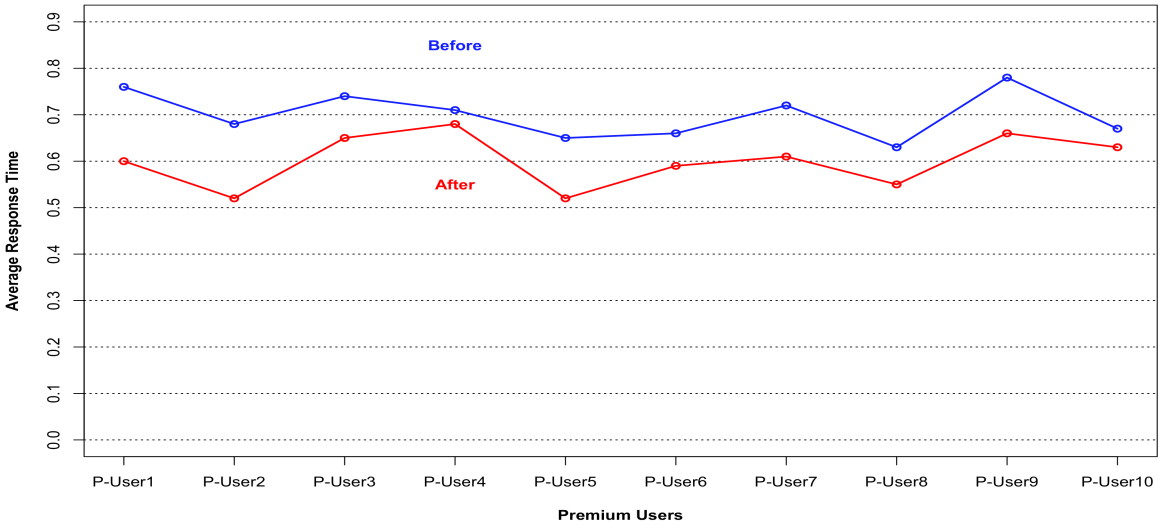
networking environments and the use of the network virtualization hypervisor which enables multiple independent virtual networks to co-exist and share the same physical network infrastructure introduce a dynamic priority-aware network request and traffic micro-segmentation and bandwidth allocation for different network user types with efficient QoS delivery.

Each network user request consists of a random number of VMs and flows with thorough specification. Priority of a request is accessed from the specification of the VM and flows by the network virtualization hypervisor. Based on the analysed information, the network hypervisor dynamically implements the appropriate host and link selection algorithm for bandwidth allocation and flow scheduling to execute requests for all network user types.

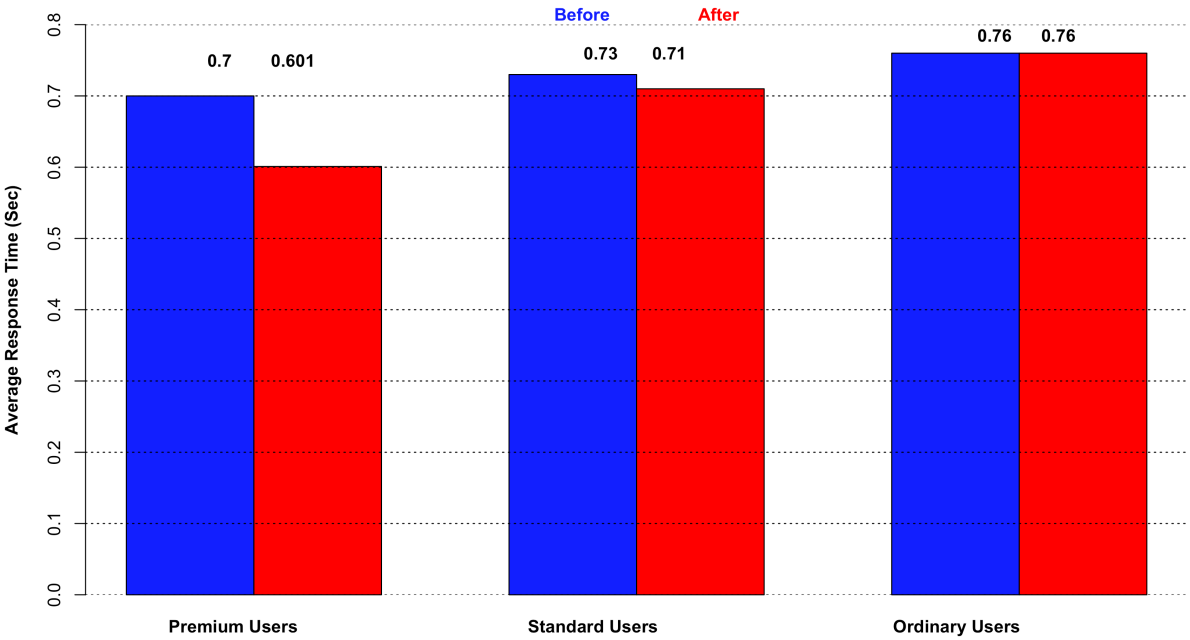
i. Setup for Experiment 2

In this scenario, we simulate a cloud datacentre network with 100 physical machines connected through 10 edge switches. The physical machines are configured with 16, 24 and 32 cores, and each core has 20,000 MIPS capacity. The network bandwidths of the links of the host-network were randomly allocated at 1Gbps, 2Gbps and 4Gbps with each having a 0.5 msec latency. In the simulation environment, the network virtualization hypervisor is able to create different channels for data flows in order to provide priority network traffic with the additional bandwidth demand, which implies virtual channels (links) connecting the VMs are dynamically set to differentiate higher priority flows over normal flows for all network user types; premium, standard and ordinary. Standard channels by default evenly distribute all packets and transfer data in the channels when there's no network traffic prioritization. However, the network hypervisor allocates a specific amount of bandwidth is exclusively for the priority channel, when traffic prioritization is required. And hence the bandwidth in these channels is not available for the other channels.

Different workloads are generated for each user using the web model (Ersoz et al., 2007). Table 3 displays the characteristics of the data which was used for the evaluating the accuracy of the CloudSimHypervisor with regards to network traffic prioritization. Each of these requests has a different start time and lifetime following exponential distribution and Pareto distribution respectively. 30 cloud infrastructure customers for the experiment have been used, 10 of them were premium users, 15 standard, and 5 ordinary users.



(a)



(b)

Figure 3.8. Effect of Network Traffic Prioritization.

Figure 3.8a displays details of performance improvement for premium users with the implementation of network traffic prioritization. The average response times for network request for all Premium Users received an improvement of about 23.75%. This also implies that the network virtualization hypervisor is efficient at dynamically allocating network

bandwidth per- flow to users an attribute which is important in enhancing QoS in software defined cloud networks.

Figure 8b, shows that average response time of Premium User requests decreased from 0.7 seconds to 0.6.1 seconds which is an average of about 14.143% performance enhancement. The figure also shows an improvement of approximately 2.74% in the response time for Standard User. However, the response time for Ordinary User, remained the same. This explains that with the implementation of the network virtualization hypervisor in software defined cloud networks, Cloud Service Providers would be able to efficiently deliver services to users with different QoS requirements.

3.10 Summary

Advances in technology has introduced innovative applications of the concept of virtualization in computer networking in cloud computing environment. This has enabled cloud service providers to implement cost efficient and scalable datacentre networks which support multi-tenancy of heterogeneous networks and virtual service functions which optimizes network resource utilization. SDN, NFV, edge computing and cloud computing are the technologies which are mainly involved in this innovative transformation in the technology industry in the past decade. Given that it's expensive to acquire, setup and deploy, these technologies, in this paper we present a simulation framework, CloudSimHypervisor which can be used to test, cross-check and re-check several large-scale heterogeneous software defined cloud networks in a controlled environment before rolling them out in real life.

CloudSimHypervisor was developed by extending CloudSimSDN-NFV framework which is an extension of the well-studied and used CloudSimSDN and CloudSim toolkits. Experiments to validate the accuracy of our simulation framework depicted that it is comparable to the CloudSimSDN-NFV. The validation experiments also showed that CloudSimHypervisor has additional functions and resources for instance the network virtualization hypervisor which when implemented in software defined cloud networks, optimizes network resources utilization which are not available in other cloud environment simulators.

Chapter 4 SLA-Aware and Energy Efficient Intelligent Network Resource Overbooking

4.1 Introduction

It has become overwhelmingly difficult to manage legacy computer networks especially in terms of scaling to the demands of cloud networks and datacentres. However, leveraging the combined advantages of software defined networks (SDN), network functions virtualization (NFV) and edge computing in a cloud computing environment address these shortcomings of legacy networks. In traditional network, distributed routers are core controllers for network management, while the routers can cooperate with each other by communicating network information, the decision is made by a single router with its discrete control logic without consideration of the entire network.

In contrast, Software defined technologies provide programmable network control logic which is capable of managing and segmenting the global network traffic of homogenous and heterogeneous networks. Hence, traffic consolidation and micro segmentation can be executed by the programmable control logic to enhance performance efficiency, energy consumption and SLA satisfaction for entire software defined cloud networks and datacentre networks. Network user requests with varying QoS requirements are received by the network virtualization hypervisor which has updated information of available network resources and the estimated resources required to execute the requests that it has received. The hypervisor further consolidates the traffic based on the SLA of the users requesting for resources from the cloud datacentre. These network transmission activities cannot be executed using traditional legacy networks as their control logic in the distributed router considers only local impact for the control decision and has limited resource capacity.

Furthermore, the network virtualization hypervisor has the ability to jointly provision virtual networks as function, a key function for datacentre optimization. VM consolidation process and micro-segmentation of network traffic considering the global view of the entire datacentre are also executed by the hypervisor. The network virtualization hypervisor's control of the network manages each of the network devices overall network. It discharges the forwarding rules of the data packets in the forward plane which form the physical network infrastructure of the network. With the hypervisor's ability to dynamically alter the network configuration without altering the network setup, the network becomes easily adjustable to the changes in the network condition. For instance, dynamic bandwidth allocation for a specific network flows can be enabled to improve QoS of mission critical networks.

Overbooking of resources (Moreno & Xu, 2012), (Baset, Wang, & Tang, 2012), (Moreno & Xu, 2011), (Lowe, 2013) is a conventional technique which Cloud Service Providers (CSPs) adopt to extend the utilization of network resource in datacentres while limiting the number of physical network components that are in use and powered on. This technique involves assigning more resources such as bandwidth, CPU and memory than are actually available in the physical or host network infrastructure to satisfy the demands of network user requests. i.e. enabling the network virtualization hypervisor to map more virtual network requests onto the physical network infrastructure than the available resources (F Caglar & Gokhale, 2014). This procedure is viable for Cloud Service Providers as service users usually overestimate the resource requirements for their services or applications; meanwhile they utilize just a fragment of the resources they pay for.

Resource overbooking techniques such as transparent page sharing, memory ballooning, swapping to disk and memory compression are methods that well known hypervisors which support configuration for overbooking ratios for compute virtualization e.g. Xen, KVM and VMware ESX server use for memory overbooking (Waldspurger, 2002a). However, Cloud Service Providers who are providing solutions for mission critical datacentres have to adopt overbooking strategies which are appropriate for meeting the demands of their service users in real time. These strategies are of the type which processes network user requests in batches as a result require a high and reliable throughput.

4.2 Related Work

This section compares a number of research work which have thoroughly explored energy efficient cloud resource management. Our focus in this chapter, would be on works with the SDN-based virtualized cloud networks frame of reference (Beloglazov, Buyya, Lee, & Zomaya, 2011) .

ElasticTree is an OpenFlow based network power monitor which is used to effectively change network data traffic and regulate network elements in datacentres for the purpose of energy saving(Therrien et al., 2012) . Elastic-Tree has the capability to consolidate network flows to a limited number of network links. The unused network switches are turned off to conserve energy consumption. The authors also considered robustness of the network and how well it handles traffic surges in further research. Although ElasticTree addresses issues regarding network power savings it does not consider isolated heterogeneous virtual networks deployed by network virtualization hypervisors in cloud networks.

Abts et al., (2010) asserted energy consumption in Datacentre networks can be proportional to the volume of network data traffic just as a CPU of a computer which is underutilized consumes less energy. They suggested a link rate adaptation that changes dynamic range with regards to the predicted network traffic load. They demonstrated that energy proportional networking is attainable with agile individual link rate adjustment. However, their submission did not consider a network traffic micro-segmentation and turning off links which are not in use in specified instances.

Wang et al., (2012) suggested an approach which is similar to ElasticTree. They applied correlational analysis of traffic flows in computer networks to minimise energy consumption in datacentres. In their approach, network traffic which are found to follow a less correlated pattern is consolidated into like network links of the same kind to secure efficient energy consumption. CARPO further asserted, that the link speed of the network ports be adjusted with regards to the volume of network traffic such that a decrease in network traffic would correlate with a slow network link speed to conserve more energy in the datacentre. This phenomenon is known as link adaptation

In recent times, a number of researchers have considered extending their research scope to include both DCN and host optimization simultaneously. (Jiang, Lan, Ha, Chen, & Chiang, 2012) assessed how VM placement and network routing problem can jointly reduce the cost of network traffic in datacentres by leveraging Markov approximation to develop an algorithm to find a near optimal solution in relevant time. They formulated the VM placement and routing problem using on-line algorithm in dynamically changing traffic loads and solved the joint optimization problem.

The investigations of (Jin et al., 2013) also accounted for the optimization of energy consumption for joint host and network factors. They used integer linear programming to formulate a joint host-network problem which remodelled VM placement problem to a routing problem in multi-objective optimization problem. They finally used the depth-first search a defining approach to determine the host which is most suitable for placing VMs. A Prototype for their investigation was implemented with an OpenFlow based system which was used to conduct several simulations and assess multiple test cases.

Optimising VM placement and network routing using VMPlanner was presented by (Fang et al., 2013) in their work in which they addressed the energy saving problem with three algorithms. With VMPlanner, VMs with higher mutual network traffic are grouped and they are assigned to the same or similar racks. The flow of traffic is then aggregated to reduce the intensity of the network traffic between the various racks so that the vacant switches can be powered off. The algorithms which were used for their investigations are distance-aware VM-group to server-rack mapping, traffic-aware VM grouping, and power-aware inter-VM traffic flow routing.

Zheng et al., (2014), suggested PowerNetS, which determines optimal VM placement using correlation between various VMs with considering joint host and network resources. The coefficient of correlation between traffic flows is evaluated using PowerNetS and it is used for traffic consolidation and VM placement. They also presented a detailed power model which included power consumptions of chassis, switch plus individual active and idle ports to enhance power consumption of a server.

Contrary to the procedures mentioned above, this research proposes an intelligent self-configuring overbooking strategy which autonomously changes with regards to workload in real-time. This assures that both SLA satisfaction and energy efficiency is maintained with unpredictable changes in workload, datacentre status, and network user demands.

4.3 Power Formulation

The energy efficient host-network resource allocation problem can be composed as a multi-commodity problem (Heller et al., 2009). The purpose of the problem is to minimize the power consumption of hosts, switches and links in a datacentre.

4.3.1 Power Models

The following notations are used for formulating the power model.

- s_i*: The ith switch in the datacenter;*
- l_i*: The ith link in the datacenter;*
- h_i*: The ith host in the datacenter;*
- R_i *: The jth virtual network request to host i;*
- *rd($VM_{j,i}$): The resource demand $VM_{j,i}$;*
- $\mathrm{C}(h_i)$*: The capacity of host i;*
- $\mathrm{C}(l_i)$*: The capacity of link i;*
- $rd(R)$*: The resource demand of R;*
- $f_{j,i}$*: The flow j on link i;*
- $d(f_{j,i})$ *: The data rate of flow j on link i;*
- $|R|$ *: The total number of virtual network requests to the datacenter;*
- $|H|$*: The total number of hosts in the datacenter;*
- $|L|$*: The total number of links in the datacenter;*
- R_h *: The number of virtual network requests allocated to host i;*

- n_i: *The number of flows assigned to link i;*
- $\mathrm{CC}(X, Y)$: *The Correlation Coefficient between two variables X; Y;*
- $P(h_i)$: *Power consumption of host i;*
- $P(s_i)$: *Power consumption of switch i;*
- P_{idle}: *Idle power consumption of host;*
- P_{peak}: *Peak power consumption of host;*
- u_i: *CPU utilization percentage of host i;*
- P_{static}: *Power consumption of switch without traffic;*
- P_{port}: *Power consumption of each port on switch;*
- q_i: *The number of active ports on switch i;*

Power consumption of host *i* is modelled based on the host CPU utilization percentage (Pelley, Meisner, Wenisch, & VanGilder, 2009):

$$P(h_i) = \begin{cases} P_{idle} + \left(P_{peak} - P_{idle}\right).u_i & if\ R_h > 0, \\ 0 & if\ \ R_h = 0 \end{cases} \quad (4.1)$$

Idle power consumption is constant factor for network hosts irrespective of the volume of workload it receives from the Network Virtualization Hypervisor. It is reduced only if the hosts are turned off. However, network hosts utilize more energy when the Network Hypervisor(s) sends more workload to it for processing which eventually impels higher CPU utilization.

In this research study we adopted linear power model described in (Heller et al., 2009). As hosts' network components are homogeneous, power consumption of a host will be same as another if the CPU utilization is same.

Power consumption of switch *i* is calculated based on the active ports (X. Wang et al., 2012):

$$P(s_i) = \begin{cases} P_{static} + P_{port} \cdot q_i & \text{if } s_i \text{ is on} \\ 0 & \text{if } s_i \text{ is off} \end{cases} \quad (4.2)$$

Similar to host's energy consumption, there is a static part of power usage in a switch regardless of the network traffic that it receives from the Network Hypervisor.

Aside the static consumption, the switch consumes more energy when more of its ports are active and receive traffic from the network virtualization hypervisor. We use the linear model addressed in (X. Wang et al., 2012), where energy consumption of a switch is proportional to the number of active ports in the switch.

The problem is to jointly optimize the host and network energy consumption in the time periods as discussed in this chapter. The Network Virtualization Hypervisor allocates virtual networks' requests $|R|$ to hosts $|H|$ for the time period where $|L|$ links are connected.

$$\textbf{Minimize} \quad \sum_{i=1}^{|H|} P(h_i) + \sum_{i=1}^{|S|} P(S_i) \quad (4.3)$$

subject to:

$$\sum_{i=1}^{|H|} R_h = |R| \quad (4.4)$$

Total virtual network requests allocated to host networks in the datacenter

$$\forall h_i \in H, \sum_{i=1}^{|R_h|} rd(R_{hi}) \leq \mathrm{C}(h_i) \quad (4.5)$$

Total bandwidth capacity of links in the datacenter where (f_{ji}) is flow through the link.

$$\forall l_i \in L, \sum_{i=1}^{n_i} d(f_{ji}) \leq \mathrm{C}(l_i) \quad (4.6)$$

$$\forall i, \sum_{j=1}^{|R|} \emptyset_{i,j} = 1, \qquad (4.7)$$

where

$$\emptyset_{ij} = \begin{cases} 1 \;\; if \;\; R_h \;\; is\, allocated\, to \;\; h_i \\ 0 \;\; otherwise \end{cases} \qquad (4.8)$$

The objectives of this power model are to reduce total energy consumed by host and switches in software defined cloud datacentre network while minimising SLA violations concurrently. SLA violation in this work, is the measure of the percentage of user requests which exceeds the expected response time. To obtain this quantity, we estimated the network response time of each user request using a baseline algorithm without overbooking to compute the number of user request which are violating SLA. The constraints that were considered for the experiment are the resource required by the compute components of virtual network request allocated to the physical host by the network virtualization hypervisor which should not be greater that the capacity of the host and the bandwidth requirements of the data allocated to links should not be greater than the capacity of the link of the physical networks.

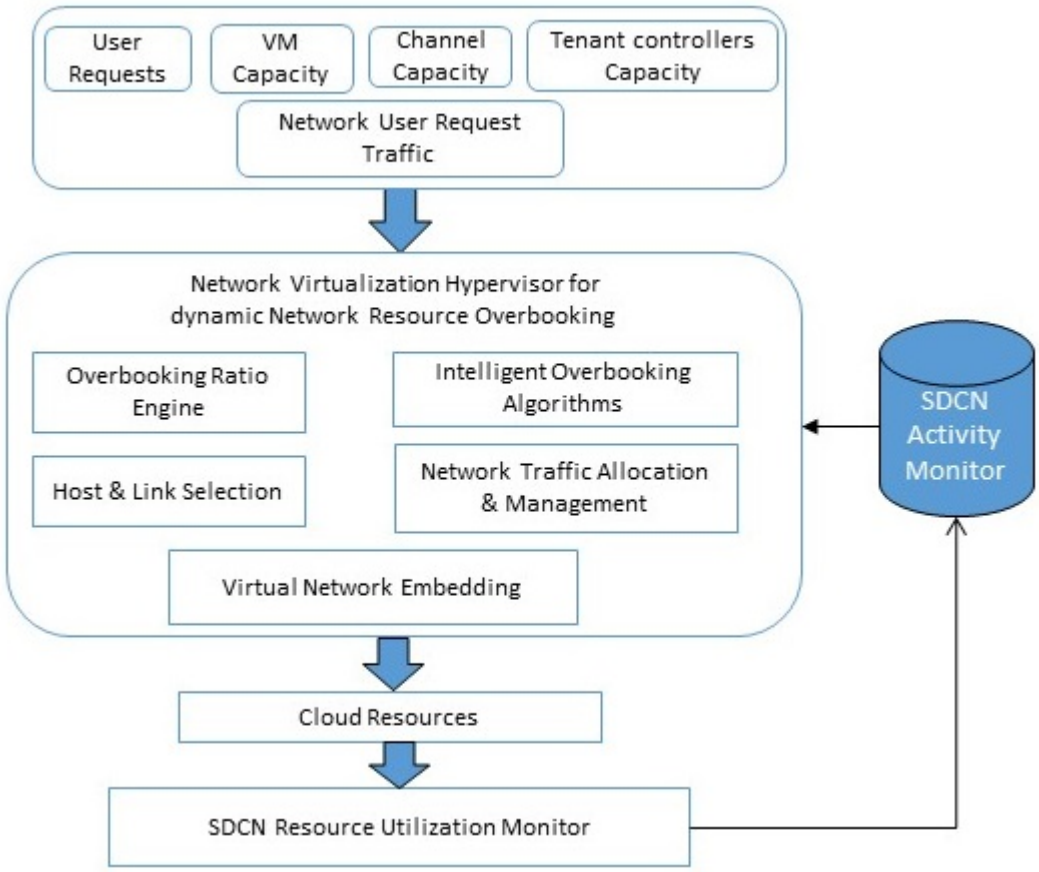


Figure 4.1 Network Resource Allocation System

4.4. Resource Allocation Architecture

The resource allocation architectural framework in Figure 4.1 was proposed to maximize energy saving and concurrently minimize SLA breaches without pre-knowledge of the network workload in the cloud datacentre with the application of network resource overbooking strategy. The Overbooking hypervisor is responsible for managing all the processes and components required for effective implementation of the strategy. Some principal processes the architecture executes are host and link selection policies and network traffic allocation and management policies. The host and link selection policies decide the preamble for allocating user requests to appropriate host network resources whilst traffic allocation and management determine the destination of host and link network resources to allocate compute and bandwidth requests to limit SLA violation.

Accurate overbooking ratios are required to efficiently execute both processes. These ratios are derived with the help of the overbooking engine which analyses the correlation between information from compute and network resource utilization. Correlation analysis data use data collected from monitoring utilization of host network resources, distribution of network user requests and traffic management by the network hypervisor in a cloud datacentre network. The consolidation policy uses current link traffic and host utilization of network user requests and flow allocation and consolidation.

SDCN Resource Utilization Monitor: This component of the network resource allocation system, monitors the levels of utilization of network resources in the SDCN. The resource utilization monitor collects utilization metrics used to monitor network resources and the utilization information of these resources in software defined cloud DCNs. Information of host network resource components such as CPU of each host and bandwidth of the links between switches and that of virtual network components such as virtual network requests, virtual network embedding and network traffic consolidation and allocation.

4.5. Network Resource Overbooking Algorithm

Consolidation of network traffic, user requests and VMs to a smaller number of physical network resources and turning off the unused ones is a useful activity to conserve energy in homogeneous and heterogeneous cloud datacentre networks. Most large-scale datacenters may consist of heterogeneous devices with different batches of servers and switches, in a single rack. The network virtualization hypervisor allocates user requests with the specified resource requirements such CPU core, memory, disk-space and bandwidth to available network host resources which are usually over-provisioned. To conserve the amount of energy that these host resources consume in the datacentre, the network hypervisor can consolidate these user request and network traffic a limited number of these host resource and then turn off the ones the ones which are left unused.

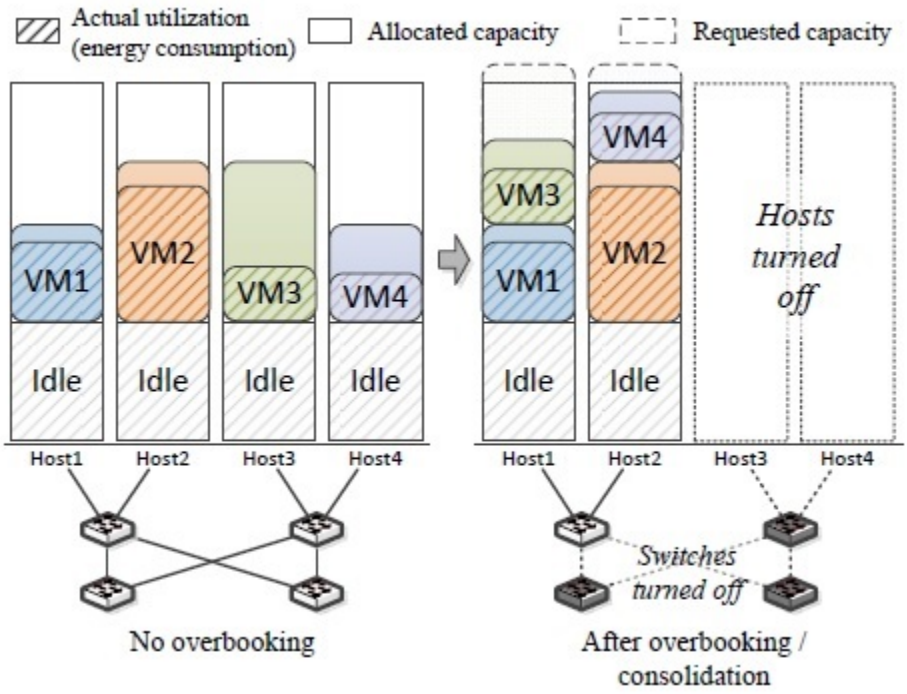


Figure 4.2. Demonstration of network events consolidation and Resource Overbooking

Prior to consolidation of network events such as (user data request and network traffic) and overbooking of host resource overbooking, the network virtualization hypervisor allocates network user requests (events) which comprise, VMs and virtual channels to all available and vacant host network resources in the datacentre. As all the host resources are in use and processing requests from the hypervisor, they all consume electric energy which contribute to high power consumption in the datacentre. From Figure 4.2., the resource requirements of the events allocated to Host 3, and Host 4 is low and hence lower resource utilization compared to the resources requirements of the events allocated to Hosts 1 & 2. When overbooking is implemented, the network hypervisor consolidates the network events with low resource requirements and intelligently allocates them to hosts network components which may already be in use but still have enough resource to execute them (Host 1 & Host 2) as displayed in Figure 4.2. and hence making vacant Host 3 and Host 4. These vacant and unused hosts network components are then turned off to reduce energy consumption in the datacentre.

To assess the network resource allocation problem discussed in the section 4.3 we apply:
The Network traffic consolidation and migration method, which is a procedure where the network hypervisor consolidates network traffic and allocates them to suitable host network

components which possess adequate resources to execute the request base availability or SLA's. The intelligence features of the network virtualization hypervisor enables requests which are allocated to network components which are overbooked to be returned to the hypervisor for reallocation. In the traditional cloud datacentre networks, a similar process is carried out, however, that focuses on migration of VMs from overloaded or underutilized host network components.

It is worth noting that distinct algorithms could be used for each of the chosen stages and hence some studies propose an integration of these methods.

The following subsections explain the joint consolidation and migration algorithm which was used for this study:

4.6. Joint VNR Consolidation and Migration Algorithm with Dynamic Overbooking

The Network Virtualization Hypervisor has a global view of the entire datacentre network and as a result has knowledge of the host network components and available resources but does not have prior knowledge of volume of virtual network requests (VNR) and resource demand, workload and data rates of the network traffic. Thus, this algorithm:

- Selects overutilized host network components with usage more than a specified threshold (for instance, 0.6) and transfer the most utilized VM to the migration list
- Selects the underutilized host components with usage below a specified threshold (for instance, 0.1) and transfer their VMs to the migration list
- Selects the overloaded network links whose average bandwidth usage is more than a defined threshold (for instance, 60% of the link capacity) and transfer the channels in the link with highest data rates into the migration list.
- Migrates virtual channels which form the flows at the finishing stages of execution as the challenge with link congestion can be mitigated with over-utilized VM migration.

The algorithm then classifies the VMs in the VNR in a migration list and rank them in descending order based on their resource requirements. It then selects the VM ($VM_{m,i}$) from the top of the migration list for which the hypervisor selects a candidate host to allocate to. An intelligent overbooking algorithm is executed as a constraint to ensure the host network components utilizes enough of their resources (CPU & bandwidth) to process the VNR workload to reduce the network traffic consolidated by the network hypervisor. Equations (4.9) and (4.10) applies to network host capacity computation.

Resource Allocation Ratio (RAR) is defined as the percentage of actual resources allocated to the requested resources. It may be regarded as a reserve of overbooking ratio.

Dynamic Resource Allocation Ratio (DRAR) is applied in this algorithm as a VM admission constraints and as an actual resource allocation for migrating VM. This allows datacentre administrators to conserve more energy with use of resource overbooking and compliance to SLA by dynamically changing overbooking ratio.

$$\sum_{i=1}^{R_h} \left(rd\left(VM_{j,i}\right)\right) + DRAR_{hm} \times rd(VM_{m,i}) < C(h_i) \qquad (4.9)$$

Total virtual network requests network hosts in the datacenter

$$DRAR_{hm} = Min_{DRAR} + \frac{Max_{DRAR} - Min_{DRAR}}{Max_{DRAR}} \times \frac{1}{R_h} \sum_{j=1}^{R_h} CC(VM_{j,i}, VM_{m,i}) \quad (4.10)$$

As network constraint applies to host and link network components, Equations (4.11) and (4.12). applies to target links.

$$\sum_{i=1}^{n_i} \left(d\left(f_{j,i}\right)\right) + DRAR_i \times d(f_{m,i}) < C(l_i) \qquad (4.11)$$

Total flow executed by the links in the datacenter

$$DRAR_i = Min_{DRAR} + \frac{Max_{DRAR} - Min_{DRAR}}{Max_{DRAR}} \quad \frac{1}{n_i}\sum_{j=1}^{n_i} CC\left(d(f_{j,i}), d(f_{m,i})\right) \quad (4.12)$$

From Equations 4.10 and 4.12 it can be observed that this algorithm calculates the Dynamic Resource Allocation Ratio (DRAR) with specific regards to the correlation VMs of the various VNRs which are allocated to the host network components.

As DRAR applies to constraints which decide admission and allocation of resource during VM migration it contributes to determining the overbooking ratio. For instance, for a DRAR of 100%, host network components allocate 100% of requested resource from the network hypervisor for VMs. However, if DRAR is reduced to 60% it offers only 60% of the requested resource thus reducing the number of host network components required to process virtual network requests from the hypervisor. In this study, we make use of the correlation coefficient derived from historic data usage, mean VM utilization form preceding time frame and variables defined preliminarily to decide the parameters. We calculated the VM- to -VM correlation using Pearson Correlation Coefficient with the range between -1 and 1 (correlation varies linearly with the coefficient i.e. the lower the coefficient the lower the correlation). If the coefficient is closer to 1, it indicates the VMs are more correlated. We use equation (4.13) to normalize the range between 0 and 1 as the coefficient ranges from -1 to 1.

$$CC(X,Y) = \left(\frac{Cov\ (X,Y)}{\sqrt{Var(X)Var(Y)}} + 1\right)/2 \quad (4.13)$$

To limit the range of the Dynamic Resource Allocation Ratio, maximum and minimum DRAR (Max_{DRAR} and Min_{DRAR}) are defined. Average usage of VM of the preceding time window is the measure which is used to differentiate Min_{DRAR}. This implies Dynamic Resource Allocation Ratio (DRAR) is impacted by both correlation and actual VM usage. To determine the allocation of each of these parameters, *a* and *b* are defined in Equation (4.14). *a* defines VM usage and *b* defines guaranteed available network resource to be

allocated to the VNR. In the experiment MAX_{OR} is set to 1.0 to allow for a possible 100% resource allocation to VNR.

A method which outlines the steps involved in migrating the VMs in the VNRs between the network hypervisor and the candidate host network resources considering DRAR and the constraints is detailed below.

Algorithm.1.

VM migration with dynamic overbooking Strategy

1: **Data:** a: User-defined constant for historical utilization fraction in Min_{OR}
2: **Data:** b: User-defined constant for minimum Resource Allocation Ratio.
3: **Data:** t_{start}, t_{end}: Start and end time of the previous time window.
4: **Data:** vm_{mig}: A VNR to be migrated.
5: **Data:** h: A candidate host.
6: **function** MIGRATE (vm_{mig}, h)
7: $VNR_h \leftarrow$ all VMs in the host h;
8: $u_{mig} \leftarrow$ utilization matrix of vm_{mig} in (t_{start}, t_{end});
9: $S_{corr} \leftarrow 0$;
10: **for each** vm_i in VNR_h do
11: u_i utilization matrix of vm_i in (t_{start}, t_{end});
12: $S_{corr} \leftarrow S_{corr}$+ CC ($u_{mig}$, u_i);
13: **end for**
14: $DRAR_h \leftarrow$ calculate with Equation (4.10);
15: C_h free resource in host h;
16: rd requested resource of VNR vm_{mig};
17: $rd_{DRAR} \leftarrow DRAR_h \times rd$;
18: migrated $\leftarrow$ false;
19: if $rd_{DRAR} < C_h$ **then**
20: $DRAR_l$ calculate with Equation (4.12);
21: $C_l \leftarrow$ free resource of the link of host h;
22: d $\leftarrow$ requested resource of the flow of vm_{mig};
23: $d_{DRAR} \leftarrow DRAR_l \times d$;
24: **if** $d_{DRAR} < C_l$ **then**
25: Migrate vm_{mig} to h;
26: $C_h \leftarrow C_h - rd_{DRAR}$;
27: $C_l \leftarrow d_{DRAR}$;
28: migrated $\leftarrow$ true;
29: **end if**
30: **end if**
31: **return** migrated
32: **end function**

4.6.1. Consolidation Algorithms

A. Most Correlated:
This is the method used to migrate VNR entities to host network components which are capable of hosting VMs from the same VNRs or which are connected. This algorithm applies the most-full bin-packing procedure if the migrated VMs are not connected among themselves or if the connected VMs in the migration list are awaiting allocation else, the network virtualization hypervisor consolidates the VNRs and reallocates them when the constraints above can be met. In the event of multiple choices of allocation, the hypervisor allocates the VNRs to candidate links and host with the most-full first procedure.

B. Least Correlated:
This algorithm for network resource migration is used to allocate entities of the VNR which have the least average correlation coefficient between migrating VMs and VMs on the migration list to host network components. The algorithm calculates correlation coefficient for each destination host network components which has the capacity to host at least one entity of the VNR and classifies them in ascending order. It then selects the least correlated host and applies the DRAR constraints. The first un-occupied host from the list of hosts is selected to host a virtual network entity if no available host is found from the non-empty ones. Connected virtual network entities are allocated to separate host network components to enhance communication and reduce host network component overbooking with this algorithm.

C. Under Utilized:
This method of migration allocates VNR entities to the host network components which is utilized the least. The algorithm first prepares a list of underutilized and un-occupied host network components. The first VM in the migration list with the highest resource requirements is allocated to the first underutilized host in the list. Like the other algorithms discussed above, the method dynamically computes the DRAR based on the number of VNR entities allocated to the host network components. The capability of the host to accept migration VNR entities is assessed by calculating DRAR from correlation and used a constraint. Which implies the most utilized VNR entity in the migration list is allocated to the least utilized host.

Algorithm 2.

Most Correlated migration algorithm with dynamic overbooking Strategy
1: **Data**: *VNR*: Selected migration VM list.
2: **Data**: *H*: List of hosts.
3: sort VNR in descending order of requested CPU resources;
4: **for each** *vm* in *VNR* **do**
5: VNR_{conn} ← List of connected VNRs of *vm*;
6: H_{conn} ← List of hosts where other *VNRs* in VNR_{conn} are placed;
7: **if** H_{conn} is empty **then**
8: Migrate *vm* to the most-full host in *H* with the constraints in Algorithm 1;
9: else
10: sort H_{conn}Hconn in ascending order of free resources;
11: migrated ← false;
12: **for each** *h* in H_{conn} **do**
13: migrated MIGRATE (*vm, h);*
14: **if** migrated = true then
15: **break**
16: **end if**
17: **end for**
18: **if** migrated = false **then**
19: Migrate *vm* to the most-full host in *H* with the constraints in Algorithm 1;
20: **end if**
21: **end if**
22: **end for**

4.6.2. Baseline Algorithms

The proposed algorithm was compared with the following baseline algorithms

NoOver: **No overbooking without any migration.**

This is a non-overbooking algorithm which allocates 100% network resources to the demands of VMs and channels. This algorithm applies the Most Full First bin-packing algorithm which allocates virtual machines to the most full host network resources which have adequate resources to execute the VNRs for VM placement. The algorithm does not consider correlation or connectivity between VMs when selecting the host machines and hence VM allocation is executed regardless of these constraints. For instance, connected VMs can be randomly placed on the same or different hosts network components based on resource availability. We used this algorithm as a baseline to evaluate the computation of the degree of SLA violation and energy conservation since it doesn't apply any form of VM migration and the resource capacity available on the host network components are not exceeded.

StaticMigration: **VM to be migrated to the most correlated host without dynamic overbooking.**

This algorithm applies a static overbooking ratio for all host network components for selecting and allocating resources instead of using dynamically changing DRAR. Thus, the selected host components to all VMs being migrated. This algorithm was applied in PowerNetS (Zheng et al., 2017).

4.7. Performance Evaluation

The overbooking algorithm proposed in this work is evaluated in a simulation environment. The dynamic overbooking strategy was implemented and compared to other algorithms including non-overbooking, static overbooking and the overbooking strategy used in the production datacentre of Google. Workload response time and total energy consumed by host network components in the software defined cloud datacentre network were computed. SLA violation was thoroughly investigated based on the response time of the workload during these experiments. The response time was calculated for all the workload with a baseline algorithm without overbooking and was compared with the response time of the proposed overbooking algorithms. SLA violation is detected when the response time of a workload calculated using the proposed algorithm is longer than the that computed with the baseline one. Energy consumption is also compared with the no overbooking and other baseline algorithms.

4.8. Testbed Configuration

To evaluate our approach, we implement the algorithms in CloudSimHypervisor which we proposed in Chapter 3. CloudSimHypervisor is a simulation framework developed from (Son et al., 2015) a CloudSim (Calheiros et al., 2011) based simulation toolkit which supports features of SDN, NFV and the network virtualization hypervisor, abstraction of entire physical network resources, dynamic network configuration programmable network control logic and network components. Information regarding the usage of network resource was gathered for use in experimenting our intelligent overbooking strategy with monitoring components we added to CloudSimHypervisor. The datacentre network setup which used for the experiment was made up of 120 homogeneous host network machines in a 6-pod software defined cloud network. Each pod contains 20 hosts machines distributed

between 4 edge switches. Communication between these edge switches is enabled by the network virtualization hypervisor, Figure.4.3 illustrates. A distributive model was adapted to set the network hypervisor up in this experiment. Other resource requirements such as memory and storage are not considered in the experiments to eliminate complexities which may have a negative effect on the results.

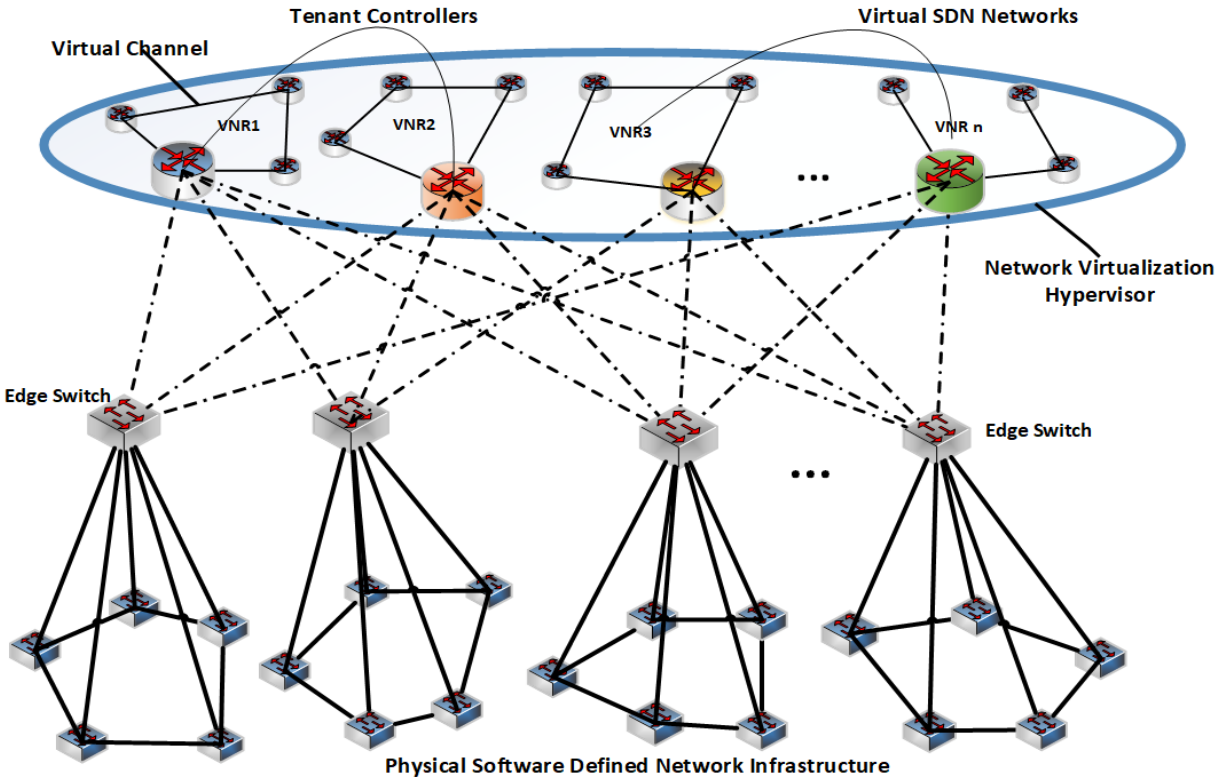


Figure .4.3. A Software Defined Cloud Networking Architecture

4.9. Workload

Network traffic in a typical datacentre vary per hourly, daily, weekly, and monthly. Characterizing this traffic in a datacentre would enable us discover patterns of changes which could be helpful in defining efficient network resource provisioning strategies. To achieve this objective, workload from the Google cluster-usage trace (Reiss, Wilkes, & Hellerstein, 2011) which is publicly available was used for the experiment. This workload consists of substantial data for more than 12,000 heterogeneous physical host machines

running 4,000 different types of applications and about 1.2 billion rows of resource-usage data. We utilized all 29 days (i.e. 696 hours) of data. The Google cluster-usage trace does not provide data for bandwidth allocated to the links connecting physical host machines in the datacentre by the network hypervisor from the VNRs. As each of these requests has a different start time and lifetime following exponential and Pareto distributions the network traffic characterization method by (Ersoz et al., 2007) was adopted to generate values for bandwidth to complete the dataset required for this experiment.

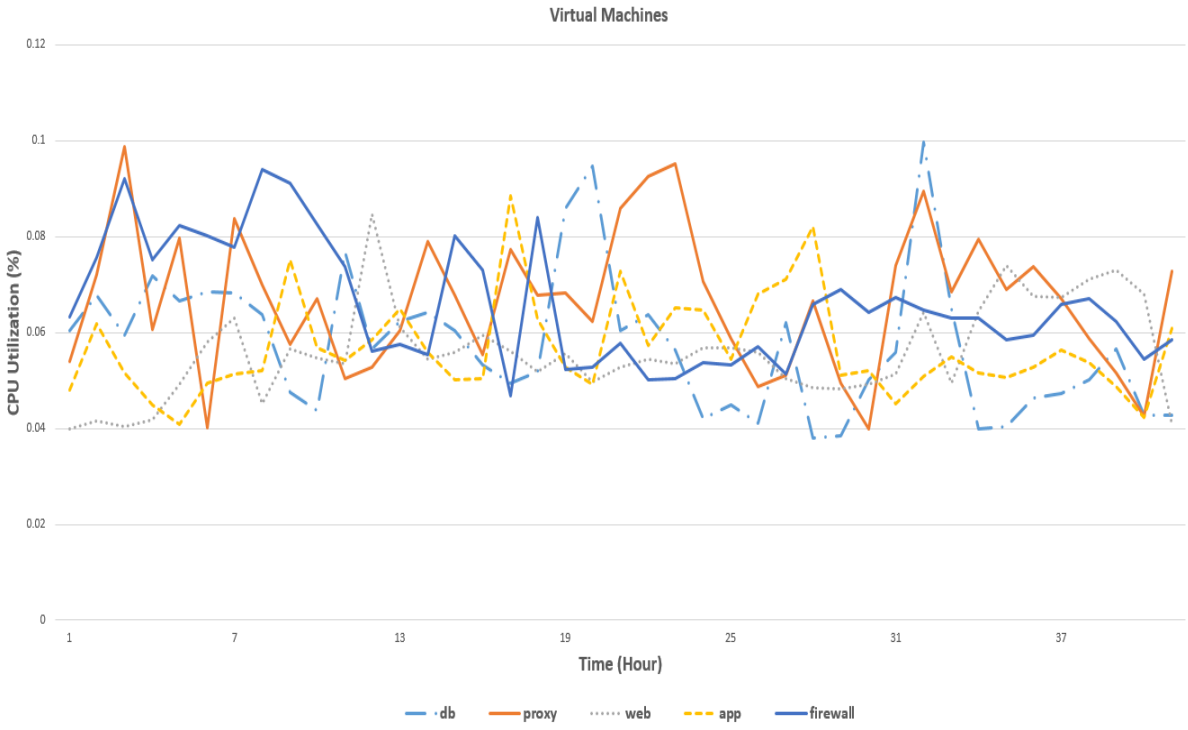


Figure 4.4. CPU request and usage distribution in a datacentre

From Figure 4.4., it is observed that workloads executed in a datacentre do not reach their peaks at the same time and they vary over time. It is assumed that each of these applications were allocated to a set of virtual machines by the network hypervisor based on the VNRs received.

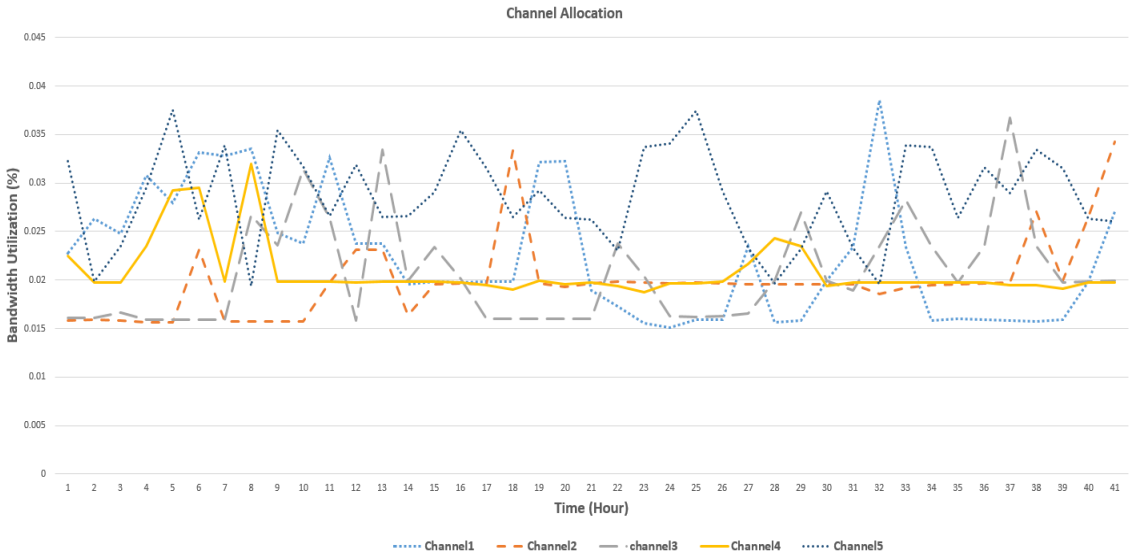


Figure 4.5. Bandwidth request and usage distribution in a datacentre

Figure 4.5 above displays a representation of usage of bandwidth of the links connecting the various network host machines in the cloud datacentre. Bandwidth demands from the channels in the VNRs are allocated to these links for processing.

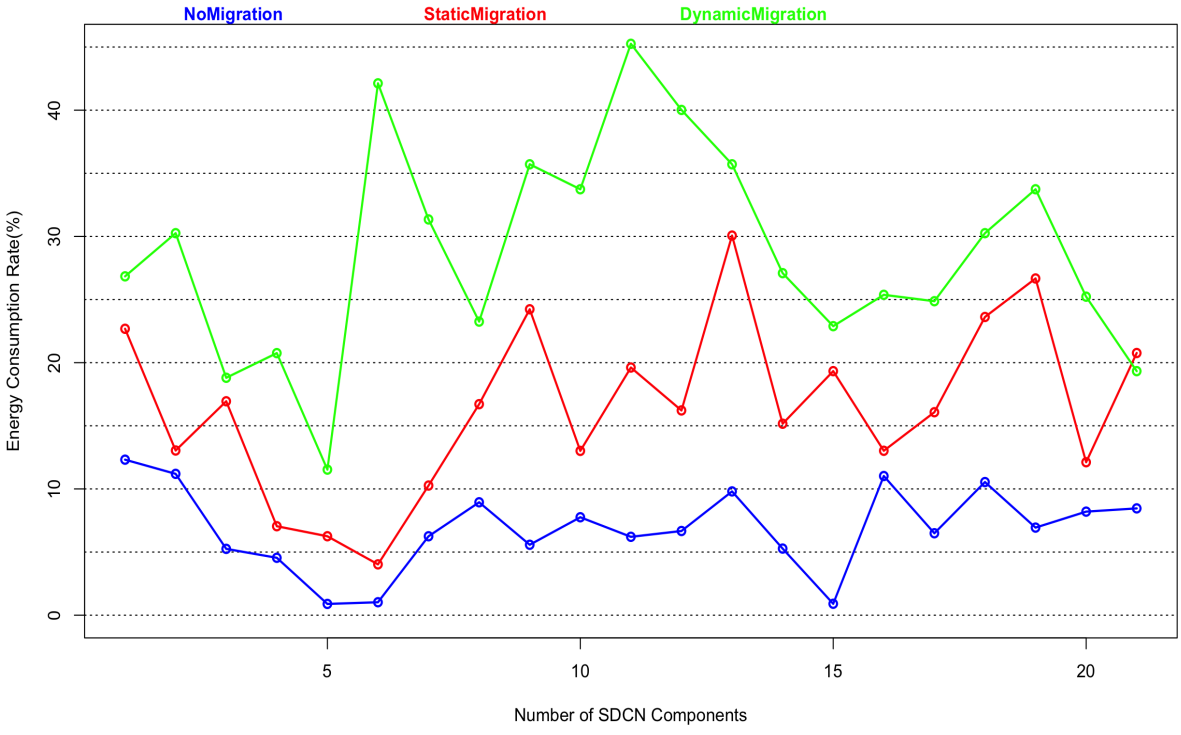
4.10. Migration Policy

Migration of network resources with well-defined policies plays a relevant role in effectively reducing overloading of host network components in datacentre networks. As overloading, which usually results from inadequate host network components is a cause of significant SLA violation, it is imperative for datacentre solution providers to adopt efficient migration policies to prevent it. In this thesis we adopt a joint consolidation and migration algorithm which utilizes dynamic overbooking ratio to reduce power consumption and SLA violation.

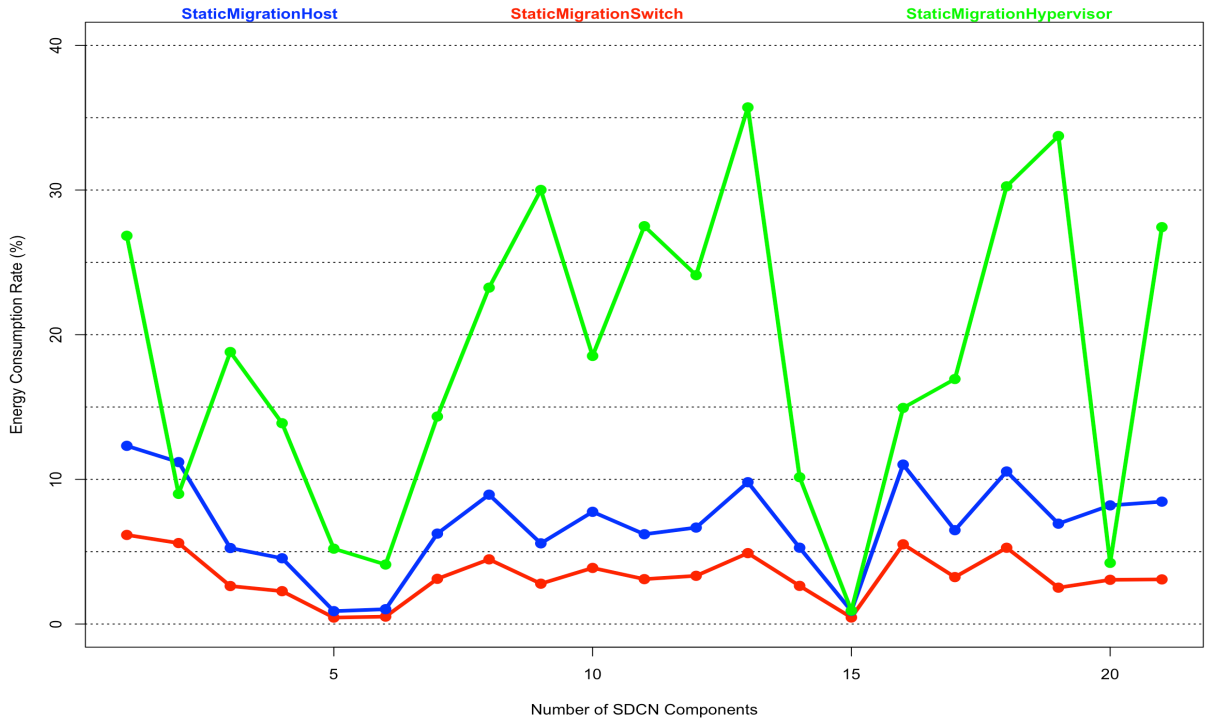
4.11. Investigating the Impact of Migration Procedures

Investigations in this section started by exploring the use of the Network Virtualization Hypervisor in implementing the migration strategies which we used as based line algorithms in the testbed experiments. The objective of these experiments is to assess the effectiveness of the algorithms in conserving energy in Software Defined Cloud Datacentre Networks with. The diagrams in Figure 4.6. displays the total energy consumption and the percentage of energy saving of different migration algorithms. In figure 4.6a, it can be observed that

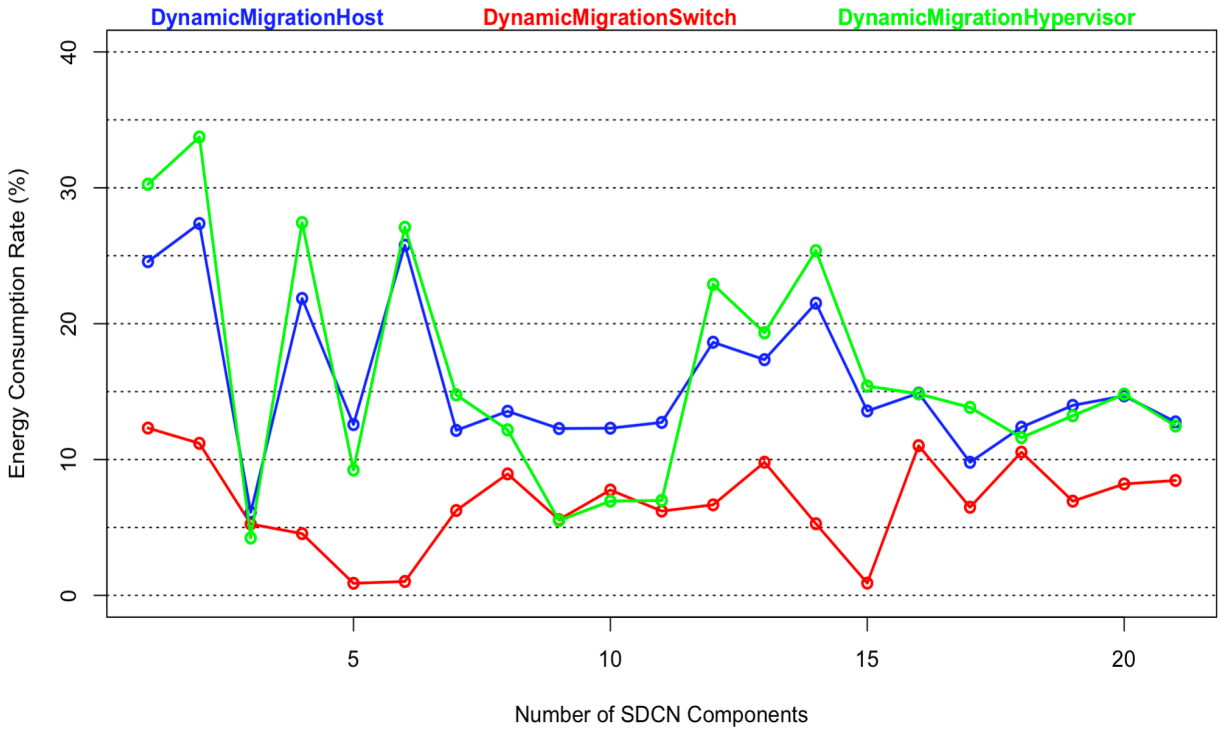
network host components utilize more energy with the implementation of static and dynamic migration algorithms compared to when there is no migration. Figure 4.6b. and 4.6c display that the energy consumption rate of specific network host components (host, switch and the network hypervisor) in the SDCDCN. The diagram shows that the network hypervisor consumes more energy than the host and the switches in the datacentre network. This is due to the fact that the network virtualization hypervisor receives all the requests for solutions and applications from network users. It is also responsible for balancing and allocating the traffic for these VNRs to the host network resources which have adequate resources to execute them. As a result, virtual software defined networks (vSDNs) which comprise VMs and channels are hosted on the hypervisor. The VMs are not placed / hosted on the physical host and switches as it is in traditional cloud datacentre networks. In Figure 4.6b. the hypervisor consumed almost 30% more energy than the amount energy consumed by the switches and hosts.



(a)



(b)



(c)

Figure 4.6. Energy Consumption for different migration strategies in SDCDCN

In Figure 4.6c, the diagram demonstrates the efficiency of the dynamic migration procedure using the same SDCDCN that was used in the experiment for the static migration strategy. The results show that the amount of energy consumed by the network virtualization hypervisor in using dynamic migration algorithm to allocate resources to the host network components in the DCN is almost 20% more than the amount of energy consumed by the host and the switch. However, the dynamic migration algorithm adopts a dynamically changing DRAR. The results above also demonstrate that with the same network setup, the network virtualization hypervisor use less energy in executing the same tasks using dynamic migration algorithm than the amount of energy it would require when using static migration algorithm.

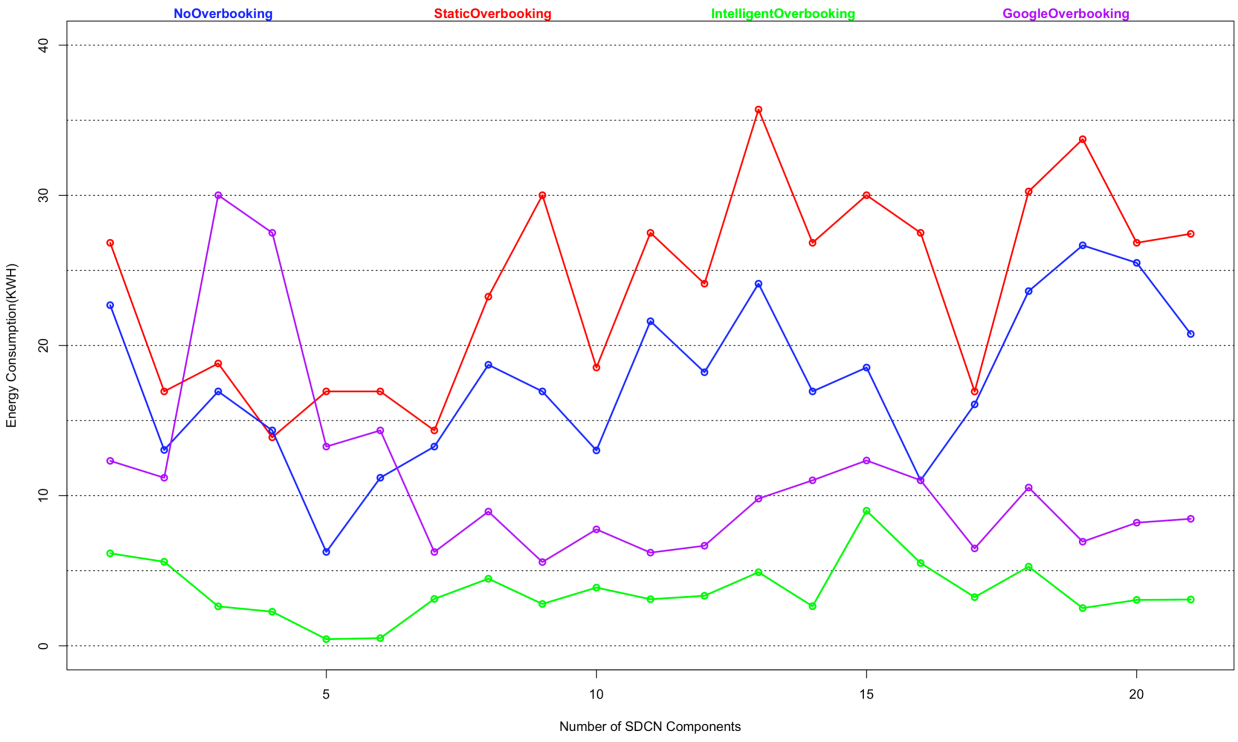


Figure 4.7. Energy consumption for different overbooking strategies in SDCDCN

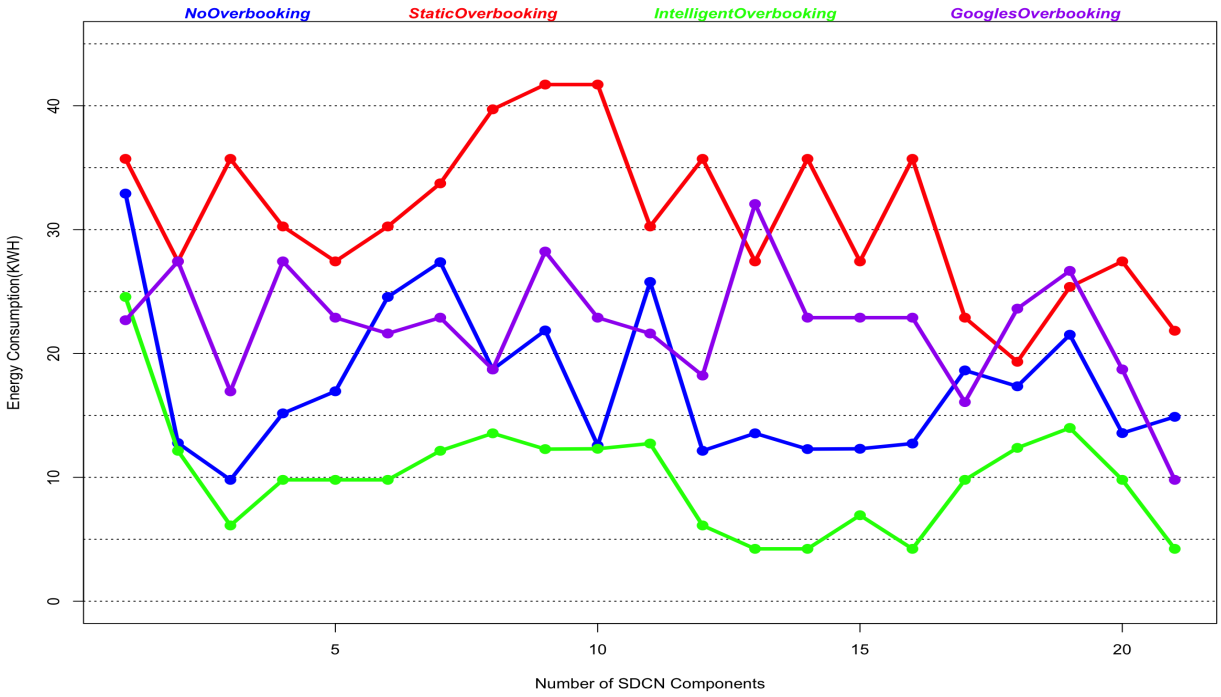
4.12. Investigating the Impact of Intelligent Overbooking Ratio.

In this section, the impact of the proposed intelligent overbooking of network host components is investigated by comparing with a correlation-aware static overbooking

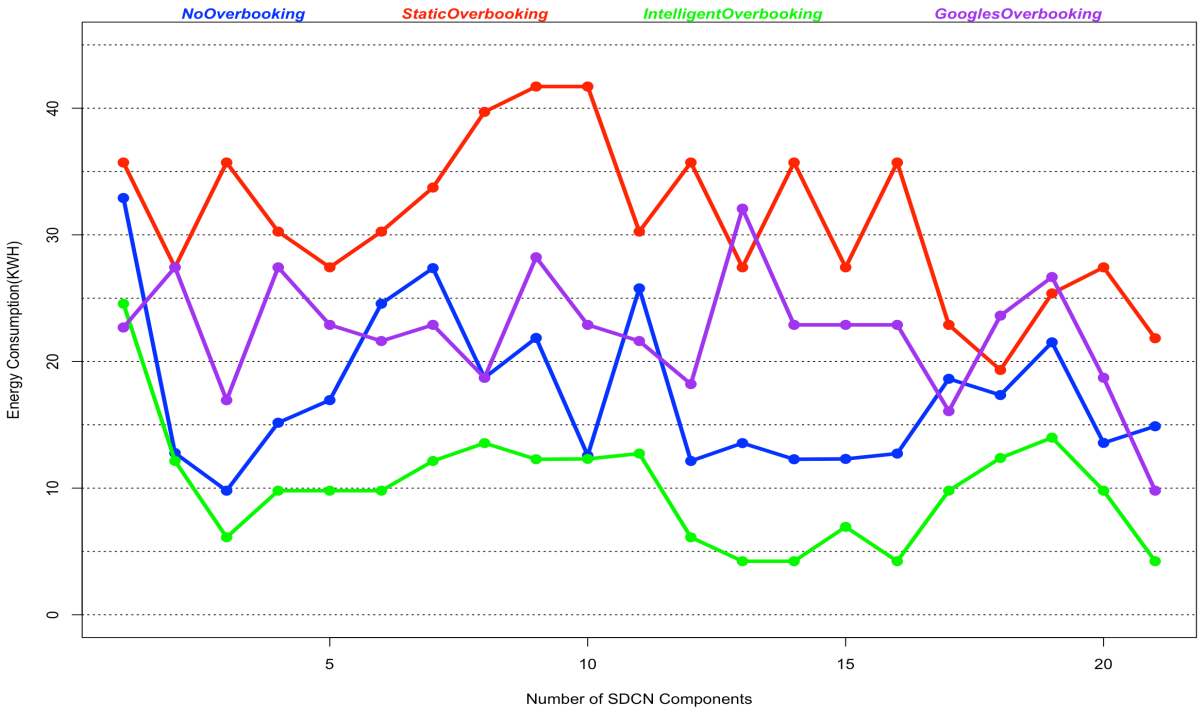
procedure and the overbooking algorithm used in Google's production datacentre. The objective of this experiment is to compare the effectiveness of our overbooking procedure

with Google's overbooking strategy and a correlation aware static overbooking. To evaluate the efficiency of our dynamic overbooking algorithm we compared it with the two overbooking procedures mentioned above because Google's production datacentre implements an online overbooking strategy like ours which has no- prior knowledge of the workload, but the correlation- aware static procedure obtain correlation of workload beforehand. This algorithm then places VMs on network hosts which are close each other and then migrates them to other close by hosts when the initial hosts are overloaded.

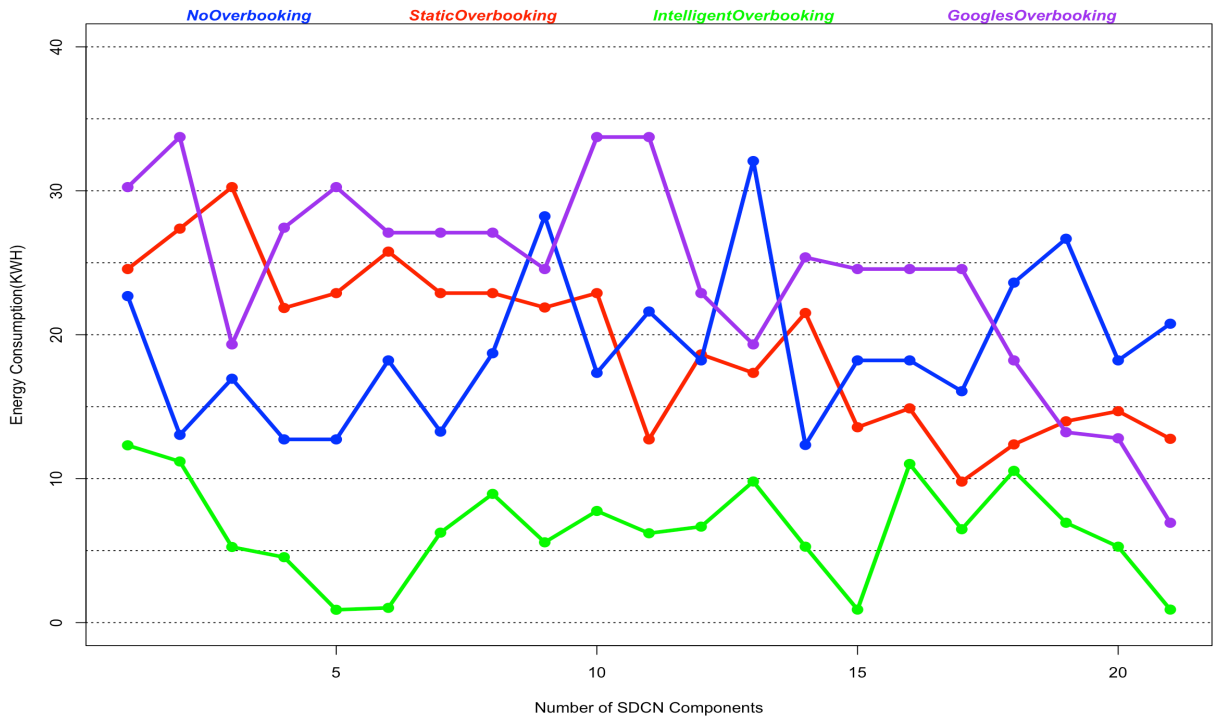
The differences between static overbooking, Google's overbooking and our overbooking strategies is presented in Figure 4.7. The figure shows that the maximum energy consumed with the use of static overbooking is about 23.3KWh, that consumed with Google's overbooking algorithm is 42.5KWh and 33.7KWh with dynamic overbooking algorithm. This implies that Google's overbooking strategy consumed the highest amount of energy. The maximum amount of energy consumed by our dynamic overbooking algorithm 10.4KWh of energy more than that of static overbooking. This is because the overbooking ratio used with the static overbooking algorithm doesn't change when migrating network resources in between hosts network components irrespective of the workload in the cloud datacentre network whereas the overbooking ratio of the our dynamic overbooking strategy changes with regards to changes in the workloads received by the network virtualization hypervisor.


NoOverbooking
StaticOverbooking
IntelligentOverbooking
GooglesOverbooking
Energy Consumption(KWH)
0
10
20
30
40
5
10
15
20
Number of SDCN Components

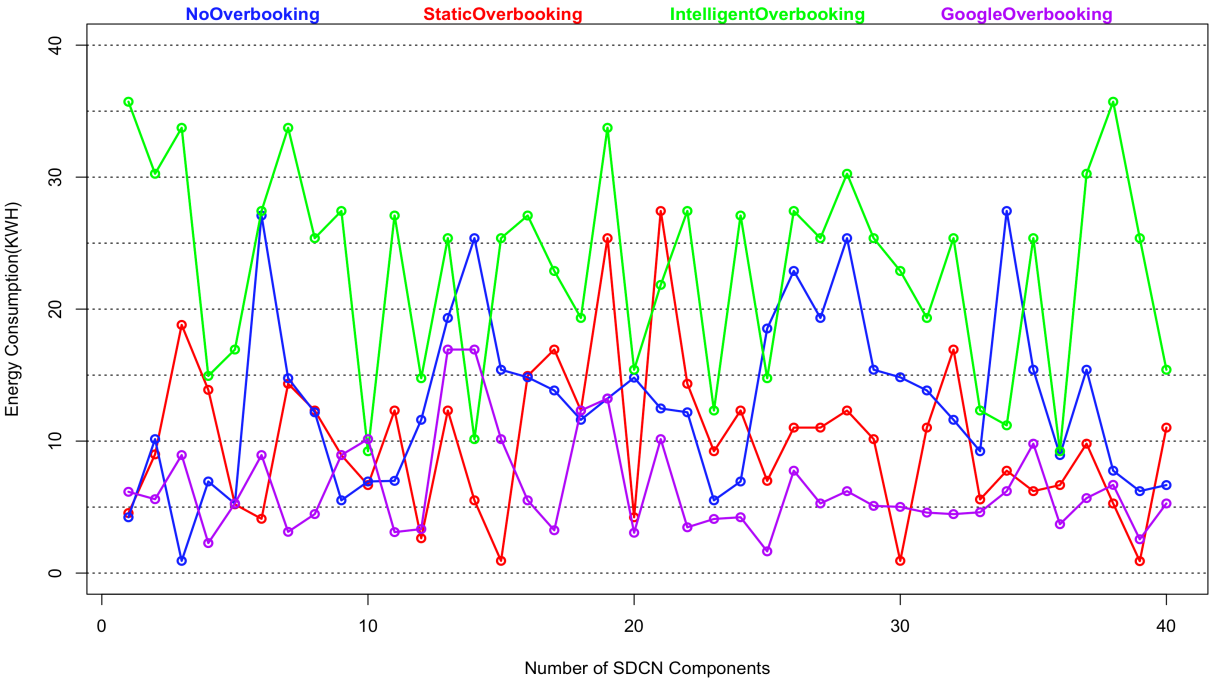
(a)


NoOverbooking
StaticOverbooking
IntelligentOverbooking
GooglesOverbooking
Energy Consumption(KWH)
0
10
20
30
40
5
10
15
20
Number of SDCN Components

(b)



(c)



(d)

Figure 4.8. Energy consumption analysis in the entire datacentre network

To explore the efficiency of our intelligent overbooking strategy with regards to energy conservation we analyse the dynamics of the power distribution and consumption in the

entire software defined cloud datacentre. Figure 4.8a, shows energy consumption by network links in the datacentre. Figure 4.8b, displays energy consumed by the network virtualization hypervisors is managing network traffic and provisioning network resources within the datacentre and Figure 4.8c depicts energy consumption by the hosts and switches network components in the datacentre. In all the three diagrams above, the energy consumption of the network and host components compared with the baseline (No Overbooking) shows that static overbooking and Google's overbooking algorithms consumes a lower amount of energy. Correlation-aware strategy employed by static overbooking has a low energy consumption rate at start of deployment but accumulates and increases as the datacentre get highly loaded with network requests and migration activities utilizing more switches and network host over time. The results also show that our intelligent overbooking algorithms constantly consumed less energy although the RAR continually changes following changes in actual workload received by the network virtualization hypervisor. The network virtualization hypervisor consolidates all VNRs and intelligently allocates them to the host network components with regards to resources availability, SLA and priority. Consolidation of the VNRs conserves huge volumes of datacentre energy. Hence, although more workload is executed using our intelligent overbooking algorithm, it has an energy conservation rate of close to 40% more than 20% more than that of Google's overbooking strategy from Figure 4.8d.

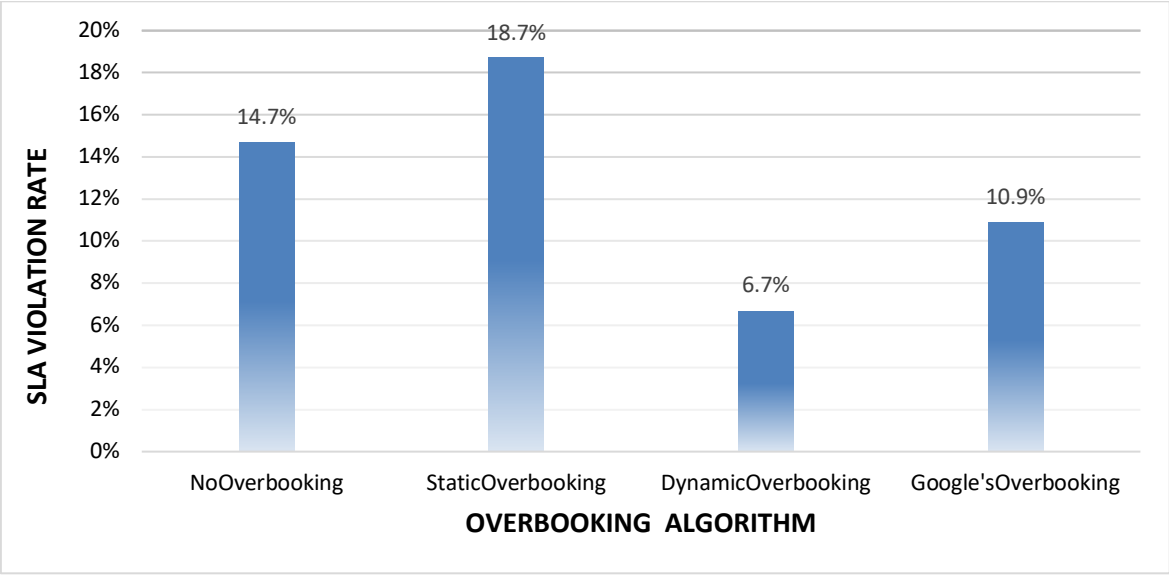


Figure 4.9. SLA Violation rates for different overbooking strategies

The effectiveness of the dynamic overbooking can be clearly seen in SLA violation percentage presented in Figure 4.9. SLA violation rate of the static overbooking algorithm (18.7%) is far higher than our dynamic algorithm (6.7%). The other two algorithms used in the experiment violation rates are (14.7%) for no overbooking and (10.9%) for the overbooking strategy in Google's production environment.

4.13 Summary

In this chapter, an intelligent overbooking strategy which dynamically allocates network and host resources based on the change in workloads and network resource utilization was presented. The variation of network workloads had a direct influence on the overbooking ratios in real-time. Leveraging the intelligent overbooking algorithm allows the network virtualization hypervisor to dynamically manage the allocation of network resources to and from host network components in order to maximize energy cost by drastically minimizing network resource overload and execution time of VNRs as well as limit SLA violations. Our implementation and testbed experiments using CloudSimHypervisor demonstrated that our intelligent overbooking strategy is efficient in conserving energy in software defined cloud networks (SDCN) and minimizes the rate of SLA violations when compared with the baseline.

Chapter 5 Performance-Aware Intelligent Overbooking Strategy to Enhance Resource Usage in Cloud Datacentre Networks.

5.1 Introduction

In chapter 4, an energy efficient intelligent overbooking strategy which dynamically allocates network and host resources based on the changes in workloads and network resource utilization was presented. In this chapter a performance – aware intelligent overbooking algorithm which optimise resource usage with dynamically changing overbooking ratios is presented.

Over-provisioning of network resources (hosts, links and switches) is major cause of a number of challenges such energy inefficiency and SLA violations in software defined cloud datacentres. As they are provisioned for peak demand, these network resources are under-utilized for a greater part of the time. For instance, research shows that the average utilization of servers in most large cloud datacentres is reported to be between 10-30% (Son, 2018), which implies that user application requirements utilize only a fraction of the allocated resources. Cloud service providers usually adopt resource overbooking to maximize the usage of network resources (F Caglar & Gokhale, 2014). This is achieved by committing more resources such as bandwidth, CPU, memory and VM on a lesser number of physical host than actually available and hence reducing the number of physical network resources that are put to use.

The emergence of software defined networks (SDN), and network virtualization technologies enables CPS to mitigate the challenge of over-provisioning of resources both on their part and on the part of their customers. With Software Defined Cloud Networks which is an integrated system of mainly SDN and NFV from which the network virtualization hypervisor is developed, cloud datacentres now, are capable of managing their

network stack through software and consider networks as one of the key elements in their consolidation techniques. The network virtualization hypervisor enables multiple heterogeneous networks to share the same physical network infrastructure providing a platform to support isolating these independent virtual networks and their custom defined tenant controllers. As a result, routing and other control-related issues are managed via the hypervisor, enabling the data forward plane/ physical network to quickly react and adapt to changes in network resource demand and application requirements (Son, 2018). The hypervisor makes it possible to control individual traffic flow between VMs and physical hosts and as such network traffic can be consolidated to a smaller number of physical network resources, i.e., links, switches and other host machines with the use of resource overbooking.

Overbooking network resources to process network requests and applications in real time for cloud datacentres is likely to cause unreliable performance due to significant rise in jitter. This can cause CPSs not to able to provide the performance requirements such availability of resources and response time which they assured their service users and hence violating the Service Level Agreement, SLA. Consequently, there is a need to having a systematic approach for overbooking the resources of the networks in cloud datacentres (F Caglar & Gokhale, 2014).

In reality, when overbooking ratios are low, network service users benefit from high service satisfaction at the expense of ineffective and uneconomical usage of resources of CSPs whiles high overbooking ratios result in effective utilisation of resources and efficient energy savings by CSPs but poor QoS delivery to their clients to relatively high contention and interference from resource overbooking (Alford, 1988), (Sufyan Beg & Ahmad, 2002), (Pu et al., 2010), (Zhang, Tune, & Hagmann, 2013).Cloud datacentres may comprise of large-scale heterogeneous computer network components and so using single overbooking ratio might not be efficient to achieve the best results (Faruk Caglar & Gokhale, 2014).

While overbooking strategies helps in maximizing network resource usage, they also increase the tendency of SLA violation when either host or network is overloaded. If the consolidated VMs or traffics reach the peak utilization at the same time, insufficient amount of resources would be allocated which will delay the workload processing. With

dynamically changing workload in mission critical cloud datacentres, we proposed a strategy which allocates more precise amount of network resources to VMs and hosts in software defined cloud datacentre networks. This strategy can increase overbooking in the host machines and networks while still providing enough resources to minimize SLA violations (Caglar & Gokhale, 2014).

Finally, with the dynamic nature of networks and cloud unification environment, and the need for real-time response to network requests and operations, offline estimation of overbooking ratios would not be suitable for software defined cloud networks (Caglar & Gokhale, 2014). Developing systematic overbooking ratios capable of making accurate trade-offs in overcoming the conflicting objectives of the datacentre network systems becomes a key requirement.

In the current state of the art, cloud service providers determine overbooking strategies for their datacentres by analysing the workloads of VMs through network resource assessment applications, and make decisions based on historical studies on optimal network resource overbooking ratios (Lowe, 2013) and fixed percentiles (Son, Dastjerdi, Calheiros, & Buyya, 2017). Others also employ dynamic overbooking strategy but for homogenous datacentre setups (Son et al., 2017) and consider initial VM placement as a major constraint and for only host computers (Caglar & Gokhale, 2014). These procedures for computing and implementing overbooking ratios, however, are not suitable for software defined cloud datacentre networks due to the heterogeneity of the workload received from network service users with different SLAs. In this chapter, a performance-aware intelligent overbooking algorithm for each host and network component in the SDCN which;

1. autonomously ensures QoS in executing real-time virtual network requests without using fixed percentiles.
2. autonomously, predicts network resource usage and performance for the next time window
3. enhances the volume of network resource usage in SDCN and reduces SLA violations is presented

Our overbooking strategy has three novelties. Firstly, our overbooking strategy dynamically adapts to the changes of workload instead of using a fixed percentile. Secondly, it is designed to work without the prior knowledge of the workload. Finally, we consider the performance of Host and network resources and how they can be used to satisfy QoS in executing real-time network user requests considering their SLAs.

5.2. Related Work

This section outlines a number of research studies which have thoroughly explored energy efficient cloud resource management. Our focus in this chapter, would be on studies relating to network resource overbooking and application of forecasting prediction algorithms for enhancing network resource utilization in software defined cloud datacentre networks.

Forecasting future usage of network resources with regards to historic information is a relevant aspect of network resource overbooking. In this thesis, we adapt machine learning based decision-making strategies predict future usage of network resources in a software defined cloud networks, it has been applied in several research studies. For instance (Edwards, New, & Parker, 2012) in their research predicted future consumption electricity in the energy industry. (Olsson, 2013) also applied machine learning to predict water production of a heat pump after 24 hours. (Imam, Miskhat, Rahman, & Amin, 2011) used machine learning based resource decision making approach to predict future allocation of network resources in cloud computing environments to save cost.

Moreno & Xu, (2012) proposed an artificial neural network-based overbooking algorithm which enhances performance requirements of online applications and optimises energy efficiency of datacentre networks. They adapted an approach which predicts the resource utilization pattern of customers based on historic information. The approach also used a cost-benefit and analysis estimate the volume of resources to be allocated to virtual machines. Although this study provided an overbooking algorithm for resource allocation, it didn't make provision for per-host overbooking ratios.

Tomás & Tordsson, (2013) suggested a management framework in cloud computing environment which consist of scheduling algorithms for vertical elasticity such (CPU,

memory and bandwidth) and an admission control for horizontal elasticity. In their study, they ignored interference effect and network resource contention and assumed that there are no SLA violations if only the capacity of used resources is within the limit of the capacity of the physical host machines. In this thesis, the network virtualization hypervisor takes into account parameters such as number of network user requests, physical host network resources and their capacities such CPU, bandwidth and memory usage to accurately study and predict the performance of the network before and after overbooking the resources. A mechanism which eliminates the challenge of network and performance interference. Our overbooking algorithms also provide asymmetric overbooking ratios within well-defined time limits.

Roy, Dubey, & Gokhale, (2011) proposed an autonomous framework for auto scaling cloud network resources based on an algorithm for predicting forecasting workloads. Although the objectives of this study performance assurance enhancement, it focused on how to proactively scale the volume of resources required to serve the need of specific applications in a datacentre with regards to predicted incoming workloads. The research team used usage traces of the 1998 Soccer World Cup.

Zhang et al., (2013) developed CPI2 for performance optimization of latency sensitive jobs in the event of performance interference. The CPI2 solution identifies CPU performance interference incidents and secures the affected jobs by throttling the triggering task. The research team showed that CPI (Cycle-per-instruction) is an effective parameter for representing application response time. We applied the multiplicative inverse of CPI (Instruction per cycle or IPC) as a key parameter in developing our algorithms and evaluating the performance of network of network resources software defined cloud networks. We also used trace of data from the production environment of Google datacentres (Reiss et al., 2011) in our work. We leveraged discoveries from (Reiss, Tumanov, & Ganger, 2012) (Liu & Cho, 2012) (Mishra, Hellerstein, Cirne, & Das, 2010) in providing statistical profile and resource utilization, workload characteristics and task classification in this chapter.

5.3. Training of Artificial Neural Networks

Neural networks are new artificial intelligence techniques. In most cases they are adaptive systems that changes its structure based on external or internal information that flows through the network during the learning phase. ANN learning methods attempt to find a set of connections **w** that gives a relation which best fits the training set. For instance, neural networks can be considered as highly nonlinear functions with the basic the form:

$$F(x, w) = y \qquad (5.1)$$

where **x** is the input vector presented to the network, **w** represents the weights of the network, and **y** is the corresponding output vector approximated or predicted by the network. The weight vector **w** is usually ordered first by layer, then by neurons, and finally by the weights of each neuron plus its bias.

A neural network has to be configured such that the application of a set of input produces either direct or through a relaxation process the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to explicitly set the weights using a prior knowledge. An alternative is to train the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule; although training the ANN with data is not traceable, it is considered as an efficient optimisation technique where the best network parameters (weights and biases) are determined in order to minimize network error. As a result, several function optimization methods from numerical linear algebra can be applied to network learning, one of which is the **Levenberg-Marquardt back propagation algorithm** which we utilize to train our ANN.

In training ANN, the internal neural networks, all 695 hours of the cluster trace was used. We exploited all 29 days (i.e. 695 hours) of data from Google cluster-usage trace except the 696th hour using our intelligent overbooking strategy. The resource usage and performance predictions were made for the 696th hour.

Google Inc. made available datacentre cluster data for a 29 days period on May 2011 a document Google cluster-usage traces: format+schema (Reiss et al., 2011). This workload consists of significant data for over 12,000 heterogeneous physical host machines executing

4,000 different requests or applications and close to 1.2 billion rows of resource-usage data. Noisy data in this training set was removed in order to avoid overfitting in the artificial neural networks (ANN).

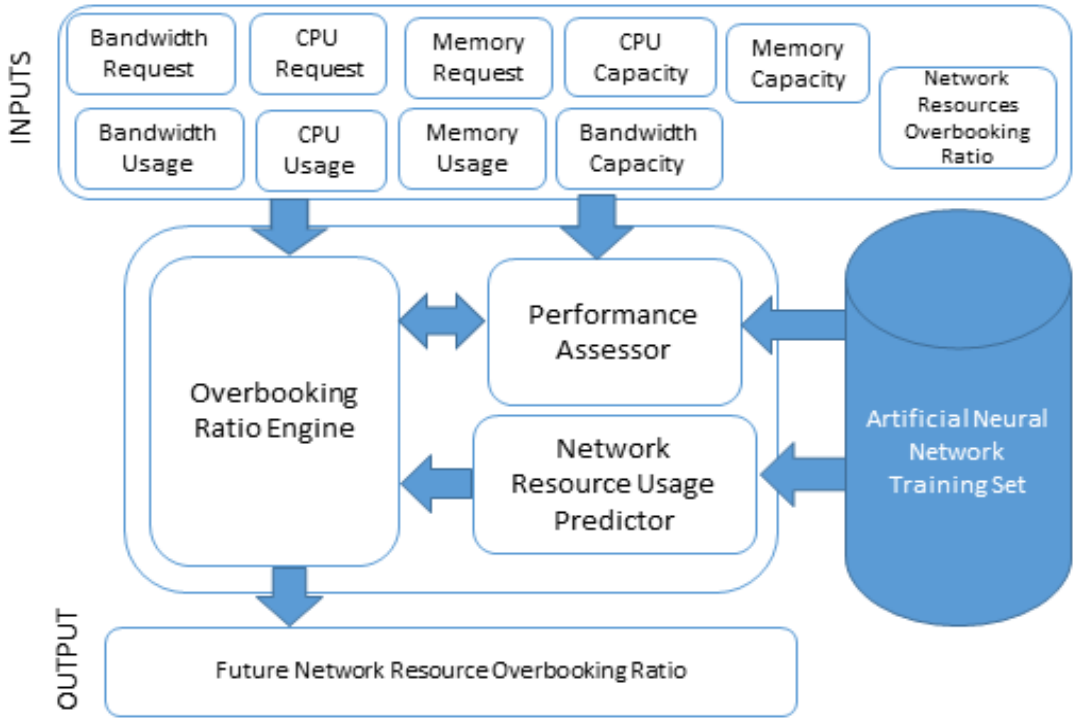


Figure 5.1. Intelligent Autonomous Overbooking System Architecture

5.4. Intelligent Autonomous Overbooking System

The architecture above aims to maximise performance of host network resources of software defined cloud datacentre networks (SDCN) whiles minimising the user SLA. This proposed architecture, which is illustrated in Figure 5.1., uses a machine learning-based online decision-making strategy to determine precise and effective overbooking ratios for host network resources in the datacentre. Our focus in this thesis is to estimate network bandwidth and CPU overbooking ratios for specific time intervals for the physical networks in SDCDN. The control unit of our over propose network resource overbooking architecture comprise (a) an overbooking ratio prediction engine (b) a resource usage predictor and (c) a performance assessor. The performance assessor and resource usage predictor components of the control unit, retrieve historic data from a training set repository to train internal ANN. Our overbooking algorithm use the following as input parameters; bandwidth, CPU and

memory overbooking ratios, average bandwidth and CPU request, average bandwidth and CPU usage, average performance and average bandwidth and CPU capacity. The overbooking algorithm uses these parameters to predict dynamic overbooking ratios for the host network resources which changes with changes in network traffic every hour. These input parameters are obtained from collecting details from network resources and the overbooking ratio inputs are estimated using equation (5.3) for all network resources in the heterogeneous cluster.

A. Resource Usage Predictor: The role of the resource usage predictor of the proposed overbooking architecture is to predict the average usage of network resources (Bandwidth and CPU) of the host network of the SDCDN for the specified time interval. A two-layer, feed forward ANN is used in making the prediction. ANNs have the capability to model and generalize both linear and non-linear relationships between input and output. The hidden layer is the layer which determines the prediction (Krose & Smagt, 1996). The sliding window average for network resource usage data, and network resource requests as well as with the host network resource capacity are features that are delivered to the resource usage predictor.

$$TotalResourceAllocated = \sum_{i=0}^{n} ResourceAllocated_i \quad (5.2)$$

$$Overbooking\ Ratio = {}^{TotalResourceAllocated}/_{Capacity} \quad (5.3)$$

$Total\ Resource\ Allocated$: $Total\ amount\ of\ resources\ allocated\ to\ all\ the\ task\ on\ host\ network\ components$

n : $Total\ number\ of\ task$

$Resource\ Allocated$: $Size\ of\ allocated\ CPU\ or\ bandwidth$

$Capacity$: $Resource\ capacity\ of\ host\ network\ resources$

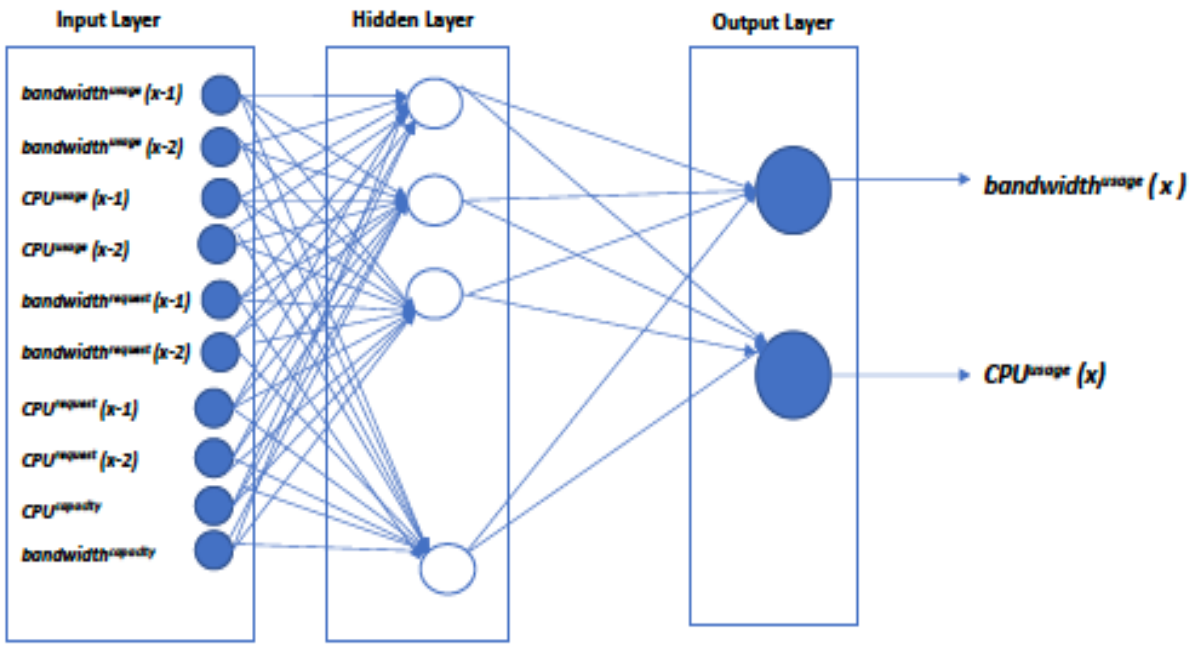


Figure.5.2 Topology of the Resource Usage Prediction Artificial Neural Network

Input layer: bandwidth^{usage} (x-1), bandwidth^{usage} (x-2), CPU^{usage} (x-1), CPU^{usage} (x-2), bandwidth^{request} (x-1), bandwidth^{request} (x-2), CPU^{request} (x-1), CPU^{request} (x-2), CPU^{capacity}, bandwidth^{capacity}

Hidden Layer: 91 neurons
Activation Function (in hidden layer)
: Tangent Sigmoid
Output Layer: bandwidth^{usage} (x); CPU^{usage} (x)
Transfer Function (in output layer)
: Pure Linear

where

x = The predicted hour

bandwidth^{usage} (x-1) = Average bandwidth usage at time (x-1)

bandwidth^{usage} (x-2) = Average bandwidth usage at time (x-2)

CPU^{usage} (x-1) = Average CPU usage at time (x-1)

CPU^{usage} *(x-2) = Average CPU usage at time (x-2)*

$bandwidth^{request}$ *(x-1) = Average bandwidth request at time (x-1)*

$bandwidth^{request}$ *(x-2) = Average bandwidth request at time (x-2)*

$CPU^{request}$ *(x-1) = Average CPU request at time (x-1)*

$CPU^{request}$ *(x-2) = Average CPU request at time (x-2)*

$bandwidth^{capacity}$ *= Bandwidth capacity of host network component*

$CPU^{capacity}$ *= CPU capacity of host network component*

$bandwidth^{usage}$ *(x) = Average bandwidth usage at time x*

CPU^{usage} *(x) = Average CPU usage at time x*

These input parameters for network resources were considered for predicting resource usage because they are common factors that influence the performance of host networks in datacentres. The capacities of these network resources which depicts the heterogeneous nature of software defined cloud datacentre networks are fed into the ANN to aid in conveying accurate correlation between input and output information.

B. Overbooking Ratio Prediction Engine: The purpose of the overbooking ratio prediction engine is to estimate the per network resource overbooking ratios after the resource usage predictor has predicted the magnitudes of network resource usage for the next specified time window. This information is then transferred to the performance assessor. The performance assessor the predicts the performance of the network resources using the new overbooking ratios and transfer the information back to the overbooking ratio prediction engine. This communication exchange between the overbooking ratio prediction engine and performance assessor iterates until the predetermined convergence values (computed manually from the historic data in the Google trace) are satisfied. This iteration uses Security Slack which is a distinct step size to concur on an exquisite overbooking ratio.

$$SecuritySlack(x) = Capacity(x)\ X\ SecurityPercentage \quad (5.4)$$

$$OverbookingRatio(x) = \beta\ X\ Capacity(x) - SecuritySlack(x) / PredictedUsage(x) \quad (5.5)$$

$$NetworkResourceRequest(x) = OverbookingRatio(x)\ X\ Capacity \quad (5.6)$$

where

x = the predicted time
β = Elastic capacity to present the best overbooking ratio if the predicted performance is too high
Capacity(x) = Capacity of Network Resource component at the time x
SecurityPercentage(x) = Elastic capacity on network resource to converge the best ratio at hour x
OverbookingRatio(x) = Network Resource overbooking ratios at the time x

C. Performance Assessor: The function of the performance assessor component of our overbooking architecture is to predict the performance of the network resources with the new overbooking ratios which are calculated by the overbooking ratio engine. Accordingly, it lays out the assurance that the new overbooking ratios received from the overbooking ratio engine does not breach SLAs. The performance assessor makes use of the instruction per cycle (IPC) as the measure of performance which implies that the higher the value, the better the performance. Violation of user SLA is determined based on historic optimal performance values on the basis of per-network resource. These threshold values are attained either from the trace log or by assignment by a domain expert. If the predicted IPC of the host network component under consideration is greater than the maximum IPC in the trace in for the component in the cluster of components our overbooking algorithm presumes there will not be SLA violation and hence deliver performance assurance for network traffic and network user request. Contrarily, if the predicted IPC breaches SLA, the overbooking algorithm exempts those network components from being overbooked.

The topology of this ANN for predicting IPC is provided in the mathematical formulation below. We assume that the amount of resources allocated by the network virtualization hypervisor and the mean number of virtual network requests it assigns to the host SDCDCN components changes when the overbooking ratio engine derives new ratios.

Hidden Layer: 96 neurons

Activation Function (in hidden layer)

: Tangent Sigmoid

Output Layer: P(x)

Transfer Function (in output layer)

: Pure Linear

Input Layer: $bandwidth^{request}(x)$, $CPU^{request}(x)$, $bandwidth^{OR}(x)$, $CPU^{OR}(x)$,

$bandwidth^{capacity}$, $CPU^{capacity,}$ $VNR(x)$

x= *The predicted time*

$bandwidth^{request}(x)$ = *Average bandwidth request at time x*

$CPU^{request}(x)$ = *Average CPU request at time x*

$bandwidth^{capacity}$ = *Bandwidth capacity of host network component*

$CPU^{capacity}$ = *CPU capacity of host network component*

$VNR(x)$ = *Average Virtual Network Requests at time x*

$bandwidth^{OR}(x)$ = *bandwidth overbooking ratio at time x*

$CPU^{OR}(x)$ = *CPU overbooking ratio at time x*

$Performance(x)$ = *average performance*

5.5. Performance Evaluation

The proposed intelligent autonomous overbooking strategy was implemented and evaluated in simulations and testbed experiments. The proposed method to measure the QoS levels of heterogeneous network traffic workload and the performance of host network resources in cloud datacentre networks was implemented. From the test experiments conducted and simulations, we evaluated actual proportions of host and network resource usage in SDCDCN and also predicted resource usage volumes for the next hour. SLA violation is assessed through per- host network resource performance of network components in the datacentre.

The performance of each host network components was evaluated, and the results compared with that of the proposed algorithm. To train the internal neural networks of our intelligent

overbooking strategy, we made use of all 29days (i.e. 695 hours) of data from Google cluster-usage trace except the 696th hour for which the prediction of network resource usage was made. The activation function was selected in the hidden layer and output layer with respect to the back propagation, expected output value constraints and on a trial-and- error performance results of the ANN.

5.6. Testbed Configuration

In order to evaluate the proposed intelligent overbooking strategy, a testbed experiment was implemented in CloudSimHypervisor which was proposed in Chapter 3. CloudSimHypervisor is a CloudSim based simulation tool (Calheiros et al., 2011) which supports divers SDN, NFV, and software defined cloud networking features such as dynamic network configuration and programmable network virtualization hypervisor. Monitoring components were added to the simulator to gather usage information of cloud datacentre network components, network traffic for use with our proposed intelligent autonomous overbooking strategy. The cloud datacentre simulated for our experiment consists of 125 heterogeneous host network machines in a 5-pod software defined cloud network. Each pod contains 25 hosts machines distributed between 4 edge switches. Communication between these edge switches is enabled by the network virtualization hypervisor as illustrated in Figure.5.3. A distributive model was adapted to set the network hypervisor up in this experiment. Other resource requirements such as memory and storage are not considered in the experiments to eliminate complexities which may have a negative impact on the results.

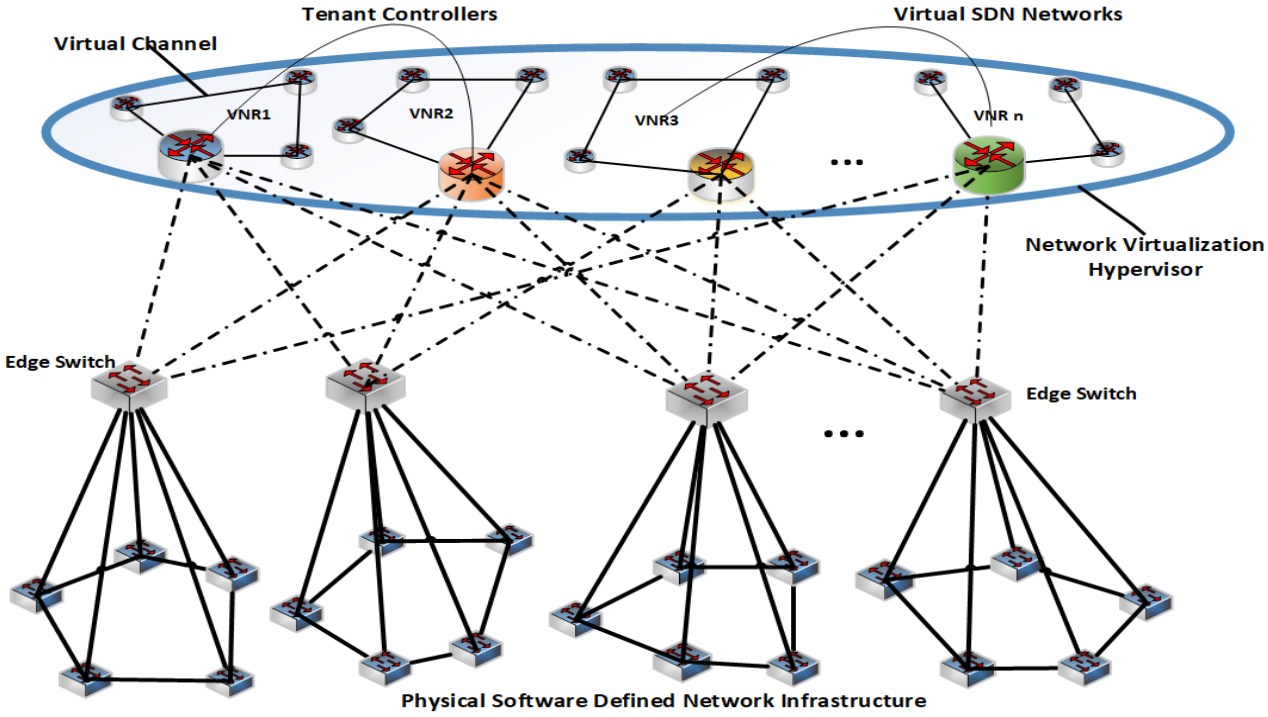


Figure .5.3. A Software Defined Cloud Networking Architecture

For this experiment, the network components with the highest average bandwidth and CPU usage within the 29days of cluster trace under consideration were used (i.e. sufficient number of host machines were used in testbed experiments). This filtering was necessary in order to set out the resource-intensive usage network components in the SDCDCN for the experiment.

5.7. Workload

In a datacentre in the real-world traffic varies hourly, daily, weekly, and monthly. Characterization of traffic of a datacentre enables us to discover patterns of change that can be utilized to predict efficiently the future resource usage in the SDCDCN. To realize the objectives for this activity, Google cluster-usage trace (Reiss et al., 2011) which is publicly available was used for the testbed experiment. This workload consists of substantial data for more than 12,000 heterogeneous physical host machines running 4,000 different types of applications and about 1.2 billion rows of resource-usage data. We utilized all 29 days (i.e. 695 hours) of data.

We utilized all 29 days (i.e. 695 hours) of data. The Google cluster-usage trace does not provide data for bandwidth allocated to the links connecting physical host machines in the datacentre by the network hypervisor from the VNRs. As each of these requests has a different start time and lifetime following exponential and Pareto distributions we adopt the network traffic characterization method by (Ersoz et al., 2007) to generate values for bandwidth to complete the dataset required for this experiment.

5.8. Investigating the Impact of Autonomous Overbooking on Datacentre Resource Utilisation

The impact of the proposed intelligent autonomous overbooking strategy in software defined cloud datacentre network (SDCDCN) is would now be investigated. The objective of this experiment is to verify the efficiency of our proposed autonomous overbooking system in predicting resource usage values using the resource usage predictor ANN in comparison with actual resource usage values from the production environment of Google datacentre. The effectiveness of the performance predictor ANN which uses the overbooking ratios provided by the overbooking ratio engine is also assessed with data from the Google traces.

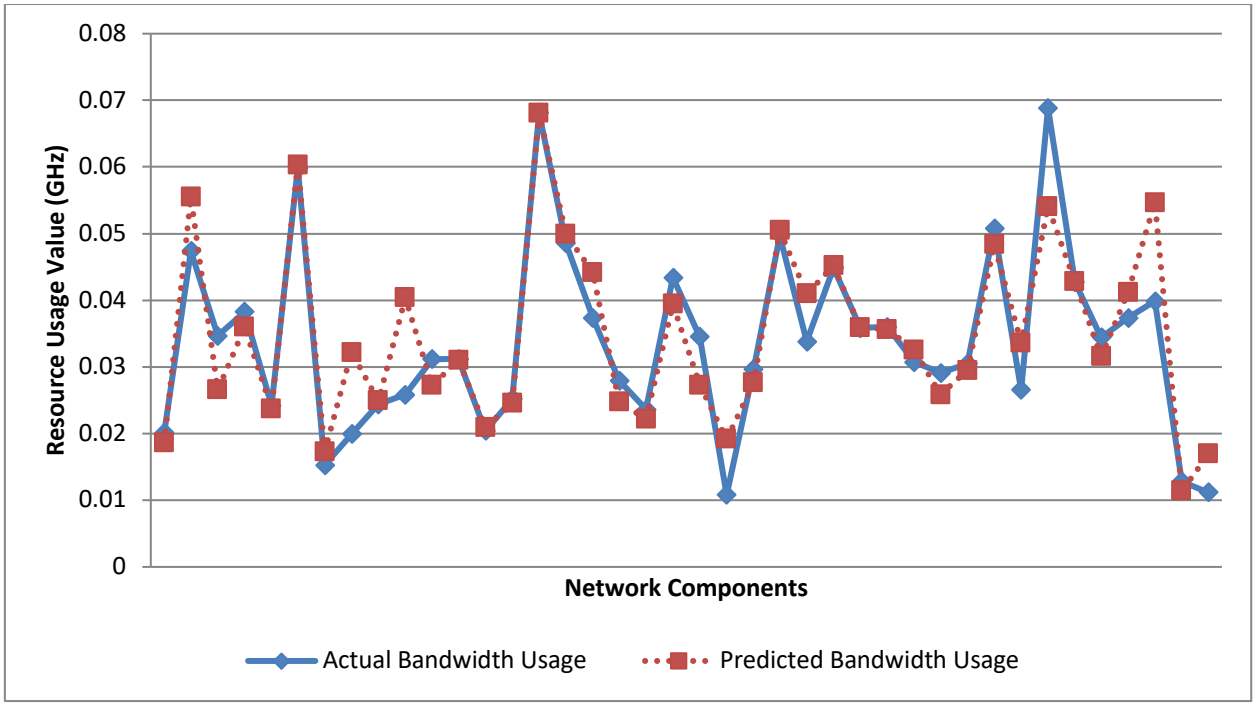
The best performance for the resource prediction ANN was produced with 91 neurons in the hidden layer using a trial-and-error approach with the mean squared error value (MSE), which is the averaged squared difference between inputs and outputs, of 0.0001.The correlation between the inputs and outputs which is also known as regression R is 0.87. The R value and MSE show that the error value of the outputs from the resource usage predictor is negligible. This implies, correlation between the input and output parameters of the ANN is optimum.

The best performance for the performance prediction ANN was at 96 neurons in the hidden layer using a trial-and-error approach, at the mean squared error value (MSE) of 0.085, and an R value of 0.79. The MSE and R values suggests that the instructions per cycle output values predicted by the performance assessor component of the autonomous intelligent system, although higher have a negligible error compared to that of the usage predictor ANN. Hence, the correlation between the outputs of the performance prediction ANN and its inputs is high.

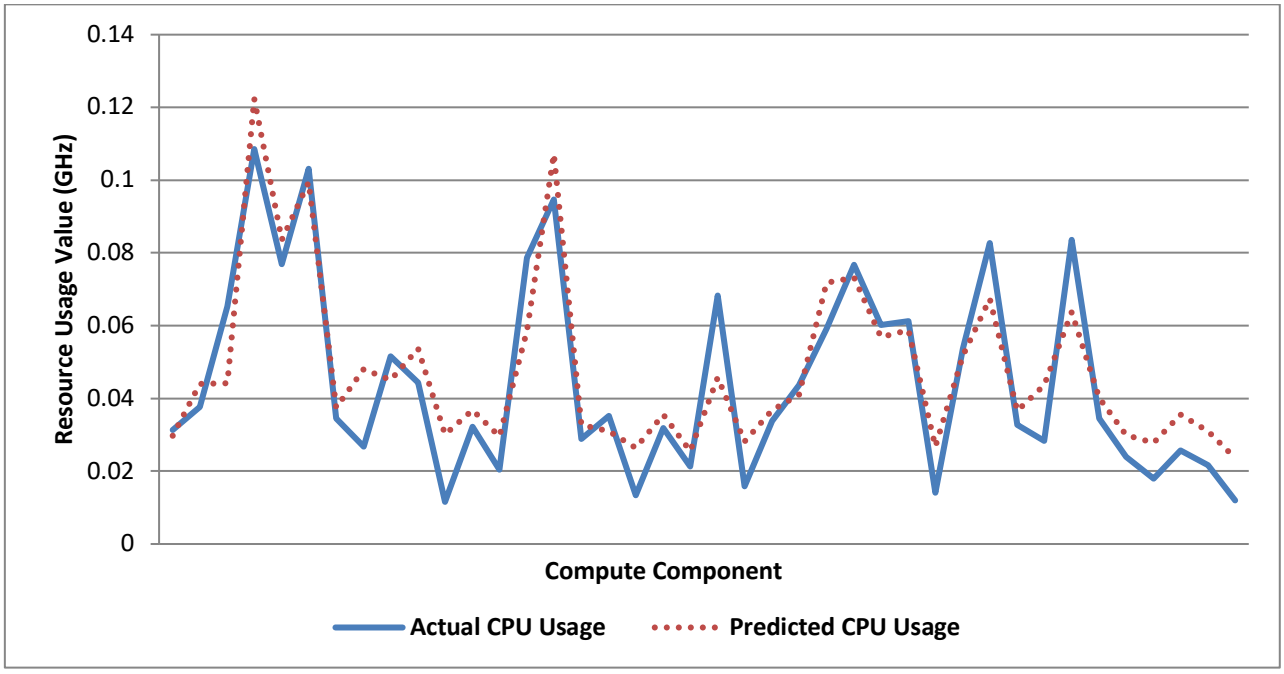
The overbooking ratio engine predicts overbooking ratios for the various network components in the SDCDCN based on values it receives from the resource usage predictor of the magnitude of network resources to be used for the next hour window. These predicted overbooking rates and the predicted performance of the network components are analysed and compared with the actual overbooking rates and performance observed from the Google cluster trace. We calculated four overbooking ratios and considered four different conditions for compute and network components in a heterogeneous SDCDCN in this experiment.

1. if the predicted performance value (P(x)) is greater than or equal to the maximum performance value of the SDCDCN components in the trace (i.e. $P(x) >= max$)
2. if the predicted performance value (P(x)) is greater than or equal to sum of the maximum performance value of the SDCDCN components and five times the standard deviation of this value (i.e., $P(x) >= max +5\sigma$).
3. if the predicted performance value (P(x)) is greater than or equal to sum of the maximum performance value of the SDCDCN components and seven times the standard deviation of this value (i.e., $P(x) >= max +7\sigma$).
4. if the predicted performance value (P(x)) is greater than or equal to sum of the maximum performance value of the SDCDCN components and ten times the standard deviation of this value (i.e., $P(x) >= max +10\sigma$).

Multiple predicted overbooking ratios which are purposed to enhance resource usage in the SDCDCN considering different performance indexes were calculated and used in this experiment to display the closeness of the performance constraints in order to validate the effectiveness of our intelligent overbooking algorithm. The range or the overbooking ratios was defined to show the effectiveness of the overbooking algorithms under resilient conditions.



(a)



(b)

Figure.5.4 Actual and predicted network resource usage

The results in Figure 5.4a and Figure 5.4b. indicates that the compute and network resources usage values predicted by the proposed autonomous overbooking system is very close to that of the actual usage values of Google's production datacentre because of the values of

the MSE and regression. From Figure 5.4a, the highest actual and predicted bandwidth usage observed during the experiment is 0.07 Gbps and the lowest bandwidth usage was observed at 0.01 Gbps.

It can also be observed from Figure 5.4b, that the highest actual CPU usage in the cloud datacentre is slightly more than 0.1 GHz whiles that of the predicted is 0.12 GHz. The lowest predicted CPU usage is slightly more than 0.02 GHz and the actual CPU usage is slightly lower than 0.02 GHz.

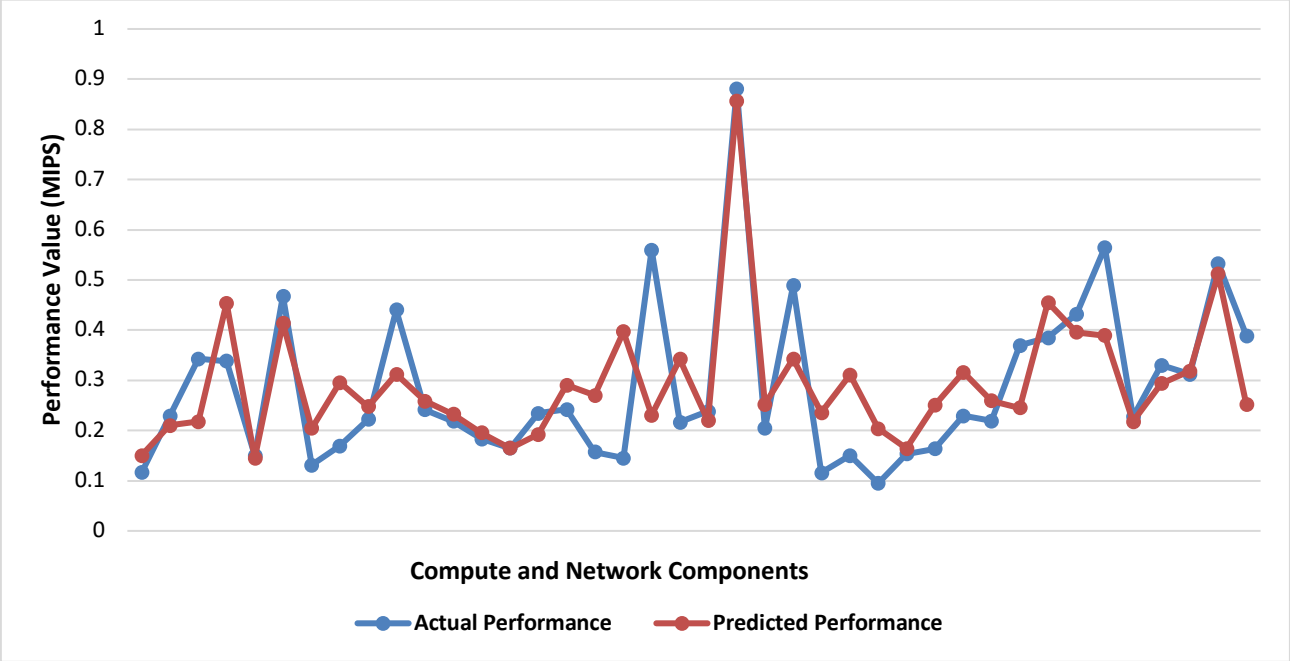


Figure.5.5. Actual and predicted IPC values

Figure 5.5 illustrates predicted instructions per cycle (IPC) values along with the actual IPC values for each host compute and network component. The hour, which was used for the prediction, 696th on the timeline is the same as that used by the overbooking ratio prediction engine. The predicted performance value follows the actual usage values closely because of a low MSE and a suitable R values for all host compute and network components in the training set. It is assumed that an increase in the input parameters would enable the ANN to reduce prediction errors further in order to enhance correlation between the input and output values.

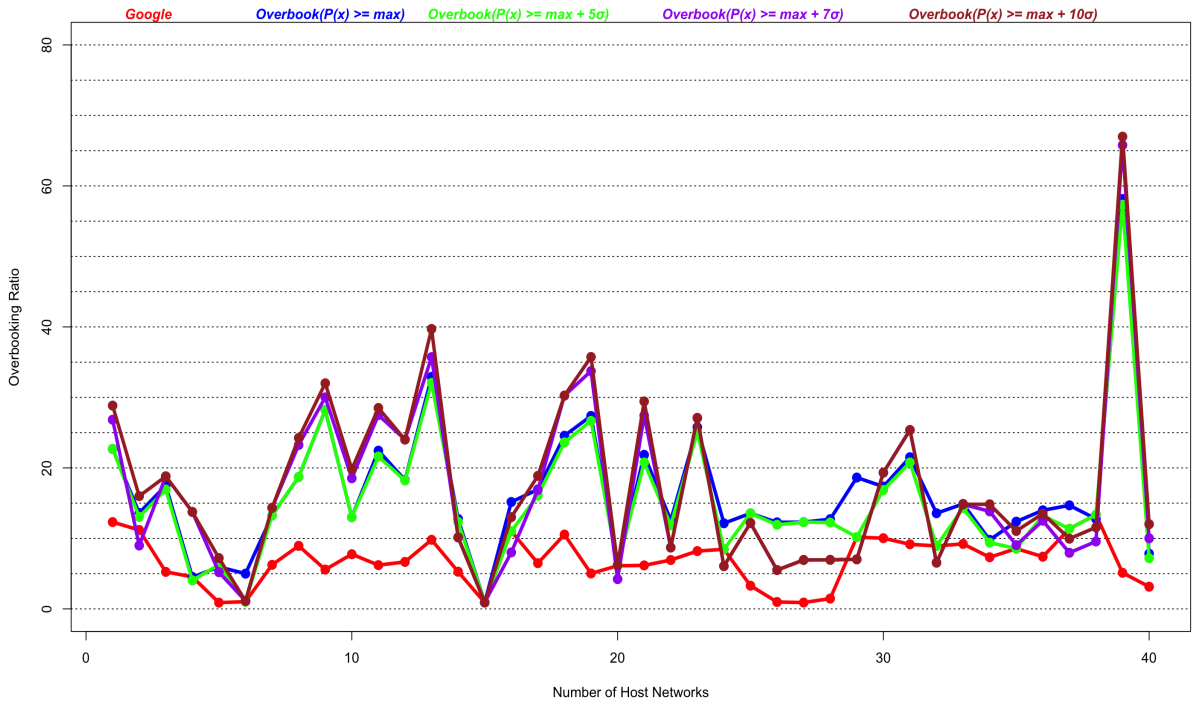


Figure.5.6. Google's host compute and network actual and Intelligent Overbooking Ratios for different Performance indexes

Figure 5.6 illustrates a comparison of our multiple intelligent overbooking ratios which were derived from the overbooking prediction engine at the 696th hour and that of the actual of Google's host and network components. It is observed from the diagram that our intelligent overbooking strategy predominantly predicts higher overbooking ratios for the host SDCDCN components without violating SLA compared to that of Google's overbooking.

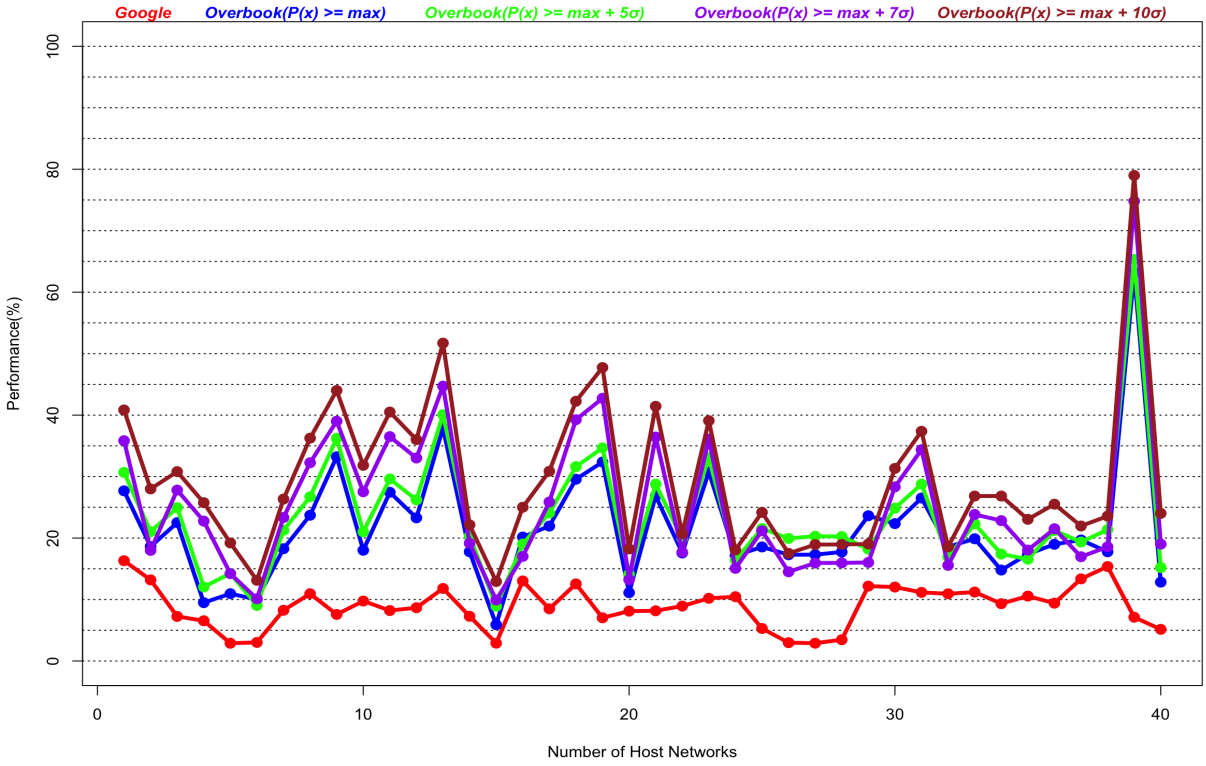


Figure.5.7. Google's host compute and network actual and Intelligent Predicted Performance rates for different Performance indexes

Figure 5.7 illustrates Google's host actual values on the 696th hour and the intelligent overbooking strategy predicted performance values associated with the overbooking ratios in Figure 5.6. An identical predicted performance values and that of actual performance based on the overbooking ratios above in Figure 5.6. would depict Violation in SLA and hence those host components exempted from overbooking by our overbooking strategy. However, as observed from Figure 5.7, the predicted performance values for the SDCDCN components under all the performance indexes are higher than the maximum performance values of the host components, which implies our intelligent overbooking strategy predicted with higher performance values for the next hour for the entire Software Defined Cloud Datacentre without violating SLAs in comparison with that of overbooking algorithm used in Google's production environment.

5.9. Summary

In this chapter, an intelligent autonomous overbooking strategy which predicts network resource usage and performance for the next hour in a SDCDCN while avoiding SLA violation was presented. Two ANN topologies were used by the proposed intelligent overbooking system to predict the network resource usage and performance of network components in the datacentre. The ANN was because, the neural network learning problem is a function optimization problem, where the best network parameters (weights and biases) are determined with higher precision in order to minimize network error. Data from Google cluster traces was used for training the ANN and for validation of the outputs. This was done with the use of two different portions or segments of the cluster traces.

Implementation and testbed experiments using the CloudSimHypervisor demonstrated that the proposed intelligent autonomous overbooking strategy predicted higher network resource usage and performance values without violating SLA based on all the overbooking ratios derived by the overbooking engine considering different performance indexes in comparison with that of the overbooking algorithm used in the production environment of Google's datacentres.

Chapter 6 Conclusion and Future Research

6.1 Conclusions and Discussion

The emergence of softwarisation of computer networks with the use of modern technologies such as SDN, NFV, edge computing and SFC in a cloud computing environment has enabled Network Cloud Unification and expanded usage potential of Software Defined Cloud Networks. SDCN streamlines complex network operations with orchestration, management and automation. The network virtualization hypervisor dynamically manages traffic from network user requests of the both monogenous and heterogeneous cloud datacentre networks. Its ability of controllability and dynamically adjust to on demand network requirements and utilization has led to significant enhancement in the quality of service delivery in cloud datacentre networks. Although each of the networking technologies (SDN, NFV, Edge computing) can function independently, integrating them leverages their corporate advantages such as programmability and customization of network control logic, management and orchestration, etc., in securing innovation in cloud datacentre network management. Networking and cloud unification provide efficient mechanisms and solutions to dynamically optimize simultaneous compute and network resource provisioning across homogeneous (*intra*) and heterogeneous (*inter*) Software Defined Cloud Networks.

However, joint provisioning of compute and network resources in cloud datacentre on a large-scale can complicated and challenging. As a result, this thesis conducted an extensive research on how deploying a Network Virtualization Hypervisor by integrating the network control attributes of SDN and the network virtualization attributes of NFV in a cloud computing environment can provide enhance strategies to optimize joint compute and network resource provisioning in cloud datacentre networks.

Considering the details, the background of key technologies enabling Software Defined Cloud Datacentre Networks and the aims and objectives of this study with regards to the research problem are outlined in Chapter 1.

Chapter 2 commenced with the definition of terms in order to clarify ambiguous terminologies and proposed concepts in the literature surveyed for this research work. Furthermore, comprehensive review of literature survey and taxonomy of Software Defined Cloud Networks for energy efficiency, performance and virtualization was presented in Chapter 2. This chapter set out the state -of- the- art SDCN architectures and empirical tools and toolkits for performance evaluation. Finally, Chapter 2 discussed in detail an analysis of current research which assisted in identifying research gaps and finding research challenges which shaped the direction of the thesis.

In Chapter 3, a Network Virtualization Hypervisor from SDN and NFV components in a cloud datacentre was modelled. The Network Hypervisor was enabled with features and functions to abstract the entire network resources (compute, switch & link) of physical networks and stitch them together to create independent virtual networks. Based on the model, a discrete event simulation application framework was proposed and implemented in the extension of CloudSimSDN. The CloudSimHypervisor is enabled with additional capability to manage, virtual networks, tenant controllers and the programmable Network Virtualization Hypervisor. Physical host network resources were modelled to be managed by the network hypervisor. Network User requests were abstracted with the hypervisor for the convenience of conducting reproducible testbed simulation experiments.

With the development of CloudSimHypervisor, two novel heuristics are proposed for joint compute and network resource provisioning in Chapters 4 and 5. An intelligent self-configuring overbooking strategy which autonomously changes with regards to workload in real-time was suggested in Chapter 4. A novel network resource allocation system in which the network virtualization hypervisor takes input from network traffic and works with the host and link selection component, the overbooking ratio engine and other components to derive suitable overbooking ratios for allocating the VNRs to host SDCDCN components was presented. The dynamic overbooking strategy also assures that both SLA satisfaction and energy efficiency is maintained with unpredictable changes in workload, datacentre status, and network user demands. The performance evaluation conducted with real-world traces showed that the effectiveness of the proposed algorithm varies depending on the characteristic of the workload.

Chapter 5 discussed a performance -aware intelligent overbooking strategy which is able to predict resource usage and performance of host compute and network components for the next hour in the SDCDCN. A novel intelligent overbooking system which applied two ANN structures to predict the resource usage and performance values of the host compute and network components for the next hour was proposed. These two ANN topologies were trained with data from Google cluster trace. The predicted network resource usage and performance values were validated in comparison with Google's actual overbooking ratio and multiple overbooking ratios under different performance conditions.

6.2. Future Directions

Although the methods investigated in this thesis contributes to exploiting the potential of software defined networks and cloud unification for joint resource provisioning, there are other aspects which require further comprehensive exploration.

6.2.1 Supporting Big Data Applications in 5G and 6G technologies

Cloud computing has become a fundamental infrastructure for Big Data applications with its massive resource requirements for computation jobs and network transmission. Using Software Defined Cloud Networks infrastructure with the network virtualization hypervisor can bring vital benefits to Big Data applications to reduce network bottlenecks during the application processing. For instance, aggregation of the mapped data from mappers to the reducer can cause a network burst in a short time when the mapping processes are completed at the similar time. The network hypervisor can be exploited for network-aware scheduling of Big Data workloads. Network bandwidth management is considered with use of SDCN scheduling to guarantee data locality and optimize task assignment among available workers. According to (Letaief, Chen, Shi, et al., 2019), big data analytics is the first natural application of Artificial intelligence, AI. The network virtualisation hypervisor has the capability to deploy the four defined applications of data analytics in 5G and 6G technologies. Descriptive analytics applies historical data to get insights on network performance, traffic profile, channel conditions, user perspectives and etc. These applications are descriptive analytics, diagnostic analytics, predictive analytics, and prescriptive analytics. This type of analytics enhances the situational awareness of network

operators and service providers. Diagnostic analytics enable autonomous detection of network faults and service impairments. It identifies the major causes of anomalies in SDCN and ultimately improves its reliability and security. Predictive analytics use data to predict future events such as traffic patterns, user locations, user behaviour and preference, and resource availability. Prescriptive analytics take advantage of the predictions to suggest decision options for resource allocation, network slicing and virtualization, cache placement and edge computing (Letaief, Chen, Shi, et al., 2019). In addition to the computation processing time of the workload, network limitation between workers can also be considered when scheduling and distributing jobs.

6.2.2 Enhancing Security and Reliability in SDCN in 5G and 6G technologies

For public cloud providers, security of the datacentre and their data is a crucial factor of the business that must be fulfilled. While security in cloud datacentres has been explored extensively with diverse approaches, security of SDCN usage, in particular, still requires investigation to provide insurance for service providers applying SDN and NFV in their datacentres. Current security issues include encrypting packets to and from the network virtualization hypervisor. As there is regular flow of network traffic and transmission between the network hypervisor and physical components as well as between the hypervisor and virtual network requests, protecting the network hypervisor from intrusions and irregular access is imperative. Furthermore, researchers put more effort on investigating the vulnerability of SDCN itself from DDoS attack (Jantila & Chaipah, 2016) and the security issue of network control (Dargahi, Caponi, Ambrosin, Bianchi, & Conti, 2017). Security is a critical issue for 6G wireless networks, especially when the STIN technique is employed. In 6G, besides the conventional physical layer security, other types of security, such as integrated network security, should be jointly considered (Yang, P., Xiao, Y., Xiao, M., & Li, S., 2019). Therefore, new security approaches, relying on low complexity and having high security levels, are worth more intensive study. Cyber and physical layer security issues within wireless networks are widespread in daily life. As a result, for wireless computing among individuals and communities, privacy leakage is a predominant concern caused by perpetual data uploading, caching, and transmitting. Providing wireless computing with trusted communications is a crucial objective. Security should therefore be considered as a

basic performance requirement of wireless computing in 6G and is referred to as secure wireless computing for private data (SWCPD) (Yang, P., Xiao, Y., Xiao, M., & Li, S., 2019).

For SDCN, it is also critical to assure the reliability and availability of the network virtualization hypervisor and switches to maintain the proper operation of network system under unexpected threats and failures. Therefore, it is important to investigate these aspects of improving security and reliability of software defined networks and cloud network unification to build fully automated and fault-tolerant Software-Defined Clouds.

References

[1] Mobile Network Design and Deployment: How Incumbent Operators Plan for Technology Upgrades and Related Spectrum Needs. Rysavy Research, June 2012.

[2] Beyond LTE: Enabling the Mobile Broadband Explosion, LTE and 5G Innovation: Igniting Mobile Broadband. Rysavy Research/4G Americas, August 2015.

[3] Abts, D. et al. (2010) 'Energy proportional datacentre networks', Proceedings - International Symposium on Computer Architecture, pp. 338–347. doi: 10.1145/1815961.1816004

[4] Alford, M. G. (1988) 'Q-clouds', Nuclear Physics, Section B, 298(2), pp. 323–332. doi: 10.1016/0550-3213(88)90269-6.

[5] Amarasinghe, H., Jarray, A. and Karmouch, A. (2017) 'Fault-tolerant IaaS management for networked cloud infrastructure with SDN', IEEE International Conference on Communications. IEEE, pp. 1–7. doi: 10.1109/ICC.2017.7996342.

[6] Antequera, R. B. et al. (2018) 'ADON: Application-Driven Overlay Network-as-a-Service for Data-Intensive Science', IEEE Transactions on Cloud Computing. IEEE, 6(3), pp. 640–655. doi: 10.1109/TCC.2015.2511753.

[7] Aujla, G. S. et al. (2018) 'Optimal decision making for big data processing at edge-cloud environment: An SDN perspective', IEEE Transactions on Industrial Informatics. IEEE, 14(2), pp. 778–789. doi: 10.1109/TII.2017.2738841.

[8] Azizian, M., Cherkaoui, S. and Hafid, A. S. (2017) 'Vehicle Software Updates Distribution with SDN and Cloud Computing', IEEE Communications Magazine. IEEE, 55(8), pp. 74–79. doi: 10.1109/MCOM.2017.1601161.

[9] Baset, S. A., Wang, L. and Tang, C. (2012) 'Towards an understanding of oversubscription in cloud', the 2nd USENIX conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services. USENIX Association. pp. 7–7. Available at: http://www.cs.columbia.edu/~salman/publications/oversub-hotice-2012.pdf.

[10] Beck, M. T. et al. (2015) 'Distributed and scalable embedding of virtual networks', Journal of Network and Computer Applications. Elsevier, 56, pp. 124–136. doi: 10.1016/j.jnca.2015.06.012

[11] Belbekkouche, A., Hasan, M. M. and Karmouch, A. (2012) 'Resource discovery and

allocation in network virtualization', IEEE Communications Surveys and Tutorials. doi: 10.1109/SURV.2011.122811.00060.

[12] Beloglazov, A. et al. (2011) A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems, Advances in Computers. doi: 10.1016/B978-0-12-385512-1.00003-7.

[13] Beloglazov, A., Abawajy, J. and Buyya, R. (2012) 'Energy-aware resource allocation heuristics for efficient management of datacenters for Cloud computing', Future Generation Computer Systems. doi: 10.1016/j.future.2011.04.017.

[14] Beloglazov, A. and Buyya, R. (2010) 'Energy efficient resource management in virtualized cloud data centers', in CCGrid 2010 - 10th IEEE/ACM International Conference on Cluster, Cloud, and Grid Computing. doi: 10.1109/CCGRID.2010.46.

[15] Blenk, A., Basta, A., Zerwas, J., et al. (2016) 'Pairing SDN with network virtualization: The network hypervisor placement problem', 2015 IEEE Conference on Network Function Virtualization and Software Defined Network, NFV-SDN 2015, pp. 198–204. doi: 10.1109/NFV-SDN.2015.7387427.

[16] Blenk, A., Basta, A., Reisslein, M., et al. (2016) 'Survey on network virtualization hypervisors for software defined networking', IEEE Communications Surveys and Tutorials, 18(1), pp. 655–685. doi: 10.1109/COMST.2015.2489183.

[17] Blenk, A., Basta, A. and Kellerer, W. (2015) 'HyperFlex: An SDN virtualization architecture with flexible hypervisor function allocation', Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, pp. 397–405. doi: 10.1109/INM.2015.7140316.

[18] Blenk, A. and Kellerer, W. (2013) 'Traffic pattern based virtual network embedding', pp. 23–26. doi: 10.1145/2537148.2537151.

[19] Bonafiglia, R. et al. (2017) 'End-to-end service orchestration across SDN and cloud computing domains', 2017 IEEE Conference on Network Softwarization: Softwarization Sustaining a Hyper-Connected World: en Route to 5G, NetSoft 2017, (October 2019). doi: 10.1109/NETSOFT.2017.8004234.

[20] Braun, W. and Menth, M. (2014) 'Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices', Future Internet, 6(2), pp. 302–336. doi: 10.3390/fi6020302.

[21] Buyya, R. et al. (2014) 'Software-Defined Cloud Computing: Architectural elements and open challenges', in Proceedings of the 2014 International Conference on

Advances in Computing, Communications and Informatics, ICACCI 2014. doi: 10.1109/ICACCI.2014.6968661.

[22] Caglar, Faruk and Gokhale, A. (2014) 'IOverbook: Intelligent resource-overbooking to support soft real-time applications in the cloud', IEEE International Conference on Cloud Computing, CLOUD, pp. 538–545. doi: 10.1109/CLOUD.2014.78

[23] Caglar, F and Gokhale, A. (2014) 'iOverbook: managing cloud-based soft real-time applications in a resource-overbooked data center', … Conference on Cloud Computing (CLOUD' Available at: http://www.dre.vanderbilt.edu/~gokhale/WWW/papers/RTAS2014.pdf.

[24] Calheiros, R. N. et al. (2011) 'CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms', Software - Practice and Experience. doi: 10.1002/spe.995.

[25] Callegati, F. et al. (2016) 'SDN for dynamic NFV deployment', IEEE Communications Magazine. IEEE, 54(10), pp. 89–95. doi: 10.1109/MCOM.2016.7588275.

[26] Cerroni, W. et al. (2015) 'Cross-layer resource orchestration for cloud service delivery: A seamless SDN approach', Computer Networks. Elsevier Ltd., 87, pp. 16–32. doi: 10.1016/j.comnet.2015.05.008.

[27] Citron, D. and Zlotnick, A. (2011) 'Testing large-scale cloud management', Ibm Journal of Research and Development, 55(6), pp. 1–10. doi: 10.1147/JRD.2011.2170913.

[28] Cui, L., Yu, F. R. and Yan, Q. (2016) 'When big data meets software-defined networking: SDN for big data and big data for SDN', IEEE Network. IEEE, 30(1), pp. 58–65. doi: 10.1109/MNET.2016.7389832.

[29] Cziva, R. et al. (2016) 'SDN-Based Virtual Machine Management for Cloud Data Centers', IEEE Transactions on Network and Service Management. IEEE, 13(2), pp. 212–225. doi: 10.1109/TNSM.2016.2528220.

[30] Dias de Assunção, M. et al. (2017) 'Designing and building SDN testbeds for energy-aware traffic engineering services', Photonic Network Communications, 34(3), pp. 396–410. doi: 10.1007/s11107-017-0709-9.

[31] Duan, Q., Yan, Y. and Vasilakos, A. V. (2012) 'A survey on service-oriented network virtualization toward convergence of networking and cloud computing', IEEE Transactions on Network and Service Management, 9(4), pp. 373–392. doi:

10.1109/TNSM.2012.113012.120310.

[32] Duan, Q., Yan, Y. and Vasilakos, A. V (2012) 'A Survey on Service-Oriented Network Virtualization Toward Convergence of Networking and Cloud Computing', 9(4), pp. 373–392.

[33] Egilmez, H. E. et al. (2012) 'OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks', 2012 Conference Handbook - Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2012.

[34] Ersoz, D., Yousif, M. S. and Das, C. R. (2007) 'Characterizing network traffic in a cluster-based, multi-tier data center', Proceedings - International Conference on Distributed Computing Systems, (1). doi: 10.1109/ICDCS.2007.90.

[35] Esposito, F., Matta, I. and Ishakian, V. (2013) 'Slice embedding solutions for distributed service architectures', ACM Computing Surveys, 46(1), pp. 1–29. doi: 10.1145/2522968.2522974.

[36] Fang, W. et al. (2013) 'VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers', Computer Networks. Elsevier B.V., 57(1), pp. 179–196. doi: 10.1016/j.comnet.2012.09.008.

[37] Fichera, S. et al. (2017) 'On experimenting 5G: Testbed set-up for SDN orchestration across network cloud and IoT domains', 2017 IEEE Conference on Network Softwarization: Softwarization Sustaining a Hyper-Connected World: en Route to 5G, NetSoft 2017. doi: 10.1109/NETSOFT.2017.8004245.

[38] Fischer, A. et al. (2012) 'Virtual Network Embedding: A Survey', 15(4), pp. 1888–1906.

[39] Fischer, A. et al. (2013) 'Virtual network embedding: A survey', IEEE Communications Surveys and Tutorials, 15(4), pp. 1888–1906. doi: 10.1109/SURV.2013.013013.00155.

[40] Garg, S. K. and Buyya, R. (2011) 'NetworkCloudSim: Modelling parallel applications in cloud simulations', Proceedings - 2011 4th IEEE International Conference on Utility and Cloud Computing, UCC 2011, (Vm), pp. 105–113. doi: 10.1109/UCC.2011.24.

[41] Gharbaoui, M. et al. (2016) 'Cloud and network orchestration in SDN data centers: Design principles and performance evaluation', Computer Networks. Elsevier B.V., 108, pp. 279–295. doi: 10.1016/j.comnet.2016.08.029.

[42] Gray, K. and Nadeau, T. D. (2016) ‘Network Function Virtualization’, p. 270. doi: 10.1109/MIC.2016.112.

[43] Habibi, P., Mokhtari, M. and Sabaei, M. (2017) ‘QRVE: QoS-aware routing and energy-efficient VM Placement for Software-Defined DataCenter Networks’, 2016 8th International Symposium on Telecommunications, IST 2016. IEEE, pp. 533–539. doi: 10.1109/ISTEL.2016.7881879.

[44] Heller, B. et al. (2009) ‘ElasticTree: Saving energy in data center networks’, Submitted to ACM. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:ElasticTree+:+Saving+Energy+in+Data+Center+Networks#0.

[45] Heller, B., Sherwood, R. and McKeown, N. (2012) ‘The controller placement problem’, ACM SIGCOMM Computer Communication Review, 42(4), p. 473. doi: 10.1145/2377677.2377767.

[46] Huang, S. and Griffioen, J. (2013) ‘Network hypervisors: Managing the emerging SDN chaos’, Proceedings - International Conference on Computer Communications and Networks, ICCCN. IEEE, pp. 1–7. doi: 10.1109/ICCCN.2013.6614160.

[47] Ishimori, A. et al. (2013) ‘Control of multiple packet schedulers for improving QoS on OpenFlow/SDN networking’, Proceedings - 2013 2nd European Workshop on Software Defined Networks, EWSDN 2013, pp. 81–86. doi: 10.1109/EWSDN.2013.20.

[48] Jain, S. et al. (2013) ‘B4: Experience with a globally-deployed software defined WAN’, Computer Communication Review, 43(4), pp. 3–14. doi: 10.1145/2534169.2486019.

[49] Jeong, K. and Figueiredo, R. (2016) ‘Self-configuring software-defined overlay bypass for seamless inter- and intra-cloud virtual networking’, HPDC 2016 - Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing, pp. 153–164. doi: 10.1145/2907294.2907318.

[50] Jiang, J. W. et al. (2012) ‘Joint VM placement and routing for data center traffic engineering’, Proceedings - IEEE INFOCOM. IEEE, pp. 2876–2880. doi: 10.1109/INFCOM.2012.6195719.

[51] Jin, H. et al. (2013) ‘Joint host-network optimization for energy-efficient data center networking’, Proceedings - IEEE 27th International Parallel and Distributed Processing Symposium, IPDPS 2013. IEEE, pp. 623–634. doi: 10.1109/IPDPS.2013.100.

[52] Kapil, D., Pilli, E. S. and Joshi, R. C. (2013) 'Live virtual machine migration techniques: Survey and research challenges', Proceedings of the 2013 3rd IEEE International Advance Computing Conference, IACC 2013. IEEE, pp. 963–969. doi: 10.1109/IAdCC.2013.6514357.

[53] Khan, A. et al. (2012) 'Network virtualization: A hypervisor for the internet?', in IEEE Communications Magazine. doi: 10.1109/MCOM.2012.6122544.

[54] Kim, Y. et al. (2016) 'SDN-based orchestration for interworking cloud and transport networks', 2016 International Conference on Information and Communication Technology Convergence, ICTC 2016. IEEE, pp. 303–307. doi: 10.1109/ICTC.2016.7763490.

[55] Kliazovich, D., Bouvry, P. and Khan, S. U. (2012) 'GreenCloud: A packet-level simulator of energy-aware cloud computing data centers', Journal of Supercomputing. doi: 10.1007/s11227-010-0504-1.

[56] Kreutz, D. et al. (2015) 'Software-defined networking: A comprehensive survey', Proceedings of the IEEE. doi: 10.1109/JPROC.2014.2371999.

[57] Lantz, B., Heller, B. and McKeown, N. (2010) 'A Network in a Laptop: Rapid Prototyping for Software-Defined Networks', in Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets). doi: 10.1145/1868447.1868466.

[58] Leivadeas, A. et al. (2017) 'Optimal virtualized network function allocation for an SDN enabled cloud', Computer Standards and Interfaces. Elsevier, 54(April 2016), pp. 266–278. doi: 10.1016/j.csi.2017.01.001.

[59] Licciardello, M. et al. (2017) 'Performance evaluation of abstraction models for orchestration of distributed datacentre networks', International Conference on Transparent Optical Networks, pp. 9–12. doi: 10.1109/ICTON.2017.8025184.

[60] Lowe, S. (2013) 'Best Practices for Oversubscription of CPU, Memory and Storage in vSphere Virtual Environments', VMware's White paper, available at: https:// Available at:http://virtualizationreview.com/~/media/23865B1BB1134F239CE170AFBB85672A.pdf.

[61] Martini, B. et al. (2016) 'Design and evaluation of SDN-based orchestration system for cloud datacenters', 2016 IEEE International Conference on Communications, ICC 2016. IEEE, pp. 1–6. doi: 10.1109/ICC.2016.7511144.

[62] McKeown, N. et al. (2008) 'Sigcomm08_Openflow.Pdf', 38(2), pp. 69–74. doi:

10.1145/1355734.1355746.

[63] Mechtri, M. et al. (2013) ‘SDN for inter cloud networking’, SDN4FNS 2013 - 2013 Workshop on Software Defined Networks for Future Networks and Services. doi: 10.1109/SDN4FNS.2013.6702552

[64] Medina, V. and GARCíA, J. M. (2014) ‘A survey of migration mechanisms of virtual machines’, ACM Computing Surveys, 46(3). doi: 10.1145/2492705.

[65] Mijumbi, R. et al. (2016) ‘Network function virtualization: State-of-the-art and research challenges’, IEEE Communications Surveys and Tutorials, 18(1). doi: 10.1109/COMST.2015.2477041.

[66] Moreno, I. S. and Xu, J. (2011) ‘Customer-aware resource overallocation to improve energy efficiency in realtime Cloud Computing data centers’, Proceedings - 2011 IEEE International Conference on Service-Oriented Computing and Applications, SOCA 2011. IEEE, pp. 1–8. doi: 10.1109/SOCA.2011.6166239.

[67] Moreno, I. S. and Xu, J. (2012) ‘Neural network-based overallocation for improved energy-efficiency in real-time cloud environments’, Proceedings - 2012 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, ISORC 2012. IEEE, pp. 119–126. doi: 10.1109/ISORC.2012.24.

[68] Networks, I. N. C. (no date) ‘A Definitive Guide to Successful Migrations’.

[69] Núñez, A. et al. (2012) ‘ICanCloud: A Flexible and Scalable Cloud Infrastructure Simulator’, Journal of Grid Computing. doi: 10.1007/s10723-012-9208-5.

[70] Pakbaznia, E. and Pedram, M. (2009) ‘Minimizing data center cooling and server power costs’, Proceedings of the International Symposium on Low Power Electronics and Design, pp. 145–150. doi: 10.1145/1594233.1594268.

[71] Panagopoulos, D. J. (2011) ‘a Nalyzing the H Ealth I Mpacts of M Odern’, 17, pp. 1–55. doi: 10.29268/stcc.2014.2.1.

[72] Petri, I. et al. (2015a) ‘Integrating software defined networks within a cloud federation’, Proceedings - 2015 IEEE/ACM 15th International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2015, pp. 179–188. doi: 10.1109/CCGrid.2015.11.

[73] Petri, I. et al. (2015b) ‘Integrating software defined networks within a cloud federation’, Proceedings - 2015 IEEE/ACM 15th International Symposium on

Cluster, Cloud, and Grid Computing, CCGrid 2015. IEEE, pp. 179–188. doi: 10.1109/CCGrid.2015.11.

[74] Popa, L. et al. (2012) 'FairCloud: Sharing the network in cloud computing', Computer Communication Review, 42(4), pp. 187–198. doi: 10.1145/2377677.2377717.

[75] Pu, X. et al. (2010) 'Understanding performance interference of I/O workload in virtualized cloud environments', Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010. IEEE, (Vmm), pp. 51–58. doi: 10.1109/CLOUD.2010.65.

[76] Risdianto, A. C., Shin, J. S. and Kim, J. (2016) 'Deployment and evaluation of software-defined inter-connections for multi-domain federated SDN-Cloud', ACM International Conference Proceeding Series, 15-17-June, pp. 118–121. doi: 10.1145/2935663.2935683.

[77] Samant, D. and Bellur, U. (2016) 'Handling boot storms in virtualized data Centers-A survey', ACM Computing Surveys, 49(1), pp. 1–36. doi: 10.1145/2932709.

[78] Sherwood, R. et al. (2009) 'Openflow-Tr-2009-1-Flowvisor.Pdf'.

[79] Smith, J. E. and Nair, R. (2005) 'The architecture of virtual machines Smith, J.E. Ravi Nair', IEEE Computer, 38(5), pp. 32–38. Available at: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1430629.

[80] Son, J. (2013) 'Automated decision system for efficient resource selection and allocation in inter-clouds, June 2013.

[81] Son, J. et al. (2015) 'CloudSimSDN: Modeling and simulation of software-defined cloud data centers', Proceedings - 2015 IEEE/ACM 15th International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2015, pp. 475–484. doi: 10.1109/CCGrid.2015.87.

[82] Son, J. et al. (2017) 'SLA-Aware and Energy-Efficient Dynamic Overbooking in SDN-Based Cloud Data Centers', IEEE Transactions on Sustainable Computing. IEEE, 2(2), pp. 76–89. doi: 10.1109/tsusc.2017.2702164.

[83] Son, J. (2018) 'Integrated Provisioning of Compute and Network Resources in Software-Defined Cloud Data Centers', (January), p. 208. Available at: https://minerva access.unimelb.edu.au/bitstream/handle/11343/212287/thesis.pdf?sequence=1&isAllowed=y.

[84] Son, J. and Buyya, R. (2018a) 'A Taxonomy of Software-Defined Networking

(SDN) -Enabled Cloud Computing', 51(3).

[85] Son, J. and Buyya, R. (2018b) 'A Taxonomy of Software-Defined Networking (SDN)-Enabled Cloud Computing', ACM Computing Surveys, 51(3), pp. 1–36. doi: 10.1145/3190617.

[86] Son, J. and Buyya, R. (2019) 'CloudSimSDN-NFV: Modeling and Simulation of Network Function Virtualization and Service Function Chaining in Edge Computing Environments'.

[87] Souza, F. R. De et al. (2017) 'QoS-aware virtual infrastructures allocation on SDN-based clouds', Proceedings - 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID 2017. IEEE, pp. 120–129. doi: 10.1109/CCGRID.2017.57.

[88] Spadaro, S. et al. (2016) 'Resource orchestration in SDN-based future optical data centres', 2016 20th International Conference on Optical Network Design and Modelling, ONDM 2016, (Ondm). doi: 10.1109/ONDM.2016.7494057.

[89] Stojmenovic, I. (2008) 'Simulations in wireless sensor and ad hoc networks: Matching and advancing models, metrics, and solutions', IEEE Communications Magazine. IEEE, 46(12), pp. 102–107. doi: 10.1109/MCOM.2008.4689215.

[90] Sufyan Beg, M. M. and Ahmad, N. (2002) 'Genetic Algorithm Based Rank Aggregation for the Web', Proceedings of the Joint Conference on Information Sciences, 6, pp. 329–333.

[91] Therrien, E. et al. (2012) 'Integrating medicinal chemistry, organic/combinatorial chemistry, and computational chemistry for the discovery of selective estrogen receptor modulators with FORECASTER, a novel platform for drug discovery', Journal of Chemical Information and Modelling, 52(1), pp. 210–224. doi: 10.1021/ci2004779.

[92] Waldspurger, Carl A. (2002) 'Memory resource management in VMware ESX server', Operating Systems Review (ACM), 36(Special Issue), pp. 181–194. doi: 10.1145/844128.844146.

[93] Wang, R., Butnariu, D. and Rexford, J. (2011) 'OpenFlow-Based Server Load Balancing Gone Wild Into the Wild : Core Ideas', Hot-ICE'11 Proceedings of the 11th USENIX conference on Hot topics in management of internet, cloud, and enterprise networks and services works and services, p. 12.

[94] Wang, W. et al. (2014) 'Autonomic QoS management mechanism in Software Defined Network', China Communications. China Institute of Communications,

11(7), pp. 13–23. doi: 10.1109/CC.2014.6895381.

[95] Wang, Xiaodong et al. (2012) ‘CARPO: Correlation-aware power optimization in data center networks’, Proceedings - IEEE INFOCOM. IEEE, pp. 1125–1133. doi: 10.1109/INFCOM.2012.6195471.

[96] Wu, Y. et al. (2017) ‘Orchestrating bulk data transfers across geo-distributed datacenters’, IEEE Transactions on Cloud Computing, 5(1), pp. 112–125. doi: 10.1109/TCC.2015.2389842.

[97] Zhang, X., Tune, E. and Hagmann, R. (2013) ‘CPI2: CPU performance isolation for shared compute clusters’, in EuroSys’13. doi: 10.1145/2465351.2465388.

[98] Zheng, K. et al. (2014) ‘Joint power optimization of data center network and servers with correlation analysis’, Proceedings - IEEE INFOCOM, pp. 2598–2606. doi: 10.1109/INFOCOM.2014.6848207.

[99] Zheng, K. et al. (2017) ‘PowerNetS: Coordinating Data Center Network with Servers and Cooling for Power Optimization’, IEEE Transactions on Network and Service Management, 14(3), pp. 661–675. doi: 10.1109/TNSM.2017.2711567.

[100] Zheng, K. and Wang, X. (2017) ‘Dynamic Control of Flow Completion Time for Power Efficiency of Data Center Networks’, Proceedings - International Conference on Distributed Computing Systems. IEEE, pp. 340–350. doi: 10.1109/ICDCS.2017.138.

[101] P. Qin, B. Dai, B. Huang, and G. Xu, “Bandwidth-aware scheduling with sdn in hadoop: A new trend for big data,” IEEE Systems Journal, vol. 11, no. 4, pp. 2337–2344, Dec 2017.

[102] S. Jantila and K. Chaipah, “A security analysis of a hybrid mechanism to defend ddos attacks in sdn,” Procedia Computer Science, vol. 86, no. Supplement C, pp. 437–440,2016.[Online].Available:http://www.sciencedirect.com/science/article/pii/S1877050916304082

[103] T. Dargahi, A. Caponi, M. Ambrosin, G. Bianchi, and M. Conti, “A survey on the security of stateful sdn data planes,” IEEE Communications Surveys Tutorials, vol. 19, no. 3, pp. 1701–1725, third quarter 2017.

[104] A. Xie, X. Wang, W. Wang, and S. Lu, “Designing a disaster-resilient network with software defined networking,” in Proceedings of the IEEE 22nd International Symposium of Quality of Service (IWQoS), May 2014, pp. 135–140.

[105] R. Buyya, R. N. Calheiros, J. Son, A. V. Dastjerdi, and Y. Yoon, “Software-defined cloud computing: Architectural elements and open challenges,” in Proceedings of the 3rd International Conference on Advances in Computing, Communications and Informatics, IEEE. IEEE Press, 2014.

[106] Y. Jararweh, M. Al-Ayyoub, A. Darabseh, E. Benkhelifa, M. Vouk, and A. Rindos, “Software defined cloud: Survey, system and evaluation,” Future Generation Computer Systems, vol. 58, pp. 56 – 74, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X15003283

[107] W. Wendong, Q. Qinglei, G. Xiangyang, H. Yannan, and Q. Xirong, “Autonomic QoS management mechanism in software defined network,” Communications, China, vol. 11, no. 7, pp. 13–23, July 2014

[108] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. H¨ olzle, S. Stuart, and A. Vahdat, “B4: Experience with a globally-deployed software defined wan,” in Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM. New York, NY, USA: ACM,2013, pp. 3–14. [Online]. Available: http://doi.acm.org/10.1145/2486001.2486019

[109] S. Seetharam, P. Calyam, and T. Beyene, “ADON: Application-driven overlay network-as-a-service for data-intensive science,” in Proceedings of the 2014 IEEE 3rd International Conference on Cloud Networking, Oct 2014, pp. 313–319.

[110] A. C. Risdianto, J.-S. Shin, and J. W. Kim, “Deployment and evaluation of software-defined inter-connections for multi-domain federated SDN-cloud,” in Proceedings of the 11th International Conference on Future Internet Technologies. New York, NY, USA: ACM, 2016, pp. 118–121. [Online]. Available: http://doi.acm.org.ezp.lib.unimelb.edu.au/10.1145/2935663.2935683

[111] M. Azizian, S. Cherkaoui, and A. S. Hafd, “Vehicle software updates distribution with SDN and cloud computing,” IEEE Communications Magazine, vol. 55, no. 8, pp.74–79, 2017.

[112] Letaief, K. et al. (2019) “The Roadmap to 6G: AI Empowered Wireless Networks,” IEEE Communications Magazine, vol. 57, no.8, pp. 84-90, Aug 2019.

[113] Yang, P., Xiao, Y., Xiao, M., & Li, S. (2019). 6G Wireless Communications: Vision and Potential Techniques. *IEEE Network*, *33*(4), 70–75. https://doi.org/10.1109/MNET.2019.1800418