



UNIVERSITY OF WEST ATTICA

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Diploma Thesis

A Survey on Next Generation Networks and Cloud Computing



Student: JULIEN MAKALU

Registration Number: ele46587

Supervisor

CHARALAMPOS Z. PATRIKAKIS

Professor Dept. of Electrical and Electronics Engineering

ATHENS-EGALEO, February 2021



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

Μελέτη Δικτύων Νέας Γενιάς και Νεφοϋπολογιστικών Υποδομών



Φοιτητής: ΖΥΛΙΕΝ ΜΑΚΑΛΟΥ

ΑΜ: ele46587

Επιβλέπων Καθηγητής

ΧΑΡΑΛΑΜΠΟΣ ΠΑΤΡΙΚΑΚΗΣ

Καθηγητής στο Τμήμα Ηλεκτρολογών Και Ηλεκτρονικών Μηχανικών

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, Φεβρουάριος 2021

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Χαράλαμπος Πατρικάκης Καθηγητής	Τάτλας Νικόλαος-Αλέξανδρος Αναπληρωτής Καθηγητής	Φειδάκης Μιχαήλ ΕΔΙΠ
(Υπογραφή)	(Υπογραφή)	(Υπογραφή)

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Ζυλιέν Μακαλού, Φεβρουάριος, 2021

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **Ζυλιέν Μακαλού**, του Ζαν Πιερ, με αριθμό μητρώου **ele46587** φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

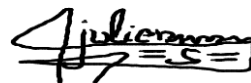
δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.»

Ημερομηνία 10 / 02 / 2021

Ζυλιέν Μακαλού



DEDICATION

I dedicate this Diploma Thesis to all those who have left their country to seek peace elsewhere, to orphan children, to all Greeks, especially the Greeks of Konitsa, to Congo's children and to all those who supported me, in particular:

to my biological Parents Jean Pierre Makalu and Marie Nzey

to my dear Yollande M. Lupi and our daughter Foteini-Clemencia Makalu

to my sister Nicole Lubembo Makalu

to my uncle Guy Mandungu Makalu

to my spiritual parents Georges Kitsaras and Mairry Tzora

to my protectors parents Mr. Panagiotis Maragos and Mrs. Loukia Makri

to my framers Dimitris Maragkos and Lina Karaiskou

to my dear friends, in particular Chris Gavalas, Theodora Mourechidi, Alexis and Denis Rembetsis

ACKNOWLEDGEMENTS

At the end of this degree thesis, my heartfelt thanks are dedicated to all those who contributed, directly or indirectly to my university study.

First, I would like to **thank God** for the miracle in my life, from the war and the park bench to the university desk. I express my gratitude to my tutor and supervisor Prof. Patrikakis Charalampos for his support and encouragement. His availability, his guidelines and his attention to detail have allowed me to constantly care for and improve the quality of this work. I would like to thank also PhD candidate Mr. Michael G. Xevgenis, acting as Assistant to the Supervisor for helping me with his experience and knowledge relatively to the Cloud Computing and Network, he has always found the time to follow up on my work and be attentive when I encounter difficulties.

I express my gratitude to Greece for the opportunity it has given me to continue my studies, to the Children's Care Center of Konitsa, to the Greek Orthodox Church and the theological boarding school of Agia Barbara, to the National Bank of Greece, to the Upstream mobile technology company, to the National Scholarships Foundation (IKY), to the Greek Council for Refugees and to all the Greek citizens who supported me during my school and university studies.

My thanks also go to Mr. Marco Veremis, for the scholarship and the opportunities offered within the Upstream mobile technology company for my graduation internship, to my sister Nicole Lubembo Makalu, to my uncle Guy Mandungu Makalu, to my dear Yollande M. Lupi, to my Greek language teacher Gianna Nikou, to my spiritual parents George Kitsaras and Mary Tzora, to Alexia Zoe, to Theodora Theodoroulaki, to the grandmother Kitsaras Sevasti and the grandfather Kitsaras Orestis for their support, to the Professor P. Skandalakis, to Mrs. Effie Bazdra, Mrs. Nelly Tzakou, Mr. Nikos Marinakos, to Mrs. Eleni Ninou, Mrs. Antonia Koliou, Mr. Xristos Tziompos, Mr. Stergios Thomas, Mr. Andrea Tefos, Niki Kona and all agents of KPM Konitsas, to the Maragkos family, Makri and Christodoulou family, Karaïskou family, Tsironi family, Mr. Dimitris Maragkos, Mrs. Lina Karaïskakou, Konstantinos Maragkos, Marilena Maragkou, Mrs. Xryssa Tsironi, Mr. Georgios Vassilogeorgis, Mrs. Aggeliki Mitsiou, Mrs. Lia Athanassopoulou, Mr. Nikos Spiropoulos, Mr. Ioannis Datsotoulos, Mrs. Pascale Durand, Mrs. Celine Anceline, Mr. Thomas Tzimas, Mrs. Ev. Kitsi, Mrs. Evi Gkarametsi, Mrs. Eleni Pangratiou, Ms. Antigoni Pavlidou, Kleio Nikolopoulou, Mrs. Ev. Liberi, Minas Vroxopoulos, Konst. Sgouros, Vasiliki and Vasilis Papatthemelis, Marianna Makri, Alex Zagkouroglou, Samoila Family and to all my student colleagues.

With immense pride I address my most distinguished thanks to all my professors of Technical School of Konitsa and of UNIWA who have passed their knowledge to us with high-standard training.

My thanks also go to all the members of the Examination Committee for the honor they have given us to participate in the review of this work.

To anyone who has the generosity to give their time and share a smile by communicating the strength to live, I extend my gratitude fully.

Περίληψη

Η δικτύωση δεδομένων IP έχει αναπτυχθεί με βάση τις αρχές των τηλεπικοινωνιών, τη στοίβα TCP / IP και την παλαιότερη δρομολόγηση μονάδων δεδομένων πληροφοριών. Οι ίδιες αρχές, σε

συνδυασμό με την τελευταία εξέλιξη της Επιστήμης και της Τεχνολογίας Υπολογιστών, χρησιμοποιούνται για τα δίκτυα της επόμενης γενιάς: δίκτυα καθορισμένα από λογισμικό και προγραμματιζόμενα, νεφουπολογιστική, εικονικοποίηση, όλα σε συνδυασμό με υψηλής ταχύτητας και αξιόπιστες γραμμές που επιτρέπουν την πρόσβαση σε μια παγκόσμια συνδεδεμένη κοινότητα . Αυτή η εργασία ερευνά τη σύγχρονη δικτύωση, δίνοντας έμφαση στο νεφουπολογιστική, την εικονικοποίηση, τα δίκτυα που καθορίζονται από λογισμικό (SDN) και το Εικονικοποίηση λειτουργίας δικτύου (NFV). Τα πλεονεκτήματα και τα μειονεκτήματα κάθε τεχνολογίας λαμβάνονται υπόψη, μαζί με τις αρχές και τις λειτουργίες τους. Στο τέλος αυτής της έρευνας, έχει ρυθμιστεί μια δοκιμαστική μονάδα SDN, με βάση το GNS3 με υποστήριξη Docker και τον ελεγκτή OpenDaylight. Διάφορες πλατφόρμες χρησιμοποιούνται σε εικονικό περιβάλλον για να καταλήξουν σε έναν λειτουργικό συνδυασμό. Ένα απλό σενάριο μηχανικής κίνησης SDN παρουσιάζεται επίσης. Το σενάριο του ελέγχου κυκλοφορίας των πλαισίων Ethernet σε ένα τοπικό δίκτυο με λεπτομερείς απαιτήσεις ασφαλείας, το οποίο προσεγγίζεται σε αυτή τη εργασία, είναι ένα σημαντικό ενδιαφέρον για τις εταιρείες δικτύου ή τις υπηρεσίες ασφαλείας που προσπαθούν να μεταναστεύσουν σε πιο ευέλικτα δίκτυα, ευέλικτα και προγραμματιζόμενα. Αυτό το σενάριο κατέστησε δυνατή τη διερεύνηση της απόδοσης και της διαχείρισης δικτύων επόμενης γενιάς στα δύο επιχειρησιακά επίπεδα του δικτύου, τα οποία είναι το σχέδιο δεδομένων και το σχέδιο ελέγχου. Η ενότητα συμπερασμάτων δίνει μια γενική άποψη αυτής εργασίας χρησιμοποιώντας αποτελέσματα ερευνών και δείχνοντας πώς το NFV και το SDN ανοίγουν το δρόμο στο δίκτυο 5G, το οποίο πρέπει να είναι έξυπνο, εικονικοποιημένο και προγραμματιζόμενο προκειμένου να βελτιωθεί η απόδοση των υπηρεσιών που προσφέρουν οι εταιρείες. Δείχνει επίσης συγκεκριμένες απαιτήσεις και ορισμένα πλεονεκτήματα, εμπόδια και μελλοντικές προκλήσεις ανάπτυξης και ανάπτυξης δικτύου στο περιβάλλον NFV / SDN.

Λέξεις-κλειδιά

Δικτύωση καθορισμένη από λογισμικό (SDN), Άνοιγμα ροής (OpenFlow), Νεφουπολογιστική, Εικονικοποίηση, GNS3, OpenDaylight

Abstract

IP Data Networking has been developed based on telecommunications principles, the TCP/IP stack, and the legacy routing/switching of information data units. The same principles, together with the latest evolution in Computer Science and Technology, are used for the next generation networking: software-defined and programmable networks, cloud, virtualization, all combined with high-speed

and reliable lines allow access to a globally connected community. This work surveys modern networking, emphasizing cloud, virtualization, software-defined networks and Network Function Virtualization (NFV). The advantages and drawbacks of each technology are considered, together with their principles and functionalities. At the end of this survey, an SDN testbed is setup, based on GNS3 with Docker support and the OpenDaylight controller. Various platforms are used in a virtualized environment to end up with a functional combination. A simple SDN traffic engineering scenario is also demonstrated. The scenario of the traffic control of Ethernet frames within a Local Area Network with granular security requirements, which is approached in this thesis, is a significant interest for the network companies or security services which are trying to migrate to more flexible networks, agile and programmable. This scenario made it possible to explore the performance and management of next-generation networks on the network's two operational planes, which are the data plan and the control plan. The conclusion section gives a general view of this thesis using results of researches and showing how NFV and SDN open the way to the 5G network, which must be intelligent, virtualized, and programmable in order to improve the performance of the services offered by companies. It explores also specific requirements such as the control of interfaces, the protection of operating systems, the reliable connectivity, the data network's security and flexibility, and some advantages, obstacles and future challenges of network development and deployment on the NFV / SDN environment are explored.

Keywords

Software-Defined Networking, OpenFlow, Virtualization, Cloud Computing, GNS3, OpenDaylight

List of Contents

List of tables	9
List of figures	10
Alphabetical Index	11

1	INTRODUCTION	13
1.1	Object of the Diploma Thesis	19
1.2	Purpose and objectives	19
1.3	Methodology.....	19
1.4	Innovation.....	20
1.5	Structure	20
1.6	Related Work	20
2	CHAPTER 1st: VIRTUALIZATION TECHNOLOGY AND CLOUD COMPUTING	21
2.1	Virtualization Technology	23
2.1.1	Virtualization Terminology and requirement	24
2.1.2	What is virtualization?	25
2.1.3	Hypervisors	26
2.1.4	Virtual Machines.....	30
2.1.5	Virtualization Techniques	31
2.1.6	Benefits of virtualization technology.....	34
2.2	Cloud Computing	35
2.2.1	Cloud computing architecture	35
2.2.2	Cloud computing service models.....	37
2.2.3	Cloud computing deployment models	38
2.2.4	Virtualization in the cloud	38
2.2.5	Open Source Cloud Computing Technologies	40
2.2.6	Benefits of cloud computing.....	41
3	CHAPTER 2nd: SOFTWARE DEFINED NETWORK AND NETWORK FUNCTION VIRTUALIZATION	43
3.1	Centralized Control and Data Planes	43
3.2	OpenFlow	46
3.3	SDN Controllers	49
3.4	Network Programmability	51
3.5	Network Function Virtualization (NFV)	52
3.5.1	The main considerations and key objectives of NFV	53
3.5.2	Relationship of SDN and NFV	54
3.5.3	Benefits of Network Function Virtualization (NFV)	54
3.6	Chapter Conclusion	55
4	CHAPTER 3rd: SDN VIRTUAL NETWORK IMPLEMENTATION USING GNS3	56
4.1	Implementation Strategy	56
4.2	The Host Machine	57
4.3	The GNS3 Environment	58
4.4	The OpenDaylight Environment	61
4.5	Basic SDN Topology	63
4.6	A Simple SDN Scenario	64
5	CONCLUSIONS	67

List of tables

Table 1: Advantages Of Virtualization.....	34
Table2: Benefits Of Cloud Computing.....	42
Table 3: The Components Of The Flow Table Entry.....	48

Table 4: The List Of Some Sdn Controllers.....	48
Table 5: Benefits Of Network Function Virtualization (Nfv).....	54

List of figures

Figure1: Three Layer Pyramid Of Cloud Computing Services [6].....	17
Figure 2: History And Evolution Of Virtualization [26].....	24
Figure 3: Virtualizing Resources [2].....	26
Figure 4: A Sample Virtual Machine Monitor [4].....	26
Figure 5: Type 1 And Type 2 Hypervisors [2].....	27

Figure 6: The Esx Architecture [4].....	28
Figure 7: The Xen Hypervisor Architecture [4].....	29
Figure 8: Microsoft Hyper-V Architecture [4].....	29
Figure 9: A Virtual Machine [4].....	30
Figure 10: A Simple Virtual Network [4].....	31
Figure 11: Full Virtualization Using Binary Translation [28].....	32
Figure 12: Paravirtualization [28].....	33
Figure 13: Hardware Assisted Virtualization [28].....	33
Figure 14: Cloud Computing: Server [3].....	36
Figure 15: Cloud Computing Service Models [29].....	37
Figure 16: Creation Of Logical Instances By Virtualization Layer[3].....	39
Figure 17: Control And Data Planes Of A Typical Network [8].....	44
Figure 18: Openflow Architecture [8].....	47
Figure 19: Packet Flow Through The Openflow Processing [32].....	48
Figure 20: Traditional Hardware-Based Network And Nfv Approach [34].....	53
Figure 21: Implementation Strategy In Terms Of Software Resources.....	56
Figure 22: The Host Machine And The Virtualization Software Installed.....	57
Figure 23: Addition On A Network Adapter To The GNS3 Vm.....	56
Figure 24: Download The Appliance From The GNS3 Marketplace.....	57
Figure 25 Illustrates The New Appliance Together With The Gns3 Vm, Running In Vmware Player 14.....	57
Figure 25: The Open Vswitch Appliance In GNS3.....	57
Figure 26: Import The Cisco Ios Image.....	58
Figure 27: Cisco Ios Router Integration.....	58
Figure 28: Opendaylight Machine Network Setup.....	59
Figure 29: Installation On Opendaylight In Xubuntu.....	62
Figure 30: The Opendaylight Dlux Interface.....	62
Figure 31: Basic Topology With Connections.....	63
Figure 32: Cisco Router Sample Configuration.....	63
Figure 33: Simple Traffic Engineering Scenario Topology.....	64
Figure 34: The Topology In The Opendaylight Dlux Interface.....	64
Figure 35: Uni-Directional Patch Panel Flow.....	65
Figure 36: The Result Of The Flow In The Traffic.....	66

Alphabetical Index

- AMD** : Advanced Micro Devices
API : Application Programming Interface

APP	: Atom Publishing Protocol
ARP	: Address Resolution Protocol
AWS	: Amazon Web Services
BSS	: Business Support System
CAPEX	: Capital Expenditure
CLI	: Command Line Interface
CMP	: Cloud Management Platform
CMS	: Content Management System
CORBA	: Common Object Response Broker Architecture
CP/CMS	: Control Program/Cambridge Monitor System
CPU	: Central Processing Unit
CRM	: Customer Relationship Management
CSPs	: Communications Service Providers
CTSS	: Compatible Time Sharing System
DCOM	: Distributed Common Object Model
EC2	: Amazon Elastic Compute Cloud
EMS	: Elements Management Elements
Eucalyptus	: Elastic Utility Computing Architecture to link your programs to useful systems
EVPN	: Ethernet Virtual Private Network
FIB	: Forwarding Information Base
GNS3	: Graphical Network Simulator-3
HTTP	: Hypertext Transfer Protocol
IaaS	: Infrastructure as a Service
IBM	: International Business Machines Corporation
IETF	: Internet Engineering Task Force
IP	: Internet Protocol
IT	: Information Technology
LISP	: Locator / ID Separation Protocol

LU	: Logical Unit
MAC	: Media Access Control address
MPLS	: Multiprotocol Label Switching
NAS	: Network-attached storage
NV	: Network Virtualization
NFV	: Network function virtualization
NFVI	: Network Functions Virtualization Infrastructure
NIC	: Network Interface Card
NIST	: US National Institute of Standards and Technology
ONF	: Open Networking Foundation
OPEX	: Operational Expenditure
OS	: Operating System
OSS	: Operation Support System
OVA	: Open Virtualization Appliance
OVF	: Open Virtualization Format
ONOS	: Open Network Operating System
PaaS	: Platform as a Service
RAID	: Redundant Array of Inexpensive Disks
RDP	: Remote Desktop Protocol
REST	: Representational State Transfer
RIB	: Routing Information Base
RPC	: Remote Procedure Call
SaaS	: Software as a Service
SAN	: Storage Area Networks
SDN	: Software Defined Networks
SOAP	: Simple Object Access Protocol
SNMP	: Simple Network Management Protocol
SSH	: Secure Shell

TSP	:Telecommunications Service Providers
TCP/IP	:Transmission Control Protocol/Internet Protocol
UI	: User Interface
VA	: Virtual Appliance
VLAN	: Virtual Local Area Network
VMM	: Virtual Machine Monitor
VMs	: Virtual Machines
VPN	: Virtual Private Network
vSwitch	:virtual Switch traffic
VTEP	: VxLAN Tunnel EndPoint
VxLAN	: Virtual eXtensible LAN
WSDL	: Web Service Description Language
XML	: Extensible Markup Language
XMPP	: Extensible Messaging and Presence Protocol

1 INTRODUCTION

The field of Information Technology is exploding around us, providing access to the resources of the Internet, clouds, and networks from which we have information when it is available. These technologies are expanding in many human life areas, including the workplace, education, business,

health services, and many others. Virtualization technology is the key to the foundation of these information technologies. As Rick F. Van Der Lans explains in his book "Data Virtualization Business Intelligence Systems," Virtualization in simple terms from the user's point of view means that virtual resources are used by different applications that do not need information about where these resources come from, the technical interface, the Platform used for their availability or how they were implemented [1]. It brings new application deployment models, new services, and other beneficial possibilities to organizations and enterprises. From a technical point of view, the definition of Virtualization, as described in Eric Bauer and Randaee Adams' book "Reliability and Availability of Cloud Computing", is "the logical abstraction of physical assets, such as the hardware platform, operating system (OS), storage devices, data stores or network interfaces" [2].

According to this definition, virtualization technology has a set of missions, starting with the fact that it enables the Virtualization of resources and facilitates their sharing. It makes it possible to operate, isolate and manage multiple virtual machines; it can also reassign them due to user mobility. In other words, Virtualization allows operating systems to run on a physical server that shares resources with others. The applications that use these resources are based on the operating systems. In virtualized mode, operating systems successfully run independently and autonomously.

Regarding its origin, as mentioned in Dr. Christopher Strachey's book "Time Sharing in Large Fast Computers", the concept of Virtualization was created in order to help engineers to use their Software while at the same time sharing resources of their computers. Its evolution led to the creation of Atlas Computer, whose main characteristics were the allocation of system resources to support multi-programming while separating physical memory from the used programs. The concept of Virtualization had provided the foundation for the advent of cloud computing, network virtualization, virtual memory, and Virtual Machine technologies to enable the consolidation of applications on a smaller number of servers. As a technology, Virtualization has several advantages, including reducing operating costs related to the hardware used, such as physical machines, floor space, and maintenance in the data center. A data center is defined as a physical space that hosts servers in a controlled environment with a clean power supply and optimized network connectivity [2].

The hypervisor or virtual machine monitor (VMM), one of Virtualization's elements, creates and operates virtual machines, allowing operating systems to run simultaneously on the same physical machine. There are two hypervisor categories: a native hypervisor that runs directly on a hardware platform and a type 2 hypervisor that runs inside another system. This is discussed in depth in the next chapter. Virtualization is seen as the foundation of cloud computing, where the virtualized Infrastructure is essential to ensure flexibility and scalability.

Cloud computing, as defined by the US National Institute of Standards and Technology (NIST), is a "model that provides ubiquitous, convenient, on-demand network access to a shared set of configurable computing resources that can be quickly provisioned and released with minimal management effort or interaction with service providers" [2]. Besides, Barrie Sosinsky, in his book "Cloud Computing Bible", states that cloud computing allows applications and services which run on a distributed network to use virtualized resources that they can access through common Internet protocols and network standards. It has the capacity of abstracting the physical layer of hardware, its resources are virtual, and it offers resources as needed [3].

Cloud computing has revolutionized the way applications are hosted and delivered. Before its emergence, the applications offered were necessarily located on a computer server hosted by the user. With its use, applications are located in an environment that consists of servers interconnected, through the process of Virtualization, and geographically distinct at the data center level. Virtualization makes it possible to create a virtual version of physical resources. For example, i) the Virtualization of servers with their installation on physical servers, ii) desktop virtualization allows the use of a central office unit stored virtually on a local server, iii) application virtualization, where applications are virtualized and run on most operating systems delivered to the end-user device, iv) storage virtualization by consolidating physical memory and v) network virtualization by enabling the user to access network resources from a single computer. Matthew Portnoy, in his book "Virtualization essentials", confirms that: "Virtualization is the engine that powers cloud computing and enables the transformation of data centers from a convenient and labor-intensive process into a set of easily consumable, self-managed, highly scalable and highly available resources "[4]. This means that virtualization technology makes it easier to pool, operate and manage IT resources. It helps cloud computing to provide greater automation possibilities that allow businesses to reduce administrative costs. Not only this, but it also increases the ability to deploy solutions dynamically.

The main characteristics of cloud computing are [5]:

- The on-Demand self-service is a function that provides users with IT capabilities automatically according to their needs.
- The Broad Network Access this function is a good answer to the problem of user mobility, as it gives them the possibility to access cloud services wherever they use a suitable IP network,
- The Resource pooling, as described in NIST-800-145, means that companies that provide the computing resources are pooled to satisfy the demands of different consumers in a multi-tenant model. These physical and virtual resources are sometimes dynamically allocated and reallocated.
- The Rapid Elasticity is a function that means that the cloud computing can quickly provide resources with elasticity and in any quantity for meeting user needs.
- The Measured service means that the cloud resources are subject to effective monitoring and control to ensure transparency for both the provider and the consumer.

Furthermore, cloud computing has two different classes based on the deployment model, such as public cloud, private cloud, community cloud, and hybrid cloud, and the service model, which describes the type of service offered, is a three-layer pyramid composed of service types such as Infrastructure as a Service (IaaS) at the bottom, Platform as a Service (PaaS) in the middle and Software as a Service (SaaS) at the top.

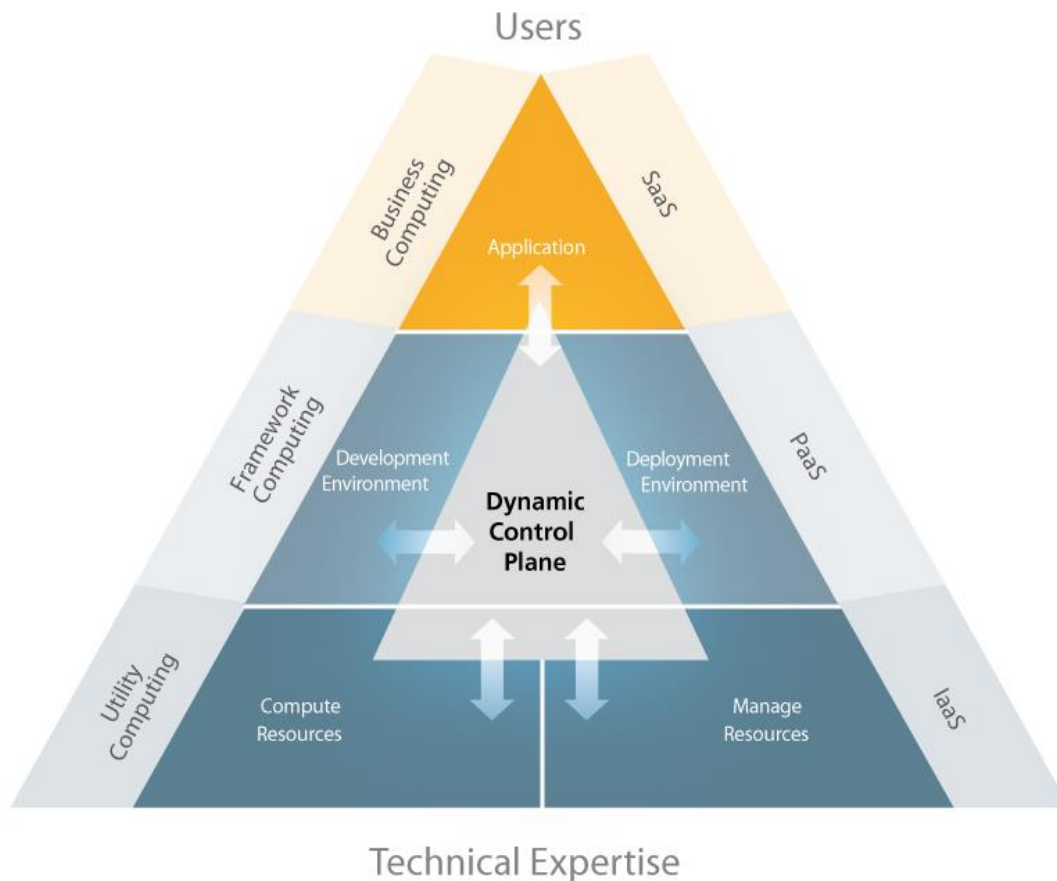


Figure 1: Three layer pyramid of cloud computing services [6]

Beyond the strategic capital expenditure through which cloud computing helps companies move IT from a capital-intensive to a revenue-generating activity, it also offers many benefits such as flexibility, efficiency, rapid implementation, and improved energy efficiency. One of the significant advantages of virtualization technology and cloud computing is the ability to automatically dispose of hardware resources, such as the server, storage space, or network, according to users' needs. In the area of networks, it is essential to bring the network's physical processes, such as routers and switches, to the Software. This migrating functions from the physical network to the Software is called network virtualization (NV). The Journal of Lightwave Technology, in the article "Network Virtualization: Technologies, Perspectives, and Frontiers", describes network virtualization as any form of partitioning or combining a set of network resources that are then made available to users in such a way that each user has a unique and distinct view of the network [7]. Traditional network resources, such as links and nodes, or topologies, are abstracted and migrated from hardware to Software or are put together in an appropriate environment, and they are easily accessible to users through a centralized management system. NV can mix several physical networks to create a software-based virtual network or divide a physical network into several separate virtual networks that operate independently of each other, increasing network efficiency. A software layer is superimposed on the hardware to drive it using tunneling technologies to interconnect the virtual networks in network virtualization. This approach meets the elasticity requirements of cloud computing, as the Software theoretically makes the network capacity unlimited and automatically adjustable according to the user's needs. It thus allows network capacities to be available alongside server and storage capacities.

As a whole, a network device is made up of two planes: a data plane, which has the task of connecting the different parts of the network to a device, and a control plane, which can be considered as the brain of a device. The control plane implements the protocol and communicates with the data plane to coordinate the network path's construction. A logical control plane sends commands to each device to manage its routing hardware and physical switching in the centralized control plane system. All this led to the creation of Software Defined Networks (SDN), which have been commercially supported by the Open Network Foundation (ONF), which is also its central standards authority [8]. Stanford University is considered the place where SDN development began, with the creation of Open Flow, the control plan housed in a centralized controller. Thomas D. Nadeau and Ken Gray describe SDN in their book *SDN: Software Defined Networks "An Authoritative Review of Network Programmability Technologies"* as an approach to architecture that enhances and simplifies network operations by closely linking communication between applications, services, and devices in a network using a logically centralized network control point [8]. This control is achieved from the coordination and orchestration layer between virtual and physical devices using an SDN controller.

Network function virtualization (NFV) is a concept that is considered to be complementary to SDN; it separates Software from hardware allowing flexible network deployment and dynamic operation. The SDN has the task of consistently controlling devices such as routers and switches through the use of SDN controllers, and NFV allows network applications to be controlled with virtualized servers; it also represents the implementation of data networking functions in Software that runs on necessary hosts. According to the European Telecommunications Standards Institute in its article "Network Functions Virtualization: An Introduction, Benefits, Enablers, Challenges & Call for Action", NFV aims at transforming network architecture by enabling the evolution of standard computer virtualization technology to consolidate many types of network equipment on servers [9].

It is known that Virtualization, cloud computing, and the software-defined network offers specific advantages to businesses. SDN provides centralized control of the network and facilitates its management. It provides a single set of APIs, allowing a single management console, from a central controller, for both physical and virtual devices. With its use, the virtual network configuration is done without impacting the physical network, thus reducing overall operating costs, offering the possibility to save hardware and reduce capital expenditure. Also, it allows unifying the cloud resources, directing and automating data traffic, which results in high quality of service. Not only that, but the specific advantage of SDN is the programmability of traffic that promotes agility and implementation of network automation [10].

This thesis addresses the subject of cloud computing and network virtualization, starting from virtualization technology, cloud infrastructure performance, and analyzing Software Defined Network (SDN) and Network Function Virtualization (NFV). Given the advantages of network virtualization, the idea of creating an SDN solution using Graphical Network Simulator 3 (GNS3) was born. The use case stems from the need to program the network and make it simple, intelligent, and transparent. To realize this idea, the Docker container, the OpenvSwitch, and Virtualization will be used in GNS3 to create a programmable and flexible network. Many network engineers adopt the GNS3 eco-system to emulate, test, troubleshoot, and configure virtual and real networks.

As described in the "Docker-curriculum" web site, Ubuntu is a Debian-based Linux operating system that runs from the desktop to the cloud, and Docker is a service that makes it easy to run an application in a container, packaging it with all its dependencies in the same place [11]. The OpenvSwitch is a

multi-layer software switch, an OpenFlow virtual switch, which provides a switch stack for hardware virtualization environment [12]. The VMware workstation can be described as a hosted hypervisor that allows virtual machines to be configured and used on a single physical machine [13]. In general, some of the significant issues that companies face are setting up their network correctly, controlling, expanding or shrinking their network effectively, and making it easier to manage. Besides, according to Zeus Kerravala in his article "The Top Five Network Problems Solved by SDN", today's enterprises are faced with many network problems such as: solving network problems and their proper use, securing resources, and supporting critical IT projects [14]. The use case will be proposed as a network solution for technology companies in Greece to solve different network problems they face and thus enable them to migrate to the next-generation network.

1.1 Object of the Diploma Thesis

This thesis is a survey on Next-Generation Networks using cloud computing. In particular, this thesis aims to analyze how cloud computing can contribute to virtualize a network, making it programmable and flexible to the need of users. Additionally, the thesis provides a study of the Software-Defined Network (SDN) technologies and evaluates how it can be used in the context of network virtualization.

1.2 Purpose and objectives

Cloud computing's overall performance allows concentrating the IT infrastructure from a company to the public or private cloud. Cloud computing's network resource refers to the provisioning of network connectivity and capacity to interconnect other resources. It is possible to easily describe and create networks in a cloud environment to interconnect resources dynamically and automatically. The network virtualization using cloud computing is possible with the use of OpenFlow architecture, which creates an open interface that allows the external controller to modify how the switch or the router can work in order to respond to the need of users. Our motivation is based on the performance and benefits that this technology offers. This leads us to analyze, explore and perform network virtualization experiments using cloud computing via the GNS3 environment. This Diploma Thesis aims to answer the research problem statement and question: *how can cloud computing contribute to virtualize a network?* This work will help readers who want to study, examine, and possibly migrate from a traditional to a virtual network to understand network virtualization using cloud computing and appreciate its benefits.

1.3 Methodology

The methodology chosen for this thesis is the theoretical and experimental research process to obtain results from a significant traffic engineering scenario. The research process consisted of the following steps: initially, the emphasis placed by telecommunications companies on the next generation networks led the author to begin reviewing the literature in order to find specific information for this study by doing the synthesis of the literature from conference papers, books, theses, web publications,

dissertations and journal articles. This review of the literature made it possible to answer the problem statement and question.

To fully present this survey on next-generation networks, the research carried out is completed by a practical implementation of a typical case, implementing an SDN virtual network using GNS3, which simulates a cloud-based network in real-time in order to understand its operation fully. The experimental research process used these instruments based on SDN virtual network whose results were analyzed and commented on.

1.4 Innovation

The experimental research process used to fully present this survey on next-generation networks has led to the development of a patch-panel scenario with the traffic control of Ethernet frames within a Local Area Network with granular security requirements, where a switch port can communicate with a specific port but not with other ports on the same switch.

1.5 Structure

The rest of this thesis is organized as follows: in Chapter 2, a theoretical analysis of Virtualization technology and Cloud Computing will occur. In Chapter 3, an analysis of Software Defined Network and Network Function Virtualization will take place. In Chapter 4, the implementation of an SDN virtual network using GNS3, together with a traffic engineering scenario, will be undertaken and a conclusion in Section 5.

1.6 Related Work

The evolution of technology in network virtualization has inspired the academic community, industries, and many researchers to suggest models and conduct projects that can explore the different aspects of network virtualization. Several researchers are motivated to continue their research in this area because network virtualization offers flexibility, security, and manageability. According to T. Anderson, in his book "Overcoming the Internet Impasse through Virtualization", network virtualization technology is considered a tool for evaluating new architectures and a fundamental element of the next generation of networks [15]. This section reviews recent work and projects related to network virtualization. From a historical perspective, it is known that the virtual local area network (VLAN), which is a group of hosts in a logical network with a single broadcast domain, the Virtual Private Network (VPN), active, programmable, and overlay networks are the four main classes of coexisting multiple logical networks which have existed in the past. The VLAN offers good isolation, and for its operation, switches use the destination MAC address and the VLAN ID to transfer frames. The VPN connects several sites using private tunnels that have an adequate security level on a shared communication network such as the Internet. Active and programmable networks allow the consistent execution, by different actors, of potentially conflicting code on network equipment. The overlay network is a logical network that is deployed over one or more physical networks that exist. [16].

Network virtualization is based on the separation between the control plane and the data plane, applying SDN technologies. These two planes are connected via a well-defined programming interface between the switches and the SDN controller, which is responsible for controlling the logic routing operation. The network virtualization technique, which uses the SDN controller to control, program, and manage the network, attract researcher's and enterprise's attention because it provides essential services for deployed applications, such as routing, access control, and multicasting. Here are some SDN and NFV controller projects:

- The NOX Controller, which has been developed by Nicira Networks, is one of the first generation OpenFlow controllers. It uses the C++ or Python languages to develop network control applications in production and education [17].
- The POX Controller is a python implementation of the NOX controller invented to be a platform that implements a network virtualization programming model while developing its control software. It supports the OpenFlow protocol [17].
- The Ryu Framework is an SDN framework that provides network services based on the use of well-defined API and python components to facilitate the development of various network management and control applications. Its integration with OpenStack, a cloud orchestration system, allows it to be deployed in cloud data centers [17].
- The Projector Controller is an OpenFlow controller that uses the Java language. Integrating with the Open Stack orchestration system is considered the fundamental element of the commercial SDN products of the Big Switch network [17].
- The Open Daylight Controller is an SDN controller that uses the Java language. It was developed by the Linux Foundation to provide a complete network programming platform for SDNs by supporting protocols such as OpenFlow, OVSDDB, and NETCONF. Besides, it can integrate the Open Stack orchestration system [17].
- The CORD is an abbreviation for Central office Re-architected as a Datacenter, an open-source project that aims to bring together SDN, NFV, and cloud computing services to offer "all-in-one" services (XaaS). It is applied to residential, mobile, and enterprise networks and can integrate open source platforms such as ONOS SDN Controller, XOS, and OpenStack. [18].
- Cloudify is an open-source cloud and NFV project initiated by Giga Space to increase the orchestration and management of NFV. Its use of Topology and Orchestration Specification for Cloud Applications (TOSCA) allows it to fully manage the service lifecycle by being compatible with virtualized and non-virtualized devices [19].
- NAPO is a project initiated by the Linux Foundation and adopted by global service providers such as Ericsson, Nokia, Cisco, and Huawei to design, build VMFs and orchestrate end-to-end compound services [20].
- The OSM is an open-source VMF management and orchestration platform hosted by the European Telecommunications Standards Institute (ETSI) and adopted by major European network operators. It has a resource orchestrator that handles orchestration in SDN and cloud technology for network service lifecycle management [21].
- The X-MANO is a cross-domain NFV orchestration platform that consists of different interfaces and modules to ensure confidentiality of information and programmability of the cross-domain service lifecycle [22].

- The OpenBaton is a project initiated by the Fraunhofer Fokuz Institute that runs on different NFVIs such as AWS, OpenStack, Docker containers, and LXC containers open-source implementation of the OVN based on the ETSI NFV MANO reference standard and the OASIS TOSCA specification [23].
- Gohan, used as an orchestration layer for network services over cloud services, is based on micro-services to simplify the deployment model. It consists of a REST API server, a database backend, a command-line interface, and a web user interface [24].

Today, the concept of network virtualization using a cloud infrastructure has an important place in information technology. It is a process in which the cloud user can access the virtual server and virtual hardware to create a virtual network solution that is flexible and programmable with the help of a hypervisor. All these projects and work around network virtualization technology clearly show a strong need for Virtualization in the network domain. Network virtualization using cloud computing is a modern solution that reduces the number and cost of physical devices needed, easily segments the network, allows rapid changeover and agile deployment, saves CAPEX and OPEX, and reduces downtime.

2 CHAPTER 1st: VIRTUALIZATION TECHNOLOGY AND CLOUD COMPUTING

Two of the most efficient technologies, Virtualization, and cloud computing, have transformed information technology (IT). This chapter examines what virtualization and cloud computing are, describing their benefits. The first part of this chapter will approach Virtualization by presenting its description, history, terminology, and requirements. This part will also look at what exactly Virtualization is, how it works, its components such as hypervisors, virtual machines, its techniques, and benefits. The second part of this chapter will present a detailed analysis of cloud computing based on its deployment models, key features, and benefits. It will also approach the topic of Virtualization in the cloud.

2.1 Virtualization Technology

As a software technique, Virtualization enables the creation of virtual machines (VMs) inside a computer. Initially, the virtual memory and time-sharing system allowed the development of what is now called virtualization technology. Today the need for Virtualization is becoming increasingly apparent due to ever-expanding modern technology. Its history, which is as old as computing itself, informs that it was built on the concept of paging and virtual memory. These two concepts had allowed more extensive programs than the available memory by sharing the same memory with other programs. The Atlas computer, developed by the University of Manchester, was the first notable system to include virtual memory [25]. However, due to various problems related to the virtual memory system's malfunction, the MAC project was launched to develop the next generation of time-sharing systems [25]. A similar project, the Compatible Time-Sharing System (CTSS), has been set up at the Massachusetts Institute of Technology to allow users to interact directly with the computer [25]. This notion of interaction had attracted the attention of IBM (International Business Machines Corporation), which subsequently initiated the development of its time-sharing system and later the CP/CMS (Control Program/Cambridge Monitor System), which is a high-performance time-sharing operating system considered as a virtual machine [25]. The first classical Virtualization was performed on IBM mainframes in the 1960s, and Gerald J. Popek and Rober P. Goldberg codified the framework that describes the computer system requirements to support virtualization [26]. Today Virtualization has emerged as a technology that enables better use of available resources by working with cloud computing and other modern technologies.

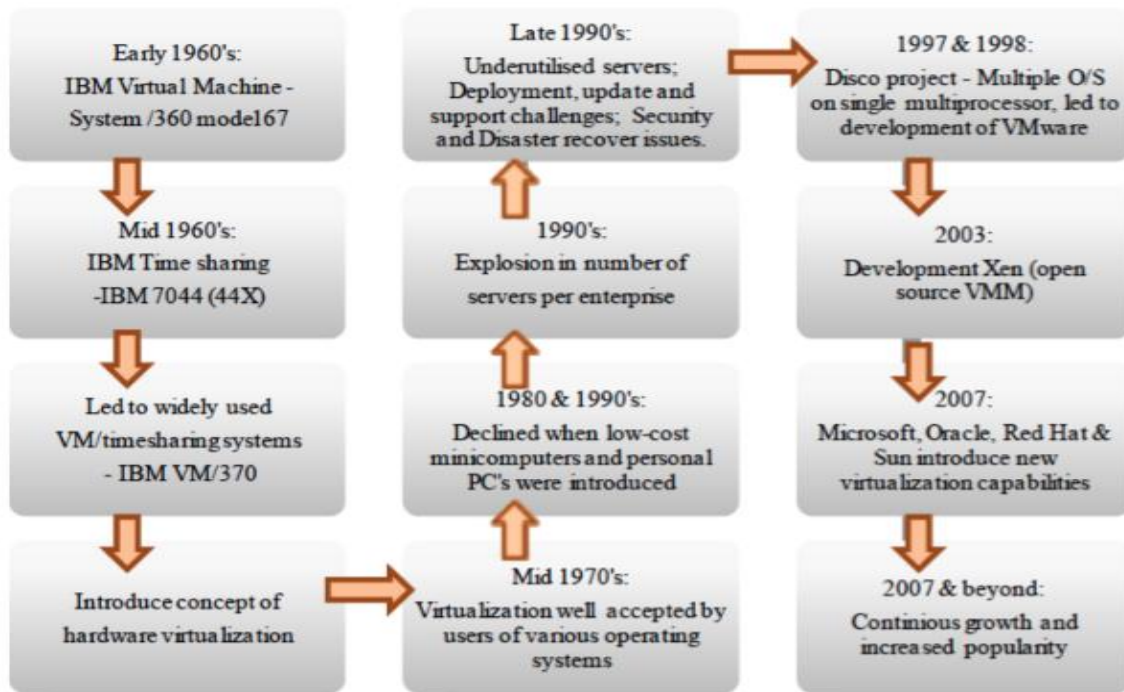


Figure 2: History and evolution of virtualization [26]

Virtualization technology allows the server to host different processes and operating systems (OS) at the same time without negatively impacting their performance. Also, it allows virtual machines to be used from anywhere, the development of virtual test benches, and facilitates maintenance. Hypervisors are one of the elements that have enabled Virtualization to make possible simultaneous access to computers that perform batch processing. By enabling efficient sharing of resources, virtualization technology has also enabled the development of the current model of cloud computing. [2]. According to Aaron Weiss, in his book "Computing in the clouds," the emergence of cloud computing has been fostered by the existence of virtualization technology, the need to use one server to meet the demands of different users appropriately the consolidation of existing servers [27].

2.1.1 Virtualization Terminology and requirement

In this section, we describe some standard terms of Virtualization that are used in this chapter, and we discuss some requirements for its proper functioning:

- A **Virtual Machine** (VM) is a software implementation of a physical machine with its operating system and its applications independent of the physical machine that hosts it.
- A **host machine** is the physical machine that runs the hypervisor by providing the actual hardware resources such as processor, network connectivity, and memory that are used to run virtual machines.
- A **Hypervisor (Virtual Machine Monitor (VMM))** is a software layer that makes Virtualization possible by summarizing the physical layer and then presenting the abstraction for virtual machines that individuals use as a subset of physical resources. Besides, it manages the inputs and outputs from the virtual machines to the physical device and vice versa. As an

essential feature for virtual environments, the hypervisor allows virtual machines to start applications and run on a single host to better use its hardware resources.

- A **Guest operating system** is an operating system installed and executed in a virtual machine.

The hypervisor needs to exhibit three properties, according to Popek and Goldberg [4]:

- *Fidelity* means that the environment created by the hypervisor for the VM must be similar to the physical hardware machine,
- *Isolation or security* means that the hypervisor must have complete control over system resources,
- *Performance* means that the VM should generally operate as a physical machine operates.

2.1.2 What is virtualization?

Virtualization is at the heart of modern information technology, where it changes the way IT services are delivered. It is creating a virtual version of something that can be a server, an operating system (OS), a storage device, or a network, using Software that simulates the hardware's functionality. It describes technology that extracts computing resources, such as processors (CPUs), storage, memory, and network connectivity, from existing underlying hardware or Software. This makes it possible for applications to run simultaneously on a single hardware platform. In order to virtualize hardware resources, Virtualization uses a virtual machine (VM), which, according to the Open Virtualization Format (OVF), is an "encapsulation of virtual hardware, virtual disks and associated metadata" [2].

As mentioned above, one of the critical elements of Virtualization is the virtual machine monitor (VMM or hypervisor), Software that enables the operation of VMs by supporting the running of multiple operating systems on a single host computer to increase the robustness and stability of the system. For example, if one OS crashed, it can not affect others, and applications can run without interruption. Besides, the hypervisor manages guest operating systems and their use of system resources. The virtualization process works as follows: the hypervisor separates the physical resources from their physical environments, allowing them to be used in the virtual environment where the applications run. If a user or program in the virtual environment needs additional resources, it can send an instruction to the hypervisor's physical system, which relays the message and stores the changes. [4].

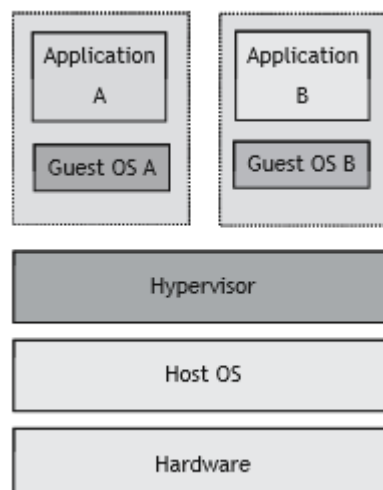


Figure 3: Virtualizing Resources [2]

As shown in **Figure 3**, VMs are isolated instances of the application software and the guest operating system that function as a separate computer. The hypervisor is responsible for supporting the isolation of VMs, just as it manages their operation on the same host computer. To facilitate application deployment, a virtual appliance is used. It is a software image delivered as a complete software stack installed on one or more VMs managed as a unit. It is usually delivered in Open Virtualization Format (OVF) [2]. Virtualization enables availability through virtual servers, which are encapsulated systems, and has disaster recovery capabilities. With the evolution of web-based computing, availability has become critical to maintaining 24/7 operations with enhanced Software and functionality. For example, Linux and newer versions of Microsoft Windows can solve resource scarcity issues without affecting application availability [4].

2.1.3 Hypervisors

The hypervisor, called initially Virtual Machine Monitor (VMM), manages the interconnection between the virtual machines and the hardware. It improves processor and memory utilization by running VMs which use their own operating systems. It is also used for developing and debugging operating systems. According to Matthew Portnoy, the VMM is now called a hypervisor because engineers were initially trying to solve a resource allocation problem, and their research led to the discovery of code that was called hypervisor by the logic that operating systems at that time were called supervisors and this code could well replace them [4].

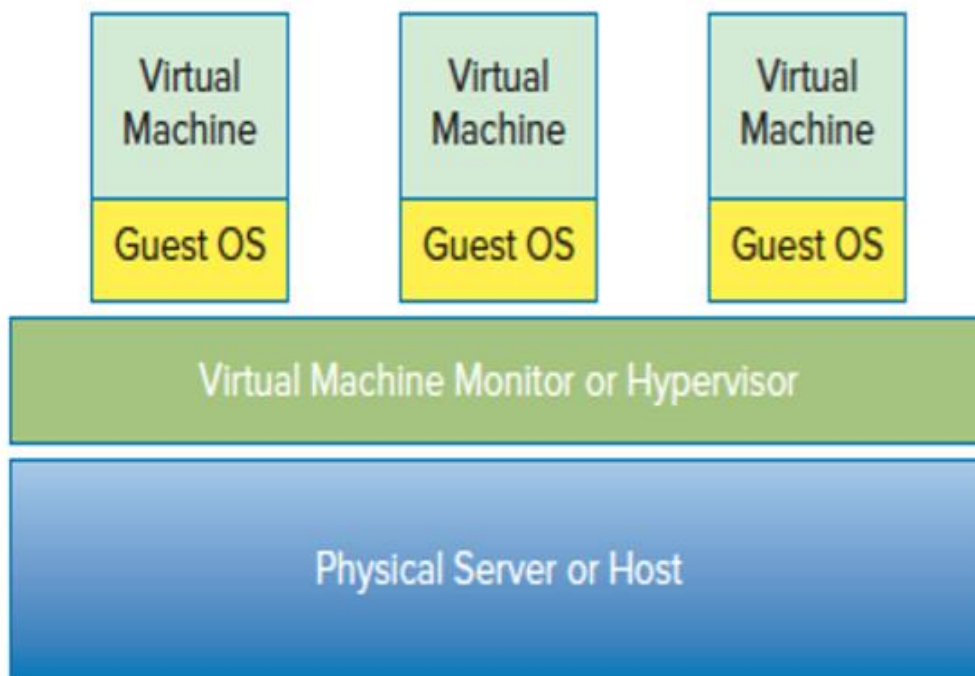


Figure 4: A sample virtual machine monitor [4]

As mentioned before, requests and read/write operations to disk, operation of network I/O or any other interaction from the virtual machine pass through the hypervisor first.

Hypervisors are divided into two types:

Type 1 (bare metal) hypervisor is installed on the host machine as the operating system. It runs directly on the host computer and provides a control for hardware access. There is no intermediate layer between the hypervisor and the physical hardware, which is why its implementation is called "bare metal." Its location below the virtual machines allows it to manage guest operating systems correctly. Among its features, the main one is performance; the fact that it communicates directly with the physical hardware resources makes it efficient and better than the type 2 hypervisor. It is specifically designed to support Virtualization by providing a large number of physical hardware resources from the host machine to guest virtual machines. Even in terms of consolidation, Type 1 hypervisor ratios are higher than those of Type 2 hypervisors. Some examples of this type of hypervisor are VMware, vSphere, ESXi, Microsoft HyperV, Citrix XenServer, Red Hat, Enterprise Virtualization (RHEV), and KVM. [4]

Type 2 (hosted) hypervisor is installed on a traditional operating system and takes advantage of the pre-existing operating system installed on the hardware to manage resources. It is installed as an application, making the installation process quick and easy. It is compatible with physical hardware devices in that it uses hardware that is already known by the underlying operation of systems such as Windows and Linux. Unlike the Type 1 hypervisor, it cannot access the hardware directly and must, in turn, go through an additional cycle to feed the request to the operating system, which manages the I/O. The request results follow the same return path through the hypervisor, which transmits the result to the virtual machine. This requires time compared to a type 1 hypervisor. Another disadvantage is that, since it runs on the operating system, any problem that causes the operating system to fail, such as malware or a destructive device driver, will also hurt virtual machines running on top of the operating system. Some examples of Type 2 hypervisor are VMware Workstation, Microsoft Virtual PC, Parallels Workstation, and Oracle VirtualBox. [4]

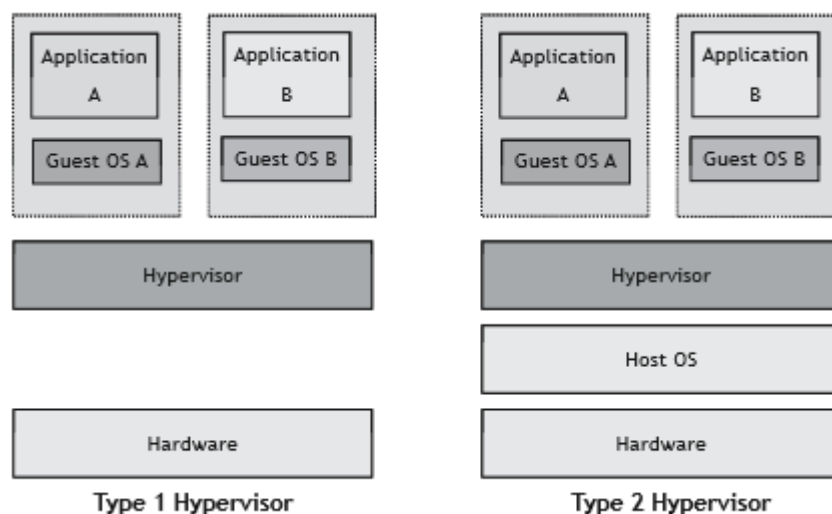


Figure 5: Type 1 and Type 2 Hypervisors [2]

Two options for the operation and role of the hypervisor are analyzed [4]:

- **Holodecks and traffic cops:** it is strongly recommended that the guest operating system access physical resources, or at least believe that it can access these resources to share them. This is the hypervisor's central role in making the guest believe that he can see and access physical resources. The second role is that the hypervisor, apart from abstracting the hardware, must balance the workload by acting as an intermediary with the guests on one side and the physical devices on the other in order to meet the guest's requests on the different subsystems. Given this function, it is clear that the hypervisor functions as a traffic agent controlling the flow.
- **Resource Allocation:** Acting as an intermediary between guests and physical devices, the hypervisor manages storage I/O requests from guests, network I/O, memory processing, and CPU work using its resource planning process to ensure that they are fully processed.

The scale of Virtualization has led to the development of several solutions using different hypervisors such as VMware ESX, Xen, Microsoft Hyper-V, and VirtualBox.

ESXi's virtualization layer enables the extraction of physical host resources in multiple virtual machines by allowing applications to access them without direct contact with the underlying hardware [37].

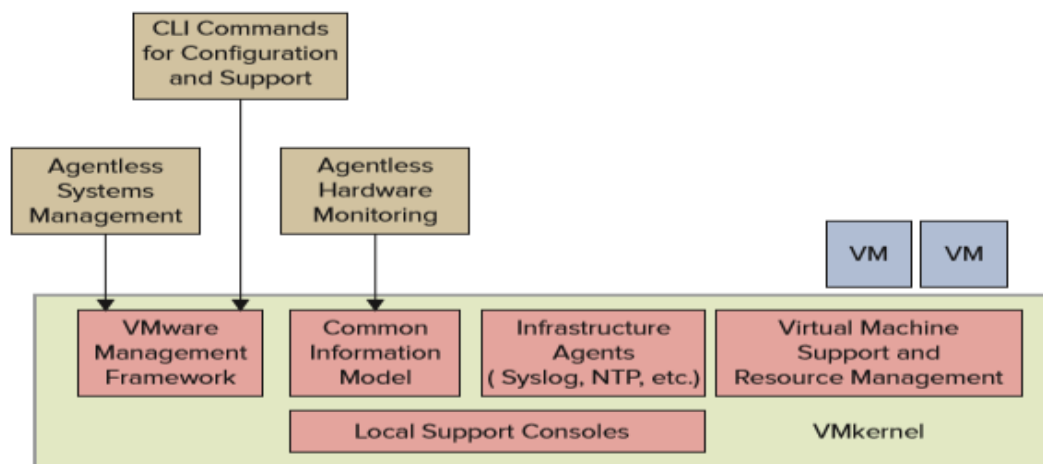


Figure 6: The ESX architecture [4]

Xen hypervisor allows the parallel execution of a large number of operating systems on a single machine. It is an open-source type 1 hypervisor[38].

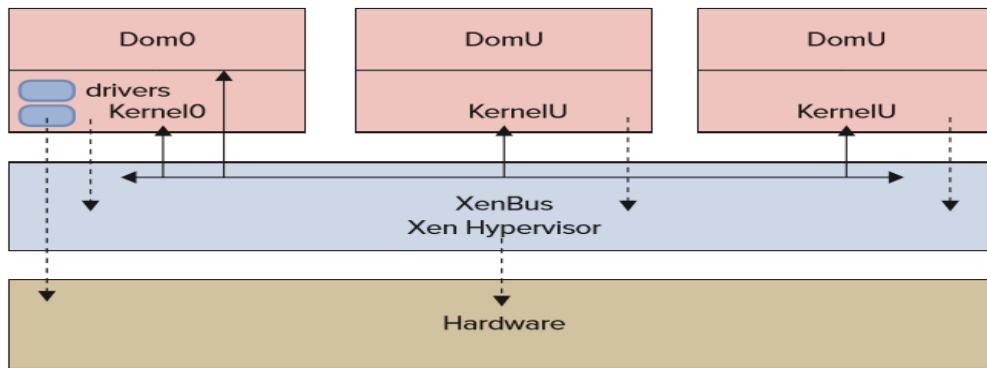


Figure 7: The Xen hypervisor architecture [4]

The Xen model, as shown in Figure 7, has a special guest called Domain0; its reference is Dom0. This guest can run when the hypervisor starts. It differs from other guests in terms of management privileges, so when other guests request the underlying hardware resources, their request goes first through the hypervisor to Dom0 and then to the resources. The response from the resources follows the same path back to the guest.

Hyper-V offers hardware virtualization by enabling the creation of virtual devices such as hard disks and virtual switches. It supports a large number of operating systems like Linux, FreeBSD, and Windows [39].FreeBSD, and Windows [39].

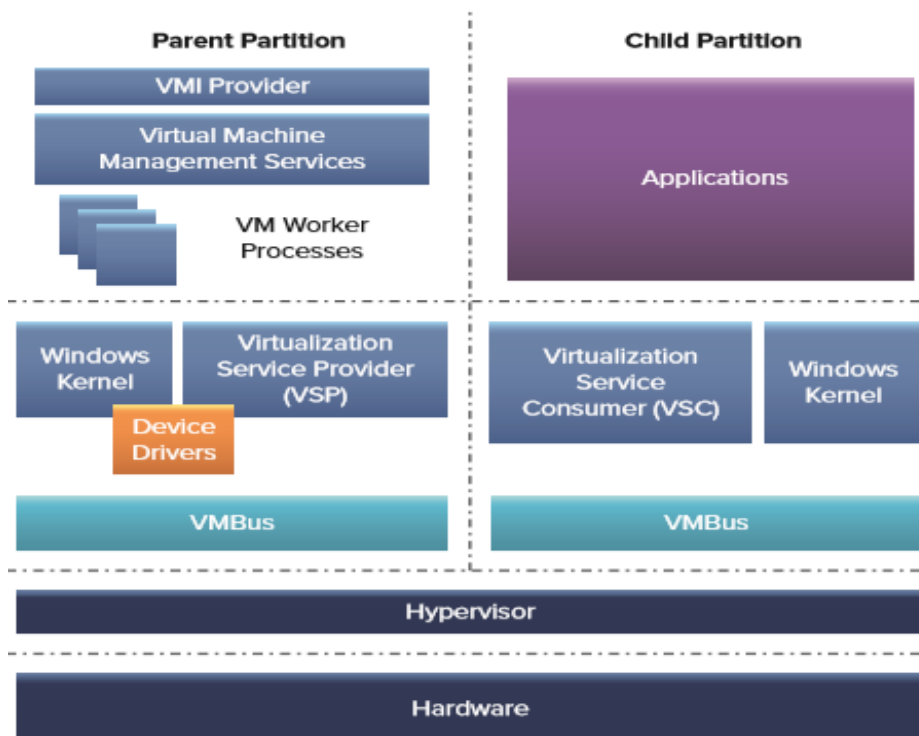


Figure 8: Microsoft Hyper-V architecture [4]

Some other virtualization solutions, such as VirtualBox, developed in 2010 by oracle, and Kernel-based virtual machine (KVM), which Red Hat acquired, use open source Xen code.

2.1.4 Virtual Machines

Virtual machines (VMs) have been created to perform specific tasks that require much attention, and that cannot be performed safely in a host environment. These tasks can be like testing operating systems. These are software computers that run applications and operating systems with the same functionality as physical computers. According to Portnoy M. in his book "Virtualization Essentials," virtual machines are essential elements of virtualization technology; they have operating systems that support traditional applications [4]. A hypervisor-based supports Their operation on a physical server. As explained in the previous section, the hypervisor decouples traditional operating systems from the hardware, transporting and regulating resource traffic to and from the VMs it supports. One of the points of difference between VMs and the physical server that supports a single operating system is based on the fact that VMs can be deployed and run multiple operating systems and applications at the same time within a single physical server. VMs provide rapid deployment and configuration capabilities; they are configured to work with processors and CPUs like a traditional computer.

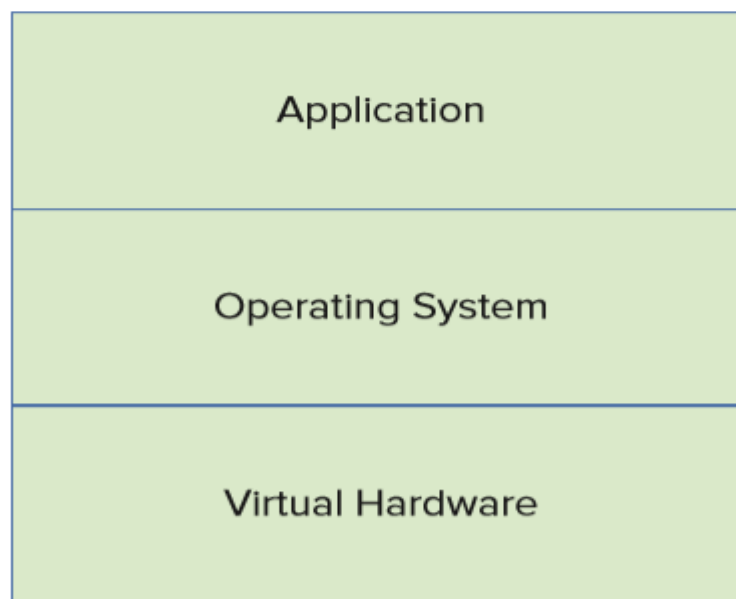


Figure 9: A virtual machine [4]

The performance of the applications that they support depends on their sufficient memory resources. If they need more memory, it is possible to reconfigure the memory amount by adding the memory capacity.

VMs communicate with the physical world via the virtual network created by the hypervisor. This network has one or more network interface cards connected to a network of virtual switches. It creates a secure environment for virtual machines that share a host.

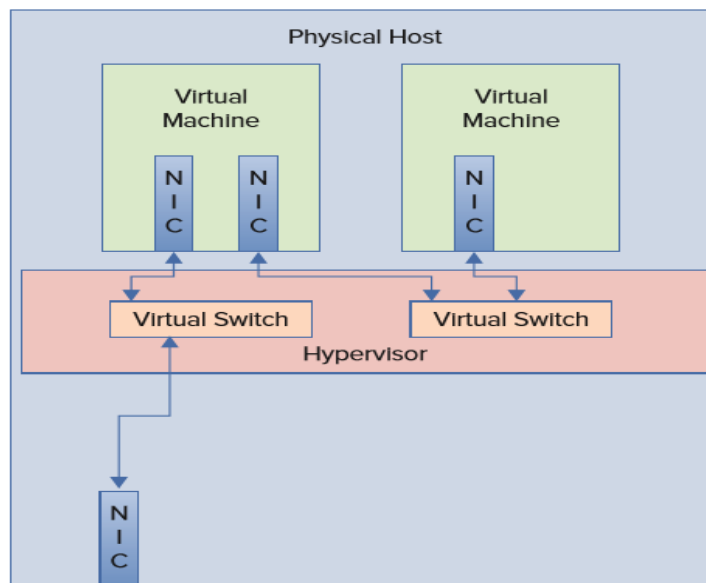


Figure 10: A simple virtual network [4]

The network connection allows interacting with the VM to manage applications that are supported by the server. One VM functionality option is cloning the existing server to create a new one using the template, a preconfigured VM. Another VM working option is a snapshot, a capture of the VM's state at a given time, which preserves its state, data, and hardware configuration by accumulating all changes in a child disk to another snapshot. The Open Virtualization Format (OVF), created to make it easier to transport virtual machines by bundling them into one or more files, is one method of packaging and distributing VMs. To do this, the Open Virtualization Appliance (OVA) is also used to consolidate data into a single file.

2.1.5 Virtualization Techniques

This section presents virtualization techniques such as "Full virtualization using binary translation," "OS-assisted virtualization or paravirtualization," "Hardware-assisted virtualization and operating system-level virtualization.

-Full Virtualization

In this technique, hypervisors are used to provide Virtualization without modifying the guest operating system. The hypervisor intercepts non-virtualizable instructions before execution and dynamically translates them into code by analyzing the VM memory. Full Virtualization provides high performance; the VM does not require any changes to the guest operating system or its applications and offers good operating system compatibility. It is a technique that meets specified requirements and allows the majority of instructions to be executed directly from the virtual processor to the physical processor. The hypervisor controls hardware resources by providing virtual machines with the ability to provide a guest operating system with an environment similar to physical hardware. [28].

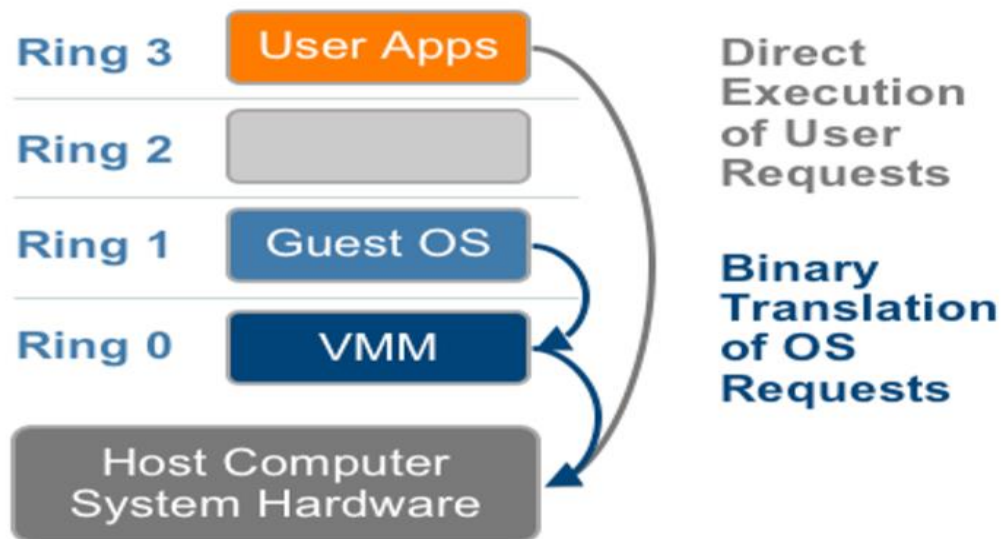


Figure 11: Full Virtualization Using Binary Translation [28]

One of the characteristics of full Virtualization is the total ignorance of the guest operating system. Because of this ignorance, the guest operating system does not communicate with and the hypervisor. Besides, no support for the underlying hardware and no changes to the guest operating system is required. This allows the hypervisor to have high performance for managing operating systems and binary translation of non-virtualizable instructions. [28].

-Paravirtualization

Paravirtualization, initially called paravirt-ops, was developed by the Xen Group to allow high performance and robust isolation of resources, with slight modifications to the guest operating system. It is a virtualization technique with an interface and uses communication between the hypervisor and the guest operating system. This technique also modifies the kernel code of the guest operating system by replacing non-virtualizable instructions by hyper-calls that communicate directly with the hypervisor. A hypercall is like a Linux system call, which admits control to the hypervisor, which uses the instruction and emulates the result to the guest operating system, constantly monitoring it for failed instructions. Paravirtualization provides a software interface or API that resembles physical hardware and allows the guest operating system to invoke hardware. One of the features of paravirtualization is the need to modify the guest operating system.

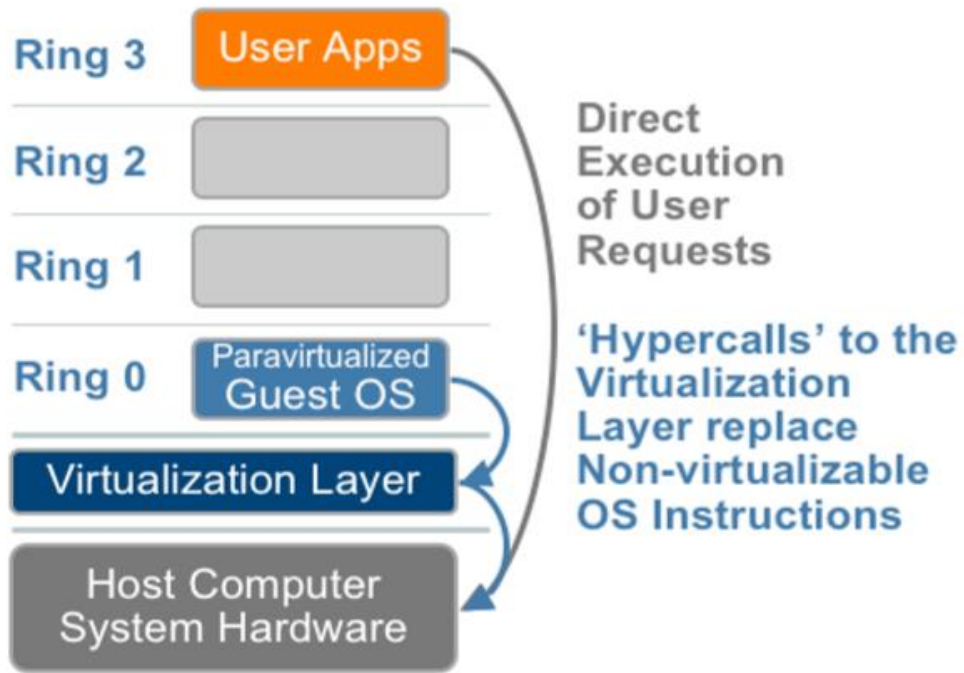


Figure 12: Paravirtualization [28]

-Hardware Assisted Virtualization

Provided by Intel and AMD, hardware-assisted Virtualization uses a virtualization-aware processor that can interact directly with the hypervisor to provide optimizations. The hypervisor enables the isolation and virtual machine's control.[2].

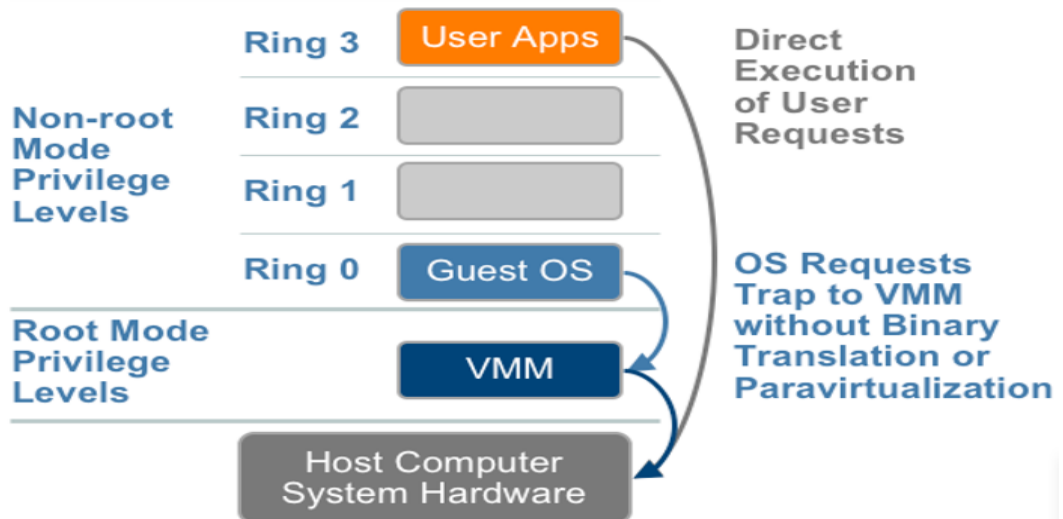


Figure 13: Hardware Assisted Virtualization [28]

The first generation of this virtualization technique was (VT-x) for Intel and AMD-V for AMD. The execution mode of the processor provided by these two manufacturers, the root mode, allows the hypervisor to run in a layer that helps it take control of the guest system's operation. One of the

characteristics of hardware-assisted Virtualization is that it is compatible with multiple operating systems and does not require any modification to the guest operating system's kernel.

-Operating System Virtualization

Operating system virtualization is the layer that runs on top of the host operating system by providing isolated partitions to run the VM in the same kernel and monitoring their interaction with the underlying operating system. This virtualization technique supports only the host OS, provides good performance, and ensures that the guest OS is the same OS as the host operation [2].

Two of the types of Virtualization are memory virtualization and network virtualization. In memory virtualization, the hypervisor is used to locate the physical location of both the machine memory and the guest memory. The shadow page tables create the overload of memory virtualization in the hypervisor, synchronized with the guest page tables when the guest operating system upgrades.

Network virtualization is an application of a virtualization technique that allows the migration of logical devices and functions to Software and the physical network. To provide good performance, network virtualization requires isolation between the virtual machine and the physical environment. This is made possible by the hypervisor that provides a network interface card (NIC) with emulation to provide logical network devices.

2.1.6 Benefits of virtualization technology

Virtualization technology, which is part of a general trend in the IT world that includes automatic and utility computing, has the usual goal of centralizing tasks while improving scalability and workloads. It offers many advantages, and its evolution has brought about significant changes in the IT field. Here are some of the benefits of Virtualization:

Table 1. Advantages of virtualization

Advantages of virtualization technology
1) Lower costs: virtualization allows to reduce the cost in the use of material resources.
2) Disaster recovery: virtualization gives the possibility to do disaster recovery even if in an emergency situation.
3) Increased utilization of resources and uptime
4) Faster backups: virtualization enables the migration and redeployment of servers and machines.
5) Energy efficient, minimized management time and easier testing

Concluding the analysis of virtualization technology, it is essential to note that Virtualization is a crucial technology for the evolution of many technologies such as Cloud Computing. The approach between virtualization and cloud computing is also confirmed by Portnoy M, which considers Virtualization as a basis for the evolution of data centers: "Cloud Computing" [4].

2.2 Cloud Computing

In the introduction to this thesis, cloud computing was described as a technology that facilitates network access to configurable computing resources. According to NIST SP 800-145, cloud computing makes it possible to access a shared set of configurable computing resources such as networks, servers, storage, applications, and so many other services that can be quickly provisioned and published [30]. This section presents an analytical study of cloud computing technology. It will describe the elements of its architecture, its service models, and its basic deployment models. This section will also approach Virtualization in cloud computing; it will present some Open Source Cloud Computing Technologies and some cloud computing benefits.

2.2.1 Cloud computing architecture

Cloud computing presents new capacities that are structured in such a way as to allow it to program its services, to ensure the capacity to create applications from components and to virtualize IT resources. Cloud computing has both hardware and software services that form a package commonly known as a platform. The different layers of the cloud computing platform use a few standard protocols, like those of the Internet, to ensure their communication. Each of its protocols plays a particular role; for example, the XML protocol is used as the messaging format, the SOAP protocol (Simple Object Access Protocol), considered as an object model, is another messaging protocol used for communication of distributed elements of an application, and several other discoveries and description protocols that are used to manage transactions based on the Web Service Description Language (WSDL) [3]. Apart from these standard network protocols, cloud computing also relies on the progress of virtualization technology to create a system of pooling of resources and their partition as needed. This is made possible by its architecture that runs Software on virtualized hardware in different locations.

Barrie Sosinsky, in his book "Cloud Computing Bible," describes the architecture of cloud computing in terms of two architectural layers: A client as a front end and the "cloud" as a backend [3]. Compared to an Iceberg, each of these two layers hides below large architectural components, standard protocols, complementary cloud computing functionalities, and services controlled by an application programming interface (API). Among the elements of the cloud computing architecture, some principal is presented here:

- **Composability:** cloud computing uses a collection of components to build applications; this is called composability. A component can be composable when it is modular, which means autonomous, cooperative, reusable, replaceable, and stateless, which means that it executes transactions freely without the influence of other transactions. However, some non-stateless transactions, such as cloud computing applications using brokers, transaction controllers, or service buses [3]. The composability of Software and hardware in cloud computing makes solutions more interoperable and facilitates system design implementation.
- **Infrastructure:** This element contains hardware and software components such as servers, storage, network, and Virtualization Software. Virtual servers have the same characteristics

as physical servers; they host virtual machines based on applications. The figure below shows the server elements such as the hypervisor, memory, CPU, TCP / IP protocol ...

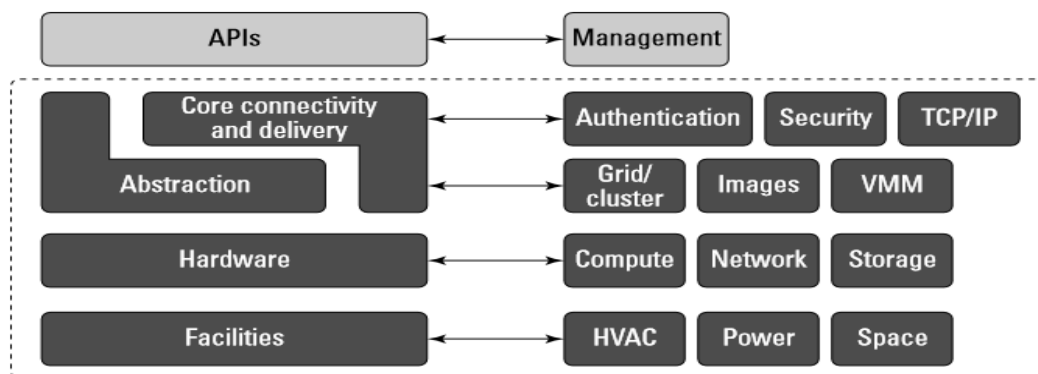


Figure 14: Cloud computing: server [3]

- **The platform:** A Platform is a virtual appliance whose installed Software is managed by the supplier's API [3]. It has tools to store, develop and test applications by measuring their performance. It is a software layer that creates services offered by certain suppliers such as the Force.com platform from Salesforce.com, Windows Azure platform, Google Apps, and Google AppEngine.
- **Virtual Appliances:** It is an ordinary deployment object, a virtual machine image file that consists of a preconfigured operating system environment and a single application. It is considered an innovation area and made up of Software installed on virtual servers to run particular machines. Often deployed as a virtual machine or a subset of a virtual machine, the Virtual Appliance (VA) operates on virtualization technology, such as VMware Workstation, and enables more complex services to be brought together. Amazon Machine Images are an example of virtual appliances; they run on a third of the Xen nodes that make up the EC2 system of Amazon Web Service [3].
- **Communication protocols:** cloud computing uses standard internet protocols, including HTTP and HTTPS transfer protocols, to ensure communication. Currently, the message transmission standard is based on the Simple Object Access Protocol (SOAP), which uses XML for its messages, and on RPC (Remote Procedure Call) and HTTP to transmit messages. The use of the SOAP protocol with the discovery and description model WSDL (Web Services Description Language), which defines a Web service's public interface, has allowed the creation of several "WSstar" extensions, the best known of which are WS-Addressing. , WS-Discovery, WS-Eventing, WS-Federation, WS-MakeConnection, WSMessaging, WS-MetadataExchange, WS-Notification, WS-Policy, WS-ResourceFramework, WS-Security, WS-Transfer and WS-Trust [3]. All these extensions use the SOAP protocol, which gives them access to multiple remote server applications. The advent of DCOM (Distributed Common Object Model) and CORBA (Common Object Response Broker Architecture) made possible the interaction between Software from different computers. To standardize resources on the Web, the Representational State Transfer (REST) gives a global identifier to each resource, ensuring their communication with users. Besides REST, cloud services use many

other data exchange standards such as Atom Publishing Protocol (APP) and Microsoft's ADO.NET [3].

- **Applications:** cloud computing virtualizes resources, with its concept of orchestration and service bus, which controls its components, by offering high performance for the proper functioning of applications such as Online File Storage, Web applications E-commerce software.

2.2.2 Cloud computing service models

Cloud computing is made up of layers on which the applications are hosted. The contribution of virtualization in storage and Connectivity has enabled cloud computing to develop its services in an architecture that makes them more efficient and easier to use. These services are SaaS (Software as a Service), which allows direct interaction with hosted applications and makes them available via the Internet, PaaS (Platform as a Service), which categorizes cloud computing capabilities, specific products and services; and the IaaS (Infrastructure as a Service), which allows the development of virtual computer systems or networks [3].

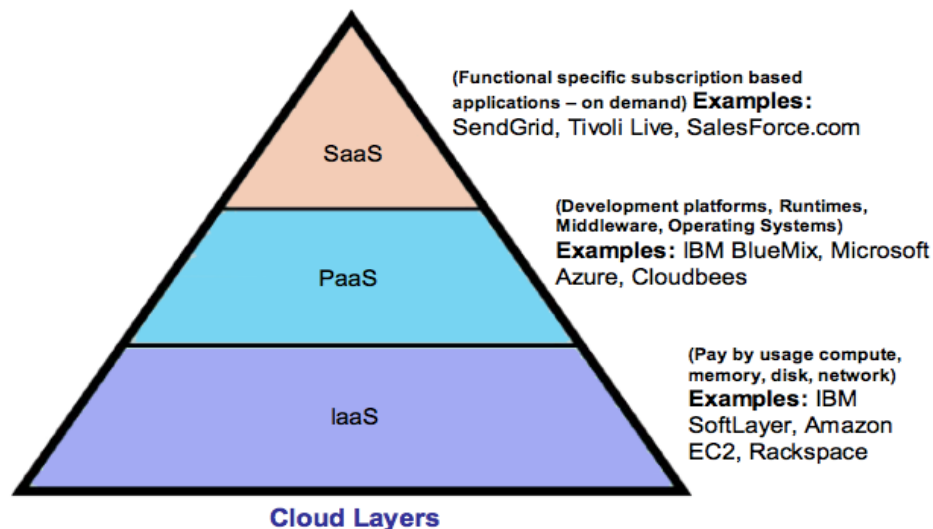


Figure 15: Cloud computing service models [29]

- **Infrastructure as a Service (IaaS):** it is a cloud computing service model that consists of virtualized hardware such as servers, storage, network infrastructure, of which the supplier is the author and owner. This service provides resources as needed. These resources, although virtualized, are based on real systems. This means that the physical servers handle user requests to virtual systems. In the IaaS infrastructure, the execution of server workloads, which relies on a pool of virtualized machines, RAID storage, and network interface capacity, is based on a logical unit consisted of LUN, logical container storage, the Cloud interconnects layer, which is a virtual network layer that has IP addresses, and the virtual application software layer. Some examples of IaaS are Amazon Elastic Compute Cloud (EC2) from Amazon Web Services (AWS). Its servers are run on a virtualization platform (Xen) and are partitioned into logical computing units of different sizes.
- **Platform as a service (PaaS):** it is a dedicated software environment for creating custom solutions. Its integrated tools allow the deployment of applications by creating user interfaces.

This model allows user interaction with the software and gives suppliers the responsibility of managing the operational aspects of service and maintenance. The Google App Engine platform, Microsoft Azure and Force.com are some examples of PaaS [3].

- **Software as a service (SaaS):** it is a complete model of cloud computing service that provides the complete infrastructure, software, and solutions accessible via the Internet. Its services are billed by suppliers based on use; they also provide monitoring, maintenance, and updates. By ensuring data sharing via a single instance multi-instance model, SaaS ensures that all users have the same software version to ensure compatibility. Google Apps, Desktop as a Service, Salesforce.com, and CRM are SaaS examples [3]. Several SaaS software uses open-source software such as Linux, APACHE, MySQL, etc., which are less expensive and more portable, thus creating the Open SaaS concept.

2.2.3 Cloud computing deployment models

This model presents a description of cloud computing's implementation, hosting, and access by users. Its operation is based on the virtualization of servers' computing power in segmented software applications capable of developing processing and storage capacities. The popular types of deployment model are:

- **The private cloud:** according to NIST (National Institute of Standards and Technology), the private Cloud is a cloud infrastructure implemented to be used, owned, and managed exclusively by a single organization or by a third party on or off its premises [NIST SP 800-145]. There are also external private clouds, the management of which is entrusted to an external service provider.
- **Community cloud:** It is a cloud infrastructure implemented by a group of organizations forming a community and sharing concerns such as mission, security requirements, policy, and compliance considerations. It can be owned, managed, and operated by one or more community organizations, or third party or a and it can exist on or off the premises [NIST SP 800-145]. This model allows users to access both local cloud resources and the resources of other community member organizations.
- **Public cloud:** It is a cloud infrastructure open to everyone and which is implemented to be used by the general public while hosted, owned, managed, and operated by a supplier who defines the same level of service for all users.
- **Hybrid cloud:** it is a composition of two or more separate cloud infrastructures such as private, community, or public.

2.2.4 Virtualization in the cloud

Virtualization technology is considered the foundation of cloud computing, which uses it to perform certain functions such as partitioning, aggregating, isolating, and abstracting resources by forming logical instances. These resources are grouped and give rise to creating a virtual infrastructure based on the physical infrastructure. This makes it possible to increase or decrease virtual resource capacities flexibly. The cloud data center uses virtualization software to create virtual machines to

manage storage resources, control virtual switches or routers from network devices, and secure virtualized network functions.

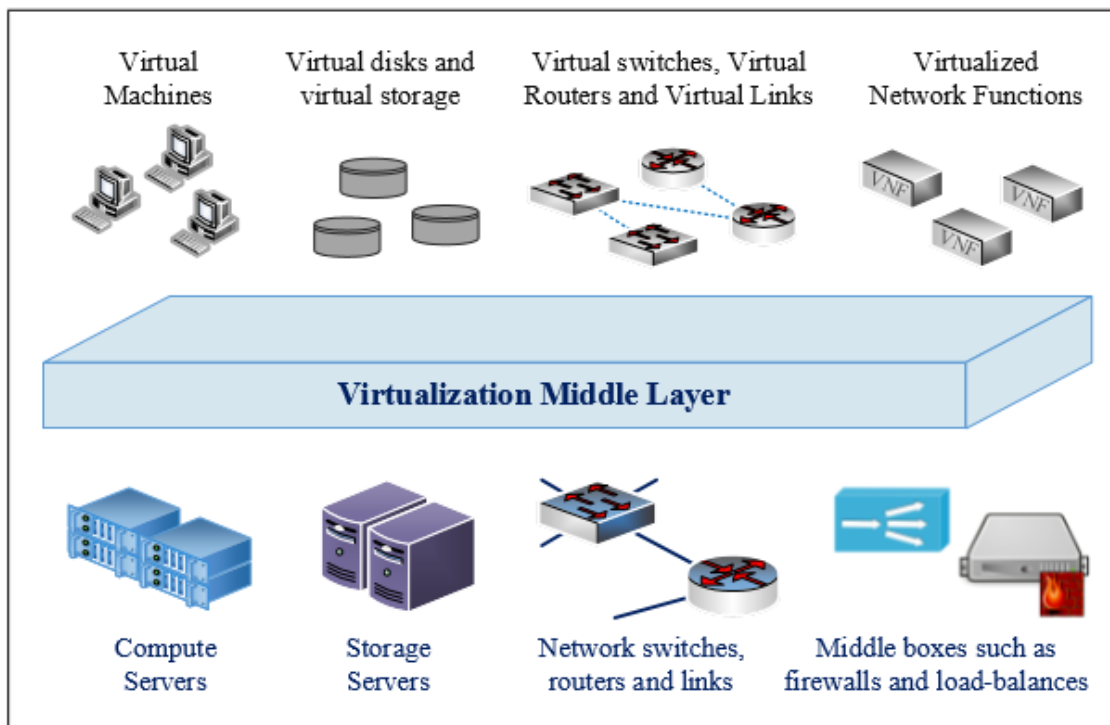


Figure 16: Creation of logical instances by virtualization layer[3].

- **Storage resource virtualization:** data storage is one of the most widely used and best-studied cloud services areas. The use of data centers and traditional computing environments, such as local physical storage drives and networked storage solutions such as storage area networks (SAN) and networked storage (NAS), has made possible data storage. Cloud computing performs three main abstractions of data storage: the database level, which makes virtualized storage resources available for platform services, Object Storage, which stores files while keeping a layout flat and giving them a unique identifier; and Bulk Storage[31].
- **Virtualization of computing resources:** the evolution of virtualization technology through virtualization of computing resources has favored creating the cloud service model and has enabled servers to support the simultaneous execution of several computer systems. The virtualization of computing resources such as the CPU and the memory is carried out via the partition, the aggregation, and the abstraction of resources, with hypervisors' use
- **Virtualization of network resources:** Virtualization has enabled network resources' abstraction and offered high performance in the network working system. Currently, virtualization of the cloud network infrastructure is based on SDN, one of the evolutions of Virtualization technology. SDN allows the creation of a flexible and programmable network.

2.2.5 Open Source Cloud Computing Technologies

The IT environment has experienced significant progress using several technologies, classified as the open-source technologies emerging in the cloud computing field. By their principles of flexibility, openness, reliability, and efficiency, these technologies contribute to the development of all sectors of cloud computing, such as infrastructure operations, platforms, and applications. Also, using an open-source cloud platform reduces the cost and offers freedom of choice for various frameworks, clouds, and services. Using open-source, businesses become fast, agile, streamlined, and pass from transaction-based services to relationship-based services. Open sources are supported by all the cloud pyramid services, namely IaaS, PaaS, and SaaS.

Infrastructure as a Service (IaaS) has been developed to support storage, hardware, servers, and network components. It provides access to the resources contained in the Cloud via the Internet. The best known of its open-source software are OpenStack, OpenNebula, Eucalyptus, and CloudStack.

OpenStack is an open-source platform that uses virtual resources to create and manage public and private clouds to deliver infrastructure as a service (IaaS). It is a series of script commands grouped in a set of projects which take over the tasks for the creation of cloud environments by relying on virtualization technology, which creates a layer of virtual resources, and on a system of basic operating system (OS), which executes OpenStack script commands. Its software platform consists of several interdependent components for the control of hardware pools, processing resources, storage, and networking. The extraction of virtual resources uses a coherent set of application programming interfaces (API), and its management is ensured by using a Web dashboard, command-line tools, or services RESTful web.

OpenNebula is an open-source project that develops solutions to build and manage a data center's virtual infrastructure to create private, public, and hybrid implementations of infrastructure as a service (IaaS). It orchestrates storage, network, virtualization, monitoring, and security technologies for service deployment. OpenNebula has a toolbox that contains integration, management, scalability, and security functionality. It offers users the possibility to choose several cloud interfaces such as Amazon EC2 Query, OGF Open Cloud Computing Interface, and vCloud and hypervisors.

Eucalyptus (Elastic Utility Computing Architecture to link your programs to good systems) is open-source computer software used to develop cloud computing environments compatible with Amazon Web Services (AWS). It uses the existing virtualized infrastructure to create cloud resources for computing, network, storage, and pooling according to the change in application workloads. The Eucalyptus user console offers an interface that facilitates these resources' management and configuration by accessing virtual instances via SSH and RDP mechanisms. It is also possible to configure its IaaS service components such as Cloud Controller, Cluster Controller, Walrus, Storage Controller, and VMware Broker to increase their resistance to failures. It runs different versions of Windows and Linux virtual machine images [36].

Apache CloudStack is an IaaS software developed for the deployment and management of large networks of virtual machines. It offers a full stack of computing orchestration features, network as a service, user and account management, a native API, resource accounting, and a user interface (UI).

Additionally, it provides an easy cloud management option using a web interface, command-line tools, and a fully RESTful API compatible with AWS EC2 and S3.

The PaaS solution provides an infrastructure for the design and deployment of software applications, networks, servers, operating systems, and programming applications. It supports several open sources, including OpenShift, WSO2 Stratos, Cloud Foundry, and Cloudify.

OpenShift Origin is a PaaS computing platform from Red Hat to create, deploy, test, and run applications. Besides, it offers disk space, CPU resources, memory, network connectivity, and an Apache or JBoss server to properly deploy the applications using a file system model PHP, Python, Ruby, or Rails [36]. The Web management console, command-line interface tools, or a REST-based API allow managing its interaction with users. The OpenShift platform is based on Kubernetes and uses a flexible installer with full API support that allows it to be extended according to user needs.

WSO2 Stratos is a complete cloud solution, an open-source PaaS based on WS02 Carbon to manage the underlying cloud infrastructure, application scalability requirements, versions, and documentation of the API. Its store concept makes it easier to test and evaluate APIs by offering direct deployment, offering options for flow control, and automatic recovery in the event of an endpoint suspension [36].

Cloud Foundry is an open-source PaaS created by VMware to facilitate rapid development, testing, deployment, and evolution of applications using a scalable architecture and workflows compatible with DevOps. It also offers tools for monitoring deployed applications, detecting failure conditions, and resolving them. Supporting languages such as Python, Ruby, PHP, Java, and Go, Cloud Foundry can be deployed on VMware vSphere and other cloud infrastructures, such as HP Helion, Azure, or AWS. Its recent version, Micro Cloud Foundry, can be downloaded for execution on laptops or local machines.

SaaS is a software service with specific functionalities that users access via a browser on a small stream. It has specific platforms that can be considered open-source in its broadest understanding, such as WordPress and SugarCRM.

WordPress is a content management system (CMS) and an open-source SaaS, installed on a Web server, written in PHP associated with a MySQL or MariaDB database for the creation and deployment of web application and media such as websites, pages Web, online stores...

SugarCRM is a cloud-based application that features customizable home pages, an "assistant" tool, a shortcut bar, and downloadable plugins. Its architecture allows it to support Windows, Linux, MySQL, SQL Server, and Oracle systems, and its operation is possible on a Windows operating system just like on a Mac.

2.2.6 Benefits of cloud computing

Table2. *Benefits of cloud computing*

Benefits of cloud computing
1) The reduction of Implementation and maintenance costs
2) The mobility, flexibility and the evolution of infrastructure
3) The reduction of implementation and deployment time of resources
4) The transformation of IT services
5) High performance of applications

In conclusion, cloud computing is a technology that puts IT resources such as data storage, computing power, network resources, and users' disposal according to their demand via the Internet. The availability of high capacity networks, virtualization technology, service-oriented architecture, and autonomous computing enabled him to come today a technology that offers many advantages and contributes significantly to IT development.

3 CHAPTER 2nd: SOFTWARE DEFINED NETWORK AND NETWORK FUNCTION VIRTUALIZATION

In the introduction of this thesis, the network device was described as a set made up of two planes: a data plane which is responsible for the connection to the different network ports that a peripheral can have, and a control plane which is considered to be the very brain of the peripheral. By the way, in the context of a centralized control plane, the devices' commands and manipulations are carried out by the brain of the network device. The advent of virtualization and cloud computing technologies allowed a remarkable evolution of these two plans of the network device until creating what is called today the software-defined network (SDN). Nadeau T. and Gray K. describe this network in their book "SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies" as an architectural approach that closely links the interaction between applications, services, and network devices to optimize and simplify network operations [8]. It makes it possible to program network devices and make them flexible using a logically centralized network control point, the SDN controller, which can ensure orchestration, mediation, and two-way communication between the applications that have an interaction with the elements of the network and vice versa. This controller also provides and manages the maintenance of all network paths and the programming of network devices under its control.

OpenFlow, a protocol designed for devices that contain only data planes, describes responses to commands from the SDN controller. This chapter discusses the network aspects of software-programmable networks while providing necessary coverage for virtualization. It also explores advances in network technology that have enabled the virtualization of the network, storage, and compute resources part of SDN. More precisely, this chapter will present the Centralized Control and Data Planes, OpenFlow, SDN Controllers, Network Programmability, Network Function Virtualization, and a conclusion with SDN's advantages.

3.1 Centralized Control and Data Planes

SDN is based on the separation of control and data planes from a network device. The network's high availability requirements logically led to the centralization of the control plane in a model where a switching device under its control is located at a distance. It is important to note that, in general, a centralized control system offers new capabilities working in conjunction with a decentralized control system without replacing or eliminating the existed device's control plan [8]. This means that certain classic functions of the device control plan, such as ARP processing or the MAC address, are retained. The SDN system works with a hybrid operation in which the sub-layer, provided by the distributed control plane, is used by a logical overlay, which is provided by the centralized control plane, as network transport.

The control plane is located at a very high level to establish a local data set used to create transfer table entries called the transfer information base (FIB). The data plane uses this table to transmit traffic that takes place on a device between the input and output ports. Communication between control plane instances in-network helps maintain a consistent routing information base (RIB), a set of data used for storage of network topology. The consistency and stability of RIB allow FIB to be

programmed after development by the controlling program entity from a good view of the network constructed using information gathered through speech with other control plan instances [8].

The mechanics of the control and data planes are illustrated in *Figure 17*, representing a network of interconnected switches.

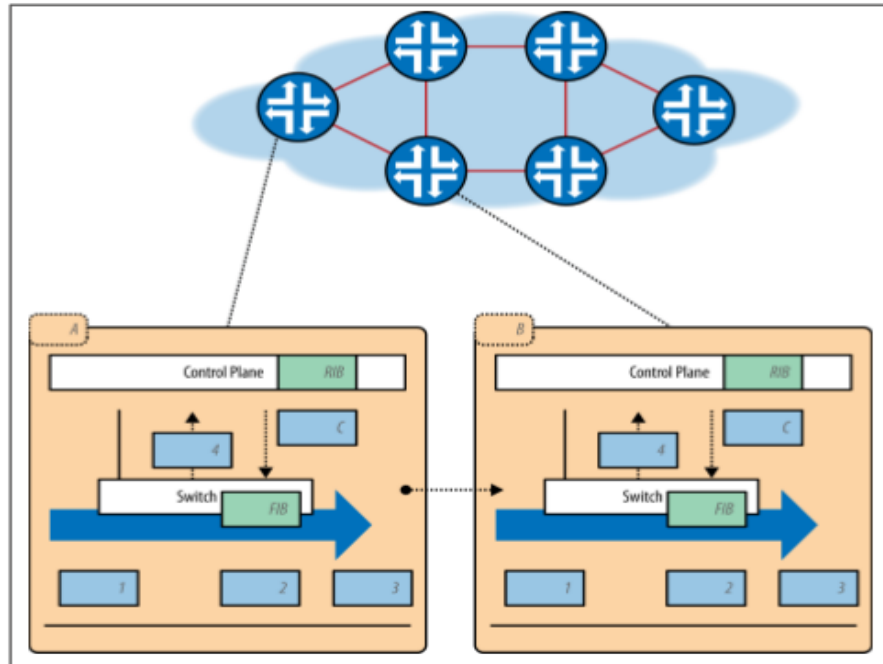


Figure 17: Control and data planes of a typical network [8]

Analyzing figure 15, the control and data planes represent a network of interconnected switches, in which switch A receives the packets through the input ports of the line card where the data plane is located and then transmits them to the switch B. Each of these extensions has a chassis that contains two separate planes: a control plane that runs on its processor and a data plane that runs on the other. The control plan processes the packets and controls the traffic, sometimes causing RIB alteration and update alerts. When the RIB regains its stability, the FIB update is carried out in the control plane and the data plane. The transmission subsequently adapts to these updates.

It is important to note that a layer two control plan mainly deals with physical layer addresses such as MAC addresses, while a layer three control plane focuses on network layer addresses such as those of the IP protocol [8]. A large number of end hosts prevents Layer 2 networks from finishing well at scale, which leads to the need to merge the problems of scaling and control plane design with those of Layer 3. In fact, in a layer two network, the transfer is based on the storage and accessibility of MAC addresses, which is sometimes tricky, especially in an extensive network such as that of companies. In a layer three network, the transfer is based on the accessibility of network addresses by a destination IP prefix and addressing families for unicast and multicast. Modern technologies have allowed the use of Layer 3 networking to segment or join Layer 2 domains to solve its scale problems. This is done by connecting Layer 2 bridges, which represent certain sets of IP subnets, to a Layer 3 router connected to other routers with which they form more extensive networks. The

protocols involved in packet traffic and transfer are the MPLS protocol (Multiprotocol Label Switching), the EVPN protocol (Ethernet Virtual Private Network), and the LISP protocol (Locator / ID Separation Protocol) [8]. The MPLS protocol is based on ATM technology, created by combining the best parts of Layer 2 forwarding with Layer 3 IP routing for fast packet forwarding sharing [8]. The EVPN protocol channels remote Layer 2 bridges in a tunnel over an MPLS infrastructure to allow the flow of addressing and Layer 2 accessibility without affecting the scale of the underlying Layer 3 networks. The LISP protocol makes an essential contribution to solving some of the traffic gaps. It adds new addressing domains and separates the supplier's site's address using a new card and a new encapsulation control and transfer protocol [8].

The data system manages, collects, and verifies the incoming datagram in wired and wireless networks. The control plan programs well in advance of the FIB table to allow packets to be processed efficiently and quickly based on identifying the destination. If the destination is unknown, packet processing is carried out by the control plane using the RIB. Other services, commonly known as transfer functionality, are implemented by the data plane and allow local modification of transfer search results. These features include an access control list entry that can specifically delete a flow and a QoS policy to map it to an output queue. The transfer of information begins when the control plane instructs the data plane to proceed with packets' transmission. The data plan then uses transfer mechanisms to guarantee the table's synchronization and verification are distributed software versions after their programming.

When network components resided on the same hardware family base, the variation due to the balance between service, management, control, and data plans created many stability and scale issues that affected network performance [8]. These designs also required a high cost because they were involved, which is why there was the need to separate the two plans based on an SDN solution that brings speed, performance, and productivity in current networks' functioning.

The concept of a centralized control plan offers several advantages, the main one being simplifying the control of a programmable network. This allows the application to manage the network by interacting with the centralized control point instead of directly contacting all the elements. The SDN solution can operate according to two models.

The first is the programmability via a controller, a model in which network equipment such as routers and switches are replaced by a universal physical machine that processes IP flows by following the SDN controller's instructions. These instructions can be VLANs (access port) assignments, routing, and specific processing of network services. This is functional by using an SDN architecture based on the OpenFlow injection protocol, on the controller, which provides an abstraction of the transport plan, and on programmable interfaces (API) of network applications. In general, the controller gives commands to network equipment and globally manages its operation. The SDN controller has a north interface called Northbound API, through which it offers high-level APIs to SDN applications, such as a network management request, service priority, or access control. After orchestrating the application request, the SDN controller goes through the south interface using CLI protocols (Command Line Interface), the OpenFlow protocol, or other protocols for transmitting the corresponding routing tables. The data plane then receives the rules transmitted by the controller and begins routing traffic.

The second model is the programmability via SDN Overlay; it is created by the applications of an "overlay" network, which is nothing other than a virtual network above the physical network to ensure the entirety of the services. In this network, packets are encapsulated by protocols such as VxLAN (Virtual eXtensible LAN) and then transmitted using the existing infrastructure [8]. Initially created to facilitate the interconnection of cloud servers based on the principle of virtualization, the Overlay is today one of the main elements for network programming. The physical network interface is virtualized by the hypervisor, which then shares it with virtual machines by creating a virtual switch that allows VMs to each other. The hypervisor also manages the encapsulation and decapsulation of the frames in the VxLANs protocols (VTEP: VxLAN Tunnel EndPoint), which cloud computing hosts use. Starting from virtualization and the fundamental principle of SDN, which is based on the separation of the control plane and the data plane, the SDN overlay orchestrates the VMs, provides automatic control and deployment of the servers using the elements of the infrastructure (storage, computation, networking) which are virtualized and grouped into resource pools. The Cloud Management Platform (CMP) is used to provide virtual networks and activate network services according to workloads [8].

3.2 OpenFlow

The OpenFlow results from research carried out by Stanford University on networks for the creation of experimental protocols on the network of their campus [8]. The original idea was that OpenFlow replaces Layer 2 and Layer 3 protocols in network devices such as switches and routers. Today it plays a vital role in implementing the control plan and in the functioning of SDN (Software Defined Network). It consists of protocols and an API that is responsible for the program for the controller's operation. Its standardization, use, and marketing are guaranteed by the Open Networking Foundation (ONF), which at the same time promotes the development of SDN. Whereas in the traditional functioning of IP networks, each network element, such as the Ethernet switch and the router, performs its functions of the control plane and the data plane. SDN provides a solution for centralized management of the control plane by leaving the data plan's equipment management. To reach this goal, the SDN controller uses the OpenFlow protocol to transmit commands to the switch to perform data plane programming. Therefore, OpenFlow is at the heart of SDN, mainly through its key components: the separation of control and data planes, instantiation of transmission state, and network programming[8]. The SDN controller provides the API to SDN applications to run services such as flow routing, filtering, and end-to-end security. OpenFlow occupies a place in the SDN architecture infrastructure, which is made up of network equipment responsible for traffic management and support for the OpenFlow protocol. The control layer contains the SDN controller, which globally manages the network and infrastructure equipment, and the application layer, which is responsible for automating network applications using programmable interfaces.

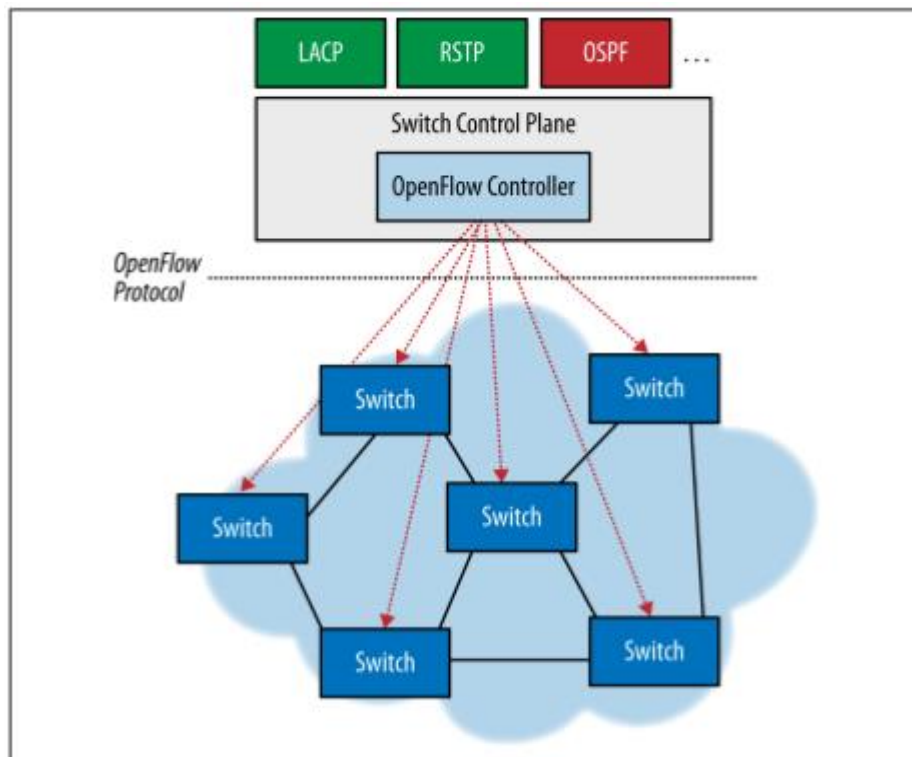


Figure 18: OpenFlow architecture [8]

The figure 16 illustrates the Open Flow architecture showing how it works. Depending on the role that the protocols play, there are two parts of OpenFlow:

- A wired protocol that manages control establishes a message structure for communicating flow level changes (flow mods), defines another structure for switch ports and tables, performs stored actions, and passes metadata. It is also responsible for collecting statistics and taking charge of the tables [8].
- A configuration and management protocol that assigns physical switch ports to a specific controller and defines his availability or unavailability.

An OpenFlow switch has a flow and group table and an OpenFlow channel that opens to another controller. Therefore, the controller uses the OpenFlow protocol for switch management and can reactively or proactively modify flow entries in flow tables. The flow tables in the switch each have a group of flow entries with match fields, counters, and an instruction. The instructions make it possible to describe the transfer, modify the packets, and the group table in processing. Those in the pipeline allow packets to be transferred to the next tables for their processing and the exchange of information between the tables in the metadata form.

OpenFlow ports are Ethernet ports, network interfaces that allow the flow of OpenFlow processing packets to the rest of the network and vice versa, and the logical connection of OpenFlow switches. Ports have associated port IDs, Ethernet MAC addresses, and the OpenFlow controller can change their configurations. Input ports receive OpenFlow packets, the OpenFlow pipeline processes them and forwards them to an output port. Their properties include support for Ethernet functionality, speed, operating mode, and type of physical medium. The three types of OpenFlow switch ports

are[32]: the physical ports, which are the hardware interface ports of the switch, the Logical Ports which are higher-level abstractions of the switch through the use of non-OpenFlow methods such as link aggregation groups and the Reserved Ports which define the generic transfer of the shipment to the controller, the flood and the transfer using non-OpenFlow methods.

The OpenFlow compatible switches have two types: simple OpenFlow and the OpenFlow hybrid switch. There is an OpenFlow pipeline or the regular pipeline with flow tables with multiple entries and processes all packets in switches.

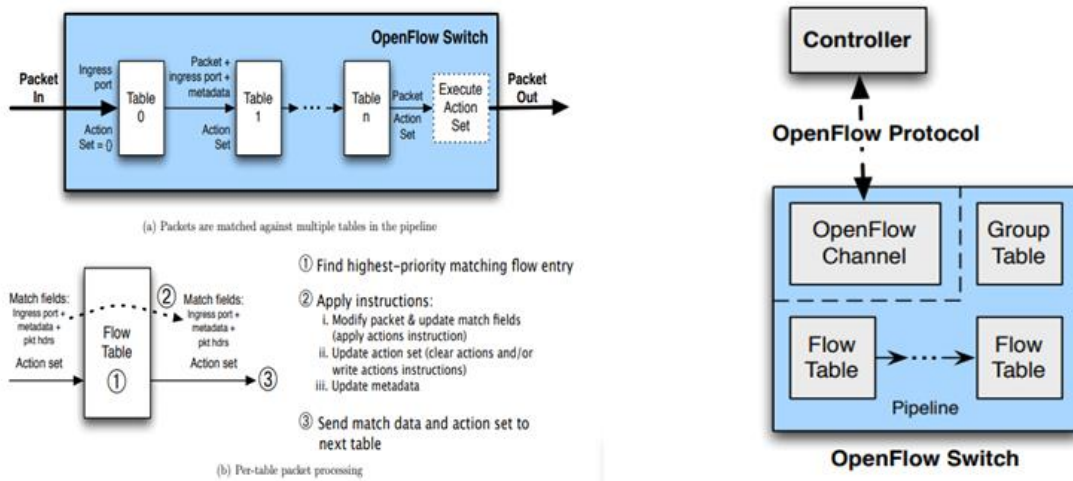


Figure19: Packet flow through the OpenFlow processing [32]

The flow tables of an OpenFlow switch have numbers that always start with the flow table 0 to organize processing. After finding a flow entry, the instructions direct the packet to another higher flow table where it is processed according to the defined specifications. The flow table entry is composed of [32]:

Table 3. The components of the flow table entry

Match fields	Priority	Counters	Instructions	Timeouts	Cookie
--------------	----------	----------	--------------	----------	--------

1. Match fields: this part is used to match the packets, consisting of the input port, headers, and metadata specified by a previous table.
2. Priority: it is a correspondence that determines the priority of the flow entry.
3. Counters: it is updated when the packages are matched.
4. Instructions: these are commands for modifying the set of actions or processing the pipeline.
5. Timeouts: this is the maximum duration of the flow before it expires by the switch.
6. Cookie: this is an opaque data value chosen by the controller to filter the flow statistics, its modification, and its deletion.

3.3 SDN Controllers

As mentioned above, the data plane contains network equipment such as switches and routers, and the control plane has a controller responsible for abstracting and managing network elements. Therefore, the SDN controller works in collaboration with a data model that links managed resources, policies, and other services. There is an application programming interface (API) that makes the controller services available to applications. They form an environment with a secure TCP control session between the controller and the network components, a standardized protocol, a device, a computing system, and so many other network services. Apart from OpenFlow and proprietary protocols, the SDN controller can also use the IP / MPLS network's functionalities for the creation of MPLS VPNs. It is considered an integrated solution for managing the state of the network and the data center's resources, for example, computation, storage, and virtual machine images. Other SDN controllers support open-source APIs (OpenStack, Cloudstack, etc.), which manage the network abstraction. The SDN controller acts as a proxy in a distributed environment and a flow provisioning and service management agent when it is a centralized environment [8].

With SDN, network intelligence is essentially found in a controller that performs all the calculations and ensures network applications and services. The modules that provide the routing service are the topology manager and the link discovery modules that discover and maintain the physical links in the network. Some features of SDN controller are [33]:

- *The programming language:* the programming language allows quick access to memory and its management. The languages that are most used for programming SDN controllers are Python, C ++, and Java.
- *Support for OpenFlow:* the OpenFlow protocol is at the very heart of the SDN solution; it allows direct management of the switch transmission plan.
- *Network programmability:* The network's programmability is the very foundation of the SDN to manage Connectivity and the deployment of network services by implementing automation and dynamicity in the management process[33].
- *Efficiency:* it is a term that determines the performance, reliability, scalability, and security of SDN controller
- *Southbound interfaces:* the efficiency of network control is made possible thanks to the Southbound APIs that the controller uses to dynamically modify the transfer rules installed in the data plan devices [33].
- *Northbound interfaces:* The application layer uses the APIs to the north for communication with the controller and the support of several applications.
- *The partnership:* the economic capacity of having partner organizations is one of the SDN controller's significant points. Because it managed to attract the attention of Cisco, Linux Foundation, Intel, IBM, Juniper, and many others.

The table below provides a list of some SDN controllers, the most used are ONOS and OpenDaylight, coded with Java to run the platforms. The ONOS controller was created for operator networks, and its architecture allows the maintenance of high-speed and large-scale networks by supporting hybrid networks. OpenDaylight was initially focused on data centers, but today it can support different applications because it has been enriched with additional interfaces to the south, such as HTTP,

COAP, and support for the OpenStack Neutron plugin [8]. Another controller that is most used by small businesses and research applications is Ryu. It is coded in Python and contains several functionalities for developing applications and modules, even though it does not have high modularity and cannot run across platforms.

Table 4. The list of some SDN controllers

Controller	Programming Language	GUI	Distributed /Centralized	Platform Support	Southbound APIs	Northbound APIs
ONOS	Java	Web based	Distributed	Linux, MAC OS, Windows	OF1.0, 1.3, NETCONF	REST API
Open-Day Light	Java	Web based	Distributed	Linux, MAC OS, Windows	OF1.0,1.3, 1.4,NETCONF/YANG, OVSDB, PCEP,BGP/LS, LISP, SNMP	REST API
NOX	C++	Python +QT4	Centralized	Linux, MAC OS, Windows	OF1.0	REST API
POX	Python	Python +QT4	Centralized	Linux, MAC OS, Windows	OF1.0	REST API
RYU	Python	Python +QT4	Centralized	Most supported on Linux,	OF1.0,1.2,1.3, 1.4,NETCONF,OF CONF	RESTfor Southbound

3.4 Network Programmability

Network programmability is at the center of the critical foundations of software-defined networks. This concept is based on how network devices are managed and can interact with each other. These devices are easily programmable and have a reliable channel to guarantee two-way communication between them by forming a tightly coupled feedback loop. This offers high speed and good performance in the execution of services. In most cases, interfaces must have critical characteristics such as bidirectional, user-friendly for applications, and self-describing to achieve Network Programmability. Some technologies used in network programmability are:

- *The management interface*: it allows the proper management of network devices by providing a coherent operational view of a device and facilitates its configuration and operating state. It consists of two key components: a protocol that describes the syntax and semantics associated with the sending or receiving specific messages such as commands, requests, or responses to previous requests, and a specification of format messages and their meanings [8].
- *The Application-Network Divide*: Unlike the traditional network where most of the elements (routers, switches, or firewall) supported specific traditional interfaces, for example, the command-line interface (CLI)[8]. This solves the old interfaces used in traditional networks towards which coding and paradigms were heavy, thus causing long time intervals between commands of the applications and their actual executions. This is called the application network division. JSON solves several shortcomings of the old interfaces based on a self-referential, hierarchical schema, easily integrated into Java applications and defined using human-readable XML [8].
- *The command-line interface*: it is an interface that allows communication with the peripheral.
- *NETCONF*: It is a network configuration protocol, developed in December 2006 and standardized by the IETF (Internet Engineering Task Force) to provide mechanisms allowing to install and manage the configuration of network devices using a data coding based on the extensible markup language [XML] and the transport protocol such as TCP, HTTP or HTTPS [8].
- *SNMP (Simple Network Management Protocol)*: created by the IETF, this protocol deals with essential network management directly or remotely. It, therefore, controls the elements of the network to determine their state and their performance characteristics.
- *publish-subscribe interfaces (pub-sub)*: This is a messaging model that message senders or publishers use to send messages to recipients or subscribers by publishing them in classes. Therefore, the recipients register in these classes for which they express interest to receive only messages belonging to the interested class.
- *XMPP*: it is an extensible messaging and presence protocol (XMPP) that belongs to pub-sub protocols. It allows you to implement several publishing and subscription systems based on XML (Extensible Markup Language). It is often used to provide instant messaging and near real-time information to a group of subscribers [8].
- *The Google Protocol Buffers*: it is an extensible mechanism independent of language and Google platform to serialize structured data. Its creation's initial objective was to refine the deficiencies of codes in XML and JSON by using binary coding. It offers the use of binary-

coded APIs only for APIs consumed internally and the use of human-readable APIs such as XML, JSON, etc., for publicly available APIs [8].

- *v Thrift*: It is similar to protocol buffers; it is a language that defines the interface used to create services and can be used as a remote procedure call (RPC) infrastructure. For example, the content service provider Facebook application, which encountered problems when it was created with large-scale data centers, is now using Thrift to meet its growing needs [8].

The concept of network programmability also involves the orchestration of some networks and data centers' orchestration to have a faster and more optimized deployment; it is necessary to ensure the monitoring of storage and calculation[8].

3.5 Network Function Virtualization (NFV)

The evolution of technology has enabled devices to be increasingly connected simultaneously to the ICT infrastructure (information and communication technologies), and their support requires a significant modification of the network architecture using some technologies such as network function virtualization (NFV). This technology resulted from the Layer123 SDN conference, which was sponsored by ETSI and took place in Darmstadt, Germany, in autumn 2012, during which the telecommunications service providers (TSP) had expressed their desire to virtualize services and sponsor the development of standards [8]. It is based on the separation of control plane and data plane, virtualization, SDN controllers, and orchestration applications. NFV offers better network administration, programmability, reduced cost of capital expenditure (CAPEX) and operational expenditure (OPEX), and overall network performance [34]. Also, NFV extends the concept of virtualization to networking by changing the way networks are designed, deployed, and maintained with virtualized network functions; it gives TSPs the possibility of creating new services, facilitates their management, increases the reliability and resilience of the network without deploying a dedicated physical architecture. Network virtualization provides virtualized networking at layer two and layer three, and NFV provides virtualized networking at layer 4-7 [34]. This allows communications service providers (CSPs) to operate, configure and deploy network environments flexibly.

The NFV has an architecture that can adapt to its changes and allow the development of open-source software that can run adequately on generic shared hardware. This architecture is mostly made up of functional blocks:

- The NFVI (Network Functions Virtualization Infrastructure) functional block combines physical hardware and virtualized resources.
- The VNF functional block consists of several VNFs, which is the virtualization of NF based on the hardware and several elements' management elements (EMS).
- The NFV-MANO functional block manages, orchestrates, deploys, and operates VNFs.
- The Operation Support System (OSS) and the Business Support System (BSS) block guarantees coordination with the traditional network system[34].

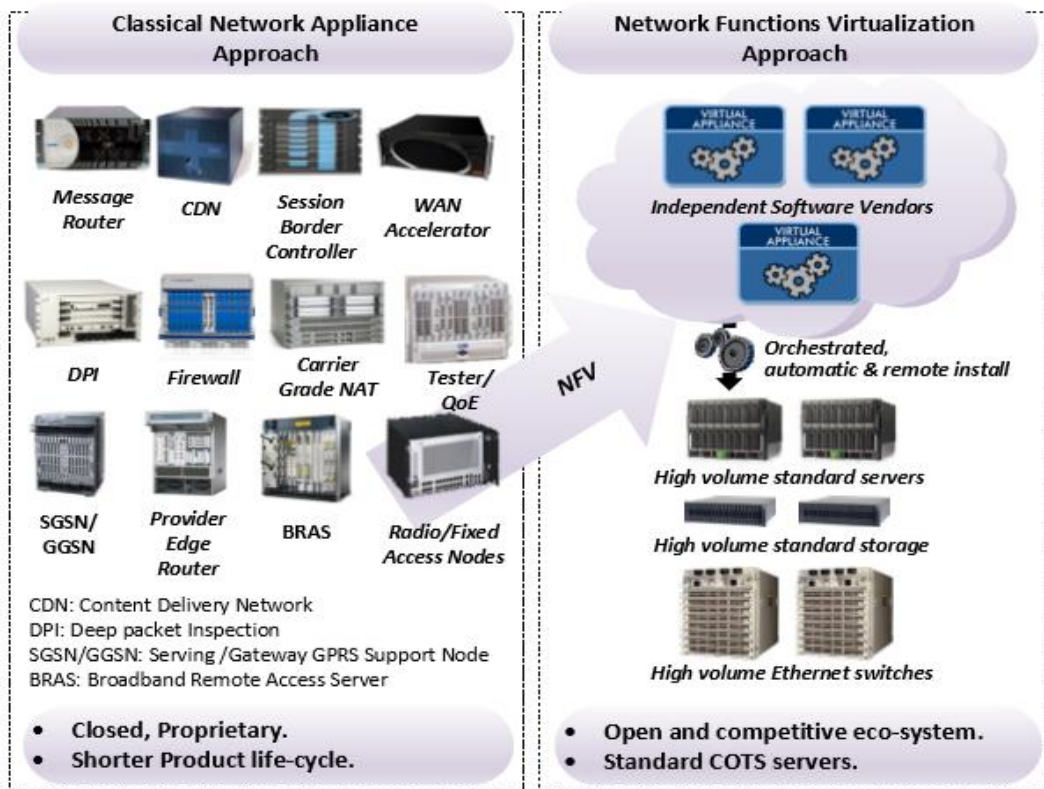


Figure 20: Traditional Hardware-based Network and NFV Approach [34]

3.5.1 The main considerations and key objectives of NFV

The NFV has considerations and objectives that must be followed to provide the network performances and function in full safety. This section presents some considerations of NFV:

- *Interoperability*: NFV is designed to be compatible and decoupled from hardware appliances when deployed in a virtualized network environment [34]. This facilitates the implementation of a dynamic range of network topology, custom routing, and handover policies on underlying systems and provides the ability to reprogram network elements as needed rather than replacing them in an expensive purchasing process.
- *Resilience*: Resilience procedures are used to overcome network problems, such as crashes, attacks, and traffic congestion, which can interrupt the operation of network services and thus degrade their performance. The NFV environment is usually spared these kinds of problems because most of its networking components and functions are software-based. Also, it has mechanisms that provide a reliable software update, thereby guaranteeing the service's continuity and availability.
- *Consistent performance*: The performance of an NFV environment is linked to several factors such as the central processing unit (CPU), memory, bidirectional virtual switch traffic (vSwitch), and hardware acceleration. These factors allow it to maximize its performance and elasticity across networks.
- *The reliability and availability of end-to-end services*: In general, network reliability is measured by the traditional low probability of network failure. Network availability is measured by the traditional low probability that the network system will operate without

failure [34]. NFV's infrastructure, its virtual links, and its defined interfaces allow it to provide a reliable network and have end-to-end services.

- *Elasticity, Automation, And Scalability:* NFV also focuses its center of interest on network capacity planning by relying on information-centric software approaches. Its elasticity and automation allow it to improve network performance.

3.5.2 Relationship of SDN and NFV

As pointed out in the introduction of this thesis, the SDN is mainly concerned with the use of resources to provide services, and the NFV is responsible for creating and supporting the life cycle of resource classes of service. SDN's profitability is based on its ability to provide services and execute them quickly and flexibly. It has a generic idea of resources that NFV mostly supplies. NFV's profitability is based on its ability to reduce time and costs when providing the class of resources. The SDN controller considers the VNF as another resource that can reside at an appropriate point in infrastructure, a node function with known connectivity points, and a known and controllable transfer function [35]. NFV enhances the agility of SDN services by quickly creating and moving virtual resources. To select the resources to be used, the SDN uses a form of coordination to inventory the capacity of the resources, the current state, the data connection, and management control channels. The NFV, in his turn, uses general-purpose computing and storage resources to adapt the quantity and location of virtual resources as needed. This makes it possible to reduce the cost, the computing, storage, and network capacities blocked [35].

When a service request arises in the operation of the network, an SDN controller orchestrates network services on non-NFV resources. The selected resources are then integrated into the end-to-end service built by the SDN controller on behalf of the client. The NFV then ensures the lifecycle maintenance by initializing resources with appropriate global attribute values [35]. To provide automation of transfer attributes, the SDN controller cooperates with a hypothetically separate NFV domain. It often proceeds by downloading, installing, and configuring applets on specific platforms at the appropriate topological points using NFV concepts and tools. If a network uses several different SDN and NFV domains, it is essential to coordinate its activities to prevent them from working against the grain.

3.5.3 Benefits of Network Function Virtualization (NFV)

Table 5: Benefits of Network Function Virtualization (NFV)

1.Reduce the need for materials and equipment
2. Helps operators achieve greater CAPEX efficiency
3. Simplifies network operations
4. Allows service providers to use the latest technology for greater OPEX efficiency
5. Allows greater scalability and upgrades

7. Economic maintenance costs

8. Significantly extends the hardware network cycles
--

3.6 Chapter Conclusion

This chapter has analyzed and presented software-defined networking (SDN), which offers an architecture approach to network control in an intelligent, centralized, and "programmed" way, using software applications and open APIs. This allows global and consistent network management, regardless of the underlying network technology. This chapter also discusses network function virtualization (NFV), which is an effective way to virtualize network services, such as routers, firewalls, and load balancers, by grouping them as virtual machines (VM) standard hardware. This enables the creation and provision of on-demand network services without the need for additional hardware resources. The NFV and SDN are not dependent on each other but instead, have similarities and complement each other. Their point of resemblance is that they both rely on virtualization using network abstraction, and their point of difference is based on how each of them separates the functions and the abstract resources. SDN focuses on separating network transfer functions and network control functions for providing a manageable network with centralized programmability and management. Moreover, NFV focuses on the summary of the hardware's network functions by supporting the SDN, to which it provides an infrastructure for the execution of its software. The use of NFV and SDN allows the creation of more intelligent network architecture.

4 CHAPTER 3rd: SDN VIRTUAL NETWORK IMPLEMENTATION USING GNS3

The fourth Chapter consists of the practical part of this thesis and also its modest contribution. A functional testbed will be created from scratch to meet and practice SDN with real Openflow commands. Numerous tools and platforms are used, such as:

- Windows Operating System
- Linux Operating System
- The gns3 environment with VM and Docker integration
- The OpenDaylight Openflow controller
- VMware Player virtualization software
- Oracle Virtual Box virtualization software

The nature of most of these tools is Open Source and, even though documentation does exist, it is always difficult to install and combine them to produce a fully functional working environment. During this effort, we encountered many obstacles that were overcome with extensive search and testing. That is why we emphasize the step-by-step setup of our testbed rather than on developing complex scenarios. Nevertheless, basic traffic engineering scenarios are included.

4.1 Implementation Strategy

The two essential components are the OpenFlow switch and the controller to build a testbed for SDN with the OpenFlow protocol. Mininet simulator is often used to create topologies with both SDN and legacy devices, but we decided to use the gns3 environment with Docker integration. The gns3, together with the associated VM, is installed on a Windows host machine, and VMware is used to run the VM. As for the controller, we opt for the OpenDaylight, installed on a xUbuntu VM, running on the same Windows physical computer. A Linux gns3 appliance could also host the controller, but this would cause performance and stability issues.

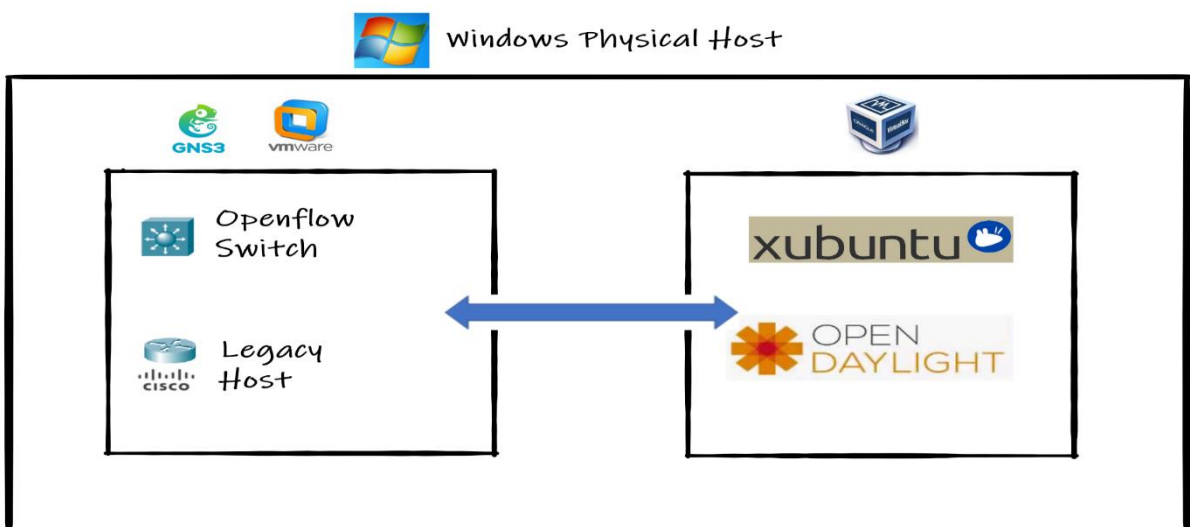


Figure 21: Implementation strategy in terms of software resources

A detailed description of the setup and configuration of each component follows. This description is necessary because sequential tests and the general experience have shown that the slightest modification can make the setup non-functional.

4.2 The Host Machine

The host machine is a Windows 7 x64, i5-3210M CPU with 8 GB RAM. The choice of the host operating system is based on the fact that it is considered to better accept the installation of the VM version of gns3. Some special settings should be made on the host machine (**Figure 22**):

- Windows Firewall is disabled.
- Windows User Account Control is disabled.
- No third-party security software is installed.
- IP configuration is assigned by a local DHCP server (the home modem-router).
- VMware 14 is installed. Although the 15 version of VMware Player can also be installed on Windows 7, the general experience shows that version 14 should be preferred.
- Oracle VirtualBox 6.1 is installed.
- Gns3 version 2.2.17, together with the VM, is installed.

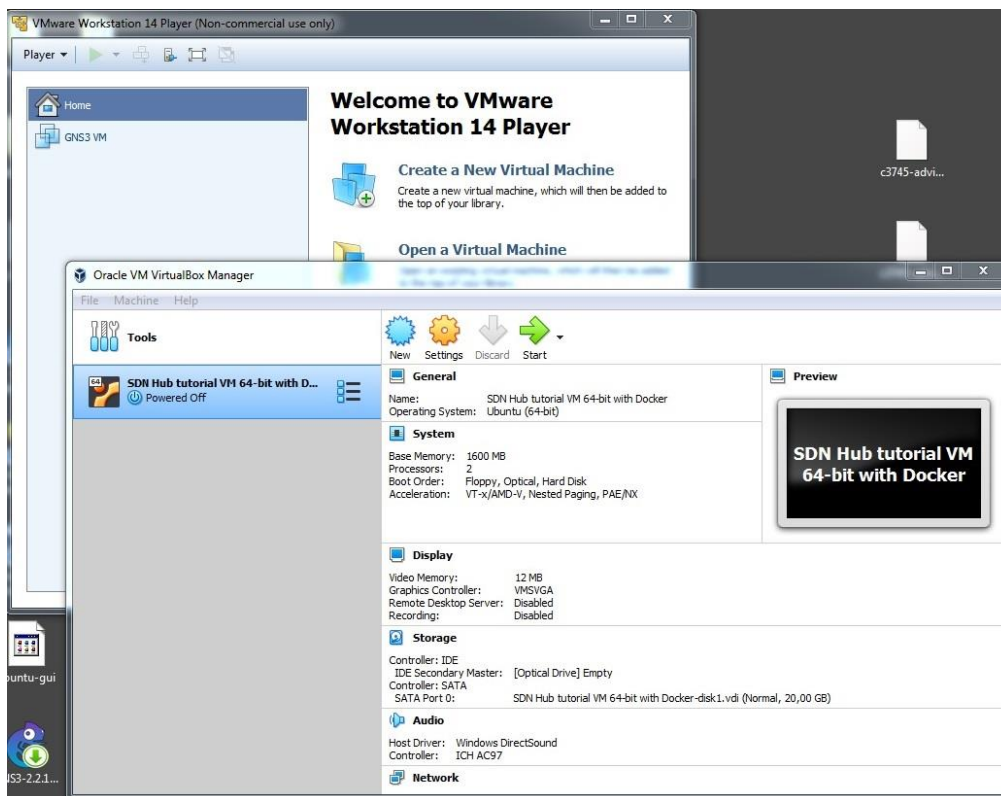


Figure 22: The host machine and the virtualization software installed

4.3 The GNS3 Environment

Once gns3 is installed, a set of additional settings are made.

➔ Connectivity to the physical world:

Connectivity to the physical world is assured through the gns3 cloud, with the addition of a third network adapter to the GNS3 VM. As **Figure 23** illustrates, the additional network card is set to be bridged with the physical one. This specific interface is to be used within the gns3 cloud to connect the openvswitch management interface (eth0).

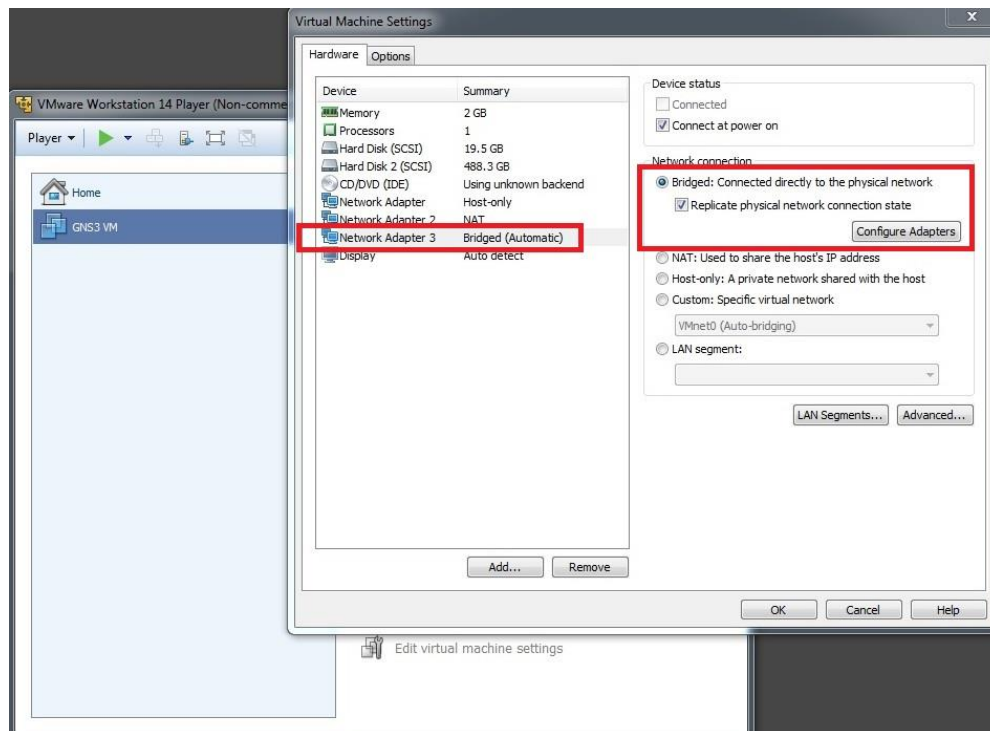


Figure 23: Addition on a network adapter to the gns3 VM

➔ OpenvSwitch Integration:

The OpenvSwitch with management interface appliance is downloaded by the gns3 marketplace (**Figure 24**) and imported to the Gns3 workspace. The new appliance is shown under the Switches family of devices in gns3.

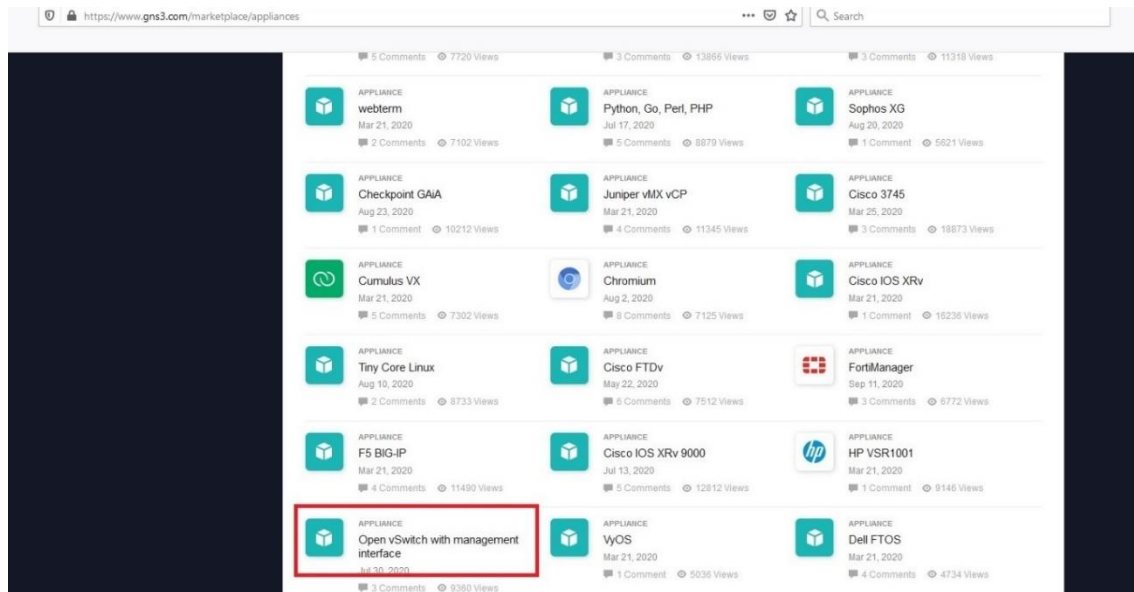


Figure 24: Download the appliance from the gns3 Marketplace

Figure 25 illustrates the new appliance together with the Gns3 VM, running in VMware Player 14.

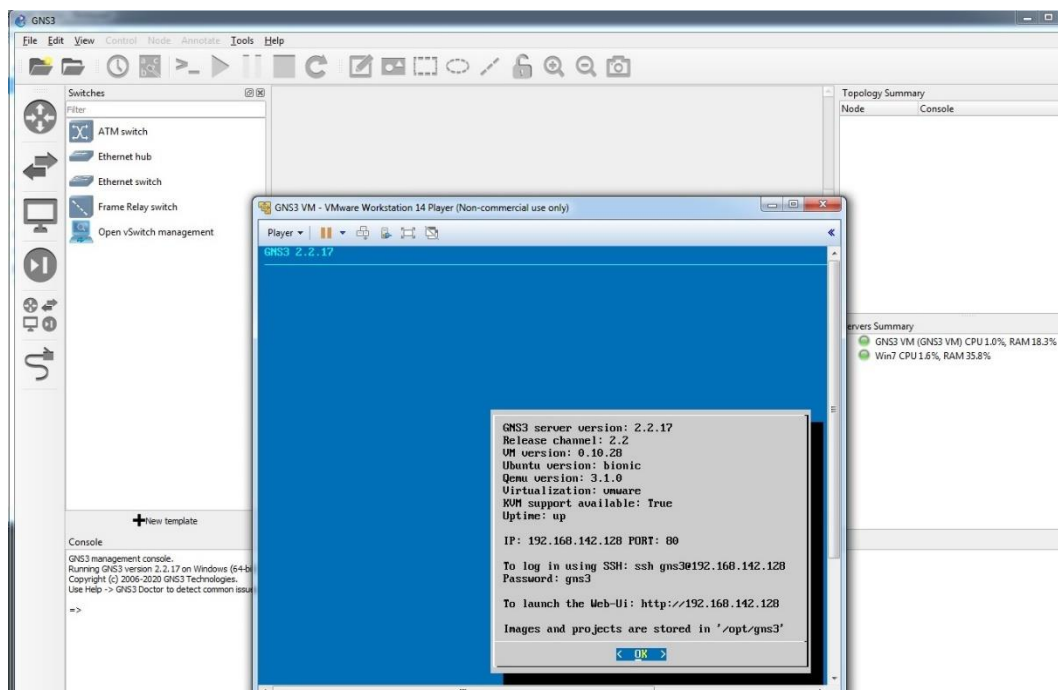


Figure 25: The Open vSwitch appliance in gns3

➔ Cisco IOS Router Integration:

Cisco IOS routers represent legacy hosts. To integrate such a device, an IOS image 3745 with Advanced IP Services is imported (**Figure 26**). The new device can be found under the Routers family of devices, as **Figure 27** illustrates.

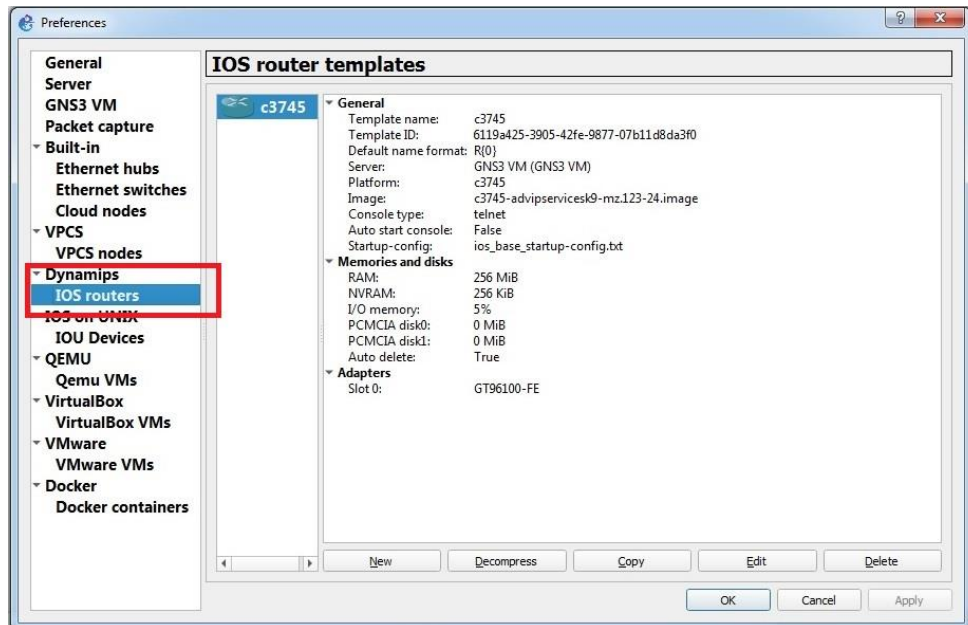


Figure 26: Import the Cisco IOS Image

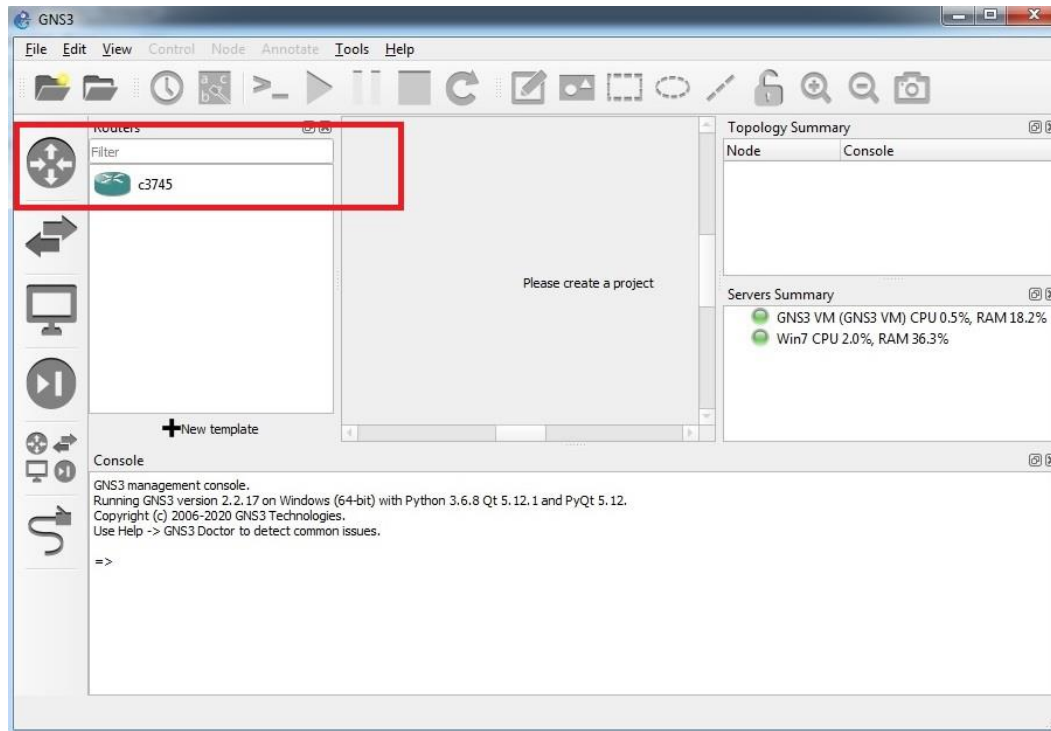


Figure 27: Cisco IOS router integration

4.4 The OpenDaylight Environment

To install the controller, the 'SDN Hub Tutorial VM' is downloaded and opened by VirtualBox. The network adapter is set to Bridged to assure Connectivity with the rest of the testbed and Internet connectivity (*Figure 28*).

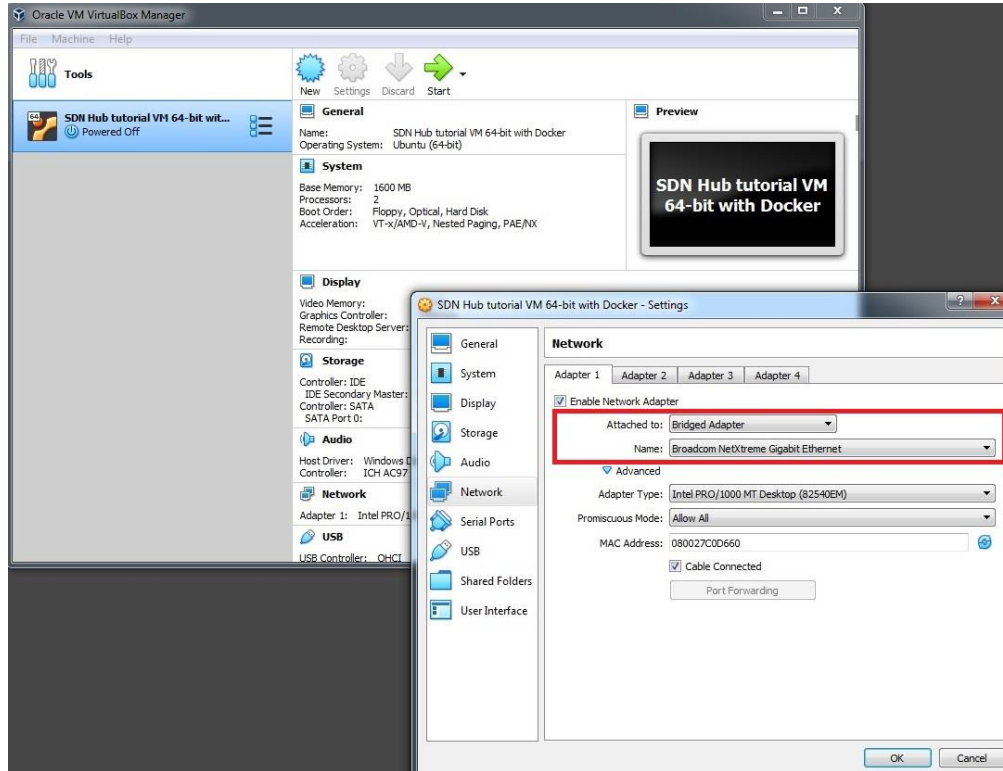


Figure 28: OpenDaylight machine network setup

Although the downloaded VM offers an integrated working environment, the Karaf-0.5.3-Boron version of Opendaylight is downloaded and installed from scratch on xUbuntu (*Figure 29*), together with the dlux graphical user interface (*Figure 30*).

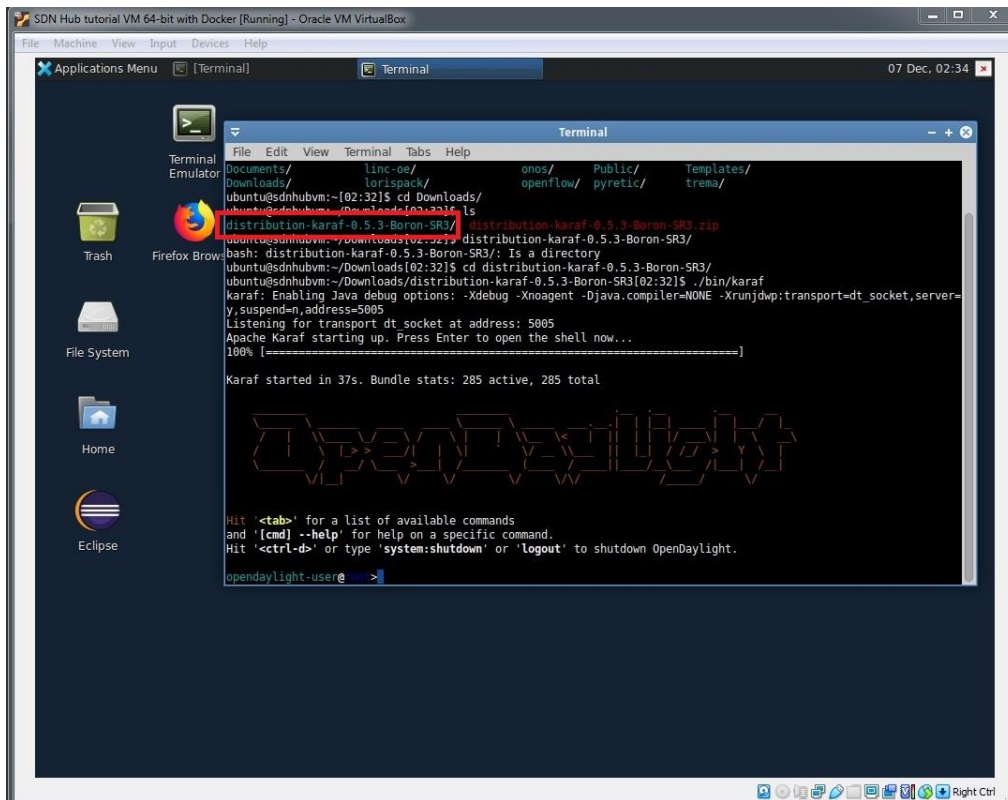


Figure 29: Installation on OpenDaylight in xUbuntu

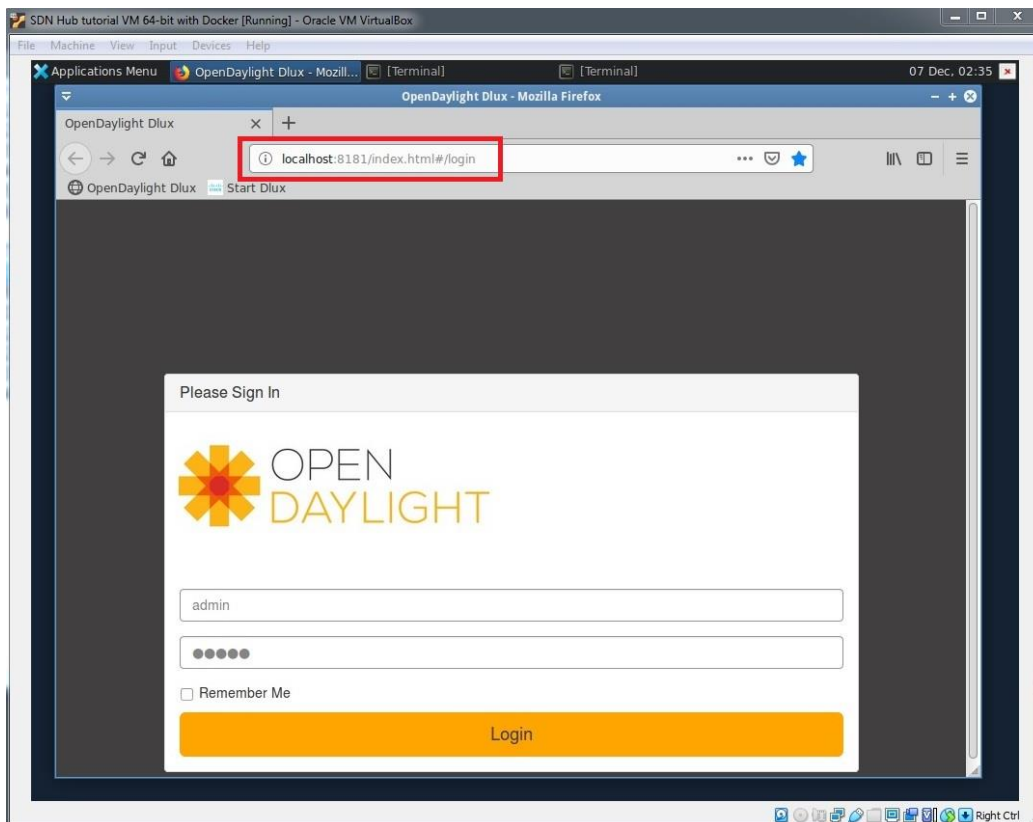


Figure 30: The OpenDaylight dlux interface

4.5 Basic SDN Topology

To build a simple, functional SDN scenario, we start with an OpenvSwitch and two Cisco routers. The Cloud is connected to the OpenvSwitch management interface with its third card, which corresponds to the extra adapter we added before to the gns3 VMware machine. The entire topology is shown in *Figure 31*.

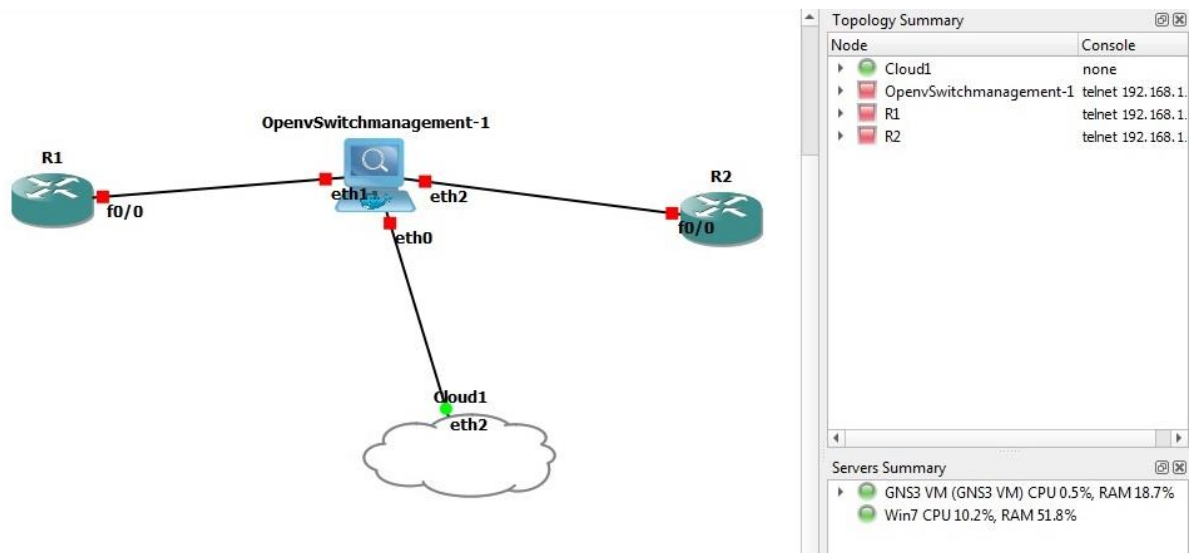


Figure 31: Basic topology with connections

We first configure the Cisco routers with IP addresses on their LAN interfaces, as *Figure 32* illustrates.

```

R1#
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface fa 0/0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#
*Mar  1 00:06:53.843: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:06:54.843: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#
    
```

Figure 32: Cisco router sample configuration

Then the OpenvSwitch is configured with an IP address of 192.168.1.94 on the management interface and with the socket to establish communication with OpenDaylight running on xUbuntu - VirtualBox (*Figure 33*). As this same Figure illustrates, the ping from the OpenDaylight machine to the OpenvSwitch is successful.

4.6 A Simple SDN Scenario

Once connectivity between the OpenvSwitch and the controller is assured, a simple traffic engineering scenario must be developed. The scenario is based on the previous initial topology, where two more Cisco routers are added, as *Figure 33* illustrates.

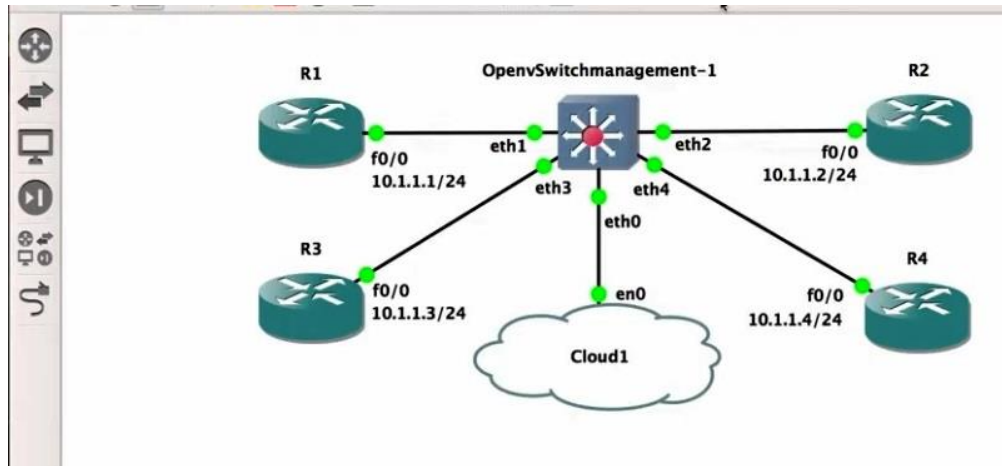


Figure 33: Simple traffic engineering scenario topology

The respective representation of the topology in OpenDaylight is shown in *Figure 34*.

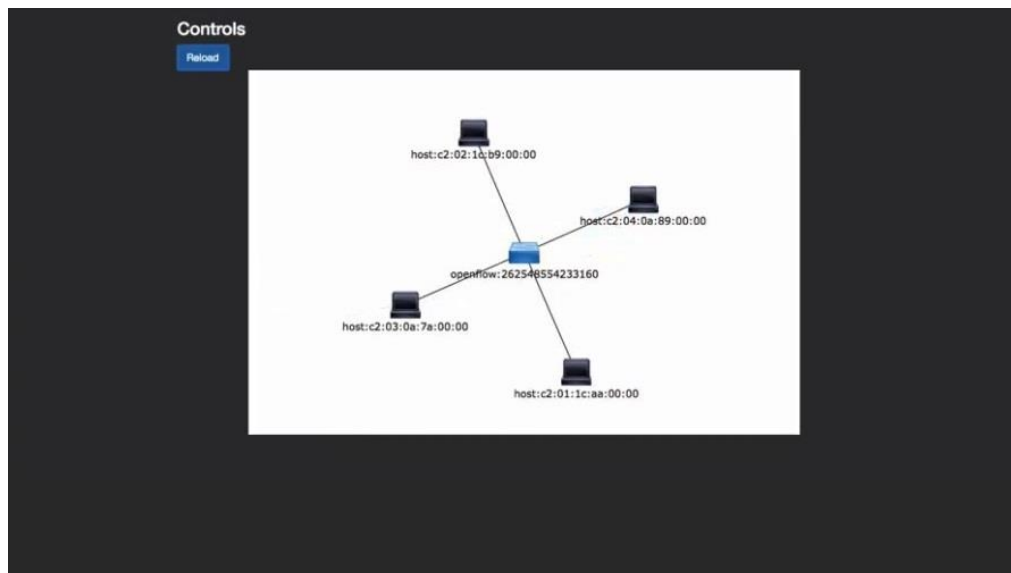


Figure 34: The topology in the OpenDaylight dlux interface

The traffic engineering scenario consists of creating a uni-directional patch panel, in software, from port eth1 to port eth2 of the OpenvSwitch. As a result, any traffic entering port eth1 is always directed to port eth2. The flow entry is described in *Figure 35*.

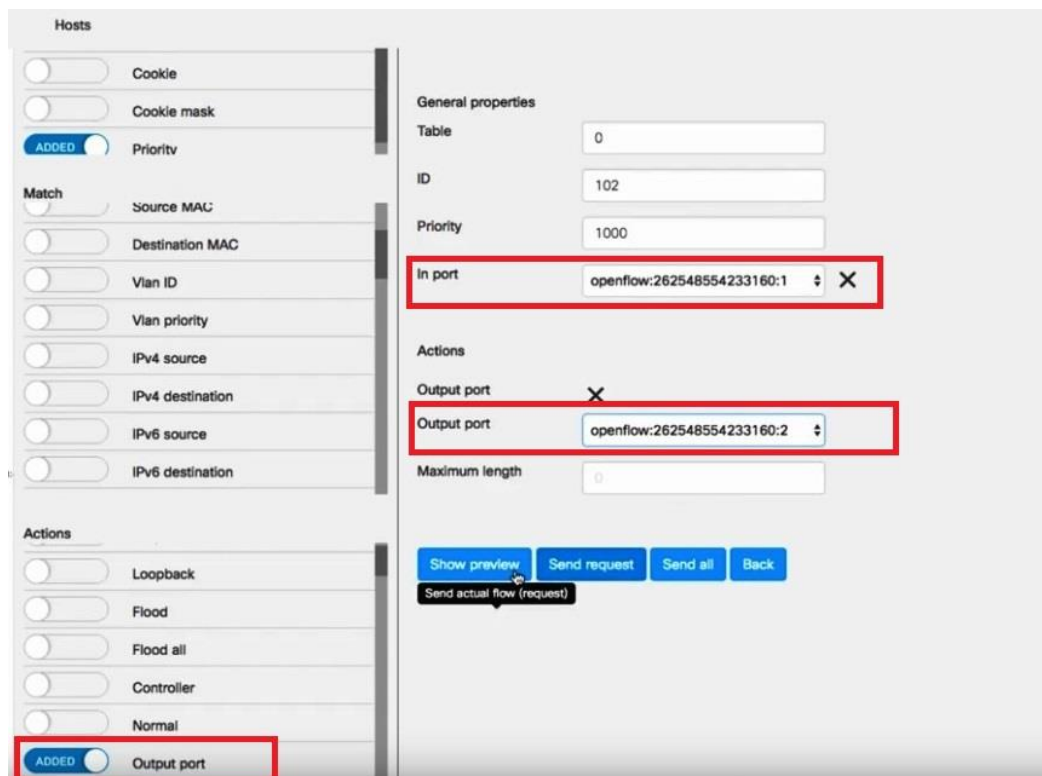


Figure 35: Uni-directional patch panel flow

Since the ingress traffic to port eth1 is directed only to port eth2, it is normal that R1 can only ping R2, and not the R3, R4 routers, as **Figure 36** illustrates.

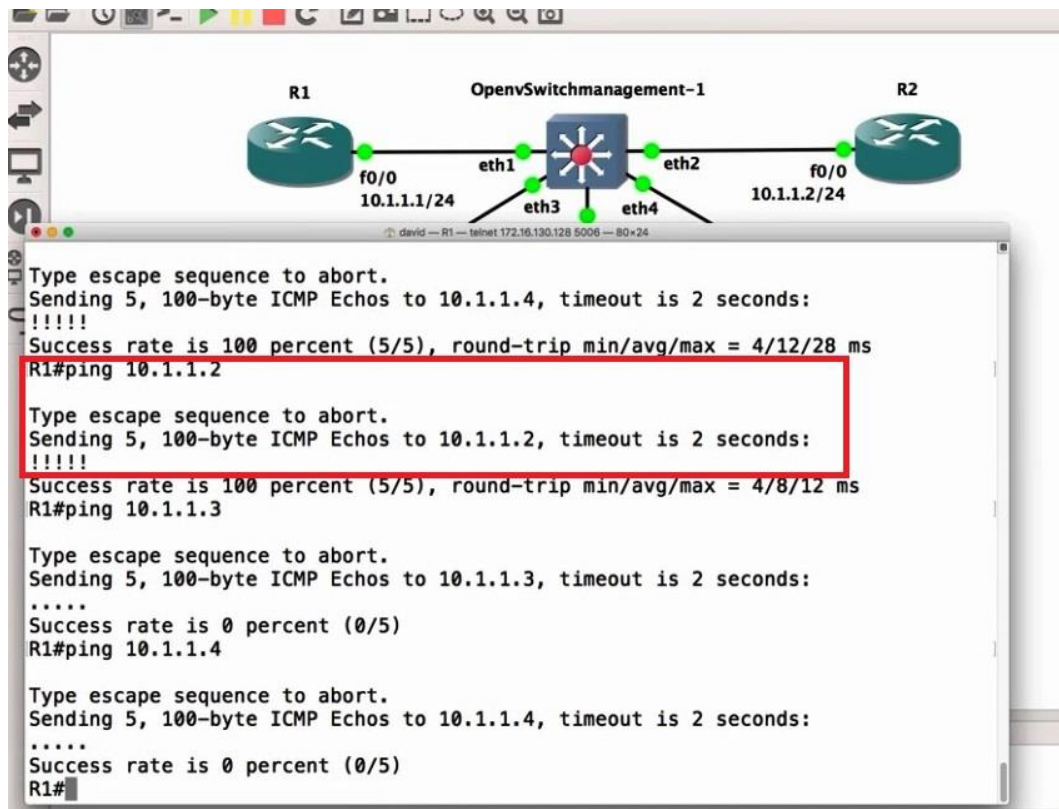


Figure 36: The result of the flow in the traffic

The idea behind this patch-panel scenario is the traffic control of Ethernet frames within a Local Area Network with granular security requirements, where a switch port can communicate with a specific port but not with other ports on the same switch. This concept is similar to the Private VLAN (PVLAN) feature on a legacy Cisco Catalyst switch. It reveals our initial position, stated at the beginning of this project: SDN is based on the fundamentals, giving practically unlimited possibilities in the management of the network infrastructure.

Virtualization technology helped develop Network virtualization, which is the key to SDN and cloud computing's current and future success. Virtualization allows running multiple virtual machines (VMs), which can be used for SDN and cloud computing components. In this sdn virtual network implementation, virtualization technology is used through VMware 14, installed together with Gns3 version 2.2.17 in host machine Windows 7 x 64 to deploy OpenFlow Switch and legacy host. Oracle VirtualBox 6.1, installed in the same host machine to contain OpenDaylight Environment, an SDN controller, and xubuntu Docker, is another example of a combination of Virtualization and SDN. The Connectivity to the physical world is assured through the Gns3 Cloud. The Cloud is connected to the OpenvSwitch management interface with its third card, which corresponds to the extra adapter added to the Gns3 VMware machine. This integration of Virtualization and SDN in the cloud environment offers high-quality service, such as programmable networks, easily partitionable and virtualizable, which is one of the required features for cloud computing where several competing entities share the network infrastructure.

5 CONCLUSIONS

The progress of technology in cloud computing is an essential basis for the new era of information technology. One of the remarkable results of this progress is the appearance of new generation networks. This is why this thesis tackled the subject of cloud computing and its contribution to the virtualization of next-generation networks. The research results within the framework of this thesis have shown that cloud computing plays a vital role in the virtualization and automation of networks. Indeed, the virtualization and automation tools of the deployment of network resources offered by cloud computing make the network smarter, more reliable, and improve its agility. This led to the emergence of SDN (Software Defined Networking) with an architecture that separates the control plane from the data plane to provide a programmable network. The SDN architecture is based on the OpenFlow protocol, and SDN controllers responsible for managing network flows on devices. NFV improves the agility of SDN services by quickly creating and moving virtual resources.

It is essential to point out that NFV and SDN have similarities and complement each other and are currently seen as solutions that have revolutionized network technology. They offer cost savings and reliable flexibility for network management and use. In this thesis's context, the controller used for the operation of the SDN implemented in the GNS3 platform is OpenDaylight. This made the network more agile and suitable for the traffic engineering scenario, which was developed to demonstrate the functionality of an intelligent, automatic, and programmable network based on cloud computing.

The scenario of the traffic control of Ethernet frames within a Local Area Network with granular security requirements, which was approached in this thesis, is a significant interest for the network companies or security services which are trying to migrate to more flexible networks, agile and programmable. This scenario made it possible to explore the performance and management of next-generation networks on the network's two operational planes, which are the data plan and the control plan.

As studied in this thesis, the data plan is based on a cloud computing platform that implements multi-tenant network virtualization for the proper deployment of NFV. After analyzing the performance of the main elements of virtualized networks, it has been proven that the Open vSwitch has shown near-optimal behavior and that an open-source cloud computing platform can be effectively used for the deployment of NFV.

In the scenario developed in this thesis, the SDN control plan is based on the OpenDaylight controller and effectively allows the piloting of specific data flows to the required NFV locations by creating a fully dynamic and adaptive chaining service. The result showed that the software-controlled network could self-adapt and dynamically reconfigure the NFV transfer to meet user needs.

Today a great use of multimedia services hence the need to adopt 5G as a future network that can easily support new services and meet their requirements. NFV and SDN, therefore, open the way to the 5G network, which must be intelligent, virtualized, and programmable in order to improve the performance of the services offered by companies. Network function virtualization and software-defined networking technologies are therefore at the heart of 5G networks.

The 5G SDN architecture is designed to meet the criteria of dynamism, reliability, and profitability. NFV is also a significant component for 5G infrastructure and network appliance virtualization; it offers a new way to build a network. It also allows the network to be divided by creating multiple virtual networks in a shared infrastructure to meet customer needs. Its role in 5G is also to enable the distributed Cloud to create flexible and programmable networks.

To achieve their objectives, SDN and NFV face specific requirements such as:

Control: SDN must standardize control interfaces, protect operating systems, control performance, and information maintenance. The NFV must control the provision in real-time, create network granularity policies and deal with virtualization - big data.

Reliability: SDN must ensure reliable Connectivity and rapid connection recovery while ensuring the transport and data network's security and flexibility. The NFV must ensure all 5G technologies' proper functioning by providing smooth Connectivity, endpoint virtualization, and high performance.

NFV and SDN's advent has brought several advantages in networking technology by encouraging the cost-effective transition from dedicated hardware appliances to a software approach. Despite these advantages, the development and deployment of the network on the NFV / SDN environment sometimes encounter obstacles due to the security issue, which is also one of their future challenges.

SDN still has a long way to go for its complete perfection to deal with security threats. As it was pointed out above, the SDN uses virtualization technology, which can sometimes lead to security issues related to the virtualization of network hypervisors and their isolation. These issues can be the security of SDN controllers, the transfer plan, and unauthorized access.

Compromise of the SDN Controller, which is the SDN architecture brain, can cause the entire network to fail. This is why next-generation networks must address the SDN Controller's security issues to avoid attacks that could compromise its operation. Because a compromised controller can transmit the wrong flow rules and cause security issues in the SDN architecture, another challenge facing next-generation networks is accessing an SDN architecture without authorization. This is often done through the SDN controller and can cause modification of network data or its components with the direct consequence of the entire network system's failure.

The most significant future challenges for NFV are security at the network function hypervisor level, virtual network functions, administrative and performance isolation, communication and functional interfaces.

One of the leading security concerns of NFV is the vulnerabilities of hypervisors. This is why next-generation networks must have an effective security system to avoid security risks regarding confidentiality, integrity, and availability. By the way, being responsible for creating VMs and controlling their operating systems, a hypervisor can experience hijacking attacks, an adversary can take his control and gain access to all VMs. In the worst case, he can even reach SDN controllers that are integrated with NFV technology with the consequence of destroying virtualized network functions.

Virtual network functions should be a top security priority of next-generation networks because they can sometimes suffer from sniffing, denial of service, and spoofing attacks if they are not well secured. By the way, in an NFV deployment, a malicious client can gain access through the network

to compromise virtual network functions and then disrupt operations by sending fraudulent instructions through the hypervisor.

When there is a lack of good isolation between virtual functions, VMs can be vulnerable to attack. This is why it is essential that next-generation networks can address the security issues of performance isolation in the NFV infrastructure by ensuring that compromises in one VM will not affect others. Because an isolation failure can allow side-channel attacks between virtual networks in an NFV shared infrastructure.

The challenges of interconnectivity between the end-to-end components of NFV are also significant problems that next-generation networks must provide effective solutions because security threats at the level of communication between NFV components and at the level of functional interfaces can compromise the proper functioning of the virtualized network infrastructure.

Today, there are several projects on the next-generation networks, particularly the 5G network, with the help of SDN and NFV to improve the quality of service and accelerate network slicing and its softwarization. SDN capabilities such as programmability, global visibility, and centralized control, and those of NFV such as flexibility, scalability, and adaptability have been discussed extensively in this thesis. With the main objective achieved, the scenario developed in this thesis about the next-generation cloud-based networks can be adopted to develop and analyze other more complex performances to meet companies' needs, thereby producing substantial socio-economic impacts.

Bibliography - References - Internet Resources

- [1] Rick F. Van der Lans in Data Virtualization for Business Intelligence Systems. Elsevier July 25, 2012 (pp 3-9)
- [2] Bauer E, Adams R. Reliability and Availability of Cloud Computing, First Edition. John Wiley & Sons; 2012 Jul 20 (pp 16-28)
- [3] Barrie Sosinsky. Cloud Computing Bible. Wiley Publishing, Inc; December 3, 2010 (pp 3-4)
- [4] Portnoy M. Virtualization essentials. John Wiley & Sons; 2012 Mar 29
- [5] Ronald L Krutz and Russell Dean Vines. Cloud Security: A Comprehensive Guide to Secure Cloud Computing. Wiley Publishing, Inc, 2010 (pp 9-10)
- [6] Figure1. Three lay of cloud computing service, URL <https://www.f5.com/services/resources/white-papers/controlling-the-cloud-requirements-for-cloud-computing>
- [7] Anjing Wang, Mohan Iyer, Rudra Dutta, George N. Rouskas and Ilia Baldine, “Network virtualization: technologies, perspectives, and frontiers,” Journal of Lightwave Technology, vol. 31, no. 4, pp. (pp 10-11), 2013.
- [8] Nadeau T, Gray K. SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies. Publisher: O'Reilly Media, August. 2013.(pp7-8)
- [9] “Network functions virtualization: An introduction, benefits, enablers, challenges & call for action.” https://portal.etsi.org/nfv/nfv_white_paper.pdf
- [10] 7 Advantages of Software Defined Networking, August 08, 2017
URL <https://imagineNEXT.ingrammicro.com/data-center/7-advantages-of-software-defined-networking>, Retrieved [Mar.6, 2020]
- [11] What is Docker, URL <https://docker-curriculum.com/#what-is-docker->, Retrieved [Mar.6, 2020]
- [12] Production Quality, Multilayer Open Virtual Switch URL <https://www.openvswitch.org/>, Retrieved [Mar.6, 2020]
- [13] VMware Workstation, URL <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>, Retrieved [Mar.6, 2020]
- [14] The Top Five Network Problems Solved by SDN, URL <https://blog.silver-peak.com/the-top-five-network-problems-solved-by-sdn/>, Retrieved [Mar.6, 2020]
- [16] T. Anderson et al., “Overcoming the Internet Impasse through Virtualization,” Computer, vol. 38, no. 4, 2005, (pp. 34–41)
- [17] Bruno Medeiros de Barros , Marcos Antonio Simplicio Jr., Tereza Cristina Melo de Brito Carvalho, Marco Antonio Torrez Rojas, Fernando Frota Redígolo, Ewerton Rodrigues Andrade, Dino

Raffael Cristofoleti Magri .Applying Software-defined Networks to Cloud Computing, May 2015, url <https://www.researchgate.net/publication/283275261>

[18] “CORD: Re-inventing Central Offices for Efficiency and Agility,” Accessed December 2018. [Online]. Available: <https://opencord.org/>

[19] GigaSpaces, “2015cloudify,” Accessed April 2019. [Online]. Available: <http://cloudify.co/>

[20] L. Foundation, “ONAP–Open Network Automation Platform,” Accessed May 2019. [Online]. Available: <https://www.onap.org/>

[21] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, “A survey on service function chaining,” *Journal of Network and Computer Applications*, vol. 75, pp. 138–155, 2016

[22] A. Francescon, G. Baggio, R. Fedrizzi, R. Ferrusy, I. G. B. Yahiaz, and R. Riggio, “X–MANO: Cross–domain management and orchestration of network services,” in *2017 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2017, pp. 1–5.

[23] Fraunhofer, “Open Baton: An open source reference implementation of the ETSI Network Function Virtualization MANO specification,” Accessed February 2019. [Online]. Available: <http://openbaton.github.io/>

[24] NTT, “Gohan-REST-based api server to evolve your cloud service very rapidly,” accessed 2019. [Online]. Available: <http://gohan.cloudwan.io/>

[25] Melinda Varian. *Vm and the vm community: Past, present, and future*. office of computing and information technology. Technical report, Princeton University, Princeton, NJ, 1997.

[26] Figure 2, History and evolution of virtualization <https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=9236&context=theses>

[27] Aaron Weiss. *Computing in the clouds*. netWorker, 11:16–25, December 2007

[28] VMware, *Understanding Full Virtualization, Paravirtualization, and Hardware Assist*. Publisher : VMware, Latest Version : March 11, 2008

[29] Figure 15 <https://www.ibm.com/blogs/cloud-computing/2014/02/17/what-is-platform-as-a-service-paas-2/>

[30] Eric Simmon, *Evaluation of Cloud Computing Services Based on NIST 800-145*, Special Publication 500-322 Draft – 20170427, by the NIST

[31] J. Wu, L. Ping, X. Ge, Y. Wang, and J. Fu, "Cloud storage as the infrastructure of cloud computing," In *Proc. International Conference on Intelligent Computing and Cognitive Informatics (ICICCI 2010)*, pp. 380-383, 2010.

[32] *OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05)* October 14, 2013 Open Networking Foundation TS-012

[33] Ola Salman, Imad Elhaji, Ayman Kayssi and Ali Chehab, *SDN Controllers: A Comparative Study*, April 2016, Conference: 2016 18th Mediterranean Electrotechnical Conference (MELECON)

- [34] A. U. REHMAN, RUI. L. AGUIAR, and JOÃO PAULO BARRACA , Network Functions Virtualization: The Long Road to Commercial Deployments, May 2019,Article in IEEE Access PP(99):1-1
- [35] Open Networking Foundation, Relationship of SDN and NFV Issue, 1 October 2015
- [36] IaaS, PaaS and SaaS OpenSource, URL <https://www.linuxlinks.com/iaas/>, Retrieved [13 of June 2020]
- [37] ESXi hypervisor,URL <https://geek-university.com/vmware-esxi/what-is-vmware-esxi>, Retrieved [11 of January 2021]
- [38] Xen hypervisor, <http://www-archive.xenproject.org/products/xenhyp.html>, Retrieved [11 of January 2021]
- [39] Hyper-V, <https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/>, Retrieved [11 of January 2021]

