



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

**ΠΛΑΤΦΟΡΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
ΕΡΓΑΣΤΗΡΙΑΚΩΝ ΜΑΘΗΜΑΤΩΝ ΜΕ ΧΡΗΣΗ ΤΕΧΝΟΛΟΓΙΩΝ
ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ**

**Φοιτητής: Νίκος Καφταντζής
ΑΜ: 50344391**

Επιβλέπων Καθηγητής

Πατρικάκης Ζ. Χαράλαμπος

Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, Σεπτέμβριος 2020



**UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING**

Diploma Thesis

A platform for monitoring and management of laboratory courses, using web technologies

**Student: Nick Kaftantzis
Registration Number:50344391**

Supervisor

Charalampos Z. Patrikakis, Dr. Ing

Professor

ATHENS-EGALEO, September 2020

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Νίκος Καφταντζής, Σεπτέμβρης, 2020

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 ΠΕΡΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ ΚΑΙ ΛΟΓΟΚΛΟΠΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπόγραφα ότι η παρούσα εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα αποκλειστικά και ότι είμαι ο αποκλειστικός συγγραφέας του κειμένου της.

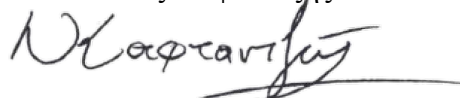
Η εργασία μου δεν προσβάλλει οποιασδήποτε μορφής δικαιώματα πνευματικής ιδιοκτησίας, προσωπικότητας ή προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής ή λογοκλοπής.

Κάθε βοήθεια που έλαβα για την ολοκλήρωση της εργασίας είναι αναγνωρισμένη και αναφέρεται λεπτομερώς στο κείμενό της. Ειδικότερα, έχω αναφέρει ευδιάκριτα μέσα στο κείμενο και με την κατάλληλη παραπομπή όλες τις πηγές δεδομένων, κώδικα προγραμματισμού Η/Υ, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών που χρησιμοποιήθηκαν, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης, και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Επιπλέον, όλες οι πηγές που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης κατά τα διεθνή πρότυπα.

Τέλος δηλώνω ενυπόγραφα ότι αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της είναι προϊόν λογοκλοπής.

Ημερομηνία

Νικόλαος Καφταντζής



(Υπογραφή)

ACKNOWLEDGEMENTS

I would like to express my special thanks to all the people who contributed to this graduated work. I would like to express sincere gratitude to my supervisor Prof. Charalabos Patrikakis for the support that he continuously provided during my research work and thesis writing. Besides my supervisor, I also would like to thank Dr. Dimitris Kogias and Michalis Xevgenis for their guidance and insightful comments.

Περίληψη

Τα τελευταία χρόνια έχουν αναπτυχθεί αρκετές πλατφόρμες προσομοίωσης δικτύων, οι οποίες έχουν βοηθήσει στην εξέλιξη των τρόπων και μέσων εκμάθησης στην εκπαίδευση. Συγκεκριμένα, η χρήση από ένα εργαστήριο δικτύων μιας αντίστοιχης πλατφόρμας εξομοίωσης της δικτυακής λειτουργίας θα βοηθήσει στην αύξηση της ποιότητας και αποτελεσματικότητας της προσφερόμενης εκπαιδευτικής λειτουργίας. Όμως, τέτοια εργαλεία παρουσιάζουν σημαντική πολυπλοκότητα στην εγκατάσταση και τη συντήρησή τους, τα οποία πρέπει να αντιμετωπιστούν για την κατάλληλη εφαρμογή τους.

Η διπλωματική αυτή διαπραγματεύεται την χρήση μιας διαφορετικής πλατφόρμας προσομοίωσης για τη δημιουργία και μελέτη τοπολογιών δικτύων με σκοπό την εκμάθηση χρήσης πρωτοκόλλων δυναμικής δρομολόγησης. Συγκεκριμένα, θα χρησιμοποιηθεί η πλατφόρμα EVE-NG και, μέσω των δυνατοτήτων που μας προσφέρει, θα αναπτυχθούν 5 τοπολογίες επικοινωνίας δικτύων.

Μάλιστα, στα πλαίσια των προαναφερθέντων, η διπλωματική εστιάζει στην επικοινωνία των δρομολογητών μεταξύ τους και στην χρήση συγκεκριμένων δυναμικών πρωτοκόλλων, όπου χρησιμοποιούν αλγόριθμους για την επιλογή των βέλτιστων διαδρομών επικοινωνίας μεταξύ των δρομολογητών του δικτύου, με σκοπό να φανούν στους χρήστες τα πλεονεκτήματα της χρήσης των πρωτοκόλλων αυτών μέσα από έναν εξομοιωτή.

Λέξεις – κλειδιά

EVE-NG, πλατφόρμα προσομοίωσης, Διαδίκτυο, Δρομολόγηση, Δυναμική δρομολόγηση, IP πρωτόκολλο, Επίπεδο Δικτύου, Junos

Abstract

As networking systems continue to evolve in complexity, more and more tools are emerging in order to better understand the emerging new technologies related to networking technologies. The computer science sector from its inception is a field of continuous changes and upgrades of existing technologies. Every day we see new protocols being incorporated and new ideas being proposed with the sole purpose of the most efficient use of the internet and its possibilities. Due to their complexity and need for backward compatibility, many challenges arise from developing, implementing and testing these technologies.

This is where network emulation enters the stage. And more specifically in the field of education where there is a great need for online classes which will offer profitable practical knowledge and an extensive theoretical basis, which can be achieved with the use of a network emulation tool.

However, these tools are challenging to setup and maintain, in the laboratory environments of universities. In addition, for network research it is very costly to deploy a complete testbed topology containing multiple networked computers, switches, routers and data links to verify a certain network protocol or to validate proof of work over a prototype topology. The network simulators and emulators, in these circumstances, can save a lot of money and time in accomplishing this task. Network emulators particularly come very handy in allowing the network designers to test new protocols or to alter the existing ones in a controlled and secured environment. For this reason, emulated platforms are researched and studied upon.

In this thesis, the goal is to study and use these emulators techniques with the purpose to use their tools to create easy, understandable and manageable exercises for universities labs. The idea is to support laboratory environments in universities using an emulation platform which will be running on a virtual machine in a Cloud service provider, thus eliminating the need for powerful hardware to be installed in the computers. The scenario is to use the *EVE-NG* emulation platform to create network topologies courses, with the purpose to study the use of dynamic routing protocols in order to comprehend how connection between router and Layer 3 routing occurs. In the proposed scenario the only requirement is the connection of the students to a specific IP address where the web UI of the platform will be visible. From there, the students will be able to enter the platform and start to interact with the labs.

Keywords

EVE-NG, Internet, OSI Layers, Routing, Dynamic Routing Protocols, OSPF, BGP, Junos, Network Layer, IP Protocol, Emulation platforms

Περιεχόμενα

ACKNOWLEDGEMENTS	5
List of Tables.....	10
List of Figures	10
1 Introduction	12
1.1 Personal motivation	12
1.2 Scope and Goals	12
1.3 Thesis Structure.....	13
2 Chapter 1: Platforms for Management Laboratories Courses.....	14
2.1 Technological background: The Internet protocol architecture (A top down approach)	14
2.2 Platforms to Emulate Networks Topologies	19
2.2.1 The GNS3 Platform.....	19
2.2.2 The Packet Tracer Platform	20
2.2.3 The EVE-NG Platform	21
2.2.4 Comparative Presentation.....	22
2.3 Introduction to EVE-NG Platform	23
2.3.1 The role of GRNET	24
2.3.2 The OKEANOS cloud and myNetlab project	25
3 Chapter 2: Creation of the networking scenarios on the EVE-NG platform	25
3.1 The purpose of this scenarios	25
3.2 An overview of IP addressing and IP routing	26
3.2.1 The IP Datagram Structure	26
3.2.2 IP Addressing Classful, Classless & Subnetting	28
3.2.3 IP Routing Static & Dynamic	32
3.3 Static Routing	32
3.3.1 Analysis of Static Routing	32
3.4 Dynamic Routing	33
3.4.1 Analysis of Dynamic Routing (Distance Vector vs Link State algorithm vs Path vector)	36
3.4.2 RIPv1 and RIPv2 Protocol	42
3.4.3 OSPF Protocol	43
3.4.4 IS-IS Protocol.....	44
3.4.5 BGP, EBGP and IBGP.....	45
3.4.6 Advantages and disadvantages	47
3.5 The JUNOS OS and its Characteristics	48
3.5.1 Configuring a JUNOS router.....	48
3.5.2 Cisco vs JUNOS.....	50
3.6 Description of the experiments	51
3.6.1 Static Routing.....	51
3.6.2 OSPF	51
3.6.3 RIPv2.....	51
3.6.4 IS-IS.....	51
3.6.5 BGP.....	51
4 Chapter 3 : Development of Laboratory Exercises Using JUNOS	52
4.1 Static Route Topology	52

4.2	RIPv2 Topology.....	55
4.3	3.3) OSPF Topology.....	59
4.4	Intermediate System-Intermediate System (IS-IS) Topology	63
4.5	Border Gateway Protocol (BGP) Topology.....	67
5	Chapter 4 : Conclusion and Future work	76
6	Chapter 5: References	77

List of Tables

TABLE 1: PLATFORMS COMPARISON	23
TABLE 2: DEFAULT MASK VALUE	30
TABLE 3: CLASSFULL NETWORK MASKS	30
TABLE 4: ROUTING TABLE	33
TABLE 5: ROUTING PROTOCOLS CLASSIFICATION	37
TABLE 6: DISTANCE-VECTOR VS LINK-STATE	38
TABLE 7: COMPARISON OF DYNAMIC ROUTING PROTOCOLS	41
TABLE 8: RIPv1 AND RIPv2 COMPARISON.....	42
TABLE 9: IBGP AND EBGP DIFFERENCES	46
TABLE 10: STATIC ROUTING VERSUS DYNAMIC ROUTING	47
TABLE 11: DEFAULT ACTIONS OF ROUTING PROTOCOLS.....	57

List of Figures

FIGURE 1: OSI MODEL LAYERS.....	15
FIGURE 2: OSI LAYERS FUNCTIONS AND PROTOCOLS.....	16
FIGURE 3: GNS3 PLATFORM[7]	19
FIGURE 4: PACKET TRACER PLATFORM[8]	20
FIGURE 5: EVE-NG[10].....	23
FIGURE 6: THE IPv4 DATAGRAM FORMAT	26
FIGURE 7: IPv4 CLASSFUL ADDRESSING	29
FIGURE 8: ROUTING PROTOCOLS EVOLUTION.....	34
FIGURE 9: DYNAMIC ROUTING PROTOCOLS CLASSIFICATION	35
FIGURE 10: DIFFERENCES BETWEEN IGP & EGP.....	35
FIGURE 11: AN ILLUSTRATION OF PATH-VECTOR PROTOCOL	38
FIGURE 12: IS-IS ROUTER'S LEVELS	44
FIGURE 13: EBGP & IBGP SESSIONS	45
FIGURE 14: LOGIN IN A JUNOS DEVICE.....	48
FIGURE 15: THE UNIX BSD SHELL.....	48
FIGURE 16: OPERATIONAL MODE	49
FIGURE 17: CONFIGURATION MODE.....	49
FIGURE 18: JUNOS HIERARCHY LEVELS	49
FIGURE 19: STATIC TOPOLOGY	52
FIGURE 20: NETWORK AREAS	53
FIGURE 21: SUCCESSFUL PING FROM OLIVE2 TO OLIVE	54
FIGURE 22: RIPv2 TOPOLOGY	55
FIGURE 23: JUNOS POLICY.....	56
FIGURE 24: PING FROM OLIVE 5 TO OLIVE1	59
FIGURE 25: OSPF TOPOLOGY	59
FIGURE 26: OSPF VIRTUAL LINK.....	60
FIGURE 27: PING OLIVE FROM OLIVE	63
FIGURE 28: IS-IS TOPOLOGY	64
FIGURE 29: PING OLIVE1 FROM OLIVE7	66

FIGURE 30: BGP NETWORK TOPOLOGY	67
FIGURE 31: PING FROM OLIVE5 TO OLIVE12	75

1 Introduction

As technological advances continue to infiltrate in different parts of our everyday lives it becomes very difficult to not be able to use the possibilities and advantages offered to you by technology. Especially over the past four decades, network technology has become a big aspect of today's Internet. Network engineers when designing a network architecture, they should keep in mind several parameters (bandwidth, cabling, ports needed etc.) to consider where they may have implications such as. Teaching and approaching networking concepts requires the use of tools that emulate network topologies in order to help students comprehend the theoretical part of networking, algorithms and conduct experimental activities, such as installation, configuration and problem-solving techniques. [1].

Although virtualization was firstly developed to make easier the management of resources utilization, now has become a vital part of the Internet. In general, there are mainly two optimal paths one can take when you want to create virtualized networks environments: *simulators* and *emulators*. *Simulators* on one hand can mimic the basic functions of network devices but are not able to provide all the functions of the devices, whereas *emulators* have the ability to behave like a real network would act. In addition, in an emulated network, a virtual computer with various operation systems, such as *Windows* or *Linux*, can be connected to network devices, routers and switches, in the created virtual network.

Thus, the selection of a network emulator over a network simulation platform is preferable when you want to recreate the real-world effects of network inside a controlled area.

1.1 Personal motivation

My main motivation is the idea of creating a series of laboratory exercises using an emulated platform for the purpose of a better and more in-depth learning experience in university laboratories that aim to explain concepts such as protocols, dynamic routing and algorithms that are part of our everyday use of the Web.

My secondary motivation is the ability of recreating and configuring a network topology through an emulated environment where it is possible to recreate the same conditions as a real network environment and be able to understand and grasp the fundamentals rules of which they operate.

1.2 Scope and Goals

The objective of this thesis is to conduct the appropriate research and study in order to familiarize with the technologies revolving around networking and routing, such as dynamic routing, network emulators and the protocols that are operating in the network layer.

The final goal is the creation of a series of exercises were the student will be able to start and learn about basic routing information such as static routing, longest prefix number up until the

usage of dynamic protocols, their differences, which algorithms they utilize, which scenario is more suited for and their characteristics

1.3 Thesis Structure

In Chapter 1 there will be an introduction to the use of network emulators (e.g., *EVE-NG*, *GNS3*) in education and the role they can play, for a better and more effective learning environment, especially for students and pre-academic to comprehend how communication through Internet works. In addition, there will be a detailed analysis of the architecture of the Internet and its role.

Chapter 2 presents the study that was performed in order to understand in-depth how the IP protocol and IP-addressing works, how the routers communicate by using various methods, such as dynamic and static routing and the different routing protocols that can be used to achieve it. Lastly the *Junos OS* is presented that will run on all the routers that will be used in the designed laboratory courses.

The experimental part will be presented in Chapter 3, and the outcome will take the form of a series of network topologies created using the *EVE-NG* platform designed for academic laboratories, where the students will be able to see step-by-step the configuration of a route through command line commands. A lot of tests were performed during the development of these exercises to ensure the proper functionality of the topologies, while also making sure those functionalities are well fitted for the purpose of the thesis and the education background of the students.

2 Chapter 1: Platforms for Management Laboratories Courses

Over the last decade, with the rapid development of the Internet, several platforms for network management have been developed. As the evolution of the internet continues daily at a rapid pace with new technologies appearing vigorously and without seeming to diminish at all in the near future, there is a need for educational workshops which will provide the necessary knowledge to the student. Given this practical relevance, it seems mandatory the education and preparation of students both in theory and in practice with the use of emulation tools within the scope of the university laboratories. Therefore, the use emulated network topologies for teaching in higher education classes and training is important in order to comprehend basic principles and protocols that encircle the Internet.

With that in mind the first step is to explain and analyze the architecture of the Internet and specifically the *Open Systems Interconnection (OSI)* model which is a reference model that standardizes the communication functions of a computing system over any network structure and technology.

2.1 Technological background: The Internet protocol architecture (A top down approach)

The OSI model is essence an easy way to understand data communications when two networked systems are interconnected. It separates the communications processes into seven distinctive parts or layers. Each layer is completely autonomous and at the same time form the basis for the layers above it and respectively offer new functions for the layers below it.

It was first conceived in the 1970's when computer networking was taking off and developed specific developed teams from mostly computer and telecommunication corporations. In 1983 the OSI was conceived and was originally intended to be a detailed documentation for how interfaces are going to connect. Instead, it came to be a common reference architecture model with which anyone could build and configure detailed network interfaces, which in turn, had the potential to be the standard for governing the transmission of data packets. Therefore, the OSI model was and officially was created and accepted as the international standard by the *International Organization for Standardization (ISO)*. The model uses layers in order to help present a visual and detailed description of what is a networking system. It is helpful for network managers for narrowing down problems as well as computer programmers [2].

The networking model of OSI, in general presents a common way to segment computer networking operations into multiple layers. Each layer depends on the layers below it to provide supporting capabilities and offers support to the layers above it. Such a model of layered functionality is, also, called a “protocol stack” or “protocol suite”. Protocols are operating in the hardware level or in the software level and some are operating in both. The first 3 layers are mainly operating in the hardware and the other 4 layers are mostly operating in software.

Primarily the OSI model is more of a reference model that standardize the requirements for the protocols between two computers. This approach offers the ability for various network components from different manufactures to operate smoothly and with no communication.[3].

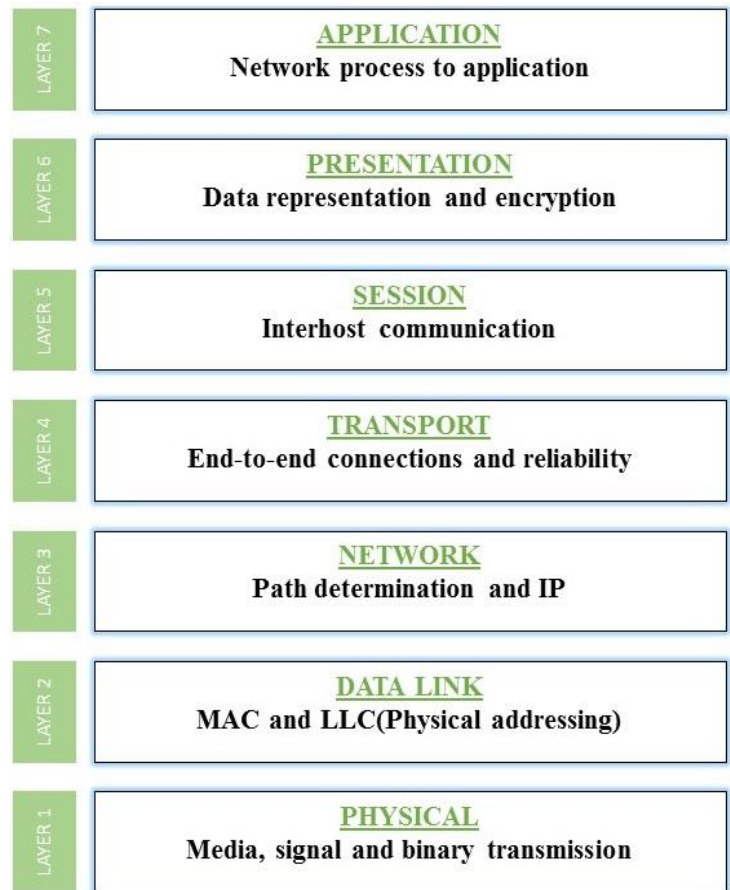


Figure 1: OSI model Layers

In the architecture illustrated in *Figure 1*, each level uses what is below it and offers its services to what is above it. Therefore, each message between two computers, there will have be a continuous flow of data packets through the layers from the source computer, across the network and then up through the layers in the receiving end computer.

Its role is not only to be an instructive guide to help students and newcomers to understand more easily how the internet operates, but its true purpose is the operability and connectivity of different communication systems with the use of the same protocols [4].

The OSI model was created for the following purposes:

- To standardize networking protocols and to make communication between networking devices across the Internet possible.
- To build a unified platform were developers and manufactures have the opportunity to create products and services with the ability to communicate over the network.
- To assist network administrators by splitting the data processes into smaller more manageable segments. Thus, making it are easier to understand, manage and troubleshoot. Therefore, the layered approach offers an easier approach to troubleshoot the faulted device which is operating in a layer.

The 7 layers of the OSI model were not picked at random. The number of levels in the model are not purposeless, but it is based on some specifications that have been applied and thus completed the model consisting of 7 levels. The principles that were used in order to create final seven layers can be described as follows [5]:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that architecture does not become unwieldy.

Using these 5 principles as guideline the OSI model was constructed with the intend to provide a guideline and to ensure, that the communication companies and institutes will be constructed

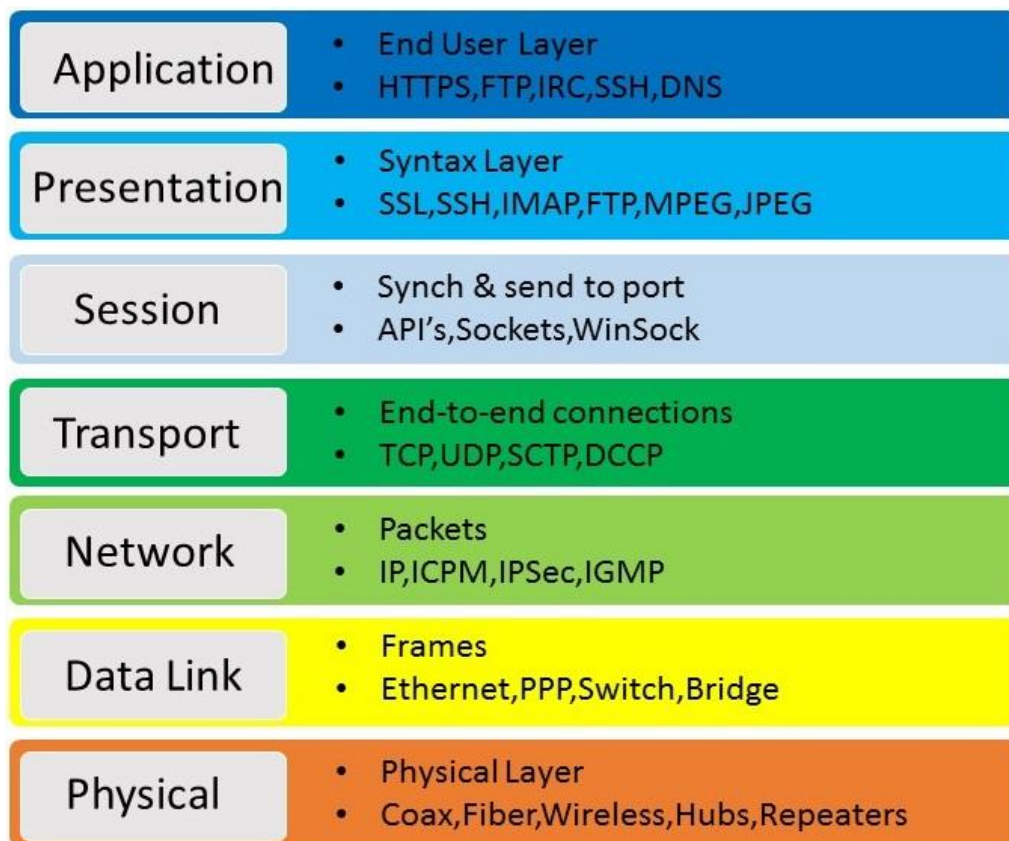


Figure 2: OSI Layers Functions and Protocols

according to this model in order to be compatible to communicate with other systems. If the OSI model had more or less layers, it wouldn't mean that the protocols or software created would have extra or less functionality from what they have today.

The OSI model consists of seven layers with the application layer, which is the layer nearer the end user, on the other hand the physical layer, were the data transfer occurs with the use of a

transmission channel [2], [5]. Even though we will follow a top down approach in the presentation, it should be noted that the numbering of the layers follows the reverse order.

- **Layer 7. Application Layer:** The seventh layer in OSI network model is the *Application Layer*. Traffic data in most cases is generated in the Application Layer. Often is a web request from a browser that is using the *Hypertext Transfer Protocol (HTTP)* protocol, a command from telnet protocol, a file download request from *File Transfer Protocol (FTP)* protocol. The purpose of this layer is to provide entrance to services into the communication system and send packets. Some protocols that are running on this layer are *FTP, Domain Name Service (DNS) etc.*
- **Layer 6. Presentation Layer:** the presentation layer essentially serves the application layer. That is, it accepts all packets which the agent at the above level checks if each one individually has the desired type and if not then converts it and sends them to the recipient. Exactly the same procedure is done at level 6 of the computer that receives the packages where again check all packages if they have the required type. The purpose of this layer is to transform data to the form with which the application is working. Formatting functions at the presentation layer may include compression, encryption, and ensuring that the character code set (ASCII, Unicode) can be interpreted on the other side. For example, if we select to compress the data from a network application that we are using, the Application Layer will pass that request to the *Presentation Layer*, but it will be the *Presentation Layer* that does the compression.
- **Layer 5. Session Layer:** The position of *Session Layer* of the Seven Layered OSI model is below the *Presentation Layer*. Session layer is the fifth layer of seven layered OSI model. The *Session Layer* is responsible for establishing, managing, and terminating connections between applications at each end of the communication. In the connection establishment phase, the service and the rules (who transmits and when, how much data can be sent at a time) for communication between the two devices are proposed. The participating devices must agree on the rules. Once the rules are established, the data transfer phase begins. Connection termination occurs when the session is complete, and communication ends. In practice, *Session Layer* is often combined with the *Transport Layer*.
- **Layer 4. Transport Layer:** The fourth layer of the seven layers of OSI network model is the *Transport layer*. The *Transport layer* handles transport functions such as reliable or unreliable delivery of the data to the destination. On the sending computer, the transport layer is responsible for breaking the data into smaller packets, so that if any packet is lost during transmission, the missing packets will be sent again. Missing packets are determined by acknowledgments packets (ACKs) from the remote device, when the remote device receives the packets. At the receiving system, the transport layer will be responsible for opening all the packets and reconstructing the original message. The transport layer also enables the option of specifying a "service address" for the services or application on the source and the destination computer to specify what application the request came from and what application the request is going to. Many network applications can run on a computer simultaneously and there should be some mechanism to identify which application should receive the incoming data.

- **Layer 3. Network Layer:** The third layer of the seven layers of *OSI* network model is the *Network layer*. The *Network layer* of the *OSI* model is responsible for managing logical addressing information in the packets and the delivery of those packets to the correct destination. Routers, which are special computers used to build the network, direct the data packet generated by *Network Layer* using information stored in a table known as *routing table*. The routing table is a list of available destinations that are stored in memory on the routers. The network layer is responsible for working with logical addresses. The logical addresses are used to uniquely identify a computer on the network, but at the same time identify the network that system resides on. The logical address is used by network layer protocols to deliver the packets to the correct network. The system addressing used in *Network Layer* is known as *IP address*. It is this layer, that this thesis will primarily focus on.
- **Layer 2. Data Link Layer:** The second layer of *OSI* network model is called the *Data Link Layer*. This level is responsible for the safe and seamless transfer of packages from the sender to the recipient. This is achieved by providing an end-to-end validation of the data being transmitted. The *Data Link Layer* is logically divided into two sublayers: The *MAC Sublayer* and the *LLC Sublayer*. *MAC Sublayer* determines the physical addressing of the hosts. The *MAC* sub-layer maintains *MAC* addresses (physical device addresses) for communicating with other devices on the network. *MAC* addresses are burned into the network cards and constitute the low-level address used to determine the source and destination of network traffic. *MAC* Addresses are also known as Physical addresses, Layer 2 addresses, or Hardware addresses. The *LLC* sublayer is responsible for synchronizing frames, error checking, and flow control.
- **Layer 1. Physical Layer:** The first layer of *OSI* network model is called the *Physical layer*. Physical circuits are used on the physical layer of *OSI* model. The *Physical layer* deals only with the physical characteristics of the channel, where the data is transferred. It basically includes the voltage of the electrical current used to transport the signal, the media type (Coaxial Cable, Optical Fiber, radio frequency), impedance characteristics, physical shape of the connector, the layout of the pins and various other physical requirements. The *Physical Layer* is responsible for the processes needed to place the communication signals over the media, and to receive signals coming from that media. The lower boundary of the physical layer of the *OSI* model is the physical connector attached to the transmission media. The lower level of the physical layer is essentially the means of transmitting information. As oxymoronic as it sounds, the material from which the information is transferred is not part of the physical layer and is a completely different field. The transmission medium is considered to be the zero level in the protocol [4],[5].

In this section, the *OSI* model and its layers were described to explain what exactly an *OSI* reference model is, why it is used and what is its role in the internet today. *OSI* model is not so much an architecture but more of a guidance model for how should the applications should communicate, which only gives us an idea how packet transfer over the network during any communication.

2.2 Platforms to Emulate Networks Topologies

There are quite a few networking simulators and emulators available on the internet today. But due to the limitation of this thesis, we will only refer to the most used emulators that exist and, will specifically refer to *GNS3* simulation (REFS), the *Packet Tracer* from *Cisco* and the *EVE-NG* emulation platform (REFS). These are some of best platforms to obtain a practical experience, can be used for testing network technologies for deployment in the real world.

Specifically, these platforms are used for the sole reason that administrators cannot always predict how things will turn out, especially when you have to manage a large number of computers where something to go wrong is very high. Thus, the replication of the networking system and the resolve of any issue that may arise before the actual deployment, offers the perfect testing environment.

A concise analysis will be made on each of the platforms and at the end will be a comparative table.

2.2.1 The GNS3 Platform

The *Graphical Network Simulator-3 (GNS3)* is a free, open-source client/server interface for network emulation and virtualization that is focused mostly on supporting *Cisco* and *Juniper* software. It was first released in 2008 and it uses virtual and real devices in order to emulate complex networks. It is primarily Python-based platform with the ability to run Dynamics images and simulate various Cisco and Juniper hardware or software machines. What offers, is a platform that is easy to use, and the user can make a series of web topologies and not worry that something may not work since the environment is completely controlled and designed for creating and testing experimental topologies. [6].



Figure 3: GNS3 Platform[7]

At the same time the team behind GNS3, performs frequently upgrades in its product so that it can support newer and multiple devices from different manufacturers around the world (*Palo Alto, Cumulus, F5* e.t.c).

Also offers the ability to install the GNS3 platform, into a Virtual Machine (VM) and to act as a server. Even if someone does not have the physical prerequisites to build a virtual machine, the company offers the possibility to install the platform on your local computer and then connect to the server of GNS3 that the company has. Once the steps are completed, you will be able to create normal network topologies.

GNS3 like any other platform, it has some advantages but also some disadvantages, some of them are:

- GNS3 is a free network emulator: “Open-source” software means that the source code of the software can be reviewed and modified by the general public. Because GNS3 is open source, anyone can review the software’s source code. The greatest advantage to the open-source nature of GNS3 is the community.
- Simple to use with well written Documentation: The GNS3 documentation is considered to be a well-organized and written documentation and from open source platforms.
- Editable Active Topology Configuration: In GNS3, in each topology each individual part of the topology is completely autonomous and changes can be made without having to stop the operation of the entire topology. At the same time one can add or remove pieces autonomously and make a finish different from the original topology.
- A variety of Connection Types: Attention has also been paid to the type of connections that can be made between devices in a topology. Therefore, in addition to the classic connection through ethernet, it also offers other ways of connection such as
- Large and active community: The GNS3 community allows you to exchange network topologies with other users, thus sharing you work, therefore exchanging knowledge and opinions between users and at the same time offering a kind of inspiration and new ideas for a different way of constructing topologies.

GNS3 may have many positive features making one of the best virtual network emulators in the market, but because it’s an emulation platform it also requires specific licensed software images like *Cisco IOS* and to be suited with GNS3 platform. Therefore, in order for someone to add additional images to the platform in addition to those with which you come pre-installed they will need to obtain its official ones so that they can use them within the platform.

2.2.2 The Packet Tracer Platform

Packet Tracer is one of the most well-known network simulators powered by *Cisco* and is widely used to educate those interested in order to obtain certification level on the products like routers, switches, firewalls, and more. This platform has been built with the purpose mainly of the educational part but also the part of the certification exams. Where someone to get a certification on CCNA will be examined based on the knowledge he has about computer networks through its platform, where Cisco supports all the products that the it has. [9].



Figure 4: Packet Tracer Platform[8]

It offers the possibility to install in almost all software and gives the opportunity to those interested to be trained not only in basic concepts of networks but also to develop a range of skills in the machinery of the company. A reminder as in its case is a simulation without having a complete correlation with the physical

machines but only a copy without taking into account all the parameters that exist in a real environment.

Packet Tracer has several advantages as a free *Cisco* network simulator, including:

- ❖ Cost: It is completely free to download and easy use with a nice UI. The only requirement is that you create and log into Packet Tracer with a Cisco Networking Academy account, which is free to create.
- ❖ Cisco Platform Compatibility: *Packet Tracer* runs on most operating systems, including all active Windows operating systems (*Windows* 7, 8.1, and 10), *macOS*, and *Ubuntu* 16.04 LTS and 18.04 LTS.
- ❖ Simulation Mode: From the moment a topology is made on the platform it will be in specific mode (Realistic mode). That is, the devices will automatically produce packages and forward them so that the topology is more realistic like in a real situation.

However, it is considered one of the most stable and reliable platforms on the market. The fact that it is free and offers access to several devices with a rich toolbox of options so that one can experiment, and the ability to give the average user a very close to real experience, automatically makes the platform one of the best one can use. for training in the field of their networks.

Finally, the limited number of the Cisco's platform IOS software devices and the fact that is a simulation platform instead of an emulation, it bestows a negative side to *Packet Tracer* making it a less attractive option for networking testing and emulation.

2.2.3 The EVE-NG Platform

EVE-NG (Emulated Virtual Environment- Next Generation) and its predecessor UNetLab are network topology emulators. They have the ability to support both commercial and open source images of routers. *EVE-NG* has been built to create and test network topologies within a fully controlled environment offered by the platform. Moreover, the *EVE-NG* is compatible with Wireshark in order to do packet capture on the topology interfaces.

The Community Edition main characteristics are:

- ❖ Cost: The Community Edition of *EVE-NG* is offered as free. The differences between the Community Edition and the Professional Edition of *EVE-NG* are mainly two. The Community Edition has a 63-node limit per lab (which is more than enough for most use cases). The Professional Edition has a few administrative features missing in the Community Edition, including support for multiple users, user roles.
- ❖ Clientless: The *EVE-NG HTML5* client is a very convenient tool that makes it different it from *Packet Tracer* and *GNS3*. In *EVE-EG*, the user design, connect, and manages network topologies through an *HTML5* client. In other words, the user doesn't need to download and install a separate application in addition to the server to virtualize, connect, and configure network devices. You simply deploy the server through a bare-metal installation or virtual

machine, and everything else can be done through the *HTML5* client. The *HTML5* client is very responsive, even when working with larger topologies.

- ❖ Modifiable Active Topology: Like *GNS3*, *EVE-NG* gives the option to the user to dynamically and or real time make changes on the topology. As it was stated and before the ability to adapt the topology is an excellent and efficient method especially when working with nodes that would normally take a long time to boot.
- ❖ A variety of connectivity options: *EVE-NG* likewise *GNS3* has the ability to supports multiply types of connectivity both serial and Ethernet interfaces.
- ❖ The option to export/import labs: *EVE-NG* provides the uses the ability to export and import network topologies from one *EVE-NG* platform to another without any compatibility issues.

The *EVE-NG* Community Edition ant be without some disadvantages itself some of them are:

- ❖ Software Image Access: As in the previous cases the platform itself does not give access to any image, so the user will have to obtain some images so that he can use them on the platform.
- ❖ Documentation: The *EVE-NG* documentation in general, the official document from the site is not considered very friendly to the inexperienced user and in some cases, it may be difficult to understand and apply the instructions given.

EVE-NG stands out among its competition as the only clientless virtual network emulator. While *GNS3* require the user to download and install a separate application to manipulate network devices on a server, *EVE-NG* only requires a lightweight terminal application.

The design, creation and execution of the entire web topology can be achieved entirely through the website. This feature alone offers great freedom to the end user as it is given the ability to connect to the platform via any device such as a mobile phone. As mentioned above, a negative is the lack of access to images, as is done on the *GNS3*, and the license or purchase from the official seller it may require.

Last but not least, the person who will install the platform will need to already have some basic knowledge of virtual machines and to have some knowledge of terminals and Linux systems.

2.2.4 Comparative Presentation

The existing environment from simulators may seem hostile and inaccessible to an inexperienced user entering this field for the first time. Where each platform has been designed separately with a different mindset to cover a specific part of the market. Therefore, a newcomer will not know what he is asking for and what solution he will make for his own needs.

For this reason, a brief but analytic table that contains the main features about the main characteristics of each option is available. of each platform that was mentioned is presented in Table 1.

<i>Criteria</i>	GNS3	Packet Tracer	EVE-NG
-----------------	-------------	----------------------	---------------

<i>License (Cost)</i>	None	None (For students)	Free Community Edition (Not the premium edition)
<i>Centralized Management</i>	Multiple Isolated Servers	Central Managed cluster with multiple compute nodes.	Central Managed cluster with multiple compute nodes.
<i>Combability and Accessibility</i>	GNU/Linux, Windows, MacOS	GNU/Linux, Windows, MacOS	Web-UI
<i>Load Balancing</i>	Manually	Yes	Only in Premium edition
<i>Required Compute Resources</i>	Medium (device dependent)	High (device dependent)	Medium (device dependent)
<i>Custom Images</i>	Images for routers needed	Extendable (node restriction only for induced Cisco nodes)	Images for routers needed

Table 1: Platforms Comparison

In summary, any of these platforms is a great choice for use in education since all of them are available for everyone with plenty of learning options and a very large and active community, but what makes *EVE-NG* to stand out, is that it's clientless virtual network emulator with huge capabilities, available to everyone ,with a *HTML5* client where through it the user can oversee and configure the topologies. Finally, the option to run *EVE-NG* in a bare-metal server or in a VM with any operating system such *Windows*, *Linux* or *MacOS*, makes it the best option to pick for a university lab emulation platform.

2.3 Introduction to EVE-NG Platform

To describe *EVE-NG* platform, it is necessary to describe what and where the *EVE-NG* platform can be used and its which situation is useful for us. *EVE-NG* provides the tools that are essential in order to build network topologies and test them in a risk-free environment and also to monitor their traffic connectivity.



Figure 5: EVE-NG[10]

Many of the features offered by the EVE-NG platform are helping to a great extent to simplify the way of managing a networking topology and to make the testing environment easier for the administrator so that in a short time he can understand a topology, edit it and share it. EVE-NG offers many features using an HTML5 web user interface which can be accessed via *Virtual Network Computing (VNC)*, *Telnet or Remote Desktop Protocol (RDP)* and gives the ability to import and export any configurations from other *EVE-NG* platforms instantly and import them to another *EVE-NG* platform. Users have the ability to add nodes from a list of templates configure them and manage them. [14].

Moreover, administrators or user with more clearance have the option to add software commercial and open-source router images that emulate an Operating System, a host or a router in order to create a virtual environment and create custom templates to recreate any scenario they want. EVE-NG can run standard networking devices such as *Dynamips* and also to runs a variety of other software like routers, on image type QEMU. *EVE-NG*, as it was mentioned previously, is an open-source project and the *EVE-NG* source code is available for everyone.

Some of *EVE-NG* main characteristics are:

- User cannot see other EVE folders, only his/her own.
- User cannot edit labs or images, only the admin can.
- Shared lab folder visible for all users.
- Timer for Lab training.
- Wireshark can be used as a packet capture and analyzer.
- Limit of nodes to run per lab.

All these characteristics make *EVE-NG* an excellent platform for a network emulator in a university lab. The reason is, that because it will be running in a VM there will be no need for the existence of high end computers at the universities, that's because a platform that emulate the functions of a device will require more resources that a simulation platform due to the fact emulators have more complexity of networking functions, and thus the whole process will be running in a VM that will be located in the *OKEANOS* Cloud service.

2.3.1 The role of GRNET

As it has already been stated, the emulation platform will be located on a VM on the cloud service *OKEANOS* which is powered by Greek Research and Technology Network (*GRNET*). *GRNET* is an integrated electronic Infrastructure provider whose mission is to support high-quality e-Infrastructure services to the academic, research and educational community of Greece.

The role of *GRNET* is to provide Cloud Computing services in the form of a public-to-service (Infrastructure as a Service, *IaaS*), called *Okeanos* platform. Through *Okeanos*, any academic user can create a virtual infrastructure by combining simpler virtual building blocks. It activates in a matter of seconds hundreds of VMs, which interconnect via virtual networks to random topologies

with the ability to store data either on virtual disks or Cloud Object storage (Cloud object storage) [15]. Also, the whole *Okeanos* service is powered by *Synnefo* which a complete open source cloud stack written in Python that provides Compute, Network, Image, Volume and Storage services, like the ones offered by *Amazon Web Services (AWS)*.

2.3.2 The OKEANOS cloud and myNetlab project

Okeanos is an Infrastructure as a Service cloud provider that was developed and maintained by the *GRNET*. The service is powered by *Synnefo*, an open source software builds on top of existing open source software platform or applications (Google Ganeti, Ceph, etc.) and has been expanded in-house in order to provide a state of the art and complete IaaS cloud service provider. This move for the creation of the first public cloud which will not charge at all for its services the users who use it community for academic research purposes which was the goal of the Greek Research and Academic community.

Using *Okeanos* platform we created the *mynetlab project* with the goal of the construction of virtual network laboratories using the *EVE-NG* platform. After creating *mynetlab project*, the creation of a VM and the installation *EVE-NG* on the VM took place [15].

3 Chapter 2: Creation of the networking scenarios on the EVE-NG platform

In this chapter the presentation of the laboratory courses that will be studied accompanied by their theoretical background will take place. More specifically, the key characteristics of routing will be represented and the different solutions for dynamic routing and their algorithms will be thoroughly discussed. Additionally, the advantages and disadvantages of each routing technique will be highlighted.

Finally, a brief analysis of the *JUNOS OS* will take place focusing on its key characteristics and the differences from the *CISCO IOS*. At the end of this chapter a description of the experiment topologies that are going to be presented in Chapter 3 will be provided.

3.1 The purpose of this scenarios

The purpose of this series of laboratory exercises is primarily focusing on learning the basic principles and concepts of how the internet “works”, especially by helping those who want to learn about the way communication through the Internet is occurred.

Specifically, scenarios focusing on the use of static and dynamic routing are selected, showing how the information flows in the form of datagrams on the Internet through the routers, using the tools that are available from the *EVE-NG* platform. These exercises can be used as an introduction to basic networking concepts and to computer protocols, such as: the *IPv4* datagram, *IP* addressing,

subnetting, *IP* routing and how to address them. Understanding of how static routing operates, its advantages and disadvantages in comparison to dynamic routing is also explained. Moreover, an in-depth analysis on dynamic routing and its protocol such as *OSPF*, *RIPv2*, *IS-IS* and *BGP* which can dynamically route packets through the internet, and their characteristics takes place.

Finally, the differences between those two routing techniques and which criteria are needed for consideration, in order to choose the best possible candidate for routing in each studied case will be highlighted.

3.2 An overview of *IP* addressing and *IP* routing

In an architecture of this scale that the internet has, there must be a very specific way in which each terminal or device can stand out from the rest of the devices. This is achieved by using addresses. As in real life where every home has its own address so on the internet a similar idea is used to these *IP* addresses [1], [2]. Each address is expressed in decimal numbers and is separated by the use of dots (ea. 192.169.10.4).

These addresses are not connected to an entire device, so that the device can have internet access. Instead what actually happens is the connection of a specific address to one (of the many) interfaces that the device or router has. Essentially the *IP* addresses are a logical identifier for an interface.

Since the *IP* addressing management is essential to every network and because it will be using it extensively in our network topologies, it's very important to understand how *IP* works because, to master the *IP* addressing is to master the Internet's network layer itself.

3.2.1 The *IP* Datagram Structure

The basic unit for data transmitted in the *IP* layer is transferred to the *IP* level in packets that have a predefined format. These packages are called *IP* datagrams and are the official format for *IPv4*

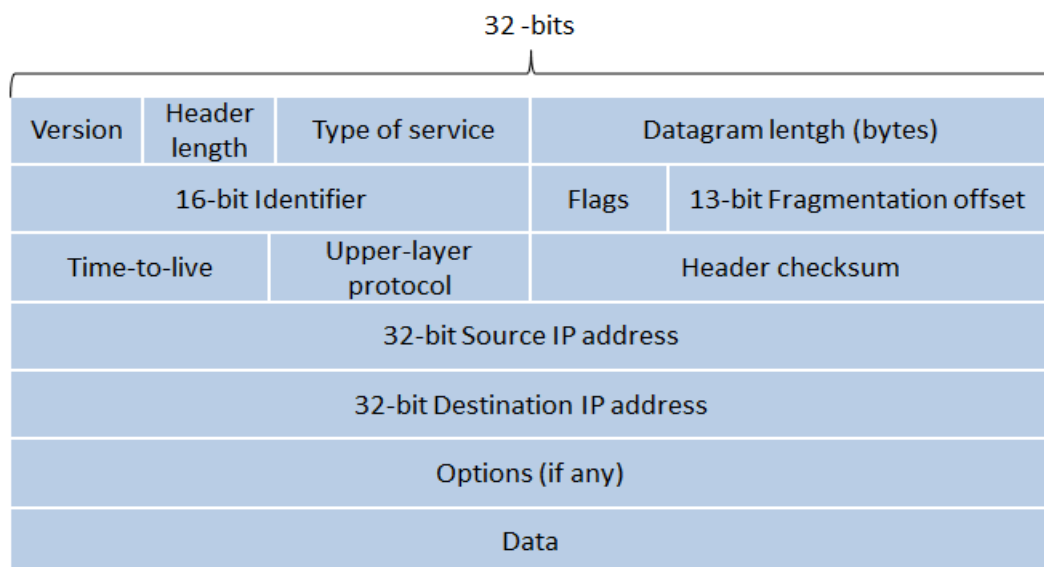


Figure 6: The *IPv4* Datagram Format

protocols (Figure 6). Although the IPv4 and IPv6 protocols each have a different datagram structure, there is the potential for them to be able to communicate with each other. In this thesis we will analyze only the structure of the IPv4 datagram. The datagram is divided into 2 major parts, the header where the addressing and control fields are located and the payload where the junk of the data is located. The IPv4 datagram is divided into the following parts:

- Version Number: Recognize which version of IP was used to create the datagram. For IPv4, this is of course the number 4. The purpose of this field is to ensure compatibility between devices that may be running different versions of IP (IPv6 for example).
- Header length: It determines the size of the IP header. The reason is because the diagram may contain many variables, and it should know up to what point is the header and at what point the payload starts.
- Type of Service: Is a series of bits and indicates whether to allow the IPv4 header to accept different types of IP datagrams (for example, datagrams particularly requiring low delay, high throughput, or reliability) to be distinguished from each other. It was a field designed to carry information to provide quality of service features, such as prioritized delivery, for IP datagrams.
- Datagram Length: It specifies the total length of the IP datagram (header plus payload), measured in bytes. Since this field is 16 bits long, the theoretical maximum size of the IP datagram is 65,535 bytes. However, datagrams are rarely larger than 1,500 bytes, which allows an IP datagram to fit in the payload field of a maximally sized Ethernet frame.
- Identifier, flags, fragmentation offset: The specific fields are responsible for the fragmentation of the load when and if needed. In the new version IPv6 there is no option to fragment the information.
- Time-to-live (TTL): One of the most important fields. It indicates the number of routers through which information can pass until it is dropped. That is, this value is reduced by one unit for each router through which it passes as a countdown. If this number goes to zero then it is considered that the package takes a long time to reach its destination with the result that a router discards it.
- Protocol: This field is used only when the package has reached its final destination. The value is corresponding to a transfer protocol for the above level. For example, if the field has the value 17 then it means that the packets use the protocol UDP while if it has the value 6 then the protocol TCP is used.
- Header Checksum: This field is responsible for checking the datagram and identifying possible errors that may have been created during the transport of the package. When a router is receiving a packet, it calculates the value of the field and if it has the same value then it means that the packet has not been damaged. Respectively, if the price it finds is different, then there has been an alteration of the information and the router discards the package.
- Source IP Address: This field includes the router IP address that created the datagram. Because the package goes through multiple routers, none of them change the specific value, only the original router can change this value.
- Destination IP Address: Respectively this field indicates the IP of the final destination of the package. Respectively, only the original router that made the package can set this price and none of the intermediate routers through which the information passes changes this price. Also,

all other routers when receiving a packet see the final destination and redirect the packet to the corresponding output. That is, the original sender does not know from the beginning the route from where the package will go but only the exact next router.

- Options: It enables the datagram to be enlarged if necessary. Such a case is rare since the use of a larger header creates simultaneous overhead problems.
- Data(payload): The last but also one of the most important fields. This field contains all the information that a router wants to send to another. Once this information reaches the final destination it is taken to the next level, the transport layer to take it to the final destination [3].

The current Internet relies on *IPv4*, which has 32-bit addresses. The *Internet Engineering Task Force (IETF)* has developed *IPv6*, which has 128-bit addresses, as the proposed replacement for *IPv4*.

3.2.2 IP Addressing Classful, Classless & Subnetting

As we know, the routing of packets on the internet is not random, but it is done in a very specific and efficient way. Once a package that carries information reaches a router and the following process unfolds. The router sees the package and checks the destination IP to which it is destined. If the packet destination is on this router then the packet has successfully reached its destination, but if this does not happen then the router must forward the packet to the next router. This is done in the router by checking a table that the router maintains and updates with all the neighboring routers and destinations that a packet can be routed to, and therefore knows which interface should forward the packet to the destination in order to arrive.

This technique of checking and selecting the most ideal route is called the longer prefix matching technique. Once it finds more IP it looks more like it then distributes the packets to the interface corresponding to the IP corresponding to the destination IP of the packet. All devices that operate on the internet and use the IP protocol (routers, switches) use this packet distribution system.[3].

To determine the best path, when a packet is received it contains information about its source and destination address. The router forwards the packet on a particular interface by matching the address against a list of routing table entries, commonly referred to as subnet prefixes. The challenge comes when an *IP* address matches against more than one such prefixes. In this case, the longest matching prefix is selected. Then the forwarding table gives the appropriate instructions to the router to send the information to the next hop. In essence, some of the features that a routing table retains are.

- Destination: The *IP* address of the packet's final destination.
- Next hop: The *IP* address to which the packet is forwarded.
- Interface: The outgoing network interface the device should use when forwarding the packet to the next hop or final destination.
- Metric: Assigns a number to each path, the larger this number indicates that this route is not ideal, while the smaller the number indicates that this route is more suited.
- Routes: Indicates which networks the router has direct access through its interfaces and which networks it knows exist and communicates but not directly but indirectly through other routers.

As in packet routing, there are two ways in the routing table that can be maintained and updated. It is done in a static way where an administrator should periodically update the available treadmills that he could use. where

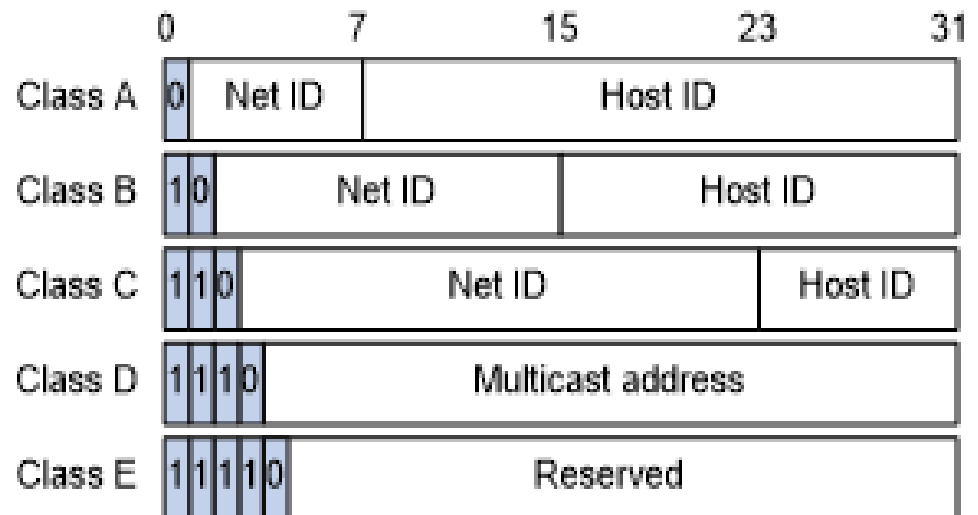


Figure 7: IPv4 Classful Addressing

it will use routing packages to exchange information with neighboring routers to update the table when there are cases where a router will have failed [16].

An IP address normally consists of 32 bits (10000110.00010111.11100010.00101101) to which 4 parts of 8 digits are separated. For better management they can be represented as decimal numbers and can take values between 0 and 255(e.g., 192.168.10.25). This method is referred as dotted-decimal notation. The general rule is that whatever is connected to an IP network then has the ability to get an IP address. If a device has multiple interfaces connected to a network then the same device can have multiple IP addresses, one for each interface. Each address that will be given is completely unique IP address and there can be not 2 identical IPs in a network [5].

As mentioned above each IP address is essentially a series of 32-bit numbers in the binary system. In general, these octaves offer the ability to create subnets for networks of different sizes. There are 5 groups of different networks that have been made. In each a group has been given a separate group of addresses where they have been defined by the first 3 digits of the first octave. This seems clearer in Figure 7 where the change of value in the first 3 digits of the octave represent the changes of the group it belongs into.

Classful addressing manages IP addresses space (0.0.0.0 to 255.255.255.255) into divide the IP's into specific groups or classes, and one can understand, just by looking at the first 4 digits of the address. These first 4 digits define the network part of IP's are already referred to as 'Most Significant Bits'. With this way of separating addresses, 5 different groups of addresses (Class A, Class B, Class C, Class D & Class E) have been created with different purposes each, and whenever a router receives an address it can very easily understand which part of the address belongs to the network and which part belongs to the host.

Depending on the amounts of numbers given in the network part, the number of available addresses in the host part will be proportional. The Table 2 below gives a breakdown of how the Classful system breaks up the IP address space.

In a Class A address, the first octet is the network portion, so the Class A example in Figure 7 has a major network address of 1.0.0.0 - 127.255.255.255. Octets 2, 3, and 4 are for the network

manager to divide into subnets and hosts as he/she sees fit. *Class A* addresses are used for networks that have more than 65,536 hosts (actually, up to 16.777.214 hosts) [6].

In a *Class B* address, the first two octets are the network portion, so the *Class B* example in Figure 7 has a major network address of 128.0.0.0 - 191.255.255.255. Octets 3 and 4 (16 bits) are for local subnets and hosts. *Class B* addresses are used for networks that have between 256 and 65.534 hosts.

For a *Class C* address, the first three octets are the network portion. The *Class C* example in Figure 7 has a major network address of 192.0.0.0 - 223.255.255.255. Octet 4 (8 bits) is for local subnets and hosts - perfect for networks with less than 254 hosts. The addresses of class D (Multicast) and class E (Experimental) cannot be given to networks and they exist to serve special reasons that are beyond the scope of this thesis.

Since we discussed the *IP* addresses and the classes that exist, we proceed to the analysis of network masks and subnetting to complete the overview of *IP* protocol.

CLASS	NETWORK & HOST	MASK VALUE
Class A	N.H.H.H	255.0.0.0
Class B	N.N.H.H	255.255.0.0
Class C	N.N.N.H	255.255.255.0

Table 2: Default mask Value

A network mask determines which portion of the address identifies the network and which portion of the address identifies the hosts [1]. *Class A*, *B*, and *C* networks have default masks or, as generally referred to them, as *classful addressing*.

Table 2 summarizes the hosts and networks for the different classes of address [14]:

Class	Network Mask	Network Bits	Host Bits	Number of Networks	Maximum hosts per network
A	255.0.0.0	8	24	2^{8-1}	$2^{24} - 2$
B	255.255.0.0	16	16	2^{16-2}	$2^{16} - 2$
C	255.255.255.0	24	8	2^{24-3}	$2^8 - 2$

Table 3: Classfull Network Masks

All the above analysis of addressing and how to distribute them into classes and subclasses has been classful addressing. But if there seemed to be a good way to make a distribution of addresses around the world, it turned out to be very unprofitable, because more and more people were asking for a part of the IP addresses. Usually when someone asked for an IP then it was very common to give him one part from Class C and this was because very few people wanted more than 256 IP.

But as the ever-increasing demand for addresses continued, the available addresses from the Class C began to run out. This led to the decision to give a chunk of blocks from Class B. Although it seems like a solution, this is not at all efficient, because when a block from Class B was given, it meant that they gave a total of 65,034 IP's. So, if someone wanted to have 400 IP's they had to give him a block from class b. As we understand this was very unprofitable, for the industry and therefore for the future of the internet since there would be no IP available for everyone.

This led to the process of finding a solution that would solve this problem of numerous dispersions of addresses in large blocks while a company may only need a small part of it.

This led to the introduction of *Classless addressing*. In the classless's world of addressing, the number of network bits was not fixed like in classful addressing at 8, 16, or 24 bits but there was more flexibility on the number of bits that have been reserved for the network which could be any number from bit 0 to 31. With that in mind the networks from now on could be partitioned into smaller ones, called subnets. All the hosts on the same network were configured with the same subnet mask, and share the same pattern of network bits, and so the utilization of *IP* addresses improved drastically [1], [5].

This meant the following very important. Every network now had a size like. That is, in the case where the network bits were 4, there could be 2^4 addresses. And within this network that was created the number of available addresses will be 2 less, for the reason that 2 addresses must be reserved for other purposes. Specifically, one address was reserved for the network ID and the other for the broadcast address.

Finally, a way had to be defined where it could be easily seen in an address which part of where the network and which part was the hosts. This led to the creation of subnet masks. It became a key tool in helping network administrators find the number of addresses available on their network. In general, networking is a very useful tool that helps one to build smaller networks within one's already main network.

What actually happens is, when someone wants to do subnetting then they borrow some bits from the host and go to the network part. This has the effect of making smaller subnets within the primary.

Another advantage of subnet mask is that it also helps the routers. When they want to see which in subnet a host belongs to, then they look at the network mask and based on that they see which bits belong to the network and then they keep only the part of the network and they know in which network the specific host belongs to.

3.2.3 IP Routing Static & Dynamic

As has been seen from previous chapters, the packet routing process itself is not a simple matter. It is actually a fairly complex process which involves many different pieces to work properly.

Specifically, in addition to the physical machines needed to transport packages, the necessary functional part is also a necessary part. That is, the appropriate algorithm who will tell the router from which interface it should send the packets in order to reach their destination. Different algorithms and protocols can be used for this job to identify the optimal path for packages [16].

Essentially routers are simple routers of information passing from one router to the next. Additionally, packets are providing network information. From the headers of a packets, as it was mentioned earlier in the *IP* datagram, we can extract information about its origin and its destination address. There are two basic types of routes:

- **Static Routing:** The simplest form of routing available on the internet is static routing. It is required the supervision and maintenance of the routes by the administrator where he will be responsible for everything. From the creation of the routes but also their change in case a router fall.
- **Dynamic Routing:** Dynamic routing is very different from static routing. Where here responsible for the creation, maintenance and update of routes is a dynamic routing algorithm and not the administrator. There are many such algorithms available with each one having different features and having to use different ones depending on the needs presented in each case. [14].

3.3 Static Routing

It may seem that static routing is no longer used and is obsolete. But this is not the reality. It may seem that in the face of dynamic routing (like OSPF or BGP), the static with the constant supervision of an administrator is not interesting, but there are many cases of topologies where the use of static routing is suggested and still considered necessary.

Although at first glance it seems that static routing does not have any positive features to prefer this is very far from reality. Static routing is in general very easy to apply in network topologies and does not need to be available the router or from the network to operate efficiently, in contrast to dynamic routing where it needs resources and speed to be able to operate.

Static routing is especially preferred in topologies of small and manageable networks where there will be no changes in the network and the change of routers is expected to remain constant without any upgrade, since everything will be known to the operator.

3.3.1 Analysis of Static Routing

The use of static routing allows the network administrator to manually enter the routes, configure it and instruct for each *IP* what the next hop would be to transfer the traffic.

Generally, in static routing the administrator has defined in the routing table the routes from where the router will send the information for each possible destination. When you receive a packet on the router it looks at the destination address and sees depending on which of its routes it is destined to reach its destination. When the router sees the IP 10.10.9.9, the next step is to go to the routing table and see from which interface it should direct the packet to go to its final destination. That is, the router sees where he wants to go the package and checks the routing table and decides to direct the package accordingly, as soon as the package reaches the next router, the same process is repeated. [15].

Router IP's	Destination IP's
10.0.0.0/16	10.10.9.9
10.0.0.0/16	10.10.10.10
3.0.0.0/16	5.5.5.5

Table 4: Routing Table

In order for the static routing to work normally and without problems, the topology administrator must define the routing not only to one router but also to all that exist in the network and to all the interfaces that are connected.

This means that the administrator must have an excellent knowledge of the topology and routing that exist, and through their configuration it is enough to manage any problem that may arise since he will be able to know everything necessary to solve the issue.

As we said in order for the static routing to work, in addition to defining all the paths by the administrator, the topology itself must remain static. This is because any other application on the topology will lead to problems because the routers will not be able to communicate and transfer packages efficiently. In order to solve this, the administrator must again inform the routing schedule to all routers that he deems necessary so that he can return to his normal state. Simply stated, **static routing is great for networks that don't change.**

Therefore, the administrator or anyone who has a very good knowledge of network topology should be available at any time. And this is the big difference with dynamic routing. Where if one path on the network disappears (hardware failure, a broken, data link), dynamic routing will tell the routers how to route around that path without manual intervention.

3.4 Dynamic Routing

If someone asked how the internet works today and how it achieves it, the answer to that question would be through the use of dynamic routing.

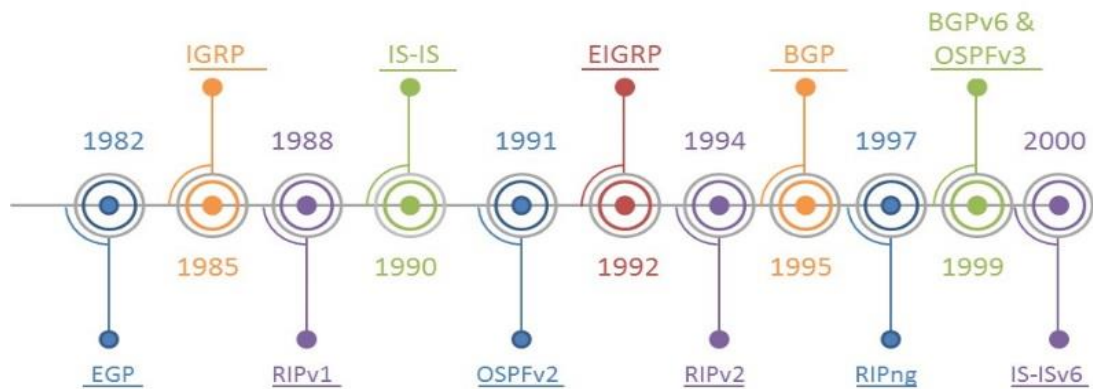


Figure 8: Routing Protocols Evolution

Strange though, dynamic routing protocols have been around for almost 40 years. As the networks grew and their architectures became larger and more complex, it became apparent that there could not be an administrator to maintain packet routing. Thus, were created the dynamic routing protocols where they served one and only one purpose. Calculating and deciding which will be the most ideal path for data transfer. The protocol does this process automatically and continuously so that it can detect possible changes or failures that may have occurred. When such a change is detected the algorithm calculates the paths again to make changes where needed.

An important role is also played by the fact that the algorithm runs on all the available routers that exist in the network and at the same time talks to the other routers to know who is operating and who is not, and to know who is making a change[15]. Compared to static routing the dynamic must use resources from the router to operate and at the same time use bandwidth to communicate.

Having had almost 40 years of existence in the heap of networks, there have been many different algorithms over time. Some of them are simpler and are best used in small and medium environments. While more modern algorithms are more complex and can operate in different environments with many factors and be more efficient at the same time.

The first and simplest algorithm built was the *RIP* algorithm. Although there are 2 versions of it (RIPv1 and RIPv2), both of them cannot work on network topologies which are large and have many limitations in their architecture due to the different requirements that existed in the past. With the increase of internet users and the corresponding increase of routers, other algorithms (IS-IS, OSPF, BGP etc.) that were more suitable for large and somewhat complex network topology environments have slowly emerged like *Open Shortest Path First (OSPF)* and *Intermediate System-to-Intermediate System (IS-IS)*.

Additionally, the need for connection between networks running different protocols had been created. The Border Gateway Protocol (BGP) was created for communication between Internet service providers (ISPs). In Figure 9 we can understand better how routing protocols work, and can be classified according to their characteristics. For now, this section provides a very brief overview of each protocol [15].

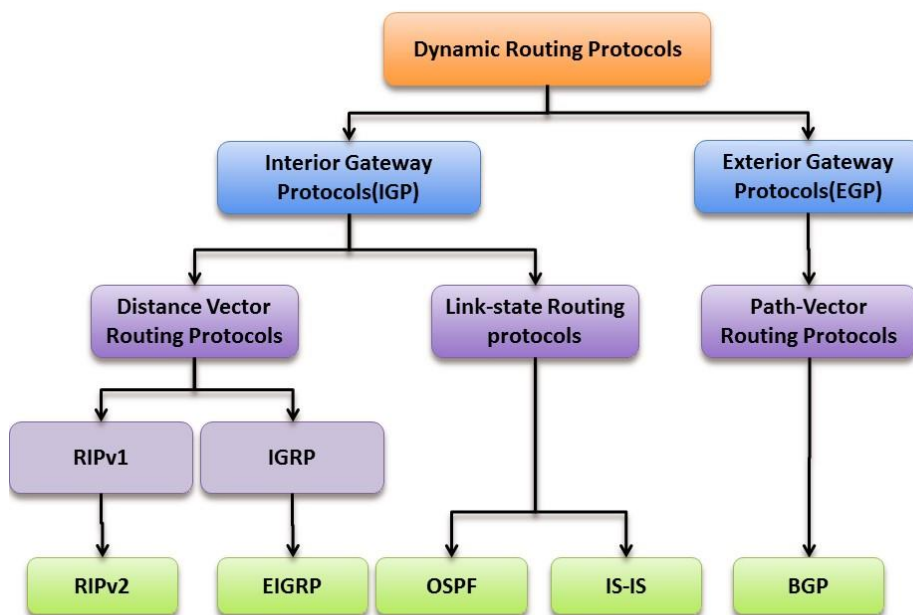


Figure 9: Dynamic Routing Protocols Classification

For instance, routing protocols have been categorized as follows:

- Ripv1 (Legacy): It's An IGP, Distance Vector, Classful Protocol.
- IGRP (Legacy): IGP, Distance Vector.
- Ripv2: IGP, Distance Vector, Classless Protocol.
- EIGRP: IGP, Distance Vector, Classless Protocol Developed by Cisco.
- OSPF: IGP, Link-State, Classless Protocol.
- IS-IS: IGP, Link-State, Classless Protocol
- BGP: EGP, Path-Vector, Classless Protocol.

Most of these protocols are used in modern world network topologies. some of them are obsolete and find use only in old topologies where they no longer have or can no longer be upgraded. As it becomes obvious, each protocol has different characteristics and performances and is mainly due to the architecture on which they were acquired but also the problems they wanted to solve such as fast communication between routers, fast, stability of the algorithm, direct information. The data is moved around different network

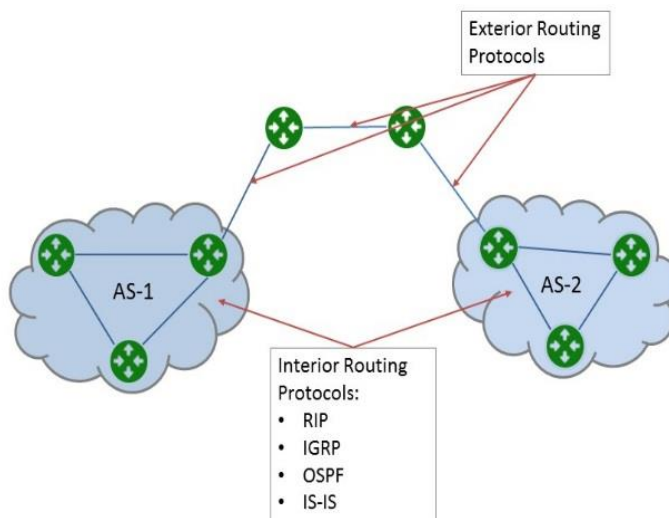


Figure 10: Differences between IGP & EGP

topologies and handled by different protocols within and outside of different of *Autonomous Systems (AS¹)*. The Internet is based on the concept of AS concept therefore, two types of routing protocols (*Figure 10*) are required [16]:

1. ***Interior Gateway Protocols (IGP)***: these algorithms are used only in internal networks, like in company topologies or in universities campus. The algorithms that are internal are *RIP, EIGRP, OSPF, and IS-IS*. These dynamic protocols transfer data from one internal topology to another.
2. ***Exterior Gateway Protocols (EGP)***: On the other hand, in order for all internal networks to be able to communicate directly with each other, it had to be done via the internet, where size and scalability take on other dimensions and a different algorithm (BGP protocol) was needed, where it could work effectively. Protocols convey information that is needed from one autonomous system to another

3.4.1 Analysis of Dynamic Routing (Distance Vector vs Link State algorithm vs Path vector)

Because in dynamic protocols there are many algorithms where they are used and each of them has a different architecture from the rest, their categorization began based on the features they present. There are 2 major subcategories in dynamic protocols. "Distance Vector" and "Link State" algorithms. The ultimate goal of each routing algorithm is the same, the correct routing of data to their final destination using the most optimal path. The main features of the two types of algorithms used today are:

1. The routing protocol selects the best routing path based on a distance metric (the distance) and an interface (the vector). In essence, each router advertises to other routers on the network the destinations it knows through its own routing table and some other information to reach the corresponding destination. These algorithms have another subcategory in which they are divided into distance-based algorithms and path-based. The distance vector algorithms are algorithms that calculate the optimal path based on how many routers will the information pass from until it reaches the final destination, and the path-vector where the optimal path is not calculated based on the distance to the final destination but the optimal path is calculated again in each router the package arrives and changes dynamically in each time.
2. Selects the best routing path by calculating the state of each link in a path and finding the path that has the lowest total metric to reach the destination. In this algorithm the algorithm logic changes. The algorithm in each router advertises the state of the interfaces for which it is responsible in the router throughout the network and the same is done by the other routers. Then each of them individually with the information

¹ An ***autonomous system (AS)*** is a group of routers where they are managed by a single entity. Typical examples of an autonomous system are a company's network.

received from the other routers makes a map of the network topology and is called the *shortest path tree* that shows the routes and each possible destination within the network.

	Distance Vector Routing Protocol		Link State Routing Protocol		Path Vector
Classfull	RIP	IGRP	None		EGRP
Classless	RIPv2	EIGRP	OSPFv2	IS-IS	BGPv4
IPV6	RIPng	EIGRP for IPV6	OSPFv3	IS-IS for IPV6	BGPv4 for IPV6

Table 5: Routing Protocols Classification

Dynamic routing protocols like OSPF, RIP and BGP are using algorithms to calculate the best possible route. But each one of them is using a completely different algorithm with a completely different approach to find the optimized route. As we observed at Figure 9 both link-state and distance-vector algorithms are utilized by interior gateway protocols (IGP) and only path-vector protocol is used by EGP [14]. Table 4 [16] shows the classification of dynamic routing protocols.

Each routing algorithm is linked with its way of locating the most efficient and at the same time shortest path in a topology. The reason it is preferred the shortest path to a destination is because generally the shortest path is more likely not to contain web magnets. Network loops are cases where packets do not reach the final destination because they are in a perpetual cycle where

one router forwards them to the next rule, basically a cycle without end, thus creating more traffic to the network [15].

BASIS FOR COMPARISON	DISTANCE VECTOR ROUTING	LINK STATE ROUTING
Algorithm	Bellman ford	Dijkstra
Network view	Topology information from the neighbor point of view	Complete information on the network topology
Best path calculation	Based on the least number of hops	Based on the cost
Updates	Full routing table	Link state updates
Updates frequency	Periodic updates	Triggered updates

CPU and memory	Low utilization	Intensive
Simplicity	High simplicity	Requires a trained network administrator
Convergence time	Moderate	Fast
Updates	On broadcast	On multicast
Hierarchical structure	No	Yes
Intermediate Nodes	Yes	Yes

Table 6: Distance-Vector vs Link-State

There are many algorithms that calculate the optimal path in a network of routers, the most common algorithm being used is the Dijkstra algorithm Shortest Path First (SPF²). Where when the status of an interface changes to a router then a special update called a Link-State Advertisement (LSA³) is automatically sent to the rest of the network stating the change in the status of the interface. As soon as the update is done, then each router updates the map he has made for the topology in order to change as many routes as he deems necessary.

So, the routers with the link-state protocol such as OSPF will complete the following generic link-state routing process to reach a state of convergence:

1. The first step is for the algorithm to learn about all the available interfaces that the router has and are in an active state.
2. The next step is for the algorithm to communicate with neighboring routers and to communicate with its respective interfaces.

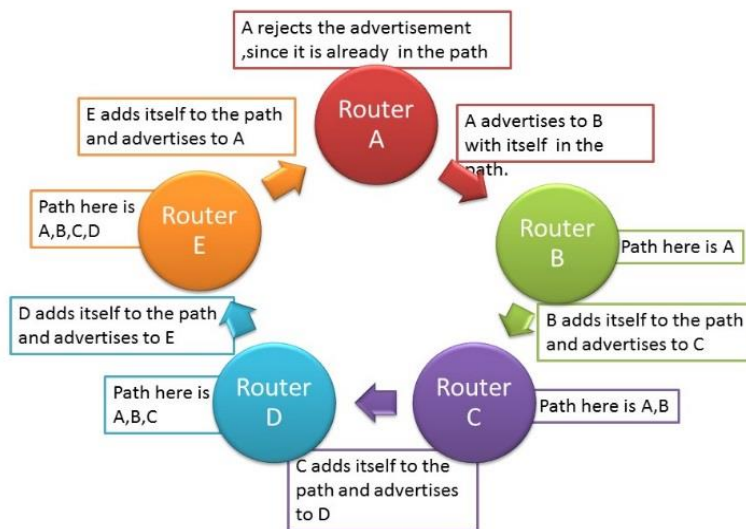


Figure 11: An illustration of Path-Vector Protocol

² SPF algorithm (Dijkstra algorithm) – The shortest path first (SPF) algorithm is a calculation performed on the database resulting in the SPF tree.

³ Link-state advertisements (LSAs) – A link-state advertisement (LSA) is a small packet of routing information that is sent between routers.

3. The next step is for the algorithm to start advertising on the network its own available interfaces and all neighboring routers. This process is performed by all routers on the network, and in addition they advertise not only there are not directly related to each other.
4. Finally, after all the routers and their interfaces have been advertised on the network, then each of the routers individually builds a map of the network topology and the path that the packages must follow to reach their destination.

In relation to the Distance vector algorithms, the link state reaches the final state much faster, where every router knows everyone in the network and is less necessary to create a network in the topology. But at the same time, they need to use more resources than the distance vector [14] ,[15].

Link State Protocols use a hierarchical structure that limits the distance that a Link-State Advertisement (LSA) need to travel. Link State Protocols use multicasts to share the routing information. Only the routers which run Link State protocol process the updates. Link State routers send updates only when there is a change in the state of the network (incremental updates). Link-state protocols work best in situations where:

- The network design is hierarchical, usually occurring in large networks.
- The administrators have a good knowledge of the implemented link-state routing protocol.
- Fast convergence of the network is crucial.
- Link-state algorithms can be more complex and expensive to implement and support.

Unlike link-state protocols, the latter have a simpler architecture and are easier to understand. To find the most efficient pathway, the algorithm calculates how many hops the package is going to do until it reaches its final destination. The most common algorithm used by the latter is the Bellman-Ford algorithm for best-path route determination.

Also, when routers within a network has completed the topology in its routing table, they can send it to neighboring routers. This leads to cases where the router table becomes huge, such as in network topologies with vast routers and this creates significant traffic on the network [16].

This algorithm, it collects a large amount of information about the other routers within the network and about other topologies that may exist. However, he does not know anything about the features of another network topology, that is, he knows the existence of another topology, which means that a package can be routed if needed, but he does not know the number of routers that exist in, the corresponding algorithms etc. Distance vector routing protocols do not have an actual map of the network topology [14].

Distance vector protocols work best in situations where:

- The network is simple and flat and does not require a hierarchical design.
- The administrators do not have enough knowledge to configure and troubleshoot link state protocols.
- Specific types of networks, such as hub-and-spoke networks, are being implemented.
- Worst-case convergence times in a network are not a concern

The major difference between Distance vector and link state algorithms is that in distance vector routing the router share the knowledge of the entire autonomous system, whereas in link state

routing the router share the knowledge of only their neighbor routers in the autonomous system. Table 6 shows a cumulative comparison of routing algorithms.

Finally, there is the Path-vector protocols and, here, we have only one protocol that is using this routing method and its BGP or else the protocol of the Internet. Although it is in the same family of path-vector algorithms, it differs in the way of calculating the weight of our path. Instead of measuring the weight based on how many hops the package will do and always choosing the shortest route in hops, instead it calculates the probability that the route it will choose to send the package will have loops or not. Only one protocol uses this algorithm and this runs the BGP, the protocol of the Internet.

In this case best shown Figure 11, router A advertises reachability to router B. When router B receives this information, it adds itself to the path, and advertises it to router C. Router C adds itself to the path and advertises to router D that the network is reachable in this direction [15].

Router E receives the route advertisement and adds itself to the path as well. However, when router E attempts to advertise that it can reach to router A, router A will reject the advertisement, since the associated path vector contained in the advertisement indicates that router A is already in the path.

Any time a router receives an advertisement in which it is already part of the path, the advertisement is rejected, since accepting the path would effectively result in a routing information loop. Same would've happened if router D would advertise to Router B also since Router B is already in the path also.

Now since we have mentioned all the dynamic protocols and their algorithms, Table 6 illustrates an extensive list of all the characteristics of the dynamic routing protocols.

	RIPv1	RIPv2	IGRP	EIGRP	OSPF	IS-IS	BGP
Interior/Exterior	Interior	Interior	Interior	Interior	Interior	Interior	Exterior
Routing Protocol	Distance Vector	Distance Vector	Distance Vector	Hybrid	Link-state	Link-state	Path Vector
Default Metric	Hop count	Hop count	Bandwidth/Delay	Bandwidth/Delay	Cost	Cost	Multiple Attributes
Administrative	120	120	100	90(Internal) 170(External)	110	115	20(Internal) 200(External)

Distance				al)			nal)
Hopcount Limit	15	15	255(default 100)	224(default 100)	None	None	EBGP Neighbors:1 IBGP Neighbors: None
Convergence	Slow	Slow	Slow	Very Fast	Fast	Fast	Average
Update Timers	30 seconds	30 seconds	90 seconds	Only when change occurs	Only when change occurs	Only when change occurs	Only when change occurs
Update Table	Full table	Full table	Full table	Only Changes	Only Changes	Only Changes	Only Changes
Classless	No	Yes	No	Yes	Yes	Yes	Yes
Support VLSM	No	Yes	No	Yes	Yes	Yes	Yes
Algorithm	Bellman-Ford	Bellman-Ford	Bellman-Ford	DUAL	Dijkstra	Dijkstra	Best Path Algorithm
Update Address	Broadcast	24.0.0.9	224.0.0.10	224.0.0.10	-	-	Unicast
Protocol & Port	UDP port 520	-	IP Protocol 9	IP Protocol 88	Protocol 89	-	TCP port 179

Table 7: Comparison of Dynamic Routing Protocols

3.4.2 **RIPv1 and RIPv2 Protocol**

The RIP protocol is the first dynamic routing protocol that was created. It is a protocol that belongs in the distant vector category protocols and is now only used in obsolete networks or topologies where there are few routers, due to its limitations.

The RIPv1 is a classful protocol that shares information between the routers in the topology using the broadcast IP of the network. The protocol informs the network of any changes that have been made every 30 seconds so as to avoid any network loops that may have been created. At the same time, it can support up to six different routes to the same destination that have exactly the same weight so they will be considered balanced. The big disadvantage of RIPv1 is that it has the limit of 15 hops that the package can do until it is finally dropped. If the packet exceeds this fixed number then on the next router that the algorithm will check the value and will see if its value is 0 then it will drop the packet and will not reach its destination [16].

As computer networks evolved and became more complex and abundant, algorithms had to be adapted to new standards. This is how the RIPv2 protocol was created, where there is an upgraded version of rip 1. Although for the most part they have remained the same, the differences they have, although small, have a great impact. RIPv2 no longer updates the topology every 30 seconds but instead of updating continuously without any changes to the network, it will update the network every time a change is occurs, so the updates now are triggered based. RIPv2 is a classless protocol instead of classful. Also, it is not the broadcast IP that is used, but the multicast for updates to inform the other routers.

Characteristics	RIP Protocol	
	Version 1	Version 2
Algorithm	Bellman-Ford	
Path Selection	Hop count	
Routing	Classful	Classless
Transmission	Broadcast	Multicast
Administrative Distance	120	
Hop count Limit	15	
Authentication	None	MD5
Protocol	UDP	

Table 8: RIPv1 and RIPv2 Comparison

3.4.3 OSPF Protocol

The OSPF protocol has evolved into one of the most widespread and well-known dynamic routing protocols in the world.

The OSPF protocol belongs to the group of link-state protocols and can only work on internal network topologies. Essentially it can do the same as the RIP protocol but much more efficiently and without of course any restrictions that the RIP has and at the same time is more scalable and easier to configure. A very important feature is that the OSPF is an open standard protocol and with the ability of operating on all routers independent of their manufacturer [18][22].

The Shortest Path First (SPF) algorithm, developed based on Dijkstra algorithm which is used to find the most efficient path, to provide a loop-free communication in the topology. Using this algorithm, we achieve the very fast completion of the routing table by each router in the network but also updating the table when there is some change in the topology.

For this reasons OSPF is the best routing protocol for most campus LAN routing and the most popular of all the dynamic protocols for use on the LAN. Based on the above features that OSPF provides, it makes it one of the best protocols on the market. Even from the point of view of the network administrator everything is easier and management [23].

OSPF makes a distinction between routers depending on their role (for example, a designated router or an area border router), and the information that is exchanged between the routers is related to the status of the interfaces. This also makes updating the routing table faster as with the use of Link State Advertisements (LSAs) where an almost immediate update of the network changes is achieved [24].

The two important parts of OSPF architecture are Autonomous Systems (AS) and Areas Autonomous System (AS). The use of the areas in the OSPF are for the better classification of the networks that exist and the better transport of information, so that no loopholes and delays are occurred within the network.

The initial architecture of the OSPF is divided into 2 levels. In the first level is the area 0 of the network, and at the same time the backbone of our entire topology. The second level is all the other areas that are exist in our topology (Areas 1 – 65,535) [19][20].

Based on these characteristics, it is necessary for all the areas that are in the second level to be directly connected or this is not possible indirectly with the area 0. Those areas that are not connected to the area 0 will not be able to communicate with the rest of the topology normally. At the same time all routers operating within the 0 should have the same routing table.

The only downside that comes with the OSPF protocol or better yet it is a prerequisite for it to work effectively is that it needs memory from the router to be able to run the algorithm. with a limited number of resources.

3.4.4 IS-IS Protocol

The IS-IS (Intermediate System - Intermediate System) protocol is very similar to OSPF. Both belong to the link-state protocols and operate only on internal networks. The main difference that it has and we will analyze, is that the IS-IS operates at the level two of the OSI protocol while the OSFF, like the other dynamic routing protocols, operates at level 3 [21][22].

Respectively with the OSPF so here too the IS-IS protocol uses the Dijkstra algorithm to find all the possible destinations in the topology and to calculate which of them is the most suitable based on the weight of each route. Information about the status of the network is shared with all routers within the topology and everyone finally has a map of what the network is like.

Having this information, the routers decide which route the packages will choose to reach their final destination. Here is the advantage of link-state protocols. All routers are aware of the network status so the routers can choose routes that have specific criteria (bandwidth). Respectively, here is the disadvantage of liquid-state algorithms. Because all the routers know about the whole network then as our topography grows then the map that the routers have, As a result more resources are required from the machines but at the same time they will need to be updated more often on the network and will receive It takes more time to calculate the best trails and there will be more routes.

This lack of scalability indicates that these protocols are not recommended for large-scale internet use [26][27].The list of features provided by the IS-IS protocol is lengthy. The followings are some of its key features, in a network topology:

- It supports hierarchical routing structure.
- It is a classless routing protocol, hence supports CIDR, VLSM, and discontinuous networks.
- It floods new information in the network very rapidly.
- Its convergence time is very fast.
- It is very scalable routing protocol.
- It provides flexible timer tuning.
- It uses Cost as the default metric but can also use optional metrics, such as Delay, Expense, and Error.
- It uses Dijkstra Shortest Path First algorithm to calculate the best path.
- It is less widely implemented on router platforms.
- It runs on data link layer of OSI model.

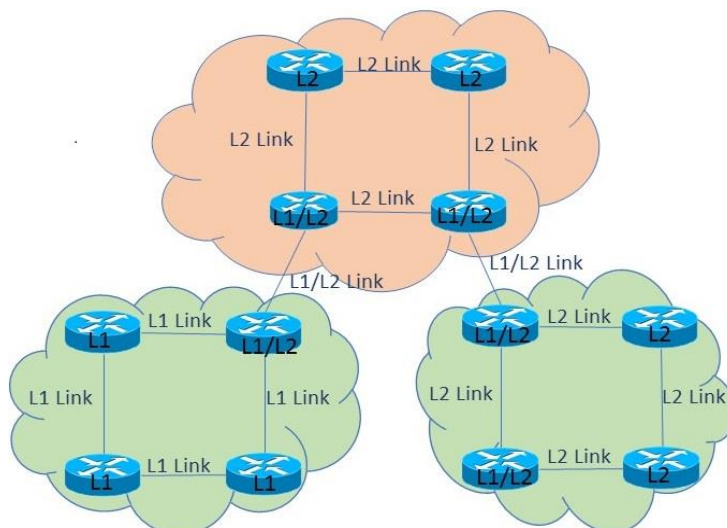


Figure 12: IS-IS Router's Levels

- It uses IS-IS Hello packets (IIHs) PDUs to form adjacencies between routers.
- It supports authentication (MD5) that allows configuring a password for a specified link, for an area/entire domain.

As we mentioned earlier IS-IS supports the hierarchal structure. To make a hierarchal structure, IS-IS protocol uses layered hierarchy [21]. IS-IS protocol has two layered hierarchy: Level 1 and Level 2.

- Level 1 hierarchy: All L1 routers know only about the local network and cannot communicate with another network. To achieve they need an L2 router.
- Level 2 hierarchy: L2 routers have the ability to communicate with other networks with which they are connected directly or indirectly. They have available all the information about the network to which they belong but also about other networks with which they communicate
- Level1/Level2 hierarchy: An IS-IS router can act as an L1 and an L2 router at the same time. These routers are referred as L1/L2 routers. An L1/L2 router may have neighbors of any area. In addition, an L1/L2 router has two separate LSDBs: level-1 LSDB for L1 routers and level-2 LSDB for L2 routers [28][29]

Unlike other *IP* routing protocols, which typically run on TCP, UDP, or *IP*, which are OSI Layer 3 or Layer 4 protocols, IS-IS runs directly on the data link layer (Layer 2). In this section, it was overviewed the basics of IS-IS protocol. The entire concept of the IS-IS protocol is so lengthy only a minor section of IS-IS protocol characteristics it was discussed [30][31].

3.4.5 BGP, EBGP and IBGP

BGP is considered the most difficult protocol to comprehend and understand of course since its purpose is enable communication for the whole world. At the same time, it is the most different protocol from the rest and is used extensively on the internet and not in internal network topologies [32].

BPG is used to exchange information between internal topologies that have different protocols. Their most common application is when we connect to our ISP through one of our devices. The most important thing is that the BPG does not have scalability problems since it works all over the world.

Unlike IGP protocols where everyone creates a topology map and calculates the weights each one individually, in BGP and the other vector-based protocols a router exchanges information only with the neighboring routers and information about who it is destined for.

Therefore, in terms of scalability the vector-based algorithms are considered much better than



Figure 13: EBGP & IBGP Sessions

the link-states. So, these protocols are preferred for use on the backbone of the internet. If the connection between two topologies is lost through the BGP, each area must contain a router where at least one of its interfaces will be connected to the internet.

When the connection is established a BGP router will send all the routes from his local routing table to that peer using a specific message called UPDATE. Then the peer uses the context of these messages to add new routes to his own local routing table, and if learn more than one route to the same destination, it will run a decision process to decide which is the most preferable. Moreover, BGP can be separated to two more subclasses (Figure 13). Just as we know an IGP process peers are between neighbors in the same AS, and an EGP is a process between neighbors in different peers [33].

BGP uses the same concept: If a BGP session is established between two neighbors in different autonomous systems, the session is external BGP (EBGP), and if the session is established between two neighbors in the same AS, the session is internal BGP (IBGP). To understand IBGP, let us first look at external BGP (EBGP). The Internet routing world consists of many autonomous systems which are interconnected. Each AS consists of multiple routers. EBGP is the version of BGP that is used to exchange BGP routing updates between two different AS's. EBGP is implemented on the edge BGP router that provides interconnection to another AS's [34].

Internal BGP, IBGP is the protocol used between the routers in the same autonomous system (AS). IBGP is used to provide information to your internal routers. IBGP requires all the devices in same AS to be aware of every other device in their AS for prefix learning. Having seen both EBGP and IBGP protocols, Table 7 compares them on various parameters [35].

Characteristics	IBGP	EBGP
Topology	Doesn't require full mesh	Requires full mesh
Neighbors IP	Both the routers forming EBGP need to be in separated AS	Both the routers forming EBGP need to be in the same AS
Route Advertisement	A route learned from EBGP will be advertised to another IBGP or EBGP	A route learned from IBGP will be not advertised to another IBGP
Administrative Distance	EBGP routes have AD of 20	IBGP routes have AD of 200

Table 9: IBGP and EBGP differences

3.4.6 Advantages and disadvantages

Having made a detailed analysis in both static and dynamic routing, we can understand that in any case there will be some positive and some negative features in the topology they are implemented [14].

We can see a list of the pros and cons of each method in Table 8[28].

Basis for Comparison	Static Routing	Dynamic Routing
Configuration	Manual	Automatic
Routes	User defined	Routes are updated according to change in topology.
Routing algorithms	Doesn't employ complex routing algorithms.	Uses complex routing algorithms to perform routing operations.
Implemented in	Small networks	Large networks
Link failure	Link failure obstructs the rerouting.	Link failure doesn't affect the rerouting.
Security	Provides high security.	Less secure due to sending broadcasts and multicasts.
Routing protocols	No routing protocols are indulged in the process.	Routing protocols such as RIP, EIGRP, etc. are involved in the routing process.
Additional resources	Not required	Needs additional resources to store the information.

Table 10: Static Routing versus Dynamic Routing

Table 8, the difference between static and dynamic routing. In summary, static routing is an offshoot which should only be recommended in cases where our network topology is small enough, and it will not grow and if it grows only at a very small percentage. The administrator must have the necessary networking skills and experience to decide whether to implement static routes in his topology based in the needs and the demands someone wants from the topology.

3.5 The JUNOS OS and its Characteristics

The core of Juniper devices is the JUNOS OS. JUNOS OS is based on the FreeBSD UNIX operating system (OS) it's the software that runs networking and security devices from Juniper Networks. It is an operating system that is used in Juniper's routing, switching and security devices [36].

JUNOS OS was formerly branded as Juniper JUNOS, and its commonly referred to as simply JUNOS. Some of the key benefits JUNOS OS include:

- Modular design: The operation performed by the system is completely isolated from the rest in order if any function malfunctions then it will not be able to affect the rest, resulting in a safer environment
- Compatibility: All the products it produces are functional with each other and have in their base the same functional so they will always be able to communicate on a basic level.

Which are some of the differences that distinguish JUNOS from other complex software architectures. In addition, through the interface it offers to the user, he can configure the device according to his needs. It supports two types of command modes the Operational Mode and the Configuration Mode.

3.5.1 Configuring a JUNOS router

As with other operating systems, JUNOS also has different levels of device configuration. The JUNOS CLI configuration command nodes are [37]:

1. UNIX BSD SHELL: JUNOS as in other cases, here too, when the user enters the device for the first time through his terminal, he must have the necessary transistors. For example, when someone login for the first time will come across something like is shown in Figure 14.



Figure 14: Login in a JUNOS Device

When someone connects to the device it enters as a user root. And it will be at the first level which is the UNIX BSD shell (Figure 15). This is the level with the fewest rights and not much can be done on the router. The UNIX BSD shell is designated with the “%” prompt. Here you can enter

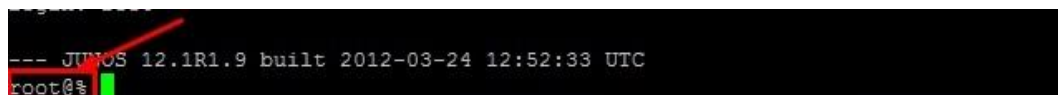


Figure 15: The UNIX BSD Shell

standard UNIX commands.

2. **OPERATIONAL MODE:** The next level is the operational level. In these modes the commands that are available are more and some functions such as ping, telnet and traceroute can be performed. To enter the router is typing cli and pressing enter. The shell is designated with the “>” prompt.

```
login: root
--- JUNOS 12.1R1.9 built 2012-03-24 12:52:33 UTC
root@% cli
root>
```

Figure 16: Operational Mode

3. **CONFIGURATION MODE:** The configuration level and last level. At this level the user has all the rights and can make any changes he wants on the router. In addition, when someone goes to a level that has more rights than the previous one, at the same time he can also perform commands that were available at the above level. (Figure 17). Configuration is designated with the “#” prompt.

```
root> edit
Entering configuration mode
[edit]
root#
```

Figure 17: Configuration Mode

Now you have entered into the configuration mode, and you are placed at the most abstract section of the configuration. At this level the root user has the right to change anything inside the router. For example, with the command “edit interfaces”, will move the user to the interface level. Once you’re at the interface level, you will only see interface-level information. This change of available commands depending on where you are on the router is very useful in an environment that is so vast.

As it is understood when someone goes “deeper” in the levels (Figure 18) of a JUNOS device, more privileges have been gained on the device so to allow for modifications and customization of more router features.

When you enter the configuration mode, we have to configure the root password. Although it seems logical the reason that asks you to set a code is completely different from what we think [37]. Essentially at this level the root user has unrestricted access and can make

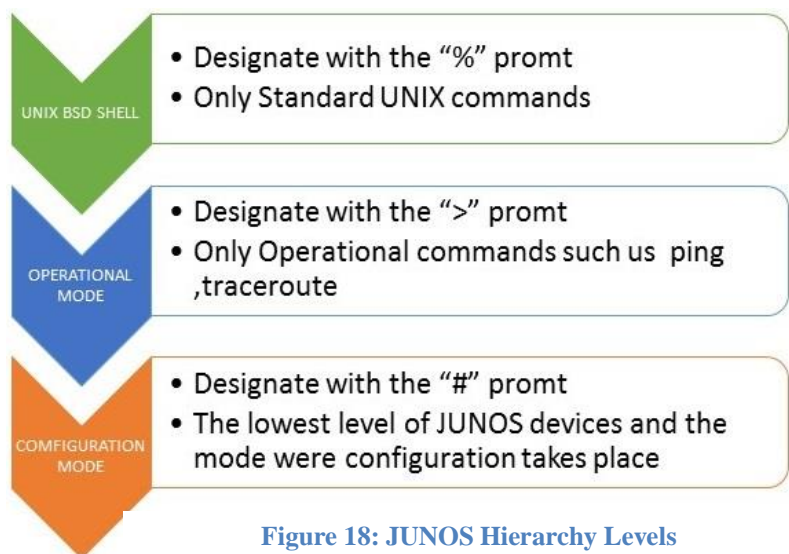


Figure 18: JUNOS Hierarchy Levels

the changes he wants without any restrictions. This is because when someone makes changes to the system and wants to finalize these changes, the router asks them to set a code for the system before saving the changes and makes it active. Any changes made to the system will not be valid until they are saved. At the same time, the operating system holds a file after the last 50 stored files with the changes that have been made. This in case the changes that are made make the system malfunction as a result of which you want to return to the previous state Otherwise it will show this message when you try to commit your configuration message

” Missing mandatory statement: 'root-authentication 'error: commit failed: (missing statements)”

To set the root password you must be in the configuration mode and execute this command

root#” set system root-authentication plain-text-password”

In this section we have presented the Basic Juniper Router Commands Modes and the initial configuration of a Juniper Router.

3.5.2 Cisco vs JUNOS

JUNOS OS and CISCO IOS are the two major contenders in the networking environment and the ultimate choice for anyone who thinks about getting evolved with networking, even on an introductory level.

IOS traditionally is a monolithic system which based on its architecture was built so that when some applications run on a machine then all of them will have access to the system resources together. This resulted in problems if there were problems with one application then it would affect the whole system and with it and the other applications they would run simultaneously on the same system. On the other side IOS is the most common and wide spread OS out there make it the best choice for newcomers as its offers them plenty of material they can learn from [38].

In contrast, in JUNOS. where it has been built later and its architecture is different. It is based on open source kernel code. Each function is isolated from the others with its own memory and functions, thus protecting and making the system as a whole safer. Moreover, it's the newest from two and thus make it the more modern and modular from IOS.

But their major difference is considered their operational performance. The reality is, IOS is old. JUNOS on the other hand stand out is, as we said, the architect itself is different from IOS, on which it is built. It is newer and configurable so it is easier to find errors and problems in the system, and you know exactly where the problem is located in contrast to IOS. Reason for that is because when JUNOS was built it was designed by networking engineers for networking engineers and in a way to contain all the benefits of IOS and none of his problems [39].

3.6 Description of the experiments

This sub-section is dedicated to explaining in brief the essence of the laboratories courses that will take place in the next chapter and describe the purpose of their laboratories.

3.6.1 Static Routing

In chapter 3 of this thesis the experimental part will take place using the virtual environment of EVE-NG to create a topology of routers with the operating system JUNOS. The purpose of this exercise is to create a topology from a series of routers that will communicate with each other via static routing using the JUNOS OS.

3.6.2 OSPF

In this topology, dynamic routing comes in place and the scope is for the student to be able to connect all the routers that will belong to different areas between them and along with the backbone area. In addition, one of the regions will not be directly connected to the backbone area but via a virtual link as we shall see in chapter 3.

3.6.3 RIPv2

Like the previous exercise there will be 3 areas between them, and the purpose will be to connect the routers between them so that they can communicate. This will be the simplest form of exercise where we will be using a dynamic protocol.

3.6.4 IS-IS

Respectively to OSPF exercise here things will be somewhat different there will be no central area, but routers and levels will be separated, and similar roaming routers would communicate to each other through the protocol

3.6.5 BGP

Finally, the most difficult and the most complex exercise of all. The creation of a topology using the BGP protocol. It's an exercise where the usage of all the knowledge we have gained so far will come handy in order to create 3 separate areas. Each one of them will be running one or more of the protocols we have seen before (OSPF, RIPv2) and these areas will be linked together through a fourth area in which BGP protocols will be running in order to a communication between areas with different protocols.

4 Chapter 3 : Development of Laboratory Exercises Using JUNOS

Packet routing is considered the basis for all networking systems. As mentioned above, the devices that are responsible for the correct routing of packets are the routers. All the work required to find the right path to the destination that the packages should go to is done externally by the dynamic routing protocols.

Of course, this work can be done with the use of static routing where the person in charge of the topology should define the paths of the packages. In most cases the solution of both combinations is followed where depending on the needs presented the case of routing.

But the routers do not know in advance what is the right path to send packages to their destination and in most cases, there are more than one route from the sender to the recipient. Five different topologies will be shown in detail and each one of them will be implemented with a routing protocol and used to configure IPv4 static and dynamic routes.

4.1 Static Route Topology

In contrast to dynamic routing, statics is the simplest form of packet routing, where along with low overhead it is an attractive solution for beginners and simple topology architectures.

A plethora of commands will be presented and explained in detail below, but also the impact they will have on the communication of the routers.

Figure 19 illustrates the topology for the first exercise. It consists of 4 routers connected to a linear way through ethernet cable, and the goal is to configure them so the routers Olive and Olive2 can communicate with each other.

As a first step in this case is to define the interfaces of all routers with a specific IP of our preference (as shown in Figure 19). This will be the basic configuration of the routers which will be repeated for all routers of this exercise, as well as in the following sections.

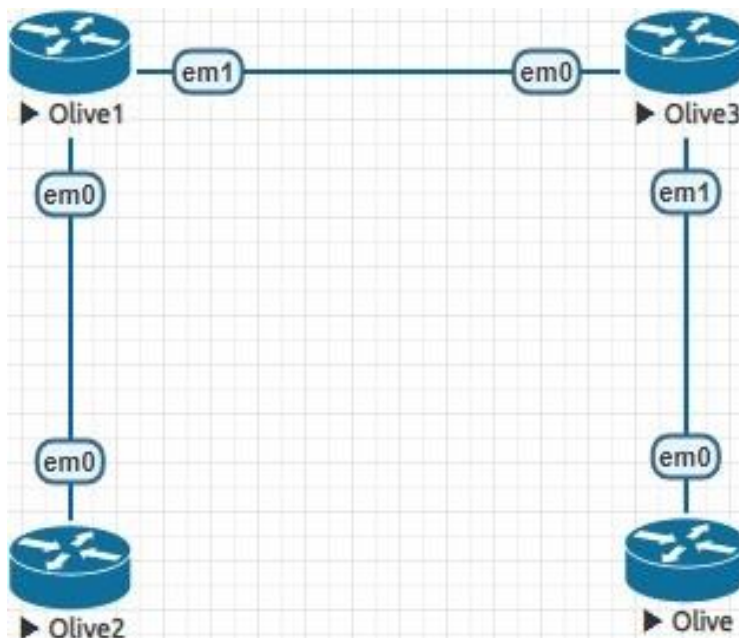


Figure 19: Static Topology

The basic configuration for Olive2 interface em0 is:

Olive2, Em0:” set interface em0 unit 0 family inet address 192.168.1.1/24”

Respectively for *Olive1* interface *em0*:

Olive1, Em0:” set interface em0 unit 0 family inet address 192.168.1.2/24”

This command defines the physical interface (*interface em0*) the logical interface (*unit 0*) the type of the IP (*family inet* for IPv4 and *family inet6* for IPv6) and which is the IP (*address 192.168.1.2/24*). In this case the network 192.168.1.0/24 was created between the two routers in order to communicate with each other. Similarly, the configuration for *Olive1* interface *em1* and *Olive3* interface *em0* are programmed respectively:

Olive1, Em1:” set interface em1 unit 0 family inet address 192.168.2.1/24”

Olive2, Em0:” set interface em0 unit 0 family inet address 192.168.2.2/24”

And for *Olive3* *em1* and *Olive* *em0* the configuration is:

Olive3, Em1:” set interface em1 unit 0 family inet address 192.168.3.1/24”

Olive, Em0:” set interface em0 unit 0 family inet address 192.168.3.2/24”

Now three networks have been created in the topology (*Figure 20*). Thus, now the routers can communicate but only with their directly connected neighbor. If they do try to reach to a router which is not in their immediate neighbor, they will fail because they do not know how to reach the destination.

For instance, *Olive3* can only reach *Olive1* and *Olive* but cannot communicate with *Olive2*. In order to achieve the desired communication, each router must be statically configured. To enter a specific path into the routing table, the route should be defined explicitly as static address and also the address of the receiver router must be associated with it. All packets that are intended for the specific router and passed

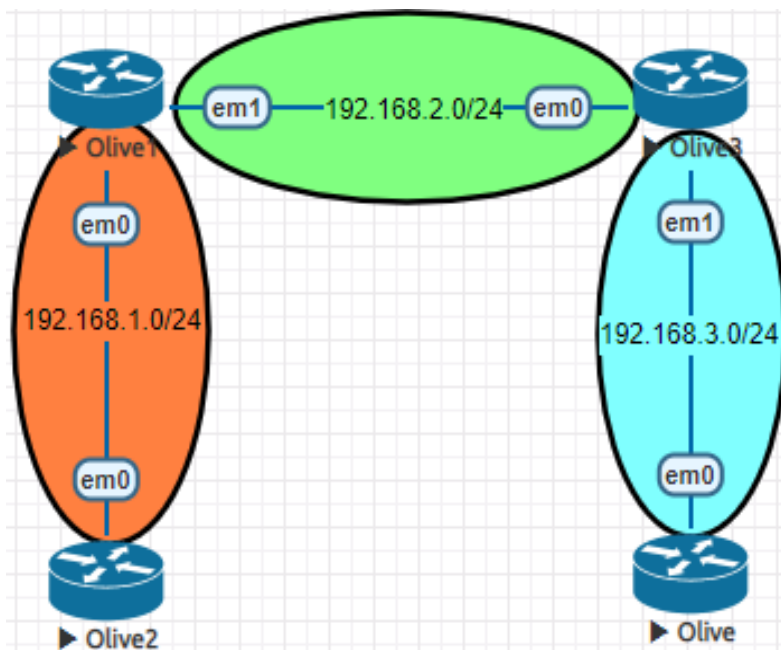


Figure 20: Network Areas

through the router that we entered the static route will be forwarded to the specific router. The static configuration for *Olive2* to reach the network *192.168.2.0/24* is:

Olive2: “*set routing-options static route 192.168.2.0/24 next-hop 192.168.1.1*”

This command sets a routing option in which all the packets that are going for the *192.168.2.0/24* network will be transmitted through the interface with IP *192.168.1.1/24*. Additionally, for *Olive2* to reach the *192.168.3.0/24* network it needed to add a second interface, from which through it the traffic will be directed to, so:

Olive2: “*set routing-options static route 192.168.2.0/24 next-hop 192.168.1.1 next-hop 192.168.2.2*”

Similar on router *Olive1*:

Olive1: “*set routing-options static route 192.168.3.0/24 next-hop 192.168.2.2*”

On router *Olive3*:

Olive 3: “*set routing-options static route 192.168.1.0/24 next-hop 192.168.2.1*”

On router *Olive*:

Olive: “*set routing-options static route 192.168.2.0/24 next-hop 192.168.3.1*”

Olive: “*set routing-options static route 192.168.1.0/24 next-hop 192.168.3.1 next-hop 192.168.2.1*”

Having completed all the configuration properly, the routers will be able to ping and direct traffic to any other router (*Figure 21*).

```
root> ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2): 56 data bytes
64 bytes from 192.168.3.2: icmp_seq=0 ttl=62 time=9.339 ms
64 bytes from 192.168.3.2: icmp_seq=1 ttl=62 time=11.031 ms
64 bytes from 192.168.3.2: icmp_seq=2 ttl=62 time=8.759 ms
^C
--- 192.168.3.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 8.759/9.710/11.031/0.964 ms
```

Figure 21: Successful ping from Olive2 to Olive

4.2 RIPv2 Topology

When it comes to internal routing protocols, *RIPv2* is considered to be the commonly dynamic protocol today. In addition, the protocol is considered to have a small degree of learning difficulty, so it makes it a very attractive solution, as it is destined only for smaller networks with no more than 15 nodes. In this network topology (*Figure 22*) the configuration must be made for 7 routers which are divided into two groups: *Group1* and *Group2*. In general, very good feature of this protocol is that a topology can run both versions of the protocol (*RIPv1* and *RIPv2*) simultaneously without any problems in the communication of the routers. This feature makes it very easy to transfer a topology from *RIPv1* to *RIPv2* without.

The first step is to set the basic configuration for the network interfaces of the routers in the topology illustrated in *Figure 22*. Having done that the next step is to configure the routers to advertise their routes. Although somewhat controversial, when the protocol is activated on the routers, the interfaces will not be adequately advertised through the *RIPv2*. This is mainly done for security reasons. In order for the interfaces to be advertised through the protocol and for the communication and transfer of packages to begin, it is necessary to create a routing policy, where it will advertise the known interfaces through the protocol to the adjacent routers. In order to achieve communication with all the routers, a routing policy will have to be created again that will report the advertising of all the interfaces that have been learned through the protocol.

This is because it is predetermined by the protocol itself not to advertise its interfaces to other routers.

Figure 22 illustrates the way of evaluation of a policy. Specifically, each policy consists of specific terms, where each of them consists of specific conditions, that each interface must meet so that the rule can then advertise them through the protocol. The application of specific rules works as well as filters for incoming or outgoing packages, where you can choose whether you will receive information if it meets certain criteria. A routing policy can be made up by thousands of terms. As the route comes in, the policy is invoked. Each route is evaluated against the policy as follows:

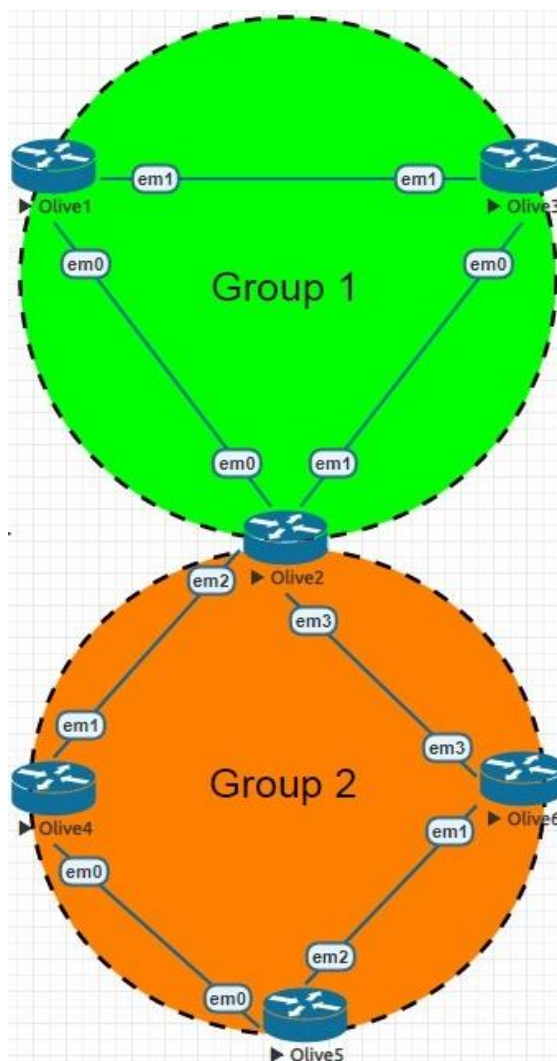


Figure 22: RIPv2 Topology

- **Step 1:** The route is evaluated against the first term. If the route matches the conditions specified, that action is taken. If the action is to accept or reject the route, that action is taken and the evaluation of the route ends. If a second term action is specified, or no action is specified, or if the route does not match, the evaluation continues in Step 2.
- **Step 2:** The route is evaluated against the second term. If it matches, the specified action is followed. If the action is to accept or reject the route, that action is taken and the evaluation of the route ends. Otherwise, the evaluation continues as described in Step 3.
- **Step 3:** If none of the terms of the policy are a match for the route in question, the next policy is evaluated, and so on until the default policy action is taken.

The protocol comes with a predefined policy where it is universally applied and affected by some variables. What protocols are running on the interfaces. In general, each protocol has its own predefined policy. In addition, an important feature is whether the rule applies to packets that are imported or exported to the router.

Table 3-1 synopsizes the default actions that some dynamic protocols have.

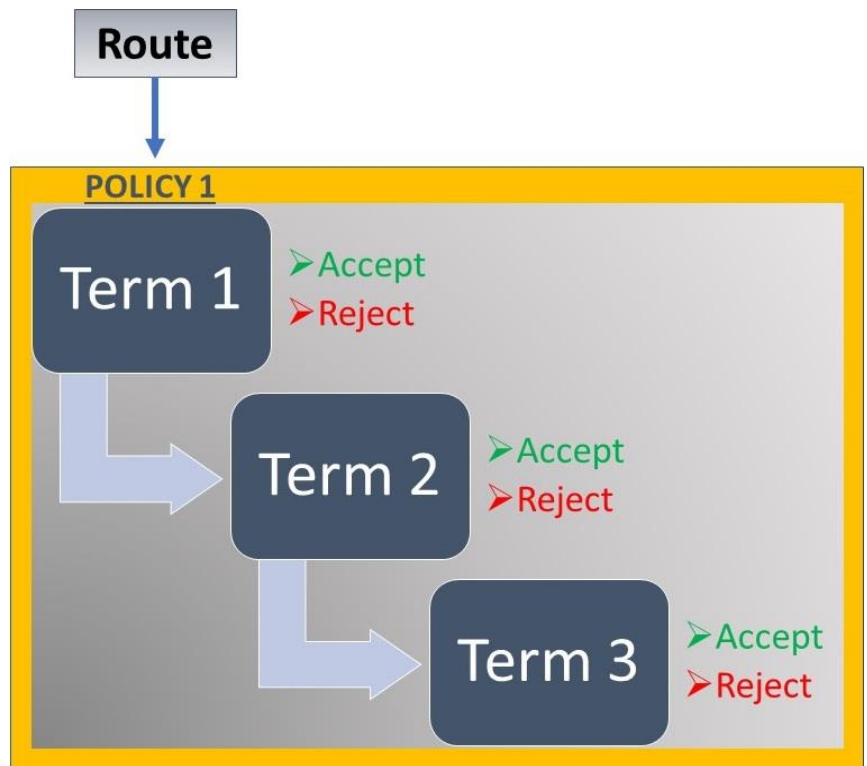


Figure 23: JUNOS Policy

In our example (Figure 22), the routing policy term configuration for *Olive1* is:

Olive1: “*edit policy-options policy-statements adv-routes term 1*”

Olive1: “*set from protocol direct*”

Olive1: “*set from protocol rip*”

Olive1: “*set then accept*”

The first command defines a routing policy statement (*routing-options policy-statements*) with a name of our choosing (*adv-routes*) and configures the first term (*term 1*). The next three commands are the conditions a route must match in order to be advertised. In this case, the route must be directly connected to *Olive1(protocol direct)*, must be running the protocol *RIPv2*

(*protocol rip*) and if it matches the previous requirements then it is accepted (*then accept*) and it is transmitted to other nodes. The above statements are configured in the same manner on all the other routers with no exception.

Having set the routing policy, the next step is to configure the routes in order to communicate not only with their neighbors but with the other routers as well, using the dynamic protocol RIPv2. The configuration for *Olive1* is:

Olive1: “*edit protocols rip group Group1*”

Olive1: “*set export adv-routes*”

<i>Protocols</i>	<i>Default Action</i>
<i>RIP</i>	<u><i>Import:</i></u> Accept all routes received on RIP-enabled interfaces.
	<u><i>Export:</i></u> Do not export any RIP routes.
<i>IS-IS</i>	<u><i>Import:</i></u> Reject everything. (The protocol uses flooding to announce local routes and any learned routes.)
	<u><i>Export:</i></u> Reject everything. (The protocol uses flooding to announce local routes and any learned routes.)
<i>OSPF</i>	<u><i>Import:</i></u> Policies can't be used for imported routes.
	<u><i>Export:</i></u> Export all routes learned via OSPF and all direct routes associated with the OSPF-enabled interfaces.
<i>BGP</i>	<u><i>Import:</i></u> Accept all routes learned from BGP neighbors.
	<u><i>Export:</i></u> all active routes learned via BGP to all BGP neighbors.

Table 11: Default Actions of Routing Protocols

Olive1: “*set neighbor em0 receive version-2*”

Olive1: “*set neighbor em1 receive version-2*”

The first command selects the protocol (*edit protocols rip*) that is going to be used for the advertisement and sets the name of the group (*set group Group1*) the router is belonging to. The next statement (*export adv-routes*) defines that this group is going to export routes that are matching this policy. Lastly the two remaining commands add the two interfaces (*em0*, *em1*) into the *rip group Group1* and specify that the *em0* and *em1* interfaces that are facing the *Olive2* and *Olive3* respectively will accept (*receive version-2*) only RIPv2 packets. For *Olive2* the configuration is similar, but requires one extra step. The reason is because it's the router that connects the two groups of the network topology, and two of its interfaces belong to Group1 and the other two in Group2.

The extra step for *Olive2* is to create a second group where the interface *em2* and *em3* are going to belong side with the routers *Olive4*, *Olive5*, *Olive6*:

Group 2 Routers: “*edit protocols rip group Group2*”

Group 2 Routers: “*set export adv-routes*”

Group 2 Routers: “*set neighbor em2 receive version-2*”

Group 2 Routers: “*set neighbor em3 receive version-2*”

Now *Olive2* is using the RIPv2 protocol and the *adv-routes* policy to advertise routes between his neighbors from two different groups. Last step is the configuration of *Olive4*, *Olive5*, *Olive6* routers which are similar, and the only difference is the name of the interfaces each router is using to communicate.

For *Olive4*:

Olive4: “*edit protocols rip group Group2*”

Olive4: “*set export adv-routes*”

Olive4: “*set neighbor em0 receive version-2*”

Olive4: “*set neighbor em1 receive version-2*”

Respectively for *Olive5*:

Olive5: “*edit protocols rip group Group2*”

Olive5: “*set export adv-routes*”

Olive5: “*set neighbor em0 receive version-2*”

Olive5: “*set neighbor em2 receive version-2*”

Similar for *Olive6*:

Olive6: “*edit protocols rip group Group2*”

Olive6: “set export adv-routes”

Olive6: “set neighbor em3 receive version-2”

Olive6: “set neighbor em1 receive version-2”

With the completion of all router’s configurations, now each one should be able to see (i.e., ping) any other router in the network in any group (Figure 24).

```

root> ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=62 time=8.237 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=62 time=8.453 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=62 time=9.989 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 8.237/8.893/9.989/0.780 ms
    
```

Figure 24: Ping from Olive 5 to Olive1

4.3 3.3) OSPF Topology

The *OSPF* routing protocol has replaced the rip protocol in most industrial and corporate networks. The reason behind this change is that the protocol makes better management of the resources of the routers and more efficient operation in the whole infrastructure with faster communication, more possibilities and less delays. An additional feature that stands out is when a change is occurs to a router, it is shared throughout the rest of the network so that all routers have

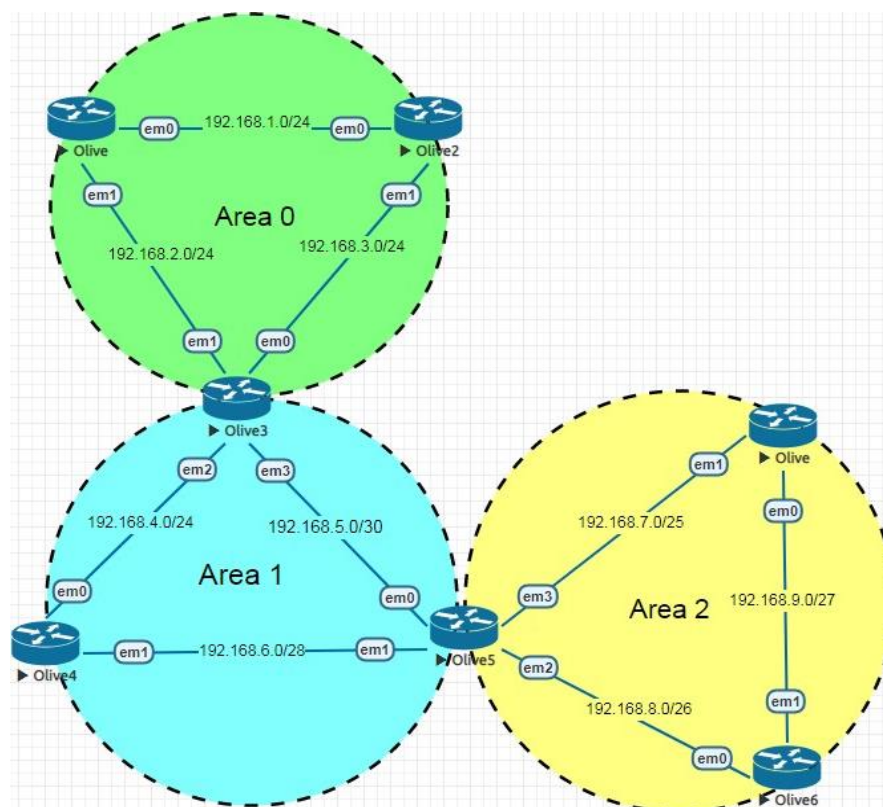


Figure 25: OSPF Topology

routing table information in the network.

This is in contrast to the RIP protocol, where instead of constant updates on the status of the topology between the routers even when no change has been made, resulting in overhead and delays in communication between routers, in the OSPF updates will only occur when there is a corresponding change in the network status.

As it is mentioned in section 2.4.3 in *OSPF* a single *Autonomous System (AS)* can be divided into smaller groups called areas, thus reducing the number of advertisements the *OSPF* is sending on the network and it reduces the size of the topology database that each router must maintain. An area can be best described as a number of routers and terminals that are in the same group and have identical topology databases.

Figure 25 displays a networking topology where there are three areas. Area0 which is also called the backbone area, Area1 and Area2. The Areas 1 and 2 are similar and consist of 3 routers each. The main area of an AS, called the backbone area has a unique ability and is always assigned the area ID *0.0.0.0* or *Area0*.

All other areas that exist or will exist in the network should be connected in some way to the central area, at least one interface from one router should be connected to another area outside of Area 0. The router that is located in between 2 areas and connects them is called area border routers (*ABRs*).

These routers are unique and are of great importance for the proper operation of an *OSPF* topology. The reason is that in a topology where we have the central area and several adjacent areas that are all connected to the central area, all data traffic that is intended to go to another area (not in area 0) will have to go through the central area first. *ABR*'s routers are responsible for routing traffic to the central area and are maintaining a unique database with all the adjacent areas that are connected. In Figure 25 only *Olive3* and *Olive5* meet the standards to be *ABRs* even if *Olive5* isn't directly connected to the backbone area.

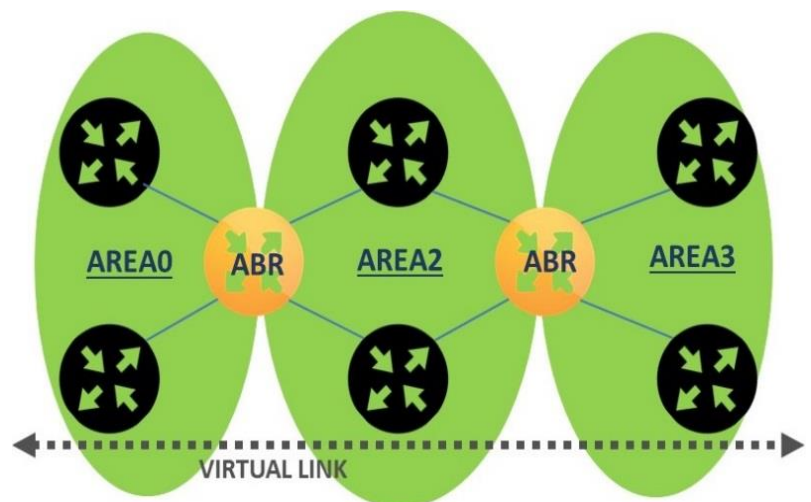


Figure 26: OSPF Virtual Link

Of course as it is logical in cases where we have too many areas that should be connected to the central area and it is not physically possible, then there is the option to create a virtual connection between that area and the central. Essentially what happens is the area that cannot be connected to the central, will be connected to an area that is directly connected to the central and will communicate through it.

This way it uses virtual links which use the ABRs routers to direct the packets into the central area. In *Figure 26* a virtual link is established between *Area3* and the backbone *Area0* through *Area2*. The virtual link transits *Area2*. All the traffic destined for other areas is routed through *Area2* to the backbone area and then to the appropriate *ABR*. All inbound traffic destined for *Area3* is routed to the backbone area and then through *Area2*. Without the virtual link all the traffic will be dropped. The first step in the *OSPF* network topology is the basic configuration of the interfaces in their appropriate network as it is shown in *Figure 26*. Next step is to enable *OSPF* on a network which mean to activate in each interface inside the network. To enable *OSPF* is our routers, the configuration is for *Olive*:

Olive: “*edit protocols ospf area 0*”

Olive: “*set interface em0*”

Olive: “*set interface em1*”

Similar for *Olive2*:

Olive2: “*edit protocols ospf area 0*”

Olive2: “*set interface em0*”

Olive2: “*set interface em1*”

The first command selects the protocol (*edit protocols ospf*) that is going to be used for the advertisement and sets the area (*area0*) where the routers interface will belong. The next two commands specify the interfaces (*em0*, *em1*) that are going to be advertised by the *OSPF* protocol to the neighbor routers. In this topology it isn't necessary to create a routing policy, because the default routing policy allows the advertisements of routes that are learned from *OSPF* to other routers inside the area.

For *Olive3* which is an ABRs router the configuration is alike to *Olive1* and *Olive2* but only for the interfaces *em0* and *em1* which are belonging to the backbone area. The interfaces *em2* and *em3* are going to be configured into the *Area1*. The configuration for *Olive3* is:

Olive3: “*edit protocols ospf area 0*”

Olive3: “*set interface em0*”

Olive3: “*set interface em1*”

Olive3: “*top*”

Olive3: “*edit protocols ospf area 1*”

Olive3: “*set interface em2*”

Olive3: “*set interface em3*”

These commands have determined in which area the interfaces will belong to. The “*top*” command is used to get back into the starting position of configuration command mode. Respectively the configuration for *Olive4* is:

Olive4: “*edit protocols ospf area 1*”

Olive4: “*set interface em0*”

Olive4: “*set interface em1*”

And for *Olive5* which is also an ABRs the configuration is:

Olive5: “*edit protocols ospf area 1*”

Olive5: “*set interface em0*”

Olive5: “*set interface em1*”

Olive5: “*top*”

Olive5: “*edit protocols ospf area 2*”

Olive5: “*set interface em2*”

Olive5: “*set interface em3*”

Accordingly, the configuration for *Olive* is:

Olive: “*edit protocols ospf area 2*”

Olive: “*set interface em0*”

Olive: “*set interface em1*”

And for *Olive6*:

Olive6: “*edit protocols ospf area 2*”

Olive6: “*set interface em0*”

Olive6: “*set interface em1*”

Last step in the topology is the configuration of the virtual link on routers *Olive3* and *Olive5*. For *Olive3*:

Olive3: “*set routing options router-id 192.168.10.1*”

Olive3: “*edit protocols ospf area 0*”

Olive3: “*set virtual-link neighbor-id 192.168.10.2 transit-area 0.0.0.1*”

The first command sets the router identifier (*router-id 192.168.10.1*) it is used by OSPF and BGP to identify the routing device from which a packet originated. The next commands specify dynamic protocol and the area (*protocols ospf area 0*) and the last command set a virtual link and specifies the IP address (*neighbor-id 192.168.10.2*) of the routing device (*Olive5*) at the other end of the virtual link and the area (*transit-area 1*) through which the virtual link transits.

Respectively for *Olive5*:

Olive5: “*set routing options router-id 192.168.10.2*”

Olive5: “*edit protocols ospf area 0*”

Olive5: “*set virtual-link neighbor-id 192.168.10.10 transit-area 0.0.0.1*”

Having completed all the configuration properly, the routes will be able to ping and direct traffic to any other route in any area, backbone or not (Figure 27).

```
root> ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=62 time=11.281 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=62 time=12.602 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=62 time=12.241 ms
^C
--- 192.168.1.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 11.281/12.041/12.602/0.557 ms
```

Figure 27: Ping Olive from Olive

4.4 Intermediate System-Intermediate System (IS-IS) Topology

The *IS-IS* protocol is an *Interior Gateway Protocol (IGP)* that uses link-state information to make routing decisions.

To achieve this protocol uses the algorithm to find the most efficient routes for sending data the protocol is very much in common with the *OSPF* protocol, and both have quick updates on any changes that may be made to the network. The big difference of *IS-IS* from the others is that unlike the rest it does not work in the third level, meaning in that of IP, but in the second level in that of the ethernet header. This change, although seemingly small has a huge impact on the networking since now in contrast to the *OSPF* where we stated which interfaces will be in which area is now the whole router and not a part of it .This feature allows it to not need a central area like the *OSPF* and to be able to grow in size without affecting its performance, that’s why many ISP prefer *IS-IS* as their backbone protocol .In *Figure 28* an overview of what *IS-IS* is and how it works is illustrated.

An *IS-IS* network topology (*Figure 28*) is a single *AS*, also called a routing domain, a single *AS* can be divided into smaller groups called areas. Routing between areas is organized hierarchically, allowing a domain to be administratively divided into smaller areas.

The default policy when IS-IS is enabled on an interface both levels (*Level 1* and *Level 2*) are enabled. To specify that an interface is on a *Level 1* link, it must disable *Level 2*. To specify that an interface is on a *Level 2* link, disable *Level 1*. It is possible to disable a level on the entire device or per-interface.

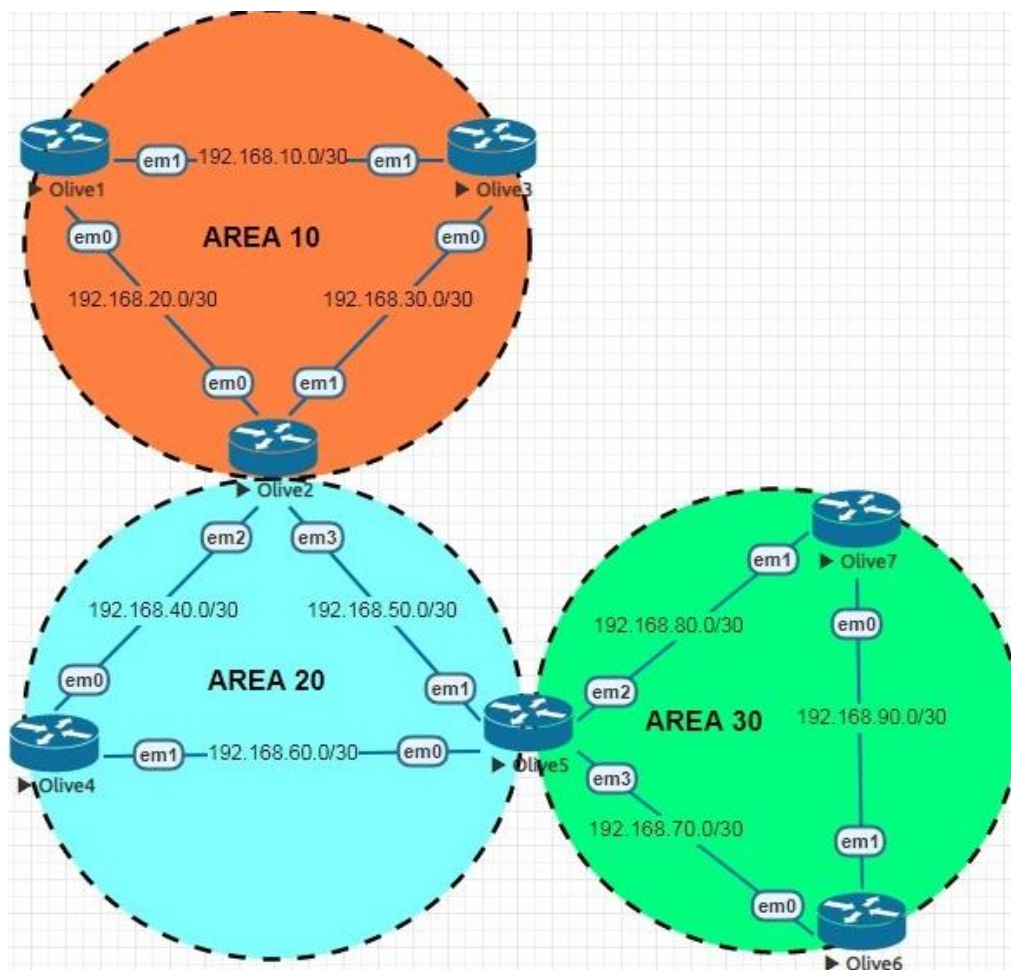


Figure 28: IS-IS Topology

The first step is to set the basic configuration for the network interfaces that connect the routers between them as it is illustrated in *Figure 28*. The basic configuration for *Olive1* is:

Olive: “set interface em1 unit 0 family inet address 192.168.10.1/30”

Olive1: “set interface em1 unit 0 family iso”

Olive1: “set interface em0 unit 0 family inet address 192.168.20.1/30”

Olive: “set interface em0 unit 0 family iso”

Olive1: “set interface lo0 unit 0 family inet address 1.1.1.1/32”

Olive1: “set interface em0 unit 0 family iso address 10.1001.0010.0100.00”

For IS-IS to work properly it requires 3 additional parameters in its basic configuration. First, there needs to be a loopback interface in each router, which it's ensures that the device will be reachable, if an interface has lost connection. Second when an interface is stated it must be configured as a "family iso" and not as a "family inet". The reason is mainly because the IS-IS in a layer 2 routing protocol and it can't be supported by the "family inet" statement which supports IP protocol traffic (OSPF, BGP).

IS-IS uses ISO network addresses, where it is connected with one of the interfaces of the router. In addition, the router will require a special network address called a *Network Entity Title (NET)* on one of the devices interfaces preferably, the lo0 interface by setting the "family iso" command. To create the NET, address the loopback address is chosen, removed all the dots(.) and were insert leading zeroes where necessary so that the string is 12 characters long (100100100100). Then a dot is added every 4th character (1001.0010.0100), and it is inserted the area number at the beginning of the string (10.1001.0010.0100) and last the selector in the end of the string (10.1001.0010.0100.00).

The basic configuration for *Olive2* is:

Olive2: "set interface em0 unit 0 family iso"

Olive2: "set interface em1 unit 0 family inet address 192.168.30.1/30"

Olive2: "set interface em1 unit 0 family iso"

Olive2: "set interface lo0 unit 0 family inet address 2.2.2.2/32"

Olive2: "set interface em0 unit 0 family iso address 10.2002.0020.0200.00"

The same process is repeated for every other router in the topology. Next step is to activate IS-IS and configure each router the hierarchy Level they will be. *Olive1*, *Olive3*, *Olive4* and their adjacent interface are Level 1 *Olive6*, *Olive7* and their adjacent interface are Level 1. *Olive2* and *Olive5* are Level 1/Level 2. To enable IS-IS, the configuration is for *Olive1*, *Olive3* and *Olive4* is:

Olive1,3,4: "set protocols isis interface em0"

Olive1,3,4: "set protocols isis interface em1"

Olive1,3,4: "set protocols isis interface lo0"

Olive1,3,4: "set protocols isis level 2 disable"

Respectively for *Olive6* and *Olive7*:

Olive6,7: "set protocols isis interface em0"

Olive6,7: "set protocols isis interface em1"

Olive6,7: "set protocols isis interface lo0"

Olive6,7: “set protocols isis level 1 disable”

The first three commands define the protocol (*protocols isis*) the router is using to direct the traffic and in which interfaces (*em0, em1*). The fourth command defines the hierarchy level in which the routers are operate. The default policy states that all interfaces in the *IS-IS* protocol is a *Level 1* and *Level 2* until one is disable.

Lastly for *Olive2*:

Olive2: “set protocols isis interface em0”

Olive2: “set protocols isis interface em0 level 2 disable”

Olive2: “set protocols isis interface em1”

Olive2: “set protocols isis interface em1 level 2 disable”

Olive2: “set protocols isis interface em2”

Olive2: “set protocols isis interface em2 level 2 disable”

Olive2: “set protocols isis interface em3”

Olive2: “set protocols isis interface lo0”

And for *Olive5*:

Olive5: “set protocols isis interface em0”

Olive5: “set protocols isis interface em0 level 2 disable”

Olive5: “set protocols isis interface em1”

Olive5: “set protocols isis interface em2”

Olive5: “set protocols isis interface em2 level 1 disable”

Olive5: “set protocols isis interface em3”

Olive5: “set protocols isis interface em3 level 1 disable”

Olive5: “set protocols isis interface lo0”

Having completed all the configuration properly, the routers will be able to ping and direct traffic to

```
root@Olive7# run ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1): 56 data bytes
64 bytes from 192.168.10.1: icmp_seq=0 ttl=62 time=16.667 ms
64 bytes from 192.168.10.1: icmp_seq=1 ttl=62 time=12.675 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=62 time=14.706 ms
^C
--- 192.168.10.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 12.675/14.683/16.667/1.630 ms
```

Figure 29: Ping Olive1 from Olive7

any other router (Figure 29).

4.5 Border Gateway Protocol (BGP) Topology

The most developed and the most difficult to understand routing algorithm. Having the responsibility to direct the packages to the whole world and to be able to do it with great success, little delay and as little overhead as possible. this robs a complex mechanism of selecting and distributing the load among countless routers. The ability to work with all other algorithms without any problems. The ability to be able to select routes based on specific rules so that it can dynamically control the redistribution of cargo makes this protocol the most complex designed and dynamically independent that exists. This protocol is divided into two subcategories. The iBGP and the eBGP. Between them the 2 protocols are identical and they are made up of almost all mechanisms with a very important difference, however. the iBGP is intended only for internal areas like the other protocols mentioned above. The eBGP purpose is to interconnect the internal AS with each one also operates by a dynamic protocol. Essentially the second subcategory is the one that makes the internet work, it is the one that connects different terminals from different areas. In the BGP network topology (Figure 30) there are 4 different areas where in each one a different dynamic protocol will be operating.

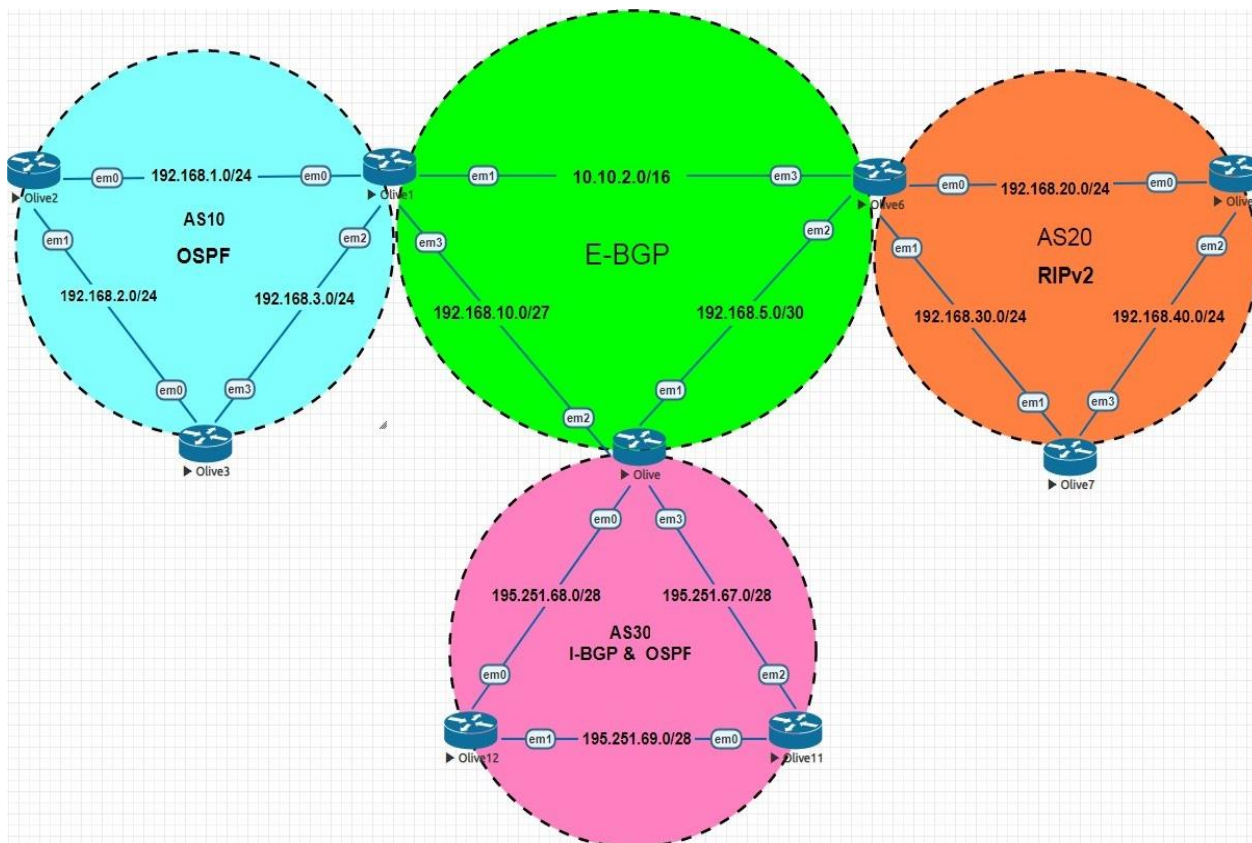


Figure 30: BGP Network Topology

Beginning with the configuration of the *OSPF* area the routers *Olive1,2* and *3* are reside in the *OSPF* area. The first step in the *OSPF* network topology is the basic configuration of the interfaces

in their appropriate network as it is shown in Figure 30. Next step is to activate *OSPF* on the networks it will be operating. It is obligated to allow *OSPF* in the interfaces where the traffic is going to be. To enable *OSPF* is our routers, the configuration for *Olive2* is:

Olive2: “*edit protocols ospf area 0*”

Olive2: “*set interface em0*”

Olive2: “*set interface em1*”

Olive2: “*top*”

Olive2: “*set routing-options autonomous system 10*”

Similar for *Olive3*:

Olive3: “*edit protocols ospf area 0*”

Olive3: “*set interface em0*”

Olive3: “*set interface em3*”

Olive3: “*top*”

Olive3: “*set routing-options autonomous system 10*”

Similar for *Olive1*:

Olive1: “*edit protocols ospf area 0*”

Olive1: “*set interface em0*”

Olive1: “*set interface em3*”

Olive1: “*top*”

Olive1: “*set routing-options autonomous system 10*”

As it was analyzed in the *OSPF* Topology, the first command selects the protocol that is going to be used for the advertisement and the area where the routers interface will belong. The next two commands specify the interfaces that will be advertised by the *OSPF* protocol. The “*top*” command is used to go to the starting level of the command mode and the last command is used to set the number of the *as* the router will belong to.

Second step is the configuration of the router’s *Olive5,6,7* which will run *RIPv2* protocol. Likewise, in the *RIPv2* Topology it’s necessary to set a policy-statement to the router be able to advertise the routes learned by the *RIPv2* protocol. Because by default the *RIPv2* don’t advertise any adjacent route from the device’s interface. The policy-statement is called *export-rip* and it’s similar for all the router’s, the configuration is:

Olive5, 6, 7: “*edit policy-options policy-statements export-rip term 1*”

Olive5, 6, 7: “*set from protocol direct*”

Olive5, 6, 7: “*set from protocol rip*”

Olive5, 6, 7: “*set then accept*”

These commands define the routing policy statement (*routing-options policy-statements*) with the name of our choosing (*export-rip*) and the configuring term (*term 1*). The next commands are the conditions a route ought to fulfill so that it can be advertised. The above statements are configured the same on all the other routers with no exception.

Having set the routing policy, the next step is to configure the routes to communicate now only with their neighbors but with the other routers as well using the dynamic protocol RIPv2. The configuration for *Olive5* is:

Olive5: “*set routing-options autonomous-system 20*”

Olive5: “*edit protocols rip group rip-group*”

Olive5: “*set export export-rip*”

Olive5: “*set neighbor em0 receive version-2*”

Olive5: “*set neighbor em2 receive version-2*”

Similar for *Olive7*:

Olive7: “*set routing-options autonomous-system 20*”

Olive7: “*edit protocols rip group rip-group*”

Olive7: “*set export export-rip*”

Olive7: “*set neighbor em1 receive version-2*”

Olive7: “*set neighbor em3 receive version-2*”

Respectively for *Olive5*:

Olive5: “*set routing-options autonomous-system 20*”

Olive5: “*edit protocols rip group rip-group*”

Olive5: “*set export export-rip*”

Olive5: “*set neighbor em0 receive version-2*”

Olive5: “*set neighbor em1 receive version-2*”

The following step is the configuration of the *iBGP* area. The routers *Olive11*, *Olive12* and *Olive* will be using *iBGP* as their *IGP*. For *iBGP* to work properly it requires some additional parameters in its configuration. Generally, in *iBGP* the loopback interface (*lo0*) is used to establish connections between *iBGP* peers. The loopback interface is always up as long as the device is operating. If there is a route to the loopback address, the *iBGP* peering session stays up. If a physical interface address is used instead and that interface goes up and down, the *iBGP* peering session also goes up and down. Thus, the loopback interface is preferable because it provides fault tolerance in case the physical interface or the link goes down, if the device has link redundancy.

While *iBGP* neighbors do not need to be directly connected, they do need to be fully meshed. In this case, fully meshed means that each device is logically connected to every other device through neighbor peer relationships. The neighbor statement creates the mesh. But because of the full mesh requirement of *iBGP*, the individual peering sessions between all *iBGP* devices in the AS must take place. The full mesh need not to be physical links.

The full-mesh requirement it exist only *iBGP* because, an *iBGP*-learned route cannot be re-advertised to another *iBGP* peer. The reason for preventing the re-advertisement of *iBGP* routes and requiring the full mesh is to avoid routing loops within an AS. Thus, to achieve the fully-mesh situation in an AS, there will need to be use another *IGP* to advertise the available networks that exist inside the AS. Consequently, the *OSPF* protocol will be used to create a fully-meshed area. Having completed the basic configuration for all three routers', the configuration for *Olive11* is:

Olive11: *“set interface lo0 unit 0 family inet address 3.3.3.3”*

Olive11: *“edit protocols ospf area 1”*

Olive11: *“set interface em0”*

Olive11: *“set interface em2”*

Olive11: *“set interface lo0 passive”*

The last command (*set interface lo0 passive*) is to set the *lo0* interface into passive mode because we don't want to receive any packet from the rest topology, only to advertise itself. Next step, a policy-option must be set for the *iBGP* to advertise its local route:

iBGP Routers Group: *“edit routing-options policy-statements send-direct term 1”*

iBGP Routers Group *“set from protocol direct”*

iBGP Routers Group *“set then accept”*

Before the *BGP* peers are established, because local routes are not automatically advertised by the *BGP* peers. At each *BGP*-enabled device, a policy configuration is required in order to export the local, static, or *IGP*-learned routes and then advertise them as *BGP* routes to the other peers. *BGP*'s advertisement policy, by default, does not advertise any non-*BGP* routes (such as local routes) to peers. Next is the *iBGP* configuration of *Olive11*:

Olive11: “*set routing-options router-id 3.3.3.3*”

Olive11: “*set routing-options autonomous-system 30*”

Olive11: “*edit protocols bgp group group1*”

Olive11: “*set type internal*”

Olive11: “*set export send-direct*”

Olive11: “*set local-address 3.3.3.3*”

Olive11: “*set neighbor 192.251.67.1*”

Olive11: “*set neighbor 192.251.69.2*”

The first two commands set the router-id (*router-id 3.3.3.3*) and the AS (*autonomous-system 30*) which the router will belong to. The third command (*protocols bgp group group1*) specifies the protocol and the group’s name for the router. The fourth command defines the *BGP* type (*internal=iBGP, external=eBGP*), next is the routing policy (*export send-direct*) for the *iBGP* peers, following the local-address statement which enables the router to specify the source information in *BGP* update messages. The last two commands set the neighbor’s IP in each interface network respectively.

The configuration for *Olive12*:

Olive12: “*set interface lo0 unit 0 family inet address 2.2.2.2*”

Olive12: “*edit protocols ospf area 1*”

Olive12: “*set interface em0*”

Olive12: “*set interface em1*”

Olive12: “*set interface lo0 passive*”

Olive12: “*edit policy-options policy-statements send-direct term 1*”

Olive12: “*set from protocol direct*”

Olive12: “*set then accept*”

Olive12: “*set routing-options router-id 2.2.2.2*”

Olive12: “*set routing-options autonomous-system 30*”

Olive12: “*edit protocols bgp group group1*”

Olive12: “*set type internal*”

Olive12: “*set export send-direct*”

Olive12: “set local-address 2.2.2.2”

Olive12: “set neighbor 192.251.67.1”

Olive12: “set neighbor 192.251.69.2”

Respectively for *Olive*:

Olive: “set interface lo0 unit 0 family inet address 1.1.1.1”

Olive: “edit protocols ospf area 1”

Olive: “set interface em0”

Olive: “set interface em3”

Olive: “set interface lo0 passive”

Olive: “edit routing-options policy-statements send-direct term 1”

Olive: “set from protocol direct”

Olive: “set then accept”

Olive: “set routing-options router-id 1.1.1.1”

Olive: “set routing-options autonomous-system 30”

Olive: “edit protocols bgp group group1”

Olive: “set type internal”

Olive: “set export send-direct”

Olive: “set local-address 1.1.1.1”

Olive: “set neighbor 192.251.68.1”

Olive: “set neighbor 192.251.67.2”

BGP by default is exchanging information with others routers from adjacent AS's. The reason for this data exchange is very important. Based on the information he receives from the other routers regarding their routing table from each router, he makes a comparison between his own routes and those he has received from the others. As a result, he can make a complete map with all the routers that exist in the network and at the same time removes or change as many paths as have the ability to create loops in the network. It achieves this by applying rules at the AS's level.

The configuration of the *eBGP* session between two routers is different for each router, because they belong to different *IGP* with different characteristics and parameters. Starting with the configuration of *Olive* after the *BGP* peers are established, those interfaces that run another dynamic protocol will not be able to advertise through the BGP. To achieve this, each router will

need to create a set of rules that will indicate that as many routes as the router knows through local or static or other dynamic protocols can be advertised through the BGP so that others can learn. the protocol.

Because in Olive two *IGP* are operating simultaneously (*iBGP* and *OSPF*) two different policies ought to be set one for each *IGP*:

Olive: *“edit routing-options policy-statements ospf-to-bgp term 1”*

Olive: *“set from protocol [bgp direct ospf]”*

Olive: *“set then accept”*

With this policy the routers that are running *OSPF* now will export the routes learned through *OSPF* to other routers. Similar for *iBGP*:

Olive: *“edit routing-options policy-statements export-all term 1”*

Olive: *“set then accept”*

With this policy the routers will be able to export the routes they know to any other router. This policy can be applied to *iBGP* and *eBGP* interfaces also, thus there is no need to set two different policies one for each protocol. Continuing with the configuration:

Olive: *“set protocols ospf export ospf-to-bgp”*

Olive: *“set protocols bgp export export-all”*

Olive: *“edit protocols bgp external-peers 0-1”*

Olive: *“set type external”*

Olive: *“set local-address 192.168.10.1”*

Olive: *“set peer-as 10”*

Olive: *“set neighbor 192.168.10.2”*

Olive: *“top”*

Olive: *“edit protocols bgp external-peers 0-6”*

Olive: *“set type external”*

Olive: *“set local-address 192.168.5.2”*

Olive: *“set peers as 20”*

Olive: *“set neighbor 192.168.5.1”*

The first command (*export ospf-to-bgp*) sets the policy to export any routes learned through *OSPF* in any *BGP* enabled device, similar for the second command (*export export-all*) where the policy enables the router to advertise any routes learned through *iBGP* and *eBGP*. The third command (*bgp external-peers 0-1*) creates a group where the variables for the connection between routers *Olive* and *Olive1* are located. Thus, the type (*type external*) of the connection must be set, alongside with the local address (*local-address 192.168.10.1*) of the routers interface, the number of the *AS* (*peers as 10*) where the neighbor belongs to and the neighbors interface IP (*neighbor 192.168.10.2*). Respectively the same variables have to be set for the connection between *Olive* and *Olive6*, the name of the group (*bgp external-peers 0-6*), the type of the connection (*type external*), the address of the router interface (*local-address 192.168.5.2*), the *AS* (*peers as 20*) of the neighbor's interface, and its IP (*neighbor 192.168.5.1*).

Mutually the configuration for *Olive1* is:

Olive1: “*edit policy-options policy-statements bgp-to-ospf term 1*”

Olive1: “*set from protocol [ospf direct bgp]*”

Olive1: “*set then accept*”

This policy specifies the routes that are learned through *eBGP* should be advertised in the *OSPF* area. Continuing:

Olive1: “*edit policy-options policy-statements export-all term 1*”

Olive1: “*set then accept*”

Olive1: “*set protocols ospf export bgp-to-ospf*”

Olive1: “*set protocols bgp export export-all*”

Olive1: “*edit protocols bgp external-peers 0-1*”

Olive1: “*set type external*”

Olive1: “*set local-address 192.168.10.2*”

Olive1: “*set peer-as 30*”

Olive1: “*set neighbor 192.168.10.1*”

Olive1: “*top*”

Olive1: “*edit protocols bgp external-peers 1-6*”

Olive1: “*set type external*”

Olive1: “*set local-address 10.10.2.1*”

Olive1: “*set peers as 20*”

Olive1: “set neighbor 10.10.2.2”

Respectively for Olive6:

Olive6: “edit policy-options policy-statements bgp-to-rip term 1”

Olive6: “set from protocol [rip direct bgp]”

Olive6: “set then accept”

Alike to Olive1 policy it allows the advertisement routes that are learned through eBGP should be advertised in the RIPv2 area. Continuing:

Olive6: “edit policy-options policy-statements export-all term 1”

Olive6: “set then accept”

Olive6: “top”

Olive6: “set protocols rip group rip-group export rip-to-ospf”

Olive6: “set protocols bgp export export-all”

Olive6: “edit protocols bgp external-peers 0-6”

Olive6: “set type external”

Olive6: “set local-address 192.168.5.1”

Olive6: “set peer-as 30”

Olive6: “set neighbor 192.168.5.2”

Olive6: “top”

Olive6: “edit protocols bgp external-peers 1-6”

Olive6: “set type external”

Olive6: “set local-address 10.10.2.2”

Olive6: “set peers as 10”

Olive6: “set neighbor 10.10.2.1”

Having completed the configuration of all areas, routers are able to communicate and direct traffic to any other node in the topology (Figure 31).

```
root@Olive5# run ping 195.251.69.1
PING 195.251.69.1 (195.251.69.1): 56 data bytes
64 bytes from 195.251.69.1: icmp_seq=0 ttl=62 time=11.359 ms
64 bytes from 195.251.69.1: icmp_seq=1 ttl=62 time=10.833 ms
64 bytes from 195.251.69.1: icmp_seq=2 ttl=62 time=10.719 ms
```

Figure 31: Ping from Olive5 to Olive12

5 Chapter 4 : Conclusion and Future work

In this thesis we presented the creation of a series of networking exercises with the use of the EVE-NG emulation platform both in practice and in theory. The purpose was firstly to provide universities and networking labs an easy to use platform from where they could create, edit and manage multiple networking topologies. Secondly the expanded learning experience for university students where they can have a better and more detailed understanding of the courses and more in-depth analysis of the services and protocols that are used for the communication on the Internet.

All this was accomplished by using the pre-existing platform of EVE-NG and the creation of specific courses based on the tools offered. Of course, the system created is very simple and with relatively small requirements in terms of cost and resources. But it also gives us extremely lucrative prospects for the platform's scalability so that it can support a larger scale environment.

With the increasing use of microservices for better scalability, more efficient resource management and overall improved application life-cycle. The next step in the thesis is the improvement of the EVE-NG platform to a Tokenized based application. Were the supervision in the EVE-NG will be easier and the maintenance of such an application will be more effortless.

6 Chapter 5: References

1. Sven Reimann, Sebastian Rieger, Christian Pape, "Using Cisco VIRL and GNS3 to Improve the Scale-out of Large Networks Testbeds in Higher Education," International Journal on Advances in Telecommunications, 2017. [Accessed 23 January 2019]
2. D. K. G. Margaret Rouse, "Search tworking," [Online]. Available: <https://searchnetworking.techtarget.com/definition/OSI>. [Accessed 19 January 2019].
3. M. Bahl, "Medium,". Available: <https://medium.com/@madhavbahl10/osi-model-layers-explained-ee1d43058c1f>. [Accessed 19 January 2019].
4. "Lifewire," [Online]. Available: <https://www.lifewire.com/layers-of-the-osi-model-illustrated-818017>. [Accessed 19 January 2019].
5. P. Simoneau, "The OSI Model: Understanding the Seven Layers of Computers Networks," p. 11[5], 2006.
6. M. Fuszner, "GNS3 Graphical Network Simulator," [Online]. Available: <https://www.csd.uoc.gr/~hy435/material/GNS3-0.5-tutorial.pdf>. [Accessed 19 January 2019].
7. Atlantic Venture Forum Image , Available : <https://atlanticventureforum.ca/program/look-whos-joining-us/gns3/> Image [Accessed 23 April 2020].
8. Available : <https://sg.carousell.com/p/cisco-packet-tracer-243629548/> [Accessed 24 April 2020].
9. "Cisco," [Online]. Available: https://www.cisco.com/c/dam/en_us/training-events/netacad/course_catalog/docs/Cisco_PacketTracer_DS.pdf. [Accessed 2019 January 2019].
10. Totally unreachable, "Eve-NG Network Emulator First Looks", [Online]. Available: <https://www.pingunreachable.com/eve-ng-network-emulator-first-looks/>. [Accessed 12 May 2020].
11. U. Dzerkals, "EVE-NG," [Online]. Available: <http://www.eve-ng.net/images/EVE-COOK-BOOK-1.7.pdf>. [Accessed 19 January 2019].
12. "GRNET," [Online]. Available: <https://oceanos.grnet.gr/home/>. [Accessed 19 January 2019].
13. StudyCCNA, "What is IP routing," [Online]. Available: <https://study-ccna.com/what-is-IP-routing/>. [Accessed 11 December 2018].
14. Network Computing, "Cisco Networking Basics: IP Addressing," 3 April 2018. [Online]. Available: <https://www.networkcomputing.com/networking/cisco-networking-basics-IP-addressing/636970615>. [Accessed 11 December 2018].
15. GRANDMETRIC, "Where to use static and where to use dynamic routing" Blog, 25 August 2017. [Online]. Available: <https://www.grandmetric.com/2017/08/25/where-to-use-static-and-where-to-use-dynamic-routing/>. [Accessed 12 December 2018].
16. CCNA Blog, "Dynamic routing protocols," [Online]. Available: <http://www.ccnablog.com/dynamic-routing-protocols/>. [Accessed 12 December 2018].

17. Traceroute, " Routing Protocols and Concepts, CCNA Exploration Companion Guide," [Online]. Available: http://ptgmedia.pearsoncmg.com/images/9781587132063/samplechapter/1587132060_03.pdf. [Accessed 12 December 2018].
18. D. M. S. S. Russ White, "Introduction to the Border Gateway Patrol," 27 August 2004. [Online]. Available: <http://www.informit.com/articles/article.aspx?p=331613&seqNum=3>. [Accessed 12 December 2018].
19. SunilKhanna, Cisco, 10 June 2017. [Online]. Available: <https://community.cisco.com/t5/networking-documents/rIPv2-routing-information-protocol/ta-p/3117425>. [Accessed 12 December 2018].
20. Network Chef BD, "Introduction to RIPv2 with configuration example," 27 September 2017. [Online]. Available: <https://networkchefbd.com/rIPv2-fundamental/>. [Accessed 12 December 2018].
21. Cisco, "OSPF Design Guide," Cisco, 10 August 2005. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/IP/open-shortest-path-first-ospf/7039-1.html>. [Accessed 12 December 2018].
22. H. FOUAD, "HOW OSPF PROTOCOL WORKS & BASIC CONCEPTS: OSPF NEIGHBOR, TOPOLOGY & ROUTING TABLE, OSPF AREAS & ROUTER ROLES, THEORY & OVERVIEW," [Online]. Available: <http://www.firewall.cx/networking-topics/routing/ospf-routing-protocol/1110-ospf-operation-basic-advanced-concepts-ospf-areas-roles-theory-overview.html>. [Accessed 12 December 2018].
23. Metaswitch, "What is Intermediate System - Intermediate System (IS-IS)?" Metaswitch, [Online]. Available: <https://www.metaswitch.com/knowledge-center/reference/what-is-intermediate-system-to-intermediate-system-isis>. [Accessed 12 December 2018].
24. IPCISCO, "IS-IS," [Online]. Available: <https://IPcisco.com/lesson/is-is/>. [Accessed 12 December 2018].
25. I. Pepelnjak, "BGP tutorial: The routing protocol that makes the Internet work," [Online]. Available: <https://searchtelecom.techtarget.com/feature/BGP-essentials-The-protocol-that-makes-the-Internet-work>. [Accessed 12 December 2018].
26. Cisco Corporation, "IP Addressing and Subnetting for New Users," Cisco, 10 August 2016. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/IP/routing-information-protocol-rIPv2/13788-3.html>. [Accessed 11 December 2018].
27. K. W. R. JAMES F. KUROSE, "The Internet Protocol (IP): IPv4, Addressing, IPv6, and More," in Computer Networking A Top-Down Approach, Pearson, 2017, pp. 357-362.
28. Chris Partsenidis, "An introduction to IP addressing and subnet masks," SearchITChannel, January 2007. [Online]. Available: <https://searchitchannel.techtarget.com/tIP/An-introduction-to-IP-addressing-and-subnet-masks>. [Accessed 11 December 2018].
29. Microsoft, "Understanding TCP/IP addressing and subnetting basics," Microsoft, 17 April 2018. [Online]. Available: <https://support.microsoft.com/en-us/help/164015/understanding-tcp-ip-addressing-and-subnetting-basics>. [Accessed 11 December 2018].
30. Metaswitch Networks, "What is IP routing?" Metaswitch, [Online]. Available: <https://www.metaswitch.com/knowledge-center/reference/what-is-ip-routing>. [Accessed 12 December 2018].

31. OmniSecu, "Types of Routes Static Routes and Dynamic Routes, Difference between static route and dynamic route," OmniSecu, [Online]. Available: <http://www.omnisecu.com/cisco-certified-network-associate-ccna/types-of-routes-static-routes-and-dynamic-routes.php>. [Accessed 12 December 2018].
32. Difference Between, "Difference Between Static and Dynamic Routing," 14 February 2015. [Online]. Available: <https://www.differencebetween.com/difference-between-static-and-vs-dynamic-routing/>. [Accessed 12 December 2018].
33. CISCO, "Cisco Networking Academy's Introduction to Static Routing," Cisco Networking Academy, 27 March 2014. [Online]. Available: <http://www.ciscopress.com/articles/article.asp?p=2180209&seqNum=4>. [Accessed 12 December 2018].
34. S. KUMAR, "Dynamic Routing," [Online]. Available: <https://www.techtutsonline.com/dynamic-routing/>. [Accessed 12 December 2018].
35. Cisco Networking Academy, "Cisco Networking Academy's Introduction to Routing Dynamically," 24 March 2014. [Online]. Available: <http://www.ciscopress.com/articles/article.asp?p=2180210&seqNum=8>. [Accessed 12 December 2018].
36. J. Doyle, "Dynamic Routing Protocols," Cisco, 16 November 2001. [Online]. Available: <http://www.ciscopress.com/articles/article.asp?p=24090&seqNum=4>. [Accessed 12 December 2018].
37. TechDifferences, "Difference Between Static and Dynamic Routing," 8 February 2018. [Online]. Available: <https://techdifferences.com/difference-between-static-and-dynamic-routing.html>. [Accessed 12 December 2018].
38. JUNIPER, "Junos OS Overview," JunIPer, 13 June 2018. [Online]. Available: https://www.junIPer.net/documentation/en_US/junos/topics/concept/junos-software-introduction.html. [Accessed 12 December 2018].
39. J. Duffy, "Cisco vs JunIPer," NetworkWorld, 7 June 2010. [Online]. Available: <https://www.networkworld.com/article/2210939/lan-wan/cisco-vs-junIPer.html>. [Accessed 12 December 2018].