



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΙΩΑΝΝΙΝΩΝ

**ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΠΜΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΔΙΚΤΥΩΝ**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΜΕΛΕΤΗ ΔΙΚΤΥΩΝ ΚΑΘΟΡΙΖΟΜΕΝΑ ΑΠΟ ΛΟΓΙΣΜΙΚΟ**  
**(ΑΠΟΔΟΣΗ ΑΝΑ ΤΟΠΟΛΟΓΙΑ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ**  
**ΠΡΩΤΟΚΟΛΛΩΝ ΑΝΑΚΑΛΥΨΗΣ ΤΟΠΟΛΟΓΙΑΣ)**

**Νικόλαος Οικονόμου**

**Επιβλέπων: Ελευθέριος Στεργίου,**  
**ΔΕΠ, Αναπληρωτής Καθηγητής**

**Άρτα, Ιούνιος 2023**



**UNIVERSITY  
OF IOANNINA**

**SCHOOL OF INFORMATICS AND TELECOMMUNICATIONS**

**DEPARTMENT OF INFORMATICS AND  
TELECOMMUNICATIONS**

**MSc INFORMATICS AND NETWORKS**

**MASTER THESIS**

**Nikolaos Oikonomou**

**STUDY OF SOFTWARE-DEFINED NETWORKS  
(PERFORMANCE BY TOPOLOGY AND EVALUATION OF  
TOPOLOGY DISCOVERY PROTOCOLS)**

**Supervisor: Eleftherios Stergiou,  
FM, Assistant Professor**

**Arta, June 2023**



## **Εγκρίθηκε από τριμελή εξεταστική επιτροπή**

Άρτα, Ημερομηνία

### **ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ**

1. Επιβλέπων καθηγητής  
Ελευθέριος Στεργίου,  
ΔΕΠ, Αναπληρωτής Καθηγητής
2. Μέλος επιτροπής  
Ευριπίδης Γλαβάς,  
ΔΕΠ, Καθηγητής Α
3. Μέλος επιτροπής  
Κωνσταντίνος Αγγέλης,  
ΔΕΠ, Καθηγητής Α

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

## **Δήλωση μη λογοκλοπής**

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εκ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Οικονόμου, Νικόλαος

Υπογραφή

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Για την εκπόνηση της παρούσας μεταπτυχιακής εργασίας θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Ελευθέριο Στεργίου για την πολύτιμη βοήθεια και καθοδήγηση που μου έδωσε για αυτή την εργασία αλλά και σε όλη την διάρκεια των σπουδών μου από όταν ξεκίνησα ως προπτυχιακός φοιτητής στο τμήμα μέχρι και σήμερα με την ολοκλήρωση των μεταπτυχιακών σπουδών μου σε αυτό καθώς και τον καθηγητή μου κ. Κωνσταντίνο Αγγέλη για τις σωστές συμβουλές και την καθοδήγηση που μου έδωσε κατά τη διάρκεια των μεταπτυχιακών μου σπουδών. Επιπλέον θα ήθελα να ευχαριστήσω και τον καθηγητή μου κ. Νικόλαο Αντωνιάδη ο οποίος με βοήθησε στο να αναπτύξω τις τεχνικές που απαιτούνται για την σωστή συγγραφή επιστημονικών εργασιών και ερευνών. Τέλος ευχαριστώ την οικογένειά μου και τους φίλους μου που ήτανε δίπλα μου και με στηρίζανε σε κάθε απόφαση που αφορούσε τις σπουδές μου.

## ΠΕΡΙΛΗΨΗ

Το Software Defined Network (SDN) είναι ένα νέο πολλά υποσχόμενο δικτυακό περιβάλλον όσον αφορά την αποτελεσματικότητα, την πολυπλοκότητα και τη διαχείριση των δικτύων. Σε αυτή την εργασία πρόκειται να μελετηθεί η απόδοση SDN δικτύων. Θα πραγματοποιηθούν δοκιμές δικτύωσης που καθορίζονται από λογισμικό (SDN) και συγκεκριμένα θα χρησιμοποιηθούν τυπικές τοπολογίες δικτύου και προσομοίωση της λειτουργίας τους. Η όλη έρευνα πρόκειται να διεξαχθεί σε υπολογιστή με λειτουργικό σύστημα Windows και θα γίνει χρήση εικονικής μηχανής για λειτουργία του προσομοιωτή Mininet.

Στόχος της έρευνας είναι να εξεταστούν τα ακόλουθα:

- i. Η απόδοση του συστήματος σε σχέση με τις διάφορες υπάρχουσες τοπολογίες
- ii. Η επίδραση του συνολικού αριθμού των switch τα οποία εμπλέκονται στην αρχιτεκτονική SDN και ο ρόλος τους στην απόδοση του δικτύου
- iii. Η απασχόληση της CPU.
- iv. Η απασχόληση της μνήμης RAM
- v. Η καθυστέρηση της μεταφοράς πακέτων μεταξύ δικτυακών κόμβων.

Επιπλέον, σε αυτή την εργασία θα αναλυθούν και θα αξιολογηθούν θέματα σχετιζόμενα με την απόδοση και την επεκτασιμότητα του πρωτοκόλλου OFDP. Η ανάλυση και η αξιολόγησή του OFDP θα υποστηριχθεί από λεπτομερή πειράματα σε μια σειρά τοπολογιών στον εξομοιωτή δικτύου Mininet το οποίο θα χρησιμοποιεί τον ελεγκτή POX.

**Λέξεις-κλειδιά:** Ελεγκτής, Εξομοιωτής, Προσομοίωση, Πρωτόκολλο, Τοπολογία.



## **ABSTRACT**

Software-defined networking (SDN) is a promising new networking environment in terms of efficiency, complexity, and management of networks. In this research the performance of SDN networks is to be studied. Software-defined networking (SDN) tests will be conducted, typical network topologies will be used to simulate their operation. The entire research is going to be conducted with a Windows computer using virtual machines operating the Mininet Simulator.

We are interested in considering the following:

The aim of the research is to examine the following:

- i. The performance of the system in relation to the various existing topologies
- ii. The effect of the total number of switches involved in the SDN architecture and their role in network performance.
- iii. CPU usage
- iv. RAM usage
- v. Transfer delay of packets between network nodes.

Additionally, performance and scalability issues of the OFDP protocol will be analyzed and evaluated in this work. The analysis and evaluation of the OFDP will be supported by detailed experiments on a series of topologies in the Mininet network simulator using the POX controller.

**Keywords:** Controller, Emulator, Simulation, Protocol, Topology

# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

## Περιεχόμενα

ΕΥΧΑΡΙΣΤΙΕΣ .....	1
ΠΕΡΙΛΗΨΗ .....	2
ABSTRACT.....	3
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	4
ΠΙΝΑΚΑΣ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ .....	9
ΑΠΟΔΟΣΗ ΟΡΩΝ / ΓΛΩΣΣΑΡΙΟ .....	11
1. ΕΙΣΑΓΩΓΗ .....	12
1.1 Αντικείμενο και στόχος της μεταπτυχιακής εργασίας .....	13
1.2 Δομή μεταπτυχιακής εργασίας.....	14
2. ΣΧΕΤΙΚΕΣ ΕΡΕΥΝΕΣ .....	15
3. ΠΕΡΙΓΡΑΦΗ ΤΟΥ SDN.....	17
3.1 Η πορεία του SDN .....	17
3.2 Ο ορισμός SDN.....	18
3.3 Απεικόνιση του SDN .....	18
3.4 Τι προσφέρει το SDN.....	20
3.5. Επιχειρήσεις που χρησιμοποιούν τεχνολογία SDN .....	22
3.5.1 Πάροχοι SDN.....	22
3.5.2 Μερίδιο αγοράς SDN.....	24
3.5.3 Παραδείγματα βιομηχανικής χρήσης SDN .....	25
3.6 Πλεονεκτήματα - Μειονεκτήματα SDN.....	27
3.6.1 Πλεονεκτήματα SDN .....	27
3.6.2 Μειονεκτήματα SDN .....	28
4. ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ SDN .....	30
4.1 Ελεγκτής (Controller).....	30
4.2 Λειτουργίες Ελεγκτών.....	31

4.2.1 Επιλογή και τοποθέτηση ελεγκτών .....	35
4.2.2 Αρχιτεκτονικές τοποθέτησης ελεγκτών .....	37
4.3 Είδη ελεγκτών .....	38
4.3.1 Ελεγκτές Ελεύθερης πρόσβασης.....	38
4.3.1.1 Ενεργοί ελεγκτές .....	39
4.3.1.1.1 Ελεγκτής OpenDaylight .....	40
4.3.1.1.2 Ελεγκτής ONOS.....	42
4.3.1.1.3 Ελεγκτής Ryu .....	43
4.3.1.1.4 Ελεγκτής Faucet .....	44
4.3.1.1.5 Ελεγκτής FRR .....	44
4.3.1.1.6 Ελεγκτής Trema .....	45
4.3.1.1.7 Ελεγκτής Open Contrail .....	45
4.3.1.1.8 Project Calico .....	46
4.3.1.2 Ανενεργοί ελεγκτές .....	46
4.3.1.2.1 Ελεγκτής Floodlight .....	47
4.3.1.2.2 Ελεγκτής LOOM.....	47
4.3.1.2.3 Ελεγκτής Open MUL .....	48
4.3.1.2.4 Ελεγκτής Maestro.....	48
4.3.1.2.5 Ελεγκτής Beacon.....	49
4.3.1.2.6 Ελεγκτής Flower .....	49
4.3.2 Ελεγκτής NOX/POX.....	50
4.3.3 Εμπορικοί Ελεγκτές .....	52
4.3.3.1 Cisco Application Centric Infrastructure (ACI).....	52
4.3.3.1.1 Αρχιτεκτονική Spine-Leaf.....	57
4.3.3.2 Juniper Networks Contrail .....	58
4.3.3.3 VMware NSX.....	59
4.3.3.4 HP-IMC.....	60
4.3.3.5 NEC.....	60

4.4 Switch.....	62
4.4.1 Ο ρόλος του Switch στο SDN.....	63
4.4.2 OpenFlow Switch.....	64
4.4.3 Εικονικό Switch .....	64
4.4.4 Φυσικό Switch.....	65
4.4.5 Edge Switch .....	66
5. ΠΡΩΤΟΚΟΛΛΑ ΛΕΙΤΟΥΡΓΙΑΣ SDN .....	67
5.1 Πρωτόκολλο OpenFlow .....	67
5.1.1 Η λειτουργία του OpenFlow .....	68
5.1.2 Πίνακες OpenFlow και καταχωρήσεις ροής .....	69
5.2 Πρωτόκολλο NetConf .....	70
5.2.1 Γλώσσα Yang.....	72
5.3 Πρωτόκολλο Open vSwitch Database Management.....	73
5.4 Πρωτόκολλο Border Gateway.....	74
5.5 Πρωτόκολλο Locator/Identifier Separation .....	75
5.6 Πρωτόκολλο Simple Network Management.....	76
5.7 Πρωτόκολλο Link Layer Discovery Protocol .....	76
6. ΠΡΟΔΙΑΓΡΑΦΕΣ ΥΛΙΚΟΥ ΚΑΙ ΛΟΓΙΣΜΙΚΟΥ .....	79
6.1 Προδιαγραφές υλικού .....	79
6.2 Προδιαγραφές λογισμικού .....	80
6.2.1 Windows 11 Pro for Workstations OS.....	80
6.2.2 VirtualBox.....	81
6.2.3 Mininet OS .....	83
6.2.4 X-Ming.....	84
6.2.5 WinSCP.....	85
6.2.6 PuTTY.....	86
6.2.7 Visual Studio Code.....	87
6.2.8 Gephi .....	88

7.ΠΡΟΔΙΑΓΡΑΦΕΣ ΠΕΙΡΑΜΑΤΩΝ .....	90
7.1. Δικτυακές τοπολογίες.....	90
7.1.1 Βασικές τοπολογίες.....	90
7.1.2 Υβριδικές τοπολογίες.....	92
7.1.3 Custom τοπολογίες.....	93
7.1.3.1 Τυχαία Τοπολογία .....	95
7.1.3.2 Μοντέλο Erdős–Rényi .....	95
7.2 Προδιαγραφές πειραμάτων .....	96
7.2.1 Μέτρηση CPU.....	98
7.2.2 Μέτρηση RAM.....	99
7.2.3 Μέτρηση Καθυστέρησης .....	101
7.2.4 Μέτρηση Δημιουργίας – Καταστροφής τοπολογίας.....	102
7.2.5 Μέτρηση Bandwidth.....	104
7.3 Εκτέλεση πειραμάτων .....	105
7.3.1 Υλοποίηση τυχαίων τοπολογιών.....	105
7.3.2 Υλοποίηση τοπολογιών ισορροπημένου δένδρου.....	106
7.3.2.1 Υλοποίηση τοπολογιών ισορροπημένου δένδρου μέσω της βασικής ενσωμάτωσης τοπολογιών στο Mininet.....	106
7.3.2.2 Υλοποίηση τοπολογιών ισορροπημένου δένδρου μέσω της βασικής ενσωμάτωσης τοπολογιών στο Mininet.....	107
7.3.3 Υλοποίηση γραμμικών τοπολογιών .....	108
7.3.3.1 Υλοποίηση γραμμικών τοπολογιών μέσω της βασικής ενσωμάτωσης τοπολογιών στο Mininet .....	108
7.3.3.2 Υλοποίηση γραμμικών τοπολογιών μέσω κώδικα.....	109
7.3.4 Εκκίνηση ελεγκτή POX στο Mininet .....	109
7.3.5 Προσαρμογή τοπολογιών σε διαφορετικό πρωτόκολλο και ελεγκτή .....	110
7.4 Συλλογή γενικών αποτελεσμάτων.....	110
7.4.1 Συλλογή αποτελεσμάτων καθυστέρησης .....	112
8.ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ .....	113

8.1 Ανάλυση CPU .....	113
8.2 Ανάλυση RAM.....	114
8.3 Ανάλυση Bandwidth .....	115
8.4 Ανάλυση Setup Time .....	116
8.5 Ανάλυση Tear Time .....	116
8.6 Ανάλυση Latency .....	117
9. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	118
10. ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ .....	122
ΠΑΡΑΡΤΗΜΑ.....	123
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	134

## ΠΙΝΑΚΑΣ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

ACI.....	Application Centric Infrastructure
API.....	Application Programming Interface
BGP.....	BORDER GATEWAY PROTOCOL
BYOD.....	Bring your own Device.
CLI.....	COMMAND LINE INTERFACE
CPU.....	Central Processing Unit
ER.....	Erdős–Rényi
IDS.....	INTRUSION DETECTION SYSTEM
IETF.....	INTERNET ENGINEERING TASKFORCE
IoT.....	Internet of Things
LAN.....	LOCAL AREA NETWORK
LLDP.....	Link Layer Discovery Protocol
MAC.....	MEDIA ACCESS CONTROL ADDRESS
MAN.....	METROPOLITAN AREA NETWORK
NOE.....	NETWORK OPERATION ENGINE
ODL.....	OPEN DAYLIGHT
OFDP.....	OpenFlow Discovery Protocol
OVSDB.....	OPEN VSWITCH DATABASE MANAGEMENT PROTOCOL
Qos.....	Quality of Service
RAM.....	Random Access Memory

REFS.....Resilient File System

RPC..... REMOTE PROCEDURE CALL

SDN.....Software Defined Networking

SNNP.....Simple Network Management Protocol

SSH.....SECURE SHELL

SSL..... Secure Sockets Layer

TLS..... Transport Layer Security

USP.....Unique Selling Proposition

VLAN..... VIRTUAL LOCAL AREA NETWORK

VM..... Virtual Machine

WAN..... WIDE AREA NETWORK

WI-FI..... WIRELESS FIDELITY

XML..... EXTENSIBLE MARKUP LANGUAGE



## ΑΠΟΔΟΣΗ ΟΡΩΝ / ΓΛΩΣΣΑΡΙΟ

Application Layer (Επίπεδο Εφαρμογών)

Application Programming Interface (Διεπαφή προγραμματισμού εφαρμογών)

Central Processing Unit (Κεντρική Μονάδα Επεξεργασίας)

Centralization (Συγκέντρωση)

Control Plane (Επίπεδο ελέγχου)

Controller (Ελεγκτής)

Custom Topology (Παρεμετροποιήσιμη Τοπολογία)

Erdős–Rényi (Μοντέλο σχεδιασμού γραφημάτων στη μαθηματική θεωρία των γραφημάτων)

Forwarding Plane (Επίπεδο Προώθησης)

Hardware (Υλικό)

Northbound Interface (Βόρεια Διεπαφή)

OpenFlow Discovery Protocol (Πρωτόκολλο Ανακάλυψης OpenFlow)

Random Access Memory (Μνήμη Τυχαίας Προσπέλασης)

Router (Δρομολογητής)

Software (Λογισμικό)

Software Defined Networking (Δίκτυα Καθοριζόμενα από Λογισμικό)

Southbound Interface (Νότια Διεπαφή)

Switch (Μεταγωγέας)

Unique Selling Proposition (Μοναδική Πρόταση Πώλησης)

Virtual Machine (Εικονική Μηχανή)

## 1. ΕΙΣΑΓΩΓΗ

Τα δίκτυα βρίσκονται παντού γύρω μας, αποτελούν κομμάτι της ζωής και της καθημερινότητάς μας . Η πλειοψηφία των αναγκών μιας σύγχρονης και τεχνολογικά ανεπτυγμένης κοινωνίας για να ικανοποιηθεί χρειάζεται ισχυρά και αξιόπιστα δίκτυα. Οι ανάγκες για αποτελεσματικά δίκτυα αυξάνονται εκθετικά με την πάροδο των ετών και την τεχνολογική ανάπτυξη. Πλέον η μεγαλύτερη μερίδα ανθρώπων από όλες τις ηλικιακές ομάδες κάνει χρήση δικτύων καθημερινά και η ποιότητα ζωής τους εξαρτάται από τα δίκτυα ακόμα και εάν αυτό δεν γίνεται με άμεσο τρόπο αντιληπτό. Στις ηλικιακές ομάδες από 0-18 τα δίκτυα είναι ο αρωγός για κυρίως για την ψυχαγωγία. Social media, online gaming, tv-streaming είναι οι συνηθέστερες από τις ατελείωτες χρήσεις των δικτύων. Στους ενήλικες από 18-55 έρχεται και προστίθεται κυρίως και η επαγγελματική αξιοποίηση των δικτύων όπου βοηθούν στο να γίνει κάποιος πιο αποτελεσματικός στην εργασία του και να εξελιχθεί. Τέλος στην ηλικιακή ομάδα των γηραιότερων 55+ τα δίκτυα χρησιμοποιούνται περισσότερο στον τομέα της υγείας και της απομακρυσμένης επικοινωνίας. Η πανδημία COVID-19 που βίωσε ο πλανήτης μας από την αρχή του έτους 2020 άλλαξε πολλά δεδομένα στον τομέα των δικτύων. Ο κόσμος αναγκαστικά έπρεπε να περνάει πολύ περισσότερο χρόνο στο σπίτι. Αυτή η κατάσταση οδήγησε τους ανθρώπους στο να βρουν έξυπνους τρόπους να εξυπηρετήσουν το μεγαλύτερο ποσοστό των καθημερινών αναγκών τους εντός των τειχών του σπιτιού τους.

Έτσι αναπόφευκτα η έννοια του δικτύου γενικότερα ήρθε στην πόρτα όλων των σπιτιών με τη μορφή του μεγαλύτερου γνωστού δικτύου παγκοσμίως το οποίο είναι διαδίκτυο. Οι νέοι πλέον έπρεπε να μορφωθούν εξ αποστάσεως με τη χρήση του διαδικτύου ( πρωτοβάθμια, δευτεροβάθμια και τριτοβάθμια εκπαίδευση ) . Οι ενήλικες κλήθηκαν να εργαστούν ως επί το πλείστον εξ αποστάσεως και οι ηλικιωμένοι και οι ευπαθείς ομάδες έπρεπε να διεκδικήσουν την φροντίδα και την περίθαλψή τους με διαφορετικό πλέον τρόπο με την βοήθεια του διαδικτύου. Αυτή η κατάσταση οδήγησε σε πρωτοφανή αύξηση όχι μόνο των χρηστών του διαδικτύου αλλά και του πλήθους των διαφορετικών συσκευών που χρησιμοποιεί ο κάθε χρήστης για να αποκτήσει πρόσβαση σε αυτό. Ως αποτέλεσμα αποκαλύφθηκαν κάποιες αδυναμίες στα ήδη υπάρχοντα δίκτυα σε παγκόσμιο επίπεδο καθώς και δημιουργήθηκαν νέες. Οι εταιρίες δικτυακών παροχών καθώς και οι ερευνητές διαπίστωσαν ότι πρέπει να αυξηθεί η χωρητικότητα των δικτύων καθώς και η ταχύτητα , διαχείριση και αξιοπιστία τους λόγω του υπερβολικού φόρτου. Τα λεγόμενα παραδοσιακά δίκτυα που χρησιμοποιούνταν για αρκετά χρόνια ξεκίνησαν να καταρρέουν και δεν

μπορούσαν να ανταπεξέλθουν καθότι στηρίζονταν κατά κύριο λόγο σε φυσικές υποδομές. Τεχνολογίες όπως το SDN όπου υπήρχαν τα τελευταία 10 χρόνια ( κυρίως από το 2013 και έπειτα) ξεκίνησαν να ερευνώνται εκτενέστερα και να εφαρμόζονται στα παγκόσμια δίκτυα προκειμένου να τα ενισχύσουν και να εξασφαλίσουν την ομαλή επιβίωση του κόσμου. Μέσω του SDN ενισχύονται οι δυνατότητες του δικτύου καθώς και γίνεται απλούστερη η δομή του και ο κύριος στόχος τους είναι η αποδοτικότερη διαχείριση των δικτύων.

## **1.1 Αντικείμενο και στόχος της μεταπτυχιακής εργασίας**

Στην παρούσα μεταπτυχιακή εργασία θα μελετηθεί η αρχιτεκτονική και η δομή των δικτύων SDN. Θα δημιουργηθούν τοπολογίες δικτύων μέσω προσομοιώσεων και θα αναλυθεί η λειτουργία τους και η απόδοσή τους. Συγκεκριμένα θα συγκριθεί η απόδοση μεταξύ των διαφορετικών τοπολογιών. Στην πορεία θα δειχθεί πως ο συνολικός αριθμός των switch, τα οποία αποτελούν βασικό πυλώνα στην δομή της αρχιτεκτονικής SDN, επηρεάζει και επιβαρύνει τη συνολική απόδοση του δικτύου. Οι κύριες μετρήσεις που θα ληφθούν προκειμένου να εξαχθούν συμπεράσματα με ακρίβεια είναι η απασχόληση της CPU, της μνήμης ram και η καθυστέρηση της μεταφοράς πακέτων μεταξύ κόμβων. Για να επιτευχθούν τα παραπάνω με τη μορφή της προσομοίωσης θα χρησιμοποιηθεί ο εξομοιωτής Mininet σε υπολογιστή με λειτουργικό σύστημα Windows. Επιπλέον λογισμικά θα χρησιμοποιηθούν σε συνεργασία με το Mininet έτσι ώστε να διασφαλιστεί η ακεραιότητα και το πλήθος των αποτελεσμάτων. Με τον τρόπο αυτόν θα μελετηθεί η συμπεριφορά και η προσαρμοστικότητα των δικτύων SDN. Ο ελεγκτής που θα χρησιμοποιηθεί στο Mininet είναι ο POX και επιπλέον θα γίνει ανάλυση του πρωτοκόλλου δημιουργίας τοπολογιών OFDP καθώς μέσω αυτού θα δημιουργηθούν τα εικονικά δίκτυα που θα μελετηθούν παρακάτω και θα εξεταστεί η απόδοση και η επεκτασιμότητά του. Στόχος είναι η εξαγωγή συμπερασμάτων για τη λειτουργία των SDN με ελεγκτή POX και ειδικότερα η χρήση του πρωτοκόλλου OFDP. Ο λόγος χρήσης των τυχαίων τοπολογιών είναι οι επιδράσεις τους στα παραδοσιακά δίκτυα και αναδεικνύει το ερώτημα για το πώς αυτά τα τυχαία γραφήματα μπορούν να επηρεάσουν το OpenFlow ως την εξέλιξή του σε ακόμα πιο σύγχρονο πρωτόκολλο δημιουργίας τοπολογιών δικτύου . Δεδομένο ότι οι τεχνολογίες δικτύου έχουν ισχυρή σχέση με γραφήματα δικτύου και κατά συνέπεια με την έννοια της θεωρίας γραφημάτων στα μαθηματικά, ένα αναλυτικό μοντέλο υπολογισμού, σύγκρισης και πρόβλεψης καθιερώνεται μέσω των προσομοιώσεων σε ρεαλιστικές πλατφόρμες έτσι ώστε να αποκτηθεί γρηγορότερη εφαρμογή σε πραγματικά δίκτυα πραγματικού χρόνου.

## 1.2 Δομή μεταπτυχιακής εργασίας

Το 1<sup>ο</sup> κεφάλαιο αποτελείται από την εισαγωγή της μεταπτυχιακής μου εργασίας , την περιγραφή του αντικειμένου της καθώς και την παρούσα δομική ανάλυσή της. Στο 2<sup>ο</sup> κεφάλαιο γίνεται αναφορά σε εργασίες και έρευνες σχετικές με το θέμα της μεταπτυχιακής εργασίας. Το 3<sup>ο</sup> κεφάλαιο περιέχει τον ορισμό SDN και περιγράφει την αρχιτεκτονική της δομής του και τα χαρακτηριστικά λειτουργίας του. Το 4<sup>ο</sup> κεφάλαιο αναφέρεται στους ελεγκτές SDN. Στο 5<sup>ο</sup> κεφάλαιο αναλύονται τα πρωτόκολλα λειτουργίας των ελεγκτών. Στο 6<sup>ο</sup> κεφάλαιο γίνεται περιγραφή του υλικού και του λογισμικού που χρησιμοποιήθηκε για την υλοποίηση των προσομοιώσεων με σημαντικότερο εκ 'αυτών να είναι ο γνωστός εξομοιωτής Mininet. Στο 7<sup>ο</sup> κεφάλαιο έχουν θέση οι προδιαγραφές και οι ιδιότητες των πειραμάτων που θα εκτελεστούν καθώς και η απαραίτητη ρύθμιση των προγραμμάτων εκτέλεσης. Στο 8<sup>ο</sup> κεφάλαιο θα εξαχθούν και θα αναλυθούν τα αποτελέσματα των στατιστικών από την χρήση και τον φόρτο του συστήματος. Τέλος στο 9<sup>ο</sup> κεφάλαιο θα παρουσιαστούν τα συμπεράσματα που προκύπτουν από την παρούσα έρευνα ενώ στο 10<sup>ο</sup> θα γίνει αναφορά στο πως η παρούσα εργασία μπορεί να αναλυθεί ακόμα περισσότερο μελλοντικά.

## 2. ΣΧΕΤΙΚΕΣ ΕΡΕΥΝΕΣ

Έχουν γίνει αρκετές δημοσιεύσεις ερευνών πάνω στο κομμάτι των SDN , με τη χρήση POX ελεγκτή καθώς και OFDP για τη δημιουργία τοπολογιών. Ελάχιστες όμως χρησιμοποιούν τυχαίες τοπολογίες για την ερμηνεία των επιδόσεων SDN. Σε παλαιότερή μου προσωπική έρευνα μελετήθηκαν τα αποτελέσματα απόδοσης των δοκιμών του Software-Defined Networking (SDN) που πραγματοποιήθηκαν σε τυπικές τοπολογίες δικτύου με χρήση προσομοίωσης. Παράλληλα συγκρίθηκε η απόδοση των τυπικών τοπολογιών με αυτή των τυχαίων. Συγκεκριμένα, οι μετρήσεις απόδοσης που εξετάστηκαν περιλάμβαναν : τον χρόνο ρύθμισης και διακοπής της τοπολογίας, τη χρήση CPU και RAM του συστήματος και την καθυστέρηση μεταφοράς πακέτων μεταξύ των κόμβων. Η όλη έρευνα διεξήχθη σε υπολογιστή με Windows χρησιμοποιώντας μια εικονική μηχανή για την εκτέλεση ενός εξομοιωτή Mininet παρόμοια με ότι θα χρησιμοποιήσουμε στην παρούσα εργασία. Από τη σχολαστική ανάλυση των αποτελεσμάτων, αξίζει να αναφερθούν τα εξής: (i) ο συνολικός αριθμός switch σε μια αρχιτεκτονική SDN έχει σημαντική επίδραση στο φόρτο της CPU. (ii) Η χρήση RAM εξαρτάται από τον αριθμό των κεντρικών υπολογιστών και σε περιπτώσεις υπερβολικού φορτίου παρουσιάζει πολύ μεγαλύτερη αύξηση σε σύγκριση με τη χρήση της CPU. (iii) Η συνολική απόδοση εξαρτάται σημαντικά από τον τύπο της τοπολογίας και τις ιδιότητές της. Τα πειράματα τότε είχαν διεξαχθεί σε τυπικό και περιορισμένο εύρος συσκευών. [15].

Στην εργασία του ο Guo δημιούργησε διαφορετικούς τύπους τοπολογιών δικτύου για ανάλυση, συμπεριλαμβανομένης της τοπολογίας δακτυλίου, τοπολογίας δέντρων και τυχαίων τοπολογιών μοντέλου Erdos-Renyi . Στα δημιουργημένα τυχαία δίκτυα, η πιθανότητα από μια άκρη μεταξύ αυθαίρετων δύο κορυφών ορίστηκε στο 0,4. Επομένως αυτά τα τυχαία δίκτυα ήταν ως επί το πλείστον πυκνά δίκτυα με μικρό μέσο μήκος διαδρομής. Τα δημιουργημένα δίκτυα δακτυλίου είχαν το μεγαλύτερο μέσο μήκος διαδρομής μεταξύ των τριών τοπολογιών ενώ το μέσο μήκος διαδρομής του δέντρου ήταν στο ενδιάμεσο, όλα τα πειράματα επαναλήφθηκαν 100 φορές. Όλοι οι κόμβοι υπόκεινταν σε κοινή πιθανότητα αποτυχίας. Πενήντα κόμβοι είχαν συνηθίσει δημιουργούν τους τρεις τύπους δικτύων. Τα αποτελέσματα δείχνουν ότι η αναμενόμενη ανθεκτικότητα του δικτύου είναι αντιστρόφως ανάλογη με την μέσο μήκος διαδρομής της τοπολογίας του δικτύου και άρα τα δίκτυα τυχαίων τοπολογιών αποδίδουν καλύτερα και τα δίκτυα δακτυλίου είναι λιγότερο ανθεκτικά. [12]

Σε άλλη έρευνα αναλύθηκε η απόδοση του προτεινόμενου μηχανισμού ανακάλυψης, που βασίζεται κυρίως σε πρωτόκολλο OFDP ως προς το γενικό φόρτο. Ερευνήθηκαν διάφορες τοπολογίες και επικεντρώθηκε σε τυχαία δίκτυα που βασίζονται στο μοντέλο Erdős–Rényi. Η μελέτη έδειξε ότι οι προσπάθειες των ερευνητών επικεντρώνονται στη μείωση του αριθμού των μηνυμάτων που φτάνουν στον ελεγκτή. Ωστόσο, η απόδοση και η επεκτασιμότητα των δικτύων SDN εξαρτάται περισσότερο από άλλους παράγοντες, όπως το φορτίο CPU, χρήση μνήμης, τοπολογία δικτύου και χρόνος που απαιτείται για την ανακάλυψη της τοπολογίας. Ελέγχθηκε η επεκτασιμότητα των OFDP και OFDPv2 για ένα ευρύ φάσμα τυχαίων δικτύων που βασίζονται στο μοντέλο Erdős–Rényi. Από το πειραματικά αποτελέσματα φάνηκε ότι τα πρωτόκολλα καταναλώνουν σχεδόν ίσους πόρους (CPU και RAM) ενώ το OFDP απαιτεί μεγαλύτερο χρόνο για την ανακάλυψη τοπολογίας από το OFDPv2 για την ίδια τοπολογία. [3].

## 3. ΠΕΡΙΓΡΑΦΗ ΤΟΥ SDN

### 3.1 Η πορεία του SDN

Το SDN είναι μια συναρπαστική τεχνολογία που επιτρέπει την καινοτομία στον τρόπο σχεδιασμού και διαχείρισης δικτύων. Αν και αυτή η τεχνολογία φαίνεται να εμφανίστηκε ξαφνικά, το SDN αποτελεί μέρος μιας μακράς ιστορίας προσπαθειών να γίνουν τα δίκτυα υπολογιστών πιο προγραμματιζόμενα. Η ιδέα δημιουργήθηκε τη δεκαετία του '90 και ύστερα από μακροχρόνιες προσπάθειες διαχωρισμού του επίπεδο ελέγχου και δεδομένων φτάνοντας στο πρωτόκολλο OpenFlow όπου και πέτυχε λειτουργικά συστήματα δικτύου. Τα τελευταία χρόνια, το SDN έχει κερδίσει σημαντικό έδαφος στη βιομηχανία. Πολλά εμπορικά switch υποστηρίζουν το OpenFlow API. Οι αρχικοί προμηθευτές που υποστήριζαν το OpenFlow ήταν οι HP, NEC και Pronto. Αυτή η λίστα έχει επεκταθεί δραματικά. Έχουν προκύψει πολλές διαφορετικές πλατφόρμες ελεγκτών [2, 7,13,21,28,29,41]. Οι προγραμματιστές έχουν χρησιμοποιήσει αυτές τις πλατφόρμες για να δημιουργήσουν πολλές εφαρμογές, όπως δυναμικό έλεγχο πρόσβασης [1,11], φόρτωση διακομιστή, εξισορρόπηση [14,18], εικονοποίηση δικτύου [17,37], ενεργειακά αποδοτική δικτύωση [4] VMs και έλεγχο κινητικότητας χρηστών [6,19]. Πρώιμες εμπορικές επιτυχίες, όπως η διαχείριση κυκλοφορίας ευρείας περιοχής της Google [20] και η πλατφόρμα εικονικοποίησης δικτύου της Nicira [37], έχουν συγκεντρώσει σημαντική προσοχή του κλάδου. Πολλές από τις μεγαλύτερες εταιρείες τεχνολογίας πληροφοριών στον κόσμο (π.χ. πάροχοι cloud, πάροχοι τηλεπικοινωνιών, προμηθευτές εξοπλισμού, και εταιρείες χρηματοοικονομικών υπηρεσιών) έχουν ενταχθεί στη βιομηχανία SDN μέσω κοινοπραξιών όπως το Open Networking Foundation [40] και το Open Daylight [39].

Αν και ο ενθουσιασμός για το SDN αυξάνεται τα τελευταία χρόνια, πολλές από τις ιδέες στις οποίες βασίζεται το SDN έχουν εξελιχθεί τα τελευταία είκοσι χρόνια (ή και περισσότερο). Κατά κάποιο τρόπο, το SDN επανεξετάζει ιδέες από τα αρχικά δίκτυα τηλεφωνίας, τα οποία χρησιμοποίησαν έναν σαφή διαχωρισμό των επιπέδων ελέγχου και δεδομένων για να απλοποιήσουν τη διαχείριση του δικτύου και την ανάπτυξη νέων υπηρεσιών. Ωστόσο, ανοιχτές διεπαφές όπως το OpenFlow επιτρέπει περισσότερη καινοτομία στις πλατφόρμες ελεγκτών και στις εφαρμογές από ό,τι ήταν δυνατό σε κλειστά δίκτυα σχεδιασμένα για ένα στενό φάσμα υπηρεσιών τηλεφωνίας. Επιπλέον το SDN μοιάζει με παλαιότερες έρευνες για την ενεργή δικτύωση, οι οποίες δημιούργησαν ένα όραμα για

προγραμματιζόμενα δίκτυα, με έμφαση στα προγραμματιζόμενα επίπεδα δεδομένων. Σχετίζεται επίσης με προηγούμενες εργασίες για τον διαχωρισμό του ελέγχου και επίπεδα δεδομένων σε δίκτυα υπολογιστών.

Αυτό που αξίζει να αναφερθεί είναι ότι αυτό που περιγράφουμε σήμερα ως SDN είναι το project που ξεκίνησε το πανεπιστήμιο του Stanford από το 2008 δημιουργώντας το πρώτο πρωτόκολλο OpenFlow και στην πορεία από το 2011 και έπειτα όπου ξεκίνησε να καθιερώνεται ο ορισμός προσαρμόστηκε στα σημερινά δεδομένα και καταλαμβάνει όλο το εύρος τεχνολογιών πέρα από το αρχικό project.[10,23]

### **3.2 Ο ορισμός SDN**

Το Software-Defined Networking (SDN) είναι μια προσέγγιση δικτύωσης που χρησιμοποιεί controllers που βασίζονται σε software ή APIs για την επικοινωνία με την υποκείμενη υποδομή υλικού και την άμεση κίνηση σε ένα δίκτυο.

Αυτό το μοντέλο διαφέρει από αυτό των παραδοσιακών δικτύων, τα οποία χρησιμοποιούν αποκλειστικά συσκευές hardware (routers και switch) για τον έλεγχο της κυκλοφορίας του δικτύου. Το SDN μπορεί να δημιουργήσει και να ελέγξει ένα εικονικό δίκτυο ή να ελέγξει ένα παραδοσιακό δίκτυο μέσω λογισμικού.

Ενώ η εικονικοποίηση δικτύου επιτρέπει την τμηματοποίηση διαφορετικών εικονικών δικτύων σε ένα ενιαίο φυσικό δίκτυο τη σύνδεση συσκευών σε διαφορετικά φυσικά δίκτυα για να δημιουργήσουν ένα ενιαίο εικονικό δίκτυο, το SDN επιτρέπει έναν νέο τρόπο ελέγχου της δρομολόγησης των πακέτων δεδομένων μέσω ενός κεντρικού διακομιστή .

Συνεπώς ως ορισμό έχουμε ότι το SDN πετυχαίνει τον επιτυχή και λειτουργικό διαχωρισμό του Control Plane σε ένα δίκτυο από το Forwarding Plane.[48]

### **3.3 Απεικόνιση του SDN**

Το SDN στον τομέα του σχεδιασμού και της αρχιτεκτονικής αποτελείται από τρία επίπεδα:

#### **i. Application Layer**

Αυτό το επίπεδο χαρακτηρίζεται ως ανοιχτή περιοχή για την ανάπτυξη όσο το δυνατόν περισσότερων καινοτόμων εφαρμογών αξιοποιώντας όλες τις πληροφορίες δικτύου σχετικά με την τοπολογία του δικτύου, την κατάσταση του δικτύου, τα στατιστικά στοιχεία του δικτύου κ.λπ. διαμόρφωση και διαχείριση, παρακολούθηση



δικτύου, αντιμετώπιση προβλημάτων δικτύου, πολιτικές και ασφάλεια δικτύου. Τέτοιες εφαρμογές SDN μπορούν να παρέχουν διάφορες λύσεις από άκρο σε άκρο για δίκτυα επιχειρήσεων και κέντρων δεδομένων του πραγματικού κόσμου. Οι προμηθευτές δικτύου έρχονται με το σύνολο των εφαρμογών SDN τους.

## **ii. Control Layer**

Εδώ βρίσκεται η βάση του επιπέδου ελέγχου όπου η έξυπνη λογική στους ελεγκτές SDN βρίσκεται για τον έλεγχο της υποδομής δικτύου. Αυτή είναι η περιοχή όπου κάθε προμηθευτής δικτύου εργάζεται για να βρει τα δικά του προϊόντα για ελεγκτή και πλαίσιο SDN. Σε αυτό το επίπεδο, πολλή επιχειρησιακή λογική γράφεται στον ελεγκτή για την ανάκτηση και διατήρηση διαφορετικών τύπων πληροφοριών δικτύου, λεπτομέρειες κατάστασης, λεπτομέρειες τοπολογίας, λεπτομέρειες στατιστικών στοιχείων και πολλά άλλα. Δεδομένου ότι ο ελεγκτής SDN προορίζεται για τη διαχείριση δικτύων, επομένως πρέπει να έχει λογική ελέγχου για περιπτώσεις χρήσης δικτύου πραγματικού κόσμου, όπως μεταγωγή, δρομολόγηση, L2 VPN, L3 VPN, κανόνες ασφαλείας τείχους προστασίας, DNS, DHCP και ομαδοποίηση. Αρκετοί προμηθευτές δικτύωσης και ακόμη και κοινότητες ανοιχτού κώδικα εργάζονται για την υλοποίηση αυτών των περιπτώσεων χρήσης στους ελεγκτές SDN τους. Μόλις εφαρμοστούν, αυτές οι υπηρεσίες εκθέτουν τα API τους στο ανώτερο επίπεδο (επίπεδο εφαρμογής), κάτι που διευκολύνει τους διαχειριστές δικτύου που στη συνέχεια χρησιμοποιούν εφαρμογές πάνω από τους ελεγκτές SDN για τη διαμόρφωση, τη διαχείριση και την παρακολούθηση του υποκείμενου δικτύου. Το επίπεδο ελέγχου βρίσκεται στη μέση και διαθέτει δύο τύπους διεπαφών - Northbound και Southbound.

## **iii. Forwarding Layer**

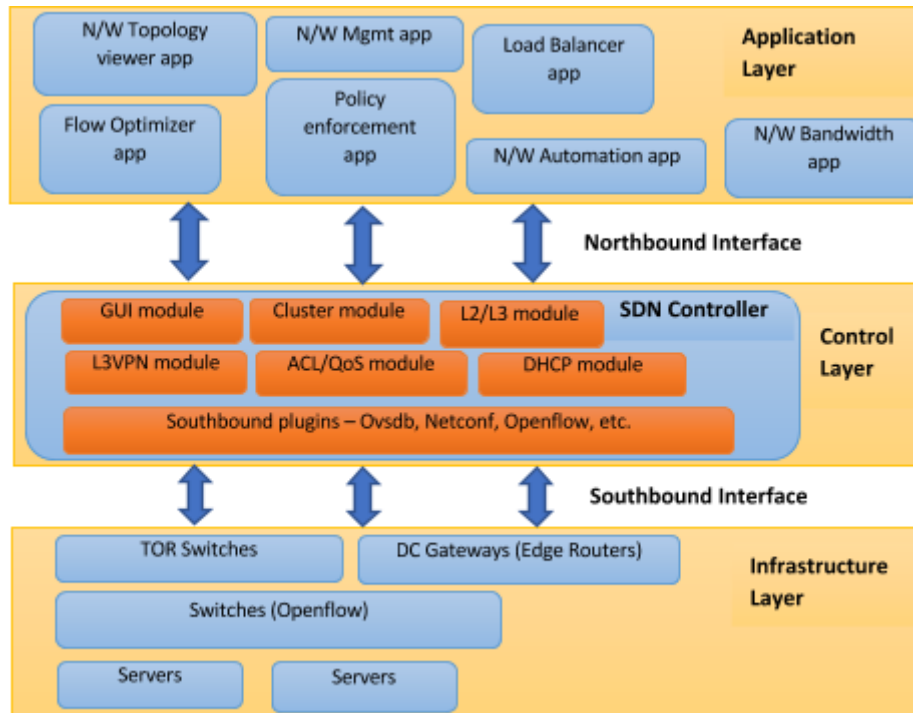
Αυτό το επίπεδο αποτελείται από διάφορους εξοπλισμούς δικτύωσης που σχηματίζουν το υποκείμενο δίκτυο για την προώθηση της κυκλοφορίας του δικτύου. Θα μπορούσε να είναι ένα σύνολο switch και router στο κέντρο δεδομένων. Αυτό το επίπεδο θα ήταν το φυσικό επίπεδο πάνω στο οποίο θα καθιερωνόταν η εικονικοποίηση δικτύου μέσω του επιπέδου ελέγχου (όπου οι ελεγκτές SDN θα διαχειρίζονταν το υποκείμενο φυσικό δίκτυο).

## **iv. Northbound Interface**

Προορίζεται για επικοινωνία με το ανώτερο επίπεδο εφαρμογής και γενικά θα πραγματοποιηθεί μέσω των API των ελεγκτών SDN.

v. **Southbound Interface**

Προορίζεται για επικοινωνία με το κατώτερο επίπεδο υποδομής στοιχείων δικτύου και γενικά θα πραγματοποιηθεί μέσω πρωτοκόλλων νότιας κατεύθυνσης – Openflow, Netconf, Ovsdb, κ.λπ.



Εικόνα 3.1 : Σχηματική απεικόνιση της αρχιτεκτονικής SDN. Προσαρμογή από [5].

### 3.4 Τι προσφέρει το SDN

Το SDN αντιπροσωπεύει ένα σημαντικό βήμα προς τα εμπρός από την παραδοσιακή δικτύωση, καθώς επιτρέπει τα ακόλουθα:

- i. **Αυξημένος έλεγχος με μεγαλύτερη ταχύτητα και ευελιξία:** Αντί να προγραμματίζουν με μη αυτόματο τρόπο πολλαπλές συσκευές υλικού για συγκεκριμένους προμηθευτές, οι προγραμματιστές μπορούν να ελέγχουν τη ροή της κυκλοφορίας σε ένα δίκτυο απλώς προγραμματίζοντας έναν ανοιχτό πρότυπο ελεγκτή που βασίζεται σε λογισμικό. Οι διαχειριστές δικτύωσης έχουν επίσης μεγαλύτερη ευελιξία στην επιλογή εξοπλισμού δικτύωσης, καθώς μπορούν να

επιλέξουν ένα μόνο πρωτόκολλο για επικοινωνία με οποιονδήποτε αριθμό συσκευών υλικού μέσω ενός κεντρικού ελεγκτή.

- ii. **Προσαρμόσιμη υποδομή δικτύου:** Με ένα δίκτυο που καθορίζεται από λογισμικό, οι διαχειριστές μπορούν να διαμορφώσουν τις υπηρεσίες δικτύου και να εκχωρήσουν εικονικούς πόρους για να αλλάξουν την υποδομή δικτύου σε πραγματικό χρόνο μέσω μιας κεντρικής τοποθεσίας. Αυτό επιτρέπει στους διαχειριστές δικτύου να βελτιστοποιούν τη ροή δεδομένων μέσω του δικτύου και να δίνουν προτεραιότητα σε εφαρμογές που απαιτούν περισσότερη διαθεσιμότητα.
- iii. **Ισχυρή ασφάλεια:** Ένα δίκτυο που καθορίζεται από λογισμικό παρέχει ορατότητα σε ολόκληρο το δίκτυο, παρέχοντας μια πιο ολιστική εικόνα των απειλών ασφαλείας. Με τον πολλαπλασιασμό των έξυπνων συσκευών που συνδέονται στο διαδίκτυο, το SDN προσφέρει σαφή πλεονεκτήματα έναντι της παραδοσιακής δικτύωσης. Οι χειριστές μπορούν να δημιουργήσουν ξεχωριστές ζώνες για συσκευές που απαιτούν διαφορετικά επίπεδα ασφαλείας ή να θέσουν αμέσως σε καραντίνα συσκευές που έχουν παραβιαστεί, ώστε να μην μπορούν να μολύνουν το υπόλοιπο δίκτυο.
- iv. **Η βασική διαφορά μεταξύ του SDN και της παραδοσιακής δικτύωσης είναι η υποδομή:** το SDN βασίζεται σε λογισμικό, ενώ η παραδοσιακή δικτύωση βασίζεται σε υλικό. Επειδή το επίπεδο ελέγχου βασίζεται σε λογισμικό, το SDN είναι πολύ πιο ευέλικτο από την παραδοσιακή δικτύωση. Επιτρέπει στους διαχειριστές να ελέγχουν το δίκτυο, να αλλάζουν τις ρυθμίσεις παραμέτρων, να παρέχουν πόρους και να αυξάνουν τη χωρητικότητα του δικτύου — όλα από μια κεντρική διεπαφή χρήστη, χωρίς να απαιτείται περισσότερο υλικό.
- v. **Υπάρχουν επίσης διαφορές ασφαλείας** μεταξύ του SDN και της παραδοσιακής δικτύωσης. Χάρη στη μεγαλύτερη ορατότητα και τη δυνατότητα καθορισμού ασφαλών διαδρομών, το SDN προσφέρει καλύτερη ασφάλεια με πολλούς τρόπους. Ωστόσο, επειδή τα δίκτυα που ορίζονται από λογισμικό χρησιμοποιούν έναν κεντρικό ελεγκτή, η ασφάλιση του ελεγκτή είναι ζωτικής σημασίας για τη διατήρηση ενός ασφαλούς δικτύου.

### 3.5. Επιχειρήσεις που χρησιμοποιούν τεχνολογία SDN

Οι επιχειρήσεις και οργανισμοί που χρησιμοποιούν τεχνολογίες SDN αποζητούν ασφάλεια και διαθέτουν μεγάλη κίνηση δεδομένων η οποία με την πάροδο των ετών πηγαίνει ως επί το πλείστον σε διαχείριση cloud δεδομένων.

Cisco
Citrix
Dell Inc.
Facebook
Fortinet
Google
HP
Kaspersky Lab
Microsoft
NSA
Riverbed Technology, Inc.
Twitter
Universidad De Granada
Vmware
Vodafone Group plc
Yahoo

Πίνακας 3.1 : Εταιρίες κολοσσοί που χρησιμοποιούν τεχνολογίες SDN

#### 3.5.1 Πάροχοι SDN

Μεγάλες επιχειρήσεις χρησιμοποιούν τεχνολογία SDN διότι τους προσφέρει ευελιξία και αυξημένη απόδοση. Προκειμένου να αποκτήσουν πρόσβαση και διαχείριση στην τεχνολογία απευθύνονται στους τεχνολογικούς κολοσσούς που παρέχουν την τεχνολογία αυτή.

1. Cisco Meraki
2. Cisco+ NaaS
3. Cradlepoint NetCloud
4. Dell SDN Solution
5. IBM SDN Services
6. Junos Space
7. Masergy
8. Nuage Networks
9. Pica8
10. Riverbed SteelConnect

Πίνακας 3.2 : Κορυφαίοι πάροχοι υπηρεσιών SDN για το έτος 2022.

COMPANIES	ΓΕΝΙΚΑ	ΛΕΙΤΟΥΡΓΙΕΣ	USP	ΤΙΜΟΛΟΓΗΣΗ
Cisco Meraki	Το Meraki εξαγοράστηκε από τη Cisco το 2012 και ειδικεύεται σε λύσεις SDN	Διαθέτει 140+ προκατασκευασμένες ενσωματώσεις και πρόσθετα όπως το Meraki Insight.	Μπορεί να αναπτυχθεί χωρίς υπερβολική διαμόρφωση.	Ξεκινάει από τα 150\$
Cisco+ NaaS	Κυκλοφόρησε το 2021 και μετατρέπει τις λειτουργίες δικτύου, την αποθήκευση και την ασφάλεια σε μια υπηρεσία που καθορίζεται από SDN στο cloud.	Είναι πλήρως ενεργοποιημένο στο cloud και υποστηρίζεται από ένα μεγάλο δίκτυο συνεργατών της Cisco για πλήρη διαχείριση του κύκλου ζωής.	Είναι μια ολοκληρωμένη λύση για όλες τις απαιτήσεις υποδομών πληροφορικής.	Αποκαλύπτεται ύστερα από ενδεχόμενο συμφωνίας
Cradlepoint NetCloud	Ενοποιεί περιβάλλοντα ασύρματου δικτύου χρησιμοποιώντας τεχνολογία SDN	Είναι εξαιρετικά προγραμματιζόμενο, διαθέτει μεγάλη γκάμα SDK και αποκλειστικές ομάδες σε βασικές περιοχές.	Συνδυαστική διαχείριση LTE, 5G, WAN και IoT μέσω μιας πλατφόρμας SDN.	Ξεκινάει από 40\$
Dell SDN Solution	Κυκλοφόρησε το 2019, συνδυάζει Dell EMC με το VeloCloud της VMware.	Η τηλεμετρία ροής παρέχει λεπτομερείς πληροφορίες και υπάρχει λεπτομερής έλεγχος μέσω ενός CLI.	Απλοποιεί την ενορχήστρωση υφάσματος του κέντρου δεδομένων καθώς και τη διαχείριση δικτύου.	Προσαρμοζόμενη τιμολόγηση
IBM SDN Services	Προσφέρει συμβουλευτικές υπηρεσίες, αξιολόγηση, στρατηγική και ανάπτυξη SDN .	Προσφέρει μεγαλύτερη ορατότητα και προγραμματιζόμενη διακυβέρνηση.	Διαθέτει δυνατότητες αυτοθεραπείας και αυτόματης κλιμάκωσης της IBM.	Προσαρμοζόμενη τιμολόγηση
Junos Space	Η Juniper Networks προσφέρει αυτήν τη λύση SDN.	Πλατφόρμα διαχείρισης ασφαλείας Δικτύου Junos για τη διαχείριση διακριτών δυνατοτήτων SDN.	Είναι σύμφωνο με το πλαίσιο FCAPS του Διεθνούς Οργανισμού Τυποποίησης	3.718\$ για μία τριετή άδεια
Masergy	Προσφέρει λύσεις SDN, ενοποιημένες επικοινωνίες, ασφάλεια και κέντρα επαφής.	Ενσωματώνεται με όλα τα δημόσια και ιδιωτικά cloud και διατίθεται σε πλήρως διαχειριζόμενες παραλλαγές.	Είναι εξαιρετικά ευέλικτο και συνεργάζεται με την Fortinet για ασφάλεια.	Προσαρμοζόμενη τιμολόγηση
Nuage Networks	Είναι ένα εργαλείο της Nokia που προσφέρει εικονικοποίηση δικτύου και αυτοματισμό δικτύου από άκρο σε άκρο.	Είναι χτισμένο σε ανοιχτή αρχιτεκτονική και διαθέτει ένα από τα κορυφαία προγράμματα εκπαίδευσης SDN του κλάδου.	Τροφοδοτείται από το κορυφαίο στον κλάδο λειτουργικό Σύστημα Service Router της Nokia.	Προσαρμοζόμενη τιμολόγηση
Pica8	Στοχεύει να παρέχει στις εταιρείες ισχυρές εγκαταστάσεις ανοιχτού δικτύου χωρίς να απαιτείται προμηθευτής	Προσφέρει έναν προγραμματιζόμενο ελεγκτή δικτύου, πίνακες εργαλείων σε πραγματικό χρόνο και μια εξαιρετική βάση γνώσεων.	Συνδυάζει απρόσκοπτα την ευελιξία και τη χρήση ανοιχτού κώδικα με την απόδοση του δικτύου.	Ξεκινάει από τα 1200\$
Riverbed SteelConnect	Το Riverbed προσφέρει υλικό και λογισμικό δικτύου, το οποίο περιλαμβάνει μια	Έχει σχεδιαστεί για το υβριδικό cloud και υποστηρίζει εκτεταμένο	Γίνεται χρήση του προγράμματος πρόγραμμα GCS Credits για αγορά	Ανάλογα με το επιλεγόμενο επίπεδο

	κεντρική πλατφόρμα διαχείρισης.	αυτοματισμό και ενοποίηση.	υπηρεσιών πριν από την ανάπτυξη.	
--	---------------------------------	----------------------------	----------------------------------	--

Πίνακας 3.3 : Σύγκριση παρόχων υπηρεσιών SDN .

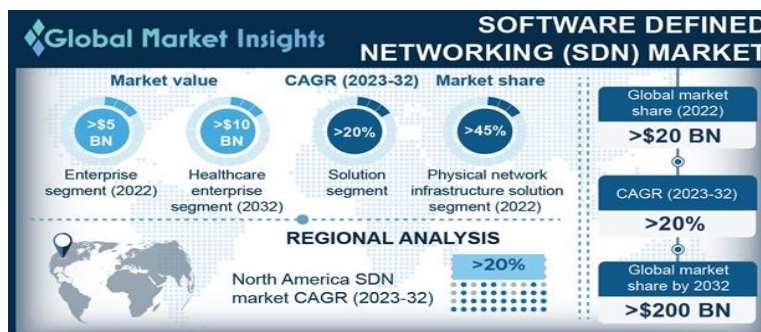
### 3.5.2 Μερίδιο αγοράς SDN

Η αγορά SDN είχε εκτιμώμενη αξία 9,2 δισεκατομμυρίων δολαρίων το 2020. Προβλέπεται ότι θα φτάσει τα 35,6 δισεκατομμύρια δολάρια έως το 2026, μετά από σύνθετο ετήσιο ρυθμό ανάπτυξης (CAGR) 25,4% από το 2022 έως το 2027. Πολλές γεωγραφικές περιοχές οδηγούν την ανάπτυξη στην αγορά SDN:

- Η αγορά των ΗΠΑ είχε εκτιμώμενη αξία 4,1 δισεκατομμυρίων δολαρίων το 2020
- Η Κίνα έχει το δεύτερο μεγαλύτερο μερίδιο αγοράς, που προβλέπεται να φτάσει τα 7,8 δισεκατομμύρια δολάρια έως το 2027, διατηρώντας ένα CAGR 26,2%
- Η Ιαπωνία και ο Καναδάς αναμένεται να αναπτυχθούν κατά 14,7% και 18,1% σε σχέση με την πρόβλεψη από το 2020 έως το 2027
- Η Γερμανία προβλέπεται να αναπτυχθεί κατά 17,2% CAGR έως το 2027
- Η αγορά Ασίας-Ειρηνικού εκτιμάται ότι θα φτάσει τα 5,7 δισεκατομμύρια δολάρια μέχρι το 2027

Ανά κλάδο, η αγορά επαγγελματικών υπηρεσιών προβλέπεται να διατηρήσει το μεγαλύτερο κομμάτι της αγοράς SDN με μερίδιο εσόδων 70%. Άλλοι αξιόλογοι κλάδοι περιλαμβάνουν:

- Τραπεζικές, χρηματοοικονομικές υπηρεσίες και ασφάλειες (BFSI)
- Λιανική και καταναλωτικά αγαθά
- Κυβέρνηση
- Φροντίδα υγείας
- Κατασκευή και υπηρεσίες με δυνατότητα πληροφορικής



Εικόνα 3.2: Παγκόσμιο μερίδιο αγοράς SDN. Προσαρμόστηκε από [44].

### 3.5.3 Παραδείγματα βιομηχανικής χρήσης SDN

Ακολουθούν αρκετά παραδείγματα για το πώς οι οργανισμοί σε διάφορους κλάδους έχουν χρησιμοποιήσει λύσεις SDN για την καλύτερη διαχείριση των εσωτερικών τους δικτύων.

- Ελικόπτερα Airbus

Η Airbus Helicopters είναι ένα κατασκευαστικό τμήμα της Airbus. Με έδρα τη Γαλλία, είναι μία από τις μεγαλύτερες εταιρείες στην αγορά της, τόσο από άποψη παραγωγής όσο και από άποψη εσόδων. Το γερμανικό τμήμα, Airbus Helicopters Germany, χρειάστηκε να αναβαθμίσει το υλικό του δικτύου του αντικαθιστώντας τα switch Nexus 2000 με νέο υλικό της σειράς Nexus 900. Μετά τη μετάβαση στην υποδομή της Cisco Application Centric Infrastructure (ACI), απέκτησε την κορυφαία στον κλάδο λύση SDN της εταιρείας. [airbus]

Σύμφωνα με την Cisco το Cisco DNA Center περιλαμβάνει ενσωματώσεις ServiceNow out-of-the-box. Μπορούμε να χρησιμοποιήσουμε τα API της Cisco για να ανοίγουμε και να κλείνουμε αυτόματα εισιτήρια τεχνικής υποστήριξης όταν προκύπτουν φυσικά και λογικά προβλήματα στο δίκτυο του campus . [cisco]

Η προκύπτουσα αλλαγή παραδείγματος βοηθάει να διαχειριστεί το δίκτυο, να ενισχυθεί η ασφάλεια, να αντιμετωπιστεί ο συνεχώς αυξανόμενος φόρτος εργασίας και να ενισχυθεί η συνδεσιμότητα IoT και BYOD».

Το πρόσφατα υλοποιημένο δίκτυο SDN συνέδεσε πολλά κτίρια και 2.000 χρήστες. Το Airbus Helicopter αντικαθιστά περίπου το 20% της υπάρχουσας υποδομής του, περίπου 43 switch. [10,23]

- Cloudseeds

Η CloudSeeds είναι πάροχος υπηρεσιών cloud engineering και συμβουλευτικών υπηρεσιών από το 2013 στο Αμβούργο της Γερμανίας. Βοηθά τους επιχειρηματικούς πελάτες της να κλιμακώσουν τα δίκτυά τους για να επιτύχουν στόχους μέσω εικονικοποίησης υποδομής και υπηρεσιών πληροφορικής. Για να προσφέρει μια νέα κατηγορία αυτοματοποιημένων υπηρεσιών, το CloudSeeds χρειαζόταν μια νέα πλατφόρμα δικτύου για να καλύψει τη ζήτηση. Μέσω μιας ανοιχτής αρχιτεκτονικής, μπορεί να προβλέψει τις μελλοντικές ανάγκες ανάπτυξης. Έτσι επέτρεψε να

δημιουργηθεί ένα δίκτυο καθορισμένο από λογισμικό που ενσωματώνει απρόσκοπτα διαφορετικές τοποθεσίες σε ένα cloud.

Όταν ο εξοπλισμός είναι φυσικά στη θέση του και η επικάλυψη λογισμικού είναι έτοιμη για χρήση, χρειάζονται μόνο λίγα λεπτά για να εμφανιστούν νέοι διακομιστές ή ακόμη και να αναπτυχθεί ένα εντελώς νέο κέντρο δεδομένων για πελάτες σε λίγες ώρες. Το αποτέλεσμα ήταν η παροχή μηδενικού αγγίγματος του δικτύου που επιτρέπει την παροχή υπηρεσιών σε λίγα λεπτά αντί για εβδομάδες. Η νέα υποδομή SDN είναι ανθεκτική και οικονομικά αποδοτική.

- Ensono

Η Ensono είναι πάροχος διαχειριζόμενων υπηρεσιών πληροφορικής για μετασχηματισμό cloud. Συνεργάζεται με εταιρείες από όλο τον κόσμο, βοηθώντας τις να υιοθετήσουν λύσεις δικτύου που επιτρέπουν την ανάπτυξη και την αποτελεσματικότητα. Η Ensono προσπαθούσε να επεκτείνει το ιδιωτικό της cloud για να καλύψει την αυξανόμενη ζήτηση για υπηρεσίες δικτύωσης και ασφάλειας με ελέγχους και απαιτήσεις συμμόρφωσης. Χρειαζόταν επίσης έναν τρόπο να υποστηρίξει την ταχεία ανάπτυξή της, διατηρώντας παράλληλα μια ελάχιστη αύξηση στον αριθμό του προσωπικού. Υλοποιώντας πολλαπλές προσφορές από το VMware, η Ensono μπόρεσε να βελτιώσει την υποδομή του δικτύου της για να καλύψει τις ανάγκες και τις μελλοντικές προσδοκίες της. Με το NSX, παρέχει πρόσβαση αυτοεξυπηρέτησης και API στους πελάτες μας, προσθέτοντας παράλληλα βελτιωμένα χαρακτηριστικά ασφαλείας, όπως το Distributed Firewall και το Micro-Segmentation. Με τον αυτοματισμό μπροστά και στο κέντρο του έργου, όχι μόνο γίνονται ευκολότερες τις λειτουργίες αλλά παρέχουν επίσης μια πλατφόρμα αυτοματοποιημένης ανάκτησης καταστροφών (DR) στο Ensono Private Cloud, με εύκολη δοκιμή SDN.

Η Ensono αναφέρει ότι κατάφερε να εξοικονομήσει μεταξύ 10 και 15 εκατομμυρίων δολαρίων κατά τη διάρκεια πέντε ετών από την εφαρμογή των λύσεων SDN της VMware. Έχει βελτιώσει τον έλεγχο του περιβάλλοντος πληροφορικής του, αυτοματοποιώντας την ανάπτυξη εικονικών μηχανών και επιπέδων ασφαλείας μέσα σε λίγα λεπτά.



### 3.6 Πλεονεκτήματα - Μειονεκτήματα SDN

Σε ότι μελετήθηκε μέχρι αυτό το σημείο μπορούμε να δούμε αρκετά πλεονεκτήματα της χρήσης SDN τα οποία θα αναλυθούν παρακάτω καθώς όμως και να συμπεράνουμε και κάποια μειονεκτήματα τα οποία παρότι εκ πρώτης όψεως φαίνονται να λάμπουν δια της απουσίας τους υπάρχουν και είναι εδώ να μας θυμίζουν ότι κάθε αρχιτεκτονική συστήματος πάντα θα έχει κάποια μειονεκτήματα όσα και να προσφέρει.

#### 3.6.1 Πλεονεκτήματα SDN

- Κόστος:

Εφόσον η διαχείριση του δικτύου είναι συγκεντρωτική και αυτοματοποιημένη, συνολικά οδηγεί σε εξοικονόμηση κόστους στο SDN. Οι χρήστες μειώνουν το περιττό κόστος χρησιμοποιώντας το SDN με καλύτερη χρήση διακομιστή και βελτιωμένη εικονικοποίηση. Εκτός από αυτά, το SDN μειώνει τις λειτουργίες δικτύου επιτρέποντας την εκτέλεση πολλαπλών εργασιών. Έτσι, οι απαιτήσεις για ακριβό υλικό, συντήρηση και συχνές αναβαθμίσεις εξαλείφονται.

- Ασφάλεια:

Το SDN περιλαμβάνει έναν ελεγκτή που παρέχει ασφάλεια σε ολόκληρο το δίκτυο. Αυτός ο ελεγκτής διασφαλίζει ότι εφαρμόζονται σωστές πολιτικές ασφαλείας και πληροφορίες εντός του δικτύου. Επίσης, το SDN είναι εξοπλισμένο με ενιαίο σύστημα διαχείρισης. Μια ενιαία οντότητα θα ελέγχει την ασφάλεια και τις δυνατότητες. Αυτός ο τύπος προσέγγισης είναι γνωστό ότι είναι εξαιρετικά ασφαλής επειδή χρησιμοποιείται μόνο ένα κεντρικό σημείο. Ο διαχειριστής θα μπορούσε εύκολα να αποκλείσει απειλές ασφαλείας και να αποτρέψει τη μόλυνση.

- Συγκεντρωτισμός:

Το SDN επιτρέπει την κεντρική διαχείριση ολόκληρου του δικτύου. Όλες οι συσκευές δικτύου μπορούν να παρακολουθούνται και να διαχειρίζονται από μια κεντρική τοποθεσία. Εξαλείφει το εμπόδιο που δημιουργούν τα παραδοσιακά συστήματα στη διαχείριση της υποδομής. Ακόμα κι αν υπάρχει ανάγκη να διαχειρίζεστε το σύστημα μεμονωμένα, το SDN το καθιστά δυνατό.

- Επεκτασιμότητα:

Ένα άλλο πλεονέκτημα του SDN είναι το επίπεδο επεκτασιμότητας που προσφέρει. Η υποδομή του δικτύου μπορεί να αλλάξει άμεσα. Όλα αυτά χωρίς την ανάγκη αγοράς και διαμόρφωσης πόρων.

- Βελτιστοποίηση:  
Ένας νέος τρόπος βελτιστοποίησης συσκευών υλικού αναπτύσσεται κατά τη χρήση του SDN. Όλο το υπάρχον και το νέο υλικό μπορεί να εκχωρηθεί με συγκεκριμένο σκοπό με τη χρήση ελεγκτών SDN. Επομένως, εξαλείφει τον περιορισμό των συσκευών υλικού όπου είναι αφιερωμένο σε έναν και μόνο σκοπό.

### 3.6.2 Μειονεκτήματα SDN

- Μη δοκιμασμένη λύση μακροπρόθεσμα:  
Εξαιτίας αυτής των λιγοστών ετών που χρησιμοποιείται, πολλοί πιστεύουν ότι οι υλοποιήσεις SDN δεν είναι έτοιμες για να πάρουν και να διατηρήσουν για χρόνια τα σκήπτρα στα δίκτυα.
- Η ενσωμάτωση-SDN:  
Προσφέρει δυνατότητα ενσωμάτωσης Southbound με οποιοδήποτε υλικό προμηθευτή και ενσωμάτωση Northbound με οποιοδήποτε λύσεις παρακολούθησης, ενορχήστρωσης και cloud, αλλά η ενσωμάτωση είναι προκλητική και απαιτεί δεξιότητες και γνώσεις σχετικά με τις λύσεις διαφόρων προμηθευτών και τον τρόπο ενσωμάτωσης με το υπάρχον περιβάλλον.
- Διαφορετικές ερμηνείες:  
Τα SDN έχουν ερμηνευτεί διαφορετικά από κάθε προμηθευτή, επομένως χρειάζεται να εξεταστεί προσεκτικά ο μελλοντικός χάρτης SDN και να τελειοποιηθούν οι αρχιτεκτονικές δομές τους πριν οριστικοποιηθεί οποιαδήποτε τελική λύση.
- Ασφάλεια:  
Το SDN αφαιρεί τα φυσικά επίπεδα από τη σχεδίαση δικτύου και τα εικονικοποιεί, στη συνέχεια τα αντικαθιστά με ένα εκτεθειμένο στρώμα επιπέδου δικτύου, ανοιχτό για επίθεση.
- Προκλήσεις διαχείρισης:  
Βελτιστοποίηση διαχείρισης στον καθορισμό του εύρους ελέγχου που δίνεται στον ελεγκτή SDN.
- Υλική μετάβαση σε δίκτυο SDN:  
Μια εταιρεία ή ένας οργανισμός που επιθυμεί να χρησιμοποιήσει δίκτυα τύπου SDN και να μεταβεί σε αυτά από την παραδοσιακή δικτύωση θα χρειαστεί να ξαναστήσει το δίκτυο του και να το προσαρμόσει δομική στην αρχιτεκτονική SDN άρα θα υπάρξουν κάποιες ανεπιθύμητες δαπάνες πάνω σε αυτό.
- Μετάβαση ανθρωπίνου δυναμικού:

Όσα ισχύουν για τη δομή του δικτύου ισχύουν και για το ανθρώπινο δυναμικό του καθώς πρέπει να εκπαιδευτεί και να προσαρμοστεί και αυτό στην συντήρηση-διαχείριση δικτύων SDN

## 4. ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ SDN

Μέχρι τώρα είδαμε μια απεικόνιση του SDN με κυριότερο κομμάτι τα επίπεδα λειτουργίας του, πώς αυτή η αρχιτεκτονική χρησιμοποιείται, την κίνηση στην αγορά καθώς και τη διαθεσιμότητα και το τι μπορεί να προσφέρει ο κάθε πάροχος από πλευράς υπηρεσιών. Θα αναλυθεί το πώς λειτουργούν τα τρία επίπεδα σε ένα δίκτυο SDN και τι ακριβώς περιέχεται στο καθένα καθώς και η γενικότερη αλληλεπίδραση τους με το δίκτυο.

### 4.1 Ελεγκτής (Controller)

Παραδοσιακά, ο όρος επίπεδο ελέγχου αναφέρεται στη λογική που είναι υπεύθυνη για να αποφασίσει πώς μια συσκευή δικτύου θα προωθήσει πακέτα. Η προώθηση πραγματοποιείται με μικρορύθμιση των πινάκων προώθησης στο επίπεδο δεδομένων της συσκευής. Στη δικτύωση που ορίζεται από λογισμικό, το επίπεδο ελέγχου είναι επίσης ένα κεντρικό σημείο - το λεγόμενο λειτουργικό σύστημα δικτύου (NOS) από το οποίο η διαχείριση του δικτύου γίνεται σε παγκόσμιο επίπεδο. Το δίκτυο νοείται ως μια συλλογή από διάφορες φυσικές ή/και εικονικές συσκευές που ελέγχονται χρησιμοποιώντας διαφορετικά πρωτόκολλα επικοινωνίας. Στο επίπεδο ελέγχου περιέχεται ο ελεγκτής ο οποίος λειτουργεί ως ο πυρήνας του SDN.

Ένας ελεγκτής δικτύωσης που καθορίζεται από λογισμικό είναι ένα κεντρικό στοιχείο της αρχιτεκτονικής SDN. Παρέχει έλεγχο στα στοιχεία δικτύου (στον διαχειριζόμενο τομέα). Στη δικτύωση, υπάρχει το επίπεδο διαχείρισης, το επίπεδο ελέγχου και το επίπεδο δεδομένων. Ένας ελεγκτής SDN παρέχει λειτουργίες επιπέδου διαχείρισης και επιπέδου ελέγχου για τα στοιχεία δικτύου στον τομέα που διαχειρίζεται. Αυτό σημαίνει ότι ένας ελεγκτής SDN, με βάση τις πληροφορίες δικτύου και ένα σύνολο προκαθορισμένων κανόνων και πολιτικών, διαχειρίζεται τα στοιχεία δικτύου και διαμορφώνει («προγράμματα») το επίπεδο δεδομένων (δηλαδή κατευθύνει τη ροή δεδομένων μέσω του δικτύου). Ένα από τα βασικά πλεονεκτήματα της χρήσης ενός ελεγκτή SDN είναι ότι επιτρέπει πιο αποτελεσματική διαχείριση δικτύου και οι αλλαγές στη διαμόρφωση του δικτύου μπορούν να εφαρμοστούν από μια κεντρική τοποθεσία αντί να χρειάζεται να ρυθμίσετε με μη αυτόματο τρόπο κάθε μεμονωμένο στοιχείο δικτύου. Επιπλέον, ένας ελεγκτής SDN μπορεί να αυτοματοποιήσει ορισμένες εργασίες, όπως η διαχείριση της κυκλοφορίας και η ασφάλεια, που μπορούν να συμβάλουν στη μείωση του κινδύνου ανθρώπινου λάθους και στη βελτίωση της συνολικής αξιοπιστίας του δικτύου. Οι ελεγκτές SDN παρέχουν ένα API γνωστό και ως διεπαφή βορρά, μέσω του οποίου εξωτερικές

εφαρμογές ή συστήματα όπως οι πλατφόρμες ενορχήστρωσης μπορούν να αλληλεπιδράσουν με το δίκτυο. Σε τέτοιες περιπτώσεις, ένας ελεγκτής SDN μεταφράζει τις απαιτήσεις επιπέδου της εφαρμογής (π.χ. περιγραφή υψηλού επιπέδου της διαμόρφωσης δικτύου) σε μια διαμόρφωση συγκεκριμένη για τα υποστηριζόμενα στοιχεία δικτύου. Οι ελεγκτές SDN μπορούν να διαχειρίζονται τόσο φυσικές συσκευές δικτύου όσο και στοιχεία λογισμικού που εκτελούν λειτουργίες δικτύου.

## 4.2 Λειτουργίες Ελεγκτών

Συνοπτικά οι κύριες λειτουργίες ενός ελεγκτή SDN περιλαμβάνουν:

- Διαχείριση της ροής δεδομένων στο διαχειριζόμενο δίκτυο
- Παροχή API για εφαρμογές και άλλα στοιχεία (π.χ. πλατφόρμες ενορχήστρωσης) για αλληλεπίδραση με το δίκτυο
- Παροχή ορατότητας στο δίκτυο, επιτρέποντας την παρακολούθηση της απόδοσης του δικτύου και τον εντοπισμό προβλημάτων
- Αυτοματοποίηση εργασιών διαχείρισης δικτύου, όπως παροχή νέων στοιχείων δικτύου και επαναδιαμόρφωση διαδρομών δικτύου

Αναλυτικότερα θα μπορούσαμε να πούμε ότι ο ελεγκτής παρέχει τις εξής δυνατότητες

- Υποστήριξη Southbound:  
Έχει οριστεί ως ο τρόπος χειρισμού ενός ελεγκτή προς τις συσκευές δικτύου για την επίτευξη βελτιστοποιημένης ροής κυκλοφορίας. Υπάρχουν πολλά πρωτόκολλα νότια κατεύθυνσης που μπορούν να χρησιμοποιηθούν. Βάση της λειτουργικότητας του κάθε πρωτοκόλλου θα πρέπει να μπορεί να υποστηρίζει αντιστοίχιση πεδίου, ανακάλυψη δικτύου με διαφορετικά πρωτόκολλα κ.τ.λ. Σε περίπτωση υποστήριξης προς τη νότια διεπαφή, οι υπεύθυνοι υλοποίησης πρέπει να λάβουν υπόψη λαμβάνοντας υπόψη όχι μόνο τα χαρακτηριστικά του πρωτοκόλλου αλλά και πιθανές επεκτάσεις, νεότερες εκδόσεις κ.α
- Υποστήριξη Northbound:

Τα Northbound API χρησιμοποιούνται για την εφαρμογή ενσωμάτωσης δικτύου και προγραμματισμού και μπορεί να χρησιμοποιηθεί από συστήματα ενορχήστρωσης που απευθύνονται σε πελάτες και εφαρμογές τρίτων. Είναι ζωτικής σημασίας για να διασφαλιστεί ότι ένας ελεγκτής έχει αναπτυχθεί κατάλληλα για την ενορχήστρωση των

επικοινωνιών μεταξύ επιπέδων. Για παράδειγμα, ο ελεγκτής θα πρέπει να μπορεί να υποστηρίζει συστήματα ενορχήστρωσης για εφαρμογές π.χ cloud όχι μόνο των ελεύθερων ελεγκτών και πρωτοκόλλων αλλά και αυτών που παρέχουν οι διάφοροι πάροχοι. Αυτές οι εφαρμογές θα μπορούσαν επίσης περιλαμβάνουν μηχανική κυκλοφορίας ή εφαρμογές που συλλέγουν δεδομένα που χρησιμοποιούνται για την εκτέλεση εργασιών όπως διαχείριση του δικτύου.

- Προγραμματιστικότητα:

Ο προγραμματισμός είναι ένα από τα πιο πολύτιμα χαρακτηριστικά του SDN και συγκεκριμένα στους ελεγκτές. Παραδοσιακά, η διαμόρφωση δικτύου επιτυγχάνεται με την εφαρμογή των αντίστοιχων κανόνων ανά συσκευή. Αναπόφευκτα, αυτή η στατική προσέγγιση είναι χρονοβόρα, επιρρεπής σε σφάλματα και σε ορισμένες περιπτώσεις ασυνεπής. Μπορεί επίσης να οδηγήσει σε υποβάθμιση απόδοσης δικτύου. Οι διεπαφές μέσω προγραμματισμού είναι τα βασικά στοιχεία κάθε ελεγκτή. Ένα από τα πιο κοινά παραδείγματα προγραμματισμού είναι η ανακατεύθυνση κίνησης που μπορεί να ζητηθεί για διάφορους σκοπούς (δηλαδή κατανομή κυκλοφορίας από άποψη χρόνου, τόπου, ασφάλεια). Ως πρώτο βήμα, το API με κατεύθυνση βόρεια κινεί τις πληροφορίες ελέγχου που έχουν συγκεντρωθεί στον ελεγκτή που είναι διαθέσιμος σε εφαρμογές δικτύου. Τότε μπορεί να αλλάξει το δίκτυο ρυθμό για να εκτελέσει εργασίες όπως η προώθηση πακέτων μέσω της διαδρομής μικρότερου κόστους ή αλλαγή των ρυθμίσεων QoS με βάση το διαθέσιμο εύρος ζώνης ή άλλους παράγοντες. Ένα άλλο παράδειγμα προγραμματισμού σε έναν ελεγκτή SDN είναι η δυνατότητα για να εφαρμόζει φίλτρα σε πακέτα για να καθορίζει εάν θα τα απορρίψει ή θα τα περάσει. Αυτά τα δυναμικά φίλτρα μπορεί να βασίζονται στην αντιστοίχιση κεφαλίδων πακέτου και μπορεί να είναι απλά ή περισσότερα εξελιγμένα, που αποτελούνται από σύνθετους συνδυασμούς πολλαπλών κεφαλίδων πακέτων. Τα φίλτρα θα πρέπει να μπορούν να αναπτύσσονται δυναμικά και είναι ο ρόλος του ελεγκτή SDN είναι να μεταωγήσει τις σχετικές καταχωρήσεις του πίνακα ροής προς τα κάτω στα switch.

- Επίβλεψη-παρακολούθηση:

Η παρακολούθηση δικτύου είναι μια άλλη δυνατότητα του ελεγκτή. Μέσω πρωτοκόλλων και σχετικών εργαλείων, ο ελεγκτής μπορεί να εντοπίσει προβλήματα στο δίκτυο και να διευκολύνει τη διαδικασία αντιμετώπισης προβλημάτων. Τα πλεονεκτήματα του ελεγκτή περιλαμβάνουν λεπτομερή ροή παρακολούθησης (και όχι τυχαία δειγματοληψία),

παρακολούθηση συγκεκριμένων κατηγοριών κυκλοφορίας κ.λπ. Ο ελεγκτής θα πρέπει να υποστηρίζει τυπικά πρωτόκολλα και τεχνικές παρακολούθησης, έτσι ώστε οι πληροφορίες μπορούν να ενσωματωθούν με άλλα συστήματα διαχείρισης και ενορχήστρωσης. Θα πρέπει, για παράδειγμα, να είναι δυνατή η παρακολούθηση της υγείας του ελεγκτή και του εικονικού

δικτύου που υποστηρίζει ο ελεγκτής χρησιμοποιώντας τέτοιου είδους εφαρμογές (SNNP) .

- Network Virtualization:

Το Network Virtualization είναι η δυνατότητα δημιουργίας λογικών, εικονικών δικτύων που είναι αποσυνδεδεμένα από το υποκείμενο υλικό. Η εικονικοποίηση δικτύου αναφέρεται και στις δύο κατηγορίες του μοντέλου OSI. Υπηρεσίες επιπέδων 2-3 (δηλαδή δρομολόγηση) και υπηρεσίες επιπέδων 4-7, όπως εξισορρόπηση φορτίου. Συνηθισμένα παραδείγματα εικονικοποίησης δικτύου που παράγονται εδώ και δεκαετίες είναι το εικονικό LAN (VLAN) και το Virtual Routing and Forwarding (VRF). Λόγω της ταχείας ανάπτυξης και αλλαγής όσον αφορά τον όγκο του δικτύου, τις απαιτήσεις απόδοσης και άλλα, οι παραπάνω μέθοδοι θεωρούνται περιορισμένες τόσο σε εύρος όσο και σε αξία. Οι ελεγκτές SDN διευκολύνουν την υλοποίηση της εικονικοποίησης δικτύου με τρόπο από άκρο σε άκρο και έτσι δίνεται οι δυνατότητα σε οργανισμούς να δημιουργούν δυναμικά εικονικά δίκτυα και να ανταποκρίνονται σε απαιτητικές διεργασίες.

Το μοντέλο Open Systems Interconnection (OSI) περιγράφει επτά επίπεδα που χρησιμοποιούν τα συστήματα υπολογιστών για να επικοινωνούν μέσω ενός δικτύου. Ήταν το πρώτο πρότυπο μοντέλο για δικτυακές επικοινωνίες, που υιοθετήθηκε από όλες τις μεγάλες εταιρείες υπολογιστών και τηλεπικοινωνιών στις αρχές της δεκαετίας του 1980. Το σύγχρονο Διαδίκτυο δεν βασίζεται στο OSI, αλλά στο απλούστερο μοντέλο TCP/IP. Ωστόσο, το μοντέλο 7 επιπέδων OSI εξακολουθεί να χρησιμοποιείται ευρέως, καθώς βοηθά στην οπτικοποίηση και την επικοινωνία του τρόπου λειτουργίας των δικτύων και βοηθά στην απομόνωση και την αντιμετώπιση προβλημάτων δικτύωσης. Το OSI εισήχθη το 1983 από εκπροσώπους των μεγάλων εταιρειών υπολογιστών και τηλεπικοινωνιών και υιοθετήθηκε από το ISO ως διεθνές πρότυπο το 1984.

ΕΠΙΠΕΔΟ	ΛΕΙΤΟΥΡΓΙΑ
Εφαρμογών	Παρέχεται στις εφαρμογές πρόσβαση στο δίκτυο
Παρουσίασης	Αναπαράσταση δεδομένων και κρυπτογράφηση
Συνόδου	Έλεγχος του διαλόγου μεταξύ των άκρων της επικοινωνίας
Μεταφοράς	Αξιόπιστη επικοινωνία από άκρο σε άκρο
Δικτύου	Καθορισμός διαδρομών και λογικών διευθύνσεων των κόμβων στα πλαίσια ενός διαδικτύου
Ζεύξης Δεδομένων	Φυσική διευθυνσιοδότηση (MAC & LLC)
Φυσικό	Δυαδική μετάδοση σήματος μέσω του φυσικού μέσου

Πίνακας 4.1 Σύνοψη περιγραφή του μοντέλου OSI

- Ευελιξία:

Η ευελιξία είναι μια άλλη απαίτηση που πρέπει να επιτύχει ένας ελεγκτής. Από τη μια πλευρά, πρέπει να υποστηρίζεται ποικιλία εφαρμογών ενώ από την άλλη, εφαρμογές ελεγκτών θα πρέπει να χρησιμοποιούν ένα κοινό πλαίσιο και μοντέλο προγραμματισμού για να διασφαλίσουν ότι τα εκτεθειμένα API είναι συνεπή και αξιοποιούνται εύκολα. Αυτό είναι σημαντικό για αρκετούς σκοπούς όπως η αντιμετώπιση προβλημάτων, η ενοποίηση συστήματος κ.α.

- Τοπολογία:

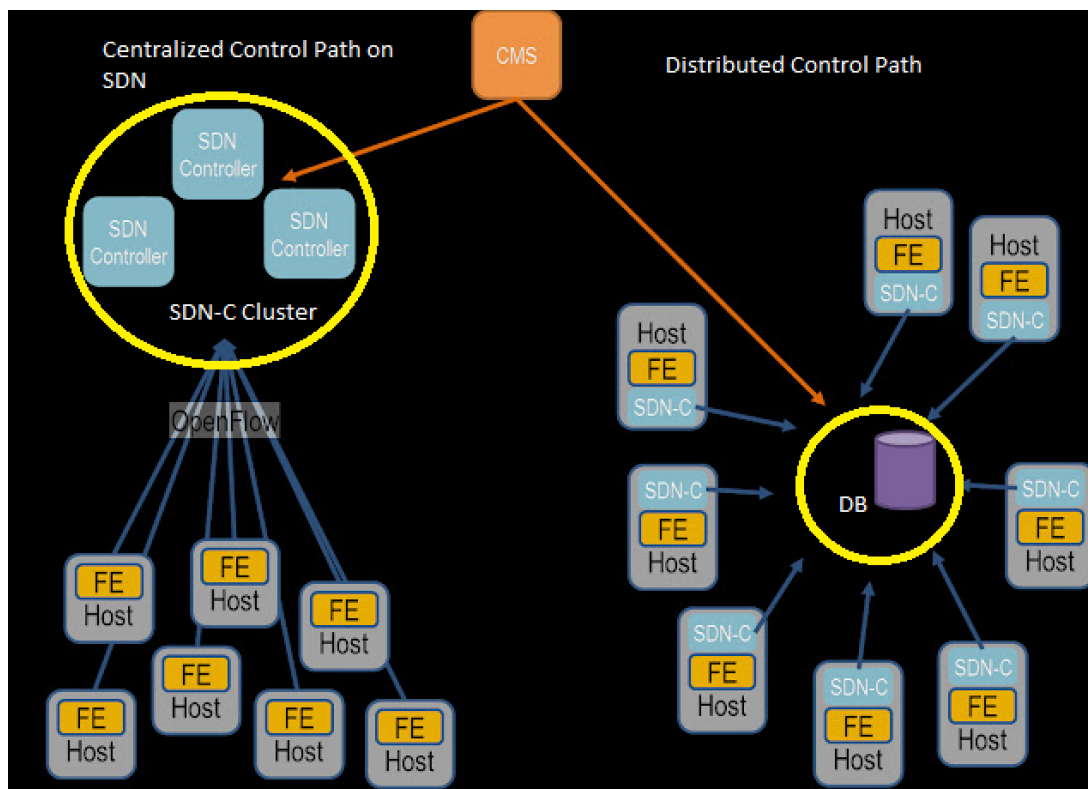
Οι ελεγκτές SDN θα πρέπει επίσης να αξιολογούνται ως προς την προσαρμοστικότητά τους όσον αφορά το δίκτυο και την τοπολογία του. Η τοπολογία δικτύου είναι η διάταξη των στοιχείων ενός δικτύου υπολογιστών. Σε περίπτωση μεγάλων δικτύων, θα πρέπει να αναλυθούν επιλογές σχετικά με την τοπολογία των επιπλέον ελεγκτών. Εξάλλου, η επικοινωνία μεταξύ των ελεγκτών είναι επίσης μέρος της τοπολογίας. Επί του παρόντος, υπάρχουν ορισμένες συζητήσεις στον κλάδο γύρω από το πώς οι ελεγκτές θα μιλήσουν μεταξύ τους. Μια κοινή τεχνική χρησιμοποιεί BGP για ανταλλαγή πληροφοριών μεταξύ των ελεγκτών. Τέλος, θέματα σχετικά με το επίπεδο συγκέντρωσης θεωρούνται αρχιτεκτονικά.

- Centralization:

Μέχρι αυτό το σημείο, το centralization έχει αναγνωριστεί ως ένα από τα κύρια χαρακτηριστικά του SDN και ειδικότερα του ελεγκτή. Ωστόσο, στην πράξη υπάρχουν περιπτώσεις όπου η μπορεί να διανεμηθεί το επίπεδο ελέγχου. Στην περίπτωση κεντρικών ελεγκτών, ένας ή περισσότεροι ελεγκτές διαχειρίζονται όλα τα στοιχεία



δικτύου στο σύστημα και διατηρούν μια συνολική εικόνα ολόκληρου του δικτύου [16]. Οι περισσότεροι ελεγκτές SDN σήμερα λειτουργούν με αυτόν τον τρόπο. Είναι μια αρκετά απλή λύση που διασφαλίζει ότι ο ελεγκτής έχει μια συνολική και ρεαλιστική άποψη της τοπολογίας. Η ενοποίηση και η συνεργασία με παλαιού τύπου συστήματα είναι επίσης ευκολότερη σε αυτήν την περίπτωση. Ωστόσο, υπάρχει ένα πρόβλημα υψηλής εξάρτησης από τη διαθεσιμότητα συμπλέγματος και όλες οι υπηρεσίες διαχειρίζονται κεντρικά. Σε περίπτωση κατανεμημένων ελεγκτών, ένας τοπικός ελεγκτής εκτελείται σε κάθε κόμβο υπολογιστών και διαχειρίζεται το απευθείας τα στοιχεία δικτύου. Η καθυστέρηση και η επεκτασιμότητα είναι καλύτερες σε αυτή την περίπτωση. Από την άλλη, δεν επιτυγχάνεται σφαιρική άποψη του δικτύου και συγχρονισμός του συνόλου και η τοπολογία γίνεται πολύπλοκη καθώς αυξάνεται ο αριθμός των υπολογιστικών κόμβων.



Εικόνα 4.1 Σύγκριση συγκεντρωτικής και κατανεμημένης αρχιτεκτονικής.

#### 4.2.1 Επιλογή και τοποθέτηση ελεγκτών

Στις περισσότερες περιπτώσεις τα δίκτυα SDN αποτελούνται από πολλούς ελεγκτές που πρέπει να συνυπάρχουν κατάλληλα τοποθετημένοι για την επίτευξη ενός βέλτιστου και

απλού κεντρικού δικτύου. Σε αυτό το σημείο, ζητήματα όπως η επεκτασιμότητα, η αξιοπιστία, η απόδοση και η διαθεσιμότητα του δικτύου εμφανίζονται. Η συνολική αρχιτεκτονική των ελεγκτών επηρεάζει κάθε πτυχή ενός αποσυνδεδεμένου επίπεδου ελέγχου, από την εξισορρόπηση φορτίου έως τις μετρήσεις ανοχής σφαλμάτων και απόδοσης. Σε ορισμένες περιπτώσεις μπορεί να επηρεαστούν οι τύποι δικτύων, η διαθεσιμότητα, η απόκριση και ο χρόνος σύγκλισης. Θα πρέπει επίσης να λάβουμε υπόψη το γεγονός ότι απαιτήσεις σχετικά με τα παραπάνω ποικίλλουν ανάλογα με τον τύπο του δικτύου. Για παράδειγμα, σε ένα δίκτυο WAN απαιτείται η ελαχιστοποίηση των καθυστερήσεων μετάδοσης, ενώ σε ένα κέντρο δεδομένων ή σε μια επιχείρηση, απαιτείται μεγιστοποίηση της ανοχής σφαλμάτων. Άρα πρέπει να γίνεται έρευνα και αναλυτική μελέτη για το ποιος είναι ο βέλτιστος τρόπος τοποθέτησης αρχιτεκτονικά και ο αριθμός ελεγκτών που αφορούν έναν συγκεκριμένο τύπο δικτύου για να επιτύχουν το επιθυμητό επίπεδο απόδοσης και ελαχιστοποίησης αποτυχιών. Η τοποθέτηση ελεγκτών σύμφωνα με μελέτες εξαρτιέται από τα επιθυμητά όρια, τις μετρικές επιλογές και την ίδια την τοπολογία του δικτύου. Για τον καθορισμό του επιθυμητού συνδυασμού αξιοπιστίας και καθυστέρησης, προτείνεται να χρησιμοποιείται ένας ελεγκτής που αποδίδει τη βέλτιστη μέτρηση αξιοπιστίας, ενώ βελτιστοποιεί τον μέσο όρο καθυστέρησης στο δίκτυο. Από την άλλη πλευρά, η βελτιστοποίηση της αξιοπιστίας αυξάνει τον μέσο όρο και σενάριο καθυστέρησης. [5]

Δεδομένου ότι το SDN είναι ένα λογικά συγκεντρωμένο δίκτυο, η αξιοπιστία του επιπέδου ελέγχου είναι κρίσιμης σημασίας. Ως εκ τούτου, μηχανισμοί για την αποφυγή ενός μόνο σημείου αστοχίας θα πρέπει να υπάρχουν. Ο πιο θεμελιώδης μηχανισμός ανάκτησης είναι η προσέγγιση «πρωτεύουσας εφεδρικής αντιγραφής». Σε αυτό το μοντέλο, οι ελεγκτές αντιγράφων ασφαλείας αναλαμβάνουν τον έλεγχο το δικτύου σε περίπτωση αποτυχίας του πρωτεύοντος δικτύου. Υπάρχουν, ωστόσο, αρκετά ζητήματα όπως ο συντονισμός εφεδρικού ελεγκτή κατά τη διάρκεια αποτυχίας, που πρέπει να μελετηθούν περαιτέρω. Υψηλή διαθεσιμότητα απαιτείται επίσης στους περισσότερους τύπους δικτύων. Αυτό μπορεί να επιτευχθεί μέσω βελτιωμένων API προς τη νότια διεπαφή και τη βέλτιστη τοποθέτηση ελεγκτή. Ευρήματα μέχρι σήμερα υποδεικνύουν ότι μεγιστοποιείται η δυνατότητα σύνδεσης στοιχείων δικτύου σε πολλούς ελεγκτές ως προς τη διαθεσιμότητα. Έχει επίσης αποδειχθεί ότι ο αριθμός των απαιτούμενων ελεγκτών εξαρτάται από την τοπολογία παρά από το μέγεθος του δικτύου. Σε ορισμένες περιπτώσεις, προτείνεται οι ελεγκτές να εκχωρούν συναρτήσεις ελέγχου, οι οποίες να χρησιμοποιούνται για την αναφορά κατάστασης και

αλλαγών σε τιμές χαρακτηριστικών, ειδοποιήσεις υπέρβασης ορίου, αστοχίες υλικού κ.λπ. Με αυτόν τον τρόπο η λειτουργική αποτελεσματικότητα αυξάνεται. [9]

#### 4.2.2 Αρχιτεκτονικές τοποθέτησης ελεγκτών

- DIFANE

Το DIFANE είναι παράδειγμα ελεγκτή λειτουργεί σε προληπτική λειτουργία. Το DIFANE είναι μια αρχιτεκτονική διαχείρισης κατανεμημένης ροής που αποτελείται από έναν ελεγκτή που δημιουργεί τους κανόνες, τα κύρια switch που είναι προγραμματισμένα από τον ελεγκτή και τα απλά switch. Τα switch εξουσιοδότησης μπορούν να είναι ένα υποσύνολο υφιστάμενων μεταγωγέων στο δίκτυο ή αποκλειστικών μεταγωγέων. Το αρχικό switch χειρίζεται το πακέτο στο επίπεδο δεδομένων και στέλνει ανατροφοδότηση στα υπόλοιπα για να αποθηκεύσει τους σχετικούς κανόνες τοπικά. Με αυτόν τον τρόπο, όλα τα επίπεδα δεδομένων λειτουργιών που απαιτούνται βασίζονται σε συγκεκριμένους κανόνες. Τα απλά switch χρησιμοποιούν cache έτσι ώστε οι περισσότερες καταχωρήσεις κίνησης δεδομένων να επεξεργάζονται από αυτήν την κρυφή μνήμη. Τα switch εξουσιοδότησης αποθηκεύουν επίσης κανόνες που είναι εγκατεστημένο και ενημερωμένο από τον ελεγκτή. Τέλος, οι κανόνες κατάτμησης εγκαθίστανται από τον ελεγκτή σε κάθε switch.

- HyperFlow

Το Hyperflow είναι μια λογικά κατανεμημένη ανάπτυξη ελεγκτή. Αποτελείται από OpenFlow switches και ελεγκτές NOX, καθένας από τους οποίους εκτελεί την HyperFlow εφαρμογή. Το HyperFlow προσαρμόζει τη λήψη αποφάσεων σε μεμονωμένους ελεγκτές και ελαχιστοποιεί τον χρόνο απόκρισης του επιπέδου ελέγχου σε αιτήματα επιπέδου δεδομένων και παρέχει επεκτασιμότητα ενώ διατηρείται ο έλεγχος του δικτύου λογικά συγκεντρωμένος. Μέσα από σχήματα συγχρονισμού, όλοι οι ελεγκτές μοιράζονται την ίδια προβολή δικτύου και τοπικά εξυπηρετούν αιτήματα χωρίς ενεργή επαφή με κάποιον απομακρυσμένο κόμβο, ελαχιστοποιώντας έτσι τη ροή χρόνους εγκατάστασης.

- Kandoo

Η ανάπτυξη ιεραρχικού ελεγκτή του Kandoo αποτελείται από μια ιεραρχία δύο επιπέδων για ελεγκτές: τοπικοί ελεγκτές που εκτελούν τοπικές εφαρμογές και μια

κεντρική ρίζα ελεγκτή που εκτελεί μη τοπικές εφαρμογές ελέγχου (δηλαδή εφαρμογές που απαιτούν πρόσβαση σε όλο το δίκτυο). Κάθε switch ελέγχεται μόνο από έναν ελεγκτή Kandoo, και κάθε ελεγκτής Kandoo μπορεί να ελέγχει πολλά switch. Εάν χρειάζεται ο ριζικός ελεγκτής μπορεί να εγκαταστήσει καταχωρήσεις ροής σε switch ενός τοπικού ελεγκτή και αναθέτει τα αιτήματα στο αντίστοιχο τοπικό ελεγκτή.

- Onix

Το Onix είναι μια φυσικά κατανεμημένη ανάπτυξη ελεγκτή, η οποία εκτελείται σε ένα σύμπλεγμα ενός ή περισσότερων φυσικών διακομιστών. Η Onix είναι υπεύθυνη να δώσει στη λογική ελέγχου πρόσβαση στο δίκτυο μέσω προγραμματισμού. Χρησιμοποιώντας το Onix API, το οποίο προσφέρει προβολή του φυσικού δικτύου οι εφαρμογές ελέγχου μπορούν να διαβάζουν και να καταγράφουν κατάσταση σε οποιοδήποτε στοιχείο του δικτύου και έτσι η διατήρηση της κατάστασης να είναι συνεπής μεταξύ των στοιχείων εντός του δικτύου και της εφαρμογής ελέγχου που εκτελείται σε πολλούς διακομιστές Onix. Το αντίγραφο της κατάστασης δικτύου που παρακολουθείται από την Onix είναι αποθηκευμένο σε μια δομή δεδομένων που ονομάζεται Network Information Base (NIB), η οποία είναι ένα γράφημα όλων των οντοτήτων δικτύου σε μια τοπολογία δικτύου.

### 4.3 Είδη ελεγκτών

Από τα παραπάνω καταλάβαμε ότι δύο είναι οι βασικές κατηγορίες ελεγκτών και πρωτοκόλλων οι οποίες κατηγοριοποιούνται ως προς την πρόσβαση που μπορούμε να έχουμε σε αυτούς :

- Ελεύθερη Πρόσβαση
- Πρόσβαση μέσω παρόχου

Σε αυτό το υποκεφάλαιο θα δούμε τους ελεγκτές και στο αμέσως επόμενο θα μελετηθούν τα πρωτόκολλά τους.

#### 4.3.1 Ελεγκτές Ελεύθερης πρόσβασης

Έχει αναφερθεί η ύπαρξη τόσο ανοιχτού κώδικα όσο και εμπορικών λύσεων. Αυτό συμβαίνει επειδή στην περίπτωση της αρχιτεκτονικής SDN, ο ανοιχτός κώδικας εξελίσσεται με αξιοσημείωτο ρυθμό. Σήμερα το λογισμικό ανοιχτού κώδικα έχει γίνει το σημείο εξέλιξης αφού επιτρέπει τη συνεργασία των ανθρώπων σε όλο τον κόσμο με στόχο την επίλυση

προβλημάτων και τη δημιουργία πιο εξελιγμένων λύσεων. Τα πανεπιστήμια έχουν δημιουργήσει πολλά έργα ανοιχτού κώδικα που σχετίζονται με το SDN και το OpenFlow. Οι λύσεις ανοιχτού κώδικα δεν περιορίζονται στην ερευνητική περιοχή, αλλά έχουν επεκταθεί και σε αυτήν της αγοράς. Πολλές εταιρείες όπως η Adara, η Bigswitch και η NEC έχουν κυκλοφορήσει ελεύθερες εκδόσεις των προϊόντων τους. Το δίλημμα ανοιχτού κώδικα έναντι αποκλειστικής χρήσης κάνει νόημα σε περιπτώσεις που κάποιος σκέφτεται να επεκτείνει τον ελεγκτή ή έχει ειδικές τροποποιήσεις ειδικά για την επιχείρησή του. Οι λύσεις ανοιχτού κώδικα μειώνουν την πιθανότητα του να περιοριστεί ένα σύστημα από μια εταιρία. Αυτό όμως σταδιακά θα εξαφανιστεί αφού γίνονται πολλές προσπάθειες αποτροπής και εννοίας τυποποίησης στοιχείων αρχιτεκτονικής. Τελικά, η επιλογή μεταξύ εμπορικών και ελεγκτών ανοιχτού κώδικα αφορά την λειτουργικότητα και τον τρόπο με τον οποίο μπορούν να χρησιμοποιηθούν σε μια υπάρχουσα αρχιτεκτονική.

Σύμφωνα με τα στατιστικά χρήσης των ελεγκτών SDN φαίνεται ότι ο πιο δημοφιλής ελεγκτής είναι ο ODL , τον οποίο χρησιμοποιούνε πολλές ασιατικές εταιρείες τηλεπικοινωνιών και διαθέτει πλήθος διαφορετικών υλοποιήσεων (περισσότερες από εκατό) . Τα ποσοστά χρήσης του τα τελευταία χρόνια ξεπερνάνε το 60%. Αξίζει να σημειωθεί ότι τα στατιστικά στοιχεία αυτά καθώς και οι ελεγκτές που παρουσιάζονται παρακάτω έχουν να κάνουν με το έτος 2022.

#### 4.3.1.1 Ενεργοί ελεγκτές

Παρακάτω θα αναλυθούν οι ελεγκτές SDN οι οποίοι είναι ακόμα ενεργοί. Ως ενεργοί ορίζονται αυτοί οι ελεγκτές που έχουνε ακόμα υποστήριξη , λαμβάνουν ενημερώσεις και βελτιώσεις και κατά κύριο λόγο χρησιμοποιούνται από επιχειρήσεις και οργανισμούς. Ο παρακάτω πίνακας δείχνει υλοποιήσεις ελεγκτών ανοιχτού κώδικα οι οποίοι είναι ενεργοί.

<b>ΕΝΕΡΓΟΙ ΕΛΕΓΚΤΕΣ</b>
OpenDaylight
ONOS
Ryu
POX
Faucet
FRR
Trema
OpenContrail
Project Calico

Πίνακας 4.2 Ενεργοί ελεγκτές.

#### 4.3.1.1.1 Ελεγκτής OpenDaylight

Ο OpenDaylight είναι ένας δημοφιλής και ευρέως χρησιμοποιούμενος ελεγκτής SDN ανοιχτού κώδικα που παρέχει μια αρθρωτή και επεκτάσιμη πλατφόρμα για την ανάπτυξη εφαρμογών και υπηρεσιών SDN. Τα ολοκληρωμένα API του, η υποστήριξη για πολλαπλά πρωτόκολλα και η ενσωμάτωση με άλλα εργαλεία τον καθιστούν μια ευέλικτη και ισχυρή πλατφόρμα για αυτοματισμό και διαχείριση δικτύου. Έχει σχεδιαστεί για να υποστηρίζει μια σειρά από αρχιτεκτονικές δικτύου, από παραδοσιακά δίκτυα τριών επιπέδων έως πιο πολύπλοκα δίκτυα πολλαπλών επιπέδων. Παρέχει ένα ολοκληρωμένο σύνολο API που μπορούν να χρησιμοποιηθούν για τον προγραμματισμό του δικτύου και την αυτοματοποίηση εργασιών όπως η διαμόρφωση δικτύου, η κυκλοφορίας πακέτων και γενικότερα η ασφάλεια.

Ένα από τα βασικά πλεονεκτήματα του OpenDaylight είναι η αρθρωτή αρχιτεκτονική του, η οποία επιτρέπει την προσθήκη νέων χαρακτηριστικών και στοιχείων, όπως απαιτείται. Αυτό τον καθιστά εξαιρετικά προσαρμόσιμο σε διαφορετικές περιπτώσεις χρήσης και αρχιτεκτονικές δικτύου. Η πλατφόρμα περιλαμβάνει επίσης μια σειρά από λειτουργίες και εργαλεία για τη διαχείριση της κυκλοφορίας του δικτύου, συμπεριλαμβανομένης της υποστήριξης για μηχανική κυκλοφορίας, QoS και ασφάλεια.

Ένα άλλο πλεονέκτημα του OpenDaylight είναι το ολοκληρωμένο σύνολο REST-API, το οποίο επιτρέπει στους προγραμματιστές να προγραμματίζουν το δίκτυο και να αυτοματοποιούν εργασίες χρησιμοποιώντας μια ποικιλία γλωσσών προγραμματισμού και εργαλείων. Αυτό καθιστά εύκολη την ενσωμάτωση του OpenDaylight με άλλες πλατφόρμες και εργαλεία, όπως το OpenStack, το Docker και το Kubernetes, και επιτρέπει την ανάπτυξη προσαρμοσμένων εφαρμογών και υπηρεσιών που πληρούν συγκεκριμένες απαιτήσεις δικτύου.

Τα SDN OpenStar, Docker και Kubernetes είναι όλες τεχνολογίες που μπορούν να χρησιμοποιηθούν σε συνδυασμό με το SDN για τη βελτίωση της απόδοσης, της επεκτασιμότητας και της ευελιξίας του δικτύου.

- **SDN OpenStar:** Είναι ένας ελεγκτής SDN ανοιχτού κώδικα που παρέχει ένα κεντρικό σημείο ελέγχου για τη διαχείριση δικτύου. Επιτρέπει στους διαχειριστές δικτύου να διαμορφώνουν και να παρακολουθούν τις συσκευές του δικτύου τους μέσω μιας φιλικής προς το χρήστη διεπαφής που βασίζεται στον ιστό. Υποστηρίζει

επίσης το πρωτόκολλο OpenFlow, το οποίο επιτρέπει τη δυναμική και ευέλικτη διαμόρφωση δικτύου.

- **Docker:** Το Docker είναι μια πλατφόρμα κοντέινερ που επιτρέπει να φτιάχνουμε και να αναπτύσσουμε εφαρμογές λογισμικού με τη μορφή container. Τα container είναι ελαφριά, φορητά και μπορούν να λειτουργήσουν σε οποιαδήποτε πλατφόρμα υποστηρίζει Docker. Αυτό διευκολύνει την ανάπτυξη και τη διαχείριση εφαρμογών σε ένα κατανεμημένο περιβάλλον, όπως ένα δίκτυο SDN.
- **Kubernetes:** Είναι μια πλατφόρμα εντοπισμού και διαχείρισης κοντέινερ που αυτοματοποιεί την ανάπτυξη, την κλιμάκωση και τη διαχείριση εφαρμογών με container. Παρέχει ένα εξαιρετικά διαθέσιμο και επεκτάσιμο περιβάλλον για την εκτέλεση εφαρμογών με container και μπορεί να χρησιμοποιηθεί για τη διαχείριση δικτύων SDN μεγάλης κλίμακας. Το Kubernetes μπορεί επίσης να χρησιμοποιηθεί σε συνδυασμό με το Docker για τη διαχείριση container και την ανάπτυξη εφαρμογών σε πολλούς κεντρικούς υπολογιστές.

Μαζί, αυτές οι τεχνολογίες παρέχουν μια ισχυρή και ευέλικτη πλατφόρμα για τη διαχείριση δικτύων SDN, επιτρέποντας στους διαχειριστές να διαμορφώνουν, να παρακολουθούν και να διαχειρίζονται εύκολα την υποδομή του δικτύου τους.

Το OpenDaylight διαθέτει επίσης μια ενεργή και υποστηρικτική κοινότητα προγραμματιστών και χρηστών που συμβάλλουν στη συνεχή ανάπτυξή του και παρέχουν πόρους και υποστήριξη στην κοινότητα. Αυτό διασφαλίζει ότι η πλατφόρμα παραμένει ενημερωμένη και σχετική με τις ανάγκες των χρηστών της.

Ωστόσο, μια πιθανή αδυναμία του OpenDaylight είναι η πολυπλοκότητά του. Λόγω της αρθρωτής αρχιτεκτονικής και του ολοκληρωμένου συνόλου δυνατοτήτων, το OpenDaylight μπορεί να είναι δύσκολο να ρυθμιστεί και να διαμορφωθεί, ειδικά για χρήστες που είναι νέοι στο SDN. Επιπλέον, η υποστήριξή του για πολλαπλά πρωτόκολλα μπορεί να οδηγήσει σε προβλήματα δυσλειτουργικού τύπου και συνολικά αυξημένη πολυπλοκότητα.

Συνολικά, το OpenDaylight είναι ένας ισχυρός και ευέλικτος ελεγκτής SDN που είναι κατάλληλος για μια σειρά περιπτώσεων χρήσης και αρχιτεκτονικές δικτύου. Η αρθρωτή αρχιτεκτονική, τα ολοκληρωμένα API και η ενσωμάτωση με άλλα εργαλεία το καθιστούν δημοφιλή επιλογή για αυτοματισμό και διαχείριση δικτύου και η ενεργή κοινότητά του διασφαλίζει ότι παραμένει σχετικό και ενημερωμένο. Ωστόσο, η πολυπλοκότητά του μπορεί

να απαιτεί επιπλέον χρόνο και πόρους για τη ρύθμιση και τη διαμόρφωση. Αποτελεί όπως αναλύθηκε και στο προηγούμενο υποκεφάλαιο τον πιο διαδεδομένο ελεγκτή SDN.

Συμπερασματικά θα μπορούσαμε να πούμε ότι τα κύρια πλεονεκτήματά του είναι:

1. Αρθρωτή αρχιτεκτονική
2. Ολοκληρωμένα API
3. Υποστήριξη πολλαπλών πρωτοκόλλων
4. Ενσωμάτωση με άλλα εργαλεία
5. Ενεργή κοινότητα

#### **4.3.1.1.2 Ελεγκτής ONOS**

Ο ONOS (Open Network Operating System) είναι ένας δημοφιλής ελεγκτής SDN ανοιχτού κώδικα που παρέχει μια επεκτάσιμη και αρθρωτή πλατφόρμα για την ανάπτυξη και την ανάπτυξη εφαρμογών και υπηρεσιών δικτύου παρόμοια με το OpenDaylight. Όπως αυτό, ο ONOS έχει σχεδιαστεί για να είναι ευέλικτος και προσαρμόσιμος σε ένα ευρύ φάσμα αρχιτεκτονικών δικτύου και περιπτώσεων χρήσης.

Ένα από τα βασικά πλεονεκτήματα του ONOS είναι η κλιμακούμενη και κατανεμημένη αρχιτεκτονική του, η οποία του επιτρέπει να υποστηρίζει δίκτυα μεγάλης κλίμακας με χιλιάδες κόμβους. Αυτό το καθιστά κατάλληλο για δίκτυα παρόχων υπηρεσιών και κέντρων δεδομένων που απαιτούν υψηλά επίπεδα επεκτασιμότητας και απόδοσης. Ο ONOS περιλαμβάνει επίσης μια σειρά λειτουργιών για τη διαχείριση της κυκλοφορίας του δικτύου, όπως εξισορρόπηση φορτίου, QoS και δυναμικό τεμαχισμό δικτύου.

Ένα άλλο πλεονέκτημα του ONOS είναι η αρθρωτή αρχιτεκτονική του, η οποία το καθιστά εξαιρετικά προσαρμόσιμο σε διαφορετικές περιπτώσεις χρήσης και αρχιτεκτονικές δικτύου και όπως και το OpenDaylight έτσι και το ONOS περιλαμβάνει επίσης ένα ολοκληρωμένο σύνολο REST API.

Το ONOS διαθέτει επίσης μια σειρά από δυνατότητες και εργαλεία για τη διαχείριση της ασφάλειας δικτύου, συμπεριλαμβανομένης της υποστήριξης για τείχη προστασίας που βασίζονται σε SDN και έλεγχο πρόσβασης δικτύου (NAC). Αυτό το καθιστά κατάλληλο για οργανισμούς που απαιτούν υψηλά επίπεδα ασφάλειας και ελέγχου δικτύου.



Μια πιθανή αδυναμία του ONOS είναι η πολυπλοκότητά του. Όπως το OpenDaylight, το ONOS μπορεί να είναι δύσκολο να ρυθμιστεί και να διαμορφωθεί, ειδικά για χρήστες που είναι νέοι στο SDN. Επιπλέον, το ONOS έχει μια πιο απότομη καμπύλη εκμάθησης από ορισμένους άλλους ελεγκτές SDN, που μπορεί να απαιτούν επιπλέον χρόνο και πόρους για να εδραιωθούν.

Συνολικά, το ONOS είναι ένας ισχυρός και επεκτάσιμος ελεγκτής SDN που είναι κατάλληλος για ανάπτυξη δικτύου μεγάλης κλίμακας.

#### **4.3.1.1.3 Ελεγκτής Ryu**

Ο RYU είναι και αυτός ένας από πολύ δημοφιλής ελεγκτής SDN ανοιχτού κώδικα που σε αντίθεση με ότι είδαμε μέχρι τώρα έχει σχεδιαστεί με κύριο γνώμονα το να παρέχει μια απλή και εύχρηστη πλατφόρμα για την ανάπτυξη εφαρμογών και υπηρεσιών SDN. Είναι γραμμένος σε Python και είναι κατάλληλος για μικρές έως μεσαίου μεγέθους αναπτύξεις δικτύου.

Ένα από τα βασικά πλεονεκτήματα του RYU είναι η απλότητα και η ευκολία χρήσης του. Σε αντίθεση με ορισμένους άλλους ελεγκτές SDN, το RYU έχει σχεδιαστεί για να είναι ελαφρύ και εύκολο στη ρύθμιση και τη διαμόρφωση. Περιλαμβάνει επίσης μια σειρά εργαλείων και λειτουργιών για τη διαχείριση της κυκλοφορίας του δικτύου, συμπεριλαμβανομένης της υποστήριξης για OpenFlow και άλλα πρωτόκολλα δικτύου.

Ο RYU διαθέτει επίσης μια σειρά από δυνατότητες για τη διαχείριση της ασφάλειας δικτύου, συμπεριλαμβανομένης της υποστήριξης για τείχη προστασίας που βασίζονται σε SDN και συστήματα ανίχνευσης εισβολής (IDS). Αυτό τον καθιστά κατάλληλο για οργανισμούς που απαιτούν υψηλά επίπεδα ασφάλειας και ελέγχου δικτύου. Τέλος ο RYU διαθέτει όπως και οι υπόλοιποι ελεγκτές που είδαμε μέχρι τώρα ως βάση την αρθρωτή αρχιτεκτονική και το μοναδικό του σύνολο REST API.

Μια πιθανή αδυναμία του RYU είναι η επεκτασιμότητά του. Ενώ το RYU είναι κατάλληλο για δίκτυα μικρού έως μεσαίου μεγέθους, μπορεί να μην είναι η καλύτερη επιλογή για μεγαλύτερες αναπτύξεις δικτύου. Επιπλέον, το RYU ενδέχεται να μην έχει το ίδιο επίπεδο υποστήριξης κοινότητας με ορισμένους άλλους ελεγκτές SDN, γεγονός που θα μπορούσε να κάνει πιο δύσκολη την εύρεση πόρων και υποστήριξης.

Συνολικά, το RYU είναι ένας απλός και εύχρηστος ελεγκτής SDN που είναι κατάλληλος για μικρές έως μεσαίου μεγέθους αναπτύξεις δικτύου. Η ελαφριά αρχιτεκτονική του, τα ολοκληρωμένα API και η γκάμα χαρακτηριστικών του το καθιστούν δημοφιλή επιλογή για προγραμματιστές που είναι νέοι στο SDN και αναζητούν μια απλή και εύχρηστη πλατφόρμα για την ανάπτυξη και την ανάπτυξη εφαρμογών και υπηρεσιών SDN. Ωστόσο, η επεκτασιμότητα του μπορεί να είναι περιοριστικός παράγοντας για μεγαλύτερες αναπτύξεις δικτύου.

#### **4.3.1.1.4 Ελεγκτής Faucet**

Ο Faucet έχει ως βάση την απλότητα και την ευελιξία χτισμένος σε Python και υποστηριζόμενος από το πρωτόκολλο OpenFlow. Παρομοίως με τον Ryu ξεχωρίζει την απλότητα και είναι ελαφρύς με λειτουργίες VLAN και QOS. Παρότι δίνει έμφαση στην ασφάλεια δεν συνίσταται να χρησιμοποιηθεί για πέρα από πολύ μικρές επιχειρήσεις και αυτό με επιφυλάξεις καθώς έχει δύσκολη υποστήριξη και πολύ μικρότερη κοινότητα χρήσης.

#### **4.3.1.1.5 Ελεγκτής FRR**

Το FRRouting (FRR) είναι μια σουίτα λογισμικού δρομολόγησης ανοιχτού κώδικα που περιλαμβάνει υποστήριξη για SDN μέσω του ελεγκτή FRRouting SDN. Ο ελεγκτής FRRouting SDN έχει σχεδιαστεί για να παρέχει μια ευέλικτη και επεκτάσιμη πλατφόρμα για τη διαχείριση και τον έλεγχο της κυκλοφορίας του δικτύου.

Ένα από τα βασικά πλεονεκτήματα του FRR είναι η υποστήριξή του για ένα ευρύ φάσμα πρωτοκόλλων δρομολόγησης, συμπεριλαμβανομένων των BGP, OSPF και IS-IS. Αυτό το καθιστά κατάλληλο για μεγάλες αναπτύξεις δικτύου που απαιτούν προηγμένες δυνατότητες δρομολόγησης. Επιπλέον, το FRR περιλαμβάνει μια σειρά εργαλείων και δυνατοτήτων για τη διαχείριση της κυκλοφορίας του δικτύου, συμπεριλαμβανομένης της υποστήριξης για δυναμική δρομολόγηση, μηχανική κυκλοφορίας και MPLS.

Επιλέγεται από πολύ μεγάλους οργανισμούς με δίκτυα υψηλής ταχύτητας και μικρού περιθωρίου αποτυχίας αλλά παρότι είναι ανοικτά επεξεργάσιμος έχει τη μεγαλύτερη δυσκολία επεξεργασίας και βελτίωσης από όλους τους άλλους ελεγκτές.

#### **4.3.1.1.6 Ελεγκτής Trema**

Το Trema είναι ένα πλαίσιο ελεγκτή OpenFlow που περιλαμβάνει όλα όσα χρειάζονται για τη δημιουργία ελεγκτών OpenFlow σε Ruby και C. Το πακέτο πηγής Trema περιλαμβάνει βασικές βιβλιοθήκες και λειτουργικές μονάδες που λειτουργούν ως διεπαφή με switch OpenFlow. Παρέχονται επίσης πολλά δείγματα εφαρμογών που αναπτύχθηκαν πάνω από το Trema, ώστε να μπορείτε να τα εκτελέσετε ως δείγμα ελεγκτών OpenFlow.

Επιπλέον, παρέχεται ένα απλό αλλά ισχυρό πλαίσιο που μιμείται ένα δίκτυο που βασίζεται στο OpenFlow και τελικούς κεντρικούς υπολογιστές για τη δοκιμή ελεγκτών. Μια προσθήκη Wireshark για τη διάγνωση εσωτερικών ροών δεδομένων μεταξύ λειτουργικών μονάδων παρέχεται ως εργαλείο εντοπισμού σφαλμάτων. Επί του παρόντος, η Trema υποστηρίζει μόνο GNU/Linux και έχει δοκιμαστεί κυρίως σε πλατφόρμες UBUNTU.

#### **4.3.1.1.7 Ελεγκτής Open Contrail**

Αναπτύχθηκε από την Juniper Networks και τώρα πλέον διατηρείται και ελέγχεται από το Linux Foundation. Ένα από τα βασικά πλεονεκτήματα του OpenContrail είναι η υποστήριξή του για εικονικά δίκτυα και επικαλύψεις δικτύου. Το OpenContrail περιλαμβάνει μια σειρά από δυνατότητες για τη διαχείριση και τον έλεγχο εικονικών δικτύων, συμπεριλαμβανομένης της υποστήριξης για εικονικούς δρομολογητές, τείχη προστασίας και εξισορροπητές φορτίου. Επιπλέον, το OpenContrail περιλαμβάνει υποστήριξη για επικαλύψεις δικτύου, η οποία επιτρέπει στους διαχειριστές δικτύου να δημιουργούν εικονικά δίκτυα που εκτείνονται σε πολλά φυσικά δίκτυα.

Ένα άλλο πλεονέκτημα του OpenContrail είναι τα χαρακτηριστικά ασφαλείας του. Το OpenContrail περιλαμβάνει μια σειρά από χαρακτηριστικά ασφαλείας, συμπεριλαμβανομένης της υποστήριξης για εικονικά τείχη προστασίας και τμηματοποίηση δικτύου, τα οποία επιτρέπουν στους διαχειριστές δικτύου να δημιουργούν ασφαλή εικονικά δίκτυα. Επιπλέον, το OpenContrail περιλαμβάνει υποστήριξη για ασφαλή επικοινωνία μεταξύ στοιχείων δικτύου, η οποία βοηθά στην προστασία από επιθέσεις δικτύου και παραβιάσεις δεδομένων.

Μεγάλες επιχειρήσεις με χιλιάδες εργαζόμενους και εγκαταστάσεις σε όλο τον κόσμο θα χρησιμοποιήσουν το OpenContrail καθώς μπορεί να υποστηρίξει αριθμό εικονικών δικτύων εκθετικά αυξανόμενο με μεγάλη αποτελεσματικότητα διαχείρισης.

#### **4.3.1.1.8 Project Calico**

Το Project Calico είναι μια πλατφόρμα ελεγκτή SDN ανοιχτού κώδικα που έχει σχεδιαστεί για να παρέχει μια επεκτάσιμη και ασφαλή πλατφόρμα για τη διαχείριση και τον έλεγχο της κυκλοφορίας του δικτύου. Αναπτύχθηκε από την Tigera και τώρα συντηρείται από το Cloud Native Computing Foundation (CNCF).

Ένα από τα βασικά πλεονεκτήματα του Project Calico είναι η υποστήριξή του για την επιβολή της πολιτικής δικτύου. Το Project Calico περιλαμβάνει μια σειρά λειτουργιών για τη διαχείριση και τον έλεγχο της κυκλοφορίας δικτύου, συμπεριλαμβανομένης της υποστήριξης για πολιτικές δικτύου που επιτρέπουν στους διαχειριστές να προσδιορίζουν ποια κίνηση δικτύου επιτρέπεται και ποια είναι αποκλεισμένη. Επιπλέον, το Project Calico περιλαμβάνει υποστήριξη για ταυτότητα φόρτου εργασίας, η οποία επιτρέπει στους διαχειριστές δικτύου να εφαρμόζουν πολιτικές που βασίζονται στην ταυτότητα μεμονωμένων φόρτων εργασίας.

Ένα άλλο πλεονέκτημα του Project Calico είναι η επεκτασιμότητά του. Το Project Calico έχει σχεδιαστεί για να είναι εξαιρετικά επεκτάσιμο και μπορεί να υποστηρίξει μεγάλες αναπτύξεις δικτύου που εκτείνονται σε πολλά κέντρα δεδομένων και περιβάλλοντα cloud. Επιπλέον, το Project Calico περιλαμβάνει υποστήριξη για επικαλύψεις δικτύου, η οποία επιτρέπει στους διαχειριστές δικτύου να δημιουργούν εικονικά δίκτυα που εκτείνονται σε πολλά φυσικά δίκτυα.

Συνολικά, το Project Calico είναι ένας εξαιρετικά επεκτάσιμος και ασφαλής ελεγκτής SDN που είναι κατάλληλος για μεγάλες αναπτύξεις δικτύου που απαιτούν προηγμένες δυνατότητες επιβολής πολιτικής δικτύου και χαρακτηριστικά ταυτότητας φόρτου εργασίας. Η υποστήριξή του για επικαλύψεις δικτύου και επεκτασιμότητα το καθιστούν δημοφιλή επιλογή για παρόχους υπηρεσιών και άλλους οργανισμούς που απαιτούν έναν εξαιρετικά επεκτάσιμο και ευέλικτο ελεγκτή SDN. Ωστόσο, η πολυπλοκότητά του μπορεί να είναι περιοριστικός παράγοντας για οργανισμούς που απαιτούν έναν απλό και εύχρηστο ελεγκτή SDN.

#### **4.3.1.2 Ανενεργοί ελεγκτές**

Παρακάτω θα αναλυθούν οι ελεγκτές SDN οι οποίοι είναι ανενεργοί. Ως ανενεργοί ορίζονται αυτοί οι ελεγκτές που δεν λαμβάνουν υποστήριξη, ενημερώσεις και βελτιώσεις

και δεν χρησιμοποιούνται από επιχειρήσεις και οργανισμούς. Ο παρακάτω πίνακας δείχνει υλοποιησεις ελεγκτών ανοιχτού κώδικα οι οποίοι είναι ανενεργοί.

<b>ΑΝΕΝΕΡΓΟΙ ΕΛΕΓΚΤΕΣ</b>
Floodlight
LOOM
OpenMUL
Maestro
Beacon
Flower
NOX

*Πίνακας 4.3 Ανενεργοί ελεγκτές.*

#### **4.3.1.2.1 Ελεγκτής Floodlight**

Ο ελεγκτής Floodlight μπορεί να είναι επωφελής για τους προγραμματιστές επειδή τους προσφέρει τη δυνατότητα να προσαρμόζουν εύκολα λογισμικό και να αναπτύσσουν εφαρμογές και είναι γραμμένο σε Java. Περιλαμβάνονται διεπαφές προγράμματος μεταφοράς αντιπροσωπευτικής κατάστασης (REST API) που διευκολύνουν τον προγραμματισμό της διεπαφής με το προϊόν και ο ιστότοπος Floodlight προσέφερε παραδείγματα κωδικοποίησης που βοηθούν τους προγραμματιστές να δημιουργήσουν το προϊόν.

Δοκιμασμένο τόσο με φυσικούς όσο και με εικονικά switch συμβατά με OpenFlow, το Floodlight Controller μπορεί να λειτουργήσει σε διάφορα περιβάλλοντα και μπορεί να συμπίπτει με αυτό που έχουν ήδη οι επιχειρήσεις στη διάθεσή τους. Μπορεί επίσης να υποστηρίξει δίκτυα όπου συνδέονται ομάδες διακοπών συμβατών με OpenFlow μέσω παραδοσιακών, μη-OpenFlow switch.

Το Floodlight Controller είναι συμβατό με το OpenStack, ένα σύνολο εργαλείων λογισμικού που βοηθούν στη δημιουργία και διαχείριση πλατφορμών υπολογιστικού νέφους τόσο για δημόσια όσο και για ιδιωτικά cloud. Το Floodlight μπορεί να εκτελεστεί ως back-end δικτύου για το OpenStack χρησιμοποιώντας μια προσθήκη Neutron που εκθέτει ένα μοντέλο δικτύωσης ως υπηρεσία με ένα REST API που προσφέρει το Floodlight.

#### **4.3.1.2.2 Ελεγκτής LOOM**

Ο σκοπός του LOOM είναι να σχεδιάσει και να πρωτοτυπήσει έναν πειραματικό ελεγκτή-μεταγωγέα δικτύου που υλοποιεί τα πρωτόκολλα OpenFlow 1.3.x και 1.4. Το έργο διερευνά την επεκτασιμότητα και την ευρωστία τέτοιων ελεγκτών σε μια κλίμακα πολύ μεγαλύτερη από ό,τι συνήθως θεωρείται: 100.000s τερματικών σημείων, 10.000s των switch. Ως εκ

τούτου, το LOOM χρησιμοποιεί μια κατανεμημένη αρχιτεκτονική με ένα μοναδικό σχήμα διαχείρισης δεδομένων για την αποθήκευση διαμορφώσεων μεταγωγέων και την τοπολογία δικτύου για την αντιμετώπιση των διπλών προκλήσεων της τεράστιας κλίμακας και της ευρωστίας και αξιοπιστίας σε επίπεδο επιχείρησης. Το LOOM είναι μια πλατφόρμα που υποστηρίζει εφαρμογές τρίτων. Η σχεδίαση LOOM διερευνά τις τακτικές για την ενσωμάτωση αυτών των εφαρμογών στην κατανεμημένη αρχιτεκτονική του LOOM προστατεύοντας παράλληλα το δίκτυο από αδίστακτες εφαρμογές. Αξίζει να σημειωθεί πως η τελευταία ενημέρωση του LOOM έγινε το 2014 και από τότε εγκαταλείφθηκε.

#### **4.3.1.2.3 Ελεγκτής Open MUL**

Το Open Mul είναι μια πλατφόρμα ελεγκτή Openflow / SDN. Έχει στον πυρήνα του μια υποδομή πολλαπλών νημάτων που βασίζεται στη γλώσσα C. Υποστηρίζει μια διεπαφή πολλαπλών επιπέδων βόρειας σύνδεσης για τη φιλοξενία εφαρμογών. Στόχος του είναι να υποστηρίξει διάφορα πρωτόκολλα SDN που ενεργοποιούν νότια, όπως Openflow 1.4, 1.3 και 1.0 μαζί με onvsdb, of-config κ.λπ.

Έχει σχεδιαστεί για απόδοση και αξιοπιστία που είναι η ανάγκη της ώρας για ανάπτυξη σε κρίσιμα για την αποστολή δίκτυα. Είναι επίσης εξαιρετικά ευέλικτο, αρθρωτό και εύκολο στην εκμάθηση. Η υποστήριξη του δεν κράτησε πολύ καθώς έληξε το 2012.

#### **4.3.1.2.4 Ελεγκτής Maestro**

Το Maestro είναι ένα «λειτουργικό σύστημα» για την οργάνωση εφαρμογών ελέγχου δικτύου. Το Maestro παρέχει διεπαφές για την εφαρμογή αρθρωτών εφαρμογών ελέγχου δικτύου για την πρόσβαση και την τροποποίηση της κατάστασης του δικτύου και τον συντονισμό των αλληλεπιδράσεων τους. Το Maestro είναι μια πλατφόρμα για την επίτευξη αυτόματων και προγραμματικών λειτουργιών ελέγχου δικτύου χρησιμοποιώντας αυτές τις αρθρωτές εφαρμογές. Αν και αυτό το έργο εστιάζει στην κατασκευή ενός ελεγκτή OpenFlow χρησιμοποιώντας το Maestro, το Maestro δεν περιορίζεται μόνο στα δίκτυα OpenFlow. Το πλαίσιο προγραμματισμού του Maestro παρέχει διεπαφές για:

- Εισαγωγή νέων προσαρμοσμένων λειτουργιών ελέγχου με την προσθήκη αρθρωτών στοιχείων ελέγχου.
- Διατήρηση της κατάστασης του δικτύου για λογαριασμό των στοιχείων ελέγχου.
- Σύνθεση στοιχείων ελέγχου καθορίζοντας την ακολουθία εκτέλεσης και την κατάσταση κοινόχρηστου δικτύου των στοιχείων.

Επιπλέον, το Maestro προσπαθεί να εκμεταλλευτεί τον παραλληλισμό μέσα σε ένα μόνο μηχανήμα για να βελτιώσει την απόδοση του συστήματος. Το θεμελιώδες χαρακτηριστικό ενός δικτύου OpenFlow είναι ότι ο ελεγκτής είναι υπεύθυνος για την αρχική εγκατάσταση κάθε ροής επικοινωνώντας με σχετικούς μεταγωγείς. Έτσι, η απόδοση του ελεγκτή θα μπορούσε να είναι το σημείο συμφόρησης. Κατά τη σχεδίαση του Maestro προσπαθούμε να απαιτήσουμε όσο το δυνατόν λιγότερη προσπάθεια από τους προγραμματιστές για τη διαχείριση της παραλληλοποίησης. Αντίθετα, το Maestro χειρίζεται το μεγαλύτερο μέρος της κουραστικής και περίπλοκης δουλειάς της διαχείρισης της κατανομής του φόρτου εργασίας και του προγραμματισμού των νημάτων των εργαζομένων. Με σχεδιασμό το Maestro είναι φορητό και επεκτάσιμο:

- Αναπτύχθηκε σε Java (τόσο η πλατφόρμα όσο και τα εξαρτήματα) - Εξαιρετικά φορητό σε διάφορα λειτουργικά συστήματα και αρχιτεκτονικές.
- Multi-threaded - Εκμεταλλεύεται πλήρως τους επεξεργαστές πολλαπλών πυρήνων.

Ήτανε μία από τις πρώτες ολοκληρωμένες υλοποιήσεις ελεγκτών και ευρέως χρησιμοποιημένος μέχρι το 2013.

#### **4.3.1.2.5 Ελεγκτής Beacon**

Υπήρξε βασική παραλλαγή του πρώτου ελεγκτή SDN του πανεπιστημίου του Standford. Οι δυνατότητές του περιοριζόταν κυρίως σε μορφή ελαφρών πειραμάτων για επιστημονικούς σκοπούς. Αναλυτικότερα βασίζεται σε Java και το πείραμα γύρω από το οποίο ξεκίνησε ήτανε η διαχείριση ενός πειραματικού κέντρου με 100 εικονικά switch και 20 φυσικά. Μπορούσε να έχει πλήρη διαχείριση για περίπου έναν ολόκληρο χρόνο χωρίς την παραμικρή υποψία σφάλματος ή πτώσης.

#### **4.3.1.2.6 Ελεγκτής Flower**

Το SDN-Flower είναι μια πλατφόρμα ελεγκτή SDN ανοιχτού κώδικα που παρέχει μια φιλική προς το χρήστη γραφική διεπαφή χρήστη (GUI) για τη διαχείριση και τον έλεγχο της κυκλοφορίας δικτύου. Αναπτύχθηκε σε Python και βασίζεται στο πρότυπο OpenFlow.

Ένα από τα βασικά πλεονεκτήματα του SDN-Flower είναι η ευκολία χρήσης του. Το SDN-Flower παρέχει ένα απλό και διαισθητικό GUI που επιτρέπει στους διαχειριστές δικτύου να δημιουργούν και να διαχειρίζονται εύκολα τις πολιτικές δικτύου και τις ροές

κυκλοφορίας. Επιπλέον, το SDN-Flower περιλαμβάνει υποστήριξη για παρακολούθηση και οπτικοποίηση δικτύου σε πραγματικό χρόνο, γεγονός που καθιστά εύκολο τον εντοπισμό και την αντιμετώπιση προβλημάτων δικτύου.

Μια πιθανή αδυναμία του SDN-Flower είναι ότι μπορεί να μην είναι τόσο επεκτάσιμο ή αποδοτικό όσο ορισμένοι άλλοι ελεγκτές SDN, ιδιαίτερα για μεγάλες αναπτύξεις δικτύου. Επιπλέον, η εστίαση του SDN-Flower στην παροχή ενός φιλικού προς τον χρήστη GUI μπορεί να το κάνει λιγότερο κατάλληλο για οργανισμούς που απαιτούν μια πιο προγραμματική προσέγγιση στη διαχείριση δικτύου.

### 4.3.2 Ελεγκτής NOX/POX

Ο NOX ήταν ο πρώτος ελεγκτής SDN. Αναπτύχθηκε αρχικά από τη Nicira Networks και ήταν η πρώτη που υποστήριξε το πρωτόκολλο OpenFlow. Δόθηκε στην ερευνητική κοινότητα το 2009 και αποτέλεσε τη βάση για πολλά ερευνητικά έργα στο SDN. Στη συνέχεια επεκτάθηκε και υποστηρίχθηκε στο Πανεπιστήμιο του Στάνφορντ με σημαντική συμβολή από το Πανεπιστήμιο της Καλιφόρνια, Μπέρκλεϋ. Μερικές δημοφιλείς εφαρμογές NOX είναι η SANE (μια προσέγγιση για που αντιπροσωπεύει το δίκτυο ως σύστημα αρχείων) και η Ethane. Σήμερα το NOX θεωρείται ανενεργό. Κατά τη διάρκεια αυτών των ετών έχουν παρουσιαστεί διαφορετικές εκδόσεις του NOX. Αυτές είναι γνωστές ως NOX, NOX-MT και POX. Το νέο NOX υποστηρίζει μόνο C++. Έχει μικρότερο δίκτυο εφαρμογών σε σύγκριση με το NOX, αλλά είναι πολύ πιο γρήγορο και έχει πολύ πιο καθαρή βάση κωδικών. Το NOX-MT, που εισήχθη στο είναι μια ελαφρώς τροποποιημένη έκδοση του ελεγκτή NOX που χρησιμοποιεί τεχνικές βελτιστοποίησης για την εισαγωγή της επεξεργασίας πολλαπλών νημάτων και για τη βελτίωση του ρυθμού και του χρόνου απόκρισης NOX. Αυτές οι τεχνικές βελτιστοποίησης περιλαμβάνουν ομαδοποίηση I/O σε ελαχιστοποίηση των γενικών εξόδων εισόδου/εξόδου και άλλων. Το POX είναι η νεότερη έκδοση που βασίζεται σε Python. Η ιδέα πίσω από την ανάπτυξή του ήταν να επιστρέψει το NOX στις ρίζες του στη C++ και να αναπτυχθεί μια ξεχωριστή πλατφόρμα βασισμένη σε Python. Διαθέτει επίσης διεπαφή Python OpenFlow, επαναχρησιμοποιήσιμα δείγματα στοιχείων για επιλογή διαδρομής, ανακάλυψη τοπολογίας κ.α. Ο πρωτεύον στόχος του POX είναι η έρευνα. Δεδομένου ότι πολλά από τα ερευνητικά έργα είναι από τη φύση τους βραχύβια, η εστίαση των προγραμματιστών του POX είναι στις καλές διεπαφές αντί για σταθερότητα API. Στην παρούσα έρευνα λόγω των πολλαπλών διεπαφών και της σταθερότητας του καθώς και λόγω του ότι στην



πορεία χρησιμοποιήθηκε η γλώσσα Python για την δημιουργία δικτυακών τοπολογιών χρησιμοποιήθηκε ο POX [8] .

Γενικά, ο ελεγκτής NOX παρέχει ένα πλήρες OpenFlow API χρησιμοποιώντας C ++ και Γλώσσες Python, χρησιμοποιεί ασύγχρονες εισόδους / εξόδους (I/O) και είναι προσανατολισμένος στη λειτουργία σε συστήματα Linux, Ubuntu και Debian. Το NOX χρησιμοποιείται τόσο ως αυτόνομος ελεγκτής όσο και ως πλαίσιο που βασίζεται σε εξαρτήματα για την ανάπτυξη εφαρμογών SDN. Είναι χτισμένο σε ένα μοντέλο προγραμματισμού που βασίζεται σε συμβάντα και υιοθετεί ένα απλό μοντέλο διεπαφών προγραμματισμού που περιστρέφεται γύρω από τρεις πυλώνες:

- συμβάντα
- χώρο ονομάτων
- προβολή δικτύου

Τα συμβάντα μπορούν να δημιουργηθούν είτε απευθείας από Μηνύματα OpenFlow ή από εφαρμογές NOX ως αποτέλεσμα επεξεργασίας συμβάντων χαμηλού επιπέδου ή άλλα συμβάντα που δημιουργούνται από εφαρμογές .

<b>ΕΝΕΡΓΟΙ ΕΛΕΥΘΕΡΟΙ ΕΛΕΓΚΤΕΣ</b>	<b>ΒΑΣΙΚΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ</b>	<b>ΒΑΣΙΚΑ ΜΕΙΟΝΕΚΤΗΜΑΤΑ</b>
NOX/POX	Απλότητα, προσαρμοστικότητα, υποστήριξη	Έλλειψη υποστήριξης από εταιρείες δικτύων, συμβατότητα, ασφάλεια
OpenDaylight	Προσαρμοστικότητα, συμβατότητα , υποστήριξη από εταιρείες δικτύων	Έλλειψη υποστήριξης , μεγάλη κατανάλωση πόρων, δύσχρηστος
ONOS	Υψηλές επιδόσεις , επεκτασιμότητα , προσαρμοστικότητα	Μηδενική υποστήριξη , δύσχρηστος , μεγάλη κατανάλωση πόρων
Ryu	Ευκολία , ελάχιστη κατανάλωση χώρου και πόρων , συμβατότητα	Περιορισμένη επεκτασιμότητα , μηδενική υποστήριξη , μηδενική ασφάλεια
Faucet	Προσαρμοστικότητα, συμβατότητα , απλότητα	Ελάχιστες δυνατότητες δικτύωσης , ελάχιστες δυνατότητες ασφαλείας , χρήση μόνο σε μικρά δίκτυα
FRR	Κλιμάκωση , Συμβατότητα , ευκολία	Μηδενικές δυνατότητες δικτύωσης , ελλιπείς γραφικό περιβάλλον , ελάχιστη υποστήριξη

Trema	Μικρή καθυστέρηση μετάδοσης, συμβατότητα , προσαρμοστικότητα	Χρονικά ξεπερασμένος , δεν τελειοποιήθηκε ποτέ , δεν βρήκε εφαρμογή στην βιομηχανία
OpenContrail	Υποστήριξη Cloud , ενισχυμένη ασφάλεια, επεκτασιμότητα	Προβλήματα συμβατότητας , περιπλοκότητα , δύσκολο περιβάλλον χρήσης
Project Calico	Ενισχυμένη ασφάλεια ,υποστήριξη Cloud , στήνεται σε δίκτυα χωρίς να αλλάξει-επιβαρύνει τη δομή τους	Χρονικά ξεπερασμένος , περιορισμένες δυνατότητες , απαιτεί πολύ εξειδικευμένους χρήστες

Πίνακας 4.4 Σύγκριση ελεύθερων ανοιχτών ελεγκτών.

### 4.3.3 Εμπορικοί Ελεγκτές

Υπάρχουν αρκετοί εμπορικοί ελεγκτές SDN διαθέσιμοι στην αγορά σήμερα. Αυτοί οι ελεγκτές αναπτύσσονται και πωλούνται από προμηθευτές που προσφέρουν διαφορετικές δυνατότητες και υπηρεσίες υποστήριξης και απευθύνονται κυρίως σε μεγάλους οργανισμούς με υψηλές απαιτήσεις απόδοσης και ασφαλείας οι οποίοι δε θέλουν να θυσιάσουν στο ελάχιστο κάτι από τα δύο και επιλέγουν να λύσουν το πρόβλημα της επιπλέον εξειδίκευσης προσωπικού που απαιτεί η παραμετροποίηση των ελεύθερων ελεγκτών.

ΕΜΠΟΡΙΚΟΙ ΕΛΕΓΚΤΕΣ
Cisco-ACI
Juniper Networks Contrail
VMware NSX
HP-IMC
NEC-ProgrammableFlow

Πίνακας 4.5 Εμπορικοί ελεγκτές.

#### 4.3.3.1 Cisco Application Centric Infrastructure (ACI)

Η Cisco Systems, Inc. είναι μια πολυεθνική εταιρεία με βάση το Σαν Χοσέ της Καλιφόρνια, η οποία σχεδιάζει και εμπορεύεται ηλεκτρονικά προϊόντα, συσκευές δικτύωσης υπολογιστών, και προϊόντα και υπηρεσίες τηλεπικοινωνιών. Έχει περισσότερους από 100 χιλιάδες υπαλλήλους σε όλο τον κόσμο, και ετήσια έσοδα 50 δισεκατομμυρίων δολαρίων το 2021.

Το Cisco ACI είναι μια λύση SDN που καθορίζει την υποδομή δικτύου της με βάση τις πολιτικές δικτύου. Για να γίνει αυτό δυνατό, η Cisco δημιούργησε το ACI Fabric OS, το οποίο τρέχει σε όλα τα συστήματα εντός του δικτύου ACI. Αυτό το κοινόχρηστο

λειτουργικό σύστημα επιτρέπει στους διάφορους μεταγωγείς εντός του δικτύου ACI να μεταφράζουν τις πολιτικές σε σχέδια υποδομής.

Η υποδομή με επίκεντρο την εφαρμογή (ACI) είναι μια προσέγγιση στην αρχιτεκτονική δικτύου που τοποθετεί τις εφαρμογές στο επίκεντρο των λειτουργιών του δικτύου. Είναι μια τεχνολογία δικτύωσης που καθορίζεται από λογισμικό που αναπτύχθηκε από τη Cisco και ενσωματώνει φυσικά και εικονικά περιβάλλοντα για να παρέχει μια ολιστική και αυτοματοποιημένη προσέγγιση στη διαχείριση δικτύου. Το ACI έχει σχεδιαστεί για να απλοποιεί την ανάπτυξη και τη διαχείριση πολύπλοκων δικτυακών υποδομών παρέχοντας ένα κεντρικό πλαίσιο διαχείρισης που επιτρέπει στους διαχειριστές να ορίζουν πολιτικές δικτύου με βάση τις επιχειρηματικές ανάγκες και τις απαιτήσεις εφαρμογών. Αυτό βοηθά στη μείωση της πολυπλοκότητας των λειτουργιών του δικτύου και στη βελτίωση της συνολικής ευελιξίας του δικτύου.

Η ACI χρησιμοποιεί μια σειρά από διαφορετικές τεχνολογίες για να επιτύχει τους στόχους της, όπως:

1. Προφίλ εφαρμογών: Το ACI χρησιμοποιεί προφίλ εφαρμογών για να καθορίσει τις απαιτήσεις συγκεκριμένων εφαρμογών, συμπεριλαμβανομένων των απαιτήσεων συνδεσιμότητας, ασφάλειας και ποιότητας υπηρεσίας (QoS).
2. Πολιτικές δικτύου: Το ACI επιτρέπει στους διαχειριστές να ορίσουν πολιτικές δικτύου που υπαγορεύουν τον τρόπο σύνδεσης των εφαρμογών και τον τρόπο κατανομής των πόρων δικτύου για να διασφαλιστεί ότι οι εφαρμογές λαμβάνουν το απαραίτητο επίπεδο απόδοσης και ασφάλειας.
3. Προγραμματιζόμενος ιστός δικτύου: Το ACI χρησιμοποιεί έναν προγραμματιζόμενο ιστό δικτύου για την αυτοματοποίηση της παροχής και της διαμόρφωσης δικτύου, επιτρέποντας στους διαχειριστές να αναπτύξουν γρήγορα και εύκολα και να κλιμακώσουν την υποδομή του δικτύου τους.
4. Ενσωμάτωση με εργαλεία τρίτων: Η ACI ενσωματώνεται με μια σειρά εργαλείων και υπηρεσιών τρίτων, συμπεριλαμβανομένων συστημάτων ασφαλείας, συστημάτων αποθήκευσης και πλατφορμών cloud, για να παρέχει μια ολοκληρωμένη λύση για τη διαχείριση της υποδομής δικτύου.

Συνολικά, το ACI παρέχει ένα ισχυρό και ευέλικτο πλαίσιο για τη διαχείριση της υποδομής δικτύου με τρόπο που είναι βελτιστοποιημένος για σύγχρονους φόρτους εργασίας με επίκεντρο την εφαρμογή.

- Αρχιτεκτονική Cisco Application Centric Infrastructure (ACI).

Ένα περιβάλλον Cisco ACI είναι κατασκευασμένο με δύο βασικά στοιχεία:

1. Cisco Application Policy Infrastructure Controller (APIC): Το APIC είναι ο ελεγκτής SDN για το Cisco ACI. Δημιουργεί τις πολιτικές που καθορίζουν την υποδομή δικτύου του κέντρου δεδομένων.
2. Switch Nexus 9000: Τα Switch Nexus 9000 χρησιμοποιούν το ACI Fabric OS για να επικοινωνούν με το APIC και να δημιουργούν υποδομή βάσει πολιτικών. Μπορούν να είναι Spine (διανομής) ή Leaf (πρόσβασης).

Όλα τα τελικά σημεία, συμπεριλαμβανομένων των APIC, συνδέονται στο δίκτυο μέσω switch Leaf. Τα Leaf switch συνδέονται μεταξύ τους χρησιμοποιώντας switch Spine .

Χρησιμοποιώντας αυτά τα στοιχεία, το ACI μπορεί να αναπτυχθεί σε μια ποικιλία διαφορετικών μοντέλων. Αυτό περιλαμβάνει υποστήριξη για επιτόπου περιβάλλοντα, που βασίζονται σε cloud (συμπεριλαμβανομένων των δημόσιων, ιδιωτικών και υβριδικών cloud) και περιβαλλόντων SD-WAN. Αυτό επιτρέπει στους οργανισμούς να χρησιμοποιούν τη διαχείριση δικτύου που βασίζεται σε πολιτικές σε όλα τα εταιρικά τους WAN.

- Βασικά χαρακτηριστικά και πλεονεκτήματα του ACI

Το Cisco ACI επιτρέπει στους οργανισμούς να δημιουργούν εύκολα ένα κέντρο δεδομένων που καθορίζεται από λογισμικό, το οποίο παρέχει πολλά οφέλη, όπως:

1. Ευελιξία: Με μια λύση SDN όπως το Cisco ACI, όλη η δικτυακή υποδομή ενός οργανισμού υλοποιείται ως κώδικας. Αυτό καθιστά εύκολη την ενημέρωση των διαμορφώσεων για την κάλυψη των εξελισσόμενων επιχειρηματικών αναγκών.
2. Συνεπής υποδομή: Το Cisco ACI αφαιρεί τις λεπτομέρειες της υποκείμενης υποδομής. Αυτό διευκολύνει τον σχεδιασμό και τη διαμόρφωση περιβαλλόντων δικτύου.

3. Αυτοματισμός και οργάνωση: Η Cisco ACI κάνει μεγάλη χρήση του αυτοματισμού για την ανάπτυξη υποδομής δικτύου με βάση τις πολιτικές δικτύου. Αυτό καθιστά εύκολη την πραγματοποίηση αλλαγών και αυξάνει την επεκτασιμότητα.
4. Υποστήριξη για υβριδικά περιβάλλοντα: Το Cisco ACI υποστηρίζει υποδομές on-prem και cloud-based, καθιστώντας δυνατή την ανάπτυξη περιβαλλόντων ACI σε πολλά διαφορετικά περιβάλλοντα.

- Αύξηση της ασφάλειας Cisco ACI με Σημείο ελέγχου

Η Cisco ACI παρέχει μια σειρά από ενσωματωμένες λύσεις ασφαλείας. Μια συνεργασία με το Check Point ενισχύει αυτές τις προστασίες για να παρέχει προηγμένη πρόληψη απειλών για περιβάλλοντα ACI.

- Ενσωμάτωση Cisco με άλλες συσκευές

Το Cisco ACI έχει κατασκευαστεί χρησιμοποιώντας το Cisco ACI Open Ecosystem. Αυτό το Open Ecosystem έχει σχεδιαστεί για να παρέχει έναν αριθμό διαφορετικών επιλογών για τη σύνδεση εργαλείων τρίτων στο Cisco ACI, όπως:

1. Ανοικτά API: Τα API της Cisco ACI είναι ανοιχτά, επιτρέποντας σε άλλα προϊόντα να συνδέονται και να αλληλεπιδρούν με ένα περιβάλλον Cisco ACI.
2. Από κοινού πιστοποιημένες λύσεις: Η Cisco έχει συνεργαστεί με περισσότερους από 65 παρόχους τεχνολογίας για τη δημιουργία ενός οικοσυστήματος SDN. Αυτές οι κοινές λύσεις είναι πιστοποιημένες από κάθε οργανισμό και παρέχονται πληροφορίες και από τα δύο μέρη για τη διασφάλιση της συμβατότητας των προκατασκευασμένων λύσεων.
3. Service Chaining: Οι λύσεις Cisco ACI υποστηρίζουν την αλυσίδα υπηρεσιών, επιτρέποντας στους οργανισμούς να δημιουργήσουν τις λύσεις που χρειάζονται για να ανταποκριθούν στις απαιτήσεις δικτύωσης και ασφαλείας.

- Check Point CloudGuard και το Cisco ACI

Το Check Point CloudGuard Network Security παρέχει συνεπή διαχείριση πολιτικής και επιβολή προηγμένων προστασιών ασφαλείας, αναπτύσσεται αυτόματα και ενορχηστρώνεται δυναμικά σε περιβάλλοντα κέντρων δεδομένων που καθορίζονται από λογισμικό. Το CloudGuardfor Cisco ACI παρέχει κορυφαία ασφάλεια στον κλάδο για

περιβάλλοντα ACI. Το CloudGuard παρέχει τις ακόλουθες δυνατότητες για τη βελτίωση της ασφάλειας Cisco ACI των πελατών:

1. **Ορατότητα και οπτικοποίηση δικτύου Cloud:** Το CloudGuard εφαρμόζει μικροτμηματοποίηση για περιβάλλοντα ACI, παρέχοντας βαθιά εικόνα για τις ροές κυκλοφορίας δεδομένων βορρά-νότου και ανατολής-δύσης. Αυτή η αναλυτική ορατότητα βοηθά στην κατανόηση των ροών δεδομένων μέσα σε ένα εταιρικό δίκτυο και στην επιβολή εταιρικών πολιτικών ασφάλειας.
2. **Προηγμένη πρόληψη απειλών:** Οι προηγμένες δυνατότητες πρόληψης απειλών του CloudGuard συνδυάζουν μια πλήρη στοίβα ασφαλείας – συμπεριλαμβανομένου τείχους προστασίας, συστήματος πρόληψης εισβολής (IPS), προστασίας από ιούς και προστασίας από ρομπότ – με ασφαλή απομακρυσμένη πρόσβαση, εξαγωγή απειλών και εξομοίωση απειλών βάσει sandbox.
3. **Αυτοματισμός και οργάνωση:** Το Cisco ACI επιτρέπει τον καθορισμό της υποδομής δικτύου με βάση τις πολιτικές δικτύου. Η ενοποίηση μεταξύ του Check Point CloudGuard και του Cisco ACI σημαίνει ότι ένας οργανισμός μπορεί αυτόματα να εισάγει και να παρέχει πύλες ασφαλείας CloudGuard σε περιβάλλοντα ACI για την επιβολή της πολιτικής ασφαλείας.
4. **Επιβολή πολιτικής και συμμόρφωσης:** Το CloudGuard λαμβάνει πλαίσιο από το APIC της Cisco, το οποίο επιτρέπει τη χρήση πληροφοριών πολιτικής που ορίζονται στο περιβάλλον ACI για τον γρήγορο καθορισμό πολιτικών ασφαλείας. Αυτές οι πολιτικές ασφαλείας μπορούν στη συνέχεια να επιβληθούν εύκολα στο ACI χρησιμοποιώντας πύλες CloudGuard.
5. **Προστασία δεδομένων:** Η ενοποίηση του CloudGuard με το ACI του επιτρέπει να εφαρμόζει την πρόληψη απώλειας δεδομένων (DLP) σε περιβάλλοντα ACI. Αυτό βοηθά στην προστασία των ευαίσθητων δεδομένων ενός οργανισμού από απώλεια ή κλοπή.
6. **Κεντρική διαχείριση ασφάλειας:** Η χρήση του CloudGuard με ACI επιτρέπει την παρακολούθηση και τη διαχείριση της ασφάλειας του οικοσυστήματος ACI από την ίδια κονσόλα με την υπόλοιπη δικτυακή υποδομή ενός οργανισμού. Αυτό διευκολύνει τους αναλυτές ασφαλείας να εντοπίζουν και να ανταποκρίνονται σε πιθανές απειλές μέσα στα δημόσια, ιδιωτικά και on-prem δίκτυά τους.

Η ολοκληρωμένη λύση Check Point και Cisco επιτρέπει στους οργανισμούς να δημιουργούν και να ασφαλίζουν εύκολα ένα κέντρο δεδομένων που καθορίζεται από λογισμικό και βελτιώνει την ασφάλεια του Cisco API. [24]

#### **4.3.3.1.1 Αρχιτεκτονική Spine-Leaf**

Μια αρχιτεκτονική Spine-Leaf είναι η τοπολογία δικτύου κέντρων δεδομένων που αποτελείται από δύο στρώματα μεταγωγής – spine και leaf. Το επίπεδο leaf αποτελείται από switch πρόσβασης που συγκεντρώνουν την κίνηση από διακομιστές και συνδέονται απευθείας στη spine ή στον πυρήνα του δικτύου. Τα spine switch διασυνδέουν όλα τα leaf switch σε μια τοπολογία πλήρους πλέγματος.

Πώς διαφέρει μια αρχιτεκτονική leaf-spine από τα παραδοσιακά σχέδια δικτύων;

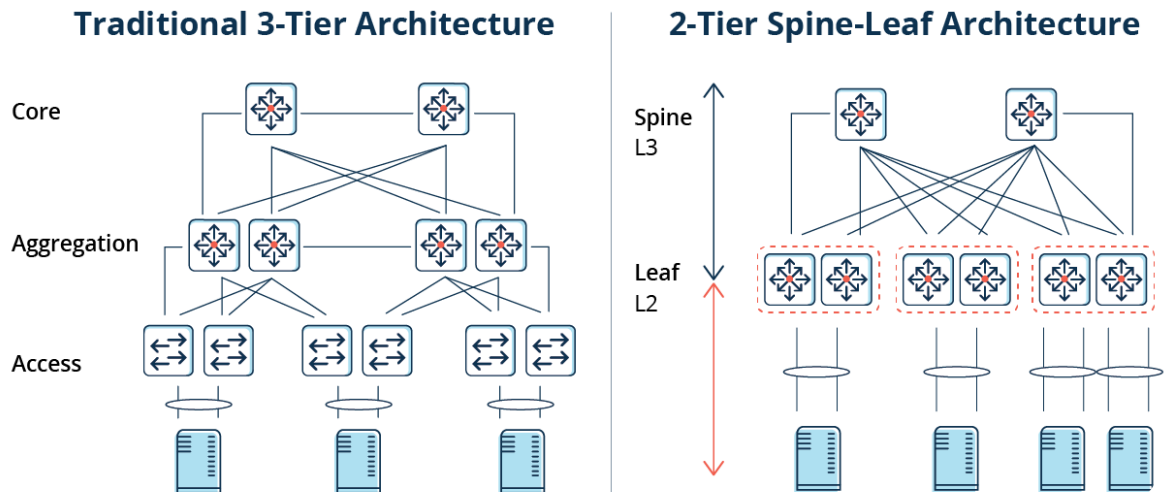
Παραδοσιακά, τα δίκτυα κέντρων δεδομένων βασίζονταν σε ένα μοντέλο τριών επιπέδων:

1. Τα switch πρόσβασης συνδέονται με διακομιστές
2. Τα switch συγκέντρωσης ή διανομής παρέχουν περιττές συνδέσεις σε switch πρόσβασης
3. Τα switch πυρήνα παρέχουν γρήγορη μεταφορά μεταξύ των switch συγκέντρωσης, συνήθως συνδεδεμένοι σε ένα πλεονάζον ζεύγος για υψηλή διαθεσιμότητα

Στο πιο βασικό επίπεδο, μια αρχιτεκτονική leaf-spine κινείται σε καθένα από αυτά τα επίπεδα, όπως απεικονίζεται στα διαγράμματα παρακάτω.

Άλλες επιπλέον διαφορές είναι:

1. Η κατάργηση του Spanning Tree Protocol (STP)
2. Αυξημένη χρήση σταθερών θυρών εναλλάσσει τα αρθρωτά μοντέλα για τον κορμό του δικτύου
3. Περισσότερη καλωδίωση για αγορά και διαχείριση, δεδομένου του υψηλότερου αριθμού διασύνδεσης
4. Αντίστροφη δικτυακή κλιμάκωση



Εικόνα 4.2: Σύγκριση παραδοσιακής αρχιτεκτονικής 3-επιπέδων με 2 επιπέδων Spine-Leaf. Προσαρμογή από [45].

#### 4.3.3.2 Juniper Networks Contrail

Το Juniper Cloud-Native Contrail Networking (CN2) είναι μια πλατφόρμα δικτύωσης που καθορίζεται από λογισμικό (SDN) που αυτοματοποιεί τη δημιουργία και τη διαχείριση εικονικών δικτύων. Επιτρέπει τη σύνδεση την απομόνωση και την ασφάλιση δεδομένων και φόρτων εργασίας τόσο σε ιδιωτικό όσο και σε δημόσιο cloud. Ενσωματώνει τεχνολογίες τα OpenStack, OpenShift και Kubernetes, το CN2 παρέχει υβριδική ενορχήστρωση SDN και κεντρικό έλεγχο εικονικής εναλλαγής, δρομολόγησης, ασφάλειας, εξισορρόπησης φορτίου και πολλά άλλα. Είναι κατάλληλο για περιβάλλοντα πολλαπλών συστάδων που μοιράζονται ομάδες, εφαρμογές και στάδια μηχανικής. Κλιμακώνεται σε εικονικά δίκτυα και σε περιπτώσεις υπολογισμού για τη διαχείριση συστάδων εικονικών δικτύων χιλιάδων κόμβων στα πιο απαιτητικά περιβάλλοντα. [26]

Βασικά χαρακτηριστικά:

1. Υβριδικό SDN για Kubernetes, OpenShift και OpenStack
2. Προηγμένη δρομολόγηση, προαιρετικές επικαλύψεις και τοπολογίες εικονικού δικτύου
3. Ασφάλεια και αλυσίδα υπηρεσιών
4. Διαχείριση / ομοσπονδία πολλαπλών συστάδων
5. Υπηρεσίες αγωγών GitOps και CI/CD



6. Οπτικοποίηση, παρακολούθηση και διαχείριση σε πραγματικό χρόνο που βασίζεται σε GUI Ιστού

### 4.3.3.3 VMware NSX

Η VMware, Inc. είναι εταιρεία που παρέχει λογισμικό εικονικοποίησης. Ιδρύθηκε το 1998 από τους Diane Greene, Mendel Rosenblum, Scott Devine, Edward Wang και τον Édouard Bugnion και η έδρα της είναι στο Palo Alto, California, ΗΠΑ. Της οποίας ο κύριος μέτοχος είναι η EMC Corporation. Τα προϊόντα της VMware είναι σχεδιασμένα για λειτουργικά συστήματα Microsoft Windows, Linux και Mac OS X. Είναι μια εταιρεία με προσωπικό περίπου 40.000 άτομα για το 2023 και θεωρείται μία από τις κορυφαίες αν όχι η κορυφαία εταιρεία εικονικών δικτύων.[49]

Το VMware NSX είναι μια λύση δικτύωσης SDN που παρέχει προηγμένες δυνατότητες αυτοματισμού δικτύου και ασφάλειας για εικονικά κέντρα δεδομένων και περιβάλλοντα cloud. Το NSX έχει σχεδιαστεί για να λειτουργεί με το λογισμικό εικονικοποίησης της VMware, συμπεριλαμβανομένων των vSphere, vCloud Director και VMware Cloud σε AWS. Ακολουθούν ορισμένα βασικά χαρακτηριστικά και πλεονεκτήματα του VMware NSX:

1. **Εικονικοποίηση δικτύου** - Το NSX παρέχει εικονικοποίηση δικτύου που επιτρέπει σε πολλά εικονικά δίκτυα να εκτελούνται στην ίδια φυσική υποδομή δικτύου. Αυτό επιτρέπει μεγαλύτερη ευελιξία στη διαμόρφωση του δικτύου και επιτρέπει την πολλαπλή μίσθωση σε περιβάλλοντα cloud.
2. **Microsegmentation** - Το NSX επιτρέπει την τμηματοποίηση δικτύου σε επίπεδο εφαρμογής, η οποία παρέχει λεπτομερή έλεγχο της κίνησης του δικτύου και βελτιώνει την ασφάλεια του δικτύου. Αυτή η δυνατότητα είναι ιδιαίτερα σημαντική σε περιβάλλοντα cloud με πολλούς χρήστες, όπου η απομόνωση μεταξύ διαφορετικών χρηστών είναι κρίσιμη.
3. **Προηγμένες υπηρεσίες δικτύου** - Το NSX παρέχει προηγμένες υπηρεσίες δικτύου, όπως εξισορρόπηση φορτίου, VPN, τείχος προστασίας και αποτροπή εισβολής. Αυτές οι υπηρεσίες μπορούν εύκολα να ενσωματωθούν σε εικονικά κέντρα δεδομένων και περιβάλλοντα cloud.
4. **Αυτοματισμός και ενορχήστρωση** - Το NSX παρέχει μια προσέγγιση βασισμένη σε πολιτικές για τη διαχείριση δικτύου που απλοποιεί τη διαμόρφωση του δικτύου και μειώνει τον κίνδυνο σφαλμάτων. Αυτή η δυνατότητα είναι ιδιαίτερα σημαντική

σε πολύπλοκα περιβάλλοντα δικτύου όπου η μη αυτόματη ρύθμιση παραμέτρων δικτύου μπορεί να είναι χρονοβόρα και επιρρεπής σε σφάλματα.

5. **Υποστήριξη πολλαπλού νέφους** - Το NSX έχει σχεδιαστεί για να λειτουργεί με περιβάλλοντα πολλαπλών νέφους, συμπεριλαμβανομένων δημόσιων, ιδιωτικών και υβριδικών νέφους. Αυτό επιτρέπει στους διαχειριστές δικτύου να διαχειρίζονται τις υπηρεσίες δικτύου σε πολλά περιβάλλοντα cloud από μια ενιαία κονσόλα διαχείρισης.

Συνοπτικά, το VMware NSX παρέχει προηγμένες δυνατότητες αυτοματισμού δικτύου και ασφάλειας για εικονικά κέντρα δεδομένων και περιβάλλοντα cloud. Το NSX επιτρέπει εικονικοποίηση δικτύου, μικροτμηματοποίηση, προηγμένες υπηρεσίες δικτύου, αυτοματισμό και ενορχήστρωση και υποστήριξη πολλαπλών νέφους, καθιστώντας το ιδανική λύση για οργανισμούς που απαιτούν μια ασφαλή και επεκτάσιμη δικτυακή υποδομή για τα εικονικά κέντρα δεδομένων και τα περιβάλλοντα cloud τους. [49]

#### **4.3.3.4 HP-IMC**

Το IMC είναι ένας εμπορικός ελεγκτής SDN που έχει σχεδιαστεί για να λειτουργεί με το υλικό δικτύωσης της HP. Παρέχει προηγμένες δυνατότητες αυτοματισμού και ασφάλειας δικτύου και μπορεί να κλιμακωθεί για να υποστηρίξει δίκτυα μεγάλης κλίμακας. Η IMC προσφέρει επίσης μια ενοποιημένη πλατφόρμα διαχείρισης τόσο για φυσικά όσο και για εικονικά δίκτυα. Με την πάροδο των ετών έχει συνεργαστεί και με άλλες εταιρείες εικονοποίησης όπως η VMware που είδαμε έτσι ώστε να μπορεί να προσφέρει ακόμα περισσότερα σε μεγαλύτερο πλήθος συσκευών. [31]

#### **4.3.3.5 NEC**

Η NEC Corporation είναι μια ιαπωνική πολυεθνική εταιρεία τεχνολογίας πληροφοριών και ηλεκτρονικών ειδών, με έδρα το Minato του Τόκιο. Η εταιρεία ήταν γνωστή ως Nippon Electric Company, Limited, πριν μετονομαστεί το 1983 ως NEC. Απασχολεί περισσότερους από 100.000 εργαζόμενους. Το NOE της NEC απλοποιεί την καθημερινή αυτοματοποιημένη και εικονική διαχείριση δικτύων παρέχοντας ένα ασφαλές, ευέλικτο και κεντρικά διαμορφώσιμο δίκτυο που γίνεται δυνατό μέσω μιας εύχρηστης διεπαφής διαχείρισης και διαμόρφωσης μεταφοράς και απόθεσης. Τα διαθέσιμα switch Gigabit PoE/non-PoE με δυνατότητα QX εταιρικού επιπέδου SDN υποστηρίζουν παραδοσιακές λειτουργίες μεταγωγής και επίσης δικτύωση που ορίζεται από λογισμικό .

- Τα switch Ethernet της σειράς QX της NEC είναι μεταγωγείς εταιρικής ποιότητας, gigabit και 10 gigabit Ethernet με δυνατότητα PoE+ (δυνατοί να παρέχουν εξ αποστάσεως ρεύμα σε τερματικά σημεία όπως τηλέφωνα IP ή σημεία πρόσβασης).
- Αποδεδειγμένη ποιότητα ανάλογη του εύρους σύνδεσης
- Υποστήριξη τόσο τις παραδοσιακές λειτουργίες μεταγωγής όσο και τις προηγμένες λειτουργίες SDN
- Ο ελεγκτής SDN της NEC (Network Operation Engine=NOE) μπορεί προαιρετικά να χρησιμοποιηθεί για την παροχή βελτιωμένης ασφάλειας, ευκολίας διαμόρφωσης και κεντρικής διαχείρισης του δικτύου.
- Εικονοποίηση: Διαιρεί ένα φυσικό δίκτυο Ethernet σε πολλαπλά ξεχωριστά εικονικά δίκτυα. Αυτό εξαλείφει την ανάγκη για δαπανηρά πολλαπλά φυσικά δίκτυα.
- Οπτικοποίηση: Κεντρική, διαισθητική διαχείριση
- Αυτοματισμός: Εύκολη ρύθμιση παραμέτρων και διαχείριση δικτύου με βάση απλούς, αφηρημένους κανόνες
- Η λύση NOE + QX έχει ανταγωνιστική τιμή. Η λύση διακρίνεται επίσης σε ανώτερες δυνατότητες εικονοποίησης δικτύου, ευκολία στη χρήση και ευκολία εγκατάστασης (χρησιμοποιώντας τη διαμόρφωση παρτίδας). [35]

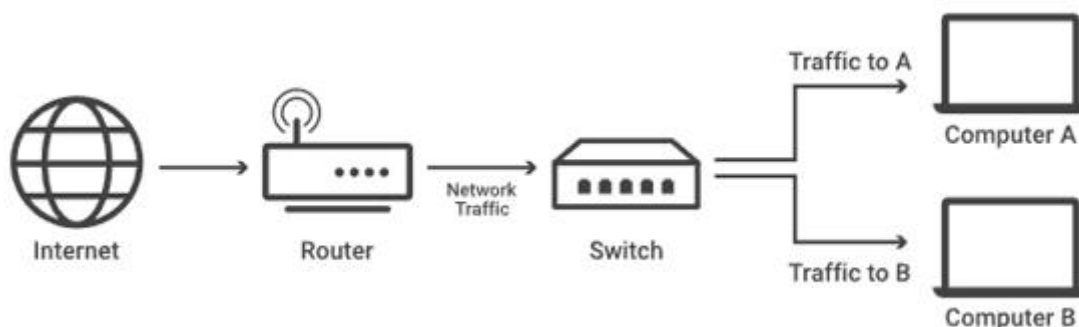
<b>ΕΜΠΟΡΙΚΟΙ ΕΛΕΓΚΤΕΣ</b>	<b>ΒΑΣΙΚΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ</b>	<b>ΒΑΣΙΚΑ ΜΕΙΟΝΕΚΤΗΜΑΤΑ</b>
Cisco ACI	Μεγάλη κλιμάκωση , κεντρική διαχείριση , αποδεδειγμένα ενισχυμένη ασφάλεια	Το μεγαλύτερο κόστος, τη μικρότερη ανεκτικότητα σε ενσωμάτωση εφαρμογών τρίτων , πολυπλοκότητα
Juniper Networks Contrail	Διαχείριση δικτύου , δημιουργία μεγάλης κλίμακας εικονικών δικτύων, πλήρης ενσωμάτωση με προϊόντα και υπηρεσίες Juniper	Κόστος , περιορισμένη ασφάλεια , ελλιπή υποστήριξη
VMware NSX	Διαχείριση δικτύου , δημιουργία μεγάλης κλίμακας εικονικών δικτύων, πλήρης ενσωμάτωση με προϊόντα και υπηρεσίες VMware	Κόστος , περιορισμένες δυνατότητες , περιορισμός ως προς την υποστήριξη προϊόντων
HP-IMC	Διαχείριση δικτύου , δημιουργία μεγάλης κλίμακας εικονικών δικτύων, προγραμματιστικότητα	Περιορισμένη προσαρμοστικότητα , κόστος , περιορισμός χρήσης καθώς υποστηρίζει κυρίως προϊόντα της εταιρείας HP.

NEC	Απεικόνιση δικτύου , υποστήριξη πολλών δικτυακών εταιρειών , ευκολία χρήσης	Μεγάλο κόστος , πολυπλοκότητα , περιορισμένες δυνατότητες
-----	---	--

Πίνακας 4.6 Σύγκριση εμπορικών ελεγκτών

## 4.4 Switch

Ένας μεταγωγέας δικτύου συνδέει συσκευές μέσα σε ένα δίκτυο (συνγά ένα τοπικό δίκτυο-LAN) και προωθεί πακέτα δεδομένων προς και από αυτές τις συσκευές. Σε αντίθεση με έναν δρομολογητή, ο μεταγωγέας στέλνει δεδομένα μόνο στη μεμονωμένη συσκευή για την οποία προορίζεται (η οποία μπορεί να είναι ένας άλλος μεταγωγέας, ένας δρομολογητής ή ένας υπολογιστής χρήστη), όχι σε δίκτυα πολλαπλών συσκευών.



Εικόνα 4.3 Κίνηση δεδομένων σε ένα δίκτυο με τη βοήθεια του μεταγωγέα-switch.

Όμως ποια είναι η βασική διαφορά μεταξύ ενός μεταγωγέα και ενός δρομολογητή; Οι δρομολογητές επιλέγουν διαδρομές για τα πακέτα δεδομένων να διασχίζουν δίκτυα και να φτάσουν στους προορισμούς τους. Το κάνουν αυτό συνδέοντας με διαφορετικά δίκτυα και προωθώντας δεδομένα από δίκτυο σε δίκτυο (συμπεριλαμβανομένων LAN) δικτύων ευρείας περιοχής (WAN) ή αυτόνομων συστημάτων, τα οποία είναι τα μεγάλα δίκτυα που συνθέτουν το Διαδίκτυο.

Στην πράξη, αυτό σημαίνει ότι οι δρομολογητές είναι απαραίτητοι για μια σύνδεση στο Διαδίκτυο, ενώ οι διακόπτες χρησιμοποιούνται μόνο για τη διασύνδεση συσκευών. Τα σπίτια και τα μικρά γραφεία χρειάζονται δρομολογητές για πρόσβαση στο Διαδίκτυο, αλλά τα περισσότερα δεν χρειάζονται μεταγωγέα δικτύου, εκτός εάν απαιτούν μεγάλο αριθμό θυρών Ethernet\*. Ωστόσο, τα μεγάλα γραφεία, τα δίκτυα και τα κέντρα δεδομένων με δεκάδες ή εκατοντάδες υπολογιστές συνήθως απαιτούν διακόπτες. Επιπλέον άξιο αναφοράς

είναι ότι το Ethernet είναι ένα πρωτόκολλο επιπέδου 2 για την αποστολή δεδομένων μεταξύ συσκευών. Σε αντίθεση με το WiFi, το Ethernet απαιτεί φυσική σύνδεση μέσω καλωδίου Ethernet.

Οι μεταγωγείς δικτύου μπορούν να λειτουργούν είτε στο επίπεδο OSI 2 (το επίπεδο σύνδεσης δεδομένων) είτε στο επίπεδο 3 (το επίπεδο δικτύου). Το επίπεδο 2 αλλάζει δεδομένα προς τα εμπρός με βάση τη διεύθυνση MAC προορισμού (δείτε παρακάτω για ορισμό), ενώ το επίπεδο 3 αλλάζει δεδομένα με βάση τη διεύθυνση IP προορισμού. Μερικοί διακόπτες μπορούν να κάνουν και τα δύο.

Οι περισσότεροι μεταγωγείς ωστόσο, είναι επιπέδου 2. Οι διακόπτες επιπέδου 2 συνδέονται συχνότερα με τις συσκευές στα δίκτυά τους χρησιμοποιώντας καλώδια Ethernet. Τα καλώδια Ethernet είναι φυσικά καλώδια που συνδέονται σε συσκευές μέσω θυρών Ethernet. Τέλος ένας μη διαχειριζόμενος μεταγωγέας δημιουργεί απλώς περισσότερες θύρες Ethernet σε ένα LAN, έτσι ώστε περισσότερες τοπικές συσκευές να έχουν πρόσβαση στο Διαδίκτυο.

Οι μη διαχειριζόμενοι μεταγωγείς μεταβιβάζουν δεδομένα εμπρός και πίσω με βάση τις διευθύνσεις MAC της συσκευής. Αντίθετα ένας διαχειριζόμενος διακόπτης εκπληρώνει την ίδια λειτουργία για πολύ μεγαλύτερα δίκτυα και προσφέρει στους διαχειριστές του δικτύου πολύ περισσότερο έλεγχο σχετικά με τον τρόπο με τον οποίο δίνεται προτεραιότητα στην κυκλοφορία. Επιτρέπουν επίσης στους διαχειριστές να ρυθμίσουν VLAN για να υποδιαιρέσουν περαιτέρω ένα τοπικό δίκτυο σε μικρότερα κομμάτια. [36]

#### **4.4.1 Ο ρόλος του Switch στο SDN**

Στο SDN, ένας μεταγωγέας είναι μια συσκευή δικτύωσης που προωθεί πακέτα δεδομένων μεταξύ διαφορετικών κόμβων δικτύου. Ο ρόλος του μεταγωγέα στο SDN είναι να λαμβάνει οδηγίες από τον ελεγκτή SDN και να τις εκτελεί στο δίκτυο.

Ο μεταγωγέας στο SDN είναι προγραμματιζόμενος και διαμορφώσιμος, επιτρέποντας στους διαχειριστές δικτύου να διαχειρίζονται το δίκτυο κεντρικά μέσω του ελεγκτή SDN. Μπορεί να ελεγχθεί και να ρυθμιστεί χρησιμοποιώντας το πρωτόκολλο OpenFlow, το οποίο επιτρέπει στον ελεγκτή να προσδιορίσει πώς θα προωθηθούν τα πακέτα δεδομένων μέσω του δικτύου. Επιπλέον είναι υπεύθυνος για την εφαρμογή των πολιτικών προώθησης που ορίζονται από τον ελεγκτή SDN. Λαμβάνει πακέτα από συσκευές δικτύου που είναι

συνδεδεμένες σε αυτό, προσδιορίζει τη διεύθυνση προορισμού των πακέτων και τα προωθεί στην κατάλληλη συσκευή δικτύου σύμφωνα με τις πολιτικές προώθησης.

#### **4.4.2 OpenFlow Switch**

Ένας μεταγωγέας OpenFlow είναι ένας τύπος μεταγωγέα που χρησιμοποιείται στο SDN που μπορεί να ελέγχεται και να ρυθμίζεται από έναν ελεγκτή SDN χρησιμοποιώντας το πρωτόκολλο OpenFlow. Το OpenFlow είναι ένα πρωτόκολλο που επιτρέπει στον ελεγκτή SDN να προσδιορίζει τον τρόπο προώθησης της κυκλοφορίας δικτύου μέσω του δικτύου από το διακόπτη OpenFlow. Έχει σχεδιαστεί για να διαχωρίζει το επίπεδο ελέγχου και το επίπεδο δεδομένων ενός δικτύου, επιτρέποντας τον κεντρικό έλεγχο και διαχείριση του δικτύου μέσω του ελεγκτή SDN. Σε ένα δίκτυο OpenFlow, ο ελεγκτής SDN επικοινωνεί με τον διακόπτη OpenFlow για να καθορίσει τη διαδρομή που πρέπει να ακολουθήσει η κυκλοφορία του δικτύου μέσω του δικτύου, αντί να βασίζεται στην παραδοσιακή προσέγγιση καταναμημένου επιπέδου ελέγχου. Έχουν συνήθως έναν πίνακα ροής, ο οποίος είναι ένα σύνολο κανόνων που καθορίζουν τον τρόπο προώθησης της κυκλοφορίας δικτύου μέσω του μεταγωγέα. Ο ελεγκτής SDN μπορεί να προσθέσει, να τροποποιήσει ή να διαγράψει αυτούς τους κανόνες όπως απαιτείται για τον έλεγχο της ροής της κυκλοφορίας του δικτύου μέσω του δικτύου.

Η χρήση αυτών των μεταγωγέων OpenFlow γίνεται πολλούς διαφορετικούς τύπους δικτύων, συμπεριλαμβανομένων δικτύων κέντρων δεδομένων, δικτύων πανεπιστημιούπολης και δικτύων ευρείας περιοχής. Αποτελούν βασικό στοιχείο του SDN, το οποίο γίνεται ολοένα και πιο δημοφιλές λόγω της ικανότητάς του να παρέχει κεντρικό έλεγχο και διαχείριση δικτύου, μεγαλύτερη ευελιξία και βελτιωμένη απόδοση δικτύου.

#### **4.4.3 Εικονικό Switch**

Ένας μεταγωγέας εικονικού δικτύου, γνωστός και ως vSwitch, βασίζεται σε λογισμικό που χρησιμοποιείται για τη σύνδεση εικονικών μηχανών (VM) μεταξύ τους και με το φυσικό δίκτυο. Συνήθως υλοποιείται ως στοιχείο εικονικοποίησης σε έναν hypervisor, ο οποίος είναι το λογισμικό που διαχειρίζεται πολλαπλά VM που εκτελούνται σε έναν μόνο φυσικό κεντρικό υπολογιστή. Λειτουργεί στο επίπεδο σύνδεσης δεδομένων (Επίπεδο 2) της στοίβας δικτύου, ακριβώς όπως ένας φυσικός μεταγωγέας δικτύου. Παρέχει την ίδια λειτουργικότητα με έναν φυσικό, όπως η προσθήκη ετικετών VLAN, QoS και το φιλτράρισμα της κυκλοφορίας, αλλά εφαρμόζεται σε λογισμικό και όχι σε υλικό.

Όταν δημιουργείται μια εικονική μηχανή σε έναν φυσικό κεντρικό υπολογιστή, συνδέεται με έναν εικονικό προσαρμογέα δικτύου, ο οποίος με τη σειρά του συνδέεται με έναν εικονικό μεταγωγέα. Ο εικονικός μεταγωγέας παρέχει στη συνέχεια συνδεσιμότητα με άλλες εικονικές μηχανές που λειτουργούν στον ίδιο κεντρικό υπολογιστή και στο φυσικό δίκτυο.

Οι εικονικοί μεταγωγείς μπορούν να διαχειρίζονται και να ρυθμίζονται μέσω μιας διεπαφής διαχείρισης που βασίζεται στο web ή μέσω εργαλείων γραμμής εντολών. Μπορούν επίσης να ενσωματωθούν με άλλα στοιχεία εικονικοποίησης, όπως εικονικούς δρομολογητές και τείχη προστασίας, για να παρέχουν μια πλήρη υποδομή εικονικού δικτύου.

Συνολικά, οι μεταγωγείς εικονικών δικτύων αποτελούν βασικό συστατικό της τεχνολογίας εικονικοποίησης και χρησιμοποιούνται εκτενώς σε κέντρα δεδομένων και περιβάλλοντα υπολογιστικού νέφους για να παρέχουν ευέλικτη, επεκτάσιμη και αποτελεσματική συνδεσιμότητα δικτύου για εικονικές μηχανές.

#### **4.4.4 Φυσικό Switch**

Ένας φυσικός μεταγωγέας είναι μια συσκευή δικτύωσης που συνδέει συσκευές σε ένα δίκτυο υπολογιστών προωθώντας πακέτα δεδομένων μεταξύ τους. Είναι μια συσκευή που βασίζεται σε υλικό που λειτουργεί στο επίπεδο σύνδεσης δεδομένων (Επίπεδο 2) και μερικές φορές στο επίπεδο δικτύου (Επίπεδο 3) της στοίβας δικτύου. Οι φυσικοί μεταγωγείς χρησιμοποιούνται για τη δημιουργία ενός τοπικού δικτύου (LAN), επιτρέποντας στις συσκευές του δικτύου να επικοινωνούν μεταξύ τους. Χρησιμοποιούνται για τη σύνδεση υπολογιστών, διακομιστών, εκτυπωτών και άλλων συσκευών δικτύου μεταξύ τους και για τη σύνδεση του LAN σε άλλα δίκτυα, όπως το Διαδίκτυο. Επιπλέον έχουν ένα σύνολο θυρών, καθεμία από τις οποίες μπορεί να συνδεθεί σε μια συσκευή δικτύου. Όταν ένα πακέτο δεδομένων φτάνει σε μια θύρα μεταγωγής, ο μεταγωγέας διαβάζει τη διεύθυνση MAC προορισμού στην κεφαλίδα του πακέτου και προωθεί το πακέτο στη θύρα που είναι συνδεδεμένη στη συσκευή με αυτήν τη διεύθυνση MAC. Εκτός από τη βασική λειτουργία προώθησης παρέχουν επίσης δυνατότητες όπως VLAN, συνάθροιση συνδέσμων, κατοπτρισμό θυρών και ποιότητα υπηρεσίας (QoS) για τη βελτίωση της απόδοσης και της ασφάλειας του δικτύου. Διατίθενται σε διαφορετικά μεγέθη, που κυμαίνονται από μικροί με λίγες μόνο θύρες έως μεγάλους εταιρικούς μεταγωγείς με εκατοντάδες ή χιλιάδες θύρες. Μπορούν να διαχειρίζονται και να ρυθμίζονται μέσω μιας διεπαφής γραμμής εντολών (CLI), μιας διεπαφής που βασίζεται στο web ή μέσω άλλων πρωτοκόλλων διαχείρισης όπως

το SNMP (Simple Network Management Protocol). Συνολικά, οι φυσικοί μεταγωγείς αποτελούν ουσιαστικό στοιχείο οποιουδήποτε δικτύου υπολογιστών, παρέχοντας αξιόπιστη και αποτελεσματική συνδεσιμότητα μεταξύ συσκευών και επιτρέποντας τη μετάδοση δεδομένων σε όλο το δίκτυο.

#### 4.4.5 Edge Switch

Ο μεταγωγέας άκρων είναι ένας τύπος διακόπτη που χρησιμοποιείται συνήθως σε δίκτυα υπολογιστών για τη σύνδεση συσκευών τελικού χρήστη, όπως επιτραπέζιους υπολογιστές, φορητούς υπολογιστές, διακομιστές και άλλες συσκευές δικτύου, στο δίκτυο. Οι μεταγωγείς άκρων βρίσκονται συνήθως στην άκρη ενός δικτύου, συνδέοντας συσκευές τελικού χρήστη με το υπόλοιπο δίκτυο. Λειτουργούν στο επίπεδο σύνδεσης δεδομένων (Επίπεδο 2) της στοίβας δικτύου και μερικές φορές στο επίπεδο δικτύου (Επίπεδο 3), ανάλογα με την αρχιτεκτονική και τις απαιτήσεις του δικτύου. Οι μεταγωγείς άκρων διατίθενται σε διαφορετικά μεγέθη και διαμορφώσεις, που κυμαίνονται από μικρούς επιτραπέζιους με λίγες θύρες έως μεγάλους διακόπτες με δυνατότητα τοποθέτησης σε rack με πολλαπλές θύρες 10 Gigabit Ethernet (GbE) ή ακόμα και 40/100 GbE. Μπορεί επίσης να περιλαμβάνουν λειτουργίες όπως Power over Ethernet (PoE), το οποίο παρέχει τροφοδοσία σε συσκευές όπως τηλέφωνα IP και σημεία ασύρματης πρόσβασης μέσω καλωδίων Ethernet. Οι μεταγωγείς άκρων παρέχουν συχνά βασικές λειτουργίες μεταγωγής, όπως VLAN που βασίζονται σε θύρες, συνάθροιση συνδέσμων και βασικές λειτουργίες ασφαλείας όπως το φιλτράρισμα διευθύνσεων MAC και η ασφάλεια θυρών. Ενδέχεται επίσης να υποστηρίζουν προηγμένες λειτουργίες όπως η Ποιότητα υπηρεσίας (QoS), ο έλεγχος πρόσβασης στο δίκτυο και τα πρωτόκολλα δυναμικής δρομολόγησης. Συνολικά, οι μεταγωγείς άκρων αποτελούν σημαντικό στοιχείο οποιασδήποτε δικτυακής υποδομής, παρέχοντας συνδεσιμότητα και πρόσβαση σε συσκευές τελικού χρήστη, ενώ παράλληλα επιτρέπουν την αποτελεσματική και ασφαλή μετάδοση δεδομένων σε όλο το δίκτυο.

ΤΥΠΟΣ SWITCH	ΒΑΣΙΚΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ	ΒΑΣΙΚΑ ΜΕΙΟΝΕΚΤΗΜΑΤΑ
OPENFLOW	Ευελιξία προσαρμοστικότητας , εκτενής και αξιόπιστος έλεγχος	Υψηλό κόστος , πολυπλοκότητα
ΕΙΚΟΝΙΚΟ	Χαμηλό κόστος , εύκολη χρήση	Επιδόσεις , επεκτασιμότητα
ΦΥΣΙΚΟ	Επιδόσεις , επεκτασιμότητα	Κόστος , συντήρηση και αξιοπιστία
EDGE	Ασφάλεια , ευκολία διαχείρισης	Περιορισμένο εύρος λειτουργίας , πολυπλοκότητα

Πίνακας 4.7 Σύγκριση ειδών Switch.



## **5. ΠΡΩΤΟΚΟΛΛΑ ΛΕΙΤΟΥΡΓΙΑΣ SDN**

Το πρωτόκολλο SDN είναι ένα σύνολο προτύπων και κανόνων που καθορίζουν τον τρόπο με τον οποίο οι ελεγκτές και οι μεταγωγείς SDN επικοινωνούν μεταξύ τους. Στην ουσία ένα πρωτόκολλο επιτρέπει στον ελεγκτή SDN να διαμορφώνει τη συμπεριφορά του μεταγωγέα, όπως να προσδιορίζει ποια πακέτα πρέπει να προωθούνται σε ποιες θύρες και να ορίζει παραμέτρους ποιότητας υπηρεσίας (QoS) για διαφορετικούς τύπους κίνησης. Το πιο συχνά χρησιμοποιούμενο πρωτόκολλο SDN είναι το OpenFlow.

### **5.1 Πρωτόκολλο OpenFlow**

Όπως αναφέρθηκε το OpenFlow είναι το πιο διαδεδομένο πρωτόκολλο SDN και καθορίζει τη ροή μεταξύ μεταγωγέα και ελεγκτή. Επιτρέπει στον ελεγκτή να διαχειρίζεται την προώθηση της κίνησης μεταξύ διαφορετικών συσκευών δικτύου ελέγχοντας τους πίνακες ροής των μεταγωγέων. Αυτό το πρωτόκολλο αναπτύχθηκε για πρώτη φορά από ερευνητές στο Πανεπιστήμιο του Στάνφορντ το 2008 και υιοθετήθηκε για πρώτη φορά από την Google στο δίκτυο κορμού τους το 2011-2012. Η διαχείριση του γίνεται πλέον από το Open Networking Foundation (ONF). Η τελευταία έκδοση που χρησιμοποιείται στη βιομηχανία μαζικά είναι η V1.5 ενώ παράλληλα τελειοποιείται η έκδοση V2.0. Σημειώνεται επίσης ότι συχνά αναφέρεται και ως OFDP δηλαδή το πρωτόκολλο δημιουργίας τοπολογιών OpenFlow διότι πρακτικά είτε αναφερθεί ως OpenFlow είτε ως OFDP εννοείται αυτόματα ακριβώς η ίδια λειτουργία.

Το OpenFlow είναι το τυπικό πρωτόκολλο νότιας διεπαφής που χρησιμοποιείται μεταξύ του ελεγκτή SDN και του μεταγωγέα. Ο ελεγκτής SDN παίρνει τις πληροφορίες από τις εφαρμογές και τις μετατρέπει σε καταχωρήσεις ροής, οι οποίες τροφοδοτούνται στο μεταγωγέα μέσω OpenFlow. Μπορεί επίσης να χρησιμοποιηθεί για την παρακολούθηση στατιστικών μεταγωγών και θυρών στη διαχείριση δικτύου.

Αξίζει να σημειωθεί πως το πρωτόκολλο OpenFlow εγκαθίσταται μόνο μεταξύ ενός ελεγκτή και του μεταγωγέα. Δεν επηρεάζει το υπόλοιπο δίκτυο. Εάν μια σύλληψη πακέτων επρόκειτο να ληφθεί μεταξύ δύο μεταγωγέων σε ένα δίκτυο, και οι δύο συνδεδεμένοι στον ελεγκτή μέσω άλλης θύρας, η σύλληψη πακέτων δεν θα αποκάλυπτε κανένα μήνυμα OF

μεταξύ των μεταγωγέων. Είναι αυστηρά για χρήση μεταξύ ενός μεταγωγέα και του ελεγκτή. Το υπόλοιπο δίκτυο δεν επηρεάζεται.[27].

### 5.1.1 Η λειτουργία του OpenFlow

Το πρωτόκολλο OpenFlow λειτουργεί στο πρωτόκολλο TCP. Το τυπικό πρωτόκολλο είναι το TCP 6633 για το OF V1.0 και το 6653 για το OF V1.3+. Πρέπει να υπάρχει σύνδεση IP μεταξύ του ελεγκτή και των διακοπών για να δημιουργηθεί μια σύνδεση OF. Το κανάλι OF σχηματίζεται μόνο μετά από μια επιτυχημένη τριπλή χειραψία TCP.

- Ο μεταγωγέας στέλνει ένα πακέτο "HELLO" για να το εισαγάγει στον ελεγκτή για να ξεκινήσει η επικοινωνία καναλιού OF. Ο διακόπτης στέλνει επίσης πληροφορίες όπως η υψηλότερη έκδοση του OF που υποστηρίζει. Ο ελεγκτής απαντά στο μήνυμα hello με την υψηλότερη υποστηριζόμενη έκδοση OF. Στη συνέχεια, ο μεταγωγέας διαπραγματεύεται το υψηλότερο επίπεδο της έκδοσης OpenFlow που υποστηρίζουν και οι δύο.
- Μόλις γίνει η διαπραγμάτευση της έκδοσης, ο ελεγκτής στέλνει ένα μήνυμα "FEATURE\_REQUEST". Αυτό το μήνυμα ουσιαστικά ζητά από τον διακόπτη τις υποστηριζόμενες δυνατότητες OF του, όπως τον αριθμό των υποστηριζόμενων πινάκων ροής, των υποστηριζόμενων ενεργειών κ.λπ. Ο μεταγωγέας απαντά σε αυτό με ένα μήνυμα "FEATURE\_REPLY" που αναφέρει όλες τις δυνατότητές του μαζί με το μοναδικό αναγνωριστικό ή το αναγνωριστικό διαδρομής δεδομένων (DPID) .

Μετά από αυτό, μπορούμε να πούμε ότι το κανάλι OpenFlow έχει εγκατασταθεί με επιτυχία μεταξύ του μεταγωγέα και του ελεγκτή. Η σύνδεση μεταξύ του ελεγκτή και του μεταγωγέα είναι απαραίτητη καθώς είναι ο μόνος τρόπος επικοινωνίας ενός διακόπτη με έναν ελεγκτή.

Για να διασφαλιστεί αυτή η σύνδεση, μπορεί επίσης να χρησιμοποιηθεί ένα πρωτόκολλο όπως το TLS αντί για μια σύνδεση TCP. Εδώ, ο ελεγκτής και ο διακόπτης πρέπει να διαθέτουν τα κατάλληλα πιστοποιητικά και κλειδιά για μια επιτυχημένη σύνδεση TLS. Αυτό αποτρέπει την κατασκοπεία στο κανάλι OF.[27]

### 5.1.2 Πίνακες OpenFlow και καταχωρήσεις ροής

Οι πίνακες ροής είναι σαν τον πίνακα MAC/CAM ενός παραδοσιακού μεταγωγέα που αποθηκεύει τις διευθύνσεις υλικού των κεντρικών υπολογιστών. Οι πίνακες ροής αποθηκεύουν καταχωρήσεις ροής ή ροές που λένε στον μεταγωγέα SDN τι να κάνει με ένα πακέτο όταν πρόκειται για μια εισερχόμενη θύρα.

Ο μεταγωγέας θα αντιστοιχίσει συγκεκριμένες παραμέτρους όπως:

- η διεύθυνση IP
- ο αριθμός θύρας
- η διεύθυνση MAC
- το αναγνωριστικό VLAN

Θα επιλέξει την καταχώριση ροής που ταιριάζει καλύτερα από τον πίνακα και θα εκτελέσει την ενέργεια που σχετίζεται με αυτήν την καταχώριση. Οι ενέργειες θα μπορούσαν να είναι η απόρριψη του πακέτου, η προώθηση του σε διαφορετική θύρα, η κατάκλιση του πακέτου ή η αποστολή του στον ελεγκτή για περαιτέρω επιθεώρηση.

Εάν ένας μεταγωγέας δεν έχει καταχώριση για ένα πακέτο, ο μεταγωγέας μπορεί να έχει μια προεπιλεγμένη καταχώριση ή καταχώριση "TABLE\_MISS". Αυτή η καταχώριση έχει τη χαμηλότερη προτεραιότητα και οι ενέργειες μπορεί να είναι είτε η απόρριψη του πακέτου είτε η αποστολή του στον ελεγκτή.

Όταν ο ελεγκτής λαμβάνει αυτό το είδος πακέτου από ένα μεταγωγέα, το στέλνει στην εφαρμογή που εκτελείται στο επίπεδο εφαρμογής, η οποία επεξεργάζεται το πακέτο και ενημερώνει τον ελεγκτή εάν χρειάζεται να εισαχθεί μια νέα καταχώριση ροής στον πίνακα ροής του μεταγωγέα. Εάν συμβαίνει αυτό, ο ελεγκτής θα εισαγάγει μια καταχώριση ροής στον μεταγωγέα. Το επόμενο πακέτο του ίδιου είδους θα αντιμετωπιστεί από τον μεταγωγέα στο επίπεδο δεδομένων καθώς έχει ήδη μια καταχώριση και θα γίνουν οι κατάλληλες ενέργειες. Αυτό βελτιώνει την αποτελεσματικότητα του δικτύου κατά έναν τεράστιο παράγοντα. Τέλος μπορούμε να πούμε ότι υπάρχουν πολλά πλεονεκτήματα του OpenFlow στο SDN .[27]

1. Το SDN επιτρέπει τον διαχωρισμό του επιπέδου ελέγχου και των δεδομένων, πράγμα που σημαίνει ότι οι μεταγωγείς μπορούν να χρησιμοποιήσουν όλους τους πόρους υλικού τους για την απλή προώθηση δεδομένων αντί για υπολογιστικές διαδρομές.

2. Το OpenFlow παρέχει έναν εύκολο τρόπο επικοινωνίας μεταξύ ελεγκτή και μεταγωγέα, που υλοποιείται εύκολα σε ένα υπάρχον δίκτυο.
3. Οι περισσότερες τρέχουσες συσκευές υποστηρίζουν το OpenFlow, δεν είναι ενεργοποιημένο από προεπιλογή, αλλά μπορούμε εύκολα να το ενεργοποιήσουμε και να το χρησιμοποιήσουμε για μετάβαση στο SDN.
4. Παρέχει ασφάλεια με σύνδεση TLS για την αποφυγή κατασκοπείας και επιθέσεων DoS στον ελεγκτή ή/και στο δίκτυο.
5. Το OpenFlow δεν αλλάζει τη διαμόρφωση ενός μεταγωγέα. Απλώς ενημερώνει τους πίνακες ροής, οι οποίοι καθορίζουν τη διαδρομή για ένα πακέτο.

## 5.2 Πρωτόκολλο NetConf

Το NetConf είναι ένα πρωτόκολλο που χρησιμοποιείται στο SDN για τη διαχείριση συσκευών δικτύου, όπως δρομολογητές και μεταγωγείς, παρέχοντας έναν τυπικό τρόπο διαμόρφωσης, παρακολούθησης και διαχείρισης αυτών των συσκευών. Είναι ένα πρότυπο IETF και βασίζεται στην κωδικοποίηση δεδομένων XML και στο πρωτόκολλο SSH για ασφαλή επικοινωνία. Η προεπιλεγμένη θύρα TCP που έχει εκχωρηθεί είναι 830. Ο διακομιστής NetConf πρέπει να ακούει για συνδέσεις με το υποσύστημα NetConf σε αυτήν τη θύρα.

Με το NetConf, οι διαχειριστές δικτύου μπορούν να διαμορφώσουν μέσω προγραμματισμού συσκευές δικτύου χρησιμοποιώντας ένα τυπικό σύνολο εντολών, αντί να βασίζονται σε ιδιόκτητες διεπαφές για συγκεκριμένες συσκευές. Αυτό βοηθά στην απλοποίηση της διαχείρισης δικτύου και διευκολύνει την αυτοματοποίηση επαναλαμβανόμενων εργασιών, όπως η ανάπτυξη νέων διαμορφώσεων δικτύου ή ενημερώσεων υλικού αλλά και λογισμικού.

Το NetConf χρησιμοποιεί ένα μοντέλο πελάτη-διακομιστή, με τον πελάτη NetConf να στέλνει αιτήματα στη συσκευή και τον διακομιστή NetConf να αποκρίνεται με ενημερώσεις δεδομένων ή κατάστασης. Το πρωτόκολλο υποστηρίζει μια σειρά λειτουργιών, όπως:

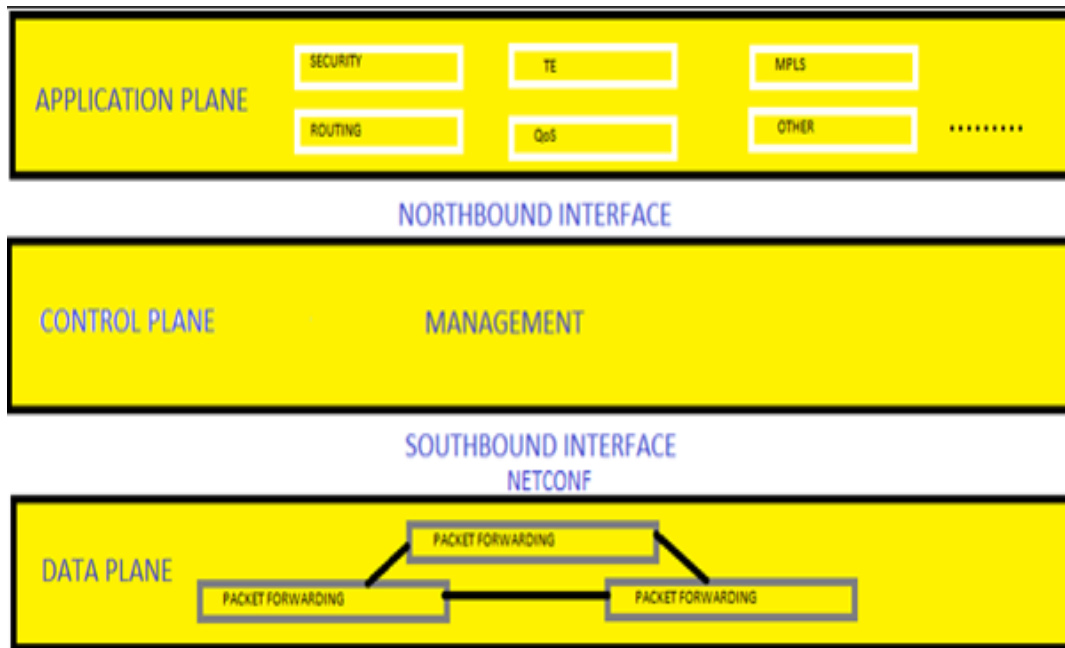
- Ανάκτηση: ανάκτηση συγκεκριμένων δεδομένων ή πληροφοριών διαμόρφωσης από τη συσκευή
- Edit-config: τροποποίηση της διαμόρφωσης της συσκευής

- Δέσμευση: πραγματοποιήστε αλλαγές στη διαμόρφωση της συσκευής
- Κλείδωμα: κλειδώνει τη διαμόρφωση της συσκευής για να εμποδίσετε πολλούς διαχειριστές να κάνουν αντικρουόμενες αλλαγές
- Ξεκλείδωμα: απελευθερώστε το κλείδωμα διαμόρφωσης

Το Netconf χρησιμοποιείται συχνά σε συνδυασμό με τη YANG, μια γλώσσα μοντελοποίησης δεδομένων που επιτρέπει στους διαχειριστές δικτύου να περιγράφουν στοιχεία δικτύου και τις διαμορφώσεις τους με δομημένο και τυποποιημένο τρόπο. Μαζί, το NetConf και το Yang αποτελούν ένα σημαντικό στοιχείο του SDN, επιτρέποντας μεγαλύτερο αυτοματισμό και έλεγχο στη διαχείριση δικτύου.[32]

Στην ουσία το NETCONF με τις λειτουργίες του καταφέρει να πετύχει τα εξής:

- Διευκολύνει τη διαχείριση δεδομένων διαμόρφωσης και τη λειτουργικότητα μεταξύ διαφορετικών συσκευών που χρησιμοποιούν κωδικοποίηση XML για τον ορισμό των μηνυμάτων και του μηχανισμού RPC για την τροποποίηση των δεδομένων διαμόρφωσης.
- Μειώνει τα σφάλματα δικτύου που προκαλούνται από σφάλματα χειροκίνητης διαμόρφωσης.
- Βελτιώνει την αποτελεσματικότητα της αναβάθμισης λογισμικού συστήματος που εκτελείται με χρήση εργαλείων διαμόρφωσης.
- Παρέχει υψηλή επεκτασιμότητα, επιτρέποντας σε διαφορετικούς προμηθευτές να ορίσουν επιπλέον λειτουργίες Netconf.



Εικόνα 5.1 Πρωτόκολλο Netconf.

### 5.2.1 Γλώσσα Yang

Η YANG είναι μια γλώσσα μοντελοποίησης δεδομένων που χρησιμοποιείται στο SDN για να περιγράψει στοιχεία δικτύου και τις διαμορφώσεις τους με δομημένο και τυποποιημένο τρόπο. Αναπτύχθηκε από το IETF για να διευκολύνει τη διαχείριση των συσκευών δικτύου με πιο αυτοματοποιημένο και προγραμματιζόμενο τρόπο. Το Yang μοντελοποιεί τις συσκευές δικτύου ως μια ιεραρχία στοιχείων δεδομένων, παρόμοια με τη δομή ενός δέντρου καταλόγου, με κάθε στοιχείο δεδομένων να ορίζεται από ένα μοναδικό αναγνωριστικό και ένα σύνολο χαρακτηριστικών. Αυτή η δομή διευκολύνει την πρόσβαση μέσω προγραμματισμού και τον χειρισμό των διαμορφώσεων συσκευών δικτύου.

Τα μοντέλα YANG χρησιμοποιούνται σε συνδυασμό με πρωτόκολλα όπως το Netconf για να επιτρέψουν στους διαχειριστές δικτύου να διαμορφώνουν, να παρακολουθούν και να διαχειρίζονται μέσω προγραμματισμού συσκευές δικτύου. Επιτρέπουν τη δημιουργία τυποποιημένων, αναγνώσιμων από μηχανή περιγραφών στοιχείων δικτύου και των διαμορφώσεων τους, οι οποίες μπορούν να χρησιμοποιηθούν για την αυτοματοποίηση των εργασιών διαχείρισης δικτύου. Σε ένα περιβάλλον SDN, τα μοντέλα Yang μπορούν να χρησιμοποιηθούν για να περιγράψουν την τοπολογία του δικτύου, να καθορίσουν πολιτικές δικτύου και να καθορίσουν τη συμπεριφορά των συσκευών δικτύου. Αυτό

επιτρέπει μεγαλύτερο αυτοματισμό και έλεγχο στη διαχείριση δικτύου, διευκολύνοντας την ανάπτυξη νέων διαμορφώσεων δικτύου, την παρακολούθηση της απόδοσης του δικτύου και την αντιμετώπιση προβλημάτων δικτύου. Τα μοντέλα Yang είναι επίσης επεκτάσιμα, επιτρέποντας στους διαχειριστές δικτύου να δημιουργούν προσαρμοσμένα μοντέλα δεδομένων για να υποστηρίξουν τις συγκεκριμένες διαμορφώσεις και απαιτήσεις δικτύου τους. Αυτό διευκολύνει την προσαρμογή της διαχείρισης δικτύου στις συγκεκριμένες ανάγκες ενός οργανισμού, αντί να βασίζεται σε μια προσέγγιση που ταιριάζει σε όλους.

### **5.3 Πρωτόκολλο Open vSwitch Database Management**

Το OVSDB είναι ένα πρωτόκολλο που χρησιμοποιείται σε δίκτυα SDN για τη διαχείριση παρουσιών Open vSwitch, οι οποίες είναι μεταγωγείς που βασίζονται σε λογισμικό και μπορούν να χρησιμοποιηθούν σε περιβάλλον SDN.

Το OVSDB παρέχει έναν τυπικό τρόπο διαμόρφωσης και διαχείρισης παρουσιών Open vSwitch, επιτρέποντας στους διαχειριστές δικτύου να αναπτύσσουν και να διαχειρίζονται συνολικά τους μεταγωγείς με πιο αυτοματοποιημένο και προγραμματικό τρόπο. Ορίζει ένα σύνολο λειτουργιών που μπορούν να χρησιμοποιηθούν για την υποβολή ερωτημάτων και την τροποποίηση της διαμόρφωσης των παρουσιών Open vSwitch, συμπεριλαμβανομένης της δημιουργίας, διαγραφής και τροποποίησης θυρών, διεπαφών και VLAN.

Το OVSDB βασίζεται σε ένα μοντέλο πελάτη-διακομιστή, με τον πελάτη OVSDB να στέλνει αιτήματα στον διακομιστή OVSDB, ο οποίος απαντά με ενημερώσεις δεδομένων ή κατάστασης. Το πρωτόκολλο χρησιμοποιεί κωδικοποίηση δεδομένων JSON και υποστηρίζει ασφαλή επικοινωνία με χρήση TLS.

Σε ένα περιβάλλον SDN, οι περιπτώσεις Open vSwitch μπορούν να χρησιμοποιηθούν για την προώθηση της κυκλοφορίας μεταξύ διαφορετικών συσκευών δικτύου και για να επιτρέψουν στους διαχειριστές δικτύου να διαχειρίζονται τις ροές κυκλοφορίας χρησιμοποιώντας έναν κεντρικό ελεγκτή SDN. Το OVSDB παρέχει έναν τυποποιημένο τρόπο ρύθμισης και διαχείρισης αυτών των μεταγωγέων, διευκολύνοντας την ανάπτυξη και τη διαχείριση δικτύων SDN μεγάλης κλίμακας.

Ανακεφαλαιώνοντας το OVSDB είναι ένα πρωτόκολλο ανοιχτού κώδικα που χρησιμοποιείται ευρέως σε περιβάλλοντα SDN και υποστηρίζεται από μια σειρά από προμηθευτές συσκευών δικτύου και ελεγκτές SDN ανοιχτού κώδικα.[32]

## 5.4 Πρωτόκολλο Border Gateway

Το Border Gateway Protocol (BGP) είναι ένα πρωτόκολλο δρομολόγησης που χρησιμοποιείται συνήθως σε περιβάλλοντα δικτύωσης SDN για την ανταλλαγή πληροφοριών δρομολόγησης μεταξύ διαφορετικών αυτόνομων συστημάτων σε δίκτυα μεγάλης κλίμακας. Το BGP είναι ένα διανυσματικό πρωτόκολλο διαδρομής που χρησιμοποιεί ένα δίκτυο διασυνδεδεμένων αυτόνομων συστημάτων για τη δρομολόγηση της κυκλοφορίας μεταξύ διαφορετικών τμημάτων του δικτύου.

Σε ένα περιβάλλον SDN, το BGP μπορεί να χρησιμοποιηθεί για να διευκολύνει την επικοινωνία μεταξύ διαφορετικών στοιχείων του δικτύου, όπως ο ελεγκτής SDN και οι συσκευές δικτύου, ή μεταξύ διαφορετικών ελεγκτών SDN. Το BGP παρέχει έναν τυποποιημένο τρόπο ανταλλαγής πληροφοριών δρομολόγησης, επιτρέποντας στους διαχειριστές του δικτύου να διαχειρίζονται τις ροές κυκλοφορίας και να βελτιστοποιούν την απόδοση του δικτύου.

Επιπλέον το BGP χρησιμοποιεί ένα ιεραρχικό σύστημα πινάκων δρομολόγησης, με κάθε αυτόνομο σύστημα να διατηρεί τον δικό του πίνακα δρομολόγησης και να ανταλλάσσει ενημερώσεις με γειτονικά αυτόνομα συστήματα. Το πρωτόκολλο υποστηρίζει τόσο εσωτερική όσο και εξωτερική δρομολόγηση, επιτρέποντας πιο αποτελεσματική δρομολόγηση εντός ενός μόνο αυτόνομου συστήματος και μεταξύ διαφορετικών αυτόνομων συστημάτων.

Οι λειτουργίες που υποστηρίζει είναι σημαντικές σε περιβάλλοντα SDN και μερικές από αυτές είναι:

- Η δρομολόγηση βάσει πολιτικής
- Η μηχανική κυκλοφορίας
- Η επιλογή διαδρομής

Αυτές οι δυνατότητες επιτρέπουν στους διαχειριστές δικτύου να προσαρμόζουν τη συμπεριφορά δρομολόγησης και να βελτιστοποιούν την απόδοση του δικτύου με βάση συγκεκριμένες απαιτήσεις δικτύου και μοτίβα κίνησης.



Τέλος το BGP χρησιμοποιείται ευρέως σε περιβάλλοντα SDN μεγάλης κλίμακας, όπως κέντρα δεδομένων και δίκτυα παρόχων υπηρεσιών, και υποστηρίζεται από μια σειρά από προμηθευτές συσκευών δικτύου και ελεγκτές SDN.[22]

## 5.5 Πρωτόκολλο Locator/Identifier Separation

Το πρωτόκολλο διαχωρισμού εντοπισμού/αναγνωριστικού (LISP) είναι ένα πρωτόκολλο που χρησιμοποιείται στο SDN για να διαχωρίσει τη θέση δικτύου μιας συσκευής (στην ουσία τη διεύθυνση IP της) από την ταυτότητα ή το αναγνωριστικό της. Το LISP παρέχει έναν τρόπο εκχώρησης πολλαπλών διευθύνσεων IP σε μια συσκευή και επιτρέπει τη δρομολόγηση της κυκλοφορίας με βάση την ταυτότητα της συσκευής και όχι τη φυσική της θέση.

Σε ένα περιβάλλον SDN, το LISP μπορεί να χρησιμοποιηθεί για να απλοποιήσει τη διαχείριση του δικτύου και να επιτρέψει την πιο αποτελεσματική δρομολόγηση της κυκλοφορίας μεταξύ διαφορετικών συσκευών. Διαχωρίζοντας την ταυτότητα της συσκευής από τη θέση της, το LISP επιτρέπει στους διαχειριστές δικτύου να μετακινούν συσκευές μεταξύ διαφορετικών τοποθεσιών δικτύου χωρίς να αλλάζουν τις διευθύνσεις IP τους, γεγονός που απλοποιεί τη διαχείριση του δικτύου και μειώνει την πιθανότητα σφαλμάτων.

Η λειτουργία του παρουσιάζει ένα ενδιαφέρον καθώς λειτουργεί χρησιμοποιώντας ένα σύστημα αντιστοίχισης που συσχετίζει ένα αναγνωριστικό (όπως ένα όνομα κεντρικού υπολογιστή ή άλλο αναγνωριστικό) με έναν ή περισσότερους εντοπιστές (διευθύνσεις IP). Αυτό το σύστημα χαρτογράφησης χρησιμοποιείται από δρομολογητές με δυνατότητα LISP για την προώθηση της κυκλοφορίας στη σωστή συσκευή, με βάση το αναγνωριστικό της και όχι τη φυσική της θέση.

Συνοψίζοντας το LISP είναι ένα πρωτόκολλο ανοιχτού κώδικα που χρησιμοποιείται ευρέως σε περιβάλλοντα SDN και υποστηρίζεται από μια σειρά από προμηθευτές συσκευών δικτύου και ελεγκτές SDN. Μπορεί να χρησιμοποιηθεί σε συνδυασμό με άλλα πρωτόκολλα SDN, όπως το OpenFlow και το Netconf για να καταστεί δυνατή η πιο αποτελεσματική και ευέλικτη διαχείριση δικτύου.[44]

## 5.6 Πρωτόκολλο Simple Network Management

Το Simple Network Management Protocol (SNMP) είναι ένα πρωτόκολλο που χρησιμοποιείται στη δικτύωση που ορίζεται από λογισμικό (SDN) για την παρακολούθηση και τη διαχείριση συσκευών δικτύου. Το SNMP παρέχει έναν τυποποιημένο τρόπο για τους διαχειριστές δικτύου να παρακολουθούν την απόδοση των συσκευών δικτύου, όπως μεταγωγείς και δρομολογητές, και να τις διαμορφώνουν εξ αποστάσεως.

Σε ένα περιβάλλον SDN, το SNMP μπορεί να χρησιμοποιηθεί για την παρακολούθηση και τη διαχείριση συσκευών δικτύου από έναν κεντρικό ελεγκτή SDN. Το SNMP επιτρέπει στους διαχειριστές δικτύου να παρακολουθούν μια σειρά από μετρήσεις της συσκευής, όπως τη χρήση της CPU, τη χρήση μνήμης και την κυκλοφορία δικτύου, και να λαμβάνουν ειδοποιήσεις όταν προκύπτουν προβλήματα απόδοσης.

Το SNMP βασίζεται σε ένα μοντέλο πελάτη-διακομιστή, με agents SNMP να εκτελούνται στις συσκευές δικτύου και διαχειριστές SNMP να εκτελούνται στον ελεγκτή SDN. Οι agents συλλέγουν δεδομένα απόδοσης και τα στέλνουν στους διαχειριστές, οι οποίοι στη συνέχεια μπορούν να αναλύσουν τα δεδομένα και να προβούν σε ενέργειες για τη βελτίωση της απόδοσης του δικτύου.

Το SNMP είναι ένα ευρέως χρησιμοποιούμενο πρωτόκολλο στη διαχείριση δικτύου και υποστηρίζεται από μια σειρά από προμηθευτές συσκευών δικτύου και ελεγκτές SDN. Μπορεί να χρησιμοποιηθεί σε συνδυασμό με όλα τα παραπάνω πρωτόκολλα SDN που αναφέρθηκαν, με σκοπό την πιο αποτελεσματική και ευέλικτη διαχείριση δικτύου.

## 5.7 Πρωτόκολλο Link Layer Discovery Protocol

Το Πρωτόκολλο Ανακάλυψης Επιπέδου Σύνδεσης (LLDP) είναι ένα πρωτόκολλο επιπέδου 2, ουδέτερο από τον προμηθευτή, που χρησιμοποιείται για την ανακάλυψη και τη διαφήμιση πληροφοριών συσκευών δικτύου σε ένα τοπικό δίκτυο (LAN). Επιτρέπει στις συσκευές δικτύου να ανταλλάσσουν πληροφορίες σχετικά με την ταυτότητα, τις δυνατότητες και τις συνδέσεις τους.

Το LLDP λειτουργεί με την αποστολή και λήψη πλαισίων LLDP, τα οποία είναι πακέτα πολλαπλής διανομής που μεταδίδονται σε κάθε διεπαφή δικτύου. Τα πλαίσια LLDP περιέχουν TLV (Τύπος-Μήκος-Τιμή) που φέρουν συγκεκριμένες πληροφορίες σχετικά

με τη συσκευή μετάδοσης, όπως όνομα συστήματος, περιγραφή θύρας, δυνατότητες συστήματος και διευθύνσεις διαχείρισης.

Τα βασικά χαρακτηριστικά και τα οφέλη του LLDP περιλαμβάνουν:

- Ανακάλυψη συσκευής: Το LLDP δίνει τη δυνατότητα στις συσκευές δικτύου να ανακαλύπτουν γειτονικές συσκευές στο LAN, παρέχοντας πληροφορίες σχετικά με την ταυτότητά τους, όπως τύπο συσκευής, προμηθευτή και μοντέλο.
- Ανακάλυψη τοπολογίας: Ανταλλάσσοντας πληροφορίες LLDP, οι συσκευές μπορούν να συγκεντρώσουν λεπτομέρειες σχετικά με τις συνδέσεις και την τοπολογία του δικτύου, συμπεριλαμβανομένων των γειτονικών συσκευών, των αριθμών θυρών και των ταχυτήτων σύνδεσης.
- Αυτόματη διαμόρφωση: Το LLDP μπορεί να χρησιμοποιηθεί από συστήματα διαχείρισης δικτύου για αυτόματη διαμόρφωση συσκευών δικτύου με βάση τις δυνατότητές τους που ανακαλύφθηκαν. Αυτό απλοποιεί τη ρύθμιση του δικτύου και μειώνει τις προσπάθειες μη αυτόματης διαμόρφωσης.
- Αντιμετώπιση προβλημάτων και παρακολούθηση: Το LLDP διευκολύνει την αντιμετώπιση προβλημάτων δικτύου παρέχοντας ορατότητα στην τοπολογία του δικτύου και τη συνδεσιμότητα της συσκευής. Επιτρέπει στους διαχειριστές να αναγνωρίζουν και να εντοπίζουν συσκευές, να εντοπίζουν αστοχίες συνδέσεων και να παρακολουθούν την κατάσταση της σύνδεσης.

Το LLDP υποστηρίζεται από ένα ευρύ φάσμα συσκευών δικτύου, συμπεριλαμβανομένων μεταγωγέων, δρομολογητών, σημείων ασύρματης πρόσβασης και τηλεφώνων IP. Συχνά χρησιμοποιείται σε συνδυασμό με άλλα πρωτόκολλα δικτύου όπως το SNMP (Simple Network Management Protocol) για να καταστεί δυνατή η ολοκληρωμένη διαχείριση και παρακολούθηση δικτύου.

Είναι σημαντικό να σημειωθεί ότι το LLDP είναι ένα πρωτόκολλο επιπέδου 2 και η λειτουργικότητά του περιορίζεται στο τμήμα τοπικού δικτύου. Δεν δρομολογεί την κυκλοφορία ούτε παρέχει ορατότητα σε όλο το δίκτυο.

<b>ΠΡΩΤΟΚΟΛΛΑ</b>	<b>ΒΑΣΙΚΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ</b>	<b>ΒΑΣΙΚΑ ΜΕΙΟΝΕΚΤΗΜΑΤΑ</b>
OpenFlow	Πλήρως προσαρμόσιμο ,επεκτασιμότητα	Σύνθετο
NetConf	Απλότητα, διαχείριση	Περιορισμένη απόδοση

OVSDB	Προσαρμόσιμο , διαχείριση	Λίγες σύνθετες επιλογές
BGP	Χρήση σε διαφορετικά δίκτυα , διευθυνσιοδότηση	Συνίσταται μόνο για πολύ μεγάλα δίκτυα
LISP	Απλότητα, αποτελεσματικός έλεγχος κίνησης	Περιορισμένες δυνατότητες
SNMP	Προηγμένος έλεγχος	Σύνθετο
LLDP	Μεγάλο φάσμα συσκευών	Περιορισμός μόνο σε LAN δίκτυα

Πίνακας5.1 Σύγκριση πρωτοκόλλων ελεγκτών SDN.

## **6. ΠΡΟΔΙΑΓΡΑΦΕΣ ΥΛΙΚΟΥ ΚΑΙ ΛΟΓΙΣΜΙΚΟΥ**

Σε αυτό το κεφάλαιο θα αναλυθεί το κάθε εργαλείο που χρησιμοποιήθηκε για την παρούσα μεταπτυχιακή εργασία . Συγκεκριμένα θα αναλυθεί τόσο το κομμάτι του υλικού όσο και το κομμάτι του λογισμικού.

### **6.1 Προδιαγραφές υλικού**

Σε σχέση με προηγούμενες σχετικές έρευνες στις οποίες χρησιμοποιήθηκαν φορητοί υπολογιστές υψηλών επιδόσεων , σταθεροί υπολογιστές χαμηλών επιδόσεων ή ακόμα και workstation στη συγκεκριμένη έρευνα χρησιμοποιήθηκε ένας καινούργιος σταθερός υπολογιστής υψηλών επιδόσεων ο οποίος παρέχει τη δυνατότητα υλοποίησης μεγαλύτερης έκτασης εικονικών δικτύων καθώς και βελτιστοποιημένη διαχείριση και κατανομή των φυσικών πόρων του συστήματος επιτρέποντας την επίτευξη καλύτερων επιδόσεων και την αποτελεσματικότερη κλιμάκωση των εικονικών δικτύων που θα δημιουργηθούν.[3,12,15]

Στον πυρήνα του συστήματος συναντάμε μια μητρική Gigabyte B550M AORUS PRO η οποία διαθέτει το chipset B550 της AMD το οποίο επιτρέπει τη λειτουργία όλων των σύγχρονων τεχνικών διασυνδεσιμότητας όπως PCI EXPRESS 4.0 και παρέχει την βάση για ένα σύστημα υψηλών επιδόσεων σε επίπεδο ταχύτητας, επεξεργασίας καθώς και απεικόνισης.

Σαν επεξεργαστής επιλέχθηκε ο AMD RYZEN 5 5600X ο οποίος διαθέτει σύγχρονη αρχιτεκτονική πυρήνων TSMC 7nm FinFET και συνολικό αριθμό πυρήνων/νημάτων ο οποίος αντιστοιχεί σε 6/12 αντίστοιχα και μπορούν να φέρουν εις πέρας μεγαλύτερη κλίμακα εικονικών δικτύων από ότι αντίστοιχες σχετικές έρευνες. Είναι υπερχρονισμένος στα 4.7 GHz και ταυτόχρονα ρυθμισμένος σε χαμηλότερη τάση λειτουργίας για χαμηλότερες θερμοκρασίες και μικρότερη κατανάλωση περίπου 70 βαθμούς κελσίου και 45watt αντίστοιχα ενώ βρίσκεται στο 100% της λειτουργίας του. Οι παραπάνω ρυθμίσεις στο υλικό του συστήματος πραγματοποιούνται σε συνάρτηση με το chipset της μητρικής.

Η μνήμη ram του συστήματος έχει χωρητικότητα 32GB είναι γενιάς DDR4 είναι χρονισμένη στα 3600MHz και αποτελείται από 2 Module των 16GB το καθένα και προφανώς λειτουργεί σε DUAL-CHANNEL. Έχει αρκετή χωρητικότητα για να δοκιμάσουμε τα όρια του

συστήματος και παράλληλα αρκετή ταχύτητα για μεγάλο αριθμό ταυτόχρονων υπολογισμών.

Στο σημαντικό κομμάτι της αποθήκευσης στο σύστημα χρησιμοποιείται ένας δίσκος Kingston KC3000 οποίος είναι αρχιτεκτονικής NVME , χρησιμοποιεί δίαυλο PCI EXPRESS 4.0 , έχει χωρητικότητα 2TB και υψηλές επιδόσεις περίπου 7000 MB/s σε εγγραφή και διάβασμα αντίστοιχα.

Τέλος για τη γραφική απεικόνιση των δεδομένων ο υπολογιστής διαθέτει μια κάρτα γραφικών AMD RADEON 6750 XT υπερσύγχρονης γενιάς Navi 22 και αρχιτεκτονικής RDNA 2 που διαθέτει μνήμη γραφικών 12GB η οποία μπορεί να ανταπεξέλθει σε οποιαδήποτε δικτυακή υλοποίηση καθώς πρακτικά λειτουργεί σε πάρα πολύ μικρό ποσοστό ξεκούραστα σε σχέση με το υπόλοιπο σύστημα. Επιπλέον ο δίαυλός της είναι και αυτός PCI EXPRESS 4.0 .

CPU	RAM	GPU	STORAGE
AMD RYZEN 5 5600X 6c/12t 4.7 GHz 45Watt	32GB DDR4 3600 MHz (2x16) DUAL CHANNEL	AMD RADEON RX 6750 XT 12GB PCI EXPRESS 4.0	KINGSTON KC3000 NVME PCI EXPRESS 4.0 7GB/s

Πίνακας 6.1 Τεχνικές προδιαγραφές συστήματος υλοποίησης εργασίας .

## 6.2 Προδιαγραφές λογισμικού

Σε αυτή την ενότητα αναλύονται οι προδιαγραφές του λογισμικού του συστήματος που χρησιμοποιήθηκε. Είναι πολύ σημαντικό όχι μόνο να γίνει έρευνα προκειμένου να χρησιμοποιηθεί το σωστό λογισμικό το οποίο μπορεί να δώσει τα αποτελέσματα τα οποία επιθυμούμε αλλά και να μπορούν όλα τα λογισμικά να συνεργάζονται αρμονικά μεταξύ τους καθώς πρέπει να υπάρχει συνοχή , συνάφεια και επανελεγχόμενος έλεγχος στα αποτελέσματα που θα εξαχθούν.

### 6.2.1 Windows 11 Pro for Workstations OS

Το κύριο λειτουργικό σύστημα που χρησιμοποιήθηκε δεν μπορούσε να είναι άλλο από το Windows καθότι περιλαμβάνει μεγάλη επεκτασιμότητα και συνδεσιμότητα με υλικό και λογισμικό καθώς και μπορεί να χρησιμοποιείται από έναν απλό και καθημερινό υπολογιστή μέχρι έναν υπολογιστή με χρήση ερευνητική και πειραματική. Επιλέχθηκε η τελευταία έκδοση 11 καθότι περιέχει τις τελευταίες ενημερώσεις στον τομέα των εξελίξεων των οδηγών του συστήματος και το λογισμικό έχει αρχιτεκτονική 64 Bit.

Τα Windows 11 Pro for Workstations είναι μια έκδοση των Windows 11 που έχει σχεδιαστεί ειδικά για υλικό και φόρτους εργασίας προηγμένης τεχνολογίας. Περιλαμβάνει πολλές λειτουργίες που δεν είναι διαθέσιμες στην τυπική έκδοση των Windows 11 Pro, όπως:

- Ανθεκτικό σύστημα αρχείων (ReFS): Το ReFS είναι ένα σύστημα αρχείων σχεδιασμένο για συσκευές αποθήκευσης προηγμένης τεχνολογίας και παρέχει αυξημένη ανθεκτικότητα δεδομένων έναντι της καταστροφής δεδομένων.
- Μόνιμη μνήμη: Τα Windows 11 Pro for Workstations υποστηρίζουν μη πτητικές μονάδες μνήμης, όπως το Intel Optane, οι οποίες παρέχουν ταχύτερη πρόσβαση στα δεδομένα και χαμηλότερη καθυστέρηση.
- Λειτουργία σταθμού εργασίας: Αυτή η λειτουργία παρέχει κορυφαία απόδοση και αξιοπιστία για απαιτητικούς φόρτους εργασίας, όπως απόδοση 3D, επεξεργασία βίντεο και επιστημονικές προσομοιώσεις.
- Εκτεταμένη υποστήριξη υλικού: Υποστηρίζουν έως και 4 CPU, 6 TB μνήμης RAM και πολλαπλές GPU, καθιστώντας το ιδανικό για σενάρια υπολογιστών υψηλής απόδοσης.
- Ταχύτερη δικτύωση: Η έκδοση περιλαμβάνει υποστήριξη για το SMB Direct, ένα πρωτόκολλο που επιτρέπει γρήγορες μεταφορές δεδομένων μέσω του δικτύου, μειώνοντας τον χρόνο που απαιτείται για τη μεταφορά μεγάλων αρχείων.

Συνολικά, τα Windows 11 Pro for Workstations είναι σχεδιασμένα για να παρέχουν αυξημένη απόδοση, αξιοπιστία και υποστήριξη για υλικό προηγμένης τεχνολογίας για χρήστες που απαιτούν απαιτητικούς φόρτους εργασίας.[33]

### **6.2.2 VirtualBox**

Το VirtualBox είναι ένα δωρεάν λογισμικό εικονικοποίησης ανοιχτού κώδικα που επιτρέπει στους χρήστες να εκτελούν πολλά λειτουργικά συστήματα σε έναν μόνο φυσικό υπολογιστή. Στην παρούσα εργασία εγκαταστάθηκε στο λειτουργικό μας σύστημα και μας παρέχει τη δυνατότητα να τρέξουμε το λογισμικό που χρειάζεται για την μελέτη των υλοποιήσεων SDN. Αναπτύχθηκε αρχικά από την Innotek GmbH και αργότερα αποκτήθηκε από την Oracle Corporation. Η έκδοση που χρησιμοποιήθηκε στο τελικό στάδιο υλοποίησης των πειραμάτων είναι η 7.0.8 με τελευταία αναβάθμιση τον

Απρίλιο του 2023. Το VirtualBox υποστηρίζει ένα ευρύ φάσμα επισκεπτών λειτουργικών συστημάτων από τα οποία πλέον δεν λείπει κανένα. Τα πιο δημοφιλή από αυτά είναι:

- Windows
- Linux
- macOS
- Solaris

Το VirtualBox επιτρέπει στους χρήστες να δημιουργούν εικονικές μηχανές (VM) που λειτουργούν σαν ξεχωριστοί υπολογιστές με το δικό τους εικονικό υλικό, όπως εικονικές CPU, εικονικές κάρτες γραφικών, διεπαφές εικονικού δικτύου και συσκευές εικονικής αποθήκευσης. Κάθε εικονική μηχανή μπορεί να τρέξει το δικό της λειτουργικό σύστημα και εφαρμογές, και μπορεί να απομονωθεί από άλλα VM που εκτελούνται στον ίδιο κεντρικό υπολογιστή.

Μερικά από τα βασικά χαρακτηριστικά του VirtualBox περιλαμβάνουν:

- Συμβατότητα μεταξύ πλατφορμών: Το VirtualBox εκτελείται σε Windows, macOS, Linux και Solaris.
- Εικονικοποίηση υλικού: Το VirtualBox υποστηρίζει επεκτάσεις εικονικοποίησης υλικού, όπως οι Intel VT-x και AMD-V, οι οποίες βελτιώνουν την απόδοση των εικονικών μηχανών.
- Στιγμιότυπο: Το VirtualBox επιτρέπει στους χρήστες να τραβούν στιγμιότυπα των VM τους, τα οποία μπορούν να χρησιμοποιηθούν για την επαναφορά του VM στην προηγούμενη κατάσταση, εάν κάτι πάει στραβά.
- Αδιάλειπτη λειτουργία: Η απρόσκοπτη λειτουργία του VirtualBox επιτρέπει στα παράθυρα του επισκέπτη λειτουργικού συστήματος να εμφανίζονται σαν να εκτελούνται στην επιφάνεια εργασίας του κεντρικού λειτουργικού συστήματος.
- Υποστήριξη USB: Το VirtualBox υποστηρίζει συσκευές USB, επιτρέποντας στους χρήστες να χρησιμοποιούν συσκευές USB όπως εκτυπωτές, σαρωτές και εξωτερικές συσκευές αποθήκευσης μέσα από τις εικονικές τους μηχανές.

Το VirtualBox χρησιμοποιείται ευρέως από προγραμματιστές, επαγγελματίες πληροφορικής και λάτρεις που χρειάζονται να τρέξουν πολλά λειτουργικά συστήματα σε έναν υπολογιστή για σκοπούς δοκιμής, ανάπτυξης ή πειραματισμού. [46]



### 6.2.3 Mininet OS

Το εικονικό λογισμικό που εγκαταστάθηκε στην εικονική μηχανή VirtualBox είναι το Mininet. Το Mininet είναι ένας εξομοιωτής δικτύου που μπορεί να χρησιμοποιηθεί για την προσομοίωση και τη δοκιμή δικτύων που καθορίζονται από λογισμικό (SDN). Πιο συγκεκριμένα η έκδοση που χρησιμοποιείται είναι η 2.3.0 η οποία βασίζεται στο Ubuntu 20.04.1 και είναι η τελευταία έκδοση mininet η οποία κυκλοφόρησε τον Φεβρουάριο του 2021.

Το Mininet μπορεί να χρησιμοποιηθεί για την προσομοίωση ενός περιβάλλοντος SDN προσομοιώνοντας ένα δίκτυο μεταγωγέων και κεντρικών υπολογιστών και εκτελώντας έναν ελεγκτή SDN πάνω από το εξομοιούμενο δίκτυο. Στη συνέχεια, ο ελεγκτής SDN μπορεί να χρησιμοποιηθεί για τη διαμόρφωση της συμπεριφοράς του δικτύου, όπως η ρύθμιση διαδρομών δικτύου, η διαμόρφωση πολιτικών δικτύου και η παρακολούθηση της κυκλοφορίας του δικτύου. Το Mininet υποστηρίζει πολλούς δημοφιλείς ελεγκτές SDN, συμπεριλαμβανομένων ελεγκτών OpenFlow όπως OpenDaylight, ONOS και Ryu.

Μερικά από τα πλεονεκτήματα της χρήσης Mininet για δοκιμές SDN περιλαμβάνουν:

- Ευελιξία: Το Mininet επιτρέπει στους χρήστες να δημιουργούν προσαρμοσμένες τοπολογίες δικτύου και να μιμούνται διαφορετικά σενάρια δικτύου, καθιστώντας εύκολη τη δοκιμή και τον πειραματισμό με διαφορετικές διαμορφώσεις SDN.
- Απομόνωση: Το Mininet παρέχει υψηλό βαθμό απομόνωσης μεταξύ στοιχείων δικτύου, επιτρέποντας στους χρήστες να δοκιμάζουν και να διορθώνουν τις εφαρμογές SDN σε ελεγχόμενο περιβάλλον χωρίς να επηρεάζουν τα δίκτυα παραγωγής.
- Ενσωμάτωση: Το Mininet μπορεί να ενσωματωθεί με άλλα εργαλεία δοκιμών, όπως το Wireshark και το tcpdump, για να παρέχει ένα πιο ολοκληρωμένο περιβάλλον δοκιμών.
- Επεκτασιμότητα: Το Mininet μπορεί να μιμηθεί δίκτυα μεγάλης κλίμακας με εκατοντάδες ή χιλιάδες κόμβους, παρέχοντας ένα ρεαλιστικό περιβάλλον δοκιμών για μεγάλης κλίμακας αναπτύξεις SDN.

Συνολικά, το Mininet είναι ένα ισχυρό εργαλείο για δοκιμές και ανάπτυξη SDN, παρέχοντας ένα ευέλικτο και προσαρμόσιμο περιβάλλον σε ερευνητές, εκπαιδευτικούς και προγραμματιστές για να πειραματιστούν και να δοκιμάσουν εφαρμογές και ελεγκτές SDN.[34]

#### 6.2.4 X-Ming

Το Xming είναι ένας δωρεάν και ανοιχτού κώδικα διακομιστής X-Window-System για λειτουργικά συστήματα Microsoft Windows. Το X-Window-System είναι ένα πρωτόκολλο δικτύου που επιτρέπει την εμφάνιση γραφικών διεπαφών χρήστη σε απομακρυσμένα μηχανήματα μέσω δικτύου. Η τελευταία έκδοση που χρησιμοποιήθηκε είναι η 7.7.0.89-Απρίλιος 2023.

Το Xming επιτρέπει στους χρήστες των Windows να εκτελούν εφαρμογές Linux και Unix που απαιτούν διακομιστή συστήματος X-window, χωρίς να χρειάζεται να εγκαταστήσουν ένα πλήρες λειτουργικό σύστημα Linux ή Unix. Με το Xming, οι χρήστες των Windows μπορούν να εκτελούν εφαρμογές γραφικών, όπως απομακρυσμένους επιτραπέζιους υπολογιστές, προγράμματα επεξεργασίας κειμένου και προγράμματα περιήγησης ιστού σε έναν απομακρυσμένο διακομιστή Linux ή Unix και να εμφανίζουν τη διεπαφή χρήστη της εφαρμογής στην επιφάνεια εργασίας των Windows.

Το Xming υποστηρίζει ένα ευρύ φάσμα λειτουργιών του συστήματος X-Window, συμπεριλαμβανομένων πολλαπλών παραθύρων, αποκοπής και επικόλλησης και απομακρυσμένου ελέγχου ταυτότητας. Μπορεί να χρησιμοποιηθεί σε συνδυασμό με δημοφιλή πρωτόκολλα απομακρυσμένης επιφάνειας εργασίας όπως το SSH και το VNC για να παρέχει μια ασφαλή και αξιόπιστη λύση απομακρυσμένης πρόσβασης. Μερικά από τα βασικά χαρακτηριστικά του Xming περιλαμβάνουν:

- Εύκολο στη χρήση: Το Xming είναι εύκολο στην εγκατάσταση και τη ρύθμιση παραμέτρων και παρέχει ένα απλό περιβάλλον εργασίας χρήστη για τη διαχείριση των περιόδων λειτουργίας X-Window-System.
- Συμβατότητα μεταξύ πλατφορμών: Υποστηρίζει Windows XP, Vista, 7, 8 και 10 και μπορεί να χρησιμοποιηθεί με ένα ευρύ φάσμα λειτουργικών συστημάτων Linux και Unix.

- Ελαφρύ: Είναι ελαφρύ και δεν απαιτεί πολλούς πόρους συστήματος, καθιστώντας το κατάλληλο για χρήση σε παλαιότερους ή λιγότερο ισχυρούς υπολογιστές.
- Προσαρμόσιμο: Μπορεί να προσαρμοστεί για να ταιριάζει σε μεμονωμένες προτιμήσεις, όπως η αλλαγή της ανάλυσης της οθόνης ή η διαμόρφωση των γραμματοσειρών του συστήματος παραθύρου X.

Συνολικά, το Xming είναι ένα ισχυρό εργαλείο για χρήστες Windows που πρέπει να εκτελούν εφαρμογές Linux ή Unix που απαιτούν διακομιστή συστήματος X Window. Παρέχει έναν απλό και αξιόπιστο τρόπο πρόσβασης σε απομακρυσμένες εφαρμογές και μπορεί να χρησιμοποιηθεί σε συνδυασμό με μια ποικιλία πρωτοκόλλων απομακρυσμένης πρόσβασης για την παροχή μιας ασφαλούς και ευέλικτης λύσης απομακρυσμένης πρόσβασης. Στη συγκεκριμένη έρευνα παρείχε τη δυνατότητα πολλαπλών τερματικών παραθύρων παράλληλης λειτουργίας του προσομοιωτή προκειμένου να γίνει σωστή και λεπτομερής καταγραφή μετρικών επιδόσεων.[51]

## 6.2.5 WinSCP

Το WinSCP είναι ένα δωρεάν και ανοιχτού κώδικα πρόγραμμα-πελάτη SFTP, FTP και SCP για Windows. Επιτρέπει στους χρήστες να μεταφέρουν με ασφάλεια αρχεία μεταξύ τοπικών και απομακρυσμένων υπολογιστών και υποστηρίζει μια ποικιλία μεθόδων κρυπτογράφησης και ελέγχου ταυτότητας για την εξασφάλιση ασφαλών μεταφορών αρχείων.

Παρέχει μια γραφική διεπαφή χρήστη (GUI) που διευκολύνει τη μεταφόρτωση και τη λήψη αρχείων, καθώς και τη διαχείριση απομακρυσμένων αρχείων και καταλόγων. Το GUI είναι διαισθητικό και φιλικό προς το χρήστη, με δυνατότητες όπως μεταφορά και απόθεση, συγχρονισμός αρχείων και απομακρυσμένη επεξεργασία αρχείων. Η τελευταία έκδοση που χρησιμοποιήθηκε είναι η 5.21.7 με τελευταία ενημέρωση τον Ιανουάριο του 2023.

Μερικά από τα βασικά χαρακτηριστικά του WinSCP περιλαμβάνουν:

- Ασφαλής μεταφορά αρχείων: Το WinSCP υποστηρίζει μια ποικιλία μεθόδων κρυπτογράφησης και ελέγχου ταυτότητας, συμπεριλαμβανομένων των SSH, SSL/TLS και Kerberos, για τη διασφάλιση ασφαλών μεταφορών αρχείων.

- **Συγχρονισμός:** Το WinSCP μπορεί να συγχρονίσει τοπικούς και απομακρυσμένους καταλόγους, διασφαλίζοντας ότι οι δύο κατάλογοι έχουν τα ίδια αρχεία και φακέλους.
- **Απομακρυσμένη επεξεργασία αρχείων:** Το WinSCP επιτρέπει στους χρήστες να επεξεργάζονται απομακρυσμένα αρχεία χρησιμοποιώντας το πρόγραμμα επεξεργασίας που προτιμούν και ανεβάζει αυτόματα τις αλλαγές στον απομακρυσμένο διακομιστή.
- **Σενάρια και αυτοματισμός:** Το WinSCP υποστηρίζει δέσμες ενεργειών και αυτοματισμούς μέσω της διεπαφής γραμμής εντολών και της μηχανής δέσμης ενεργειών, επιτρέποντας στους χρήστες να αυτοματοποιούν επαναλαμβανόμενες εργασίες και να ενσωματώνουν το WinSCP με άλλα εργαλεία και εφαρμογές.

Συνολικά, το WinSCP είναι ένας ισχυρός και αξιόπιστος πελάτης μεταφοράς αρχείων για Windows, με φιλική προς το χρήστη διεπαφή και ένα ευρύ φάσμα λειτουργιών για την υποστήριξη ασφαλών και αποτελεσματικών μεταφορών αρχείων. [50]

## 6.2.6 PuTTY

Το PuTTY είναι ένας δωρεάν εξομοιωτής τερματικού ανοιχτού κώδικα, σειριακή κονσόλα και εφαρμογή μεταφοράς αρχείων δικτύου για Windows. Υποστηρίζει μια ποικιλία πρωτοκόλλων δικτύου, συμπεριλαμβανομένων των SSH, Telnet και Rlogin, και μπορεί να χρησιμοποιηθεί για ασφαλή σύνδεση σε απομακρυσμένους διακομιστές και συσκευές. Παρέχει μια γραφική διεπαφή χρήστη (GUI) που διευκολύνει τη δημιουργία και τη διαχείριση συνδέσεων δικτύου. Υποστηρίζει επίσης ένα ευρύ φάσμα δυνατοτήτων εξομοίωσης τερματικού, συμπεριλαμβανομένων προσαρμόσιμων γραμματοσειρών και χρωμάτων, και υποστηρίζει Unicode και αμφίδρομο κείμενο. Η έκδοση που χρησιμοποιήθηκε είναι η 0.78 με τελευταία ενημέρωση τον Οκτώβρη του 2022.

Μερικά από τα βασικά χαρακτηριστικά του PuTTY περιλαμβάνουν:

- **Ασφαλείς συνδέσεις δικτύου:** Το PuTTY υποστηρίζει μια ποικιλία από ασφαλή πρωτόκολλα δικτύου, συμπεριλαμβανομένου του SSH, το οποίο χρησιμοποιεί κρυπτογράφηση και έλεγχο ταυτότητας για την προστασία των συνδέσεων δικτύου από μη εξουσιοδοτημένη πρόσβαση.

- Πολλαπλά πρωτόκολλα δικτύου: Το PuTTY υποστηρίζει μια ποικιλία πρωτοκόλλων δικτύου, συμπεριλαμβανομένων των SSH, Telnet και Rlogin, καθιστώντας το ένα ευέλικτο εργαλείο για τη διαχείριση των συνδέσεων δικτύου.
- Συμβατότητα μεταξύ πλατφορμών: Το PuTTY μπορεί να χρησιμοποιηθεί σε συστήματα Windows, Linux και Unix και μπορεί να εκτελεστεί από μονάδα USB ή άλλα φορητά μέσα.
- Προσαρμόσιμη διεπαφή χρήστη: Η διεπαφή χρήστη του PuTTY μπορεί να προσαρμοστεί ώστε να ταιριάζει στις μεμονωμένες προτιμήσεις, συμπεριλαμβανομένων των γραμματοσειρών, των χρωμάτων και άλλων ρυθμίσεων.

Συνολικά, το PuTTY είναι ένα ισχυρό και αξιόπιστο εργαλείο για τη διαχείριση των συνδέσεων δικτύου, με φιλική προς το χρήστη διεπαφή και ένα ευρύ φάσμα λειτουργιών για την υποστήριξη της ασφαλούς και αποτελεσματικής διαχείρισης δικτύου. [42]

## 6.2.7 Visual Studio Code

Το Visual Studio Code (VS Code) είναι ένα δωρεάν πρόγραμμα επεξεργασίας κώδικα ανοιχτού κώδικα που αναπτύχθηκε από τη Microsoft για Windows, Linux και macOS. Παρέχει ένα ελαφρύ και προσαρμόσιμο πρόγραμμα επεξεργασίας που υποστηρίζει ένα ευρύ φάσμα γλωσσών προγραμματισμού και εργαλείων. Το VS Code προσφέρει ένα πλούσιο σύνολο λειτουργιών για τη βελτίωση της παραγωγικότητας και τη βελτίωση της εμπειρίας ανάπτυξης, όπως η επισήμανση και η συμπλήρωση κώδικα, ο εντοπισμός σφαλμάτων και η ενσωμάτωση Git. Υποστηρίζει επίσης επεκτάσεις και πρόσθετα, τα οποία επιτρέπουν στους χρήστες να προσαρμόζουν και να βελτιώνουν τη λειτουργικότητα του προγράμματος επεξεργασίας. Έγινε χρήση της τελευταίας έκδοσης 1.77 Μάρτιος 2023. Μερικά από τα βασικά χαρακτηριστικά του Visual Studio Code περιλαμβάνουν:

- Επεξεργασία κώδικα: Το VS Code προσφέρει μια ισχυρή εμπειρία επεξεργασίας κώδικα, με υποστήριξη για επισήμανση σύνταξης, συμπλήρωση κώδικα και πλοήγηση κώδικα.
- Εντοπισμός σφαλμάτων: Ο κώδικας VS περιλαμβάνει μια ισχυρή μηχανή εντοπισμού σφαλμάτων που υποστηρίζει ένα ευρύ φάσμα γλωσσών και πλατφορμών, συμπεριλαμβανομένων των Node.js, Python και .NET.
- Επεκτάσεις: Το VS Code υποστηρίζει ένα ευρύ φάσμα επεκτάσεων και προσθηκών, οι οποίες μπορούν να χρησιμοποιηθούν για τη βελτίωση της

λειτουργικότητας του προγράμματος επεξεργασίας και την υποστήριξη πρόσθετων γλωσσών και εργαλείων.

- Ενσωματωμένο τερματικό: Ο κώδικας VS περιλαμβάνει ένα ενσωματωμένο τερματικό, το οποίο επιτρέπει στους χρήστες να εκτελούν εντολές και να εκτελούν σενάρια χωρίς να εγκαταλείπουν το πρόγραμμα επεξεργασίας.
- Ενοποίηση Git: Το VS Code περιλαμβάνει ενσωματωμένη υποστήριξη Git, η οποία επιτρέπει στους χρήστες να διαχειρίζονται αποθετήρια κώδικα και να εκτελούν κοινές λειτουργίες Git απευθείας από τον επεξεργαστή.

Συνολικά, το Visual Studio Code είναι ένα ισχυρό και ευέλικτο πρόγραμμα επεξεργασίας κώδικα, με ένα πλούσιο σύνολο δυνατοτήτων και υποστήριξη για ένα ευρύ φάσμα γλωσσών και εργαλείων. Η ευελιξία και η δυνατότητα προσαρμογής του το καθιστούν δημοφιλή επιλογή μεταξύ των προγραμματιστών για μια ποικιλία έργων και ροών εργασίας. Στην παρούσα εργασία η χρήση του ήταν η δημιουργία τοπολογιών μέσω συγγραφής Python Scripts. Η έκδοση της Python που χρησιμοποιήθηκε είναι η 3.12. [47]

## 6.2.8 Gephi

Το Gephi είναι ένα λογισμικό ανοιχτού κώδικα για οπτικοποίηση και εξερεύνηση γραφημάτων και δικτύων. Είναι ένα ισχυρό εργαλείο που επιτρέπει στους χρήστες να αναλύουν και να οπτικοποιούν πολύπλοκα δίκτυα, όπως κοινωνικά δίκτυα, βιολογικά δίκτυα και δίκτυα μεταφορών. Η έκδοση που χρησιμοποιήθηκε είναι η 0.10.1 Ιανουάριος 2023. Το Gephi παρέχει μια σειρά από λειτουργίες για την υποστήριξη της ανάλυσης και της οπτικοποίησης δικτύων, όπως:

- Διαδραστική απεικόνιση: Η Gephi προσφέρει μια σειρά από εργαλεία διαδραστικής απεικόνισης, συμπεριλαμβανομένων 2D και 3D διατάξεων, για να βοηθήσει τους χρήστες να εξερευνήσουν και να κατανοήσουν τη δομή των δικτύων τους.
- Ανάλυση δικτύου: Η Gephi παρέχει μια σειρά από αλγόριθμους για την ανάλυση των ιδιοτήτων του δικτύου, όπως η κεντρικότητα, η ομαδοποίηση και ο εντοπισμός κοινότητας, οι οποίοι μπορούν να βοηθήσουν τους χρήστες να αναγνωρίσουν βασικούς κόμβους και δομές στα δίκτυά τους.
- Εισαγωγή και εξαγωγή δεδομένων: Η Gephi υποστηρίζει μια σειρά μορφών δεδομένων, συμπεριλαμβανομένων των CSV, Excel και GEXF, καθιστώντας εύκολη την εισαγωγή δεδομένων από διάφορες πηγές και την εξαγωγή οπτικοποιήσεων και αποτελεσμάτων ανάλυσης.

- Προσαρμογή και επεκτασιμότητα: Το Gexhi είναι εξαιρετικά προσαρμόσιμο και επεκτάσιμο, με μια σειρά από πρόσθετα και API που επιτρέπουν στους χρήστες να προσαρμόσουν το λογισμικό στις συγκεκριμένες ανάγκες τους.

Συνολικά, το Gexhi είναι ένα ισχυρό εργαλείο για την ανάλυση και την οπτικοποίηση πολύπλοκων δικτύων, με μια σειρά λειτουργιών που υποστηρίζουν την εξερεύνηση δεδομένων και τη δημιουργία πληροφοριών. Η φύση του ανοιχτού κώδικα και η επεκτασιμότητα του το καθιστούν δημοφιλή επιλογή μεταξύ ερευνητών και αναλυτών που εργάζονται με δεδομένα δικτύου. [30]

## 7.ΠΡΟΔΙΑΓΡΑΦΕΣ ΠΕΙΡΑΜΑΤΩΝ

### 7.1. Δικτυακές τοπολογίες

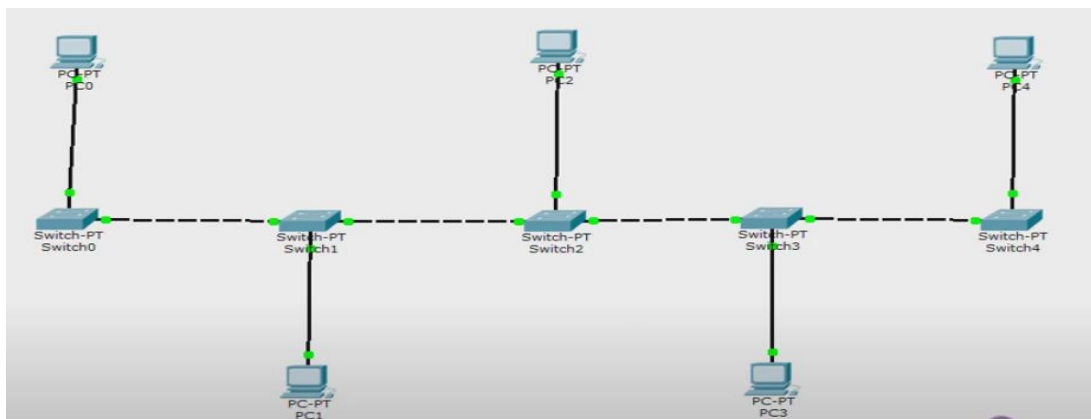
Ο όρος τοπολογία ορίζει τη γεωμετρική αναπαράσταση της σύνδεσης σε ένα δίκτυο. Υπάρχουν τρεις τύποι τοπολογιών. Στην έρευνά μου χρησιμοποιήθηκε μία από κάθε κατηγορία για αντικειμενική μελέτη. Συγκεκριμένα στις βασικές τοπολογίες επιλέχθηκε η τοπολογία διαύλου, στις υβριδικές επιλέχθηκε η τοπολογία ισορροπημένου δένδρου και στις Custom επιλέχθηκε η τυχαία τοπολογία.

- Βασική
- Υβριδική
- Custom

#### 7.1.1 Βασικές τοπολογίες

Υπάρχουν πολλές βασικές τοπολογίες δικτύου που χρησιμοποιούνται συνήθως στη δικτύωση υπολογιστών. Αυτές περιλαμβάνουν:

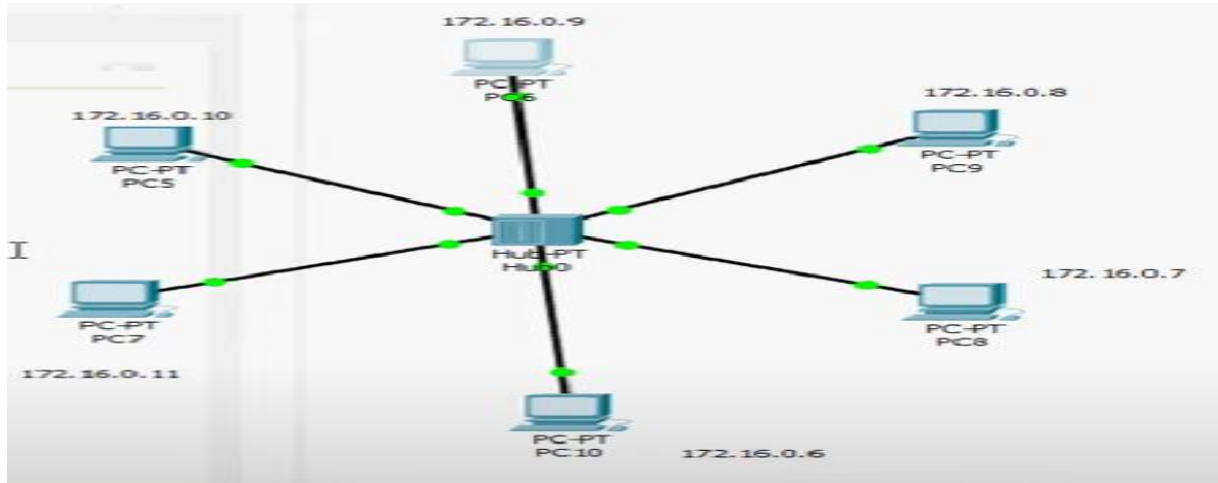
- Τοπολογία διαύλου: Όλες οι συσκευές είναι συνδεδεμένες σε μία μόνο γραμμή επικοινωνίας ή καλώδιο, γνωστό ως δίαυλος. Τα δεδομένα ταξιδεύουν και προς τις δύο κατευθύνσεις κατά μήκος του διαύλου και όλες οι συσκευές στο δίκτυο μπορούν να λάβουν το ίδιο μήνυμα ταυτόχρονα.



Εικόνα 7.1: Παράδειγμα τοπολογίας διαύλου.

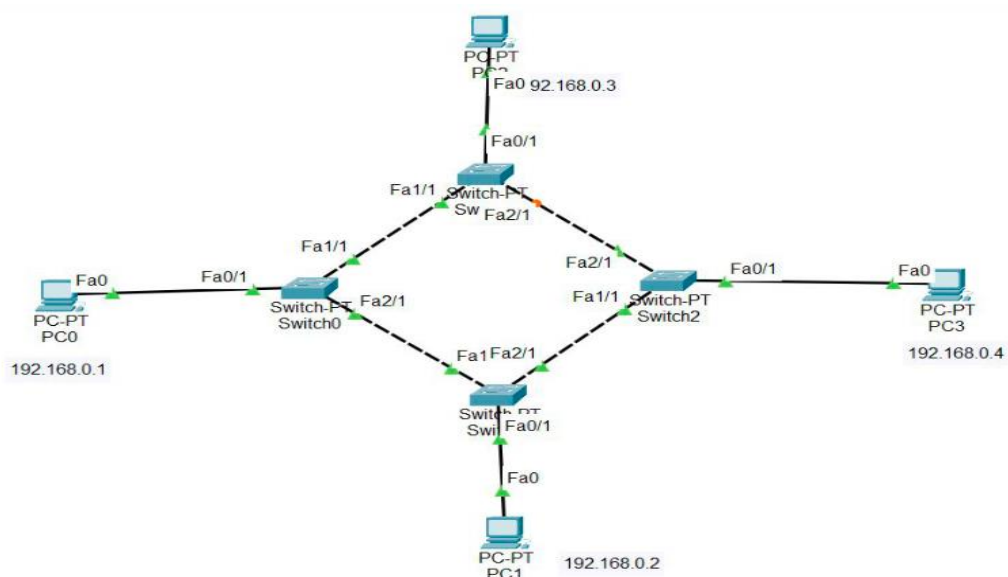


- Τοπολογία αστέρα: Όλες οι συσκευές συνδέονται σε έναν κεντρικό διανομέα ή διακόπτη και τα δεδομένα ρέουν μέσω του διανομέα ή του διακόπτη για να φτάσουν στον προορισμό τους. Κάθε συσκευή έχει μια αποκλειστική σύνδεση με τον διανομέα ή τον διακόπτη, η οποία μπορεί να βοηθήσει στη μείωση της συμφόρησης του δικτύου και στη βελτίωση της απόδοσης.



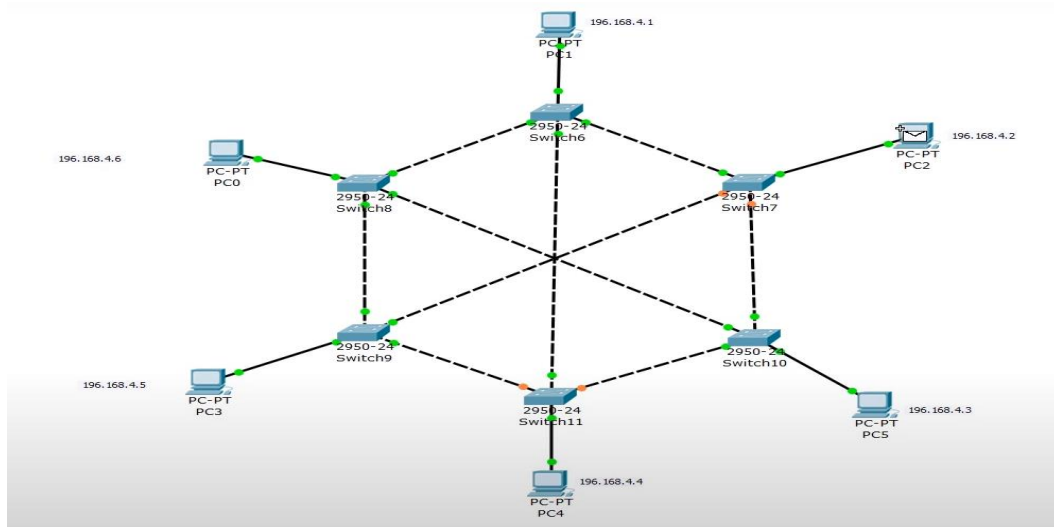
Εικόνα 7.2: Παράδειγμα τοπολογίας αστέρα.

- Τοπολογία δακτυλίου: Όλες οι συσκευές συνδέονται σε κλειστό βρόχο, με δεδομένα να ρέουν προς μία κατεύθυνση γύρω από τον βρόχο. Κάθε συσκευή λαμβάνει δεδομένα από την προηγούμενη συσκευή στον βρόχο και στέλνει δεδομένα στην επόμενη συσκευή στον βρόχο.



- Εικόνα 7.3: Παράδειγμα τοπολογίας δακτυλίου.

- Τοπολογία Πλέγματος: Κάθε συσκευή συνδέεται με κάθε άλλη συσκευή στο δίκτυο, δημιουργώντας ένα πλήρως διασυνδεδεμένο δίκτυο. Αυτό μπορεί να παρέχει υψηλή πλεονασμό και ανοχή σφαλμάτων, αλλά μπορεί να είναι πολύπλοκο στη διαχείριση και να απαιτεί πολλή καλωδίωση.



Εικόνα 7.4: Παράδειγμα τοπολογίας πλέγματος.

### 7.1.2 Υβριδικές τοπολογίες

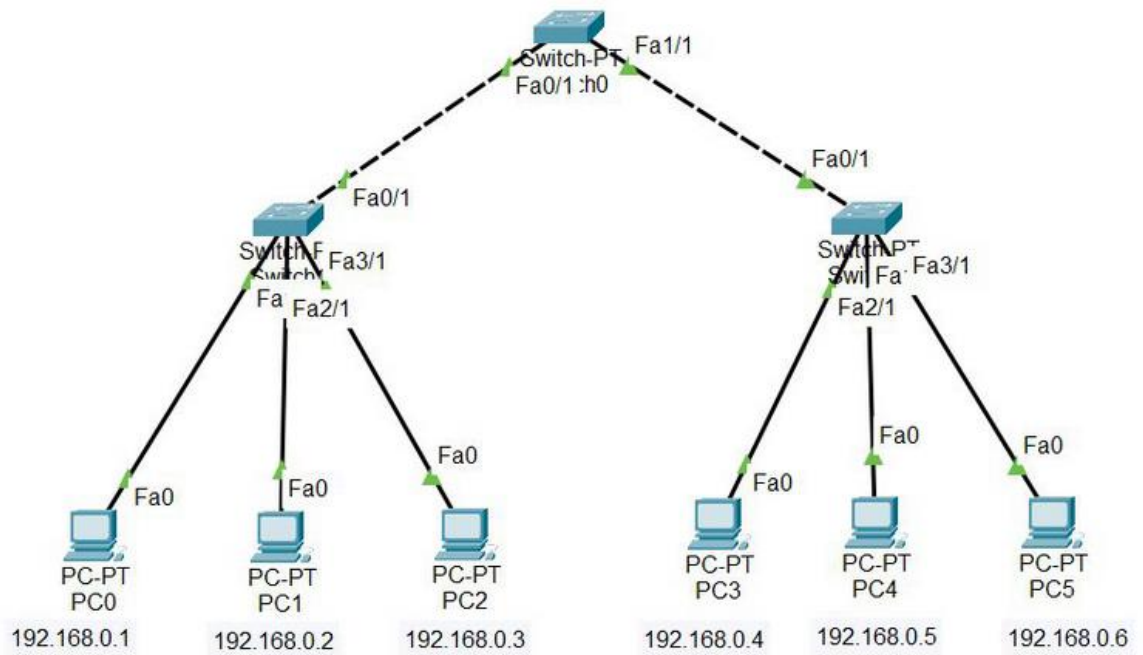
Υβριδική Τοπολογία αποτελεί ένας συνδυασμός δύο ή περισσότερων βασικών τοπολογιών, όπως μια τοπολογία αστέρα-διαύλου ή μια τοπολογία δακτυλίου-πλέγματος. Αυτό μπορεί να προσφέρει μια ισορροπία μεταξύ απόδοσης, πλεονασμού και ευκολίας διαχείρισης.

Η τοπολογία δέντρων, επίσης γνωστή ως ιεραρχική τοπολογία, είναι ένας τύπος τοπολογίας δικτύου που βασίζεται σε μια ιεραρχική δομή. Σε αυτήν την τοπολογία, πολλαπλές τοπολογίες αστέρα συνδέονται με μια τοπολογία διαύλου, δημιουργώντας μια δομή που μοιάζει με δέντρο. Σε μια τοπολογία δέντρου, ο κεντρικός διαύλος λειτουργεί ως ο κύριος κορμός του δέντρου, με πολλαπλούς κλάδους που εκτείνονται από αυτόν. Κάθε κλάδος είναι μια ξεχωριστή τοπολογία αστεριού, με έναν διανομέα ή μεταγωγέα στο κέντρο του και πολλαπλές συσκευές συνδεδεμένες σε αυτό. Αυτό επιτρέπει τη δημιουργία υποδικτύων εντός του μεγαλύτερου δικτύου, με κάθε υποδίκτυο να έχει τον δικό του αποκλειστικό διανομέα ή μεταγωγέα.

- Το κύριο πλεονέκτημα μιας τοπολογίας δέντρου είναι η επεκτασιμότητά της, καθώς μπορεί να υποστηρίξει μεγάλο αριθμό συσκευών και υποδικτύων. Παρέχει επίσης μια καλή ισορροπία μεταξύ απόδοσης και πλεονασμού, καθώς κάθε υποδίκτυο

μπορεί να λειτουργήσει ανεξάρτητα και τυχόν προβλήματα σε ένα υποδίκτυο δεν θα επηρεάσουν το υπόλοιπο δίκτυο.

- Ωστόσο, το κύριο μειονέκτημα μιας τοπολογίας δέντρου είναι η πολυπλοκότητά της, καθώς απαιτεί σημαντική ποσότητα καλωδίωσης και διαμόρφωσης. Μπορεί επίσης να είναι δύσκολη η αντιμετώπιση προβλημάτων και η διαχείριση, καθώς προβλήματα σε ένα μέρος του δικτύου μπορεί να επηρεάσουν ολόκληρο το δέντρο.



Εικόνα 7.5 Παράδειγμα υβριδικής τοπολογίας δέντρου.

Μια τοπολογία ισορροπημένου δέντρου είναι ένας συγκεκριμένος τύπος τοπολογίας δέντρου όπου κάθε κλάδος του δέντρου έχει τον ίδιο αριθμό επιπέδων. Αυτό σημαίνει ότι κάθε υποδίκτυο είναι ίσου μεγέθους και έχει τον ίδιο αριθμό συσκευών συνδεδεμένων σε αυτό. Σε μια ισορροπημένη τοπολογία δέντρου, ο κεντρικός διάυλος συνδέεται με ένα σύνολο μεταγωγέων επιπέδου-1, καθένας από τους οποίους συνδέεται με ένα σύνολο μεταγωγέων επιπέδου δύο και ούτω καθεξής, έως ότου επιτευχθεί το τελικό επίπεδο διακοπών. Κάθε μεταγωγέας στο δέντρο έχει ίσο αριθμό διακλαδώσεων συνδεδεμένων σε αυτό, γεγονός που βοηθά στην εξισορρόπηση της κίνησης του δικτύου και στην αποφυγή συμφόρησης.

### 7.1.3 Custom τοπολογίες

Οι προσαρμοσμένες τοπολογίες δικτύου αναφέρονται σε αρχιτεκτονικές δικτύου που έχουν σχεδιαστεί για να ικανοποιούν συγκεκριμένες απαιτήσεις ή να επιλύουν

συγκεκριμένα προβλήματα. Μπορεί να είναι ένας συνδυασμός δύο ή περισσότερων βασικών τοπολογιών ή μπορεί να είναι εντελώς μοναδικές και προσαρμοσμένες σε μια συγκεκριμένη εφαρμογή ή περιβάλλον. Οι προσαρμοσμένες τοπολογίες δικτύου μπορούν να δημιουργηθούν από σχεδιαστές και διαχειριστές δικτύων χρησιμοποιώντας διάφορες συσκευές και τεχνολογίες δικτύωσης, όπως μεταγωγείς, δρομολογητές, τείχη προστασίας, εξισορροπητές φορτίου και άλλα. Αυτές οι συσκευές μπορούν να ρυθμιστούν ώστε να εφαρμόζουν συγκεκριμένα πρωτόκολλα δρομολόγησης, VLAN, πολιτικές ελέγχου πρόσβασης και άλλες δυνατότητες για την επίτευξη της επιθυμητής συμπεριφοράς και απόδοσης δικτύου. Παραδείγματα προσαρμοσμένων τοπολογιών δικτύου περιλαμβάνουν:

- Τοπολογία πλέγματος με δυναμική δρομολόγηση: Αυτή η τοπολογία μπορεί να χρησιμοποιηθεί σε δίκτυα ασύρματων δικτύων μεγάλης κλίμακας για να παρέχει υψηλή πλεονασμό και ανοχή σφαλμάτων. Πρωτόκολλα δυναμικής δρομολόγησης όπως OLSR ή B.A.T.M.A.N. μπορεί να χρησιμοποιηθεί για τη βελτιστοποίηση της απόδοσης του δικτύου και τη μείωση της συμφόρησης.
- Τοπολογία Hub-and-Spoke με VPN: Αυτή η τοπολογία μπορεί να χρησιμοποιηθεί για τη σύνδεση πολλών απομακρυσμένων γραφείων ή υποκαταστημάτων σε μια κεντρική τοποθεσία χρησιμοποιώντας σήραγγες VPN. Ένας δρομολογητής hub ή ένα τείχος προστασίας χρησιμοποιείται για τη διαχείριση της ροής της κυκλοφορίας και την παροχή ασφαλούς συνδεσιμότητας μεταξύ των ακτινών.
- Τοπολογία συμπλέγματος με εξισορρόπηση φορτίου: Αυτή η τοπολογία μπορεί να χρησιμοποιηθεί για τη δημιουργία συμπλέγματος διακομιστών ιστού ή εφαρμογών υψηλής διαθεσιμότητας. Οι συσκευές εξισορρόπησης φορτίου χρησιμοποιούνται για τη διανομή της κυκλοφορίας σε πολλούς διακομιστές στο σύμπλεγμα, παρέχοντας υψηλή απόδοση και επεκτασιμότητα.

Οι προσαρμοσμένες τοπολογίες δικτύου μπορούν να παρέχουν μοναδικά πλεονεκτήματα και να λύσουν συγκεκριμένα προβλήματα, αλλά απαιτούν επίσης προσεκτικό σχεδιασμό και διαχείριση για να διασφαλιστεί η αποτελεσματικότητα και η ασφάλειά τους. Οι διαχειριστές δικτύου θα πρέπει να εξετάσουν τις συγκεκριμένες ανάγκες του οργανισμού τους και να συμβουλευτούν έμπειρους σχεδιαστές δικτύων για να δημιουργήσουν μια προσαρμοσμένη τοπολογία που να καλύπτει αυτές τις ανάγκες.

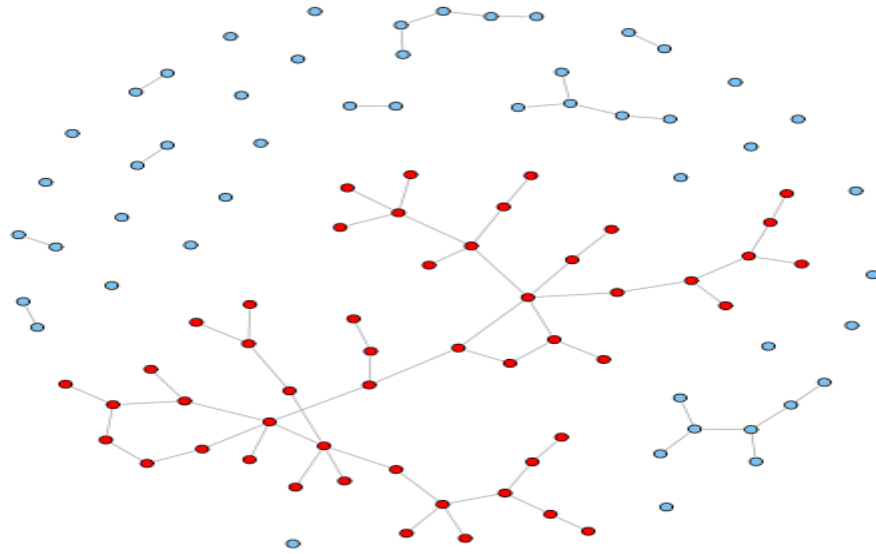
### 7.1.3.1 Τυχαία Τοπολογία

Στη δικτύωση υπολογιστών, μια τυχαία τοπολογία δικτύου αναφέρεται σε μια τοπολογία δικτύου όπου οι συνδέσεις μεταξύ των κόμβων δημιουργούνται με τυχαίο ή στοχαστικό τρόπο. Σε μια τέτοια τοπολογία, δεν υπάρχει προκαθορισμένο σχέδιο ή δομή στις συνδέσεις μεταξύ των κόμβων. Οι τυχαίες τοπολογίες δικτύων χρησιμοποιούνται σε διάφορες εφαρμογές, όπως στη μελέτη κοινωνικών δικτύων, βιολογικών δικτύων και δικτύων επικοινωνίας. Χρησιμοποιούνται επίσης στην ανάλυση των ιδιοτήτων του δικτύου, όπως η συνδεσιμότητα, η ευρωστία και η αποτελεσματικότητα. Έχει αποδειχθεί ότι παρουσιάζουν κάποιες ενδιαφέρουσες και απροσδόκητες συμπεριφορές, όπως η εμφάνιση δικτύων μικρόκοσμου και δικτύων χωρίς κλίμακα.

### 7.1.3.2 Μοντέλο Erdős–Rényi

Ένα παράδειγμα τυχαίας τοπολογίας δικτύου είναι το μοντέλο Erdős–Rényi. Το μοντέλο Erdős–Rényi, γνωστό και ως μοντέλο ER, είναι ένα μαθηματικό μοντέλο για τη δημιουργία τυχαίων γραφημάτων. Εισήχθη στον κλάδο των μαθηματικών από τους μαθηματικούς Paul Erdős και Alfréd Rényi το 1959. Στο μοντέλο ER, δημιουργείται ένα τυχαίο γράφημα με « $n$ » κόμβους ξεκινώντας με « $n$ » απομονωμένους κόμβους και στη συνέχεια συνδέοντας τυχαία ζεύγη κόμβων με μια ορισμένη πιθανότητα « $p$ ». Οι ακμές μεταξύ των κόμβων είναι ανεξάρτητες και εμφανίζονται με πιθανότητα « $p$ ». Υπάρχουν δύο παραλλαγές του μοντέλου ER: το μοντέλο  $G(n,m)$ , το οποίο δημιουργεί ένα τυχαίο γράφημα με « $n$ » κόμβους και  $m$  άκρες, και το μοντέλο  $G(n,p)$ , το οποίο δημιουργεί ένα τυχαίο γράφημα με « $n$ » κόμβους και ένα άκρη μεταξύ κάθε ζεύγους κόμβων με πιθανότητα « $p$ ».

Το μοντέλο ER έχει χρησιμοποιηθεί για τη μελέτη διαφόρων ιδιοτήτων τυχαίων γραφημάτων, συμπεριλαμβανομένης της εμφάνισης του γιγαντιαίου στοιχείου, της μετάβασης φάσης της συνδεσιμότητας και της κατανομής βαθμών του γραφήματος. Το μοντέλο έχει επίσης εφαρμοστεί σε διάφορους τομείς όπως τα κοινωνικά δίκτυα, τα δίκτυα υπολογιστών και η βιολογία. Ωστόσο, θα πρέπει να σημειωθεί ότι το μοντέλο ER υποθέτει μια εντελώς τυχαία και ομοιόμορφη κατανομή των ακμών, η οποία μπορεί να μην αντικατοπτρίζει πάντα την πραγματική δομή πολλών δικτύων. Ως αποτέλεσμα, άλλα μοντέλα δικτύων, όπως δίκτυα μικρού κόσμου και δίκτυα χωρίς κλίμακα, έχουν προταθεί για την καλύτερη αποτύπωση των ιδιοτήτων των πραγματικών δικτύων.



Εικόνα 7.6 Παράδειγμα Custom Τυχαίας τοπολογίας με χρήση του μαθηματικού μοντέλου Erdős-Rényi.

## 7.2 Προδιαγραφές πειραμάτων

Τα πειράματα υλοποιούν τοπολογίες βασισμένες στην τυχαία τοπολογία η οποία με τη σειρά της ακολουθεί το μαθηματικό μοντέλο ER . Ο ελεγκτής SDN που χρησιμοποιείται είναι ο POX λόγω της συμβατότητάς του και με τις τοπολογίες καθώς και με την δημιουργία και παραμετροποίηση τοπολογιών μέσω της γλώσσας PYTHON. Το πρωτόκολλο δημιουργίας τοπολογιών είναι το OFDP ή αλλιώς OpenFlow.

Αυτό που εξετάζουν τα πειράματα είναι τα εξής:

1. Σύγκριση απόδοσης συστήματος αναλόγως τοπολογιών.
2. Σύγκριση απόδοσης συστήματος αναλόγως πρωτοκόλλου δημιουργίας τοπολογίας.
3. Σύγκριση απόδοσης συστήματος αναλόγως αριθμού switch και πως ο συνολικός αριθμός των switch επηρεάζει την απόδοση.
4. Η χρήση της CPU.
5. Η χρήση της μνήμης RAM
6. Η καθυστέρηση της μεταφοράς πακέτων μεταξύ δικτυακών κόμβων.

7. Ο χρόνος δημιουργίας και καταστροφής μιας τοπολογίας.

Οι παραπάνω μετρήσεις θα πραγματοποιηθούν συγκρίνοντας :

1. Τοπολογίες
  - Γραμμική
  - Ισορροπημένου Δένδρου
  - Τυχαία Τοπολογία
2. Πρωτόκολλα δημιουργίας
  - OFDP
  - LLDP
  - BGP
  - LSDP
  - SNMP
  - OVSDB

Ο αριθμός των switch θα παραμένει σταθερά αυξανόμενος και σε κάθε switch θα συνδέεται ένας host με τον τρόπο που φαίνεται στον παρακάτω πίνακα. Επισημαίνεται ότι επιτεύχθηκε μεγαλύτερο πλήθος αριθμού switch από ότι στις περισσότερες αντίστοιχες έρευνες . Αυτό και μόνο ως δεδομένο επιτρέπει καλύτερη ερμηνεία αποτελεσμάτων και πραγματοποιείται κυρίως λόγω των διαθέσιμων υλικών πόρων (hardware).

<b>SWITCHES</b>	<b>HOSTS</b>
2	2
4	4
8	8
16	16
32	32
64	64
128	128
256	256
512	512
1024	1024
2048	2048
4096	4096
8192	8192

Πίνακας 7.1 Πίνακας κλίμακας διεξαγομένων πειραμάτων.

## 7.2.1 Μέτρηση CPU

Μπορούμε να μετρήσετε τη χρήση της CPU ενός δικτύου Mininet χρησιμοποιώντας μια ποικιλία εργαλείων. Εδώ υπάρχουν δύο επιλογές:

- **top:** Η εντολή `top` είναι ένα βοηθητικό πρόγραμμα Unix που παρέχει πληροφορίες σε πραγματικό χρόνο σχετικά με τις διεργασίες που εκτελούνται σε ένα σύστημα, συμπεριλαμβανομένης της χρήσης της CPU. Για να χρησιμοποιηθεί η `top` για την παρακολούθηση της χρήσης CPU ενός δικτύου Mininet:
  1. Ξεκινάμε το δίκτυο Mininet χρησιμοποιώντας την τοπολογία και τη διαμόρφωση που επιθυμούμε.
  2. Ανοίγουμε ένα νέο παράθυρο τερματικού μέσω του PuTTY σε συνεργασία με το X-ming και εκτελούμε την εντολή για να ξεκινήσουμε να παρακολουθούμε τη χρήση της CPU του συστήματος.
  3. Αναζητούμε τις διεργασίες που σχετίζονται με το δίκτυο Mininet, οι οποίες συνήθως ονομάζονται `python` ή `mnexec`. Η χρήση της CPU κάθε διεργασίας θα εμφανίζεται στη στήλη `%CPU`.
  4. Αφήνουμε την εντολή να εκτελείται για όσο διάστημα θέλουμε να παρακολουθούμε τη χρήση της CPU του δικτύου Mininet.
- **htop:** Το `htop` είναι μια διαδραστική έκδοση της εντολής `top` που παρέχει μια πιο φιλική προς το χρήστη διεπαφή για την παρακολούθηση των διαδικασιών του συστήματος. Λόγω της μεγαλύτερης ακρίβειάς της και τις αναλυτικότερης λειτουργίας της θα χρησιμοποιηθεί αυτή στα συγκεκριμένα πειράματα. Για να χρησιμοποιήσετε το `htop` για την παρακολούθηση της χρήσης της CPU ενός δικτύου Mininet:
  1. Πραγματοποιούμε εγκατάσταση του `htop` στο σύστημά εάν δεν είναι ήδη εγκατεστημένο (`sudo apt install htop`).
  2. Εκκινούμε το δίκτυο Mininet χρησιμοποιώντας την τοπολογία και τη διαμόρφωση που επιθυμούμε.
  3. Ανοίγουμε ένα νέο παράθυρο τερματικού μέσω του PuTTY σε συνεργασία με το X-ming και εκτελούμε την εντολή `htop` για να αρχίσουμε να παρακολουθούμε τη χρήση της CPU του συστήματος.



- Αναζητούμε τις διεργασίες που σχετίζονται με το δίκτυο Mininet, οι οποίες συνήθως ονομάζονται rython ή mnexec. Η χρήση της CPU κάθε διεργασίας θα εμφανίζεται στη στήλη %CPU.
- Αφήνουμε την εντολή htop να εκτελείται για όσο διάστημα θέλουμε να παρακολουθούμε τη χρήση της CPU του δικτύου Mininet.

Παρατηρούμε ότι η παρακολούθηση της χρήσης της CPU ενός δικτύου Mininet μπορεί να έχει αντίκτυπο στην απόδοση του ίδιου του δικτύου. Αξίζει να γίνει αναφορά στο ότι χρησιμοποιούμε αυτά τα εργαλεία με σύνεση και μόνο όταν είναι απαραίτητο για να αποφύγουμε την παρέμβαση στα πειραματικά μας αποτελέσματα. Όπως αναφέρθηκε και παραπάνω στα πειράματα μας χρησιμοποιείται η εντολή htop για μέτρηση της χρήσης CPU. Μπορεί επίσης να χρησιμοποιηθεί και για τη μέτρηση της χρήσης της μνήμης RAM αλλά όπως βλέπουμε στην επόμενη υπο-ενότητα είναι προτιμότερο να χρησιμοποιηθεί ένα λίγο διαφορετικό εργαλείο με τη μορφή εντολής.

```

Mininet-VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 [ 0.0% 4 [ 0.0% 7 [ 0.0% 10 [ 0.0%
2 [ 0.0% 5 [ 0.0% 8 [ 0.0% 11 [ 0.0%
3 [ 0.0% 6 [ 0.0% 9 [ 0.0% 12 [ 1.3%
Mem [ ] 151M/28.96M Tasks: 20, 7 thr: 1 running
Sup [ ] 0K/353M Load average: 1.86 0.61 0.21
Uptime: 00:00:49

  PID USER   PRI  NI  UIRT  RES  SHR  S  CPU% MEM%  TIME+  Command
 594 syslog  20   0  219M 4888 3852 S  0.0  0.0  0:00.89 /usr/sbin/rsyslogd -n -iNONE
 703 ntp      20   0  74636 3972 3404 S  0.0  0.0  0:00.16 /usr/sbin/ntpd -p /var/run/ntpd.pid
 779 mininet  20   0  11008 3968 3188 R  0.0  0.0  0:00.01 htop
    1 root    20   0   99M 11200 8624 S  0.0  0.0  0:04.73 /sbin/init text
 358 root    19  -1 59864 15264 14164 S  0.0  0.1  0:00.88 /lib/systemd/systemd-journald
 388 root    20   0  22312 6376 4956 S  0.0  0.0  0:10.83 /lib/systemd/systemd-udevd
 396 systemd-n 20   0  26784 7848 6892 S  0.0  0.0  0:01.24 /lib/systemd/systemd-networkd
 549 systemd-r 20   0  24000 12224 8220 S  0.0  0.0  0:01.47 /lib/systemd/systemd-resolved
 601 root    20   0  232M 7288 6460 S  0.0  0.0  0:00.08 /usr/lib/accounts-service/accounts-d
 619 root    20   0  232M 7288 6460 S  0.0  0.0  0:00.06 /usr/lib/accounts-service/accounts-d
 580 root    20   0  232M 7288 6460 S  0.0  0.0  0:00.36 /usr/lib/accounts-service/accounts-d
 581 root    20   0  9696 2864 2656 S  0.0  0.0  0:00.03 /usr/sbin/cron -f
 582 messagebu 20   0  7576 4672 4032 S  0.0  0.0  0:00.64 /usr/sbin/dbus-daemon --system --add
 600 root    20   0  81820 3612 3316 S  0.0  0.0  0:00.00 /usr/sbin/irqbalance --foreground
 592 root    20   0  81820 3612 3316 S  0.0  0.0  0:00.19 /usr/sbin/irqbalance --foreground
 593 root    20   0  31996 17964 10308 S  0.0  0.1  0:00.49 /usr/bin/python3 /usr/bin/networkd-
 614 syslog  20   0  219M 4888 3852 S  0.0  0.0  0:00.43 /usr/sbin/rsyslogd -n -iNONE
 615 syslog  20   0  219M 4888 3852 S  0.0  0.0  0:00.06 /usr/sbin/rsyslogd -n -iNONE
 616 syslog  20   0  219M 4888 3852 S  0.0  0.0  0:00.25 /usr/sbin/rsyslogd -n -iNONE
 596 root    20   0  16876 7744 6780 S  0.0  0.0  0:00.95 /lib/systemd/systemd-logind
 604 root    10  -10 9500 4708 3940 S  0.0  0.0  0:00.14 ovsdb-server /etc/openvswitch/conf.
 672 root    10  -10 13356 13080 11616 S  0.0  0.0  0:00.00 ovs-vsitchd unix:/var/run/openvswi
 698 root    20   0  5972 4132 3368 S  0.0  0.0  0:00.09 /bin/login -p --
 706 ntp      20   0  74636 3972 3404 S  0.0  0.0  0:00.04 /usr/sbin/ntpd -p /var/run/ntpd.pid
 760 mininet  20   0  18448 9824 8352 S  0.0  0.0  0:00.06 /lib/systemd/systemd --user
 762 mininet  20   0  100M 2944 4 S  0.0  0.0  0:00.00 (sd-pan)
 767 mininet  20   0  11068 4920 3336 S  0.0  0.0  0:00.01 -bash
  
```

Εικόνα 7.7 Πεικόνιση εντολής htop για μέτρηση χρήσης CPU και Ram.

## 7.2.2 Μέτρηση RAM

Μπορούμε να μετρήσουμε τη χρήση της μνήμης RAM ενός δικτύου Mininet χρησιμοποιώντας την εντολή free, η οποία είναι ένα βοηθητικό πρόγραμμα Unix που εμφανίζει την ποσότητα της ελεύθερης και χρησιμοποιημένης μνήμης στο σύστημα.

Για να χρησιμοποιήσουμε την εντολή free για την παρακολούθηση της χρήσης RAM ενός δικτύου Mininet:

1. Εκκινούμε το δίκτυο Mininet χρησιμοποιώντας την τοπολογία και τη διαμόρφωση που επιθυμούμε.
2. Ανοίγουμε ένα νέο παράθυρο τερματικού και εκτελούμε την εντολή free για να εμφανίσουμε την τρέχουσα χρήση RAM του συστήματος.
3. Αναζητούμε τη στήλη που χρησιμοποιείται, η οποία δείχνει την ποσότητα της μνήμης RAM που χρησιμοποιείται αυτήν τη στιγμή από το σύστημα.
4. Αφήνουμε την εντολή να εκτελείται για όσο διάστημα θέλουμε να παρακολουθήσουμε τη χρήση της μνήμης RAM του δικτύου Mininet.

Αξίζει να σημειωθεί ότι όπως η παρακολούθηση της χρήσης της CPU, η παρακολούθηση της χρήσης RAM ενός δικτύου Mininet μπορεί να έχει αντίκτυπο στην απόδοση του ίδιου του δικτύου.

The screenshot shows a terminal window with a task manager (htop) running. The top part of the terminal displays system statistics: Mem [||||] 150M/28.96G, Supr [ 0K/353M], Tasks: 20, 7 thr: 1 running, Load average: 0.08 0.32 0.16, and Uptime: 00:03:51. Below this is a table of running processes with columns for PID, USER, PRI, NI, UIRT, RES, SHR, S, CPU%, MEM%, TIME+, and Command. The process mininet is highlighted in blue. At the bottom of the terminal, there are two boxes: a green box with a downward arrow labeled 'H εντολή free' and a red box with an upward arrow labeled 'H εντολή htop'. Below these boxes, the output of the 'free' command is shown:

```
mininet@mininet-vm:~$ free
              total        used        free      shared    buff/cache   available
Mem:           30334964      152752      30042936         688      139276      29858056
Swap:             361404              0           361404
```

Εικόνα 7.8 Σύγκριση εντολών free και Htop για την μέτρηση χρήσης μνήμης RAM.

### 7.2.3 Μέτρηση Καθυστέρησης

Για να μετρήσουμε την καθυστέρηση πακέτων στο Mininet, μπορούμε να χρησιμοποιήσουμε την εντολή `ping` για να στείλουμε πακέτα μεταξύ δύο κεντρικών υπολογιστών στο δίκτυο και να μετρήσουμε το χρόνο που χρειάζεται για τη μετάδοση και τη λήψη των πακέτων. Τα βήματα που πρέπει να ακολουθήσουμε:

1. Ξεκινάμε το δίκτυο Mininet με την επιθυμητή τοπολογία και διαμόρφωση.
2. Ανοίγουμε δύο τερματικά μέσω του PuTTY σε συνεργασία με το X-ming και ξεκινάμε ένα παράθυρο για κάθε κεντρικό υπολογιστή στον οποίο θέλουμε να μετρήσουμε την καθυστέρηση μεταξύ τους. Μπορούμε να το κάνουμε αυτό εκτελώντας την εντολή `xterm h1-h2` στην κονσόλα Mininet, αντικαθιστώντας τα `h1` και `h2` με τα ονόματα των κεντρικών υπολογιστών που θέλουμε να χρησιμοποιήσουμε.
3. Σε ένα από τα παράθυρα, εκτελούμε την εντολή `ping` για να στείλουμε πακέτα από τον έναν κεντρικό υπολογιστή στον άλλο. Για παράδειγμα, μπορούμε να εκτελέσουμε το `ping -c 10 h2` για να στείλετε 10 πακέτα στον κεντρικό υπολογιστή `h2`. Η έξοδος θα δείχνει τον χρόνο που χρειάζεται για να μεταδοθεί και να ληφθεί κάθε πακέτο.
4. Μετράμε την καθυστέρηση υπολογίζοντας τον μέσο χρόνο που χρειάζεται για να μεταδοθεί και να ληφθεί ένα πακέτο. Αυτό μπορεί να γίνει λαμβάνοντας το άθροισμα των χρόνων για όλα τα πακέτα και διαιρώντας με τον αριθμό των πακέτων που αποστέλλονται. Για παράδειγμα, αν στείλουμε 10 πακέτα και οι χρόνοι ήταν 1ms, 2ms, 3ms κ.λπ., τότε η μέση καθυστέρηση θα ήταν  $(1+2+3+...)/10$ .

Τέλος όπως είδαμε και στις προηγούμενες μετρήσεις έτσι εδώ οι μετρήσεις λανθάνοντος χρόνου μπορεί να επηρεαστούν από διάφορους παράγοντες, όπως η συμφόρηση δικτύου, οι περιορισμοί εύρους ζώνης και το φόρτο της CPU.

```

mininet@mininet-vm:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=87.7 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 87.658/87.658/87.658/0.000 ms
mininet> _

```

Εικόνα 7.9 Μέτρηση καθυστέρησης μετάδοσης πακέτου.

## 7.2.4 Μέτρηση Δημιουργίας – Καταστροφής τοπολογίας

Μπορούμε να μετρήσουμε τόσο τον χρόνο δημιουργίας όσο και το χρόνο καταστροφής μιας τοπολογίας στο Mininet τροποποιώντας την εντολή που χρησιμοποιείται για την εκκίνηση και τη διακοπή του δικτύου μέσω των εξής βημάτων:

1. Ανοίγουμε ένα παράθυρο τερματικού και μεταβαίνουμε στον κατάλογο όπου βρίσκεται το σενάριο τοπολογίας Mininet.
2. Πριν ξεκινήσουμε το δίκτυο Mininet, εκτελούμε την εντολή ημερομηνίας-date για να λάβουμε την τρέχουσα ημερομηνία και ώρα:

```

* Documentation: https://help.ubuntu.com/
mininet@mininet-vm:~$ date
Sat Apr 22 20:58:40 PDT 2023
mininet@mininet-vm:~$ _

```

Εικόνα 7.10 Εντολή date.

Αυτό θα εμφανίσει την τρέχουσα ημερομηνία και ώρα στο παράθυρο του τερματικού.

3. Εκτελούμε την εντολή για να ξεκινήσουμε το δίκτυο Mininet με την εντολή time και προσθέτουμε την τρέχουσα ημερομηνία και ώρα στην έξοδο χρησιμοποιώντας την εντολή tee:

```
time sudo mn --custom mytopo.py --topo mytopo --controller remote | tee output.txt
```

Εικόνα 7.11 Εντολή time.

Αυτή η εντολή ξεκινά το δίκτυο Mininet χρησιμοποιώντας το προσαρμοσμένο σενάριο τοπολογίας mytopo.py με την τοπολογία mytopo και τον απομακρυσμένο default ελεγκτή. Η εντολή time θα μετρήσει το χρόνο που χρειάζεται για να ολοκληρωθεί αυτή η εντολή και η εντολή tee θα αποθηκεύσει την έξοδο σε ένα αρχείο που ονομάζεται output.txt ενώ θα το εμφανίζει επίσης στο παράθυρο του τερματικού.

4. Μόλις ξεκινήσει το δίκτυο Mininet, περιμένουμε να σταθεροποιηθεί και αυτό-ρυθμιστεί. Αυτό μπορεί να πάρει κάποιο χρόνο ανάλογα με την πολυπλοκότητα της εκάστοτε τοπολογίας.
5. Όταν επιθυμούμε να καταστρέψουμε το δίκτυο Mininet, εκτελούμε ξανά την εντολή ημερομηνίας-date για να λάβουμε την τρέχουσα ημερομηνία και ώρα.
6. Πατάμε ‘ Ctrl + D ’ για να σταματήσει το δίκτυο Mininet.
7. Η εντολή time θα δώσει τον χρόνο εκτέλεσης για την εντολή sudo mn και η εντολή tee θα αποθηκεύσει την έξοδο στο αρχείο output.txt. Αναζητάμε τη γραμμή που ξεκινά με πραγματικό-real:

```
1 real 0m5.239s
2 user 0m1.695s
3 sys 0m1.831s
```

Εικόνα 7.12 Εντολή time για μέτρηση χρόνο δημιουργίας τοπολογίας.

Ο χρόνος εκτέλεσης αναγράφεται στην πραγματική στήλη. Σε αυτό το παράδειγμα, ο χρόνος εγκατάστασης για την τοπολογία είναι περίπου 5.239 δευτερόλεπτα.

8. Ανοίγουμε το αρχείο output.txt για να βρούμε την έξοδο από το δίκτυο Mininet. Οι τελευταίες γραμμές της εξόδου θα πρέπει να δείχνουν τις εντολές που χρησιμοποιήθηκαν για την καταστροφή του δικτύου και τον χρόνο που χρειάστηκε για να ολοκληρωθεί:

```
1  *** Stopping 2 controllers
2  c0
3  c1
4  *** Stopping 8 links
5  .....
6  *** Stopping 3 switches
7  s1
8  s2
9  s3
10 *** Done
11 real    0m3.419s
12 user    0m0.009s
13 sys     0m0.016s
```

Εικόνα 7.13 Εντολή `time` για μέτρηση χρόνου καταστροφής τοπολογίας.

Ο χρόνος εκτέλεσης αναγράφεται στην πραγματική στήλη. Σε αυτό το παράδειγμα, ο χρόνος διάλυσης για την τοπολογία είναι περίπου 3,419 δευτερόλεπτα.

## 7.2.5 Μέτρηση Bandwidth

Μπορούμε να κάνουμε μέτρηση του bandwidth της κάθε τοπολογίας με τη χρήση της εντολής `iperf`. Απλά το μόνο που χρειάζεται είναι να μη πραγματοποιηθεί τερματισμός της εκάστοτε τοπολογίας και να παραμείνει ενεργή. Τότε μέσα στην εκτελούμενη τοπολογία γίνεται χρήση της εντολής `iperf` και λαμβάνουμε το αποτέλεσμα.

```
mininet@mininet-vm:~$ sudo mn --topo=linear,2,2
*** Creating network
*** Adding controller
*** Adding hosts:
h1s1 h1s2 h2s1 h2s2
*** Adding switches:
s1 s2
*** Adding links:
(h1s1, s1) (h1s2, s2) (h2s1, s1) (h2s2, s2) (s2, s1)
*** Configuring hosts
h1s1 h1s2 h2s1 h2s2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> iperf h1s2 h2s1
*** Iperf: testing TCP bandwidth between h1s2 and h2s1
*** Results: ['46.1 Gbits/sec', '46.1 Gbits/sec']
```

Εικόνα 7.14 Εντολή `iperf` για μέτρηση bandwidth.

## 7.3 Εκτέλεση πειραμάτων

Η εκτέλεση των πειραμάτων περιλαμβάνει την χρήση κατάλληλου κώδικα έτσι ώστε να δημιουργηθούν οι τοπολογίες στο Mininet. Ο κώδικας αυτός γράφεται σε γλώσσα Python και εισάγεται στο mininet με τη μορφή python scripts. Με τις κατάλληλες εντολές εκτελούνται τα scripts δημιουργείται η τοπολογία και αντλούνται τα στοιχεία που απαιτούνται από αυτήν. Όπως ήδη τονίστηκε η κλίμακα των πειραμάτων κυμαίνεται από 2 Switch και 2 Host αντίστοιχα και φτάνει μέχρι τον αριθμό 8192 και στα δύο αντίστοιχα με κλιμάκωση της δύναμης του 2.

### 7.3.1 Υλοποίηση τυχαίων τοπολογιών

Σε αυτή την υπο-ενότητα παρουσιάζεται ο κώδικας δημιουργίας τυχαίων τοπολογιών. Είναι γραμμένος σε Python και πάνω σε αυτόν προσαρμόζονται τα σενάρια των τυχαίων τοπολογιών. Εφόσον το Mininet τρέχει τον ελεγκτή POX που είναι συμβατός και με τη γλώσσα Python ( ο οποίος είναι ο προκαθορισμένος στο mininet) η δημιουργία επικοινωνίας με αυτόν γίνεται στις γραμμές 9-12 όπου γίνεται και επιλογή της θύρας που ακούει θύρα 6653 είτε 6633 ( φαίνονται και οι 2 θύρες για λόγο ότι αυτές είναι οι συνηθέστερες που χρησιμοποιούνται για την επικοινωνία με τον ελεγκτή και προφανώς αλλάζουν στο ίδιο νούμερο για επιτυχημένη επικοινωνία) και η εκκίνησή του αντίστοιχα στις γραμμές 54-56. Η ρύθμιση του βασικού πρωτοκόλλου δημιουργίας τοπολογιών γίνεται στις γραμμές 45-49 όπου ορίζεται το πρωτόκολλο δημιουργίας OpenFlow στη λειτουργία των switch. Τα χαρακτηριστικά του Erdős-Rényi φαίνονται στις γραμμές, 14-19 όπου ορίζεται ο αριθμός Switch , 21-26 όπου ορίζεται αριθμός Hosts , 28-43 όπου ορίζεται η συνδεσμολογία και πιθανότητα επιτυχούς σύνδεσης. Αρκεί η αλλαγή αυτών των αριθμών για την επιτυχή δημιουργία των τοπολογιών κλιμακωτά. Σαν p δίνεται μια τιμή στα πειράματα 0.8 η οποία δίνει μεγάλη πιθανότητα επιτυχίας αλλά εμπεριέχει και έναν ‘κίνδυνο’ της τάξεως του 20% να μην πραγματοποιηθεί επιτυχή σύνδεση μεταξύ κόμβων πράγμα που ρίχνει λίγο την αποδοτικότητα.

```

1 from mininet.net import Mininet
2 from mininet.node import RemoteController, OVSSwitch
3 from mininet.link import TCLink
4 from mininet.log import setLogLevel
5 import random
6
7 def topology():
8     # Create Mininet object
9     net = Mininet(controller=RemoteController, switch=OVSSwitch, link=TCLink)
10
11     # Add controller
12     c0 = net.addController('c0', controller=RemoteController, ip='127.0.0.1', port=6633)
13
14     # Add switches
15     num_switches = 5
16     switches = []
17     for i in range(num_switches):
18         switch = net.addSwitch('s{}'.format(i+1), dpid='0000000000000000{}'.format(i+1), cls=OVSSwitch, protocols='OpenFlow13')
19         switches.append(switch)
20
21     # Add hosts
22     num_hosts = 10
23     hosts = []
24     for i in range(num_hosts):
25         host = net.addHost('h{}'.format(i+1), ip='10.0.0.{}'.format(i+1))
26         hosts.append(host)
27
28     # Add links
29     p = 0.5 # Probability of a link between two nodes
30     for i in range(num_hosts):
31         for j in range(i+1, num_hosts):
32             if random.random() < p:
33                 net.addLink(hosts[i], hosts[j])
34
35     for i in range(num_switches):
36         for j in range(i+1, num_switches):
37             if random.random() < p:
38                 net.addLink(switches[i], switches[j])
39
40     for i in range(num_switches):
41         for j in range(num_hosts):
42             if random.random() < p:
43                 net.addLink(switches[i], hosts[j])
44
45     # Enable OFDP on switches
46     for switch in switches:
47         switch.cmd('ovs-vsctl set bridge', switch, 'protocols=OpenFlow13,OpenFlow14')
48         switch.cmd('ovs-vsctl set bridge', switch, 'other-config:datapath-id-{}'.format(switch.dpid))
49         switch.cmd('ovs-appctl -t ovs-ofctl ofp-discovery ' + switch.name + ' enable=true')
50
51     # Start Mininet
52     net.start()
53
54     # Start POX controller
55     pox_controller = net.addController(['pox_controller', controller=RemoteController, ip='127.0.0.1', port=6653])
56     pox_controller.start()
57
58     # Stop Mininet
59     net.stop()
60
61 if __name__ == '__main__':
62     setLogLevel('info')
63     topology()
64

```

Εικόνα 7.15 Κώδικας δημιουργίας τυχαίων τοπολογιών βάσει μοντέλου Erdős-Rényi.

## 7.3.2 Υλοποίηση τοπολογιών ισορροπημένου δένδρου

### 7.3.2.1 Υλοποίηση τοπολογιών ισορροπημένου δένδρου μέσω της βασικής ενσωμάτωσης τοπολογιών στο Mininet

Το mininet περιέχει ενσωματωμένη στις υβριδικές τοπολογίες την τοπολογία δέντρου και για αυτό το λόγο μπορούμε να την χρησιμοποιήσουμε μέσω των ενσωματωμένων εντελών στο λειτουργικό σύστημα όπως φαίνονται παρακάτω, αλλά χρειάζεται προσοχή ώστε το δέντρο να γίνεται ισορροπημένο. Η βασική εντολή είναι της μορφής :

```
sudo mn -controller remote -topo=tree,depth=2,fanout=4
```

Στο πλαίσιο μιας τοπολογίας δέντρου στο Mininet, οι όροι "fanout" και "depth" αναφέρονται στη δομή και το μέγεθος του δέντρου.



- Fanout: Το fanout αντιπροσωπεύει τον αριθμό των παιδιών που έχει κάθε διακόπτης στο δέντρο. Καθορίζει τον παράγοντα διακλάδωσης σε κάθε επίπεδο του δέντρου. Για παράδειγμα, εάν το fanout είναι 2, κάθε μεταγωγέας στο δέντρο θα έχει δύο θυγατρικούς ακόμα. Μια υψηλότερη τιμή fanout οδηγεί σε ένα ευρύτερο και πιο διασυνδεδεμένο δέντρο.
- Depth: Το βάθος αντιπροσωπεύει τον αριθμό των επιπέδων ή των επιπέδων στο δέντρο. Καθορίζει πόσοι μεταγωγείς υπάρχουν από τον μεταγωγέα ρίζας μέχρι τους μεταγωγείς φύλλων. Το βάθος του δέντρου καθορίζει το συνολικό ύψος ή μήκος του δέντρου. Μια υψηλότερη τιμή βάθους οδηγεί σε μια πιο ψηλή δομή δέντρου.

Συνδυάζοντας τις παραμέτρους fanout και depth, μπορούμε να δημιουργήσουμε διάφορες τοπολογίες δέντρων με διαφορετικά χαρακτηριστικά. Για παράδειγμα, ένα δέντρο με fanout 2 και depth 3 θα έχει  $2^3 = 8$  μεταγωγείς φύλλων συνδεδεμένους σε έναν ενιαίο μεταγωγέα ρίζας, σχηματίζοντας ένα ισορροπημένο δυαδικό δέντρο.

### **7.3.2.2 Υλοποίηση τοπολογιών ισορροπημένου δένδρου μέσω της βασικής ενσωμάτωσης τοπολογιών στο Mininet**

Εδώ παρουσιάζεται ο κώδικας δημιουργίας τοπολογιών ισορροπημένου δένδρου μέσω κώδικα. Για τον ελεγκτή ισχύει ότι έχει ήδη δηλωθεί καθώς και για την επιλογή των θυρών και η δημιουργία επικοινωνίας με αυτόν γίνεται στις γραμμές 23-24 . Η ρύθμιση του βασικού πρωτοκόλλου δημιουργίας τοπολογιών γίνεται στις γραμμές 23 όπου ορίζεται το πρωτόκολλο δημιουργίας OpenFlow στη λειτουργία των switch. Τα χαρακτηριστικά της τοπολογίας φαίνονται στις γραμμές, 6-22 όπου ορίζεται το γενικότερο βάθος και εύρος της τοπολογίας δέντρου. Αρκεί η αλλαγή αυτών των αριθμών για την επιτυχή δημιουργία των τοπολογιών κλιμακωτά.

```

1  from mininet.topo import Topo
2  from mininet.net import Mininet
3  from mininet.cli import CLI
4  from mininet.node import RemoteController
5
6  class BalancedTreeTopo(Topo):
7      def __init__(self, depth=1, fanout=2):
8          super(BalancedTreeTopo, self).__init__()
9          self.create_tree(depth, fanout)
10
11     def create_tree(self, depth, fanout):
12         if depth == 0:
13             return
14
15         switch = self.addSwitch('s{}'.format(depth))
16
17         for i in range(fanout):
18             host = self.addHost('h{}_{}'.format(depth, i+1))
19             self.addlink(host, switch)
20             self.create_tree(depth-1, fanout)
21
22     topo = BalancedTreeTopo(depth=3, fanout=2)
23     net = Mininet(topo, controller=lambda name: RemoteController(name, ip='127.0.0.1', port=6633), switch=OVSSwitch, build=False)
24     net.addController('pox_controller', controller=RemoteController, ip='127.0.0.1', port=6633)
25     net.build()
26     net.start()
27     CLI(net)
28     net.stop()

```

Εικόνα 7.16 Κώδικας δημιουργίας τοπολογιών ισορροπημένου δέντρου.

### 7.3.3 Υλοποίηση γραμμικών τοπολογιών

#### 7.3.3.1 Υλοποίηση γραμμικών τοπολογιών μέσω της βασικής ενσωμάτωσης τοπολογιών στο Mininet

Η γραμμική τοπολογία ανήκει στις βασικές τοπολογίες και υπάρχει ήδη προεγκατεστημένη στο Mininet και για αυτό το λόγο μπορούμε να την χρησιμοποιήσουμε μέσω των ενσωματωμένων εντελών στο λειτουργικό σύστημα όπως φαίνονται παρακάτω. Η βασική εντολή είναι της μορφής:

```
sudo mn -controller remote -topo=linear2,4
```

Ενδεικτικά η παραπάνω εντολή δημιουργεί την τοπολογία που ορίζεται ως γραμμική και στον πρώτο αριθμό ορίζουμε τον αριθμό των switch ενώ στον δεύτερο αριθμό ορίζουμε τον αριθμό των hosts ,έστω 2 Switch και 4 Hosts στο παράδειγμά μας και προσαρμόζουμε ανάλογα την κλίμακα του δικτύου που θέλουμε να δημιουργήσουμε.

### 7.3.3.2 Υλοποίηση γραμμικών τοπολογιών μέσω κώδικα

Μπορούμε βέβαια να δημιουργήσουμε και γραμμικές τοπολογίες μέσω αρχείων Python αλλά στη συγκεκριμένη περίπτωση είναι πολύ πιο εύκολο και απλό να χρησιμοποιήσουμε τις ενσωματωμένες εντολές του Mininet. Αν πάντως δημιουργούταν ένα αρχείο θα έμοιαζε κάπως έτσι και θα έχριζε ιδιαίτερης προσοχής στη συνδεσμολογία ούτως ώστε να είναι βέβαιη η δημιουργία γραμμικών τοπολογιών.

```
1 from mininet.net import Mininet
2 from mininet.node import RemoteController, OVSSwitch
3 from mininet.link import TCLink
4 from mininet.log import setLogLevel
5
6 def topology():
7     # Create Mininet object
8     net = Mininet(controller=RemoteController, switch=OVSSwitch, link=TCLink)
9
10    # Add controller
11    c0 = net.addController('c0', controller=RemoteController, ip='127.0.0.1', port=6633)
12
13    # Add switches
14    s1 = net.addSwitch('s1', dpid='0000000000000001', cls=OVSSwitch, protocols='OpenFlow13')
15    s2 = net.addSwitch('s2', dpid='0000000000000002', cls=OVSSwitch, protocols='OpenFlow13')
16    s3 = net.addSwitch('s3', dpid='0000000000000003', cls=OVSSwitch, protocols='OpenFlow13')
17    s4 = net.addSwitch('s4', dpid='0000000000000004', cls=OVSSwitch, protocols='OpenFlow13')
18
19    # Add hosts
20    h1 = net.addHost('h1', ip='10.0.0.1')
21    h2 = net.addHost('h2', ip='10.0.0.2')
22    h3 = net.addHost('h3', ip='10.0.0.3')
23    h4 = net.addHost('h4', ip='10.0.0.4')
24
25    # Add links
26    net.addLink(h1, s1)
27    net.addLink(h2, s1)
28    net.addLink(h3, s1)
29    net.addLink(h4, s1)
30    net.addLink(s1, s2)
31    net.addLink(s2, s3)
32    net.addLink(s3, s4)
33
34    # Enable OFDP on switches
35    for switch in [s1, s2, s3, s4]:
36        switch.cmd('ovs-vsctl set bridge', switch, 'protocols=OpenFlow13,OpenFlow14')
37        switch.cmd('ovs-vsctl set bridge', switch, 'other-config:datapath-id={}'.format(switch.dpid))
38        switch.cmd('ovs-appctl -t ovs-ofctl ofp-discovery ' + switch.name + ' enable=true')
39
40    # Start Mininet
41    net.start()
42
43    # Start POX controller
44    pox_controller = net.addController('pox_controller', controller=RemoteController, ip='127.0.0.1', port=6653)
45    pox_controller.start()
46
47    # Stop Mininet
48    net.stop()
49
50 if __name__ == '__main__':
51     setLogLevel('info')
52     topology()
```

Εικόνα 7.17 Κώδικας δημιουργίας γραμμικών τοπολογιών.

### 7.3.4 Εκκίνηση ελεγκτή POX στο Mininet

Για να ενεργοποιηθεί ο ελεγκτής POX στο Mininet, μπορούμε να ακολουθήσουμε τα παρακάτω βήματα:

1. Εκκινούμε το Mininet ανοίγοντας ένα παράθυρο τερματικού και εισάγοντας την εντολή «sudo mn».
2. Στο Mininet CLI, δημιουργούμε μια τοπολογία και εκχωρούμε τον ελεγκτή πληκτρολογώντας την εντολή `net = Mininet(topo=topo,`

controller=RemoteController). Εδώ, το topo είναι η τοπολογία που θέλουμε να χρησιμοποιήσετε και το RemoteController καθορίζει ότι θέλετε να χρησιμοποιήσετε έναν εξωτερικό ελεγκτή.

3. Ξεκινάμε τον ελεγκτή POX ανοίγοντας ένα νέο παράθυρο τερματικού και μεταβαίνοντας στον κατάλογο όπου είναι εγκατεστημένο το POX. Στη συνέχεια, πληκτρολογούμε την εντολή  

```
«./pox.py log.level --DEBUG openflow.of_01 --port=<PORT> forwarding.l2_learning,»
```

 όπου <PORT> είναι ο αριθμός θύρας OpenFlow που θέλουμε να χρησιμοποιήσουμε. Δίνεται προσοχή ο αριθμός θύρας να ταιριάζει με τον αριθμό θύρας που καθορίζεται στην εντολή Mininet.
4. Στο Mininet CLI, δημιουργούμε την τοπολογία εισάγοντας την εντολή `net.start()`.
5. Δοκιμάζεται η συνδεσιμότητα του δικτύου κάνοντας ping μεταξύ των κεντρικών υπολογιστών.

Ακολουθώντας αυτά τα βήματα, να μπορούμε να ενεργοποιήσουμε τον ελεγκτή POX στο Mininet και να ελέγξουμε τη συνδεσιμότητα του δικτύου . Μπορεί να χρειαστεί να διαμορφωθεί επιπλέον ο ελεγκτής POX ώστε να ταιριάζει στις συγκεκριμένες απαιτήσεις δικτύου [34].

### 7.3.5 Προσαρμογή τοπολογιών σε διαφορετικό πρωτόκολλο και ελεγκτή

Στο παράρτημα στο τέλος της παρούσας μεταπτυχιακής εργασίας παρατίθενται κώδικες εναλλαγής πρωτοκόλλου δημιουργίας τοπολογιών. Στην παρούσα εργασία έμφαση δόθηκε στα πειράματα κυρίως στο πρωτόκολλο OFDP. Παρόλα αυτά στο κεφάλαιο 10 θα αναφερθούν μελλοντικές ιδέες περαιτέρω ανάλυσης και εμπάθυνσης της παρούσας έρευνας για αναλυτικότερη μελέτη όλων των ελεγκτών.

## 7.4 Σύλλογή γενικών αποτελεσμάτων

Σε αυτή την ενότητα θα παρουσιαστούν οι στατιστικοί πίνακες των δεδομένων που συλλέχθηκαν από τα παραπάνω πειράματα. Ο ελεγκτής είναι ο POX και το πρωτόκολλο δημιουργίας τοπολογιών είναι το OFDP. Αξίζει να σημειωθεί πως το κάθε πείραμα εκτελέστηκε περίπου χίλιες φορές για να εξακριβωθεί το ποσοστό ακρίβειας των αποτελεσμάτων και οι αποκλίσεις ήτανε ελάχιστες και σύμφωνες με το αναμενόμενο μοτίβο. Οπότε τα αποτελέσματα που παρουσιάζονται είναι ο συνολικός μέσος όρος.

- Τυχαία

CPU (%)	MEMORY (MB)	SWITC H	HOST S	BW (Gbps)	SETUP TIME (sec)	TEAR TIME (sec)
1,9	150	2	2	41	0,092	0,085
2,6	170	4	4	42	0,145	0,136
6,4	210	8	8	48	0,326	0,413
11,2	250	16	16	48	1,256	1,646
13,5	290	32	32	47	2,719	6,167
18,1	330	64	64	38	12,752	22,390
22,4	390	128	128	38	18,393	29,712
27,8	440	256	256	36	26,715	39,513
33,6	625	512	512	33	39,212	58,004
39,2	1100	1024	1024	32	57,454	74,981
44,5	2000	2048	2048	28	83,757	119,046
47,3	3500	4096	4096	26	183,908	244,901
62,9	6800	8192	8192	27	274,483	368,271

Πίνακας 7.2 : Αποτελέσματα πειραμάτων με χρήση τυχαίων τοπολογιών.

- Γραμμική

CPU (%)	MEMORY (MB)	SWITC H	HOST S	BW (Gbps)	SETUP TIME (sec)	TEAR TIME (sec)
1,2	300	2	2	45	0,098	0,067
1,9	340	4	4	44	0,182	0,224
4,8	380	8	8	49	0,295	0,313
9,3	420	16	16	47	0,542	0,621
10,3	480	32	32	48	0,894	1,128
14,5	560	64	64	41	1,889	2,359
19,3	680	128	128	39	3,319	4,858
23,6	1050	256	256	38	6,822	7,254
28,1	1680	512	512	39	14,841	18,952
33,6	2200	1024	1024	37	33,713	39,701
38,4	3500	2048	2048	36	55,915	62,113
42,5	7300	4096	4096	33	98,009	127,989
53,6	12300	8192	8192	31	181,411	229,410

Πίνακας 7.3 : Αποτελέσματα πειραμάτων με χρήση γραμμικών τοπολογιών.

- Ισορροπημένου δένδρου

CPU (%)	MEMORY (MB)	SWITC H	HOST S	BW (Gbps)	SETUP TIME (sec)	TEAR TIME (sec)
3,2	180	2	2	43	0,150	0,141
4,5	220	4	4	42	0,265	0,181
8,9	270	8	8	38	0,429	0,284

14,7	380	16	16	44	1,854	1,678
17,6	490	32	32	48	3,535	3,280
21,2	600	64	64	41	6,614	7,252
24,8	710	128	128	43	8,325	10,053
29,1	930	256	256	40	17,783	19,993
36,1	1450	512	512	39	26,977	41,900
44,9	2580	1024	1024	41	56,672	77,451
52,4	4310	2048	2048	37	128,334	168,513
59,9	8200	4096	4096	38	190,985	212,717
74,4	15200	8192	8192	30	260,511	332,557

Πίνακας 7.4 : Αποτελέσματα πειραμάτων με χρήση τοπολογιών ισορροπημένου δέντρου.

### 7.4.1 Συλλογή αποτελεσμάτων καθυστέρησης

Η μέτρηση του latency θα γίνει με διαφορετικό και τρόπο καθώς σε κάθε δίκτυο μετριέται σε παρόμοιες συνθήκες με σταθερό μέγεθος πακέτου αυξάνοντας τον αριθμό των πακέτων και βλέποντας πως αυτό επηρεάζει το δίκτυο. Γίνεται συλλογή του μέσου και τον συνολικού χρόνου μεταφοράς. Το μέγεθος του κάθε πακέτου ορίζεται ως 1024Byte(1KB) και θα γίνει εκτέλεση εξομοιώσεων με αριθμό πακέτων αντίστοιχα [1,10,50,100,500] . Σε προηγούμενες έρευνες είχε παρατηρηθεί σαν όριο ορθής χρήσης τουMininet περίπου τα 600 πακέτα σε αυτό το μέγεθος πακέτου.

PACKET NUMBER	TREE AVERAGE LATENCY(ms)	LINEAR AVERAGE LATENCY(ms)	ERDOS RENYI AVERAGE LATENCY (ms)
1	0,048	0,018	0,013
10	0,053	0,027	0,016
50	0,044	0,026	0,018
100	0,031	0,023	0,021
500	0,041	0,015	0,022
TOTAL AVERAGE	0,0434	0,0218	0,0181

Πίνακας 7.5 : Αποτελέσματα καθυστέρησης πειραμάτων σε σύνολο των τοπολογιών.

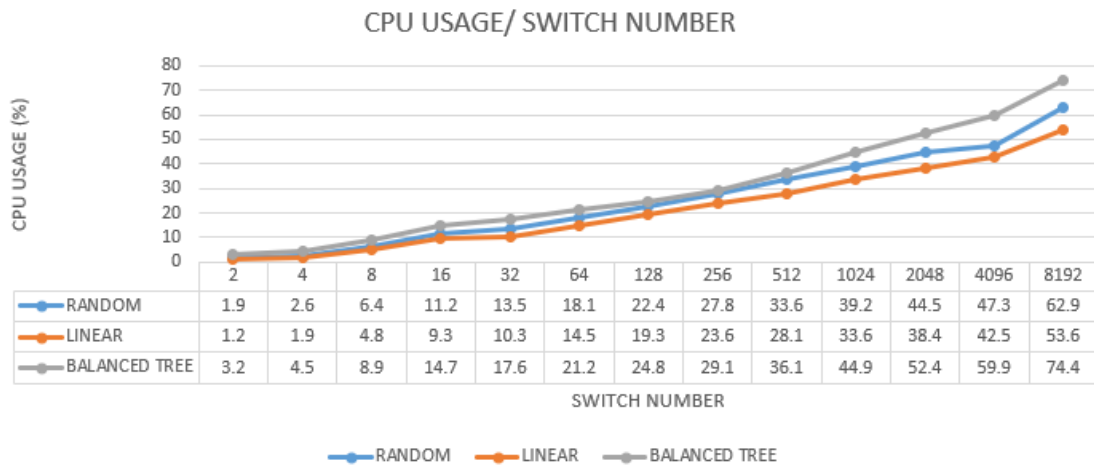
## 8.ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Τα αποτελέσματα που λήφθηκαν στο προηγούμενο κεφάλαιο μετατρέπονται με κατάλληλο τρόπο σε διαγράμματα. Στον κάθετο άξονα γίνεται κατανομή του κάθε στοιχείου που μελετήθηκε ( CPU , RAM , Bandwidth , Setup Time , Tear Time , Latency) ενώ στον οριζόντιο άξονα υπάρχει ο αριθμός των switch που χρησιμοποιήθηκαν και έτσι βγαίνουν συμπεράσματα με βάση το πλήθος των Switch. Στο διάγραμμα Latency στον οριζόντιο άξονα ο αριθμός των Switch αντικαθίσταται από τον αριθμό των πακέτων.

### 8.1 Ανάλυση CPU

Στο παρακάτω διάγραμμα φαίνεται το ποσοστό χρήσης του επεξεργαστή σε σχέση με τον αριθμό των Switch. Το προφανές λογικό συμπέρασμα είναι ότι όσο αυξάνεται ο αριθμός των Switch τόσο μεγαλύτερη είναι η απασχόληση του επεξεργαστή.

- Σε ότι αφορά τις τρεις τοπολογίες αυτή που λαμβάνει περισσότερους πόρους από τον επεξεργαστή είναι η balanced-tree.
- Αυτό γίνεται λόγω της περιπλοκότητας της τοπολογίας και της δημιουργίας δένδρου-κλαδιών που στην ουσία είναι το διαθέσιμο μονοπάτι .
- Είναι πολύ κοντά με την τυχαία τοπολογία .
- Όσο αυξάνεται ο αριθμός των Switch τόσο και κορυφώνεται η χρήση επεξεργαστή και αυξάνει η τοπολογία ισορροπημένου δένδρου τη διαφορά της.
- Ως κορυφή στο διάγραμμα στα 8192 Switch όπου η χρήση επεξεργαστή φτάνει το 74.4% μια διαφορά της τάξης του 11.5% από την τυχαία τοπολογία και του 20.8% από τη γραμμική.
- Η γραμμική παρουσιάζει μικρότερη χρήση λόγω της απλότητας της συνδεσμολογίας της.
- Αξίζει να παρατηρηθούν κάποιες αστάθειες στην τυχαία τοπολογία τις οποίες δημιουργεί η πιθανότητα μη επιτυχούς σύνδεσης δύο κόμβων.

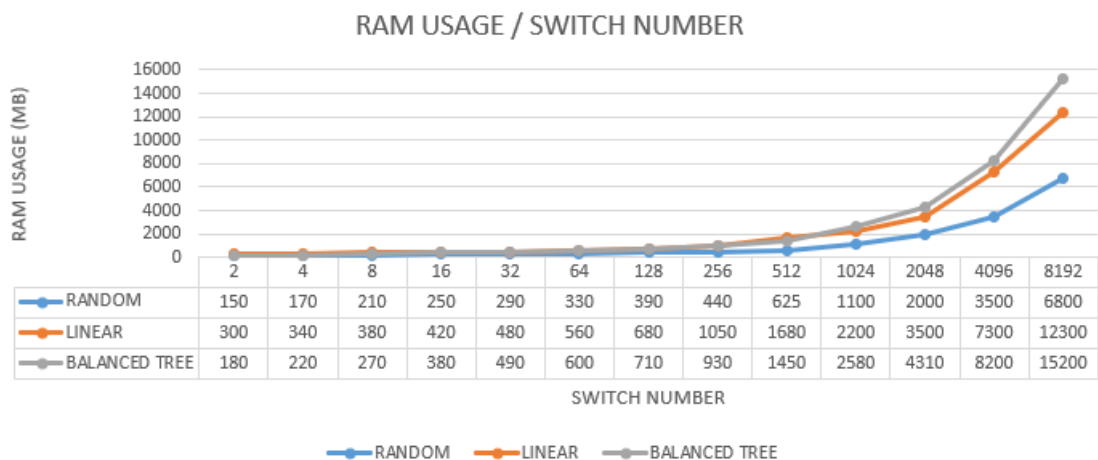


Πίνακας 8.1: Συγκριτικό διάγραμμα χρήσης CPU για τις τρεις τοπολογίες.

## 8.2 Ανάλυση RAM

Στο παρακάτω διάγραμμα καταγράφεται η χρήση μνήμης RAM σε MB. Στην περίπτωση της μνήμης η κατάσταση και βάσει αρχιτεκτονικής Mininet αλλά και των τοπολογιών. Αυτό ερμηνεύεται ως εξής:

- Η γραμμική τοπολογία αρχικά λόγω της αρχιτεκτονικής της παρότι χρησιμοποιεί λιγότερο CPU από το ισορροπημένο δένδρο φορτώνει περισσότερο τη μνήμη.
- Η διαφορά μεταξύ του εξισώνεται όσο αυξάνεται ο αριθμός των Switch και αντιστρέφεται σε μεγάλο φόρτο.
- Η τυχαία τοπολογία λόγω της παραμετροποίησής της καθώς και το ότι έχει φτιαχτεί βελτιωμένα εκ των υστέρων και εισάγεται στο Mininet παρουσιάζει πολύ μικρότερη κατανάλωση μνήμης RAM της τάξης του 45-55% σε σύγκριση με τις άλλες δύο τοπολογίες.



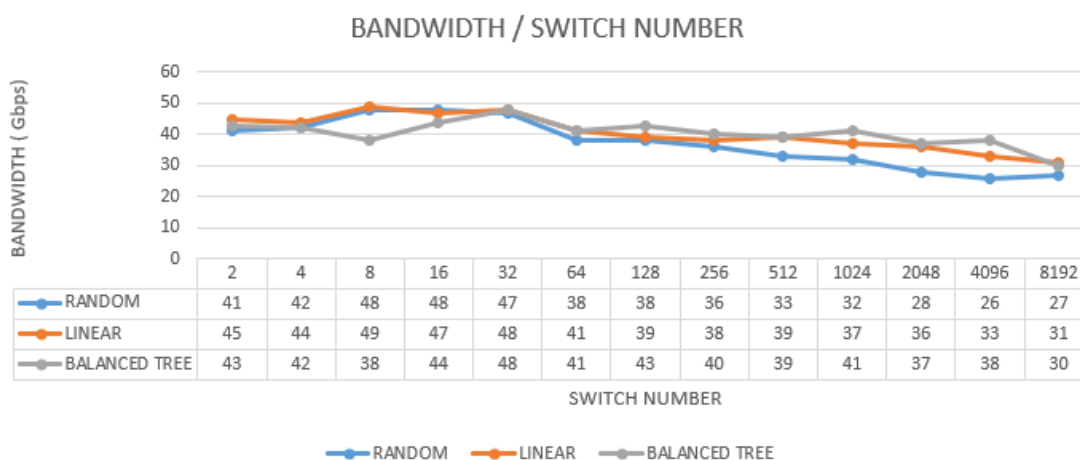
Πίνακας 8.2: Συγκριτικό διάγραμμα χρήσης RAM για τις τρεις τοπολογίες.



### 8.3 Ανάλυση Bandwidth

Ο πίνακας του Bandwidth δείχνει το διαθέσιμο εύρος δικτύου σε κάθε σύνδεση. Παρουσιάζει ενδιαφέρον καθώς εδώ φαίνονται πιο καθαρά και τα χαρακτηριστικά των τριών τοπολογιών καθώς και οι αδυναμίες τους.

- Παρατηρούνται πολλές εναλλασσόμενες κορυφές στο διάγραμμα
- Όσο αυξάνεται το μέγεθος του δικτύου τόσο και προφανώς μειώνεται το εύρος του
- Η τοπολογία του ισορροπημένου δένδρου παρουσιάζει εναλλαγές καθώς αλλάζει δομή το δένδρο που δημιουργείται.
- Η γραμμική τοπολογία παρουσιάζεται πιο σταθερή και μειώνεται γραμμικά το διαθέσιμο εύρος όσο αυξάνεται το μέγεθος του δικτύου.
- Η τυχαία τοπολογία παρουσιάζει το μικρότερο διαθέσιμο εύρος λόγω της πολυπλοκότητας και της πιθανότητας των μη επιτυχών συνδέσεων μεταξύ των κόμβων της.
- Παρατηρούμε ότι σε καμία τοπολογία η κορύφωση δε γίνεται στο μικρότερο μέγεθος δικτύου αλλά σε ένα σχετικά μικρό μέγεθος.
- Αυτό το φαινόμενο συμβαίνει διότι παρότι υπάρχει υψηλό διαθέσιμο εύρος σε μικρό δίκτυο, όταν το δίκτυο είναι πολύ μικρό δεν μπορεί να εκμεταλλευτεί πλήρως τις δυνατότητές του.

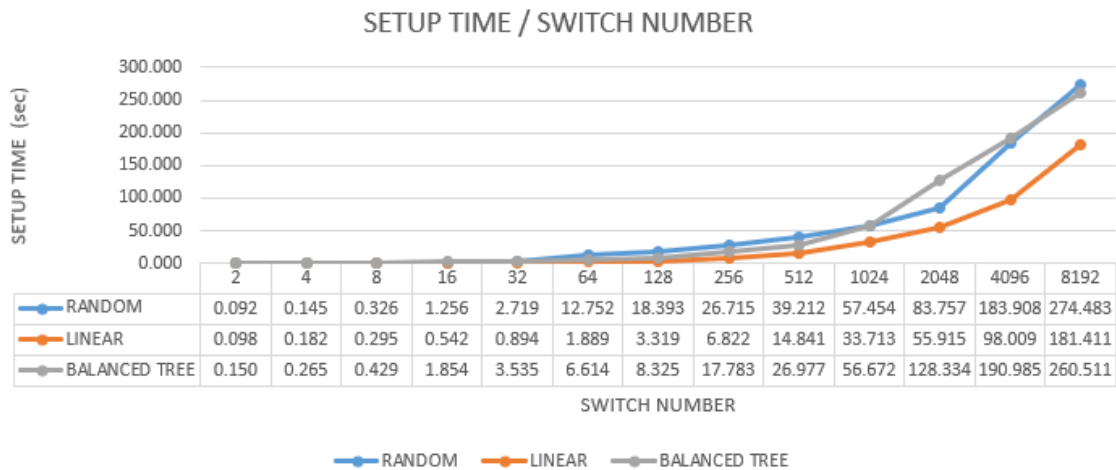


Πίνακας 8.3: Συγκριτικό διάγραμμα Bandwidth για τις τρεις τοπολογίες.

## 8.4 Ανάλυση Setup Time

Ο χρόνος δημιουργίας μιας τοπολογίας μετρείται σε δευτερόλεπτα από τη στιγμή που θα ξεκινήσει να εκτελείται η εντολή δημιουργίας της.

- Η τυχαία τοπολογία μαζί με την τοπολογία ισορροπημένου δένδρου παρουσιάζουν τον μεγαλύτερο χρόνο δημιουργίας μιας τοπολογίας.
- Οριακά η τυχαία τοπολογία είναι αυτή που χρειάζεται τον περισσότερο χρόνο.
- Η γραμμική τοπολογία παρουσιάζεται μειωμένη αισθητά σε μεγάλο ποσοστό.
- Ενδεικτικά σε πολύ μεγάλα δίκτυα η διαφορά της γραμμικής είναι περίπου 30-40%.



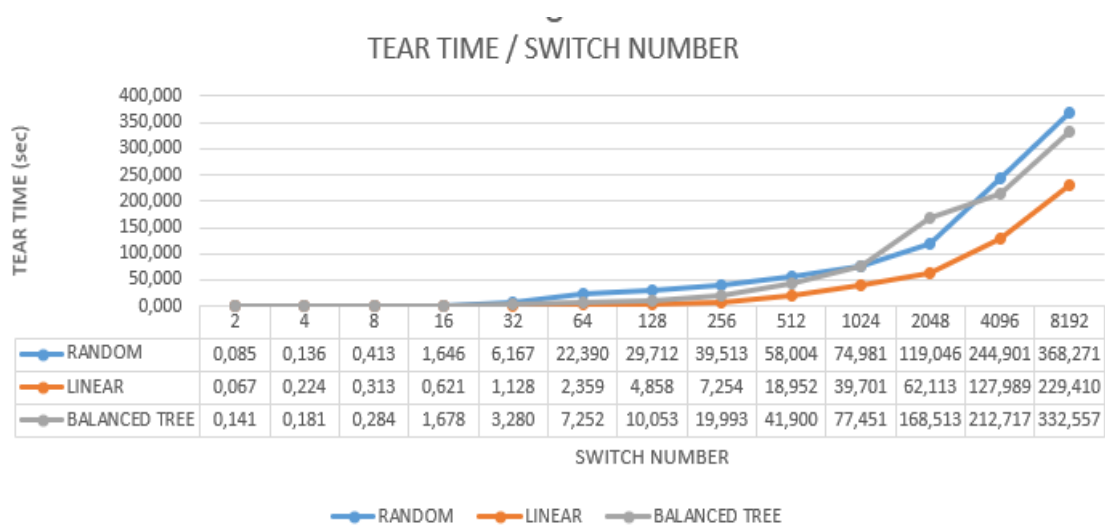
Πίνακας 8.4: Συγκριτικό διάγραμμα setup time για τις τρεις τοπολογίες.

## 8.5 Ανάλυση Tear Time

Ο χρόνος καταστροφής μιας τοπολογίας μετρείται σε δευτερόλεπτα από τη στιγμή που θα ξεκινήσει να εκτελείται η εντολή καταστροφής της.

- Αναμενόμενα όπως και στη δημιουργία η Linear έχει λιγότερο χρόνο.
- Στα μεγάλα δίκτυα παρουσιάζει περίπου 30-40% καλύτερο χρόνο όπως και στη δημιουργία.
- Αποδεικνύονται πάλι οι τυχαίες και ισορροπημένου δένδρου οι πιο χρονοβόρες.
- Κορυφή υψηλότερου χρόνου έχουμε στην τυχαία τοπολογία.
- Παρατηρούνται κάποιες αποκλίσεις στην γραμμικότητα του χρόνου στην τοπολογία ισορροπημένου δένδρου.
- Αυτές υπάρχουν λόγω της αλλαγής του δένδρου.

- Αστάθειες υπάρχουν φυσιολογικά και στις τυχαίες τοπολογίες.

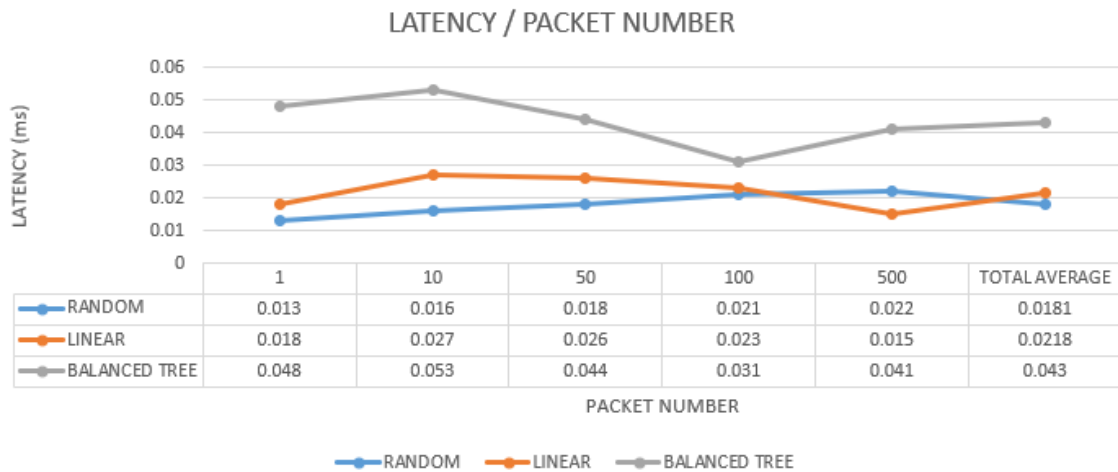


Πίνακας 8.5: Συγκριτικό διάγραμμα tear time για τις τρεις τοπολογίες.

## 8.6 Ανάλυση Latency

Στη μέτρηση Latency γίνεται καταμέτρηση του χρόνου μετάδοσης ενός πακέτου πληροφορίας από έναν κόμβο του δικτύου σε έναν άλλον. Αυτός ο χρόνος μετριέται σε ms. Το μέγεθος του πακέτου πληροφοριών ορίστηκε σε 1024 Byte = 1Kilobyte και το όριο του αριθμού των μεταφερόμενων πακέτων σε 500 καθότι σε προηγούμενες έρευνες είχε παρατηρηθεί αυτός ο αριθμός ως το όριο αξιόπιστων και επιτυχών μετρήσεων στο Mininet. [15]

- Το Ισορροπημένο δένδρο έχει τη μεγαλύτερη καθυστέρηση
- Παρόμοιες καθυστερήσεις παρουσιάζονται στις άλλες δύο τοπολογίες.
- Τουλάχιστον διπλάσια η καθυστέρηση στην τοπολογία ισορροπημένου δένδρου.



Πίνακας 8.6: Συγκριτικό διάγραμμα latency για τις τρεις τοπολογίες.

## 9. ΣΥΜΠΕΡΑΣΜΑΤΑ

Η δικτύωση που ορίζεται από λογισμικό (SDN) εμφανίστηκε ως μια μετασχηματιστική προσέγγιση στην αρχιτεκτονική και τη διαχείριση του δικτύου. Αποσυνδέοντας το επίπεδο ελέγχου από το επίπεδο δεδομένων και συγκεντρώνοντας τον έλεγχο του δικτύου μέσω λογισμικού, το SDN παρέχει πολλά οφέλη και έχει επιπτώσεις σε διάφορους κλάδους.

### Τα βασικά συμπεράσματα για το SDN έχουν ως εξής:

- **Ενισχυμένη ευελιξία δικτύου:** Το SDN επιτρέπει στους οργανισμούς να παρέχουν γρήγορα, να διαμορφώνουν και να τροποποιούν υπηρεσίες δικτύου μέσω λογισμικού, οδηγώντας σε βελτιωμένη ευελιξία δικτύου. Επιτρέπει τη δυναμική κατανομή των πόρων του δικτύου, καθιστώντας ευκολότερη την προσαρμογή στις μεταβαλλόμενες επιχειρηματικές ανάγκες και τα πρότυπα κίνησης ενός δικτύου.
- **Απλοποιημένη διαχείριση δικτύου:** Το SDN συγκεντρώνει τη διαχείριση δικτύου μέσω ενός ελεγκτή που διαχειρίζεται από λογισμικό, παρέχοντας ένα ενιαίο σημείο ελέγχου και εποπτείας. Αυτό απλοποιεί τη διαχείριση του δικτύου, μειώνει την πολυπλοκότητα και ενισχύει τις δυνατότητες αντιμετώπισης προβλημάτων.
- **Επεκτασιμότητα και ευελιξία:** Το SDN προσφέρει επεκτασιμότητα αφαιρώντας τη λειτουργικότητα του δικτύου από το υποκείμενο υλικό. Οι οργανισμοί μπορούν να κλιμακώσουν τα δίκτυά τους πιο εύκολα,

προσθέτοντας ή ανακατανέμοντας πόρους ανάλογα με τις ανάγκες. Επιπλέον, το SDN επιτρέπει την ευελιξία στην ανάπτυξη νέων υπηρεσιών και εφαρμογών χωρίς σημαντικές αλλαγές στην υποδομή.

- Προγραμματισμός δικτύου: Το SDN επιτρέπει τον προγραμματισμό δικτύου, επιτρέποντας στους διαχειριστές να αυτοματοποιούν τις λειτουργίες του δικτύου και να ελέγχουν τη συμπεριφορά του δικτύου μέσω λογισμικού. Αυτή η δυνατότητα προγραμματισμού διευκολύνει την ανάπτυξη καινοτόμων εφαρμογών και υπηρεσιών που μπορούν να αλληλεπιδρούν άμεσα με το δίκτυο.
- Βελτιωμένη ασφάλεια: Το SDN παρέχει βελτιωμένες δυνατότητες ασφάλειας αξιοποιώντας τον κεντρικό έλεγχο και τον προγραμματισμό. Οι πολιτικές ασφαλείας μπορούν να οριστούν και να επιβληθούν με συνέπεια σε ολόκληρο το δίκτυο, καθιστώντας ευκολότερο τον εντοπισμό και την απόκριση σε απειλές.
- Βελτιστοποίηση κόστους: Το SDN προσφέρει εξοικονόμηση κόστους μειώνοντας τις εξαρτήσεις υλικού και αυξάνοντας τη χρήση πόρων. Με τη δυνατότητα δυναμικού ελέγχου και κατανομής πόρων δικτύου, οι οργανισμοί μπορούν να βελτιστοποιήσουν την υποδομή τους, οδηγώντας σε καλύτερη απόδοση κόστους.
- Καινοτομία και ανάπτυξη “οικοσυστήματος”: Το SDN προωθεί την καινοτομία επιτρέποντας την ανάπτυξη νέων υπηρεσιών και εφαρμογών δικτύου. Ενθαρρύνει την ανάπτυξη ενός ‘οικοσυστήματος’ όπου οι πωλητές, οι προγραμματιστές και οι ερευνητές μπορούν να συνεργαστούν για να δημιουργήσουν νέες λύσεις και να προωθήσουν την πρόοδο στη δικτύωση.
- SD-WAN και Συνδεσιμότητα Cloud: Το SDN διαδραματίζει κρίσιμο ρόλο στην υιοθέτηση δικτύων ευρείας περιοχής που ορίζονται από λογισμικό (SD-WAN) και στη σύνδεση δικτύων εσωτερικής εγκατάστασης σε περιβάλλοντα cloud. Απλοποιεί τη διαχείριση των κατανεμημένων δικτύων, παρέχει καλύτερη ορατότητα και έλεγχο και βελτιώνει τη συνδεσιμότητα με τις υπηρεσίες cloud.

### **Προκλήσεις και ζητήματα**

- Ενώ το SDN προσφέρει σημαντικά οφέλη, παρουσιάζει επίσης προκλήσεις. Αυτά περιλαμβάνουν τη συνδυασμένη λειτουργία μεταξύ διαφορετικών λύσεων SDN.
- Ανησυχίες για την ασφάλεια που σχετίζονται με τη συγκέντρωση του ελέγχου
- Την ανάγκη για εξειδικευμένο προσωπικό για τη διαχείριση και τη λειτουργία περιβαλλόντων SDN.
- Ο προσεκτικός σχεδιασμός, οι δοκιμές και η συνεργασία με έμπειρους προμηθευτές είναι απαραίτητα για να ξεπεραστούν αυτές οι προκλήσεις.

### **Πρωτόκολλα SDN:**

- Τα πρωτόκολλα SDN διαδραματίζουν κρίσιμο ρόλο στην υλοποίηση και λειτουργία περιβαλλόντων δικτύωσης που ορίζονται από λογισμικό (SDN). Αυτά τα πρωτόκολλα ορίζουν την επικοινωνία και την αλληλεπίδραση μεταξύ διαφορετικών στοιχείων μιας αρχιτεκτονικής SDN, διευκολύνοντας τον έλεγχο και τη διαχείριση του δικτύου.
- Το OpenFlow είναι ένα από τα πιο ευρέως διαδεδομένα πρωτόκολλα SDN. Παρέχει μια τυπική διεπαφή μεταξύ του επιπέδου ελέγχου και των συσκευών προώθησης (switch). Το OpenFlow επιτρέπει τον κεντρικό έλεγχο δικτύου διαχωρίζοντας τη λογική ελέγχου από τα switch και επιτρέποντας στον ελεγκτή να προγραμματίζει κανόνες προώθησης. Συνέβαλε καθοριστικά στην ανάπτυξη και την ανάπτυξη λύσεων SDN.

### **Ελεγκτές SDN:**

- Οι ελεγκτές SDN χρησιμεύουν ως η κεντρική νοημοσύνη των αρχιτεκτονικών δικτύων που ορίζονται από λογισμικό (SDN). Είναι υπεύθυνοι για τη διαχείριση και την ενορχήστρωση πόρων του δικτύου, τη διευκόλυνση της επικοινωνίας μεταξύ του επιπέδου ελέγχου και του επιπέδου δεδομένων και τη δυνατότητα προγραμματισμού δικτύου.

### Απόδοση Τοπολογιών:

- Μέσω των πειραματικών διαδικασιών μπορούμε να καταλάβουμε πως λειτουργεί καλύτερα το SDN καθώς και τη λειτουργία των ξεχωριστών τοπολογιών. Δημιουργούνται δίκτυα ευρείας κλίμακας και μελετώνται τα χαρακτηριστικά τους. Έτσι μέσα από τον παρούσα μεταπτυχιακή εργασία επιτυγχάνεται η κατανόηση των δικτυακών αυτών δομών σε πραγματικό χρόνο και το πως είναι δυνατή η αποτελεσματική εφαρμογή τους σε πραγματικό χρόνο.
- Η τυχαία τοπολογία πετυχαίνει βελτιωμένα αποτελέσματα λόγω της προσαρμοστικότητας και της δυνατότητας παραμετροποίησης.
- Στην τυχαία τοπολογία υπάρχει πάντα το ενδεχόμενο μη επιτυχούς σύνδεσης πράγμα που διαμορφώνει τις επιδόσεις με ανάλογη επιβάρυνση.
- Πέρα από την επιβάρυνση αυτό προσδίδει στις τυχαίες τοπολογίες και μια πιο "ρεαλιστική" εφαρμογή.
- Η γραμμική τοπολογία παραμένει απλή και για αυτό διατηρεί κορυφαίες επιδόσεις.
- Η τοπολογία ισορροπημένου δένδρου λόγω της αρχιτεκτονικής της όσο επιβαρύνεται και επεκτείνεται μειώνει τις επιδόσεις εξαιτίας της περιπλοκότητας της και των υπολογισμών που γίνονται για την επιτυχή δημιουργία του δένδρου.
- Η ανάπτυξη του πρωτοκόλλου OFDP οδήγησε σε βελτιωμένες επιδόσεις και ομαλή συνεργασία.
- Χάρη στον ελεγκτή POX και την συνεργασία με το πρωτόκολλο OFDP έγινε εφικτή η επέκταση του μεγέθους δικτύων SDN μέσω της παραμετροποίησης.

ΚΑΤΗΓΟΡΙΑ ΜΕΤΡΗΣΗΣ	ΚΑΛΥΤΕΡΗ ΤΟΠΟΛΟΓΙΑ
CPU	LINEAR
RAM	RANDOM
BANDWIDTH	LINEAR
SETUP TIME	LINEAR
TEAR TIME	LINEAR
LATENCY	RANDOM

Πίνακας 9.1 Πίνακας καλύτερης τοπολογίας ανά κατηγορία μέτρησης.

## **10. ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ**

Το περιεχόμενο της συγκεκριμένης μεταπτυχιακής εργασίας αποτελεί βασικό πυλώνα της έρευνας επί των δικτύων SDN και επέκταση ήδη υπαρχουσών ερευνών πάνω στον τομέα της πληροφορικής των δικτύων και των τηλεπικοινωνιών. Σε μελλοντικές έρευνες θα μπορούσε να επεκταθεί σε ακόμα μεγαλύτερη κλίμακα με την βοήθεια υπερυπολογιστών μεγάλων ακαδημαϊκών δομών έτσι ώστε να φανεί πως θα επηρέαζε το διαδίκτυο και τον κόσμο γενικότερα μια ολική στροφή της δικτύωσης ως προς το SDN. Με τους κατάλληλους διαθέσιμους πόρους θα ήτανε δυνατές ακόμα ρεαλιστικότερες προσομοιώσεις με σκοπό την απευθείας ενσωμάτωση, βελτίωση και προσαρμογή σταδιακά , αρχικά σε ακαδημαϊκές δομές και στη συνέχεια στην κοινωνία με σκοπό ένα ισχυρότερο παγκόσμιο δίκτυο το οποίο θα είναι αποδοτικότερο , αξιόπιστο και θα αντέχει τις συνεχώς αυξανόμενες ανάγκες της σύγχρονης κοινωνίας. Τέλος ως επέκταση των όσων μελετήθηκαν είναι εφικτός και ο συνδυασμός των ήδη ενεργών πρωτοκόλλων για τη δημιουργία ενός νέου βελτιωμένου.



## ΠΑΡΑΡΤΗΜΑ

*Κώδικας δημιουργίας Τυχαίων τοπολογιών.*

```
from mininet.net import Mininet
from mininet.node import RemoteController, OVSSwitch
from mininet.link import TCLink
from mininet.log import setLogLevel
import random

def topology():
    # Create Mininet object
    net = Mininet(controller=RemoteController, switch=OVSSwitch, link=TCLink)

    # Add controller
    c0 = net.addController('c0', controller=RemoteController, ip='127.0.0.1', port=6633)

    # Add switches
    num_switches = 5
    switches = []
    for i in range(num_switches):
        switch = net.addSwitch('s{}'.format(i+1), dpid='0000000000000000{}'.format(i+1),
cls=OVSSwitch, protocols='OpenFlow13')
        switches.append(switch)

    # Add hosts
    num_hosts = 10
    hosts = []
    for i in range(num_hosts):
        host = net.addHost('h{}'.format(i+1), ip='10.0.0.{}'.format(i+1))
        hosts.append(host)

    # Add links
    p = 0.5 # Probability of a link between two nodes
    for i in range(num_hosts):
        for j in range(i+1, num_hosts):
            if random.random() < p:
                net.addLink(hosts[i], hosts[j])

    for i in range(num_switches):
        for j in range(i+1, num_switches):
            if random.random() < p:
                net.addLink(switches[i], switches[j])

    for i in range(num_switches):
        for j in range(num_hosts):
            if random.random() < p:
                net.addLink(switches[i], hosts[j])

    # Enable OFDP on switches
```

```

for switch in switches:
    switch.cmd('ovs-vsctl set bridge', switch, 'protocols=OpenFlow13,OpenFlow14')
    switch.cmd('ovs-vsctl set bridge', switch, 'other-config:datapath-
id={}'.format(switch.dpid))
    switch.cmd('ovs-appctl -t ovs-ofctl ofp-discovery ' + switch.name + ' enable=true')

# Start Mininet
net.start()

# Start POX controller
pox_controller = net.addController('pox_controller', controller=RemoteController,
ip='127.0.0.1', port=6653)
pox_controller.start()

# Stop Mininet
net.stop()

if __name__ == '__main__':
    setLogLevel('info')
    topology()

```

*Κώδικας δημιουργίας Γραμμικών τοπολογιών*

```

from mininet.net import Mininet
from mininet.node import RemoteController, OVSSwitch
from mininet.link import TCLink
from mininet.log import setLogLevel

def topology():
    # Create Mininet object
    net = Mininet(controller=RemoteController, switch=OVSSwitch, link=TCLink)

    # Add controller
    c0 = net.addController('c0', controller=RemoteController, ip='127.0.0.1', port=6633)

    # Add switches
    s1 = net.addSwitch('s1', dpid='0000000000000001', cls=OVSSwitch,
protocols='OpenFlow13')
    s2 = net.addSwitch('s2', dpid='0000000000000002', cls=OVSSwitch,
protocols='OpenFlow13')
    s3 = net.addSwitch('s3', dpid='0000000000000003', cls=OVSSwitch,
protocols='OpenFlow13')
    s4 = net.addSwitch('s4', dpid='0000000000000004', cls=OVSSwitch,
protocols='OpenFlow13')

    # Add hosts
    h1 = net.addHost('h1', ip='10.0.0.1')
    h2 = net.addHost('h2', ip='10.0.0.2')

```

```

h3 = net.addHost('h3', ip='10.0.0.3')
h4 = net.addHost('h4', ip='10.0.0.4')

# Add links
net.addLink(h1, s1)
net.addLink(h2, s1)
net.addLink(h3, s1)
net.addLink(h4, s1)
net.addLink(s1, s2)
net.addLink(s2, s3)
net.addLink(s3, s4)

# Enable OFDP on switches
for switch in [s1, s2, s3, s4]:
    switch.cmd('ovs-vsctl set bridge', switch, 'protocols=OpenFlow13,OpenFlow14')
    switch.cmd('ovs-vsctl set bridge', switch, 'other-config:datapath-
id={}'.format(switch.dpid))
    switch.cmd('ovs-appctl -t ovs-ofctl ofp-discovery ' + switch.name + ' enable=true')

# Start Mininet
net.start()

# Start POX controller
pox_controller = net.addController('pox_controller', controller=RemoteController,
ip='127.0.0.1', port=6653)
pox_controller.start()

# Stop Mininet
net.stop()

if __name__ == '__main__':
    setLogLevel('info')
    topology()

```

*Κώδικας δημιουργίας τοπολογιών ισορροπημένου δέντρου*

```

from mininet.topo import Topo
from mininet.net import Mininet
from mininet.cli import CLI
from mininet.node import RemoteController

class BalancedTreeTopo(Topo):
    def __init__(self, depth=1, fanout=2):
        super(BalancedTreeTopo, self).__init__()
        self.create_tree(depth, fanout)

    def create_tree(self, depth, fanout):
        if depth == 0:

```

```

    return

    switch = self.addSwitch('s{}'.format(depth))

    for i in range(fanout):
        host = self.addHost('h{}_{}'.format(depth, i+1))
        self.addLink(host, switch)
        self.create_tree(depth-1, fanout)

topo = BalancedTreeTopo(depth=3, fanout=2)
net = Mininet(topo, controller=lambda name: RemoteController(name, ip='127.0.0.1',
port=6633), switch=OVSSwitch, build=False)
net.addController('pox_controller', controller=RemoteController, ip='127.0.0.1', port=6633)
net.build()
net.start()
CLI(net)
net.stop()

```

*Κώδικας δημιουργίας τοπολογιών με πρωτόκολλο BGP*

```

from mininet.net import Mininet
from mininet.node import OVSSwitch, Controller
from mininet.link import Link, TCLink
from mininet.log import setLogLevel

def topology():
    # Create Mininet object
    net = Mininet(controller=Controller, switch=OVSSwitch, link=TCLink)

    # Add controller
    c0 = net.addController('c0')

    # Add switches
    s1 = net.addSwitch('s1', protocols='OpenFlow13')
    s2 = net.addSwitch('s2', protocols='OpenFlow13')

    # Add hosts
    h1 = net.addHost('h1', ip='10.0.0.1')
    h2 = net.addHost('h2', ip='10.0.0.2')

    # Add links
    Link(h1, s1)
    Link(h2, s2)
    Link(s1, s2)

    # Start Mininet
    net.start()

```

```

# Configure BGP on switches
for switch in [s1, s2]:
    switch.cmd('ovs-vsctl set bridge', switch, 'protocols=OpenFlow13')
    switch.cmd('ovs-vsctl set-fail-mode', switch, 'secure')
    switch.cmd('ovs-vsctl add-port', switch, switch.name + '-eth0', '-- set interface',
switch.name + '-eth0', 'type=internal')
    switch.cmd('ip addr add', '169.254.0.1/24', 'dev', switch.name + '-eth0')
    switch.cmd('ip link set', switch.name + '-eth0', 'up')
    switch.cmd('echo 1 > /proc/sys/net/ipv4/ip_forward')
    switch.cmd('ip route add', '10.0.0.0/24', 'dev', switch.name + '-eth0')
    switch.cmd('bgpd -d -f', switch.name + '-bgpd.conf', '-g', switch.name, '-m', switch.name
+ '-rib')

# Display BGP information
for switch in [s1, s2]:
    print('BGP information for switch', switch.name)
    print(switch.cmd('vtysh -c "show ip bgp"'))

# Stop Mininet
net.stop()

if __name__ == '__main__':
    setLogLevel('info')
    topology()

```

*Κώδικας δημιουργίας τοπολογιών με πρωτόκολλο LLDP*

```

from mininet.net import Mininet
from mininet.node import OVSSwitch, Controller
from mininet.link import Link, TCLink
from mininet.log import setLogLevel

def topology():
    # Create Mininet object
    net = Mininet(controller=Controller, switch=OVSSwitch, link=TCLink)

    # Add controller
    c0 = net.addController('c0')

    # Add switches
    s1 = net.addSwitch('s1')
    s2 = net.addSwitch('s2')

    # Add hosts
    h1 = net.addHost('h1', ip='10.0.0.1')
    h2 = net.addHost('h2', ip='10.0.0.2')

    # Add links

```

```

Link(h1, s1)
Link(h2, s2)
Link(s1, s2)

# Start Mininet
net.start()

# Enable LLDP on switches
for switch in [s1, s2]:
    switch.cmd('ovs-vsctl set bridge', switch, 'other-config:lldp-enable=true')

# Display LLDP information
for switch in [s1, s2]:
    print('LLDP information for switch', switch.name)
    print(switch.cmd('ovs-appctl -t ovsdb-server ovsdb-client dump Interface'))

# Stop Mininet
net.stop()

if __name__ == '__main__':
    setLogLevel('info')
    topology()

```

*Κώδικας δημιουργίας τοπολογιών με πρωτόκολλο OVSDDB*

```

from mininet.net import Mininet
from mininet.node import Controller, OVSKernelSwitch
from mininet.cli import CLI
from mininet.link import Link

def create_topology():
    net = Mininet(controller=Controller, switch=OVSKernelSwitch)
    c0 = net.addController('c0')

    # Add switches and hosts
    s1 = net.addSwitch('s1', dpid="0000000000000001")
    h1 = net.addHost('h1')

```

```
h2 = net.addHost('h2')

# Add links
net.addLink(h1, s1)
net.addLink(h2, s1)

# Set OVSDDB manager on switches
s1.cmd('ovs-vsctl set-manager tcp:6632')

# Start the network
net.start()

# Start the controller
net.get('c0').start()

# Configure hosts with IP addresses
h1.cmd('ip addr add 10.0.0.1/24 dev h1-eth0')
h2.cmd('ip addr add 10.0.0.2/24 dev h2-eth0')

# Open a Mininet command prompt
CLI(net)

# Stop the network
net.stop()

if __name__ == '__main__':
    create_topology()
```

*Κώδικας δημιουργίας τοπολογιών με πρωτόκολλο Netconf*

```

from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSSwitch
from mininet.cli import CLI
from mininet.link import Link

def create_topology():
    net = Mininet(controller=RemoteController, switch=OVSSwitch)
    c0 = net.addController('c0', ip='127.0.0.1', port=6633)

    # Add switches and hosts
    s1 = net.addSwitch('s1')
    h1 = net.addHost('h1')
    h2 = net.addHost('h2')

    # Add links
    net.addLink(h1, s1)
    net.addLink(h2, s1)

    # Enable NETCONF on switches
    s1.cmd('ovs-vsctl set-manager tcp:6640')

    # Start the network
    net.start()

    # Start the controller
    net.get('c0').start()

    # Configure hosts with IP addresses
    h1.cmd('ip addr add 10.0.0.1/24 dev h1-eth0')
    h2.cmd('ip addr add 10.0.0.2/24 dev h2-eth0')

    # Open a Mininet command prompt
    CLI(net)

    # Stop the network
    net.stop()

if __name__ == '__main__':
    create_topology()

```



*Κώδικας δημιουργίας τοπολογιών με πρωτόκολλο SNMP*

```
from mininet.net import Mininet
from mininet.node import Controller, OVSKernelSwitch
from mininet.cli import CLI
from mininet.link import Link

def create_topology():
    net = Mininet(controller=Controller, switch=OVSKernelSwitch)
    c0 = net.addController('c0')

    # Add switches and hosts
    s1 = net.addSwitch('s1')
    h1 = net.addHost('h1')
    h2 = net.addHost('h2')

    # Add links
    net.addLink(h1, s1)
    net.addLink(h2, s1)

    # Enable SNMP on switches
    s1.cmd('snmpd -Lf /dev/null -p 161 -c public')

    # Start the network
    net.start()

    # Start the controller
    net.get('c0').start()

    # Configure hosts with IP addresses
    h1.cmd('ip addr add 10.0.0.1/24 dev h1-eth0')
    h2.cmd('ip addr add 10.0.0.2/24 dev h2-eth0')

    # Open a Mininet command prompt
    CLI(net)

    # Stop the network
    net.stop()

if __name__ == '__main__':
    create_topology()
```

*Κώδικας δημιουργίας τοπολογιών με πρωτόκολλο LISP*

```
from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSKernelSwitch
from mininet.cli import CLI
from mininet.link import Link

def create_topology():
    net = Mininet(controller=RemoteController, switch=OVSKernelSwitch)
    c0 = net.addController('c0', ip='127.0.0.1', port=6633)

    # Add switches and hosts
    s1 = net.addSwitch('s1')
    h1 = net.addHost('h1')
    h2 = net.addHost('h2')

    # Add links
    net.addLink(h1, s1)
    net.addLink(h2, s1)

    # Set LISP configuration on switches
    s1.cmd('echo 1 > /proc/sys/net/ipv4/ip_forward')
    s1.cmd('sysctl net.ipv4.conf.all.rp_filter=0')
    s1.cmd('sysctl net.ipv4.conf.default.rp_filter=0')
    s1.cmd('ovs-vsctl set bridge s1 other_config="lisp-config=true"')

    # Start the network
    net.start()

    # Start the controller
    net.get('c0').start()

    # Configure hosts with IP addresses
    h1.cmd('ip addr add 10.0.0.1/24 dev h1-eth0')
    h2.cmd('ip addr add 10.0.0.2/24 dev h2-eth0')

    # Open a Mininet command prompt
    CLI(net)
```

```
# Stop the network
net.stop()

if __name__ == '__main__':
    create_topology()
```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

### ΕΠΙΣΤΗΜΟΝΙΚΕΣ ΑΝΑΦΟΡΕΣ

- [1] A. Nayak, A. Reimers, N. Feamster, and R. Clark. Resonance: Dynamic access control in enterprise networks. In Proc. Workshop: Research on Enterprise Networking, Barcelona, Spain, Aug. 2009
- [2] A. Voellmy and P. Hudak. Nettle: Functional reactive programming of OpenFlow networks. In Proc. Workshop on Practica
- [3] A. Zacharis, S. V. Margariti, E. Stergiou and C. Angelis, "Performance evaluation of topology discovery protocols in software defined networks," *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Heraklion, Greece, 2021, pp. 135-140, doi: 10.1109/NFV-SDN53031.2021.9665006.
- [4] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. ElasticTree: Saving energy in data center networks. Apr. 2010.
- [5] Brandon Heller, Rob Sherwood, Nick McKeown, "The Controller Placement Problem", HotSDN'12, 2012
- [6] D. Erickson et al. A demonstration of virtual machine mobility in an OpenFlow network, Aug. 2008. Demo at ACM SIGCOMM.
- [7] D. Erickson. The Beacon OpenFlow controller. In Proc. HotSDN, Aug. 2013.
- [8] Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky and Steve Uhlig, Member, "Software-Defined Networking: A Comprehensive Survey", <https://arxiv.org/abs/1406.0440>, 2014
- [9] Ian F. Akyildiz, Ahyoung Lee, Pu Wang, Min Luo, Wu Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks", Elsevier Computer Networks 71 (2014) 1–30
- [10] K. Greene. TR10: Software-defined networking. MIT Technology Review, March/April 2009. <http://www2.technologyreview.com/article/412194/tr10-software-defined-networking/>.

- [11] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker. Ethane: Taking control of the enterprise. In ACM SIGCOMM '07, 2007.
- [12] M. Guo and P. Bhattacharya, "Controller Placement for Improving Resilience of Software-Defined Networks," *2013 Fourth International Conference on Networking and Distributed Computing*, Los Angeles, CA, USA, 2013, pp. 23-27, doi: 10.1109/ICNDC.2013.15.
- [13] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. NOX: Towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110, July 2008.
- [14] N. Handigol, M. Flajslik, S. Seetharaman, N. McKeown, and R. Johari. Aster\*x: Load-balancing as a network primitive. In *ACLD '10: Architectural Concerns in Large Datacenters*, 2010.
- [15] N. V. Oikonomou, S. V. Margariti, E. Stergiou and D. Liarokapis, "Performance Evaluation of Software-Defined Networking Implemented on Various Network Topologies," *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, Preveza, Greece, 2021, pp. 1-6, doi: 10.1109/SEEDA-CECNSM53056.2021.9566213.
- [16] Pritesh Ranjan, Pankaj Pande, Ramesh Oswal, Zainab Qurani, Rajneeshkaur Bedi, "A Survey of Past, Present and Future of Software Defined Networking", Volume 2, Issue 4, *International Journal of Advance Research in Computer Science and Management Studies*, 2014
- [17] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar. Can the production network be the testbed? In *Proc. 9th USENIX OSDI*, Vancouver, Canada, Oct. 2010
- [18] R. Wang, D. Butnariu, and J. Rexford. OpenFlow-based server load balancing gone wild. In *Hot-ICE*, Mar. 2011.
- [19] Rodriguez-Natal, Alberto & Marc, Portoles-Comeras & Ermagan, Vina & Lewis, Darrel & Farinacci, Dino & Maino, Fabio & Cabello, Albert. (2015). LISP: a southbound

SDN protocol?. Communications Magazine, IEEE. 53. 201-207.

10.1109/MCOM.2015.7158286.

[20] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hlzle, S. Stuart, and A. Vahdat. B4: Experience with a globally deployed software defined WAN. In ACM SIGCOMM, Aug. 2013.

[21] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker. Onix: A distributed control platform for large-scale production networks. In OSDI, volume 10, pages 1–6, 2010.

[22] X. Zhao, S. S. Band, S. Elnaffar, M. Sookhak, A. Mosavi and E. Salwana, "The Implementation of Border Gateway Protocol Using Software-Defined Networks: A Systematic Literature Review," in IEEE Access, vol. 9, pp. 112596-112606, 2021, doi: 10.1109/ACCESS.2021.3103241.

ΣΥΝΔΕΣΜΟΙ

[23] Airbus technologies <https://www.airbus.com/en/newsroom/news/2018-06-airbus-technologies-take-the-spotlight-at-one-of-the-worlds-largest-defence>

[24] Cisco Applications <https://www.checkpoint.com/cyber-hub/cloud-security/what-is-cisco-application-centric-infrastructure-aci-security/>

[25] Cisco Corporation <https://www.cisco.com/site/us/en/index.html>

[26] Contrail <https://www.juniper.net/us/en/products/sdn-and-orchestration/contrail/cloud-native-contrail-networking.html>

[27] Engineering education SDN <https://www.section.io/engineering-education/openflow-sdn/>

[28] Enterprise storage Forum <https://www.enterprisestorageforum.com>

[29] Floodlight OpenFlow Controller. <http://floodlight.openflowhub.org/>.

[30] Gephi <https://gephi.org>

[31] HPE <https://www.hpe.com/us/en/home.html>

- [32] Huawei forum <https://forum.huawei.com/enterprise/en/index.html>
- [33] Microsoft <https://www.microsoft.com>
- [34] Mininet OS <http://mininet.org>
- [35] Necam <https://www.necam.com/sdn/>
- [36] Network Switch <https://www.cloudflare.com/learning/network-layer/what-is-a-network-switch/>
- [37] Nicira. It's time to virtualize the network, 2012. <http://nicira.com/en/network-virtualization-platform>.
- [38] ON.Lab. ONOS: Open network operating system, 2013. <http://tinyurl.com/pjs9eyw>
- [39] Open Daylight <http://www.opendaylight.org/>.
- [40] Open Networking Foundation <https://www.opennetworking.org/>.
- [41] POX Controller <http://www.noxrepo.org/pox/about-pox/>
- [42] PuTTY <https://www.putty.org>
- [43] SDN Architecture and role of OpenFlow  
<https://www.howtoforge.com/tutorial/software-defined-networking-sdn-architecture-and-role-of-openflow/>
- [44] SDN Market <https://www.gminsights.com/industry-analysis/software-defined-networking-sdn-market>
- [45] Spine-Leaf Architecture <https://www.arubanetworks.com/faq/what-is-spine-leaf-architecture/>
- [46] VirtualBox <https://www.virtualbox.org>
- [47] Visual Studio Code <https://code.visualstudio.com>
- [48] VMWARE Glossary <https://www.vmware.com/topics/glossary/content/software-defined-networking.html>

[49] VMWARE <https://www.vmware.com>

[50] WinSCP <https://winscp.net/eng/index.php>

[51] X-ming <http://www.straightrunning.com/XmingNotes/>



[Οπισθόφυλλο. Κενή σελίδα]