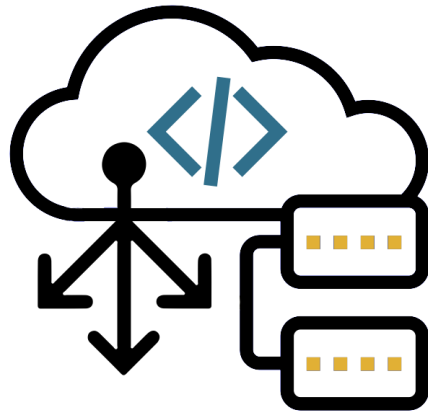




ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΑΛΓΟΡΙΘΜΩΝ ΕΞΙΣΟΡΡΟΠΗΣΗΣ ΚΙΝΗΣΗΣ ΣΕ ΔΙΚΤΥΑ ΚΑΘΟΡΙΣΜΕΝΑ ΑΠΟ ΛΟΓΙΣΜΙΚΟ



ΤΟΥ
Τάσου Κουρουνιάδη

Επιβλέπων Καθηγητής: Αναπληρωτής Καθηγητής Παναγιώτης Σαρηγιαννίδης

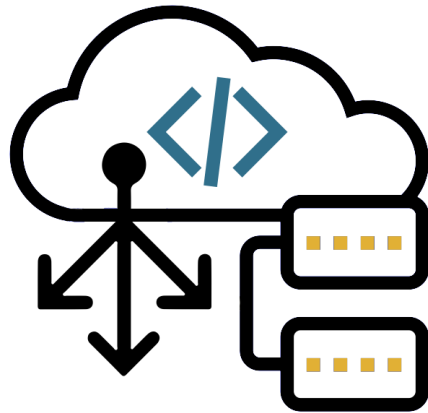
Φεβρουάριος 2024, Κοζάνη



HELLENIC DEMOCRACY
UNIVERSITY OF WESTERN MACEDONIA
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL
& COMPUTER ENGINEERING

THESIS

COMPARISON OF LOAD-BALANCING ALGORITHMS IN SDN ENVIRONMENTS



of

Tasos Kourouniadis

Supervisor: Associate Professor Panagiotis Sarigiannidis

February 2024, Kozani



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο **“ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΑΛΓΟΡΙΘΜΩΝ ΕΞΙΣΟΡΡΟΠΗΣΗΣ ΚΙΝΗΣΗΣ ΣΕ ΔΙΚΤΥΑ ΚΑΘΟΡΙΣΜΕΝΑ ΑΠΟ ΛΟΓΙΣΜΙΚΟ”** καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Σαρηγιαννίδη Παναγιώτη αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Ονοματεπώνυμο Φοιτητή & Επιβλέποντα, Έτος, Πόλη

Copyright (C) Τάσος Κουρουσιάδης, Παναγιώτης Σαρηγιαννίδης, 2024, Κοζάνη

Υπογραφή Φοιτητή:

Περίληψη

Είναι ευρέως αποδεκτό ότι τα Δίκτυα Καθορισμένα από Λογισμικό (Software Defined Networks - SDN) έχουν επιδείξει σημαντικά πλεονεκτήματα σε διάφορους τομείς σε σχέση με τα παραδοσιακά δίκτυα. Ωστόσο, η ανισορροπία φόρτου είναι ένας τομέας στον οποίο ενδέχεται να αντιμετωπίσουν προκλήσεις. Η ανισόρροπη διανομή φόρτου μπορεί να επηρεάσει σημαντικά την απόδοση ενός δικτύου σε περιβάλλοντα SDN. Για να αντιμετωπιστεί αυτό το ζήτημα, οι τομείς της μηχανικής λογισμικού και δικτύων προσπαθούν να προτείνουν στρατηγικές εξισορρόπησης φόρτου (load balancing) για να επιτύχουν την επιθυμητή απόδοση των δικτύων αυτών.

Ο κύριος στόχος οποιουδήποτε πάροχου τηλεπικοινωνιών και υπηρεσιών νέφους, είναι να μεγιστοποιήσει τη χρήση της υποδομής του δικτύου του. Οι παραδοσιακοί αλγόριθμοι δρομολόγησης όπως ο αλγόριθμος Dijkstra, είναι ανεπαρκείς πλέον για να ανταποκριθούν στις απαιτήσεις των σύγχρονων εφαρμογών.

Επομένως, οι αλγόριθμοι πολλαπλών μονοπατιών (multipath routing algorithms) προσφέρουν μια λύση στην πρόκληση της βελτιστοποίησης των δικτυακών πόρων, παρέχοντας ομοιόμορφη κατανομή της κίνησης σε πολλαπλά μονοπάτια.

Αυτή η διπλωματική εργασία έχει ως στόχο να ερευνήσει διάφορους αλγόριθμους εξισορρόπησης φόρτου για την κίνηση σε Δίκτυα Καθορισμένα από Λογισμικό, προκειμένου να εντοπίσει την πιο αποτελεσματική προσέγγιση βάσει συγκεκριμένων συνθηκών και περιορισμών, όπως οι τοπολογίες των δικτύων και διάφορα περιβάλλοντα.

Abstract

It is widely accepted that Software Defined Network (SDN) environments, have demonstrated and produced significant advantages in various areas over conventional networks. However, load imbalance is a notable area in which they may face challenges. Uneven load distribution can significantly affect network performance in SDN environments. To tackle this issue, the fields of software and network engineering, strive to propose load-balancing strategies to achieve the desired performance of these networks.

The primary goal of any telecommunications and cloud service provider is to maximize network infrastructure utilization. Traditional routing algorithms, such as Dijkstra's, are inadequate to meet the demands of modern applications.

Therefore, multipath routing algorithms offer a solution to optimizing network resources, by evenly distributing traffic across multiple paths, utilizing the numerous links found in modern networks.

This diploma thesis aims to investigate various load-balancing algorithms for network traffic, to identify the most effective approach based on specific conditions and limitations, such as network topologies and different environments.

Είμαι ευγνώμων που μπορώ να πω ότι όλα φτάνουν σε ένα τέλος με αισιόδοξους ρυθμούς. Τα τελευταία χρόνια η πορεία μου προς ολοκλήρωση του πτυχίου, δεν έπαψε να είναι γεμάτη με απροσδόκητες στιγμές γεμάτες γνώση και εκπλήξεις.

Πρώτα από όλους θα ήθελα να ευχαριστήσω την οικογένεια μου, για την υποστήριξη που μου πρόσφερε σε όλο το ακαδημαϊκό μου ταξίδι. Στην συνέχεια ευχαριστώ θερμά τους φίλους και τους ανθρώπους που γνώρισα, για τις εμπειρίες και στιγμές που μοιραστήκαμε. Με την δική μου σειρά θα ήθελα να ευχαριστήσω τον κ. Παναγιώτη Σαρηγιαννίδη, για την εμπιστοσύνη που μου είχε, και την επίβλεψη της διπλωματικής μου εργασίας. Τέλος θα ήθελα ιδιαιτέρως να ευχαριστήσω τον Υποψήφιο Διδάκτορα του Πανεπιστημίου Δυτικής Μακεδονίας, Αθανάσιο Λιατίφη για τον βοήθεια που μου πρόσφερε.

Ανυπομονώ για τις προκλήσεις που ακολουθούν..

Περιεχόμενα

Περίληψη	i
Abstract	ii
Περιεχόμενα	iv
Λίστα Εικόνων	v
Λίστα Σχημάτων	vi
Λίστα Ακρωνυμίων	viii
1. Εισαγωγή	1
1.1 Το πρόβλημα	1
1.2 Κίνητρα και Στόχοι.....	1
1.3 Δομή και Σύνοψη.....	2
2. Δίκτυα Καθορισμένα από Λογισμικό (SDN)	3
2.1 Επίπεδα των Δικτύων Υπολογιστών	3
2.2 Περιορισμοί των Παραδοσιακών Δικτύων	5
2.3 Αρχιτεκτονική Παραδοσιακών Δικτύων	8
2.4 Δίκτυα Καθορισμένα από Λογισμικό.....	9
2.4.1 Βασικές Αρχές των SDN.....	10
2.4.2 Αρχιτεκτονική Δικτύων Καθορισμένων από Λογισμικό	12
2.4.3 Πρωτόκολλο OpenFlow	17
2.5 Επόμενη Γενιά Δικτύων Καθορισμένων από Λογισμικό	22
2.5.1 Αρχιτεκτονική NG-SDN.....	24
2.5.2 Γλώσσα προγραμματισμού P4	27
3. Εξισορροπητές Φόρτου	33
3.1 Η Εξέλιξη των Εξισορροπητών Φόρτου.....	34
3.2 Κατηγορίες Εξισορροπητών Φόρτου.....	36
3.3 Νέα Γενιά Εξισορροπητών Φόρτου	37
4. Μεθοδολογία & Υλοποίηση	39
4.1 ONOS	39
4.2 Mininet	40
4.3 Stratum	42
4.4 Αλγόριθμοι εξισορόπισης φόρτου.....	43
4.5 Υλοποίηση.....	48
5. Αποτελέσματα	55
5.1 Τοπολογία 2x2.....	56
5.2 Τοπολογία 3x4.....	57
5.3 Τοπολογία 7x8.....	58
5.4 Τοπολογία RedIris Spain	59
5.5 Συγκεντρωτικά Αποτελέσματα.....	60
6. Συμπεράσματα & Μελλοντικές Επεκτάσεις	61
6.1 Συμπεράσματα	61
6.2 Μελλοντικές Επεκτάσεις.....	62
Βιβλιογραφικές Αναφορές	63

Λίστα Πινάκων

Πίνακας 1 Δίκτυα Καθορισμένα από Λογισμικό VS Παραδοσιακά Δίκτυα	11
Πίνακας 2 Γνωστοί Ελεγκτές Δικτύων Καθοριζόμενων από Λογισμικό.....	14
Πίνακας 3 Κύρια στοιχεία μιας εγγραφής ομάδας στον πίνακα ομάδας.....	18
Πίνακας 4 Τα κριτήρια αντιστοίχισης σε μια εγγραφή Openflow	19
Πίνακας 5 Εφαρμογές και μικροϋπηρεσίες του ONOS.....	49

Λίστα Εικόνων

Εικόνα 1 Αρχιτεκτονική Δικτύων Υπολογιστών.....	5
Εικόνα 2 Ο Κύκλος Ζωής των Συστημάτων Δικτύου	6
Εικόνα 3 Αρχιτεκτονική Παραδοσιακών Δικτύων.....	9
Εικόνα 4 Αρχιτεκτονική Δικτύων Καθοριζόμενων από Λογισμικό	15
Εικόνα 5 Αρχιτεκτονική Μεταγωγέα Καθορισμένου από Λογισμικό OpenFlow.....	17
Εικόνα 6 Κύρια στοιχεία μιας εγγραφής OpenFlow	19
Εικόνα 7 Η αλληλουχία πινάκων OpenFlow σε ένα μεταγωγέα SDN.....	20
Εικόνα 8 Η Διαδικασία αντιστοίχισης πακέτων στο OpenFlow	21
Εικόνα 9 Δυνατότητες και Οφέλη SDN επόμενης γενιάς	23
Εικόνα 10 Αρχιτεκτονική NG-SDN	25
Εικόνα 11 Δομή προγράμματος P4	31
Εικόνα 12 Προγραμματισμός Συσκευής με P4 και P4Runtime.....	32
Εικόνα 13 Ένα δίκτυο με εξισορροπητές φόρτου.....	33
Εικόνα 14 Η Εξέλιξη των Εξισοροπητών Φόρτου	35
Εικόνα 15 Αρχιτεκτονική του μONOS.....	40
Εικόνα 16 Το λογισμικό Mininet	41
Εικόνα 17 Αλγόριθμος Cheetah χωρίς κατάσταση	46
Εικόνα 18 Αλγόριθμος Cheetah με κατάσταση	47
Εικόνα 19 Μεταγλώττιση αρχείου .p4.....	50

Εικόνα 20 Τοπολογία 1. 2x2.....	51
Εικόνα 21 Τοπολογία 2. 3x4.....	51
Εικόνα 22 Τοπολογία 3. 7x8.....	52
Εικόνα 23 Τοπολογία 4. RedIris Spain.....	53
Εικόνα 24 Εκκίνηση περιβάλλοντος NG-SDN.....	54
Εικόνα 25 Λειτουργία δικτύου στο γραφικό περιβάλλον ONOS και Mininet	54
Εικόνα 26 Λήψη μετρήσεων	55
Εικόνα 27 Αποτελέσματα τοπολογίας 2x2.....	56
Εικόνα 28 Αποτελέσματα τοπολογίας 3x4.....	57
Εικόνα 29 Αποτελέσματα τοπολογίας 7x8.....	58
Εικόνα 30 Αποτελέσματα τοπολογίας RedIris Spain.....	59

Λίστα Ακρωνυμίων

- ACL** – Access Control List
- ADC** – Application Delivery Controllers
- API** – Application Programming Interface
- ASIC** – Application Specific Integrated Circuits
- BGP** – Border Gateway Protocol
- BMv2** – Behavioral Model version 2
- CPU** – Central Processing Unit
- DNS** – Domain Name Server
- DSL** – Domain Specific Language
- ECMP** – Equal-Cost Multi-Path
- FPGA** – Field Programmable Gate Arrays
- FTP** – File Transfer Protocol
- gNMI** – gRPC Network Management Interface
- gNOI** – gRPC Network Operations Interface
- gRPC** – google Remote Procedure Calls
- HTTP** – Hypertext Transfer Protocol
- HULA** – Hop-by-Hop Utilization-aware Load Balancing Architecture
- IoT** – Internet of Things
- IP** – Internet Protocol
- L4** – Layer 4 OSI
- L7** – Layer 7 OSI
- LAN** – Local Area Network
- NETCONF** – Network Configuration Protocol
- NG-SDN** – Next Generation-Software Defined Networks
- NIC** – Network Interface Card
- NPU** – Network Processing Unit
- ONF** – Open Networking Foundation
- OSI** – Open Systems Interconnection
- OSPF** – Open Shortest Path First
- PCC** – Per Connection Consistency

QoS – Quality of Service

REST – Representational State Transfer

RIP – Routing Information Protocol

RPC – Remote Procedure Call

SDN – Software Defined Networks

SNMP – Simple Network Management Protocol

TCP – Transport Control Protocol

VLAN – Virtual Local Area Network

WAN – Wide Area Network

Κεφάλαιο 1

Εισαγωγή

1.1 Το πρόβλημα

Ο κάθε πάροχος υπηρεσιών τηλεπικοινωνιών και νέφους, έχει ως κύριο στόχο την μέγιστη αξιοποίηση των δικτυακών υποδομών του. Λόγο της ραγδαίας ανάπτυξης στον τομέα των τηλεπικοινωνιών τις τελευταίες δεκαετίες, τα δίκτυα έχουν υποστεί τεράστια αύξηση στο φόρτο δικτυακής κίνησης. Για αυτό ευθύνη έχουν καταναλωτές και οργανισμοί, οι οποίοι βασίζονται όλο και περισσότερο στη δικτυακή συνδεσιμότητα για την πραγματοποίηση των καθημερινών τους αναγκών [1]. Ως εκ τούτου, τις ανάγκες των σύγχρονων εφαρμογών δεν είναι σε θέση να καλύψουν παραδοσιακοί αλγόριθμοι δρομολόγησης και κατασκευής μονοπατιού, με αποτέλεσμα πόροι όπως πολλαπλές ζεύξεις, μονοπάτια και εύρος ζώνης να μένουν ανεκμετάλλευτα.

1.2 Κίνητρα και Στόχοι

Μέχρι τώρα, στα σύγχρονα δίκτυα, η εξισορρόπηση φόρτου επιτυγχάνεται είτε μέσω ειδικών συσκευών είτε μέσω ενός εικονικού περιβάλλοντος. Ωστόσο, πολλές υπάρχουσες μέθοδοι δεν διαθέτουν την απαραίτητη επεκτασιμότητα και ευελιξία, έτσι ώστε να μπορούν να ανταποκριθούν στις εξελισσόμενες και αυστηρές απαιτήσεις των σύγχρονων δικτύων. Τα Δίκτυα Καθορισμένα από Λογισμικό (Software Defined Networks - SDN) [2] προσφέρουν μια πιο προηγμένη και ευέλικτη λύση για την εξισορρόπηση φόρτου δικτύων, επιτρέποντας τον κεντρικό έλεγχο και τη δυναμική προσαρμογή για τις μεταβαλλόμενες συνθήκες που υπάρχουν.

Μια λύση στο πρόβλημα της βέλτιστης αξιοποίησης των δικτυακών πόρων είναι οι αλγόριθμοι δρομολόγησης πολλαπλών μονοπατιών. Οι αλγόριθμοι αυτοί προσπαθούν να μοιράσουν ομοιόμορφα την κίνηση σε περισσότερο από ένα μονοπάτια αξιοποιώντας τις πολλαπλές ζεύξεις που υπάρχουν στα σύγχρονα δίκτυα.

Αντικείμενο της διπλωματικής εργασίας, είναι η μελέτη διάφορων αλγορίθμων εξισορρόπησης δικτυακής κίνησης, με σκοπό την εύρεση της καλύτερης προσέγγισης ανάλογα από τις συνθήκες και τους περιορισμούς που μπορεί να υπάρχουν.

Για τα SDN περιβάλλοντα έχει αναπτυχθεί μια γκάμα αλγορίθμων εξισορρόπησης κίνησης, ο κάθε ένας με τα δικά του πλεονεκτήματα και μειονεκτήματα. Η σύγκριση αυτών των αλγορίθμων είναι σημαντική έτσι ώστε να γίνει κατανόηση των επιδόσεων τους και να μπορεί να επιτευχθεί κατάλληλη επιλογή, ανάλογα με την τοπολογία δικτύου και τους περιορισμούς πόρων.

1.3 Δομή και Σύνοψη

Η διπλωματική εργασία ως σύνολο αποτελείται από έξι κεφάλαια. Από αυτό το σημείο και μετά ακολουθεί το Κεφάλαιο δύο, όπου επικεντρώνεται και αναλύει τα Δίκτυα Καθορισμένα από Λογισμικό. Παρουσιάζει τα μειονεκτήματα των παραδοσιακών δικτύων και τους λόγους στροφής της τεχνολογίας προς την νέα γενιά SDN. Ακολούθως παρουσιάζεται η νέα γενιά δικτύων και οι κύριες τεχνολογίες τους, που έχουν σχέση με την διπλωματική εργασία. Ο τομέας των SDN είναι πολύ μεγάλος, οπότε έγινε σημαντική προσπάθεια στο να επιλεγθούν προσεχτικά τα θέματα και οι τεχνολογίες που θα αναλυθούν χωρίς να γίνει υπερφόρτωση πληροφορίας.

Στο Κεφάλαιο τρία αρχικά γίνεται μια ιστορική αναδρομή για τους εξισοροποιητές φόρτου, τις κατηγορίες που υπάρχουν, σα πλεονεκτήματα και τον τρόπο λειτουργίας της κάθε μιας.

Το Κεφάλαιο τέσσερα παρουσιάζονται τα εργαλεία και οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της συγκριτικής μελέτης. Δείχνει τις τοπολογίες δικτύων που αναπτύχθηκαν και παρουσιάζει τους αλγόριθμους που συγκρίθηκαν.

Στο Κεφάλαιο πέντε φαίνονται τα αποτελέσματα των μετρήσεων που λήφθηκαν, και στο Κεφάλαιο έξι τα συμπεράσματα της διπλωματικής εργασίας και μερικές μελλοντικές επεκτάσεις που μπορούν να ακολουθήσουν.

Κεφάλαιο 2

Δίκτυα Καθορισμένα από Λογισμικό (SDN)

Με την πάροδο των χρόνων και την ταχύτατη εξέλιξη της τεχνολογίας ο τομέας των δικτύων, οδηγήθηκε στην ανάγκη για επανεξέταση των υπάρχον παραδοσιακών δικτυακών υποδομών και αρχιτεκτονικών. Η τεράστια αύξηση του διαδικτυακού περιεχομένου, των έξυπνων συσκευών, της αλλαγής των μοτίβων δικτυακής κυκλοφορίας, και γενικότερα η ανάπτυξη του Διαδικτύου των Πραγμάτων (Internet of Things - IoT), καθόρισε την ανάγκη για τη δημιουργία μιας νέας αρχιτεκτονικής.

Το κεφάλαιο που ακολουθεί αρχικά παρουσιάζει τα επίπεδα που υπάρχουν στα δίκτυα υπολογιστών και τους περιορισμούς των παραδοσιακών δικτύων. Αμέσως μετά γίνεται μια σύντομη αναφορά στην αρχιτεκτονική τους, και έπειτα μια ανάλυση των SDN δικτύων, συμπεριλαμβάνοντας την παρουσίαση της αρχιτεκτονικής τους. Τέλος γίνεται μια εισαγωγή στα SDN επόμενης γενιάς, στο γιατί αναπτύχθηκαν και στο πώς λειτουργούν.

2.1 Επίπεδα των Δικτύων Υπολογιστών

Στην σημερινή ψηφιακή εποχή τα δίκτυα υπολογιστών έχουν γίνει σημαντικά σχεδόν για όλους τους οργανισμούς. Τα δίκτυα αυτά όμως, για να τεθούν σωστά σε λειτουργία είναι βασισμένα σε μια αρχιτεκτονική η οποία χωρίζεται σε τρία λογικά επίπεδα. Όσο πιο χαμηλό είναι το επίπεδο τόσο πιο κοντά στο φυσικό επίπεδο βρίσκεται, και όσο πιο ψηλό το επίπεδο, τόσο πιο κοντά στο επίπεδο διαχείρισης. Στην Εικόνα 1 φαίνονται τα τρία επίπεδα:

- **Υψηλού Επιπέδου:**
 - Επίπεδο Ελέγχου (Control Plane)
 - Επίπεδο Διαχείρισης (Management Plane)
- **Χαμηλού Επιπέδου:**
 - Επίπεδο Δεδομένων – Προώθησης (Data Plane)

- **Επίπεδο Ελέγχου (Control Plane):**

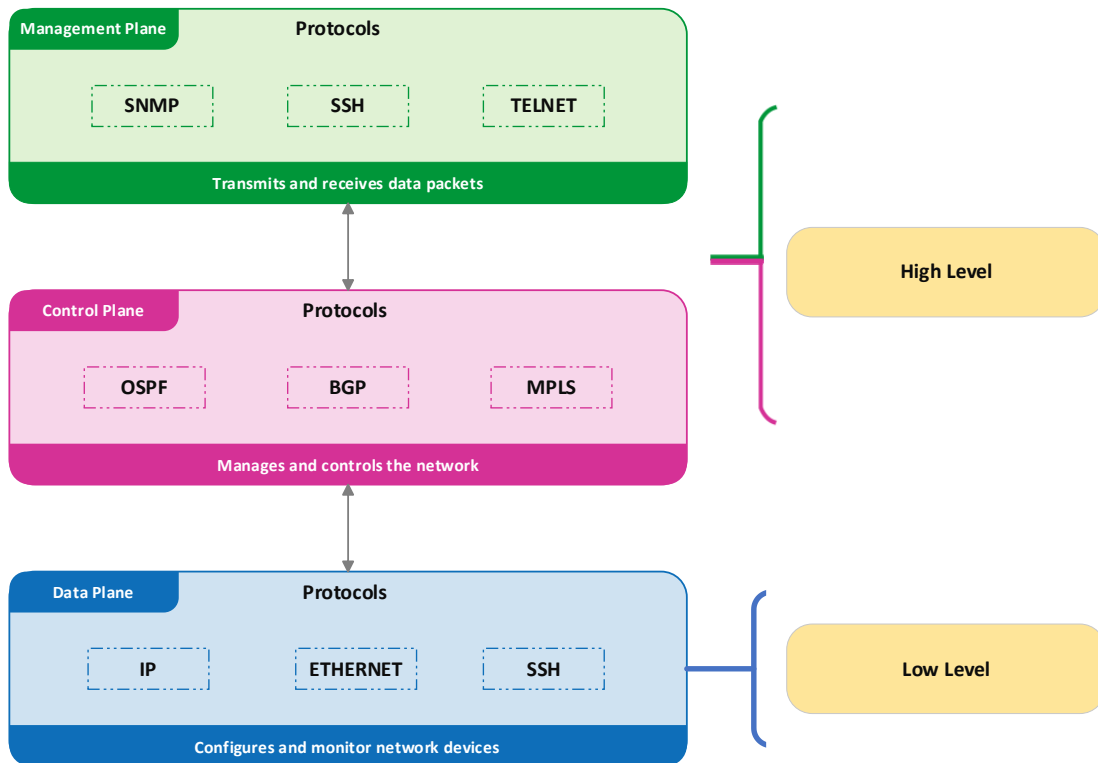
Το επίπεδο ελέγχου έχει την ευθύνη για τη διαχείριση της επικοινωνίας μεταξύ δικτυακών συσκευών, μέσω της ανάθεσης πρωτόκολλων και δρομολόγησης. Η κύρια του λειτουργία είναι να θέτει τους πίνακες και τους κανόνες δρομολόγησης όπου το επίπεδο δεδομένων συμβουλευεται για την προώθηση πακέτων. Κάποια πρωτόκολλα που χρησιμοποιεί αυτό το επίπεδο είναι: Πρωτόκολλο Πληροφοριών Δρομολόγησης (Routing Information Protocol - RIP), Πρωτόκολλο Συνοριακής Πύλης Δικτύου (Border Gateway Protocol - BGP) και Ανοιχτή Πρώτα η Βραχύτερη Διαδρομή (Open Shortest Path First - OSPF)

- **Επίπεδο Διαχείρισης (Management Plane):**

Το επίπεδο διαχείρισης είναι υπεύθυνο για την παρακολούθηση και διαχείριση ολόκληρου του δικτύου. Χρησιμοποιείται κυρίως από τους διαχειριστές δικτύων και σε αυτό το επίπεδο γίνεται η παρακολούθηση απόδοσης και η διαμόρφωση συσκευών. Συνολικά διασφαλίζει ότι ένα δίκτυο είναι ασφαλές και αξιόπιστο. Κάποια πρωτόκολλα που χρησιμοποιεί αυτό το επίπεδο είναι: Πρωτόκολλο Διαμόρφωσης Δικτύου (Network Configuration Protocol - NETCONF), Απλό Πρωτόκολλο Διαχείρισης Δικτύου (Simple Network Management Protocol - SNMP), Πρωτόκολλο Μεταφοράς αρχείων (File Transfer Protocol - FTP)

- **Επίπεδο Δεδομένων (Data Plane):**

Το επίπεδο δεδομένων (Data Plane) ή αλλιώς επίπεδο προώθησης δεδομένων (Forwarding Plane), είναι υπεύθυνο για την προώθηση των πακέτων σε ένα δίκτυο από μια συσκευή σε μια άλλη [δρομολογητές (routers) και μεταγωγείς (switches)], χρησιμοποιώντας επικοινωνία από άκρη σε άκρη (end-to-end). Υπακούει στους πίνακες και κανόνες δρομολόγησης που παρέχει το επίπεδο ελέγχου και είναι κρίσιμο για την συνολική απόδοση του δικτύου.



Εικόνα 1 Αρχιτεκτονική Δικτύων Υπολογιστών

2.2 Περιορισμοί των Παραδοσιακών Δικτύων

Συνήθως τα τέσσερα στάδια, που αποτελούν τον κύκλο ζωής ενός συστήματος δικτύου είναι: η έρευνα ζήτησης, ο στρατηγικός σχεδιασμός, η ανάπτυξη και υλοποίηση, και η λειτουργία και συντήρηση του συστήματος, όπως φαίνεται στην Εικόνα 2. Με βάση τα παραπάνω στάδια που έχουν αναφέρει οι *F. Liu, και λοιποί* [3], έχουν διαμορφωθεί οι αρχιτεκτονικές των παραδοσιακών δικτύων επικοινωνιών που ακόμα εξακολουθούν να βρίσκονται σε ευρεία χρήση στον κόσμο.



Εικόνα 2 Ο Κύκλος Ζωής των Συστημάτων Δικτύου

Ωστόσο οι αρχιτεκτονικές αυτές δεν έχουν σχεδιαστεί με τρόπο ώστε να μπορούν να υποστηρίξουν τις σημερινές απαιτήσεις που έχουν οι χρήστες, οι επιχειρήσεις και οι διάφοροι φορείς.

Παρακάτω παρουσιάζονται ορισμένοι περιορισμοί των παραδοσιακών δικτύων σύμφωνα με την έρευνα του οργανισμού Open Networking Foundation (ONF)[4]:

- **Εξάρτηση από προμηθευτές:** Ως ανταπόκριση στις μεταβαλλόμενες ανάγκες και απαιτήσεις των χρηστών, φορείς και επιχειρήσεις επιδιώκουν να ανανεώνουν τις υπηρεσίες τους και να προσφέρουν νέες δυνατότητες. Ωστόσο η υποστήριξη που προφέρουν οι προμηθευτές των εξοπλισμών δικτύου, κάνουν την προσπάθεια αυτή να γίνεται δυσκολότερη λόγω της μη μεταβλητότητας των συσκευών που προσφέρουν στην αγορά.
- **Πολυπλοκότητα:** Υπάρχει μια τεράστια γκάμα εξοπλισμών δικτύου όπως: διακομιστές (servers), routers, switches ακόμα και εικονικές συσκευές. Για την σωστή επικοινωνία των διαφορετικών συσκευών που μπορεί να υπάρχουν σε ένα δίκτυο, έχουν δημιουργηθεί τα πρωτόκολλα [5]. Ουσιαστικά επιτρέπουν την επικοινωνία μεταξύ τυχών διαφορετικών συσκευών, ανεξάρτητα από διαφορές στη δομή, το σχεδιασμό, ή ακόμα και τις εσωτερικές τους διεργασίες. Τα πρωτόκολλα συχνά σχεδιάζονται με σκοπό το κάθε ένα να επιλύει ένα

συγκεκριμένο ζήτημα. Χωρίς αυτά, τα σημερινά τοπικά δίκτυα (Local Area Networks - LAN) και τα δίκτυα ευρείας περιοχής (Wide Area Networks - WAN) δεν θα μπορούσαν να λειτουργούν.

Ένα παράδειγμα είναι η περίπτωση πρόσθεσης ή αφαίρεσης μιας συσκευής. Η αλλαγή αυτή μπορεί να επηρεάσει διάφορες συσκευές ενός δικτύου όπως switches, routers, τείχη προστασίας (firewalls). Μπορεί επίσης να οδηγήσει στην αλλαγή των οποιονδήποτε εικονικών τοπικών δικτύων (Virtual Local Area Networks - VLAN) που υπάρχουν, της λίστας ελέγχου πρόσβασης (Access Control List - ACL), της ποιότητας υπηρεσιών (Quality of Service - QoS) και άλλων διαφόρων μηχανισμών που βασίζονται σε πρωτόκολλα. Ως αποτέλεσμα η πολυπλοκότητα αυτή πολλές φορές θέτει τα παραδοσιακά δίκτυα σχετικά αμετάβλητα.

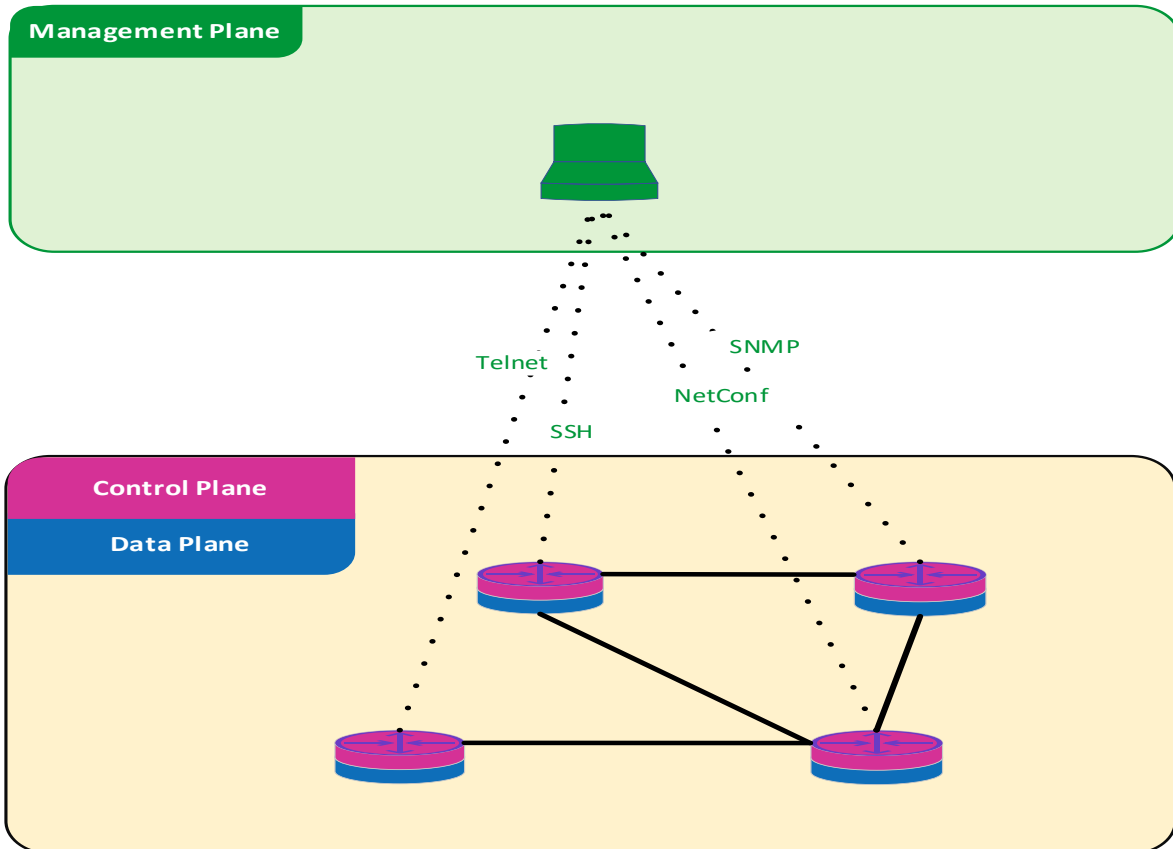
- **Αδυναμία κλιμάκωσης (Scaling):** Στην περίπτωση των κέντρων δεδομένων, μπορεί κάποιος να φανταστεί πόσο μεγάλη είναι η πολυπλοκότητα των δικτύων τους, αφού συνήθως χρειάζεται να διαχειριστούν χιλιάδες, ή και εκατοντάδες χιλιάδες δικτυακές συσκευές. Για την εξυπηρέτηση ενός χρήστη που για παράδειγμα χρειάζεται ένα αποτέλεσμα από τον παγκόσμιο ιστό όσο το δυνατόν συντομότερα, οι πάροχοι υπηρεσιών, λόγω του τεράστιου όγκου δεδομένων που έχουν να διαχειριστούν χρησιμοποιούν μεγάλης κλίμακας αλγόριθμους παράλληλης επεξεργασίας. Οι φορείς αυτοί για να παραμείνουν ανταγωνιστικοί, πρέπει να παρέχουν συνδεσιμότητα υψηλής ποιότητας, και προδιαγραφών σε πιθανόν εκατομμύρια διακομιστές. Ως αποτέλεσμα οποιαδήποτε κλιμάκωση καθιστάτε αδύνατη να γίνει χειροκίνητα. Σε παραδοσιακά δίκτυα τεράστιου μεγέθους βασικές λειτουργίες όπως η καθοδήγηση κίνησης (traffic steering) είναι πολύ πολύπλοκες να υλοποιηθούν.
- **Ασυνέπειες στις πολιτικές:** Σε περίπτωση που χρειάζεται να γίνει εφαρμογή η αλλαγή μιας πολιτικής σε όλο το δίκτυο, ο μεγάλος αριθμός μηχανισμών και συσκευών, καθιστά πολύ δύσκολη την εφαρμογή πολιτικών όπως, πολιτικές πρόσβασης, ασφάλειας και QoS με συνέπεια. Ως αποτέλεσμα μπορεί να

υπάρξουν παραβιάσεις ασφαλείας, μη συμμόρφωση με κανονισμούς και γενικά αρνητικές συνέπειες που αφήνουν κάποιο φερέα ευάλωτο.

2.3 Αρχιτεκτονική Παραδοσιακών Δικτύων

Σε ένα δίκτυο χρησιμοποιούνται συσκευές όπως routers, switches, firewalls και εξισορροπητές φόρτου (load balancers) για συγκεκριμένες εργασίες. Όπως έχει προαναφερθεί, τα παραδοσιακά δίκτυα υπολογιστών παρουσιάζουν κάποιους περιορισμούς. Ανάμεσα σε αυτούς είναι και η αρχιτεκτονική.

Σε κάθε δικτυακή συσκευή που ανήκει σε ένα παραδοσιακό δίκτυο, είναι ενσωματωμένα τα επίπεδα ελέγχου και προώθησης δεδομένων τους όπως φαίνεται στην Εικόνα 3. Για την διαμόρφωση του επιπέδου ελέγχου, οι διαχειριστές ενός δικτύου χρησιμοποιούν την διεπαφή γραμμής εντολών (Command Line Interface), και συγκεκριμένες διεπαφές προμηθευτή (vendor specific interface), για να ορίσουν τους κανόνες με τους οποίους πρέπει να προωθούνται τα πακέτα από το επίπεδο δεδομένων. Το μοντέλο αυτό έχει ως αποτέλεσμα να δημιουργεί την πολυπλοκότητα και τη στατικότητα των δικτύων, αφού καθώς οι διαχειριστές έχουν ήδη καθορίσει τις ροές ελέγχου και τις ροές δεδομένων, οποιαδήποτε τροποποίηση χρειαστεί μπορεί να πραγματοποιηθεί μόνο μέσω επαναδιαμόρφωσης των συσκευών [6]. Συμπεραίνοντας, τα παραδοσιακά δίκτυα όπως αποδεικνύουν οι επιδώσεις τους (υπολογιστικές και οικονομικές), δεν μπορούν να ανταποκριθούν στις ανάγκες της τεχνολογίας, αφού έχει καταστεί απαραίτητο για ένα δίκτυο να πρέπει να προσαρμόζεται γρήγορα και αποτελεσματικά σε αλλαγές.



Εικόνα 3 Αρχιτεκτονική Παραδοσιακών Δικτύων

2.4 Δίκτυα Καθορισμένα από Λογισμικό

Ο τομέας των δικτύων παρουσιάζει μια στροφή προς τον τρόπο με τον οποίο οι υπολογιστές ανταλλάσσουν πληροφορία. Το 2007 [7] προτάθηκε η πρώτη αρχιτεκτονική που είχε ως στόχο, τον κεντρικοποιημένο έλεγχο του δικτύου και τον διαχειρισμό της δρομολόγησης των ροών δεδομένων. Ακολούθως οι δικτυακοί πόροι, οι συσκευές και εφαρμογές άρχισαν να εικονικοποιούνται και να προσφέρουν υπηρεσίες χωρίς ανθρώπινη παρέμβαση, γεγονός που οδήγησε στη λύση των προβλημάτων που παρουσίαζαν τα παραδοσιακά δίκτυα. Η λύση αυτή είναι τα Δίκτυα Καθορισμένα από Λογισμικό (SDN). Παρέχουν μια πλήρη εικόνα στη διαχείριση των συσκευών δικτύου, όπου η ρύθμιση παραμέτρων, η παρακολούθηση και η αντιμετώπιση προβλημάτων μπορούν να γίνουν με ευκολία.

2.4.1 Βασικές Αρχές των SDN

Τα SDN περιβάλλοντα μπορούν να οριστούν σαν σύνολο, ότι αποσυνδέουν το επίπεδο ελέγχου από το επίπεδο προώθησης δεδομένων [8], ενώ στα παραδοσιακά δίκτυα, στις συσκευές δικτύου όπως routers και switches τα δύο αυτά επίπεδα συνδέονται μεταξύ τους

Η αποσύνδεση αυτή επιτρέπει τον κεντρικό έλεγχο του δικτύου μέσω μίας οντότητας που ονομάζεται ελεγκτής (controller). Ο ελεγκτής έχει μια πλήρη εικόνα του δικτύου και μπορεί να λάβει αποφάσεις σχετικά με την ροή δεδομένων.

Οι συσκευές δικτύου που βρίσκονται σε ένα SDN περιβάλλον απλοποιούνται (λόγο της αποσύνδεσης των επιπέδων), έτσι ώστε να μπορούν να επικεντρωθούν αποκλειστικά στην προώθηση πακέτων, με βάση τους κανόνες και τον πίνακα δρομολόγησης που έχει θέσει ο ελεγκτής. Επομένως καθίσταται επιτρεπτός ο ευέλικτος και ο αυτοματοποιημένος προγραμματισμός των δικτυακών συσκευών. Ο προγραμματισμός αυτός μπορεί να γίνει μέσω λογισμικών από τους διαχειριστές.

Ο κεντρικός έλεγχος που διαθέτει ο ελεγκτής, του επιτρέπει να χρησιμοποιεί μια διεπαφή που ονομάζεται Νότια Διεπαφή (Southbound Interface), για τον απευθείας προγραμματισμό των συσκευών δικτύου. Ως αποτέλεσμα υπάρχει αυξημένος έλεγχος του δικτύου και εισάγονται ευκαιρίες για την εικονικοποίηση του.

Η δυνατότητα προγραμματισμού επιτρέπει στους διαχειριστές και τους ερευνητές, να χρησιμοποιούν ανοιχτά πρότυπα και υπηρεσίες δικτύου, χωρίς να βασίζονται σε ιδιόκτητες συσκευές υλικού. Ως εκ τούτου αντιμετωπίζεται το πρόβλημα των παραδοσιακών δικτύων όπου κάθε συσκευή από κάθε προμηθευτή, προσφέρει διαφορετικές δυνατότητες.

Τα SDN αποτελούν μια επαναστατική τεχνολογία για τη διαχείριση δικτύων, αναδιαμορφώνοντας ουσιαστικά τον τρόπο που σχεδιάζονται, ελέγχονται και χρησιμοποιούνται τα δίκτυα. Οι βασικές αρχές των SDN, ανοίγουν το δρόμο για σημαντικά πλεονεκτήματα σε σχέση με τις παραδοσιακές αρχιτεκτονικές δικτύων.

Μερικά από τα πλεονεκτήματα των SDN έναντι των παραδοσιακών δικτύων είναι κάποια από τα παρακάτω, όπως φαίνονται και τον Πίνακα 1:

- **Προγραμματισμός δικτύου:** Στα παραδοσιακά δίκτυα η πρόσθεση και η αφαίρεση μιας συσκευής μπορεί να δημιουργήσει διάφορα προβλήματα σε διάφορα μέρη του δικτύου. Με τον δυναμικό προγραμματισμό μέσω λογισμικού, οι διαχειριστές μπορούν να κάνουν μετατροπές στο δίκτυο και να αναπτύξουν νέες εφαρμογές και υπηρεσίες εύκολα και γρήγορα.
- **Απλοποιημένη διαχείριση δικτύου:** Λόγο του κεντρικού σημείου ελέγχου και διαχείρισης, εξαλείφεται η πολυπλοκότητα.
- **Ευελιξία δικτύου:** Ένα SDN περιβάλλον υποστηρίζει συχνές αλλαγές στις πολιτικές και στις απαιτήσεις του δικτύου, σε πραγματικό χρόνο μέσω λογισμικού. Αυτό παρέχει μεγαλύτερη ευελιξία σε σύγκριση με τα παραδοσιακά δίκτυα, όπου οι αλλαγές απαιτούν χειροκίνητη διαμόρφωση μεμονωμένων συσκευών.
- **Υποστήριξη εικονικοποίησης:** Με τη βοήθεια της εικονικοποίησης δικτύου, ένας διαχειριστής είναι σε θέση να διαμορφώσει ένα δίκτυο ανά πάσα στιγμή και οπουδήποτε σύμφωνα με τις απαιτήσεις. Επίσης επιτρέπει την παρουσία πολλαπλών εικονικών δικτύων σε ένα κοινό φυσικό πόρο, που έχει ως αποτέλεσμα την μείωση κόστους λειτουργίας.
- **Βελτιωμένη διαχείριση:** Τα SDN επιτρέπουν την διαχείριση της ροής δεδομένων, με την εφαρμογή μηχανικής κίνησης (traffic engineering) και εξισορρόπησης φόρτου. Ως αποτέλεσμα οι πόροι του δικτύου χρησιμοποιούνται πιο αποτελεσματικά.

Χαρακτηριστικά	Δίκτυα Καθορισμένα από Λογισμικό	Παραδοσιακά Δίκτυα
Προγραμματισμός	✓	
Κεντρικός έλεγχος	✓	
Επιρρεπής σε σφάλματα		✓
Σύνθετος έλεγχος δικτύου		✓
Ευελιξία δικτύου	✓	
Βελτιωμένη απόδοση	✓	
Εύκολη υλοποίηση	✓	
Αποτελεσματική διαμόρφωση	✓	
Βελτιωμένη διαχείριση	✓	

Πίνακας 1 Δίκτυα Καθορισμένα από Λογισμικό VS Παραδοσιακά Δίκτυα [8]

2.4.2 Αρχιτεκτονική Δικτύων Καθορισμένων από Λογισμικό

Όπως έχει προαναφερθεί, τα Δίκτυα Καθορισμένα από Λογισμικό αποσυνδέουν το επίπεδο ελέγχου από το επίπεδο προώθησης δεδομένων. Η Εικόνα 4 παρουσιάζει την αρχιτεκτονική των SDN. Τα τρία βασικά επίπεδα είναι εξαρτημένα το ένα από το άλλο και η επικοινωνία μεταξύ τους γίνεται μέσω κάποιων διεπαφών [9].

- **Επίπεδο Εφαρμογών (Application Layer)**

Το επίπεδο εφαρμογών αποτελείται από εφαρμογές και υπηρεσίες, που η κύρια τους λειτουργία είναι να επικοινωνούν με το επίπεδο ελέγχου [10], και να ορίζουν την επιθυμητή συμπεριφορά του δικτύου σύμφωνα με τις απαιτήσεις. Στην περίπτωση των παραδοσιακών δικτύων, θα μπορούσε σε αυτό το επίπεδο να χρησιμοποιηθεί μια συσκευή τοίχους προστασίας, ή ενός εξισορροπητή φόρτου. Σε ένα SDN περιβάλλον όμως, θα χρησιμοποιηθεί μια εφαρμογή ή υπηρεσία στην θέση αυτών των συσκευών, για τον έλεγχο και τη διαχείριση του επιπέδου δεδομένων. Η επικοινωνία αυτή γίνεται μέσω ορισμένων διεπαφών που ονομάζονται Βόρειες Διεπαφές (Northbound Interface). Συνήθως η διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface - API) που χρησιμοποιείται είναι το REpresentational State Transfer - REST API.

Το REST δεν είναι ούτε ένα πρωτόκολλο, ούτε ένα πρότυπο. Είναι ένα σύνολο αρχιτεκτονικών περιορισμών που επιτρέπει την επικοινωνία μεταξύ πελάτη και διακομιστή, χρησιμοποιώντας το πρωτόκολλο μεταφοράς υπερ-κειμένου (Hypertext Transfer Protocol – HTTP). Παρέχει τα πρότυπα επικοινωνίας μεταξύ συσκευών και διευκολύνει την επικοινωνία μεταξύ τους [11].

- **Επίπεδο Ελέγχου (Control Layer)**

Το επίπεδο ελέγχου σε ένα SDN περιβάλλον, συνδέει το επίπεδο εφαρμογής με το επίπεδο υποδομής. Ο ελεγκτής βρίσκεται σε αυτό το επίπεδο και υπεύθυνος για την διαχείριση των πολιτικών που μπορεί να θέσει κάποιος διαχειριστής, την υποβολή κανόνων, και την δημιουργία των πινάκων δρομολόγησης. Οι εντολές αυτές με την σειρά τους θα αποσταλούν στο επίπεδο δεδομένων, αφού πρώτα μεταφραστούν σε εντολές που θα είναι κατανοητές από αυτό. Η επικοινωνία των δύο επιπέδων γίνεται μέσω της Νότιας Διεπαφής, και συνήθως χρησιμοποιούνται πρωτόκολλα όπως το OpenFlow [12] και NETCONF μαζί με την τυποποιημένη γλώσσα για μοντέλα δεδομένων YANG [13]. Τέλος, η επικοινωνία μεταξύ κόμβων που βρίσκονται στο επίπεδο ελέγχου (στην περίπτωση μη-κεντροποιημένης αρχιτεκτονικής), γίνεται μέσω μιας διεπαφής που ονομάζεται Ανατολική-Δυτική Διεπαφή (East-West Interface). Κάποια πρωτόκολλα που χρησιμοποιούνται για την επικοινωνία αυτή είναι το Border Gateway Protocol – BGP και Path Computation Element Communication Protocol – PCEP

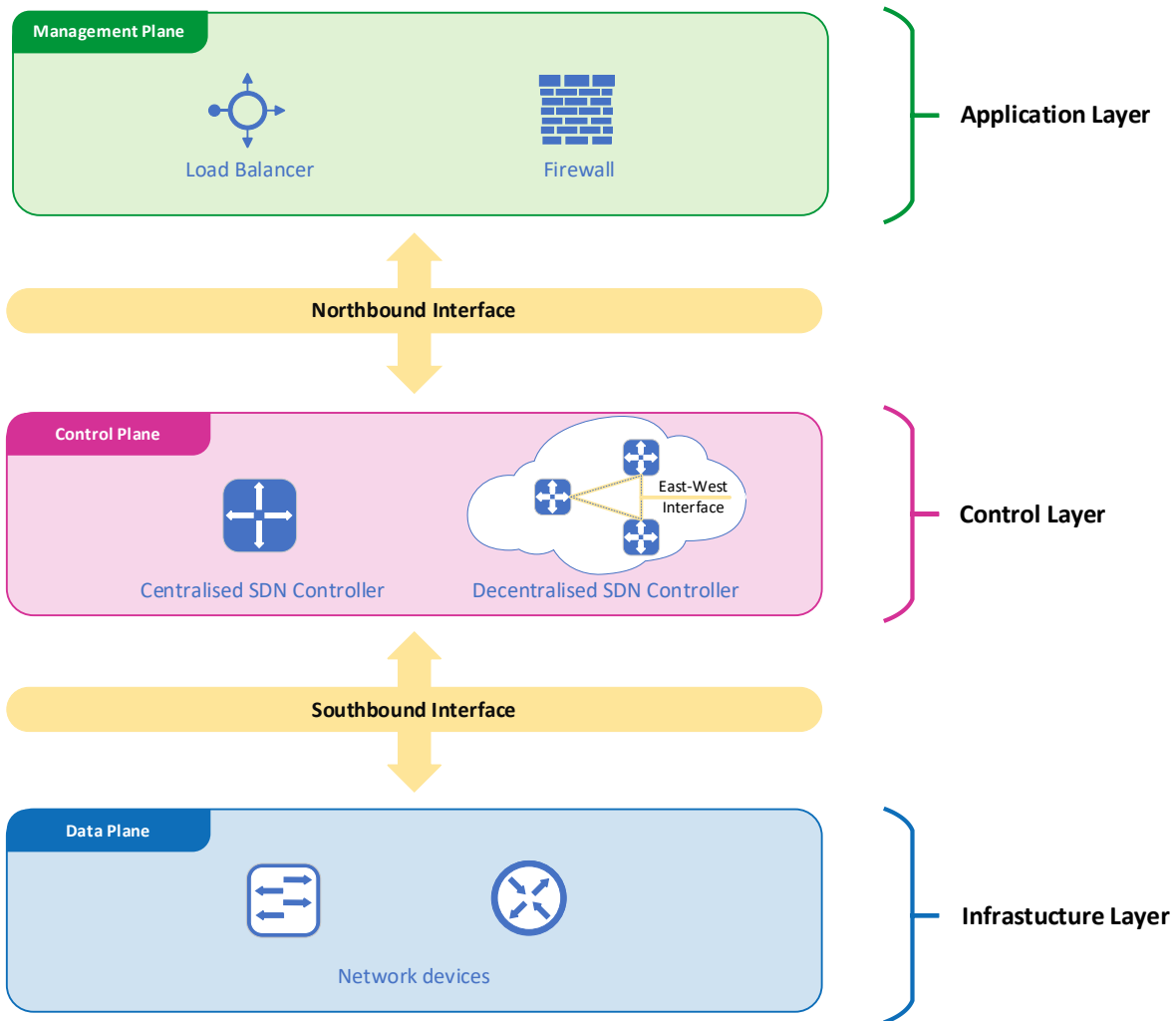
Ο ελεγκτής είναι ο εγκέφαλος ενός SDN [14]. Ανάλογα με την κατηγορία του μπορεί να αποτελείται από ένα μοναδικό κόμβο, «Κεντροποιημένη» αρχιτεκτονική, ή από ένα σύνολο κόμβων «Μη-Κεντροποιημένη» ή «Διανεμημένη» αρχιτεκτονική. Ένας κεντροποιημένος ελεγκτής εφαρμόζει όλες τις λειτουργίες του μόνο σε ένα κόμβο, έτσι επιτυγχάνεται απλότητα αφού η διαχείριση συμβαίνει μόνο από ένα σημείο. Στην περίπτωση ενός διανεμημένου ελεγκτή, το πλεονέκτημα είναι στην περίπτωση της επεκτασιμότητας, και στην υψηλή απόδοση που προσφέρει λόγω της κατανεμημένης του φύσης, όμως αυξάνει την πολυπλοκότητα του δικτύου. Στον Πίνακα 2 φαίνονται μερικοί από τους πιο γνωστούς ελεγκτές SDN.

	Όνομα	Αρχιτεκτονική	API	Γλώσσα προγραμματισμού
	Ryu[15]	Κεντροποιημένη	REST	Python3
	ONOS[16]	Κατανεμημένη	REST	Java
	Hyperflow[17]	Κατανεμημένη	REST	C++
	Beacon[18]	Κεντροποιημένη	Ad-hoc	Java
	FloodLight[19]	Κατανεμημένη	REST	Java

Πίνακας 2 Γνωστοί Ελεγκτές Δικτύων Καθοριζόμενων από Λογισμικό

- **Επίπεδο Υποδομής (Infrastructure Layer)**

Το επίπεδο υποδομής ή αλλιώς Επίπεδο Δεδομένων είναι υπεύθυνο για την προώθηση πακέτων από μία συσκευή σε άλλη. Χρησιμοποιούνται μεταγωγείς ορισμένοι από λογισμικό (SDN switches), και ο σκοπός τους είναι να συμπεριφέρονται όπως ακριβώς έχει ορίσει ο ελεγκτής. Για την επικοινωνία μεταξύ επιπέδου ελέγχου και επιπέδου υποδομής, χρησιμοποιείται το Southbound API και το πρωτόκολλο που χρησιμοποιείται πιο συχνά είναι το OpenFlow Protocol.



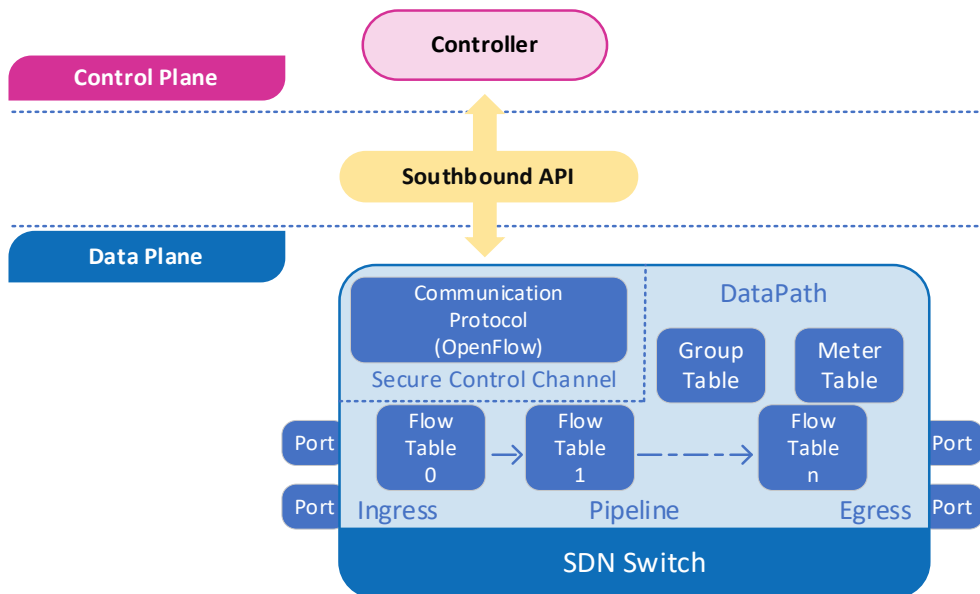
Εικόνα 4 Αρχιτεκτονική Δικτύων Καθοριζόμενων από Λογισμικό

- **Συσκευές Δικτύου**

Η συσκευές δικτύου είναι οντότητες όπως τον ελεγκτή, που η αρμοδιότητα τους είναι να λαμβάνουν πακέτα στις θύρες τους και να εκτελούν μία ή περισσότερες λειτουργίες. Οι λειτουργίες αυτές θα μπορούσαν να είναι: προώθηση πακέτου, απόρριψη πακέτου, αλλαγή κεφαλίδας ή φορτίου. Πρόσθετα παραδείγματα περιλαμβάνουν στοιχεία δικτύου που μπορεί να λειτουργούν σε επίπεδο πάνω από το IP (όπως, δρομολογητές, τείχη προστασίας, εξισορροπητές φόρτου) ή κάτω από το IP (όπως μεταγωγείς επιπέδου 2 και στοιχεία δικτύου οπτικών ινών ή μικροκυμάτων) [10].

- **Μεταγωγέας Καθορισμένος από Λογισμικό (SDN Switch)**

Σε ένα παραδοσιακό μεταγωγέα τα δύο επίπεδα ελέγχου και δεδομένων είναι ενσωματωμένα, οπότε εκτελούνται στην ίδια συσκευή και οι αποφάσεις δρομολόγησης υψηλού επιπέδου και οι λειτουργίες προώθησης. Ως αντίθεση, σε ένα μεταγωγέα SDN τα δύο αυτά επίπεδα είναι χωρισμένα όπως φαίνεται στην Εικόνα 5 [20]. Αποτελείται από δύο μέρη. Το πρώτο μέρος αποτελείται από τον πίνακα ροής που ορίζει τον τρόπο επεξεργασίας και προώθησης πακέτων, και τους πίνακες μετρικών και ομάδων. Το δεύτερο μέρος αποτελείται από μια ασφαλής διεπαφή πρωτοκόλλου επικοινωνίας με τον ελεγκτή, όπως για παράδειγμα το OpenFlow. Η συσκευή απλά εκτελεί λειτουργίες προώθησης, βάσει των κανόνων και οδηγιών που δίνονται από τον ελεγκτή. Οι πίνακες δρομολόγησης συμπληρώνονται επίσης από τον ελεγκτή. Στην περίπτωση που κάποιο πλαίσιο δεν μπορεί να ταυτιστεί με καμία εγγραφή, τότε μέσω του καναλιού ελέγχου αποστέλλεται πίσω στον ελεγκτή, έτσι ώστε να βρεθεί η θύρα εξόδου που πρέπει να ακολουθήσει το πλαίσιο, αν θα σταλεί πίσω στον μεταγωγέα, ή αν θα απορριφθεί. Ένας μεταγωγέας SDN λειτουργεί με ολοκληρωμένα κυκλώματα ειδικού σκοπού (Application Specific Integrated Circuits - ASIC) όπως ακριβώς και ένας παραδοσιακός.



Εικόνα 5 Αρχιτεκτονική Μεταγωγέα Καθορισμένου από Λογισμικό OpenFlow

2.4.3 Πρωτόκολλο OpenFlow

OpenFlow, είναι το πιο διαδεδομένο πρωτόκολλο επικοινωνίας Νότιας Διεπαφής [12]. Επιτρέπει στο επίπεδο ελέγχου να δίνει εντολές στο επίπεδο δεδομένων, έτσι ώστε ο ελεγκτής να μπορεί να προγραμματίσει τους πίνακες προώθησης και δρομολόγησης, και να διαμορφώσει τον τρόπο που χειρίζονται μεταγωγείς και δρομολογητές, τα πακέτα τα δικτύου. Η επικοινωνία γίνεται μέσω του πρωτόκολλου ελέγχου μετάδοσης (Transport Control Protocol - TCP) και κρυπτογραφείται μέσω του πρωτόκολλου Transport Layer Security – TLS. Το πρωτόκολλο OpenFlow ορίζει την δομή και το πλήθος των μηνυμάτων μεταξύ ενός SDN μεταγωγέα και του ελεγκτή [21]. Επίσης λειτουργίες όπως ερωτήματα μεταγωγέων, δομές πινάκων, εγκατάσταση, τροποποίηση, αφαίρεση κανόνων ροής, συλλογή στατιστικών στοιχείων, διαμόρφωση θυρών και ουρών, ορίζονται από το πρωτόκολλο αυτό. Ένας μεταγωγέας που υποστηρίζει το πρωτόκολλο OpenFlow ονομάζεται OpenFlow Switch [22], οι πίνακες (και οι εγγραφές των πινάκων) που εμπεριέχονται σε αυτό, επιτρέπουν στις ροές πακέτων να κατευθύνονται σύμφωνα με τους κανόνες και τις πολιτικές που ορίζονται.

Οι κυριότεροι πίνακες OpenFlow είναι: Πίνακες Ροής (Flow Tables), Πίνακας Ομάδων(Group Table) και Πίνακας Μετρητών (Meter Table) [23].

- **Πίνακες Ροής (Flow Tables):** Οι πίνακες ροής OpenFlow είναι μια ακολουθία από πίνακες που εμπεριέχουν εγγραφές. Η αρίθμηση τους ξεκινάει από το 0 και καθώς ένα πακέτο εισέρχεται σε έναν μεταγωγέα, αρχίζει η σύγκριση των εγγραφών του από τον μικρότερο πίνακα ακολουθιακά, μέχρι να βρεθεί η εγγραφή που να ταιριάζει.
- **Πίνακας Ομάδων (Group Table):** ο πίνακας ομάδων (Group Table) αντιπροσωπεύει σύνολα ενεργειών ομάδας για προώθηση πακέτων, όπως για παράδειγμα η πολυδιανομή (multicast). Στον Πίνακα 3 φαίνεται μια εγγραφή του πίνακα ομάδων και ακολούθως επεξηγούνται τα σημεία της.

Αναγνωριστικό ομάδας	Τύπος ομάδας	Μετρητές	Κουβάδες Ενεργειών
----------------------	--------------	----------	--------------------

Πίνακας 3 Κύρια στοιχεία μιας εγγραφής ομάδας στον πίνακα ομάδας[22]

Αναγνωριστικό ομάδας (Group Identifier): ένας ακέραιος αριθμός χωρίς πρόσημο (Unsigned Integer) 32-bit, που προσδιορίζει μοναδικά την ομάδα στο OpenFlow switch.

Τύπος ομάδας (Group Type): Ο τύπος ομάδας που υποστηρίζει η εγγραφή.

Μετρητές (Counters): Μετράει τον αριθμό πακέτων που έχουν επεξεργαστεί από τις ομάδες.

Κουβάδες ενεργειών (Action Buckets): Μια λίστα που περιέχει "κουβάδες" ενεργειών, δηλαδή σύνολα από ενέργειες και τις σχετικές τους παραμέτρους. Οι ενέργειες σε έναν κουβά εφαρμόζονται πάντα ως σύνολο ενεργειών.

- **Πίνακας Μετρητών (Meter Table):** Ένας πίνακας μετρητών αποτελείται από καταχωρήσεις μετρητών, ανά ροή. Οι μετρητές ανά ροή επιτρέπουν την εφαρμογή πολιτικών όπως ο περιορισμός ρυθμού, μια λειτουργία QoS που

περιορίζει ένα σύνολο ροών σε ένα επιλεγμένο εύρος ζώνης. Σε περίπτωση ταύτισης της εγγραφής του πλαισίου δεδομένων με μια εγγραφή του πίνακα μετρητών, το πλαίσιο θα ακολουθήσει την πολιτική που ορίζεται από την εγγραφή.

Στην Εικόνα 6 παρουσιάζονται τα κύρια στοιχεία μιας εγγραφής OpenFlow και στον Πίνακα 4, ο πίνακας με τα κριτήρια αντιστοίχισης.

Κριτήρια αντιστοίχισης	Προτεραιότητα	Μετρητές	Ενέργειες	Χρόνοι Ακύρωσης	Cookie
------------------------	---------------	----------	-----------	-----------------	--------

Εικόνα 6 Κύρια στοιχεία μιας εγγραφής OpenFlow[22]

Switch Port	VLAN ID	VLAN pcp	MAC Src	MAC Dst	Eth Type	IP Src	IP Dst	IP ToS	IP Prot	L4 Sport	L4 Dport
-------------	---------	----------	---------	---------	----------	--------	--------	--------	---------	----------	----------

Πίνακας 4 Τα κριτήρια αντιστοίχισης σε μια εγγραφή Openflow

Κριτήρια αντιστοίχισης (Match Fields): Κριτήρια αντιστοίχισης είναι τα πεδία που συγκρίνονται με τα πεδία των πακέτων (Πίνακας 4).

Προτεραιότητα (Priority): Δηλώνει την προτεραιότητα της εγγραφής. Οι τιμές που δέχεται το πεδίο αυτό είναι από 0-65535, και η σειρά αντιστοίχισης αρχίζει από τους κανόνες με τη μεγαλύτερη προτεραιότητα.

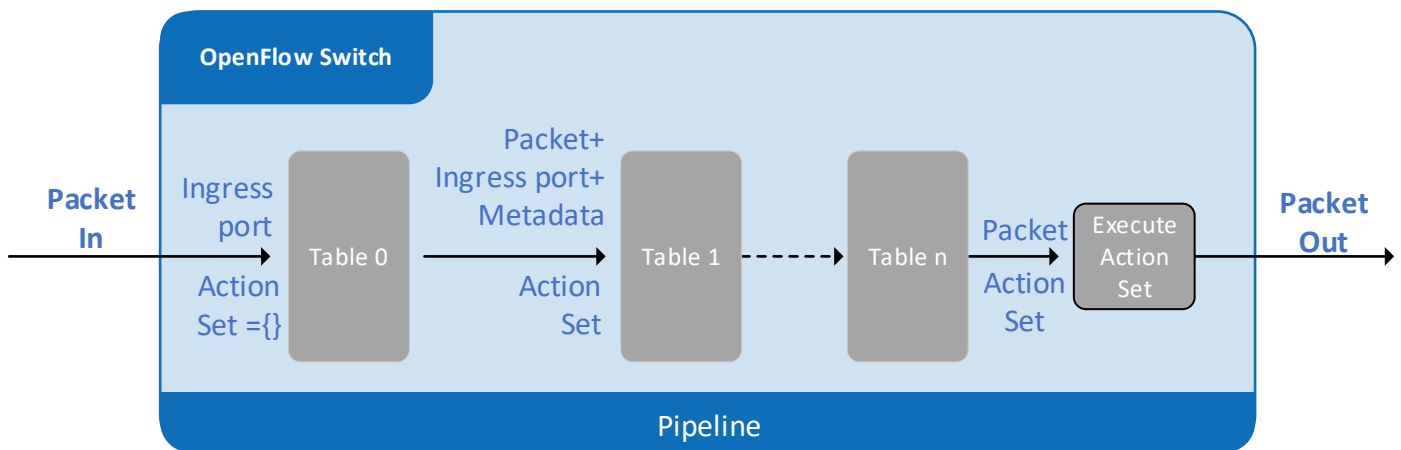
Μετρητές (Counters): Οι αντιστοίχιση μιας εγγραφής από ένα πλαίσιο, αυξάνει το πλήθος των μετρητών. Για παράδειγμα ένας μετρητής που να μετράει το πλήθος των πακέτων, ή των Bytes που έχουν επεξεργαστεί από τον κανόνα.

Ενέργειες (Actions): Κάθε εγγραφή ροής περιέχει ένα σύνολο εντολών που εκτελούνται όταν ένα πακέτο ταιριάζει με την καταχώρηση. Αυτές οι ενέργειες οδηγούν σε αλλαγές στο πακέτο, στο σύνολο πεδίων ή και στην επεξεργασία του αγωγού (pipeline).

Χρόνοι Ακύρωσης (Timeouts): Το μέγιστο χρονικό διάστημα ή ο χρόνος αδράνειας πριν από τη λήξη της εγγραφής από το μεταγωγέα. Όταν ξεπεραστεί ο χρόνος αυτός η εγγραφή αφαιρείται.

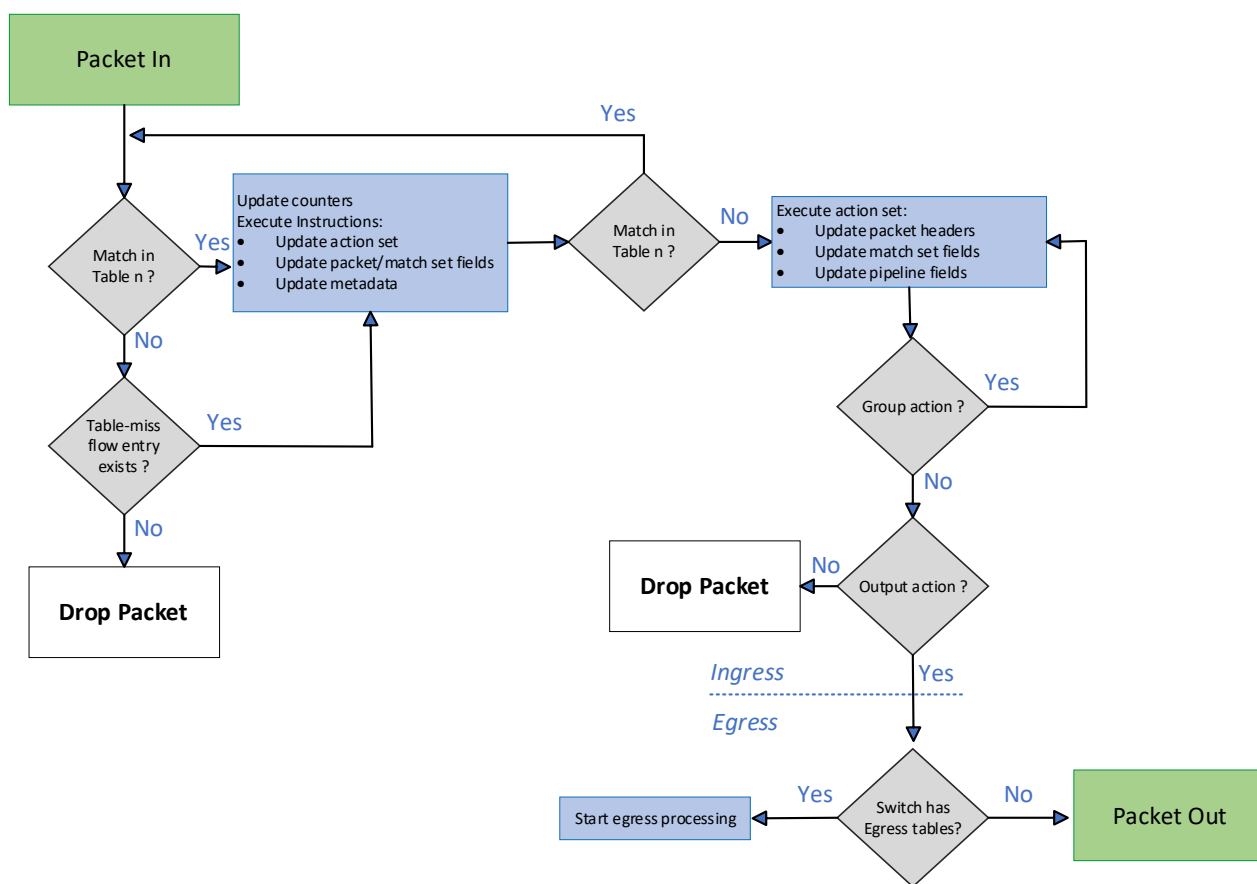
Cookie: Τιμή δεδομένων που επιλέγεται αυτόματα από τον ελεγκτή. Μπορεί να χρησιμοποιηθεί από τον για το φιλτράρισμα, την αναγνώριση και την ταυτοποίηση των εγγραφών. Δεν χρησιμοποιείται κατά την επεξεργασία πακέτων, και εγγραφές που έχουν την ίδια λειτουργία και υπάρχουν σε πολλούς πίνακες, έχουν την ίδια τιμή.

Η επεξεργασία των πακέτων σε ένα SDN Switch γίνεται μέσω του αγωγού (Pipeline). Στην Εικόνα 7 παρουσιάζεται ένας OpenFlow αγωγός. Η ροή επεξεργασίας του OpenFlow Pipeline μπορεί να κινηθεί μόνο προς τα εμπρός (από αριστερά προς τα δεξιά). Οι πίνακες ροής σε ένα μεταγωγέα OpenFlow αριθμούνται διαδοχικά ξεκινώντας από το 0. Όταν ένα πακέτο φτάσει σε ένα μεταγωγέα, στο Pipeline ξεκινά η σύγκριση πακέτων από τον πίνακα 0. Το πακέτο αντιστοιχίζεται πρώτα στα δεδομένα ροής στον πίνακα 0. Στη συνέχεια, το πακέτο προωθείται στον επόμενο πίνακα ροής με βάση το αποτέλεσμα της αντιστοίχισης. Η λειτουργία αυτή συνεχίζεται μέχρι να παρθεί μια απόφαση (για παράδειγμα ενέργεια απόρριψης πακέτου).[22]



Εικόνα 7 Η αλληλουχία πινάκων OpenFlow σε ένα μεταγωγέα SDN

Ο αγωγός OpenFlow (OpenFlow pipeline) ορίζει πώς θα δράσει κάθε καταχώρηση ροής σε ένα πακέτο. Στην περίπτωση που ένα πακέτο δεν ταιριάζει με καμία εγγραφή ροής, προκύπτει αστοχία πίνακα. Μια εγγραφή αντιστοίχισης πακέτων ορίζει τον τρόπο επεξεργασίας αυτών των πακέτων. Η διαδικασία αντιστοίχισης φαίνεται στην Εικόνα 8.



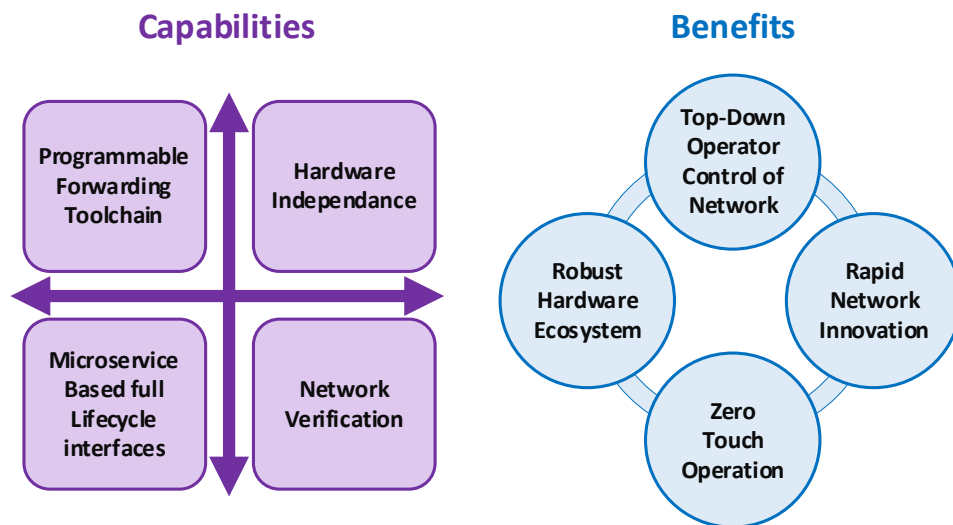
Εικόνα 8 Η Διαδικασία αντιστοίχισης πακέτων στο OpenFlow [22]

2.5 Επόμενη Γενιά Δικτύων Καθορισμένων από Λογισμικό

Όπως κάθε αναδυόμενη τεχνολογία, τα SDN αντιμετωπίζουν μερικές προκλήσεις. Σύμφωνα με τους A. Dawod και N. I. Abdullah κάποιες από αυτές είναι [24]:

- **Ποιότητα υπηρεσιών (QoS):** Η παροχή ποιότητας υπηρεσιών αποτελεί πρόκληση, καθώς στα δίκτυα SDN δεν υπάρχει τυπικός τρόπος καθορισμού πολιτικών και απαιτήσεων ελέγχου κυκλοφορίας υψηλού επιπέδου. Η ποιότητα υπηρεσιών πρέπει να λαμβάνει υπόψη παραμέτρους όπως για παράδειγμα το εύρος ζώνης, την καθυστέρηση, το jitter, την απώλεια. Για την υποστήριξη διαφόρων εφαρμογών, διαφορετικών μεταξύ τους, η πρόκληση αυτή πρέπει να αντιμετωπιστεί.
- **Ασφάλεια:** Το κεντροποιημένο επίπεδο ελέγχου, εισάγει κινδύνους ασφαλείας, όπως επιθέσεις DDoS προς τον ελεγκτή και μεμονωμένα σημεία αποτυχίας. Ωστόσο η προστασία του ελεγκτή και η διασφάλιση της κατάλληλης ταυτοποίησης μεταξύ ελεγκτών, μεταγωγέων και εφαρμογών είναι σημαντική. Είναι επίσης σημαντικό οι πολιτικές ασφαλείας να επιβάλλονται ομοιόμορφα στους διαφορετικούς τομείς, για την διασφάλιση της ασφαλείας.
- **Επεκτασιμότητα:** Οι ελεγκτές ως κεντρικό σημείο ελέγχου μπορεί να γίνει εμπόδιο λόγω περιορισμών στους υπολογιστικούς πόρους που διαθέτει. Καθώς το μέγεθος του δικτύου αυξάνεται με όλο και περισσότερες ροές, ο όγκος αιτημάτων από μεταγωγείς μπορεί να τον υπερφορτώσει.
- **Τυποποίηση:** Υπάρχει έλλειψη κοινής αρχιτεκτονικής και λύσεων για την ομοιόμορφη αντιμετώπιση ζητημάτων όπως η QoS, η εξισορρόπηση φόρτου, η ασφάλεια και η επεκτασιμότητα στο SDN. Υπάρχουν διαφορετικές προτάσεις αλλά όχι τυποποιημένη προσέγγιση.

Απάντηση στις προκλήσεις που αναφέρθηκαν πιο πάνω είναι η Επόμενη Γενιά Δικτύων Καθορισμένων από Λογισμικό (Next Generation Software Defined Networks - NG-SDN) όπως ανακοίνωσε ο οργανισμός ONF [25]. Τα SDN επόμενης γενιάς είναι μια εξέλιξη των παραδοσιακών SDN, που αντιμετωπίζουν τους περιορισμούς και ευνοούν τις αναδυόμενες τεχνολογίες. Ο κύριος στόχος είναι η προσφορά καινούριων δυνατοτήτων και οφελών όπως φαίνεται στην Εικόνα 9 είναι:



Εικόνα 9 Δυνατότητες και Οφέλη SDN επόμενης γενιάς[25]

Δυνατότητες

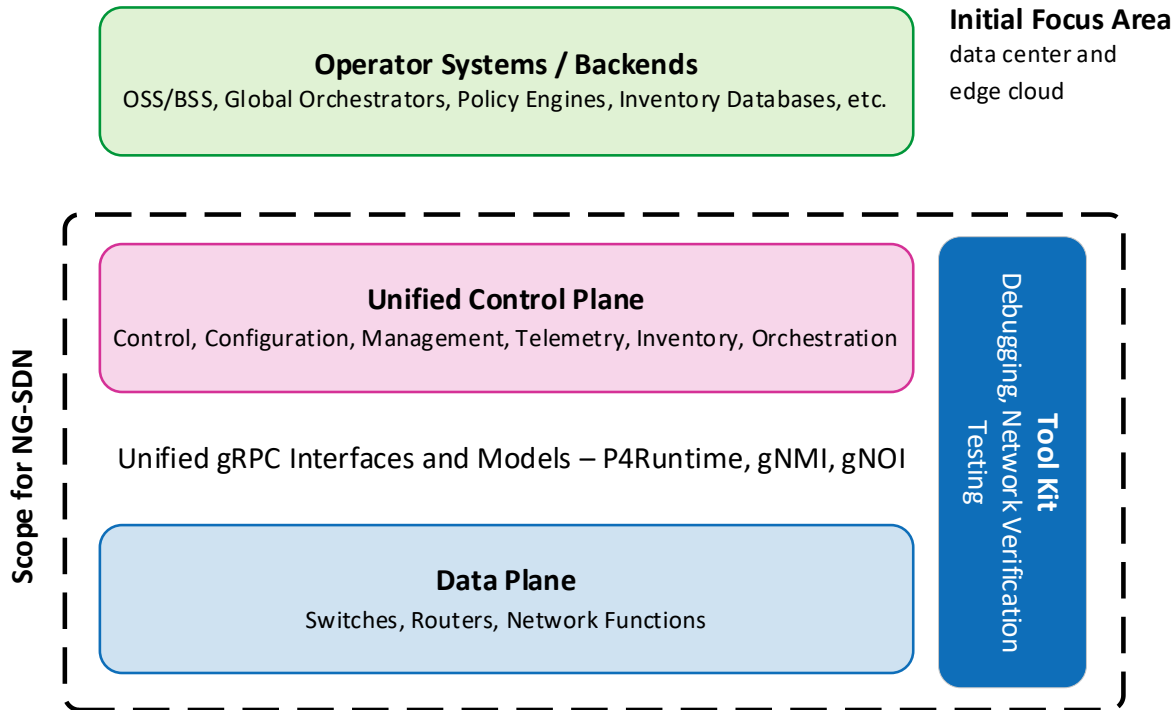
- **Προγραμματιζόμενη αλυσίδα εργαλείων προώθησης:** Πλήρως προγραμματιζόμενες διεπαφές μπορούν να ορίσουν την επιθυμητή συμπεριφορά στο δίκτυο
- **Ανεξαρτησία Υλικού:** Ο εξοπλισμός είναι πλήρως προγραμματίσιμος και χωρίς εξάρτηση από προμηθευτές.
- **Διεπαφές που βασίζονται σε μικροϋπηρεσίες:** Cloud-native μικροϋπηρεσίες δημιουργούν τις διεπαφές.
- **Πλήρης έλεγχος δικτύου:** Λεπτομερής και σε πραγματικό χρόνο μέτρηση κάθε πακέτου και ροής που διασχίζει το δίκτυο.

Οφέλη

- **Έλεγχος Δικτύου από διαχειριστή από πάνω προς τα κάτω:** Τα δίκτυα που έχουν υλικό που λειτουργεί με κυκλώματα ASIC προσεγγίζονται από την μέθοδο από κάτω προς τα πάνω. Λόγο του προγραμματισμού του δικτύου των κυκλωμάτων ASIC οι διαχειριστές και προγραμματιστές μπορούν να χρησιμοποιήσουν την μέθοδο από πάνω προς τα κάτω.
- **Ταχεία καινοτομία δικτύου:** Το δίκτυο μπορεί να επαναπρογραμματιστεί χωρίς να περιορίζει την λειτουργία του.
- **Λειτουργία Zero-Touch:** Αυτοματοποιώντας τον κύκλο ζωής μιας λειτουργίας, ελαχιστοποιούνται τα ανθρώπινα λάθη.
- **Ισχυρό οικοσύστημα υλικού:** Η απεξάρτηση από τους προμηθευτές ευνοεί την γρήγορη ανάπτυξη υλικού και παρέχει στους νέους προμηθευτές ευκολότερη πρόσβαση στην αγορά.

2.5.1 Αρχιτεκτονική NG-SDN

Η αρχιτεκτονική των επόμενης γενιάς SDN, βασίζεται στις καινούργιες δυνατότητες και στα οφέλη που δημιουργούνται, μέσω της αντιμετώπισης των περιορισμών που έχουν τα παραδοσιακά SDN [26]. Το πεδίο εφαρμογής της αρχιτεκτονικής των SDN επόμενης γενιάς, όπως φαίνεται στην Εικόνα 10. Αποτελείται από το σύνολο του επιπέδου ελέγχου και του επιπέδου προώθησης δεδομένων. Αξιοποιεί τα στοιχεία αφαίρεσης και ανοιχτού κώδικα, εξαλείφοντας έτσι τον περιορισμό που δημιουργούταν από το ότι τα επίπεδα αυτά γράφονταν και δοκιμάζονταν σε συγκεκριμένες μόνο σκευές, με αποτέλεσμα να μην υποστηρίζονται από ένα αρκετά σημαντικό ποσοστό εξοπλισμού [25]. Η εισαγωγή της αλυσίδας εργαλείων (Tool-Kit) για αποσφαλμάτωση, δοκιμή και επαλήθευση, επιταχύνει την ανάπτυξη εφαρμογών και διεπαφών τόσο για την λειτουργία των επιπέδων, όσο και για την επικοινωνία τους.



Εικόνα 10 Αρχιτεκτονική NG-SDN

Το επίπεδο ελέγχου με βάση τις δυνατότητες και τα οφέλη που προσφέρει το NG-SDN έχει εξελιχθεί με τέτοιο τρόπο έτσι ώστε να επιτρέπει [27]:

Ανεξαρτησία υλικού: Το επίπεδο ελέγχου δεν αποτελείται μόνο από συγκεκριμένες πλατφόρμες υλικού, επιτρέποντας έτσι την διαχείριση και τον έλεγχο δικτύων που αποτελούνται από διαφορετικές και εξελισσόμενες αρχιτεκτονικές υλικού άσπρου κουτιού. Έτσι προάγεται ο αγνωστικισμός του υλικού και η εύκολη ενσωμάτωση νέων συσκευών.

Αρχιτεκτονική μικροϋπηρεσιών: Αναλύοντας σύνθετες λειτουργίες επιπέδου ελέγχου σε μικρότερες, αρθρωτές και αυτόνομες υπηρεσίες, διευκολύνεται η δοκιμή και η ανάπτυξη με αποτέλεσμα να ενισχύεται η επεκτασιμότητα.

Παροχή μηδενικής αφής (Zero-Touch): Αυτοματοποιώντας τη διαδικασία διαμόρφωσης και ανάπτυξης συσκευών δικτύου, μειώνονται τα λειτουργικά έξοδα και απλοποιείται η διαχείριση του δικτύου.

Επαλήθευση δικτύου: Ενσωματώνοντας δυνατότητες επαλήθευσης δικτύου, επιτρέπεται η παρακολούθηση, η επικύρωση και η διαμόρφωση της συμπεριφοράς του δικτύου σε πραγματικό χρόνο.

Ενσωμάτωση ανοιχτών προτύπων: Με τη συμμόρφωση του επιπέδου ελέγχου με ανοιχτά πρότυπα όπως για παράδειγμα τη, γλώσσα προγραμματισμού P4 [28], την δομή (Framework) gRPC (google Remote Procedure Calls) [29] με τις διεπαφές gNMI (gRPC Network Management Interface) [30] και gNOI (gRPC Network Operations Interface) [31], προωθείται η διαλειτουργικότητα και επιτρέπεται η ανάπτυξη μεταξύ πλατφόρμων. Έτσι ευνοείται η συνεργασία και η καινοτομία στο οικοσύστημα των NG-SDN.

Το επίπεδο ελέγχου, για τη διαμόρφωση, τη διαχείριση, την τηλεμετρία και την οργάνωση του επιπέδου δεδομένων, χρησιμοποιεί κάποιο πρωτόκολλα και δομές επικοινωνίας ανοιχτού πρότυπου. Το gRPC είναι ένα γενικό πλαίσιο κλήσης απομακρυσμένης διαδικασίας (Remote Procedure Call - RPC) ανοιχτού κώδικα υψηλής απόδοσης, που μπορεί να εκτελεστεί σε οποιοδήποτε περιβάλλον. Χρησιμοποιεί το πρωτόκολλο HTTP/2 ως επίπεδο μεταφοράς, και τη γλώσσα περιγραφής διεπαφής Protocol Buffer για τη σειροποίηση δεδομένων [32]. Η εξέλιξη των διεπαφών καθιστάτε εύκολη, αφού υποστηρίζεται η συμβατότητα προς τα πίσω [29]. Το gNMI είναι ένα πρωτόκολλο που βασίζεται στη δομή gRPC για τη διαχείριση, τη διαμόρφωση και την τηλεμετρία συσκευών δικτύου. Έχει σχεδιαστεί με τη δυνατότητα να υποστηρίζει ένα ευρύ φάσμα μοντέλων δεδομένων, συμπεριλαμβανομένου του OpenConfig [29][33]. Το gNOI ορίζει ένα σύνολο μικροϋπηρεσιών που βασίζονται στο gRPC, για την εκτέλεση λειτουργικών εντολών σε συσκευές δικτύου. Παρέχει μια τυποποιημένη διεπαφή για λειτουργικές εντολές δικτύου που βρίσκονται συνήθως στη διεπαφή γραμμής εντολών (CLI) [30].

2.5.2 Γλώσσα προγραμματισμού P4

Υπάρχουν διάφορες γλώσσες και περιβάλλοντα προγραμματισμού για την ανάπτυξη εφαρμογών, και τον προγραμματισμό του επιπέδου δεδομένων σε ένα NG-SDN περιβάλλον. Μερικές λύσεις προγραμματισμού επιπέδου δικτύων είναι:

- **Lucid** [34]: Είναι μια γλώσσα ειδικού σκοπού (Domain Specific Language - DSL) υψηλού επιπέδου και ροής δεδομένων για την υλοποίηση εφαρμογών ελέγχου σε επίπεδα δεδομένων. Μπορεί να χρησιμοποιηθεί σε συσκευές PISA (Protocol Independent Switch Architecture)
- **Frenetic** [35]: Είναι μία DSL υψηλού επιπέδου, δηλωτική γλώσσα για περιγραφή της επεξεργασίας των πακέτων από τις συσκευές δικτύου. Βασίζεται στην έννοια της κατάστασης (Stateful) επεξεργασίας πακέτων, η οποία επιτρέπει στους προγραμματιστές να ορίζουν πολύπλοκους αγωγούς προώθησης, που μπορούν να προσαρμοστούν στην μεταβαλλόμενη συμπεριφορά της κυκλοφορίας του δικτύου.
- **P4**: Είναι μια DSL γλώσσα υψηλού επιπέδου για τον προγραμματισμό επεξεργαστών πακέτων ανεξάρτητων από πρωτόκολλο. Χρησιμοποιείται για τον καθορισμό της συμπεριφοράς προώθησης για προγραμματιζόμενα τσιπ μεταγωγών καθώς και για κυκλώματα ASIC. Ένα πρόγραμμα P4 υλοποιείται από το επίπεδο δεδομένων και μπορεί να προγραμματιστεί από το επίπεδο ελέγχου.
- **Data Plane Development Kit (DPDK)** [36]: Είναι ένα πλαίσιο που χρησιμοποιείται για την ανάπτυξη εφαρμογών επιπέδου δεδομένων. Μπορεί να εφαρμοστεί σε μια μεγάλη ποικιλία στόχων όπως μονάδες επεξεργασίας δικτύου (Network Processing Units - NPU), κεντρικές μονάδες Επεξεργασίας (Central Processing Units - CPU) και κάρτες διεπαφής δικτύου (Network Interface Card - NIC). Οι εφαρμογές που αναπτύσσονται σε DPDK δεσμεύουν όλους τους πόρους εκ των προτέρων, όπως ένα πρόγραμμα στην C, και μπορούν να υλοποιήσουν στοίβες πρωτόκολλων δικτύου χωρίς να χρειάζεται εξειδικευμένος εξοπλισμός συσκευών [37].

Η πιο διαδεδομένη γλώσσα προγραμματισμού επιπέδου δεδομένων είναι η P4, και προτάθηκε για πρώτη φορά το 2014 [38]. Είναι μια DSL υψηλού επιπέδου για τον προγραμματισμό επεξεργαστών πακέτων ανεξάρτητων από πρωτόκολλο (Programming Protocol-independent Packet Processors), από όπου πήρε και το όνομά της. Δημιουργήθηκε για να δουλεύει παράλληλα με το πρωτόκολλο επικοινωνίας OpenFlow και να αντιμετωπίσει τους περιορισμούς του [35]. Χαρακτηρίζεται ως μια γλώσσα προγραμματισμού για τον έλεγχο των επιπέδων προώθησης πακέτων σε συσκευές δικτύων, όπως δρομολογητές και μεταγωγείς. Η P4 στοχεύει στην παροχή δυνατοτήτων επαναδιαμόρφωσης, ανεξαρτήτως στόχου και πρωτόκολλου:

Δυνατότητα επαναδιαμόρφωσης (Reconfigurability): Οι μεταγωγείς θα μπορούν να επαναπροσδιορίσουν τον τρόπο με τον οποίο γίνεται η ανάλυση και η επεξεργασία πακέτων τους, ακόμα και μετά την ανάπτυξή τους.

Ανεξαρτησία πρωτοκόλλου (Protocol Independence): Το πρόγραμμα P4 ορίζει τη μορφή των πακέτων και των πινάκων αντιστοίχισης ενεργειών (match-action tables) για την επεξεργασία κεφαλίδων, με αποτέλεσμα να μπορούν να περιγραφούν τα πρωτόκολλα από τον πρόγραμμα.

Ανεξαρτησία στόχου (Target Independence): Το πρόγραμμα P4 περιγράφει την επεξεργασία πακέτων ανεξάρτητα από το υλικό. Ένας μεταγλωττιστής αντιστοιχίζει το πρόγραμμα P4 σε έναν μεταγωγέα, ο οποίος μπορεί πολλών διαφορετικών τύπων μηχανών εκτέλεσης, όπως CPU γενικής χρήσης, Field Programmable Gate Arrays - FPGAs, NPU και ASIC.

Σε αντίθεση με μια γλώσσα γενικού σκοπού όπως η C ή η Python, η P4 είναι μια γλώσσα βελτιστοποιημένη για προώθηση δεδομένων δικτύου. Μπορεί να περιγράψει στοιχεία όπως κεφαλίδες πακέτων, αναλυτές (parsers), match-action tables, και προγράμματα ελέγχου για τον καθορισμό του τρόπου επεξεργασίας των πακέτων. Ο στόχος είναι να μπορεί να μεταγλωττιστεί σε διαφορετικούς τύπους μεταγωγών, από μεταγωγείς λογισμικού έως μεταγωγείς υλικού, επιτρέποντας έτσι σε έναν μόνο ελεγκτή να μπορεί να προγραμματίσει πολλούς τύπους μεταγωγών. Οι βασικές αφαιρέσεις που παρέχονται από τη γλώσσα P4 είναι [39]:

Τύπος Κεφαλίδας (Header Type): Ο τύπος κεφαλίδας περιγράφει τη μορφή κάθε κεφαλίδας μέσα σε ένα πακέτο (το σύνολο των πεδίων και τα μεγέθη τους).

Αναλυτές (Parsers): Οι αναλυτές ανοίγουν τα πακέτα και περιγράφουν τις επιτρεπόμενες ακολουθίες κεφαλίδων, τον τρόπο αναγνώρισης αυτών των ακολουθιών και τα πεδία που πρέπει να εξαχθούν από τα αυτά.

Deparser: Ο Deparser είναι το αντίθετο του Parser. Δηλαδή ανακατασκευάζουν το πακέτο που έχει ανοίξει ο Parser.

Πίνακες (Tables): Οι πίνακες P4 γενικεύουν τους παραδοσιακούς πίνακες μεταγωγών. Μπορούν να χρησιμοποιηθούν για την υλοποίηση πινάκων δρομολόγησης και αναζήτησης ροής, λιστών ελέγχου πρόσβασης και άλλων τύπων πινάκων που μπορούν να οριστούν από το χρήστη, συμπεριλαμβανομένων σύνθετων αποφάσεων με πολλαπλές μεταβλητές.

Ενέργειες (Actions): Οι ενέργειες είναι τμήματα κώδικα που περιγράφουν τον τρόπο χειρισμού των πεδίων κεφαλίδας και των μεταδεδομένων των πακέτων. Οι ενέργειες μπορεί να περιλαμβάνουν δεδομένα, τα οποία παρέχονται από το επίπεδο ελέγχου κατά το χρόνο εκτέλεσης.

Μονάδες Αντιστοίχισης Ενεργειών (Match-action units): Οι μονάδες αντιστοίχισης ενεργειών εκτελούν την ακόλουθη σειρά λειτουργιών. Για παράδειγμα:

1. Κατασκευή κλειδιών αναζήτησης από πεδία των πακέτων ή από τα υπολογισμένα μεταδεδομένα.
2. Αναζήτηση στον πίνακα χρησιμοποιώντας το κατασκευασμένο κλειδί, και επιλογή της κατάλληλης ενέργειας για εκτέλεση.
3. Εκτέλεση της επιλεγμένης ενέργειας.

Ροή Ελέγχου(Control Flow): Καθορίζει τη ροή ελέγχου μεταξύ πινάκων για την επεξεργασία πακέτων

Εξωτερικά Αντικείμενα (Extern Objects): Αντικείμενα που αφορούν την αρχιτεκτονική του υλικού που μπορούν να χρησιμοποιηθούν από προγράμματα P4 μέσω API, αλλά δεν μπορούν να τροποποιηθούν.

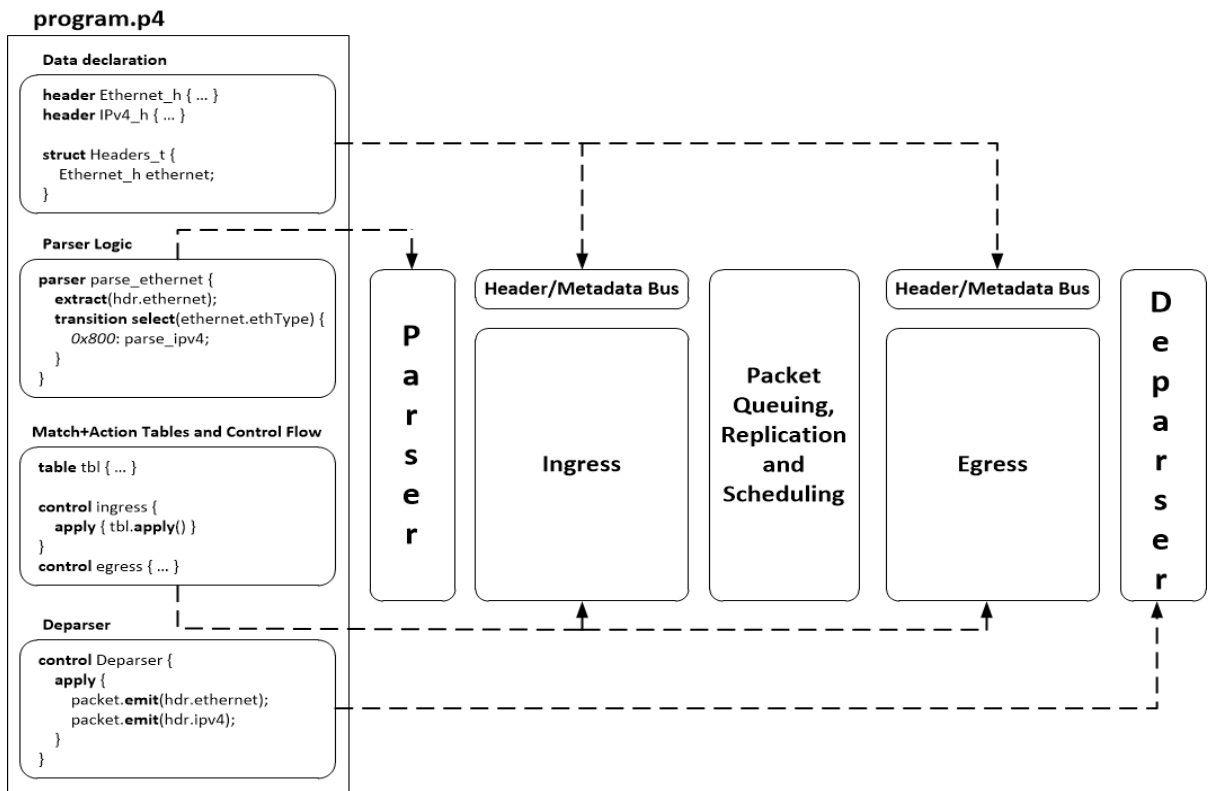
Μεταδεδομένα (Metadata): Τα μεταδεδομένα μπορεί να είναι καθορισμένα από τον χρήστη: δομές δεδομένων που σχετίζονται με κάθε πακέτο, ή εγγενή μεταδεδομένα που παρέχονται από την αρχιτεκτονική που σχετίζεται με κάθε πακέτο. Για παράδειγμα ένα εγγενή μεταδεδομένο είναι η θύρα εισόδου όπου έχει ληφθεί ένα πακέτο.

Αρχιτεκτονικό Μοντέλο (Architecture Model): Παρέχει μια διεπαφή για τον προγραμματισμό ενός στόχου μέσω κάποιου συνόλου προγραμματιζόμενων στοιχείων P4. Για παράδειγμα μερικά αρχιτεκτονικά μοντέλα είναι [40]:

- **Protocol-Independent Switch Architecture (PISA):** Πρόκειται για μια αρχιτεκτονική προώθησης αγωγού που προσδιόρισε την ανάγκη προγραμματισμού του επίπεδου δεδομένων, και αρχικά στοχοποιήθηκε από την έκδοση της P4_14. Έπειτα ξεπεράστηκε με την έκδοση της P4_16 όπου υπήρξε η ανάγκη στόχευσης πολλαπλών προγραμματιζόμενων συσκευών με διαφορετικές αρχιτεκτονικές.
- **Portable Switch Architecture (PSA):** Πρόκειται για μια προδιαγραφή αρχιτεκτονικής μεταγωγέα που μπορεί να εφαρμοστεί σε οποιονδήποτε στόχο. Στοχεύει στη συνθεσιμότητα και τη φορητότητα.
- **Αρχιτεκτονικές που βασίζονται σε λογισμικό:** Αυτές περιλαμβάνουν αρχιτεκτονικές για μεταγωγείς λογισμικού όπως Open vSwitch, eBPF, DPDK.
- **Αρχιτεκτονικές που βασίζονται σε υλικό:** Αρχιτεκτονικές για συσκευές υλικού όπως για παράδειγμα Intel, Broadcom, Cisco, Juniper, Barefoot Tofino.

Στην Εικόνα 11 [41] παρουσιάζεται η δομή ενός προγράμματος P4. Αποτελείται από τρεις κύριες ενότητες: Ορισμός πρωτοκόλλων (δήλωση δεδομένων), Λογική ανάλυσης (Parser & Deparser) και ένα μπλοκ ελέγχου που περιέχει πίνακες Match-Action. Η πρώτη ενότητα ορίζει τις κεφαλίδες των πρωτόκολλων που θα μπορεί να αναγνωρίσει η συσκευή δικτύου, για την ανάλυση των εισερχόμενων δεδομένων, και την αναγνώριση του τύπου των πακέτων. Η λογική Parser είναι μια μηχανή πεπερασμένης κατάστασης που ορίζει τα βήματα για την ανάγνωση και την ανάλυση των εισερχόμενων πακέτων. Επιπλέον, στο πρόγραμμα P4 ορίζεται ένας αριθμός

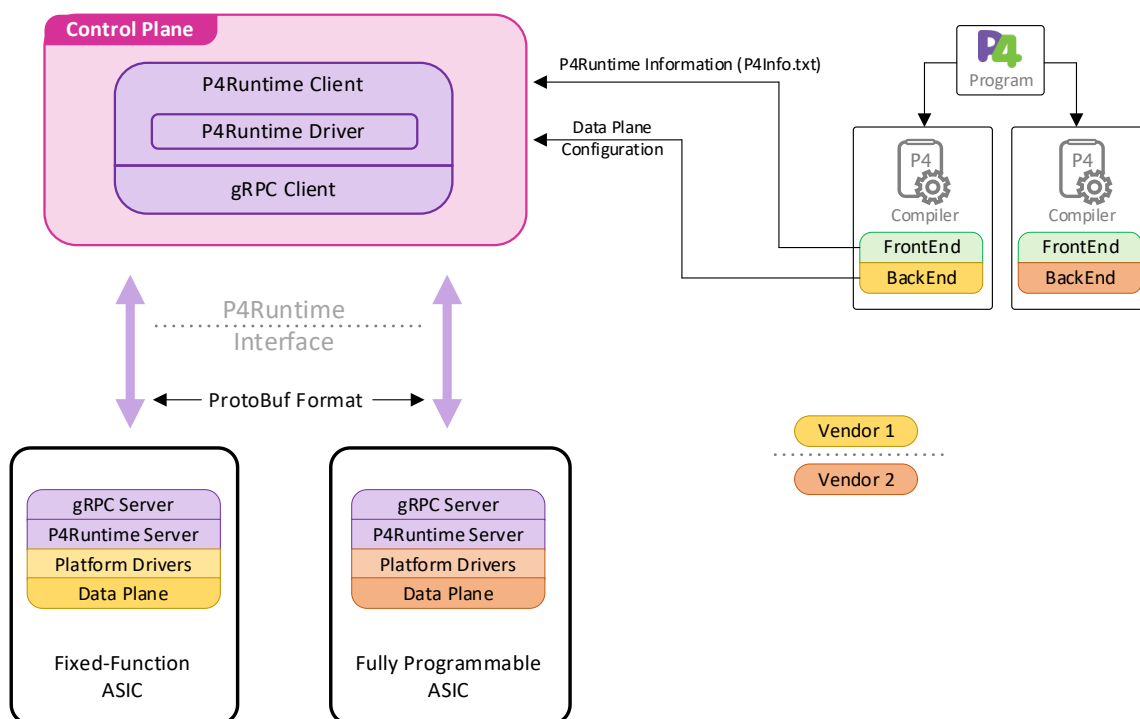
μπλοκ ελέγχου, τα οποία περιέχουν τους πίνακες Match-Action. Στη συνέχεια, σε πακέτα που ταιριάζουν με τον κανόνα, μπορούν να εκτελεστούν τρεις ενέργειες: προώθηση, αποβολή και καμία ενέργεια. Το τελευταίο μέρος είναι το Deparser, το οποίο ορίζει και ανακατασκευάζει τη σειρά των κεφαλίδων του πακέτου για τα εξερχόμενα πακέτα.



Εικόνα 11 Δομή προγράμματος P4 [41]

Για τον έλεγχο των στοιχείων επιπέδου δεδομένων μιας συσκευής που ορίζεται ή περιγράφεται από ένα πρόγραμμα P4, χρησιμοποιείται η διεπαφή χρόνου εκτέλεσης P4Runtime [42]. Είναι ένα API που βασίζεται σε RPC που επιτρέπει στους ελεγκτές λογισμικού να προγραμματίζουν και να διαχειρίζονται τη συμπεριφορά προώθησης δεδομένων σε συσκευές P4. Χρησιμοποιεί το gRPC ως το υποκείμενο πρωτόκολλο επικοινωνίας και τη δομή δεδομένων Protocol Buffers. Για τον απομακρυσμένο προγραμματισμό και τη διαχείριση της συμπεριφοράς μιας συσκευής P4 από τους ελεγκτές, ένας διακομιστής P4Runtime εκτελείται στη συσκευή στόχου. Ως αποτέλεσμα της εξ' αποστάσεως επικοινωνίας, επιτρέπεται σε

πολλούς ελεγκτές να συνδέονται ταυτόχρονα σε μία συσκευή. Για τη διασφάλιση ότι μόνο ένας ελεγκτής έχει πρόσβαση εγγραφής σε κάθε πόρο τη φορά, χρησιμοποιεί έναν μηχανισμό διαιτησίας βασισμένο σε ρόλους. Επίσης παρέχει μηχανισμούς εισόδου εξόδου (I/O) που επιτρέπουν στους ελεγκτές να ανταλλάζουν πακέτα με το επίπεδο δεδομένων. Τέλος υποστηρίζει τη ρύθμιση των παραμέτρων σε έναν ελεγκτή (με το αρχείο p4info.txt που δημιουργείτε), και τον έλεγχο και τη διαχείριση διαφόρων αντικειμένων του προγράμματος P4 όπως πίνακες, καταχωρητές, μετρητές κτλ. [42]. Στην Εικόνα 12 παρουσιάζεται ο προγραμματισμός μίας προγραμματίσιμης και μίας ειδικού σκοπού συσκευής, από δύο διαφορετικούς προμηθευτές με τη χρήση της γλώσσας P4 και της διεπαφής P4Runtime.



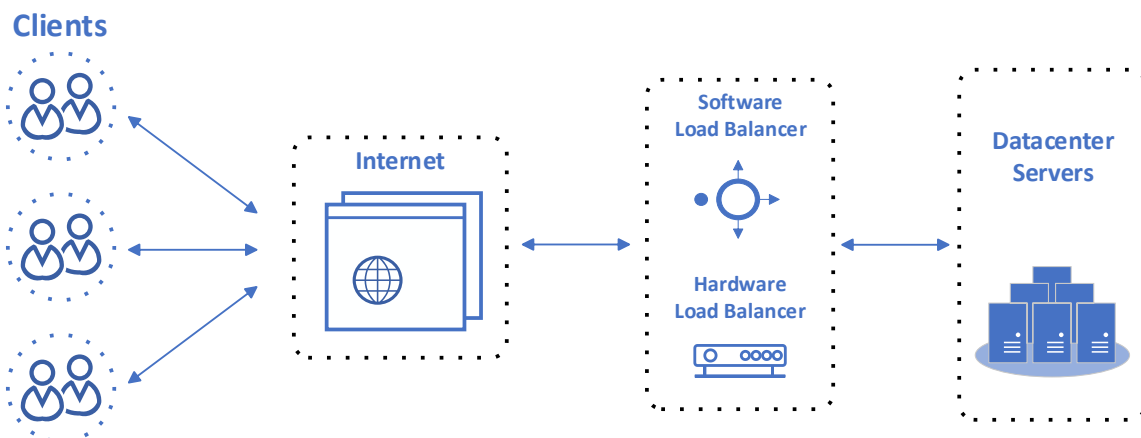
Εικόνα 12 Προγραμματισμός Συσκευής με P4 και P4Runtime

Κεφάλαιο 3

Εξισορροπητές Φόρτου

Οι εξισορροπητές φόρτου στα δίκτυα συνήθως είναι μια φυσική συσκευή, ή μια εικονική παρουσία που λειτουργεί είτε σε εξειδικευμένο υλικό, είτε σε λογισμικό. Η λειτουργία και ο σκοπός ενός εξισορροπητή φόρτου είναι να διανέμει την κίνηση του δικτύου σε πολλούς διακομιστές έτσι ώστε να αποφεύγεται η υπερφόρτωση του συστήματος.

Ένα παράδειγμα είναι οι μοντέρνες ιστοσελίδες υψηλής κίνησης, που μπορεί να έχουν από εκατοντάδες χιλιάδες, μέχρι και εκατομμύρια ταυτόχρονες συνδέσεις και αιτήματα χρηστών. Τα αιτήματα αυτά μπορεί να είναι: επιστροφή απλών δεδομένων εφαρμογής, εικόνες, κείμενο, η ακόμα και βίντεο. Καταλαβαίνει κάποιος ότι για την γρήγορη και αξιόπιστη εξυπηρέτηση των αιτημάτων αυτών, και για την κάλυψη του όγκου δεδομένων που ανταλλάσσεται, το υπολογιστικό σύστημα του πάροχου θα αποτελείται από πάρα πολλούς διακομιστές. Στην παρούσα περίπτωση όπως φαίνεται στην Εικόνα 13, ένας εξισορροπητής φόρτου θα βρίσκεται ενδιάμεσα από τους διακομιστές και τους χρήστες. Θα διαχειρίζεται την κίνηση που λαμβάνει από τους χρήστες, και θα την αποστέλλει στους διακομιστές με τέτοιο τρόπο, έτσι ώστε να μεγιστοποιεί την ταχύτητα απάντησης και να διασφαλίζει ότι κανένας διακομιστής δεν θα είναι υπερφορτωμένος [43].



Εικόνα 13 Ένα δίκτυο με εξισορροπητές φόρτου.

Το κεφάλαιο αυτό παρουσιάζει την εξέλιξη των εξισορροπητών φόρτου, τις κατηγορίες τους, τι προσφέρει η κάθε μία, καθώς και τους τύπους που υπάρχουν.

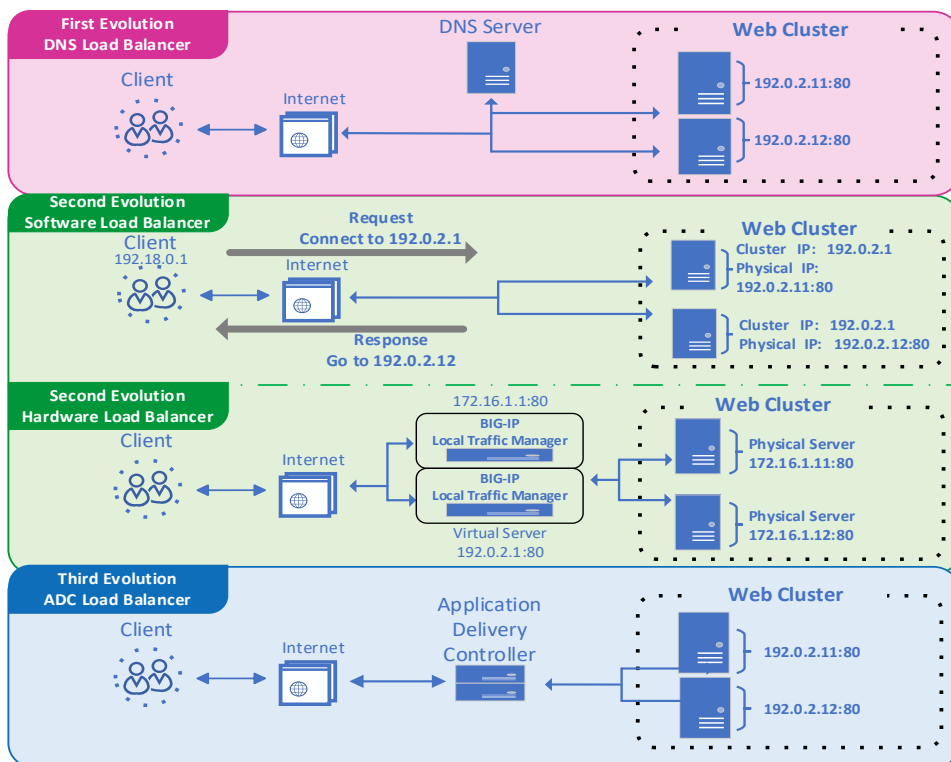
3.1 Η Εξέλιξη των Εξισορροπητών Φόρτου

Η εξισορρόπηση φόρτου γεννήθηκε από μία ανάγκη. Την ανάγκη εξυπηρέτησης όλο και περισσότερου όγκου δεδομένων. Όταν το διαδίκτυο σιγά σιγά άρχισε να μεγαλώνει, οι τότε μεγαλο-εταιρίες ανακάλυψαν ότι τα Mainframes που είχαν, δεν διέθεταν λογισμικό διακομιστή ιστού (Web Server). Η πιο οικονομική λύση, ήταν να χρησιμοποιήσουν το τυπικό και διαθέσιμο υλικό διακομιστή, που πουλούσαν οι κατασκευαστές υπολογιστών. Το πρόβλημα ήταν όμως ότι οι διακομιστές δεν μπορούσαν να επεξεργαστούν τον όγκο δεδομένων, με αποτέλεσμα οι επιχειρήσεις να τρέπονται εκτός λειτουργίας. Έτσι εμφανίστηκε η πρώτη εξέλιξη των εξισορροπητών φόρτου. Δημιουργήθηκε με την τότε τρέχουσα τεχνολογία και ήταν το σύστημα ονομάτων περιοχών (Domain Name Server - DNS), με τον αλγόριθμο Round Robin [44]. Μετέφραζε ονόματα (www.domain.com) σε διευθύνσεις IP που αναγνωρίζονταν από τη μηχανή. Το κυριότερο πρόβλημα του Round Robin DNS ήταν ότι δεν υπήρχε επιμονή (Persistence). Επιμονή είναι η έννοια της διασφάλισης ότι ένας χρήστης, δεν θα μεταφερθεί σε άλλο διακομιστή από την στιγμή που έχει γίνει η αρχική σύνδεση, ή όταν συνεχίσει μία συνεδρία που είχε ανασταλεί.

Η λύση οδήγησε στην δεύτερη εξέλιξη των εξισορροπητών φόρτου, με την εισαγωγή στον προγραμματισμό λογισμικού [31]. Εφαρμόστηκαν τεχνικές απευθείας στο λογισμικό εφαρμογών ή στο λειτουργικό σύστημα των διακομιστών. Με την ομαδοποίηση των διακομιστών σε συστάδες, προστέθηκε και μία δεύτερη διεύθυνση IP (εκτός από την φυσική τους IP) που την ονόμασαν IP συστάδας (Cluster IP). Όταν ένας χρήστης ήθελε να συνδεθεί σε μία υπηρεσία, συνδεόταν με την Cluster IP και τον αναλάμβανε όποιος διακομιστής απαντούσε πρώτος. Μερικά σύγχρονα λειτουργικά συστήματα διακομιστών είναι το Windows Server 2016 και Red Hat's High Availability Linux Server [45]. Ωστόσο, με την πάροδο του χρόνου

αμφισβητήθηκε η επεκτασιμότητα της εξισορρόπησης φόρτου βάσει εφαρμογών. Έτσι στα μέσα του 1990 [46] δημιουργήθηκαν οι πρώτοι εξισορροπητές φόρτου ως φυσικές συσκευές δικτύου. Χρησιμοποιούσαν τεχνικές όπως η μετάφραση διεύθυνσης δικτύου (Network Address Translation – NAT), και παρουσίαζαν διευθύνσεις εικονικών διακομιστών στον έξω κόσμο. Όταν κάποιος χρήστης προσπαθούσε να συνδεθεί, θα προωθούσε τη σύνδεση στον πιο κατάλληλο πραγματικό διακομιστή με βάση το έλεγχο υγείας που πραγματοποιούσε.

Οι πρώτες φυσικές συσκευές εξισορρόπησης φόρτου οδήγησαν στην τρίτη εξέλιξη, η οποία είναι οι ελεγκτές παράδοσης εφαρμογών (Application Delivery Controllers - ADC) [47]. Τα ADC βρίσκονται σε ευρεία χρήση σήμερα σε κέντρα δεδομένων. Λαμβάνουν τα βασικά χαρακτηριστικά των εξισορροπητών φόρτου όπως επεκτασιμότητα, υψηλή διαθεσιμότητα και προβλεψιμότητα, και επεκτείνονται σε αυτά προσφέροντας δυνατότητες όπως ασφάλεια, προσωρινή αποθήκευση (caching), συμπίεση και διαμόρφωση ρυθμού. Στην Εικόνα 14 παρουσιάζονται οι τρεις εξελίξεις των εξισορροπητών φόρτου.



Εικόνα 14 Η Εξέλιξη των Εξισορροπητών Φόρτου

3.2 Κατηγορίες Εξισορροπητών Φόρτου

Οι εξισορροπητές φόρτου που χρησιμοποιούνται σήμερα χωρίζονται σε δύο κατηγορίες:

Εξισορροπητές Φόρτου βασισμένοι σε υλικό:

- Μπορεί να περιέχουν ενσωματωμένες δυνατότητες εικονικοποίησης. Σε μία συσκευή μπορεί να υπάρχουν πάνω από μία περιπτώσεις εικονικού εξισορροπητή φόρτου.
- Είναι συνήθως συσκευές υψηλής απόδοσης, ικανές να επεξεργάζονται με ασφάλεια μεγάλο όγκο κίνησης από διάφορους τύπους εφαρμογών.
- Προσφέρουν ευέλικτες αρχιτεκτονικές πολλαπλών περιβάλλοντων και την πλήρη απομόνωση τους.

Εξισορροπητές Φόρτου βασισμένοι σε λογισμικό:

- Προσφέρουν εξοικονόμηση χώρου και μείωση κόστους
- Μπορούν να αντικαταστήσουν πλήρως τις συσκευές εξισορρόπησης φόρτου με την ανάλογη λειτουργικότητα και μεγαλύτερη ευελιξία.
- Μπορούν να λειτουργούν σε HyperVisors [48] και σε κοντέινερ (Docker Containers) [49] χωρίς να επιβαρύνουν σημαντικά το λειτουργικό σύστημα.

Η εξισορρόπηση φόρτου μπορεί να γίνει σε διάφορα επίπεδα του μοντέλου διασύνδεσης ανοικτών συστημάτων (Open Systems Interconnection - OSI). Συνήθως ο διαχωρισμός γίνεται ενδιάμεσα στο Επίπεδο 4 (Layer 4 – L4) και Επίπεδο 7 (Layer 7 – L7) [50].

Η εξισορρόπηση φόρτου L4 λειτουργεί στο επίπεδο μεταφοράς (Transport Layer) του OSI, το οποίο είναι υπεύθυνο για την παράδοση μηνυμάτων, χωρίς να λαμβάνει υπόψη το τι εμπεριέχεται σε αυτά. Δύο πρωτόκολλα που ανήκουν σε στο L4 είναι το TCP και το HTTP. Η διαδικασία εξισορρόπησης φόρτου γίνεται με την προώθηση των πακέτων δικτύου προς και από το διακομιστή, εξετάζοντας μόνο τα αρχικά

πακέτα στην κεφαλίδα της ροής TCP, δηλαδή τη διεύθυνση IP και τις θύρες προέλευσης και προορισμού. Δεν γίνεται επιθεώρηση στο περιεχόμενο τους, και μπορούν να λάβουν μόνο περιορισμένες αποφάσεις δρομολόγησης.

Η εξισορρόπηση φόρτου L7 λειτουργεί στο επίπεδο εφαρμογής (Application Layer) του OSI, το οποίο είναι υπεύθυνο για το πραγματικό περιεχόμενο του μηνύματος. Το κυρίαρχο πρωτόκολλο σε αυτό το επίπεδο είναι το HTTP. Η διαδικασία εξισορρόπησης φόρτου είναι πιο περίπλοκη από το L4. Όταν φτάσει ένα μήνυμα, αποσυνδέεται από την κυκλοφορία του δικτύου, και εξετάζεται το περιεχόμενό του. Η απόφαση δρομολόγησης γίνεται ανάλογα με το περιεχόμενό του μηνύματος (για παράδειγμα, τη διεύθυνση URL, ή το cookie). Μόλις ληφθεί μια απόφαση, δημιουργείται μια νέα σύνδεση TCP (ή επαναχρησιμοποιείται η ίδια) στον καθορισμένο διακομιστή, και αποστέλλεται το μήνυμα.

Η εξισορρόπηση φόρτου L7 είναι πιο απαιτητική σε πόρους από την εξισορρόπηση φόρτου L4 που βασίζεται σε πακέτα, αλλά σε έναν σύγχρονο διακομιστή σπάνια προκαλεί υποβάθμιση της απόδοσης. Η εξισορρόπηση φόρτου L7 επιτρέπει στον εξισορροπητή να λαμβάνει πιο έξυπνες αποφάσεις, και μπορεί να εφαρμόσει βελτιστοποιήσεις και στο περιεχόμενο, όπως συμπίεση και κρυπτογράφηση.

3.3 Νέα Γενιά Εξισοροποιητών Φόρτου

Όπως έχει προαναφερθεί στο Κεφάλαιο 3.1, οι εξισοροποιητές φόρτου έχουν εξελιχθεί σε τρία στάδια. Η τέταρτη εξέλιξη όπως καταλαβαίνει κάποιος, αφορά τα SDN περιβάλλοντα.

Υπάρχουν διάφοροι αλγόριθμοι εξισορρόπησης φόρτου. Μερικοί από τους πιο γνωστούς είναι, ο Round Robin, Weighted Round Robin και Equal-Cost Multi-Path – ECMP. Τα SDN περιβάλλοντα λόγω της ευελιξίας και επεκτασιμότητας που παρέχουν, επιτρέπουν την εγκατάσταση εξισοροποιητών φόρτου που είναι βασισμένοι σε λογισμικό, σε μεταγωγείς άσπρου κουτιού. Έτσι αξιοποιούνται τα πλεονεκτήματα που προσφέρει ο προγραμματισμός εξοπλισμού, έτσι ώστε να

ταιριάζει στις απαιτήσεις κάποιου δικτύου και επίσης να επιτυγχάνεται μείωση κόστους σχετικά με την ανάπτυξη και διαχείριση. Τέλος η εξισορρόπηση φόρτου είναι ένα από τα σημαντικότερα θέματα σχετικά με την επίλυση του προβλήματος της υπερφόρτωσης των δικτύων, έτσι είναι και μία από τις πρώτες ελκυστικές και σημαντικές εφαρμογές στα δίκτυα SDN [45].

Κεφάλαιο 4

Μεθοδολογία & Υλοποίηση

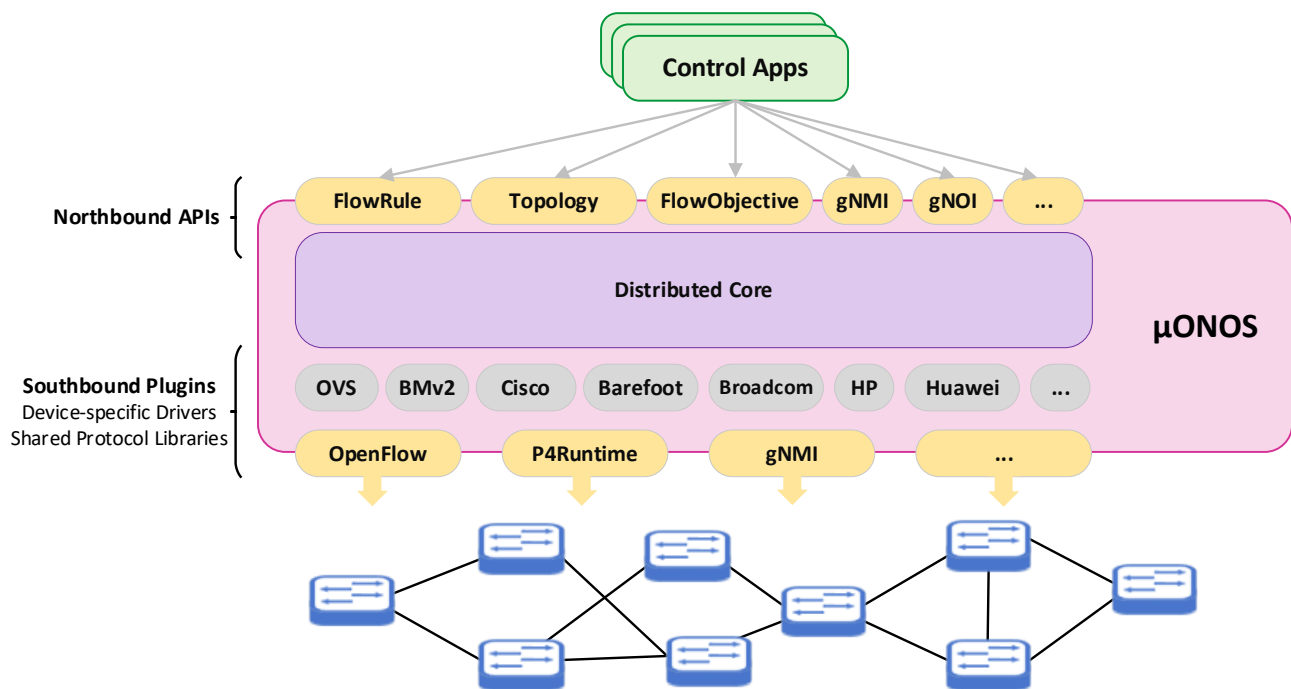
Στο κεφάλαιο που ακολουθεί, παρουσιάζεται η μεθοδολογία και η υλοποίηση, της συγκριτικής μελέτης αλγόριθμων εξισορρόπησης κίνησης, σε δίκτυα καθορισμένα από λογισμικό. Αρχικά θα εξερευνηθούν τα εργαλεία που χρησιμοποιήθηκαν, και οι δυνατότητες τους. Έπειτα θα παρουσιαστούν οι αλγόριθμοι που χρησιμοποιήθηκαν για τη μελέτη και τέλος η συνολική υλοποίηση του συστήματος.

4.1 ONOS

Σε προηγούμενο κεφάλαιο έχει αναλυθεί η λειτουργία της οντότητας του ελεγκτή, που βρίσκεται στο πεδίο ελέγχου. Στον Πίνακα 2 αναφέρθηκαν μερικοί από τους ελεγκτές που χρησιμοποιούνται για τον έλεγχο του επιπέδου δεδομένων σε ένα περιβάλλον SDN. Ο ελεγκτής που επιλέχθηκε για την υλοποίηση της μελέτης είναι το Open Network Operating System – ONOS [16]. Είναι ο πιο ευρέως χρησιμοποιημένος ελεγκτής ανοιχτού κώδικα για δίκτυα SDN. Το ONOS σχεδιάστηκε για να ανταποκρίνεται στις ανάγκες των διαχειριστών δικτύου, που επιθυμούν να δημιουργήσουν λύσεις υψηλής ποιότητας, και να αξιοποιούν τις συσκευές λευκού κουτιού. Παράλληλα προσφέρουν την ευελιξία για τη δημιουργία και την ανάπτυξη νέων δυναμικών υπηρεσιών δικτύου, με απλοποιημένες διεπαφές προγραμματισμού. Ως αποτέλεσμα προσφέρει υψηλή διαθεσιμότητα, επίδοση και επεκτασιμότητα.

Τα SDN εξελίχθηκαν σε NG-SDN, έτσι και η επόμενη γενιά της πλατφόρμας ONOS είναι το κωδικό όνομα μONOS [51]. Βασίζεται στη νέα γενιά διεπαφών, μηχανισμών, και προτύπων ελέγχου και διαμόρφωσης όπως είναι τα P4/P4Runtime, gRPC/gNMI/gNOI και OpenConfig. Επίσης παρέχει βάση για υποστήριξη λειτουργιών μηδενικής αφής, και εφαρμόζεται σε γλώσσες επιπέδου συστήματος όπως η Go, και C/C++ ανάλογα με το τι απαιτείται. Αποτελείται από ένα σύνολο

μικροϋπηρεσιών και μπορεί να αναπτυχθεί σε υποδομές νέφους και κέντρων δεδομένων, και να ελέγχεται από εντοπιστές όπως το Kubernetes [52]. Επιπρόσθετα είναι εξαιρετικά διαθέσιμο, δυναμικά επεκτάσιμο, και προσφέρει υψηλή απόδοση όσον αφορά την απόδοση (throughput) και την καθυστέρηση για την υλοποίηση των λειτουργιών και βρόγχων ελέγχου. Τέλος το μONOS διατίθεται σε μορφή που είναι έτοιμη για ανάπτυξη, με ένα σύνολο εργαλείων που απαιτούνται για συνεχή λειτουργία όπως για παράδειγμα Docker Images, Helm Charts, εργαλεία παρακολούθησης και αντιμετώπισης προβλημάτων. Στην Εικόνα 15 παρουσιάζεται η αρχιτεκτονική του μONOS



Εικόνα 15 Αρχιτεκτονική του μONOS

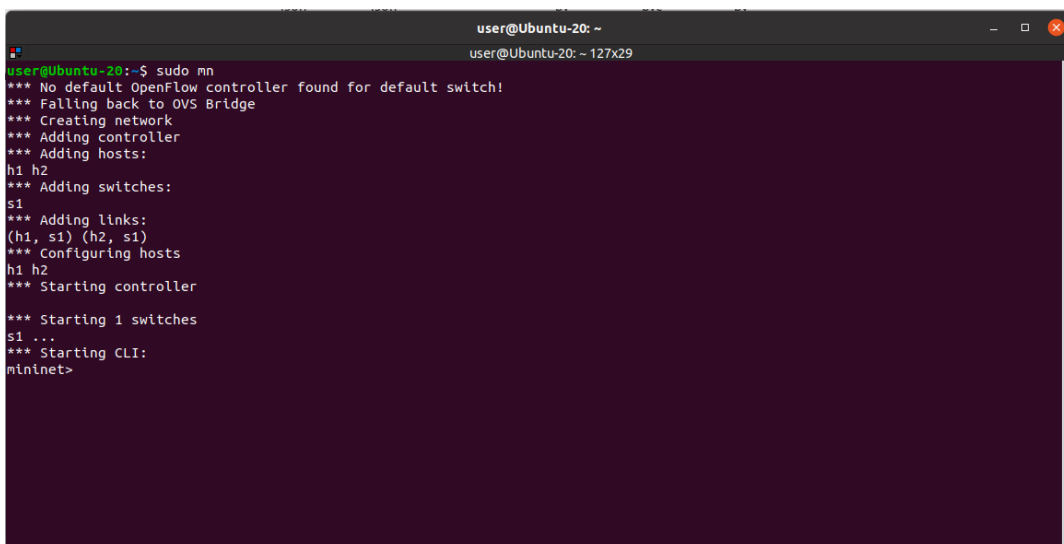
4.2 Mininet

Το λογισμικό ανοιχτού κώδικα Mininet [53], είναι ένας εξομοιωτής δικτύων που χρησιμοποιείται για την προσομοίωση εικονικών τοπολογιών SDN. Επιτρέπει την ανάπτυξη σε οποιοδήποτε υπολογιστή, και τα σχέδια ενός δικτύου μπορούν να μεταφέρονται απρόσκοπτα μεταξύ του Mininet και του πραγματικού υλικού που

τρέχει ήδη σε ζωντανές αναπτύξεις. Το μεγαλύτερο μέρος του είναι υλοποιημένο σε Python με υποστήριξη στη καινούργιες εκδόσεις (v2.3.0) για Python2 και Python3. Οι εικονικοί χρήστες είναι διεργασίες που εκτελούνται από το υποκείμενο λειτουργικό σύστημα Linux/Unix, και οι μεταγωγείς υποστηρίζουν το πρωτόκολλο OpenFlow για ευέλικτη δρομολόγηση και δικτύωση.

Ως προεπιλογή στο πρόγραμμα Mininet, ένας εικονικός μεταγωγέας δημιουργείται από τον μεταγωγέα λογισμικού πολλαπλών επιπέδων Open vSwitch [54]. Λόγο αυτού εκτελεί και λειτουργίες δρομολογητή, αφού μπορεί να επεξεργαστεί κίνηση στα επίπεδα δικτύου και ζεύξης δεδομένων. Είναι προγραμματισμένο στην γλώσσα C και υποστηρίζει όλες τις λειτουργίες ενός φυσικού μεταγωγέα. Ωστόσο, είναι σημαντικό να σημειωθεί ότι το Mininet είναι ευέλικτο και μπορεί να ρυθμιστεί έτσι ώστε να χρησιμοποιεί και άλλους τύπους μεταγωγών όπως το Behavioral Model version 2 – BMv2 [55]. Είναι γραμμένο σε C++ και ερμηνεύει ένα αρχείο JSON που δημιουργείται από τη μεταγλώττιση του προγράμματος P4, για τον καθορισμό της συμπεριφοράς επεξεργασίας πακέτων.

Τα πλεονεκτήματα του Mininet είναι ότι έχει υψηλή διαθεσιμότητα και είναι εύκολο στην εγκατάσταση και χρήση. Επίσης χρειάζεται πολύ λίγο χρόνο για την εκκίνηση και λειτουργία του και εξοικονομεί χρήματα λόγο του ότι οι εξομοιωτές είναι οικονομικά αποδοτικοί, αντί να χρησιμοποιούνται συσκευές υλικού Τέλος η σύνδεση με συσκευές δικτύου πραγματικού κόσμου είναι πολύ εύκολη.



```
user@Ubuntu-20: ~  
user@Ubuntu-20: ~ 127x29  
user@Ubuntu-20:~$ sudo mn  
*** No default OpenFlow controller found for default switch!  
*** Falling back to OVS Bridge  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2  
*** Adding switches:  
s1  
*** Adding links:  
(h1, s1) (h2, s1)  
*** Configuring hosts  
h1 h2  
*** Starting controller  
*** Starting 1 switches  
s1 ...  
*** Starting CLI:  
mininet>
```

Εικόνα 16 Το λογισμικό Mininet

4.3 Stratum

Το Stratum [56] είναι ένα λειτουργικό σύστημα μεταγωγέα ανοιχτού κώδικα, ανεξάρτητο από πυρίτιο (silicon-independent) για δίκτυα καθορισμένα από λογισμικό. Είναι ελεύθερα διαθέσιμο και μπορεί να εκτελεστεί σε ένα ευρύ φάσμα πλατφόρμων υλικού (όπως BMv2, Barefoot-Tofino). Αυτό το καθιστά πιο οικονομική και ευέλικτη επιλογή από τα ιδιόκτητα λειτουργικά συστήματα μεταγωγών. Επίσης το Stratum χρησιμοποιεί τη διεπαφή P4Runtime για τον καθορισμό της συμπεριφοράς προώθησης του μεταγωγέα. Ως αποτέλεσμα επιτρέπει στους διαχειριστές δικτύου να προσαρμόζουν τη συμπεριφορά του μεταγωγέα, με τέτοιο τρόπο έτσι ώστε να ανταποκρίνεται στις συγκεκριμένες ανάγκες τους. Επίσης εκθέτει ένα σύνολο ανοιχτών διεπαφών, συμπεριλαμβανομένων των gRPC, gNMI και gNOI, οι οποίες επιτρέπουν τον έλεγχο του από τον SDN ελεγκτή. Έτσι καθιστά εύκολη την ενσωμάτωση των μεταγωγέων Stratum σε υπάρχοντα δίκτυα SDN. Τέλος, οι μεταγωγείς με λειτουργικό σύστημα Stratum είναι ένα ισχυρό και ευέλικτο εργαλείο για τη δημιουργία δικτύων που καθορίζονται από λογισμικό. Αποτελούν μια καλή επιλογή για φορείς που θέλουν να μειώσουν την εξάρτηση από προμηθευτές, να βελτιώσουν την επεκτασιμότητα και να μειώσουν το κόστος των δικτύων τους.

4.4 Αλγόριθμοι εξισσορόπισης φόρτου

Ένας αλγόριθμος εξισορρόπησης φόρτου είναι η λογική που χρησιμοποιεί ένας εξισορροπητής φόρτου για τη διανομή της κυκλοφορίας δικτύου μεταξύ των διακομιστών. Για την μελέτη αυτή χρησιμοποιήθηκαν τέσσερεις αλγόριθμοι εξισορρόπισης φόρτου.

- **Equal Cost Multi-Path (ECMP)**

Το ECMP είναι μια στρατηγική δρομολόγησης δικτύου που επιτρέπει σε πακέτα κίνησης από την ίδια ροή ή συνεδρία, να μεταδίδονται κατά πολλές διαδρομές που έχουν ίσο κόστος. Το κόστος υπολογίζεται με βάση τους αλγόριθμους κατακερματισμού ECMP και τους μετρικούς υπολογισμούς δρομολόγησης. Αυτή η τεχνική χρησιμοποιεί πλήρως το εύρος ζώνης κατά μήκος συνδέσμων προς τον ίδιο προορισμό που διαφορετικά θα παρέμεναν αχρησιμοποίητοι, αυξάνοντας την απόδοση και την κίνηση εξισορρόπησης φόρτου [57].

Όταν ένας δρομολογητής ανακαλύψει πολλές διαδρομές προς έναν προορισμό, ο πίνακας δρομολόγησης ενημερώνεται με πολλαπλές καταχωρήσεις για αυτόν τον προορισμό. Κάθε φορά που γίνεται η προώθηση πακέτων, αποφασίζεται ποια θα είναι το επόμενο βήμα (next-hop) που θα χρησιμοποιηθεί. Για τη λήψη της απόφασης αυτής, η συσκευή λαμβάνει υπόψη πεδία κεφαλίδας πακέτων επιπέδου 3 που προσδιορίζουν τις ροές. Τέλος τα πακέτα για ένα δεδομένο ζεύγος πηγής-προορισμού είναι εγγυημένα ότι ακολουθούν την ίδια διαδρομή, ακόμη και αν υπάρχουν πολλές διαδρομές ενώ οι ροές κυκλοφορίας που προορίζονται για διαφορετικά ζεύγη τείνουν να ακολουθούν διαφορετικές διαδρομές.

- **Round Robin**

Ο αλγόριθμος Round Robin [58] είναι από τους πιο διαδεδομένους αλγόριθμους εξισορρόπησης φόρτου. Είναι ένας απλός τρόπος για τη διανομή μιας ροής δεδομένων σε μια ομάδα διακομιστών. Μία ροή δεδομένων χρήστη δρομολογείται σε διαθέσιμους διακομιστές σε κυκλική βάση. Όταν ο εξισορροπητής φόρτου, δρομολογήσει κίνηση σε όλους τους διακομιστές που υπάρχουν στην λίστα με την σειρά, τότε ο αλγόριθμος επαναλάβει τη διαδικασία ξανά από την αρχή. Χρησιμοποιώντας αυτήν τη μέθοδο, τα αιτήματα πελατών δρομολογούνται σε διαθέσιμους διακομιστές σε κυκλική βάση. Η εξισορρόπηση φόρτου Round Robin λειτουργεί καλύτερα όταν οι διακομιστές έχουν περίπου ίδιες υπολογιστικές δυνατότητες και χωρητικότητα αποθήκευσης.

- **HULA (Hop-by-Hop Utilization-aware Load Balancing Architecture)**

Ο αλγόριθμος εξισορρόπησης φόρτου HULA [59] είναι ένας αλγόριθμος που έχει σχεδιαστεί για μεγάλα δίκτυα κέντρων δεδομένων, που χρησιμοποιούν προγραμματιζόμενα επίπεδα δεδομένων. Ο HULA λειτουργεί χρησιμοποιώντας περιοδικά πακέτα ανιχνευτών (probes) για τη διανομή πληροφοριών χρήσης δικτύου σε μεταγωγείς στη μορφή διανύσματος απόστασης. Αυτό επιτρέπει στους μεταγωγείς να λαμβάνουν έγκαιρες και αποτελεσματικές αποφάσεις προώθησης για την ασταθή κίνηση των κέντρων δεδομένων. Οι ανιχνευτές καλύπτουν όλες τις επιθυμητές διαδρομές για εξισορρόπηση φόρτου και είναι ξεχωριστοί από τα πακέτα δεδομένων. Οι πληροφορίες που συλλέγονται από αυτούς τους ανιχνευτές, συνοψίζονται και αποθηκεύονται σε κάθε μεταγωγέα ως ένας πίνακας, που δίνει το καλύτερο επόμενο βήμα προς οποιονδήποτε προορισμό. [61] Αυτή η προσέγγιση επιτρέπει στους μεταγωγείς να ενημερώνουν τον ανιχνευτή για την καλύτερη downstream διαδρομή, και να τον στέλνουν σε άλλους upstream μεταγωγείς με αποτέλεσμα να γίνεται η διάδοση των πληροφοριών βέλτιστης διαδρομής σε ολόκληρο το δίκτυο.

Ένα από τα βασικά χαρακτηριστικά του HULA είναι ότι μειώνει τις απαιτήσεις κατάστασης (state requirements) για μεταγωγείς. Αντί να διατηρούν πληροφορίες

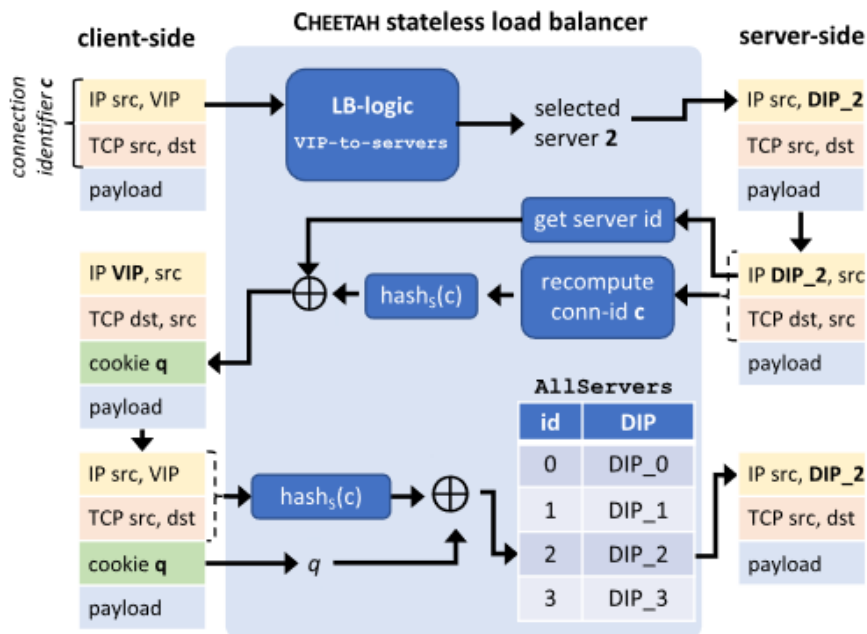
για την συμφόρηση που υπάρχει ανά διαδρομή για κάθε προορισμό, οι μεταγωγείς διατηρούν μόνο έναν πίνακα που αντιστοιχίζει τον προορισμό στο καλύτερο επόμενο βήμα. Τέλος ο αλγόριθμος HULA υλοποιείται στη γλώσσα P4, με αποτέλεσμα να επιτρέπεται σε ένα διαχειριστή, να τον τροποποιήσει ανάλογα με τις απαιτήσεις του δικτύου [60].

- **CHEETAH**

Το CHEETAH [62], [63] είναι ένα πλαίσιο εξισορρόπησης φόρτου υψηλής ταχύτητας που έχει σχεδιαστεί για να αντιμετωπίζει τις προκλήσεις της που υπάρχουν σε μεγάλα δίκτυα όπως κέντρα δεδομένων και πάροχοι υπηρεσιών. Στοχεύει στην ομοιόμορφη κατανομή φόρτου των εισερχόμενων συνδέσεων μεταξύ διακομιστών και στην συνέπεια ανά σύνδεση (Per Connection Consistency - PCC), η οποία διασφαλίζει ότι τα πακέτα που ανήκουν στην ίδια σύνδεση προωθούνται στον ίδιο διακομιστή, ακόμη και αν υπάρξει αλλαγή στον αριθμό των ενεργών διακομιστών. Το πλαίσιο έχει σχεδιαστεί για να υποστηρίζει υπάρχον μηχανισμούς εξισορρόπησης φόρτου (πχ. Round Robin) και μπορεί να εφαρμοστεί με κατάσταση (stateful), και χωρίς κατάσταση (stateless).

Στην υλοποίηση χωρίς κατάσταση όταν ο εξισοροποιητής φόρτου λάβει το πρώτο πακέτο μιας σύνδεσης, εξάγει το σύνολο των διακομιστών που υπάρχουν από στον πίνακα *VIP-to-Servers* Εικόνα 17, και επιλέγει έναν από τους διακομιστές σύμφωνα με τον προρυθμισμένο μηχανισμό εξισορρόπησης φόρτου (πχ. Round Robin), και προωθεί το πακέτο. Έπειτα, κωδικοποιεί τις πληροφορίες αντιστοίχισης διακομιστή στις κεφαλίδες όλων των πακέτων χρησιμοποιώντας μια προσέγγιση που βασίζεται σε *cookie*. Υπολογίζει τον κατακερματισμό του αναγνωριστικού σύνδεσης με ένα αλάτι *S* (το οποίο είναι άγνωστο στους πελάτες), και έπειτα το κάνει *XOR* με το αναγνωριστικό του διακομιστή, και προσθέτει την έξοδο του *XOR* στην κεφαλίδα του πακέτου ως *cookie*. Το αλάτι που χρησιμοποιείται είναι το ίδιο για όλες τις συνδέσεις. Στη συνέχεια, όταν ο εξισοροποιητής φόρτου λάβει οποιοδήποτε επόμενο πακέτο που ανήκει στην ίδια σύνδεση, εξάγει το *cookie* από την κεφαλίδα του πακέτου, υπολογίζει τον κατακερματισμό του αναγνωριστικού σύνδεσης με το αλάτι *S*, κάνει *XOR* την έξοδο του κατακερματισμού με το *cookie* και χρησιμοποιεί

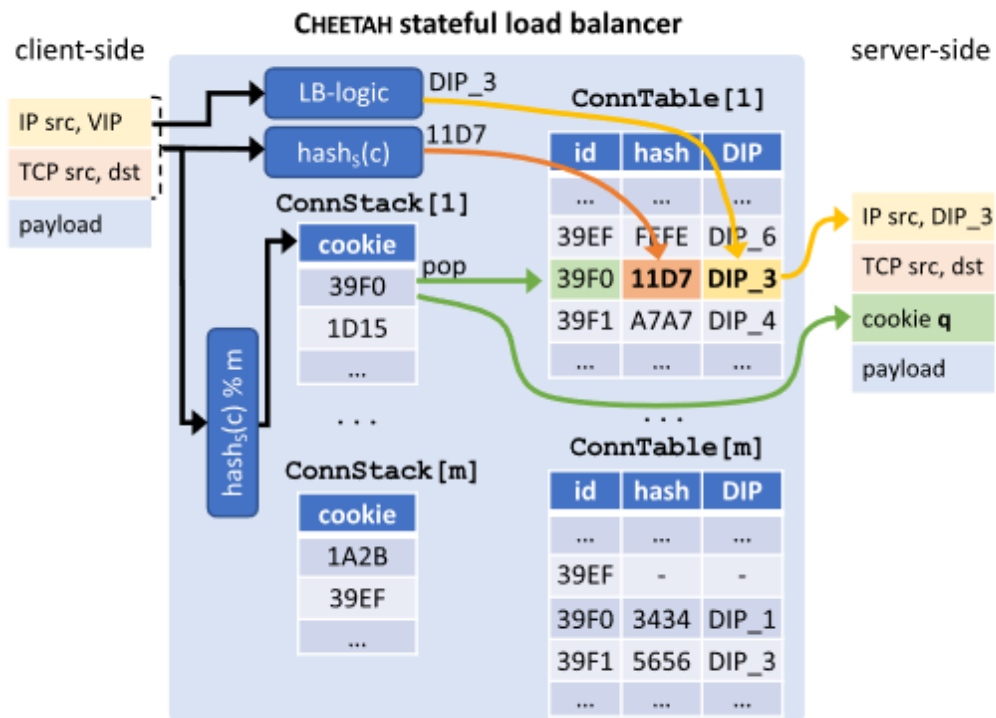
την έξοδο του XOR ως αναγνωριστικό του διακομιστή. Αυτό επιτρέπει στον εξισορροπητή φόρτου να ανακτήσει την *DIP* προορισμού του διακομιστή από τον πίνακα *AllServers* και να προωθήσει το πακέτο ανάλογα.



Εικόνα 17 Αλγόριθμος Cheetah χωρίς κατάσταση [63]

Στην υλοποίηση με κατάσταση, το CHEETAH χρησιμοποιεί μια υβριδική προσέγγιση που αποθηκεύει αναγνωριστικά σύνδεσης, και αντιστοιχίσεων διακομιστή στο επίπεδο δεδομένων. Κάθε φορά που χρειάζεται να εγκατασταθεί μια νέα σύνδεση, το CHEETAH εφαρμόζει ένα ευρετήριο (lookup) στη δομή συνδέσεων *ConnStack*, Εικόνα 18 και το ενσωματώνει ως μέρος του *cookie* στην κεφαλίδα του πακέτου. Αποθηκεύει επίσης τον επιλεγμένο διακομιστή και τον κατακερματισμό του αναγνωριστικού σύνδεσης με ένα αλάτι *S*, στην αντίστοιχη καταχώρηση πίνακα. Αυτή η τιμή κατακερματισμού επιτρέπει στον εξισορροπητή φόρτου να φιλτράρει κακόβουλες προσπάθειες παρέμβασης σε νόμιμες ροές κυκλοφορίας. Κάθε φορά που ένα πακέτο που ανήκει σε μια υπάρχουσα σύνδεση φθάνει στον εξισορροπητή φόρτου, το CHEETAH εξάγει το ευρετήριο από το *cookie* και το χρησιμοποιεί για να

εκτελέσει γρήγορα μια αναζήτηση μόνο στο πίνακα συνδέσεων *ConnTable*. Ως αποτέλεσμα ξεπερνά τους περιορισμούς των υφιστάμενων εξισορροπητών φορτίου με κατάσταση, εφαρμόζοντας έναν γρήγορο μηχανισμό που βασίζεται σε στοίβα. Αυτή η προσέγγιση εξασφαλίζει γρήγορους χρόνους εισαγωγής και επεκτασιμότητα για μεγάλο αριθμό συνδέσεων.



Εικόνα 18 Αλγόριθμος Cheetah με κατάσταση [63]

Το κριτήριο με τα οποίο έγινε η επιλογή των παραπάνω αλγόριθμων είναι το επίπεδο λειτουργίας τους στο OSI. Ο HULA και ο CHEETAH είναι L7 αλγόριθμοι εξισορρόπησης φόρτου και ο ECMP και ο Round Robin είναι L4. Μπορεί επίσης να χαρακτηριστεί ότι ο ECMP και ο Round Robin είναι αλγόριθμοι όπου οι δυνατότητες και τα μειονεκτήματά τους είναι γνωστά λόγω της ευρείας χρήσης τους ενώ ο HULA και ο CHEETAH έχουν αναπτυχθεί πρόσφατα.

4.5 Υλοποίηση

Για την υλοποίηση και την σύγκριση των αλγόριθμων εξισορρόπησης φόρτου ECMP, Round Robin, HULA και CHEETAH, χρειάστηκε η υλοποίηση ενός δικτύου σε περιβάλλον NG-SDN. Το περιβάλλον υλοποιήθηκε σε λειτουργικό σύστημα Ubuntu 20.04 LTS. Το Ubuntu Linux παρέχει ένα ευέλικτο και αρθρωτό σύστημα που βοηθά στην εύκολη ανάπτυξη και τον εντοπισμό σφαλμάτων. Τα δίκτυα που υλοποιήθηκαν δοκιμάστηκαν σε υπολογιστή με επεξεργαστή Ryzen5 3600 και 16GB RAM.

Χρησιμοποιήθηκε το περιβάλλον `ngsdn-tutorial` [64] που παρέχεται από τον ONF. Ο αλγόριθμος ECMP συμπεριλαμβανόταν στο `ngsdn-tutorial`, ενώ οι υπόλοιποι πάρθηκαν από το Github [65]. Επίσης χρησιμοποιήθηκε το εργαλείο Docker Compose για την λειτουργία της δομής των Docker Containers, τα οποία περιλαμβάνουν τον ελεγκτή ONOS έκδοσης `v2.2.2`, και τον εξομοιωτή δικτύων Mininet έκδοσης `v2.3.0`. Ο τύπος των εικονικών μεταγωγών που παρέχει το `ngsdn-tutorial` μέσω του Mininet είναι ο μεταγωγέας BMv2 με το λειτουργικό σύστημα Stratum.

Οι εφαρμογές και μικροϋπηρεσίες που χρησιμοποιούνται από τον ελεγκτή ONOS για την λειτουργία του δικτύου φαίνονται στον Πίνακα 5.

Title	App ID	Category
BMv2 Drivers	org onosproject.drivers.bmv2	Drivers
Basic Pipelines	org onosproject.pipelines.basic	Pipeline
DHCP Relay Agent	org onosproject.dhcprelay	Utility
Default Drivers	org onosproject.drivers	Drivers
FIB Push Manager {FPM} Route Receiver	org onosproject.fpm	Utility
Fabric Pipeline	org onosproject.pipelines.fabric	Pipeline
General Device Provider	org onosproject.generaldeviceprovider	Provider
Host Location Provider	org onasproject.hostprovider	Provider
Host Probing Provider	org onosproject.hostprobingprovider	Provider
IPV6 RA Generator	org onosproject.routeradvertisement	Traffic Engineering
LLDP Link Provider	org onasproject.idpprovider	Provider

Multicast traffic control	org.onosproject.mcast	Traffic Engineering
Network Config Host Provider	org onosproject.netcfgghostprovider	Provider
ONOS GUI2	org onosproject gui2	Graphical User Interface
OpenFlow Base Provider	org onosproject.openflow - base	Provider
OpenFlow Provider Suite	org.onosproject.openflow	Provider
Optical Network Model	org.onosproject.optical - model	Optical
P4Runtime Drivers	org onosproject.drivers.p4runtime	Drivers
P4Runtime Protocol Subsystem	org onasproject.protocols.p4runtime	Protocol
P4Runtime Provider	org.onosproject p4runtime	Provider
Port Load Balance Service	org onasproject portloadbalancer	Utilities
Route Service Server	org onosproject.route - service	Utility
Segment Routing	org.onosproject.segmentrouting	Traffic Steering
Stratum Drivers	org onasproject.drivers.stratum	Drivers
Trellis Troubleshooting Toolkit	org onosproject.t3	Utilities
gNMI Drivers	org.onosproject.drivers.gnmi	Drivers
gNMI Protocol Subsystem	org.onosproject.protocols.gnmi	Protocol
gNOI Drivers	org.onosproject.drivers.gnoi	Drivers
gNOI Protocol Subsystem	org onasproject protocols.gnoi	Protocol
gRPC Protocol Subsystem	org onasproject.protocols.grpc	Protocol

Πίνακας 5 Εφαρμογές και μικροϋπηρεσίες του ONOS

Οι αλγόριθμοι είναι υλοποιημένοι στην γλώσσα προγραμματισμού P4 και όταν το αρχείο .p4 μεταγλωττιστεί εξάγει δύο αρχεία Εικόνα 19:

bmv2.json: Περιέχει την διαμόρφωση και την λειτουργία του μεταγωγέα Stratum_bmv2 σε μορφή JSON.

p4info.txt: Περιέχει το μήνυμα P4Info, το οποίο είναι μια αναπαράσταση του προγράμματος P4 ανεξάρτητη από πρωτόκολλο. Το μήνυμα P4Info περιλαμβάνει πληροφορίες σχετικά με τους πίνακες, τις ενέργειες και άλλα αντικείμενα του προγράμματος P4 και χρησιμοποιείται από το επίπεδο ελέγχου για την κατασκευή κλήσεων API του P4Runtime. Είναι ένα κρίσιμο στοιχείο για την επικοινωνία μεταξύ του προγράμματος P4 (επίπεδο δεδομένων) και του επιπέδου ελέγχου.

```
user@Ubuntu-20: ~/Downloads/ngsdn-tutorial/advanced_new93x30
user@Ubuntu-20:~/Downloads/ngsdn-tutorial/advanced_new$ make p4-build
*** Building P4 program...
docker run --rm -v /home/user/Downloads/ngsdn-tutorial/advanced_new:/workdir -w /workdir open
networking/p4c:stable \
  p4c-bm2-ss --arch v1model -o p4src/build/bmv2.json \
  --p4runtime-files p4src/build/p4info.txt --wdisable=unsupported \
  p4src/main.p4
*** P4 program compiled successfully! Output files are in p4src/build
user@Ubuntu-20:~/Downloads/ngsdn-tutorial/advanced_new$ ls p4src/build
bmv2.json  p4info.txt
user@Ubuntu-20:~/Downloads/ngsdn-tutorial/advanced_new$
```

Εικόνα 19 Μεταγλώππιση αρχείου .p4

Έπειτα για τη σύγκριση των αλγορίθμων δημιουργήθηκαν τέσσερις διαφορετικές τοπολογίες μέσω του λογισμικού Mininet. Οι τοπολογίες δημιουργήθηκαν με βάση το Leaf-Spine σχέδιο αναφοράς Trellis [66]. Οι τοπολογίες που χρησιμοποιήθηκαν φαίνονται παρακάτω.

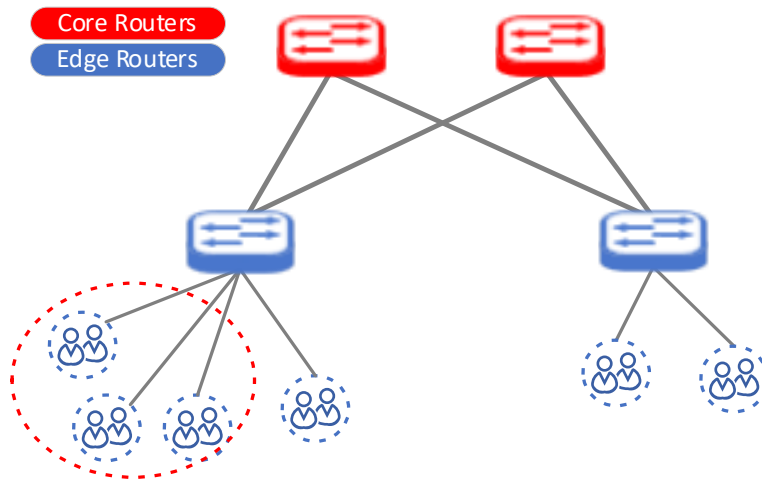
Στην πρώτη τοπολογία (Εικόνα 20) το δίκτυο 2x2 αποτελείται από δύο κεντρικούς δρομολογητές και δύο δρομολογητές άκρης. Ο κάθε δρομολογητής άκρης έχει κάποιους χρήστες συνδεδεμένους σε αυτόν.

Στην δεύτερη τοπολογία (Εικόνα 21) το δίκτυο 3x4 αποτελείται από τρεις κεντρικούς δρομολογητές και τέσσερις δρομολογητές άκρης. Ο κάθε δρομολογητής άκρης έχει κάποιους χρήστες συνδεδεμένους σε αυτόν.

Στην τρίτη τοπολογία (Εικόνα 22) το δίκτυο 7x8 αποτελείται από επτά κεντρικούς δρομολογητές και οκτώ δρομολογητές άκρης.

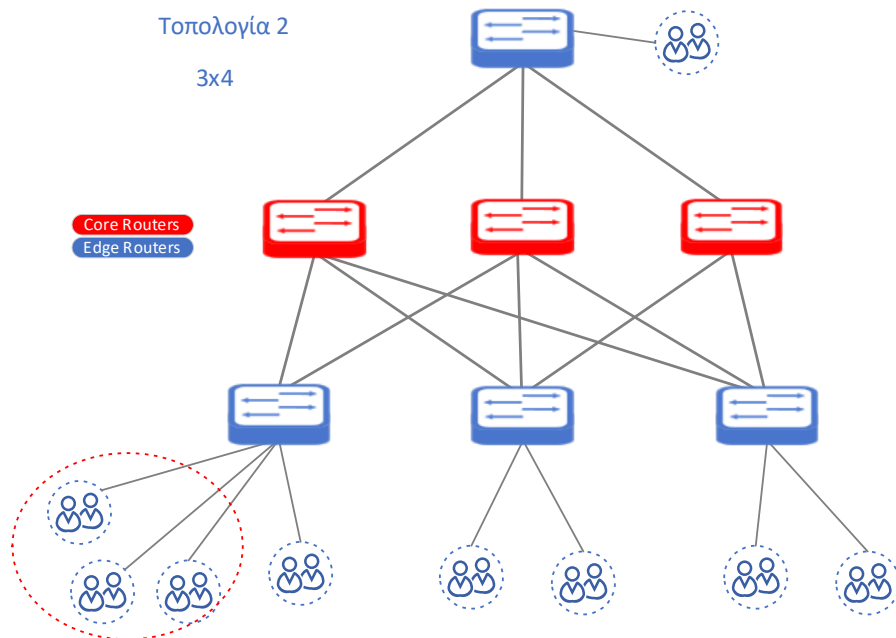
Στην τέταρτη τοπολογία (Εικόνα 23) το δίκτυο που επιλέχθηκε είναι το κεντρικό δίκτυο της Ισπανίας RedIris, το οποίο πάρθηκε από το The Internet Topology Zoo [67]. Αποτελείται από πέντε κεντρικούς δρομολογητές και οκτώ δρομολογητές άκρης.

Τοπολογία 1
2x2



Εικόνα 20 Τοπολογία 1. 2x2

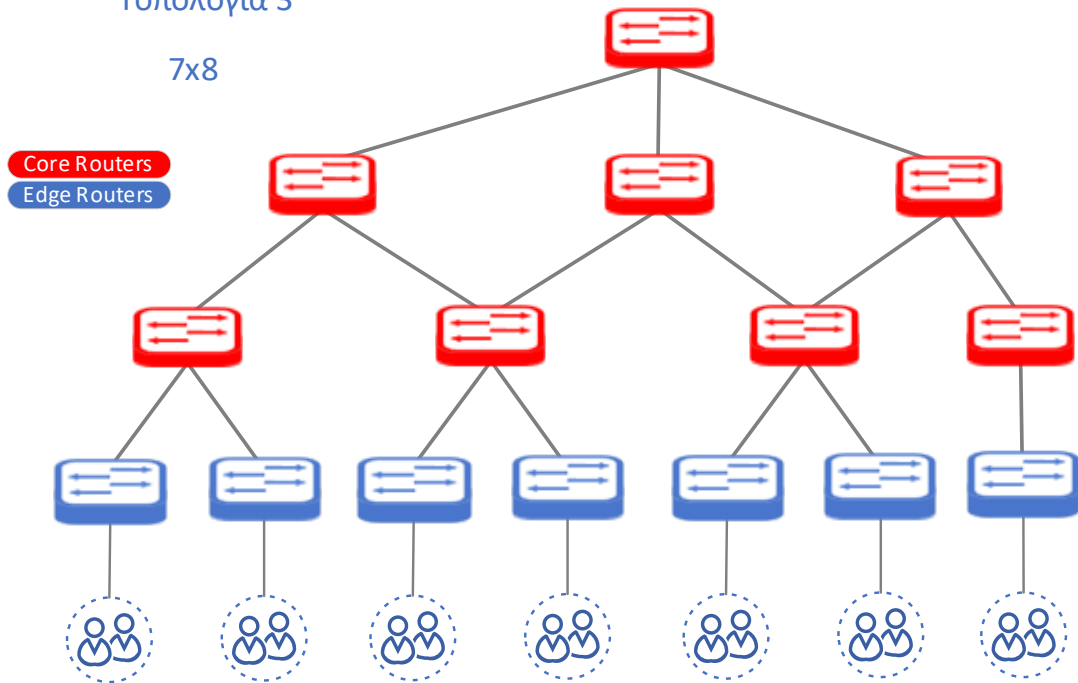
Τοπολογία 2
3x4



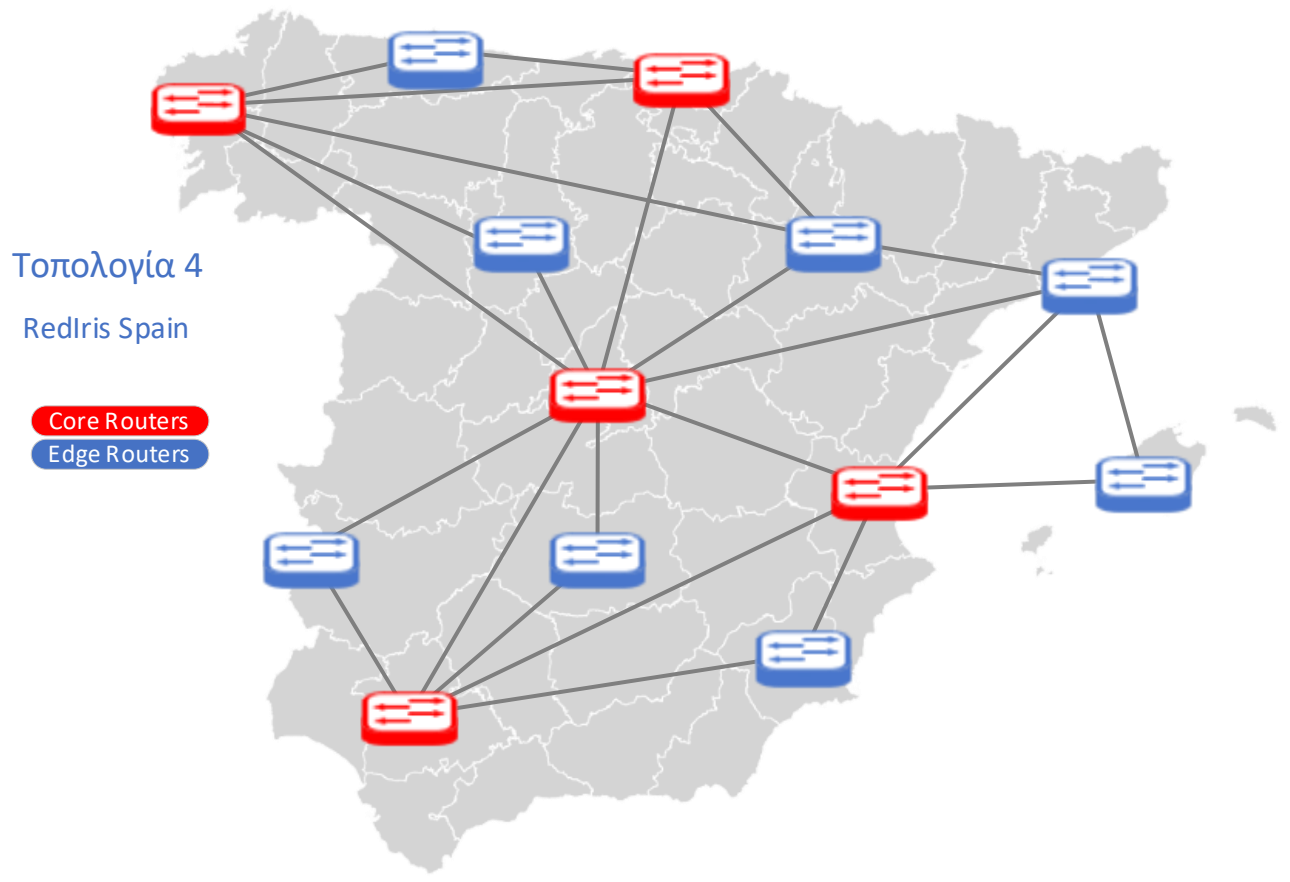
Εικόνα 21 Τοπολογία 2. 3x4

Τοπολογία 3

7x8



Εικόνα 22 Τοπολογία 3. 7x8



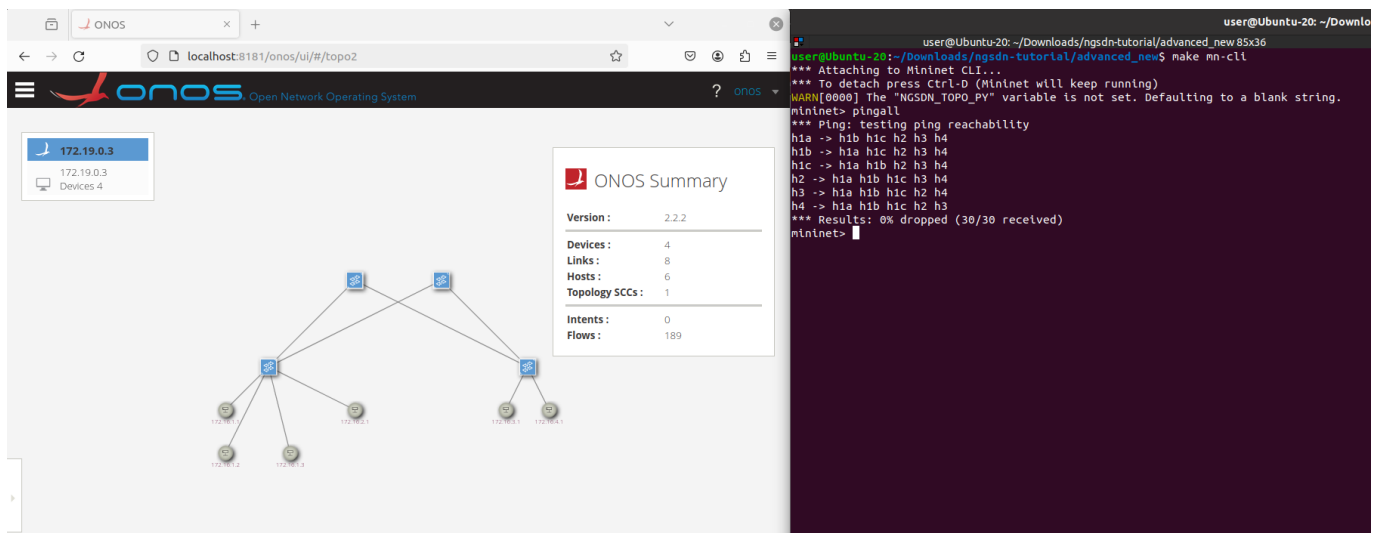
Εικόνα 23 Τοπολογία 4. RedIris Spain

Ακολούθως, γίνεται εκκίνηση του περιβάλλοντος NG-SDN, και οι τοπολογίες εισάγονται στο λογισμικό Mininet με το αρχείο topology.py που περιέχει σε Python την διαμόρφωση του δικτύου, και στον τον ελεγκτή ONOS με το αρχείο netcfg.json που περιέχει επίσης την διαμόρφωση του δικτύου σε μορφή JSON.

```
user@Ubuntu-20:~/Downloads/ngsdn-tutorial/advanced_new$ make start
*** Starting ONOS and Mininet (topology.py)...
[+] Running 3/3
  ✓ Network advanced_new_default Created 0.1s
  ✓ Container mininet Started 0.1s
  ✓ Container onos Started 0.0s
user@Ubuntu-20:~/Downloads/ngsdn-tutorial/advanced_new$ make netcfg
*** Pushing netcfg.json to ONOS...
curl --fail -sSL --user onos:rocks --no-proxy localhost -X POST -H 'Content-Type:application/json' \
  http://localhost:8181/onos/v1/network/configuration -d@./mininet/netcfg.json
user@Ubuntu-20:~/Downloads/ngsdn-tutorial/advanced_new$
```

Εικόνα 24 Εκκίνηση περιβάλλοντος NG-SDN

Τέλος αφού ξεκινήσει το ONOS, διαμορφωθούν σωστά οι μεταγωγείς, και εισαχθούν τα κατάλληλα αρχεία διαμόρφωσης δικτύου στο ONOS και στο Mininet, από το γραφικό περιβάλλον του ONOS μπορεί να φανεί η τοπολογία και από το Mininet η σωστή λειτουργία του δικτύου.

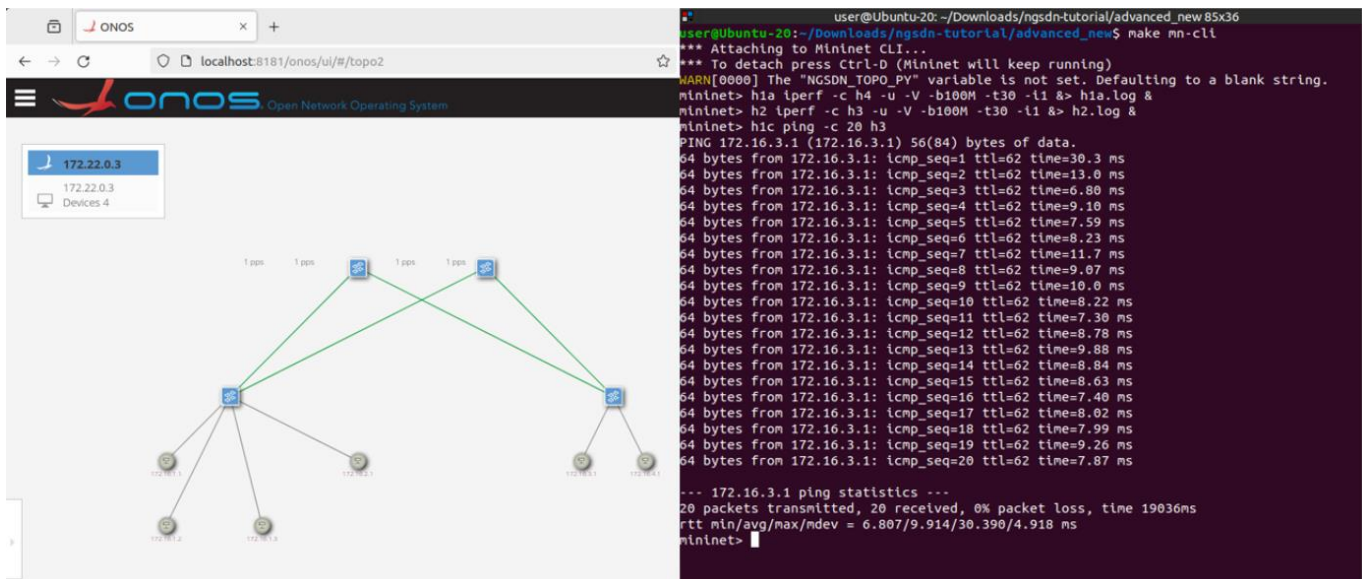


Εικόνα 25 Λειτουργία δικτύου στο γραφικό περιβάλλον ONOS και Mininet

Κεφάλαιο 5

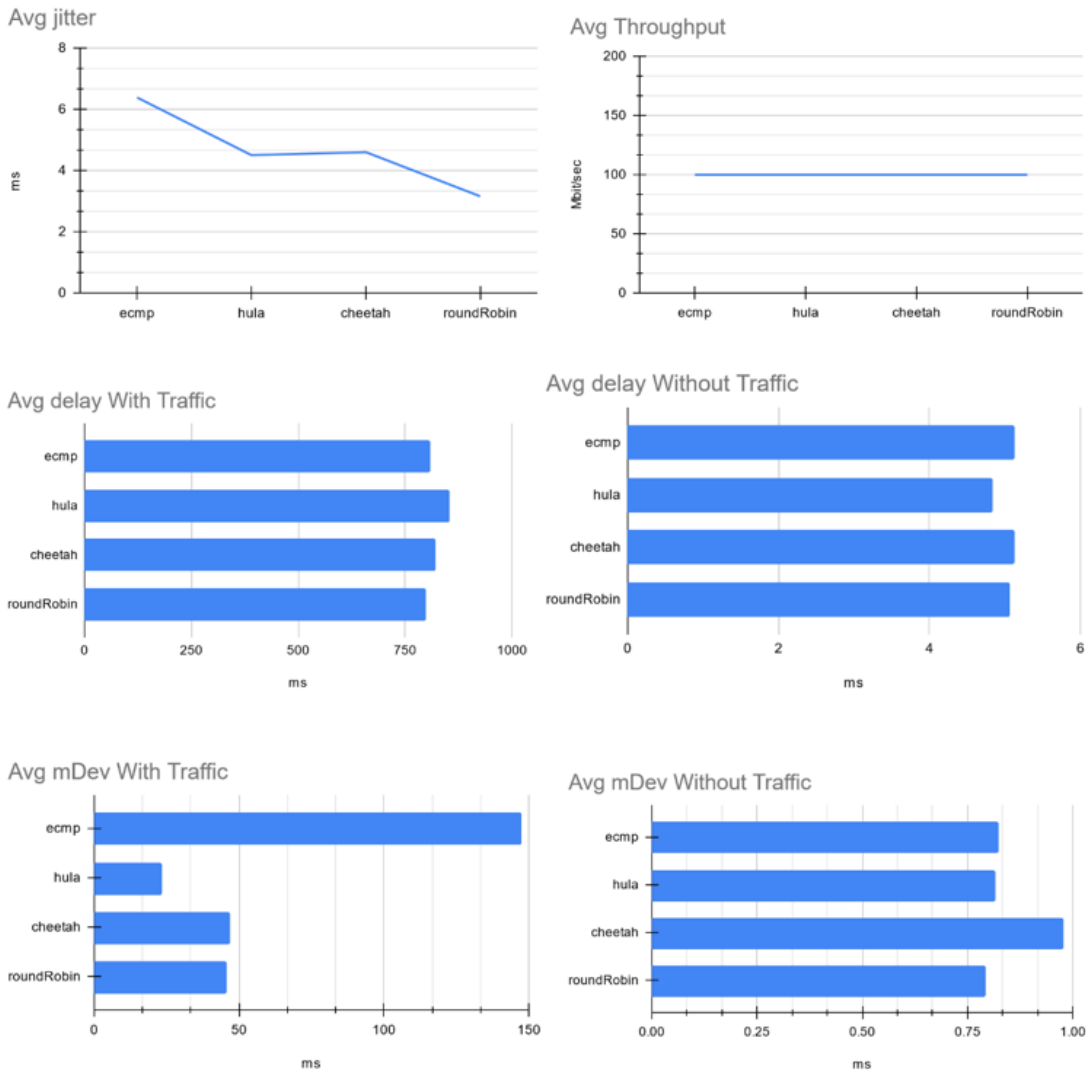
Αποτελέσματα

Για τις μετρήσεις και τα αποτελέσματα, από κάθε τοπολογία λήφθηκαν μερικά δείγματα. Οι μετρήσεις που λήφθηκαν είναι ο μέσος όρος της διακύμανσης καθυστέρησης (Average Jitter), η μέση απόδοση του δικτύου (Average Throughput), η μέση καθυστέρηση με και χωρίς κίνηση (Average Delay), και η μέση τυπική απόκλιση με και χωρίς κίνηση (Average mDev). Οι μετρήσεις για τα Jitter, Delay και mDev λήφθηκαν μέσω της διεργασίας Ping, και το Throughput μέσω της διεργασίας Iperf που προσφέρεται από το περιβάλλον του Mininet. Στην Εικόνα 26 φαίνεται η κίνηση του δικτύου στο γραφικό περιβάλλον του ONOS και στο Mininet οι εντολές που δόθηκαν για την λήψη των μετρήσεων. Η διαδικασία που παρουσιάζεται στην εικόνα επαναλήφθηκε τέσσερις φορές για κάθε αλγόριθμο που συγκρίνεται σε όλες τις τοπολογίες. Ακολουθούν τα αποτελέσματα.



Εικόνα 26 Λήψη μετρήσεων

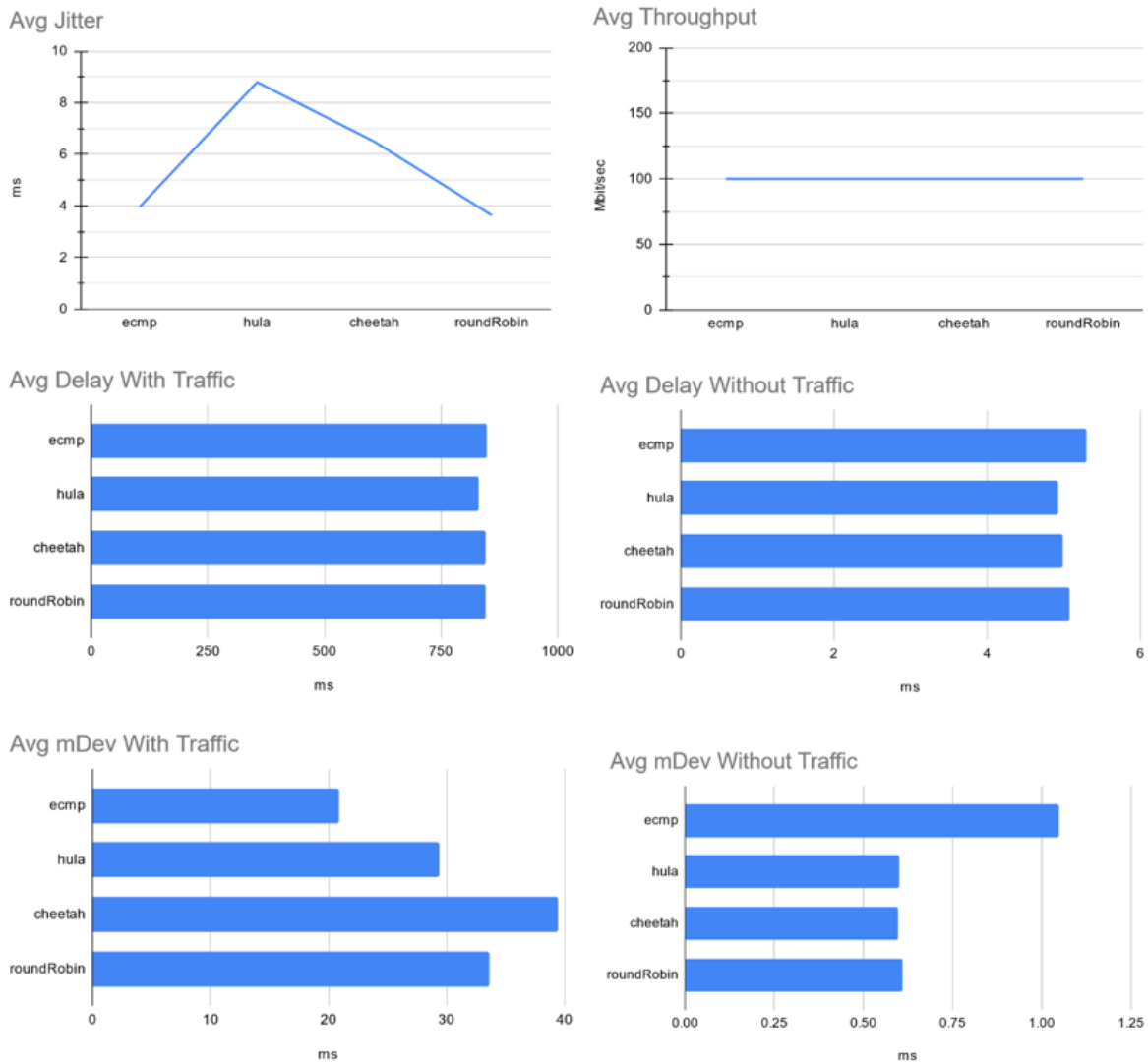
5.1 Τοπολογία 2x2



Εικόνα 27 Αποτελέσματα τοπολογίας 2x2

Από τις μετρήσεις στην Τοπολογία 2x2 παρατηρείται μέσω της Εικόνας 27, ότι ο αλγόριθμος με την μεγαλύτερη διακύμανση καθυστέρησης είναι ο ECMP και με την μικρότερη ο Round Robin. Η απόδοση του δικτύου είναι η ίδια για όλους τους αλγορίθμους. Για την καθυστέρηση παρατηρείται ότι η επίδοση όλων των αλγορίθμων είναι σχεδόν ίδια, αλλά ο HULA φαίνεται να έχει την περισσότερη καθυστέρηση με κίνηση, και την λιγότερη χωρίς, με μικρή όμως διαφορά. Την μεγαλύτερη μέση τυπική απόκλιση με κίνηση την έχει ECMP, ενώ την μικρότερη ο HULA, και χωρίς κίνηση την μεγαλύτερη τυπική απόκλιση ο CHEETAH και οι υπόλοιποι αλγόριθμοι περίπου το ίδιο.

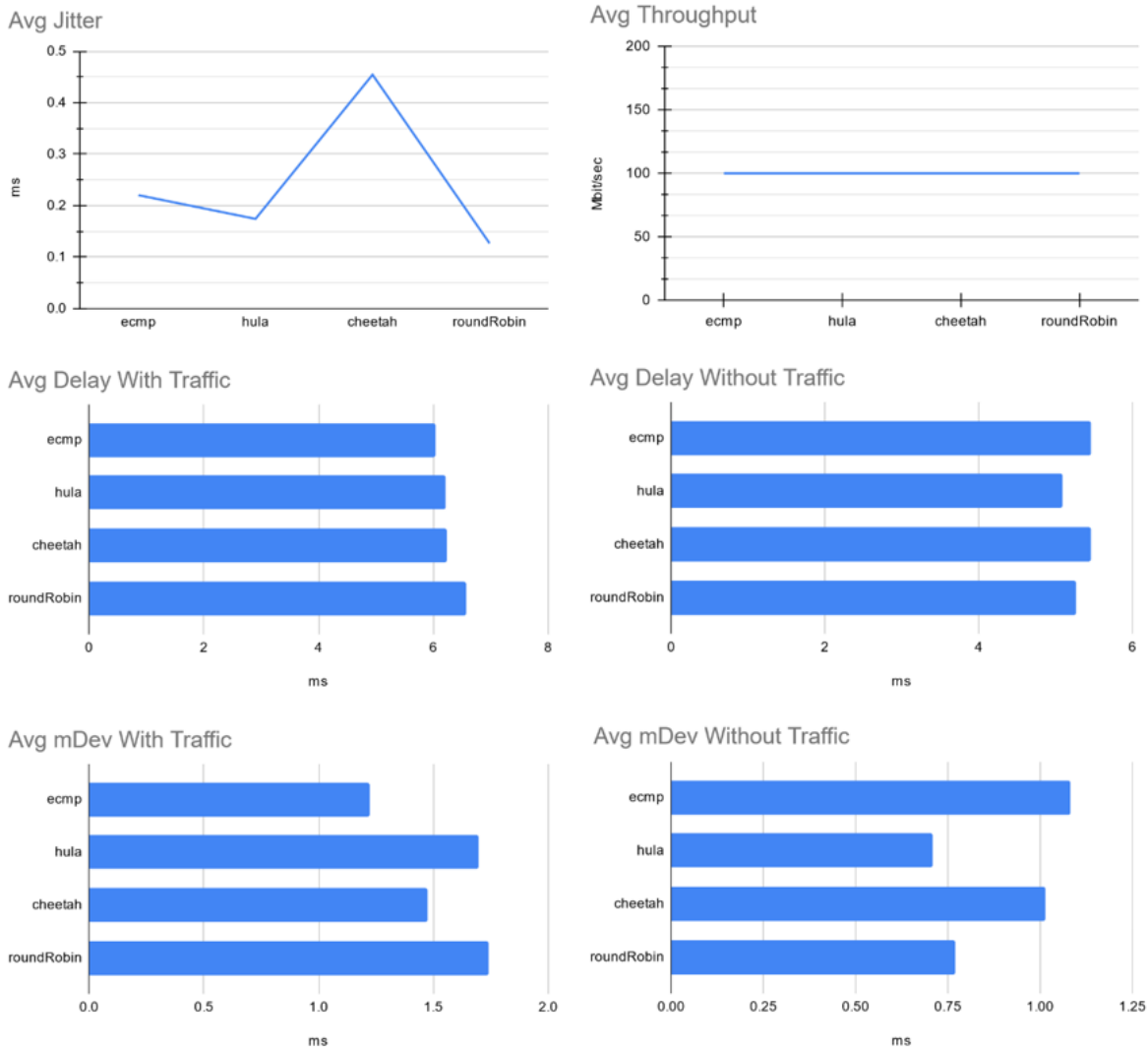
5.2 Τοπολογία 3x4



Εικόνα 28 Αποτελέσματα τοπολογίας 3x4

Στην Τοπολογία 3x4 παρατηρείται από την Εικόνα 28 ότι ο αλγόριθμος με την μεγαλύτερη διακύμανση καθυστέρησης είναι ο HULA και με την μικρότερη ο Round Robin με μικρή διαφορά από τον ECMP. Η απόδοση του δικτύου είναι η ίδια για όλους τους αλγόριθμους. Για την καθυστέρηση παρατηρείται ότι η επίδοση όλων των αλγορίθμων είναι σχεδόν ίδια, αλλά ο ECMP φαίνεται να έχει την περισσότερη καθυστέρηση με κίνηση και χωρίς, με μικρή όμως διαφορά. Την μεγαλύτερη μέση τυπική απόκλιση με κίνηση την έχει CHEETAH, ενώ την μικρότερη ο ECMP. Χωρίς κίνηση την μεγαλύτερη τυπική απόκλιση την έχει ο ECMP και οι υπόλοιποι αλγόριθμοι περίπου το ίδιο.

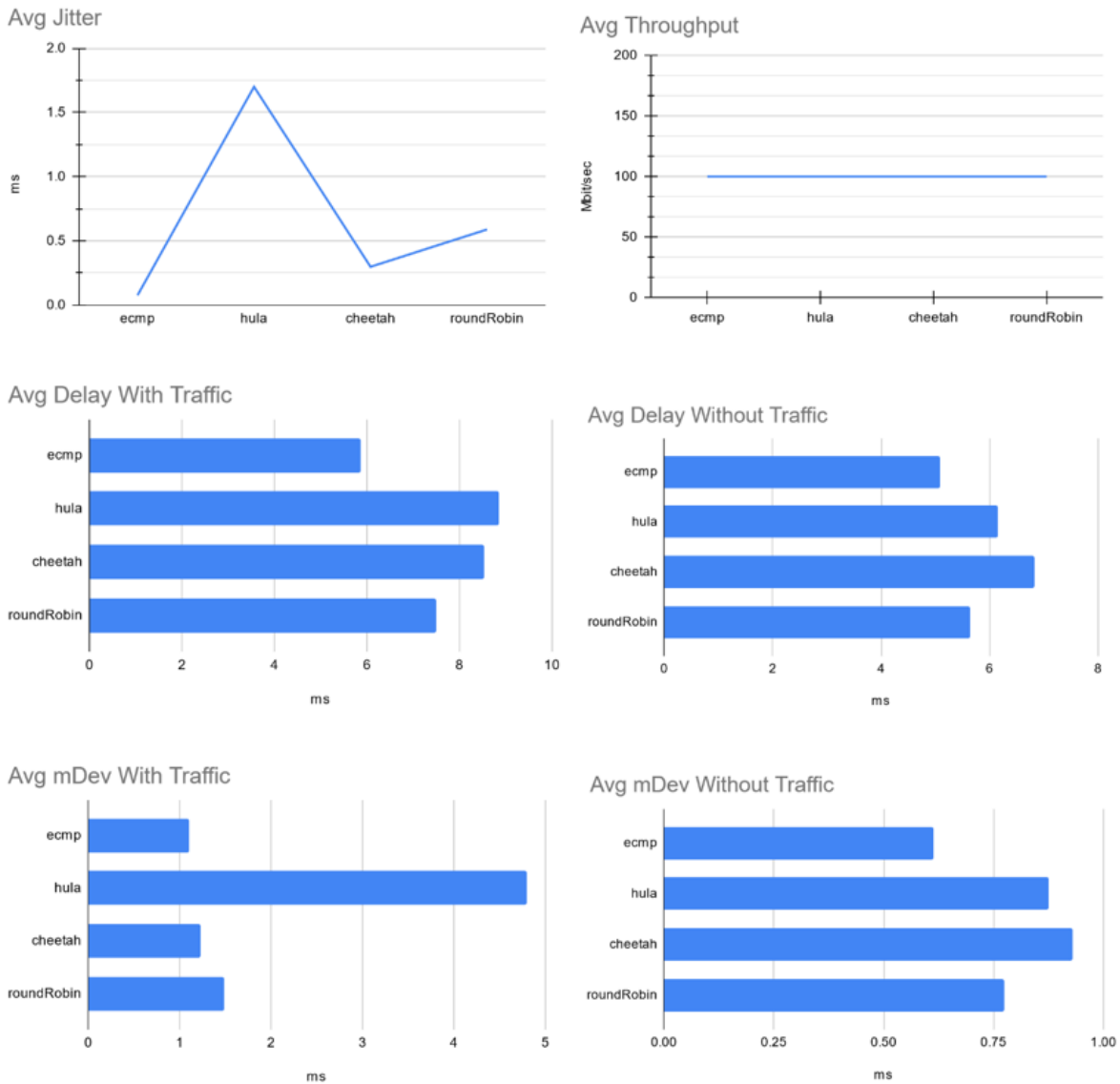
5.3 Τοπολογία 7x8



Εικόνα 29 Αποτελέσματα τοπολογίας 7x8

Στην Τοπολογία 7x8 παρατηρείται από την Εικόνα 29 ότι ο αλγόριθμος με την μεγαλύτερη διακύμανση καθυστέρησης είναι ο CHEETAH και με την μικρότερη ο Round Robin. Η απόδοση του δικτύου είναι η ίδια για όλους τους αλγόριθμους. Για την καθυστέρηση παρατηρείται ότι η επίδοση όλων των αλγορίθμων είναι σχεδόν ίδια, αλλά ο Round Robin φαίνεται να έχει την περισσότερη καθυστέρηση με κίνηση. Χωρίς κίνηση Ο ECMP και CHEETAH φαίνεται να έχουν την μεγαλύτερη καθυστέρηση με μικρή όμως διαφορά. Την μεγαλύτερη μέση τυπική απόκλιση με κίνηση την έχει RoundRobin, ενώ την μικρότερη ο ECMP, και χωρίς κίνηση την μεγαλύτερη τυπική απόκλιση ο ECMP και την μικρότερη ο HULA.

5.4 Τοπολογία RedIris Spain



Εικόνα 30 Αποτελέσματα τοπολογίας RedIris Spain

Στην Τοπολογία RedIris Spain παρατηρείται από την Εικόνα 30 ότι ο αλγόριθμος με την μεγαλύτερη διακύμανση καθυστέρησης είναι ο HULA και με την μικρότερη ο ECMP. Η απόδοση του δικτύου είναι η ίδια για όλους τους αλγόριθμους. Για την καθυστέρηση παρατηρείται ο HULA φαίνεται να έχει την περισσότερη καθυστέρηση με κίνηση και ο ECMP την μικρότερη. Χωρίς κίνηση ο CHEETAH φαίνεται να έχει την μεγαλύτερη καθυστέρηση και την λιγότερη ο ECMP. Την μεγαλύτερη μέση τυπική απόκλιση με κίνηση την έχει HULA, ενώ την μικρότερη ο ECMP, και χωρίς κίνηση την μεγαλύτερη τυπική απόκλιση ο CHEETAH και την μικρότερη ο ECMP.

5.5 Συγκεντρωτικά Αποτελέσματα

Στις τέσσερις παραπάνω τοπολογίες ο κάθε αλγόριθμος είχε πολλές φορές αναμενόμενη ανταπόκριση και συμπεριφορά και άλλες όχι. Ο ECMP παρατηρείται ότι σχεδόν σε όλες τις τοπολογίες, ενώ μπορεί να έχει την μικρότερη καθυστέρηση με κίνηση έχει όμως υψηλή διακύμανση. Ο HULA και ο CHEETAH ενώ φτιάχτηκαν για μεγάλες τοπολογίες δεν φαίνεται να έχουν σταθερή απόδοση σε αυτές. Ο Round Robin σχεδόν σε όλες τις τοπολογίες δεν είναι ούτε ο καλύτερος αλλά ούτε και ο χειρότερος, όμως βρίσκεται σε κατάλληλες συνθήκες λειτουργίες, αφού δουλεύει καλύτερα όταν οι συσκευές δικτύου έχουν ίδιους υπολογιστικούς πόρους πράγμα που ισχύει για τις τοπολογίες. Οι τοπολογίες 2x2 και 3x4 όπως φαίνεται και στα σχήματά τους έχουν VLAN, και παρατηρείται στα αποτελέσματά τους ότι οι αριθμοί είναι πολύ υψηλότεροι από τις τοπολογίες 7x8 και RedIris που δεν έχουν. Τέλος αν μετρήσει κάποιος τον αριθμό των συσκευών στην μεγαλύτερη τοπολογία, θα διαπιστώσει ότι είναι 15. Ο αριθμός αυτός δεν είναι τυχαίος αφού στην περίπτωση που οι συσκευές γίνονταν 16, ο ελεγκτής αδυνατούσε να δημιουργήσει σωστά τις συνδέσεις. Φαίνεται να υπάρχει κάποιος περιορισμός στο περιβάλλον ngsdn-tutorial που να αποτρέπει την δημιουργία περισσότερων συνδέσεων.

Κεφάλαιο 6

Συμπεράσματα & Μελλοντικές Επεκτάσεις

6.1 Συμπεράσματα

Τα Δίκτυα Καθορισμένα από Λογισμικό έχουν φέρει επανάσταση στη διαχείριση των δικτύων, αποσυνδέοντας το επίπεδο ελέγχου από το επίπεδο δεδομένων, επιτρέποντας έτσι τον προγραμματισμό και τη δυναμική διαμόρφωση των δικτύων. Η εξισορρόπηση φόρτου, είναι μια κρίσιμη λειτουργία που στοχεύει τόσο στη βελτιστοποίηση της χρήσης των πόρων, όσο και τη διασφάλιση της ποιότητας των υπηρεσιών. Στην προσπάθεια ανακάλυψης των δυνατοτήτων των SDN έχουν προκύψει διαφορετικοί αλγόριθμοι εξισορρόπησης φορτίου, ο καθένας από τους οποίους καλύπτει συγκεκριμένες απαιτήσεις και προτεραιότητες δικτύου.

Αυτή η διπλωματική εργασία, διερεύνησε την ενσωμάτωση των εξισορροπητών φόρτου στην αρχιτεκτονική SDN, εξετάζοντας διάφορες τοπολογίες, αλγόριθμους και μετρήσεις, έτσι ώστε να επιτευχθεί η αποτελεσματικότερη κατανομή της κυκλοφορίας σε ένα δίκτυο. Με βάση τη συγκριτική μελέτη αλγορίθμων που αναπτύχθηκε, οι μετρήσεις παίζουν καθοριστικό ρόλο στην καθοδήγηση των αποφάσεων επιλογής κάποιου αλγορίθμου για εξισορρόπηση φόρτου. Τα αποτελέσματα που εξάχθηκαν καθιστούν ορατό ότι, ο κάθε αλγόριθμος μπορεί να έχει την καλύτερη απόδοση σε μία τοπολογία και την χειρότερη στην επόμενη. Για την βέλτιστη χρήση του κάθε αλγορίθμου, είναι καλό να ληφθεί υπόψιν το περιβάλλον χρήσης, το μέγεθος δικτύου που πρόκειται να χρησιμοποιηθεί, καθώς και οι σωστές συνθήκες που ο κάθε ένας φτιάχτηκε για να λειτουργεί.

Συμπερασματικά, οι εξισορροπητές φόρτου που βασίζονται σε SDN, αντιπροσωπεύουν μια σημαντική πρόοδο στη διαχείριση της κυκλοφορίας του δικτύου. Η εγγενής ευελιξία, η δυνατότητα προγραμματισμού και οι προηγμένες μετρήσεις τους, επιτρέπουν την αποτελεσματική χρήση των πόρων, τη βελτιωμένη ποιότητα υπηρεσιών και τη δυναμική προσαρμογή στις μεταβαλλόμενες απαιτήσεις του δικτύου. Καθώς η έρευνα συνεχίζει να διερευνά το πλήρες δυναμικό του SDN, οι εξισορροπητές φορτίου θα διαδραματίσουν αναμφίβολα έναν ολοένα και πιο κρίσιμο ρόλο στη διασφάλιση της ευελιξίας, της επεκτασιμότητας και της απόδοσης των μελλοντικών δικτύων.

6.2 Μελλοντικές Επεκτάσεις

Η μελέτη που παρουσιάζεται σε αυτή τη διπλωματική εργασία μπορεί να επεκταθεί και να γίνει σκαλοπάτι για την περαιτέρω έρευνα στον τομέα των Αλγορίθμων Εξισορρόπησης Φόρτου σε Δίκτυα Καθορισμένα από Λογισμικό. Σε μελλοντικές μελέτες και επεκτάσεις θα μπορούσαν να γίνουν τα ακόλουθα:

- Η ανάπτυξη έξυπνων αλγορίθμων εξισορρόπησης φορτίου με επίγνωση του περιεχομένου που αξιοποιούν πλήρως τις δυνατότητες των SDN. Οι τεχνικές μηχανικής μάθησης και τεχνητής νοημοσύνης μπορούν να τη βελτιστοποιήσουν την κατανομή των πόρων και την πρόβλεψη των προτύπων κυκλοφορίας.
- Καθώς οι εξισορροπητές φορτίου γίνονται κεντρικά στοιχεία δικτύου, η ανθεκτικότητα τους έναντι επιθέσεων και τρωτών σημείων πρέπει να αντιμετωπιστεί μέσω ασφαλών πρωτοκόλλων επικοινωνίας και μηχανισμών ελέγχου πρόσβασης.
- Η συγκριτική μελέτη αλγορίθμων έχει υλοποιηθεί σε εικονικοποιημένο περιβάλλον με τον ελεγκτή και τις συσκευές να είναι συσκευές λογισμικού. Σε επόμενη επέκταση η χρήση υλικών συσκευών θα αντιπροσώπευε ορθότερα την λειτουργία των εξισορροπητών φόρτου χωρίς τους περιορισμούς που μπορεί να υπάρχουν από το σύστημα εικονικοποίησης.
- Με τον αυξανόμενο ρυθμό ανάπτυξης των δορυφορικών δικτύων πλέγματος χαμηλής τροχιάς Γης (Low Earth Orbit - LEO), είναι σημαντικό να διασφαλιστεί η βέλτιστη χρήση πόρων που διαθέτει κάθε δορυφόρος. Μηχανισμοί εξισορρόπησης φόρτου είναι πιθανόν να χρειαστούν σε αυτά τα δίκτυα όσο ο πλανήτης στρέφεται ολοένα και περισσότερο προς τα δορυφορικά δίκτυα.

Βιβλιογραφικές Αναφορές

- [1] 'Cisco VNI Complete Forecast Highlights'. Accessed: Jan. 10, 2024. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2022_Forecast_Highlights.pdf
- [2] A. Nunez, J. Ayoka, M. Z. Islam, and P. Ruiz, *A Brief Overview of Software-Defined Networking*. 2023. doi: 10.48550/arXiv.2302.00165.
- [3] F. Liu, G. Kibalya, S. Kumar Svn, and P. Zhang, 'Challenges of Traditional Networks and Development of Programmable Networks', 2022, pp. 37–61. doi: 10.1007/978-3-030-89328-6_3.
- [4] 'Software-Defined Networking: The New Norm for Networks'. Accessed: Jan. 12, 2024. [Online]. Available: <https://opennetworking.org/wp-content/uploads/2011/09/wp-sdn-newnorm.pdf>
- [5] 'Network Protocol Definition | Computer Protocol | Computer Networks | CompTIA', CompTIA.org. Accessed: Jan. 13, 2024. [Online]. Available: <https://www.comptia.org/content/guides/what-is-a-network-protocol>
- [6] N. Sabitha, H. Jayasree, and A. V. K. Prasad, 'Virtualization of Traditional Networks using SDN', vol. 7, no. 3, 2019.
- [7] M. Casado, M. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, *ETHANE: Taking Control of the Enterprise*, vol. 37. 2007, p. 12. doi: 10.1145/1282380.1282382.
- [8] K. Benzekki, A. El Fergougui, and A. El Belrhiti El Alaoui, 'Software-defined networking (SDN): A survey', *Secur. Commun. Netw.*, vol. 9, Feb. 2017, doi: 10.1002/sec.1737.
- [9] S. Badotra, 'A Review Paper on Software Defined Networking', *Int. J. Adv. Comput. Res.*, vol. 8, Mar. 2017.
- [10] E. Haleplidis, K. Pentikousis, S. Denazis, J. Hadi Salim, D. Meyer, and O. Koufopavlou, 'Software-Defined Networking (SDN): Layers and Architecture Terminology', RFC Editor, RFC7426, Jan. 2015. doi: 10.17487/rfc7426.
- [11] 'What is a REST API?' Accessed: Jan. 18, 2024. [Online]. Available: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
- [12] 'OpenFlow - ONF', Open Networking Foundation. Accessed: Jan. 19, 2024. [Online]. Available: <https://opennetworking.org/product-certification/>
- [13] H. Gang, 'What Is YANG?', Huawei. Accessed: Jan. 22, 2024. [Online]. Available: <https://info.support.huawei.com/info-finder/encyclopedia/en/YANG.html>
- [14] M. Paliwal, D. Shrimankar, and O. Tembhurne, 'Controllers in SDN: A Review Report', *IEEE Access*, vol. 6, pp. 36256–36270, 2018, doi: 10.1109/ACCESS.2018.2846236.
- [15] 'Ryu SDN Framework'. Accessed: Jan. 19, 2024. [Online]. Available: <https://ryu-sdn.org/>
- [16] 'Open Network Operating System (ONOS) SDN Controller for SDN/NFV Solutions', Open Networking Foundation. Accessed: Jan. 19, 2024. [Online]. Available: <https://opennetworking.org/onos/>
- [17] A. Tootoonchian and Y. Ganjali, 'HyperFlow: a distributed control plane for OpenFlow', in *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, in INM/WREN'10. USA: USENIX Association, Apr. 2010, p. 3.
- [18] D. Erickson, 'The beacon openflow controller', in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, in HotSDN '13. New York, NY, USA: Association for Computing Machinery, Aug. 2013, pp. 13–18. doi: 10.1145/2491185.2491189.
- [19] 'Floodlight Controller - Confluence'. Accessed: Jan. 19, 2024. [Online]. Available: <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/overview>

- [20] P. Bojović, Z. Bojovic, J. Šuh, and N. Gavrić, *Design of SDN manageable switch*. 2019.
- [21] N. McKeown *et al.*, 'OpenFlow: enabling innovation in campus networks', *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008, doi: 10.1145/1355734.1355746.
- [22] 'OpenFlow Switch Specification v1.5.1'. Accessed: Jan. 19, 2024. [Online]. Available: <https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>
- [23] 'OpenFlow Switch Specifications v1.3.1'. Accessed: Jan. 19, 2024. [Online]. Available: <https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf>
- [24] 'Software Defined Networks Challenges and Future Direction of Research', ResearchGate. Accessed: Jan. 22, 2024. [Online]. Available: https://www.researchgate.net/publication/330599431_Software_Defined_Networks_Challenges_and_Future_Direction_of_Research
- [25] 'NG-SDN', Open Networking Foundation. Accessed: Jan. 22, 2024. [Online]. Available: <https://opennetworking.org/reference-designs/ng-sdn/>
- [26] B. O'Connor, 'Realizing Next Generation SDN/NFV'.
- [27] 'Next-Gen SDN - ONF'. Accessed: Jan. 28, 2024. [Online]. Available: <https://onfstaging1.opennetworking.org/wp-content/uploads/2019/09/Connect-2019-NG-SDN.pdf>
- [28] 'P4 – Language Consortium'. Accessed: Jan. 22, 2024. [Online]. Available: <https://p4.org/>
- [29] 'gRPC', gRPC. Accessed: Jan. 28, 2024. [Online]. Available: <https://grpc.io/>
- [30] 'OpenConfig - gRPC Network Management Interface (gNMI) specification'. Accessed: Jan. 28, 2024. [Online]. Available: <https://www.openconfig.net/docs/gnmi/gnmi-specification/>
- [31] 'OpenConfig - Guidelines for developing gNOI microservices'. Accessed: Jan. 28, 2024. [Online]. Available: https://www.openconfig.net/docs/gnoi/best_practices/
- [32] 'Protocol Buffers'. Accessed: Jan. 28, 2024. [Online]. Available: <https://protobuf.dev/>
- [33] 'OpenConfig'. Accessed: Jan. 28, 2024. [Online]. Available: <https://www.openconfig.net/>
- [34] J. Sonchack, D. Loehr, J. Rexford, and D. Walker, 'Lucid: a language for control in the data plane', in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, Virtual Event USA: ACM, Aug. 2021, pp. 731–747. doi: 10.1145/3452296.3472903.
- [35] 'The Frenetic Project'. Accessed: Jan. 27, 2024. [Online]. Available: <http://frenetic-lang.org/>
- [36] 'Data Plane Development Kit (DPDK)', DPDK. Accessed: Jan. 31, 2024. [Online]. Available: <https://www.dpdk.org/>
- [37] A. Liatifis, P. Sarigiannidis, V. Argyriou, and T. Lagkas, 'Advancing SDN from OpenFlow to P4: A Survey', *ACM Comput. Surv.*, vol. 55, no. 9, p. 186:1-186:37, Jan. 2023, doi: 10.1145/3556973.
- [38] P. Bosshart *et al.*, 'Programming Protocol-Independent Packet Processors'. arXiv, May 15, 2014. Accessed: Jan. 28, 2024. [Online]. Available: <http://arxiv.org/abs/1312.1719>
- [39] 'P4~16~ Language Specification'. Accessed: Jan. 28, 2024. [Online]. Available: <https://p4.org/p4-spec/docs/P4-16-v1.0.0-spec.html#sec-grammar-intro>
- [40] 'P4 Architectures - Barefoot Networks'. Accessed: Jan. 31, 2024. [Online]. Available: <https://opennetworking.org/wp-content/uploads/2020/12/p4-ws-2017-p4-architectures.pdf>
- [41] T. Osiński, 'Network Prototyping Made Easy with P4 and Python', Open Networking Foundation. Accessed: Jan. 31, 2024. [Online]. Available: <https://opennetworking.org/news-and-events/blog/network-prototyping-made-easy-with-p4-and-python/>
- [42] 'P4Runtime Specification'. Accessed: Jan. 22, 2024. [Online]. Available: <https://p4.org/p4-spec/p4runtime/main/P4Runtime-Spec.html>
- [43] 'What Is Load Balancing? How Load Balancers Work', NGINX. Accessed: Jan. 24, 2024. [Online]. Available: <https://www.nginx.com/resources/glossary/load-balancing/>
- [44] 'Load Balancing 101: The Evolution to Application Delivery Controllers'. Accessed: Jan. 24, 2024. [Online]. Available: <https://www.f5.com/pdf/white-papers/evolution-adc-wp.pdf>

- [45] M. Albowarab, N. Zakaria, and Z. Zainal Abidin, 'Load balancing algorithms in software defined network', vol. 7, pp. 686–693, Apr. 2018.
- [46] X. Qin, H. Jiang, A. Manzanares, X. Ruan, and S. Yin, 'Dynamic load balancing for I/O-intensive applications on clusters', *ACM Trans. Storage*, vol. 5, no. 3, pp. 1–38, Nov. 2009, doi: 10.1145/1629075.1629078.
- [47] 'Application Delivery Controller (ADC) - Load Balancer & More', NGINX. Accessed: Jan. 24, 2024. [Online]. Available: <https://www.nginx.com/resources/glossary/application-delivery-controller/>
- [48] 'Oracle VM VirtualBox'. Accessed: Jan. 24, 2024. [Online]. Available: <https://www.virtualbox.org/>
- [49] 'Docker: Accelerated Container Application Development'. Accessed: Jan. 24, 2024. [Online]. Available: <https://www.docker.com/>
- [50] T. Bourke, *Server Load Balancing*. O'Reilly Media, Inc., 2001.
- [51] 'Open Network Operating System (ONOS)'. Accessed: Jan. 29, 2024. [Online]. Available: <https://docs.onosproject.org/>
- [52] 'Kubernetes -Production Grade Container Orchestration'. Accessed: Jan. 29, 2024. [Online]. Available: <https://kubernetes.io/>
- [53] 'Mininet: An Instant Virtual Network on Your Laptop (or Other PC) - Mininet'. Accessed: Jan. 30, 2024. [Online]. Available: <http://mininet.org/>
- [54] 'Open vSwitch'. Accessed: Jan. 30, 2024. [Online]. Available: <https://www.openvswitch.org/>
- [55] 'p4lang/behavioral-model'. p4language, Jan. 30, 2024. Accessed: Jan. 30, 2024. [Online]. Available: <https://github.com/p4lang/behavioral-model>
- [56] 'Stratum - Switch Operating System', Open Networking Foundation. Accessed: Jan. 30, 2024. [Online]. Available: <https://opennetworking.org/stratum/>
- [57] F. Rhamdani, N. A. Suwastika, and M. A. Nugroho, 'Equal-Cost Multipath Routing in Data Center Network Based on Software Defined Network', in *2018 6th International Conference on Information and Communication Technology (ICoICT)*, May 2018, pp. 222–226. doi: 10.1109/ICoICT.2018.8528730.
- [58] 'LoadBalancing-P4/roundRobin at main · itsprajwalshavi21/LoadBalancing-P4', GitHub. Accessed: Jan. 31, 2024. [Online]. Available: <https://github.com/itsprajwalshavi21/LoadBalancing-P4/tree/main/roundRobin>
- [59] V. Safronov, 'vadisaf/HULApp'. Jan. 28, 2021. Accessed: Jan. 31, 2024. [Online]. Available: <https://github.com/vadisaf/HULApp>
- [60] N. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford, 'HULA: Scalable Load Balancing Using Programmable Data Planes', in *Proceedings of the Symposium on SDN Research*, Santa Clara CA USA: ACM, Mar. 2016, pp. 1–12. doi: 10.1145/2890955.2890968.
- [61] M. Kulkarni, B. Goswami, and J. Paulose, 'HULA: Dynamic and Scalable Load Balancing Mechanism for Data Plane of SDN', in *2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Feb. 2023, pp. 1–9. doi: 10.1109/ICECCT56650.2023.10179849.
- [62] 'cheetahlb/cheetah-p4'. The Cheetah Load Balancer - NSLab @ KTH, Dec. 14, 2023. Accessed: Jan. 31, 2024. [Online]. Available: <https://github.com/cheetahlb/cheetah-p4>
- [63] T. Barbette, E. Wu, D. Kostic, G. Q. Maguire, P. Papadimitratos, and M. Chiesa, 'Cheetah: A High-Speed Programmable Load-Balancer Framework With Guaranteed Per-Connection-Consistency', *IEEEACM Trans. Netw.*, vol. 30, no. 1, pp. 354–367, Feb. 2022, doi: 10.1109/TNET.2021.3113370.

- [64] 'opennetworkinglab/ngsdn-tutorial: Hands-on tutorial to learn the building blocks of the Next-Gen SDN architecture'. Accessed: Jan. 31, 2024. [Online]. Available: <https://github.com/opennetworkinglab/ngsdn-tutorial/tree/advanced>
- [65] 'GitHub: Developer Platform', GitHub. Accessed: Feb. 21, 2024. [Online]. Available: <https://github.com/>
- [66] 'Trellis - Reference Design', Open Networking Foundation. Accessed: Jan. 31, 2024. [Online]. Available: <https://opennetworking.org/reference-designs/trellis/>
- [67] 'The Internet Topology Zoo'. Accessed: Jan. 31, 2024. [Online]. Available: <http://www.topology-zoo.org/dataset.html>