



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

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

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΕΞΙΣΟΡΡΟΠΗΣΗ ΦΟΡΤΙΟΥ ΣΤΑ ΔΙΚΤΥΑ ΚΑΘΟΡΙΖΟΜΕΝΑ ΑΠΟ
ΛΟΓΙΣΜΙΚΟ**

Παναγιώτης Μπούσης

Επιβλέπων: Ελευθέριος Στεργίου,
ΔΕΠ, Καθηγητής

Άρτα, Ιούνιος, 2025

Load Balancing in Software Defined Networks (SDN)

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής

Ελευθέριος Στεργίου,

ΔΕΠ, Καθηγητής

2. Μέλος επιτροπής

Γρηγόριος Δουμένης

ΔΕΠ, Επίκουρος Καθηγητής

3. Μέλος επιτροπής

Σπυριδούλα Μαργαρίτη,

ΕΔΙΠ, Α

© Μπούσης, Παναγιώτης, 2025.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

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

Μπούσης, Παναγιώτης

Υπογραφή

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εκπονήθηκε στο τμήμα Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Ιωαννίνων για το μεταπτυχιακό πρόγραμμα Πληροφορικής και Δικτύων. Για την εκπόνησή αυτής θα ήθελα να ευχαριστήσω θερμά τον επιβλέπων καθηγητή κ. Ελευθέριο Στεργίου καθώς επίσης την καθηγήτρια κα. Σπυριδούλα Μαργαρίτη για την πολύτιμη βοήθεια της κατά την διάρκεια εκπόνησης της διπλωματικής μου εργασίας. Τέλος, ευχαριστώ την οικογένειά μου και τους φίλους μου για την στήριξή τους σε όλα τα έτη των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Τα κέντρα δεδομένων είναι εγκαταστάσεις όπου στεγάζονται υποδομές πληροφορικής και δικτύων για την παροχή υπηρεσιών και εφαρμογών. Στα κέντρα δεδομένων χρησιμοποιούμε τοπολογίες 2-tier, 3-tier και FatTree. Η τοπολογία FatTree είναι μία ειδική Clos τοπολογία με χαμηλό κόστος υλοποίησης, πανομοιότυπο εύρος ζώνης σε οποιαδήποτε διχοτόμηση, απλότητα κατασκευής.

Βασική απαίτηση σε τέτοιου είδους υποδομές είναι η εξασφάλιση αποδοτικής και ορθής λειτουργίας των κέντρων δεδομένων. Για την ενίσχυση της ποιότητας υπηρεσιών χρησιμοποιούνται διάφορες τεχνικές. Μία από αυτές είναι και η εξισορρόπηση φορτίου. Η εξισορρόπηση φορτίου είναι μία μέθοδος για την ανάθεση φορτίου στους πόρους του δικτύου. Στα δίκτυα καθοριζόμενα από λογισμικό υπάρχουν διάφοροι αλγόριθμοι εξισορρόπησης φορτίου. Στα πλαίσια αυτής της εργασίας, περιγράφονται συνοπτικά τα δίκτυα SDN, η τοπολογία FatTree, η εξισορρόπηση φορτίου και ταξινομούνται οι αλγόριθμοι εξισορρόπησης φορτίου. Επιπλέον παρουσιάζονται αναλυτικά οι αλγόριθμοι ECMP, Hedera, PureSDN και EFatTree. Οι αλγόριθμοι αυτοί χρησιμοποιούνται στην πειραματική μελέτη και εκεί παρουσιάζονται τα συγκριτικά αποτελέσματα απόδοσής τους

Με την βοήθεια του Mininet εφαρμόσαμε σε τοπολογία FatTree τους αλγορίθμους ECMP, Hedera, PureSDN. Η σύγκριση έγινε με βάση τις μετρικές throughput, First packet RTT, Packet loss, mean deviation of round trip delay.

Όπως διαπιστώνεται ο ECMP αλγόριθμος παρουσιάζει καλύτερη συμπεριφορά στις FatTree τοπολογίες δικτύων.

Λέξεις-κλειδιά: Software defined networks, FatTree, ECMP, Load Balancing.

ABSTRACT

Data centers are the facilities that stores IT and the network infrastructure to provide services and applications. In data centers, we use 2-tier, 3-tier and FatTree topologies. FatTree is a special Clos topology with low implementation cost, identical bandwidth in any partition, simplicity of construction.

A basic requirement in such infrastructures is to ensure efficient and proper operation of data centers. Various techniques are used to enhance the quality of services. One of them is load balancing. Load balancing is a method for assigning load to network resources. In software-defined networks, there are various load balancing algorithms. In this paper, we studied the ECMP, Hedera, PureSDN algorithms.

With the assistance of Mininet, we executed the ECMP, Hedera, PureSDN algorithms on the FatTree network. The comparison was made based on the metrics throughput, First packet RTT, Packet loss, mean deviation of round-trip delay.

In this paper, SDN networks, FatTree topology, load balancing is briefly described and load balancing algorithms are classified. In addition, ECMP, Hedera, PureSDN and Conga algorithms are presented in detail. These algorithms are used in experimental study and their comparative performance results are presented there. As can be seen, the ECMP algorithm exhibits better behavior in FatTree network topologies.

Keywords: Software defined networks, FatTree, ECMP, Load Balancing.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΥΧΑΡΙΣΤΙΕΣ	iv
ΠΕΡΙΛΗΨΗ	v
ABSTRACT	vi
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	vii
1 Εισαγωγή.....	2
1.1 Κίνητρο.....	2
1.2 Ορισμός προβλήματος	3
1.3 Δομή εργασίας.....	3
2 Θεωρητικό υπόβαθρο.....	4
2.1 SDN – Software Defined networks	4
2.1.1 Η σύνδεση του SDN με το NFV.....	4
2.1.2 Πλεονεκτήματα της χρήσης των δικτύων καθοριζόμενα από λογισμικό	5
2.2 Τα κέντρα δεδομένων – data center	14
2.2.1 Η Τοπολογία Fat-Tree.....	15
2.3 Load Balancing – Εξισορρόπηση Φορτίου	16
2.3.1 Τί είναι η εξισορρόπηση φορτίου;	16
2.4 Ταξινόμηση αλγορίθμων εξισορρόπησης φορτίου στα SDN	27
2.4.1 Κατάταξη με βάση το επίπεδο εφαρμογής.....	27
2.4.2 Κατάταξη με βάση τη συχνότητα αλλαγής αποφάσεων για εξισορρόπηση φορτίου	29
2.4.3 Reactive vs Proactive Load Balancing.....	30
2.5 Μετρήσεις απόδοσης για τις τεχνικές εξισορρόπησης φορτίου.	30
2.6 Σχετικές εργασίες.....	32
3 Αλγόριθμοι προληπτικής (proactive) εξισορρόπησης φορτίου	34
3.1 Ο αλγόριθμος ECMP	34
3.1.1 Προβλήματα του ECMP	35
3.2 Ο αλγόριθμος Hedera	36
3.3 Ο αλγόριθμος PureSDN	37

3.4	Ο αλγόριθμος Conga.....	37
3.5	Σύνοψη	38
4	Πειραματική μελέτη.....	39
4.1	Το Mininet.....	39
4.2	Ο ελεγκτής Ryu	39
4.3	Το πλαίσιο αξιολόγησης των αλγορίθμων εξισορρόπησης φορτίου	41
4.3.1	Εργαλεία λήψης μετρήσεων (Iperf -ping)	41
4.3.2	Τοπολογίες και μοτίβα κυκλοφορίας	41
4.3.3	Σενάρια, Μετρικές, Παράμετροι	42
4.4	Αποτελέσματα	42
4.4.1	Ρυθμοαπόδοση (Throughput).....	42
4.4.2	First packet RTT	45
4.4.3	Μέση διακύμανση RTT.....	47
4.4.4	Average packet round-trip delay.....	48
4.4.5	Packet Loss – απώλεια πακέτων	49
4.5	Σύνοψη	50
5	Συμπεράσματα.....	52
6	Βιβλιογραφία	54

1 Εισαγωγή

Τα τελευταία χρόνια οι απαιτήσεις των δικτύων αλλάζουν δραματικά και όγκος των δεδομένων που μεταδίδεται αυξάνεται εκθετικά με αποτέλεσμα να ασκείται πίεση στα υπάρχοντα δίκτυα ως προς τη ζήτηση, τις απαιτήσεις ποιότητας υπηρεσιών και τη δυναμική διαχείριση. Στα παραδοσιακά δίκτυα οι αλγόριθμοι δρομολόγησης προωθούν την κίνηση με βάση το λιγότερο κόστος διαδρομής. Αυτός ο τρόπος δρομολόγησης της κίνησης μπορεί να οδηγήσει σε υπερφόρτωση κάποιων συνδέσμων μέσα στο δίκτυο.

Τα κέντρα δεδομένων είναι μία κρίσιμη υποδομή για τις νέες τεχνολογίες. Το κέντρο δεδομένων είναι ένας χώρος όπου στεγάζονται υπολογιστές και αποθηκευτικοί κόμβοι, τα οποία συνδέονται και δημιουργούν το δίκτυο κέντρων δεδομένων. Το δίκτυο κέντρων δεδομένων λειτουργεί σαν ένα κέντρο επικοινωνίας και έχει καθοριστικό ρόλο στην βελτιστοποίηση των λειτουργιών του.

Προκειμένου να ικανοποιηθεί η ανάγκη για ευελιξία και προσαρμογή στις νέες αυξημένες απαιτήσεις αναπτύχθηκαν νέα δικτυακά παραδείγματα όπως το Software-Defined Networking (SDN). Το SDN υπόσχεται την ενεργοποίηση νέων λύσεων που δυναμικά και προγραμματιζόμενα επιτρέπει τη δημιουργία αποτελεσματικών διαμορφώσεων δικτύου. Αυτή η νέα φιλοσοφία δικτύωσης, αποτέλεσε το κίνητρο για αυτή την εργασία, καθώς θα μπορούσε να συμβάλει στην εξεύρεση λύσεων για την εξισορρόπηση φορτίου σε ένα δίκτυο. Η εξισορρόπηση φορτίου είναι η διαδικασία ανάθεσης πόρων μέσα σε ένα δίκτυο με τέτοιο τρόπο όπου να έχει το δίκτυο την μέγιστη απόδοση του.

1.1 Κίνητρο

Το πλήθος των αυξανόμενων υπηρεσιών του εικοστού πρώτου αιώνα και οι σύγχρονες απαιτήσεις του χρήστη έχουν αυξήσει τις απαιτήσεις των δικτύων αλλάζουν συνεχώς και την αύξηση της ποσότητας κυκλοφορίας. Ωστόσο, ο τρόπος αρχιτεκτονικής δικτύων έχει παραμείνει στάσιμος και αυτό έχει ως αποτέλεσμα να αυξάνεται η πολυπλοκότητα και πολλαπλές ρυθμίσεις. Ο νέος τρόπος δικτύωσης, το SDN, μπορεί να προσαρμοστεί στα νέα δεδομένα και να συμβάλει στη διεύρυνση των αλγορίθμων εξισορρόπησης φορτίου και τη δημιουργία περισσότερο αποτελεσματικών λύσεων.

Η νέα αρχιτεκτονική δικτύων SDN θα βοηθήσει να λυθούν τα προβλήματα που αντιμετωπίζουν οι παραδοσιακές αρχιτεκτονικές δικτύων. Το SDN διαχωρίζει τα επίπεδα δεδομένων και ελέγχου και ενοποιεί τον έλεγχο. Το κεντρικό επίπεδο ελέγχου, έχει συνολική εικόνα του δικτύου και γίνεται πιο εύκολη και αποτελεσματική η διαχείριση του.

Σε αυτή την εργασία, θα μελετηθούν θέματα που σχετίζονται με την εξισορρόπηση φορτίου σε κέντρα δεδομένων και τοπολογίες FatTree. Μέχρι σήμερα έχουν προταθεί σημαντικές μέθοδοι για την εξισορρόπηση φορτίου που περιλαμβάνουν τον equal-cost multipath routing (ECMP), Hedera, PureSDN, Conga. Στόχος μας είναι να εξετάσουμε αυτές τις λύσεις συγκριτικά ως προς την αποτελεσματικότητα, απόδοση, τα πλεονεκτήματα και μειονεκτήματά τους. Η σύγκριση θα βασιστεί τόσο σε ποιοτικά κριτήρια όσο και σε πειραματική αξιολόγηση.

1.2 Ορισμός προβλήματος

Στα παραδοσιακά δίκτυα, λόγω έλλειψης καθολικής γνώσης για το δίκτυο (τοπολογία, φόρτος, διαθεσιμότητα πόρων), οι αποφάσεις δρομολόγησης λαμβάνονται ανά βήμα (hop) και βασίζονται συνήθως στους αλγορίθμους δρομολόγησης συντομότερης διαδρομής. Αυτό έχει ως αποτέλεσμα το φορτίο να κατανέμεται ανομοιόμορφα στους πόρους του δικτύου και μπορεί να προκαλέσει υπερφόρτωση ή συμφόρηση σε διάφορα μέρη του δικτύου.

Για την αποδοτική, ορθή και απρόσκοπτη λειτουργία των κέντρων δικτύων είναι απαραίτητη η εφαρμογή μηχανισμών διαχείρισης της κυκλοφορίας δεδομένων.

Μια προφανής λύση στο πρόβλημα αυτό είναι η υιοθέτηση της τεχνολογίας SDN, που επιτρέπει την απόκτηση καθολικής γνώσης για το δίκτυο και η εφαρμογή μηχανισμών εξισορρόπησης φορτίου. Στα πλαίσια αυτής της εργασίας, εξετάζουμε τους αλγορίθμους: ECMP, Hedera, PureSDN.

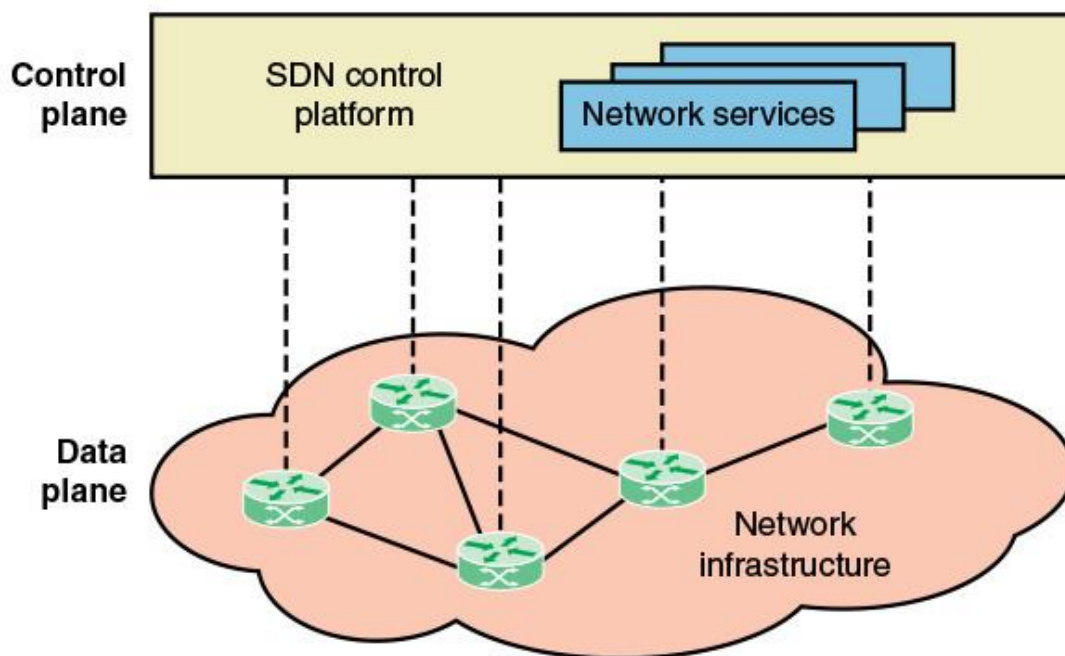
1.3 Δομή εργασίας

Η εργασία αποτελείται από πέντε επιμέρους κεφάλαια. Στο δεύτερο κεφάλαιο γίνεται εισαγωγή στη θεωρία για τα δίκτυα SDN και στα πλεονεκτήματα της χρήσης των δικτύων αυτών. Σε αυτό το κεφάλαιο γίνεται αναφορά στα κέντρα δεδομένων και στην εξισορρόπηση φορτίου. Επίσης, εξηγείται τι είναι η εξισορρόπηση φορτίου και γίνεται μία ταξινόμηση των αλγορίθμων εξισορρόπησης και παρουσιάζονται οι μετρικές απόδοσης. Στο τρίτο κεφάλαιο αναφέρονται οι αλγόριθμοι προληπτικής εξισορρόπησης φορτίου: ECMP, Hedera, PureSDN, Conga. Στο τέταρτο κεφάλαιο παρουσιάζεται η πειραματική μελέτη και τα εργαλεία που χρησιμοποιήθηκαν, η μεθοδολογία, το περιβάλλον αξιολόγησης, τα σενάρια και τα αποτελέσματα. Στο πέμπτο κεφάλαιο παρουσιάζονται τα συμπεράσματα και τα σχόλια από τα πειράματα.

2 Θεωρητικό υπόβαθρο

2.1 SDN – Software Defined networks

Τα δίκτυα SDN είναι μία προσέγγιση δικτύωσης η οποία διαχωρίζει το επίπεδο ελέγχου από το επίπεδο δεδομένων. Για να το πετύχει αυτό, χρησιμοποιεί ελεγκτές, δηλαδή κατάλληλο λογισμικό που χρησιμοποιεί διεπαφές προγραμματισμού εφαρμογών – API, για την επικοινωνία με το φυσικό επίπεδο δικτύου – hardware ή με το επίπεδο των εφαρμογών του δικτύου. Ο ελεγκτής είναι λογισμικό που διαχειρίζεται το φυσικό επίπεδο του δικτύου. Με βάση αυτήν την προσέγγιση ένας ελεγκτής έχει τη δυνατότητα να διαχειρίζεται – εκτελεί όλες τις πολύπλοκες λειτουργίες του δικτύου (δρομολόγηση, πολιτικές, έλεγχοι ασφαλείας, κ.ά.) και είναι σχεδιασμένος έτσι ώστε να ανταποκρίνεται καλύτερα στις εξελισσόμενες απαιτήσεις του. Το φυσικό επίπεδο του δικτύου περιλαμβάνει: routers, switches, access points, client devices, firewalls, κ.ά. και υλοποιεί την μεταφορά/προώθηση των δεδομένων.



Εικόνα 2-1: Software-defined Networks

Η κεντρική ιδέα πίσω από αυτήν την προσέγγιση του SDN είναι να διαχωριστούν οι λειτουργίες στα switch, σε επίπεδο δεδομένων και επίπεδο ελέγχου. Το επίπεδο δεδομένων απλά προωθεί τα πακέτα και το επίπεδο ελέγχου είναι υπεύθυνο για να λαμβάνει αποφάσεις, για να σχεδιάζει τη δρομολόγηση, να καθορίζει προτεραιότητες και πολιτικές παραμέτρων δρομολόγησης για να ανταποκρίνεται στις απαιτήσεις QoS. Επίσης είναι υπεύθυνο για την ανακατεύθυνση της κυκλοφορίας σε περίπτωση βλάβης καθώς επίσης και την ανάκτηση λειτουργίας σε περίπτωση σφάλματος.

2.1.1 Η σύνδεση του SDN με το NFV

Το Network Function Virtualization ή NFV ή αλλιώς “εικονικοποίηση λειτουργιών δικτύου” είναι μία ιδέα η οποία συμπληρώνει το SDN. Το NFV διαχωρίζει το λογισμικό από το υλικό για να υπάρχει πιο ευέλικτη ανάπτυξη δικτύου και δυναμική λειτουργία. Το SDN ελέγχει,

κατευθύνει, διαμορφώνει, τροποποιεί αυτές τις λειτουργίες για τις συγκεκριμένες χρήσεις, επίσης αφαιρεί τις φυσικές συσκευές δικτύωσης όπως routers, switches και τοποθετεί την επιλογή απόφασης στον ελεγκτή. Η ευελιξία NFV παρέχεται για τις εφαρμογές δικτύου χρησιμοποιώντας διακομιστές, έτσι ώστε το δίκτυο να μπορεί να αναπτυχθεί χωρίς την ανάγκη για περισσότερες συσκευές.

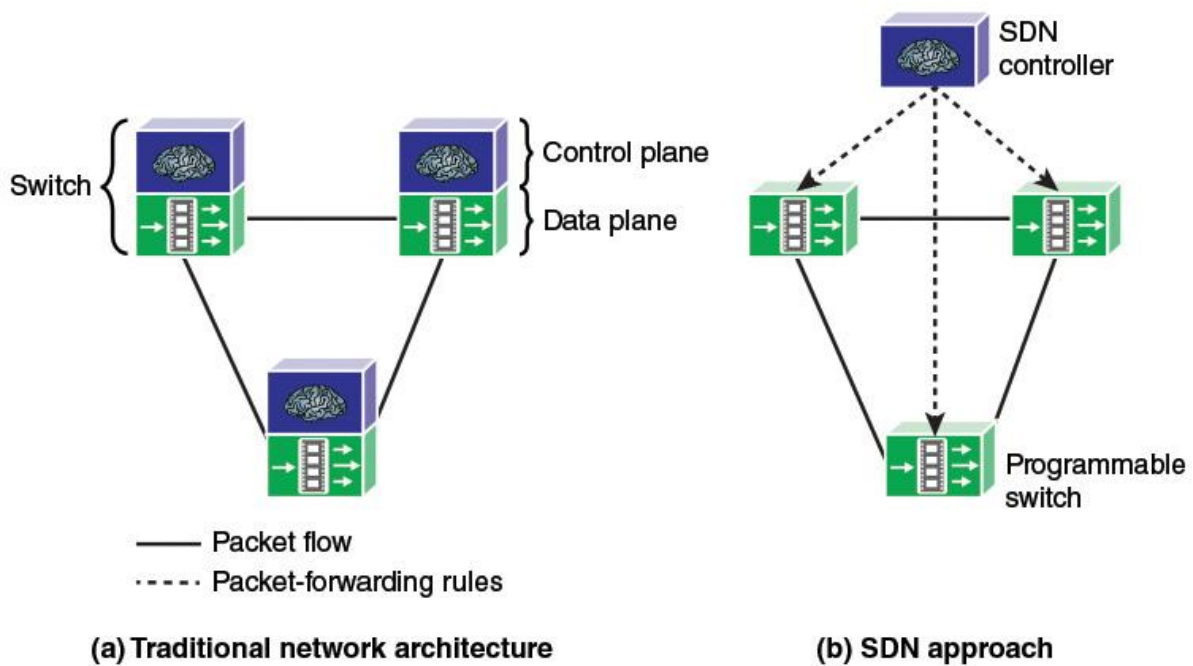
2.1.2 Πλεονεκτήματα της χρήσης των δικτύων SDN

Τα πλεονεκτήματα της χρήσης των δικτύων SDN είναι:

- Κεντρική διαχείριση: ευκολότερη διαχείριση των πόρων του δικτύου.
- Καλύτερη ορατότητα του δικτύου: ελαχιστοποιεί τα τυφλά σημεία του δικτύου.
- Επεκτασιμότητα: γίνεται πιο εύκολο να κλιμακωθούν τα δίκτυα με βάση τη ζήτηση.
- Ευελιξία: με την χρήση κώδικα μπορούν τα δίκτυα να ανταποκρίνονται με βάση της εκάστοτε ανάγκης της στιγμής.
- Ανοιχτού κώδικα: είναι εύκολο για τον κάθε προγραμματιστή να δημιουργήσει τον δικό του ελεγκτή που θα κάνει μία συγκεκριμένη εργασία για το δίκτυο του.
- Αυτοματισμός εργασιών: με την χρήση της κεντρικής διαχείρισης γίνεται πιο εύκολα, γρήγορα και ταυτόχρονα η διαχείριση του δικτύου και μειώνεται το ανθρώπινο λάθος.
- Ασφάλεια: πιο γρήγορες απαντήσεις σε απειλές.
- Μείωση του κόστους, οι διαδικασίες γίνονται κεντριοποιημένες και αυτόματες και μειώνονται τα έξοδα για ακριβό υλικό.

Τα μειονεκτήματα της χρήση των δικτύων SDN είναι:

- Περίπλοκη εγκατάσταση, κατά την αρχική εγκατάσταση και τη μετάβαση μπορεί να είναι περίπλοκη γιατί απαιτούνται σημαντικές αλλαγές.
- Κίνδυνοι και ρίσκα ασφάλειας, η ασφάλεια είναι κεντριοποιημένη οπότε υπάρχει ο όρος single point of failure.
- Πολλαπλές δεξιότητες, απαιτούνται δεξιότητες δικτύων και προγραμματισμού.



Εικόνα 2-2: Διαφορετικές προσεγγίσεις δικτύωσης α) παραδοσιακό δίκτυο, β) σύγχρονο δίκτυο

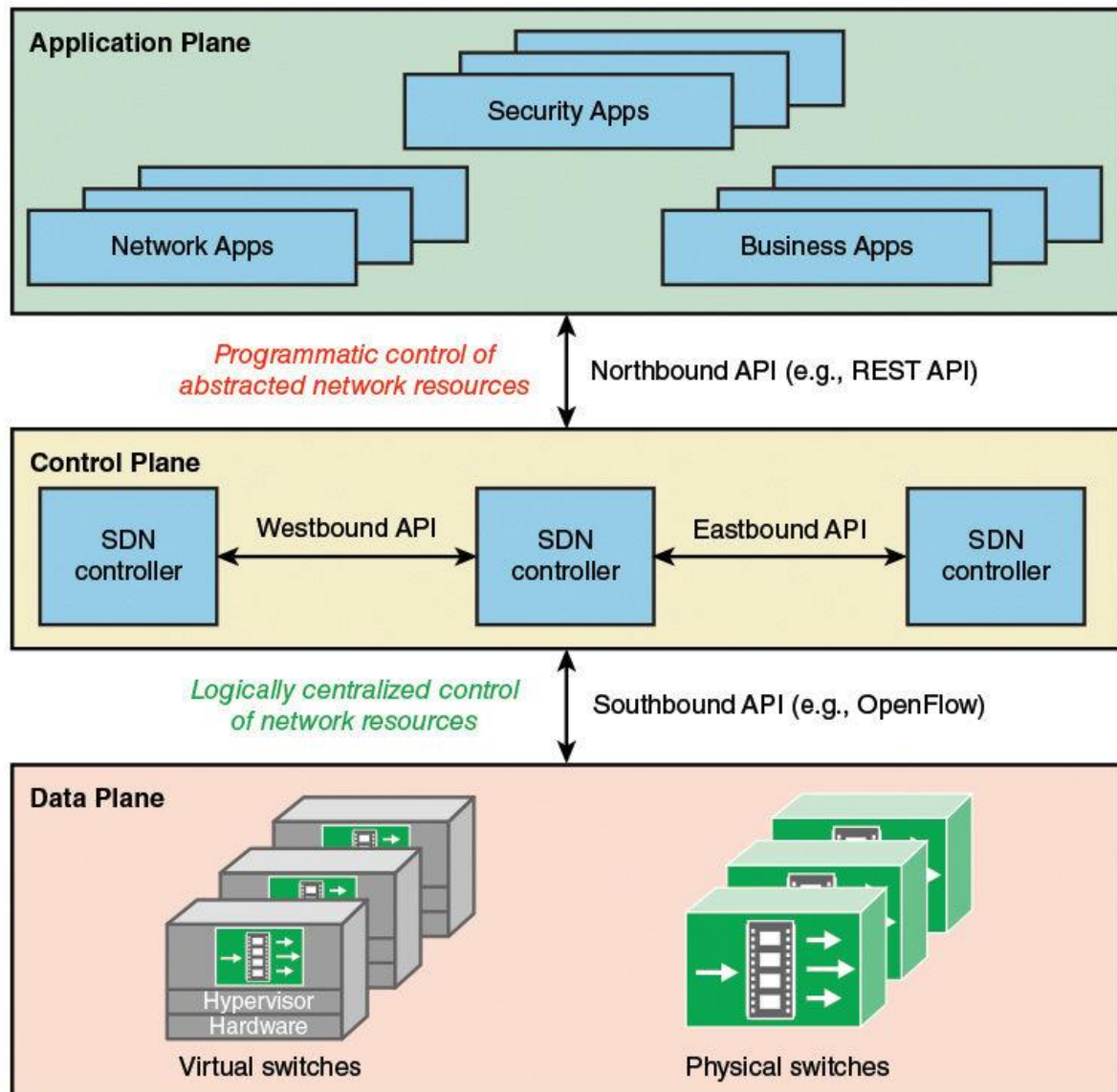
Οι βασικές απαιτήσεις για τις οποίες σχεδιάστηκαν τα δίκτυα καθοριζόμενα από λογισμικό είναι οι ακόλουθες:

- Προσαρμοστικότητα: τα δίκτυα θα πρέπει να προσαρμόζονται και να ανταποκρίνονται δυναμικά, βασιζόμενα στις ανάγκες των εφαρμογών.
- Αυτοματοποίηση: οι αλλαγές πολιτικής πρέπει να γίνονται αυτόματα για να μειώνονται τα λάθη και η συνεχόμενη εργασία (manual work).
- Συντήρηση: εισαγωγή νέων χαρακτηριστικών και δυνατοτήτων του hardware με ελάχιστη διακοπή λειτουργίας και με λιγότερη εργασία από τον διαχειριστή δικτύου.
- Κινητικότητα (mobility): ο έλεγχος λειτουργικότητας πρέπει να διευκολύνει την κινητικότητα συμπεριλαμβανομένων των φορητών συσκευών χρήση και εικονικών διακομιστών (servers).
- Security: οι εφαρμογές δικτύου πρέπει να ενσωματώνουν την ασφάλεια ως κύρια υπηρεσία και όχι ως πρόσθετη λύση.
- On-demand επεκτασιμότητα (scaling): το δίκτυο θα πρέπει να έχει την δυνατότητα να αυξάνεται ή να μειώνεται (scale – up or down) με βάση τις απαιτήσεις της υπηρεσίας.

2.1.1 Η αρχιτεκτονική των δικτύων SDN

Η αρχιτεκτονική των δικτύων SDN (Εικόνα 2-3) αποτελείται από τρία στοιχεία:

- Επίπεδο Εφαρμογών – application plane
- Επίπεδο ελέγχου – control plane
- Επίπεδο Δεδομένων – data plane

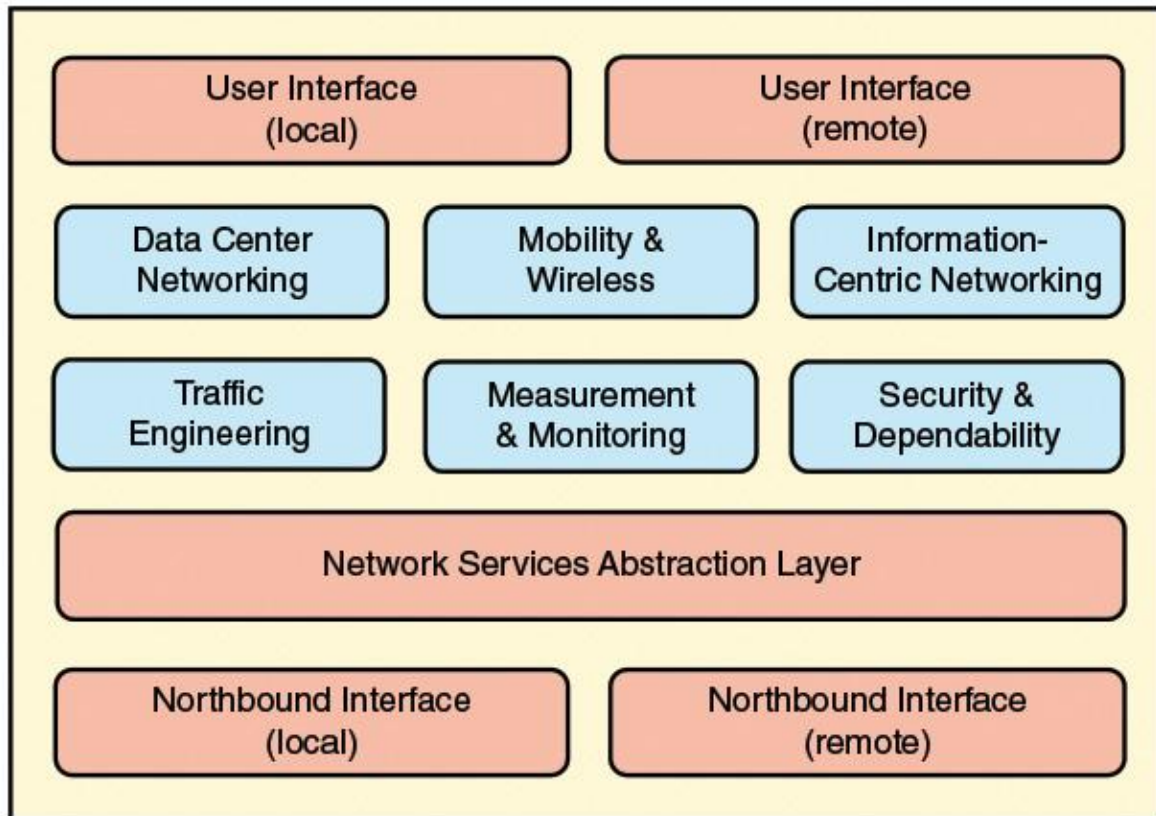


Εικόνα 2-3: Η αρχιτεκτονική SDN

Application Plane – Επίπεδο εφαρμογών

Το επίπεδο εφαρμογών βρίσκονται οι δικτυακές εφαρμογές όπως: δρομολόγηση, εξισορρόπηση φορτίου, QoS, προώθηση πακέτων, και υπηρεσίες που ορίζουν, παρακολουθούν και ελέγχουν τους πόρους και την συμπεριφορά του δικτύου. Αυτές οι εφαρμογές αλληλοεπιδρούν με το επίπεδο ελέγχου μέσω των APIs, ώστε το επίπεδο ελέγχου SDN να προσαρμόζει αυτόματα στην συμπεριφορά και τις ιδιότητες των πόρων δικτύου. Το “northbound interface” – βόρεια διεπαφή, που συνδέει το επίπεδο ελέγχου με το επίπεδο εφαρμογών, επιτρέπει στις εφαρμογές να έχουν πρόσβαση σε λειτουργίες και υπηρεσίες στο επίπεδο δεδομένων χωρίς να χρειάζεται να γνωρίζουν τις λεπτομέρειες των συσκευών. Προσφέρει μία αφηρημένη οπτική της υποδομής των συσκευών διαδικτύου. Η διεπαφή παρέχεται είτε από τον ελεγκτή μέσω ενός API είτε μέσω μίας γλώσσας προγραμματισμού των SDN δικτύων. Οι περισσότεροι ελεγκτές διαθέτουν ένα REST API με το οποίο παρέχουν

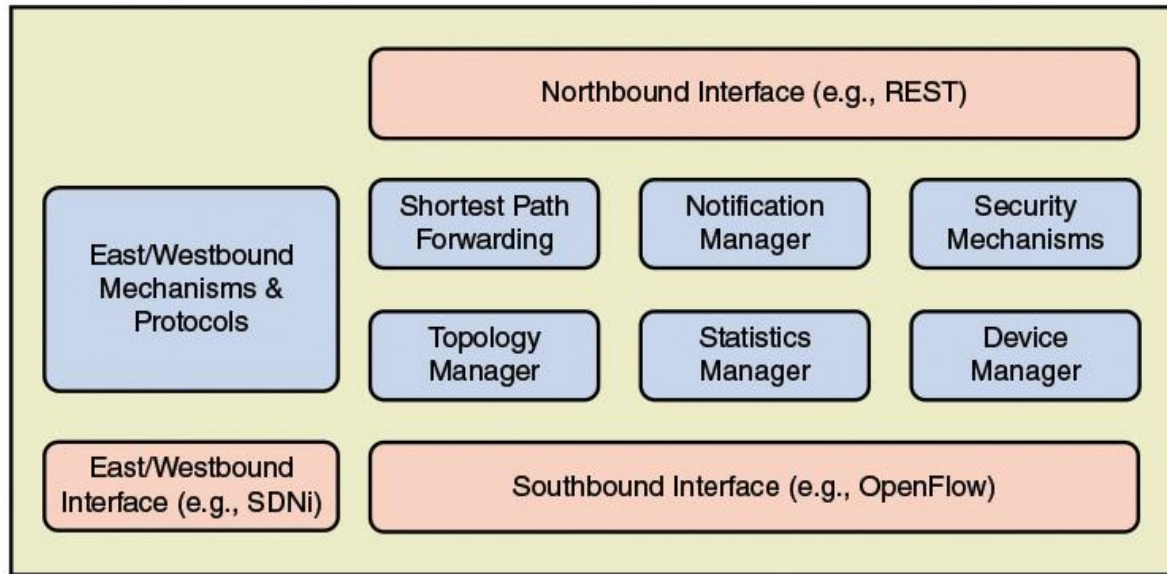
ένα προγραμματιστικό περιβάλλον για την καλύτερη διαχείριση και την αλληλεπίδραση μαζί τους.



Εικόνα 2-4: Επίπεδο εφαρμογών

Control Plane – επίπεδο ελέγχου

Η κύρια λειτουργία του επιπέδου ελέγχου SDN είναι να αντιστοιχίζει τα αιτήματα των υπηρεσιών από το επίπεδο εφαρμογής, σε συγκεκριμένες διαδικασίες αφού λάβει αποφάσεις διαβιβάζει εντολές στο επίπεδο υποδομής. Παρέχει επίσης πληροφορίες στις εφαρμογές που είναι στο application layer σχετικά με την τοποθεσία και τη δραστηριότητα του επιπέδου υποδομής. Ο ελεγκτής SDN “controller” είναι ο ελεγκτής του δικτύου.



Εικόνα 2-5: Επίπεδο ελέγχου

Κάποιες από τις κύριες λειτουργίες του επιπέδου ελέγχου είναι οι ακόλουθες:

- Προώθηση συντομότερης διαδρομής: χρησιμοποιεί πληροφορίες δρομολόγησης που τις συλλέγει από τα switches, για να καθορίσει προτεινόμενες διαδρομές
- Διαχείριση ειδοποιήσεων: λαμβάνει επεξεργάζεται και προωθεί συμβάντα “events” από διάφορες εφαρμογές σε μία άλλη.
- Μηχανισμός ασφαλείας: παρέχει απομόνωση και ασφάλεια ανάμεσα σε εφαρμογές και υπηρεσίες.
- Διαχείριση τοπολογίας: κατασκευάζει και διατηρεί πληροφορίες τοπολογίας.
- Διαχείριση στατιστικών: συλλέγει δεδομένα κίνησης μέσω των switch.
- Διαχειριστής συσκευών: διαμορφώνει τις παραμέτρους και τα χαρακτηριστικά από τα switch και διαχειρίζεται τους πίνακες ροής “flow tables”.

Σήμερα υπάρχει ένας μεγάλος αριθμός από ελεγκτές SDN που έχουν αναπτυχθεί από διάφορες ομάδες προγραμματιστών ή εταιριών για εμπορικούς ή εταιρικούς σκοπούς. Οι πιο κοινώς διαδεδομένοι είναι:

- OpenDaylight
- Open Network Operating System – ONOS
- POX
- Beacon
- Floodlight
- Ryu
- Onix

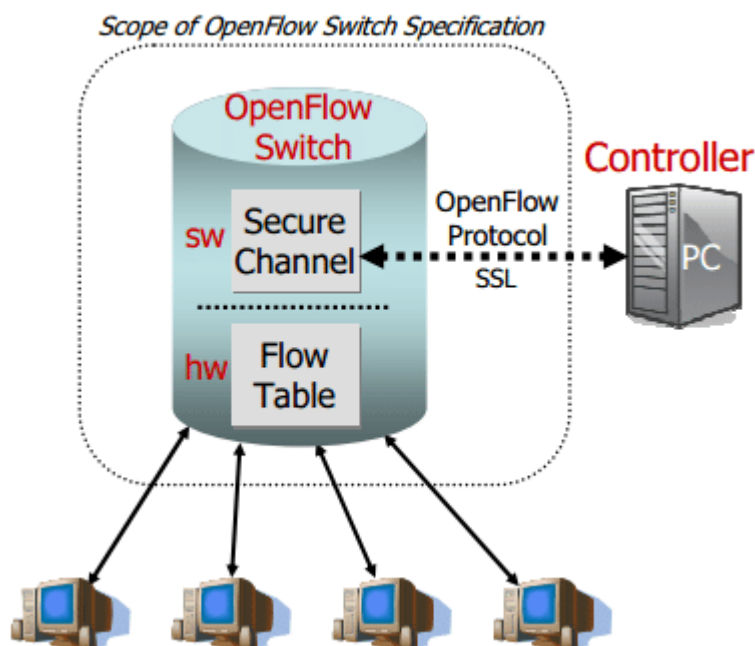
Southbound Interface – Νότια διεπαφή

Η νότια διεπαφή παρέχει την λογική σύνδεση, ενώνει μεταξύ του SDN controller και των switches που βρίσκονται στο επίπεδο υποδομής “infrastructure”. Το πιο κοινά εφαρμοζόμενο southbound API είναι το OpenFlow αλλά δεν είναι το μόνο. Υπάρχει ακόμα και το Netconf, Open vSwitch Database Management Protocol (OVSDB), Forwarding and Control Element Separation (ForCES), Protocol Oblivious Forwarding (POF)

Το πρωτόκολλο OpenFlow

Το OpenFlow προτάθηκε αρχικά από το Stanford University και είναι τώρα ένα στάνταρ πρωτόκολλο επικοινωνίας. Το OpenFlow πρωτόκολλο είναι το πιο συχνά χρησιμοποιούμενο πρωτόκολλο για την νότια διεπαφή της αρχιτεκτονικής SDN, η οποία διαχωρίζει το επίπεδο δεδομένων με το επίπεδο ελέγχου και επιτρέπει την επικοινωνία μεταξύ τους. Σκοπός του είναι η παροχή άμεσης πρόσβασης στο υλικό με αποτέλεσμα την γρήγορη παραμετροποίηση του.

Αρχιτεκτονική του OpenFlow



Εικόνα 2-6: Αρχιτεκτονική του OpenFlow

Η αρχιτεκτονική του OpenFlow όπως φαίνεται και στην Εικόνα 2-6 συμπεριλαμβάνει έναν ελεγκτή – controller, το OpenFlow switch και ένα ασφαλές κανάλι επικοινωνίας. Ο ελεγκτής ελέγχει το δίκτυο για να εφαρμόσει τις λειτουργίες του επιπέδου ελέγχου. Το switch είναι υπεύθυνο να προωθεί τα πακέτα στο επίπεδο δεδομένων και ανταλλάζει μηνύματα με τον ελεγκτή μέσω του ασφαλές καναλιού για να λαμβάνει καταχωρήσεις και να αναφέρει την κατάσταση του. Ο ελεγκτής είναι το «μυαλό» της αρχιτεκτονικής SDN και είναι

τοποθετημένος στο επίπεδο ελέγχου για να καθοδηγεί την προώθηση δεδομένων μέσω του πρωτοκόλλου OpenFlow. Το ασφαλές κανάλι OpenFlow είναι κρυπτογραφημένο με την χρήση της TLS (Transport Layer Security). Τα μηνύματα που μεταδίδονται μέσω του καναλιού OpenFlow είναι:

- Controller-to-Switch message: στέλνεται από τον ελεγκτή στο OpenFlow switch για να λάβει την κατάσταση του switch.
- Asynchronous message: στέλνεται από το OpenFlow switch στον ελεγκτή για ενημέρωση συμβάντων στο δίκτυο ή αλλαγών κατάστασης.
- Symmetric message: χρησιμοποιείται για την δημιουργία μίας σύνδεσης και για το αν μία συσκευή είναι συνδεδεμένη.

Το βασικό στοιχείο του OpenFlow είναι το switch το οποίο είναι υπεύθυνο για την προώθηση των πακέτων στο επίπεδο δεδομένων με βάση έναν πίνακα ροής ο οποίος περιέχει ένα σύνολο εγγράφων πολιτικής για τον τρόπο επεξεργασίας της κυκλοφορίας. Μπορεί να είναι φυσικό “physical” ή εικονικό “virtualized”. Τα OpenFlow switches ταξινομούνται στους ακόλουθους τύπους:

- Dedicated OpenFlow: υποστηρίζει μόνο την προώθηση OpenFlow και αυτό το switch επεξεργάζεται όλη την κίνηση που διέρχεται από αυτόν και δεν μπορεί να εκτελέσει προώθηση Layer 2 ή Layer 3.
- OpenFlow-compatible: υποστηρίζει την προώθηση OpenFlow και την προώθηση Layer 2/3

Data plane – Επίπεδο δεδομένων

Το επίπεδο δεδομένων ή αλλιώς επίπεδο υποδομής αποτελούν οι συσκευές δικτύου που εκτελούν τη μεταφορά και την επεξεργασία δεδομένων σύμφωνα με τις οδηγίες που λαμβάνουν από το επίπεδο ελέγχου (control plane) του SDN. Το πιο σημαντικό χαρακτηριστικό αυτών των συσκευών του δικτύου είναι ότι οι συσκευές αυτές εκτελούν μία απλή λειτουργία προώθησης χωρίς να εμπεριέχουν καθόλου λειτουργικό σύστημα το οποίο μπορεί να λάβει αυτόνομες αποφάσεις. Οι κύριες λειτουργίες είναι:

- Λειτουργία υποστήριξης: το switch επικοινωνεί με τον ελεγκτή και ο ελεγκτής διαχειρίζεται το switch μέσω του πρωτοκόλλου OpenFlow.
- Λειτουργία προώθησης δεδομένων: δέχεται δεδομένα από άλλες συσκευές δικτύου και τα προωθεί σε άλλες συσκευές σύμφωνα με τους κανόνες που έχουν οριστεί από τις εφαρμογές SDN.

2.1.2 Εφαρμογές της τεχνολογίας SDN

Η τεχνολογία SDN προσφέρει στους διαχειριστές συστημάτων μεγάλη προσαρμοστικότητα κατά την διαχείριση δικτύων. Αυτή η προσαρμοστικότητα βοήθησε στην βελτίωση της αποτελεσματικότητας και σε ένα ευρύ φάσμα χρήσεων σε πολλούς τομείς.

- Network Virtualization (εικονικοποίηση δικτύου): Η πρώτη ευρέως αποδεκτή περίπτωση χρήσης για το SDN ήταν η εικονικοποίηση του δικτύου. Χρησιμοποιώντας τους ίδιους φυσικούς πόρους δικτύου για τη δημιουργία ξεχωριστών εικονικών δικτύων. Η βασική ιδέα ήταν ότι το σύγχρονο “cloud” απαιτούσαν δίκτυα που θα μπορούσαν να δημιουργηθούν, να διαχειρίζονται και να καταρρίπτονται μέσω προγραμματισμού, χωρίς ο διαχειριστής δικτύου να χρειάζεται να ρυθμίσει με μη αυτόματο τρόπο, για παράδειγμα, ετικέτες VLAN σε κάποιο αριθμό switch του δικτύου. Διαχωρίζοντας το επίπεδο ελέγχου από το επίπεδο δεδομένων και συγκεντρώνοντας λογικά το επίπεδο ελέγχου, κατέστη δυνατή η έκθεση ενός μόνο σημείου εισόδου API για τη δημιουργία, την τροποποίηση και τη διαγραφή εικονικών δικτύων.
- Software-Defined WANs (SD-WAN): Μία άλλη περίπτωση χρήσης για το SDN είναι η χρήση του για τα δίκτυα ευρείας περιοχής. Οι επιχειρήσεις για πολλά χρόνια αγόραζαν υπηρεσίες WAN για έτσι ώστε να έχουν την δυνατότητα να μπορούν να συνδέσουν τις διάφορες τοποθεσίες μεταξύ τους. Η τεχνική προσέγγιση που χρησιμοποιείται ευρέως για την κατασκευή WAN που αγόραζαν οι εταιρίες ονομάζεται MPLS. Η τεχνική είναι γνωστή ως MPLS-BGP με τη χρήση VPN. Με τα SD-WAN μεταφέρεται η χρήση των VPN στον κεντρικό ελεγκτή. Η επιχείρηση συνδέει τις τοποθεσίες μεταξύ τους και εφαρμόζει τις πολιτικές ασφαλείας, την κυκλοφορία και την πρόσβαση σε κοινές υπηρεσίες.
- Data Center (κέντρα δεδομένων): Οι εφαρμογές απαιτούν τα δίκτυα να είναι ευέλικτα και να ανταποκρίνονται στις συνεχείς αλλαγές. Τα κέντρα δεδομένων είναι ο κόμβος για τους διακομιστές και τους υπολογιστικούς πόρους, το SDN ενισχύει σημαντικά την ικανότητα του δικτύου να ανταποκρίνεται στις δυναμικές απαιτήσεις του κέντρου δεδομένων. Επίσης παρέχει στους οργανισμούς μεγαλύτερο και λεπτομερή έλεγχο της σύγχρονης υποδομής που είναι επεκτάσιμη, ευέλικτη και ασφαλής.

2.1.3 Η γλώσσα προγραμματισμού P4

Η P4 γλώσσα προγραμματισμού, που βγαίνει από τα αρχικά της φράσης Programming Protocol-independent Packet Processors, χρησιμοποιείται για να καθορίσει πώς επεξεργάζεται την προώθηση πακέτων το επίπεδο δεδομένων, δηλαδή το switch. Η P4 είναι protocol independent και target independent, αυτό σημαίνει ότι ο κώδικας μπορεί να τρέξει σε διάφορες πλατφόρμες υλικού – hardware, απλά θα πρέπει να συμμορφώνονται με μία κοινή προδιαγραφή υλικού. Αρχικά προτάθηκε το 2014 και οι κύριες δυνατότητες της P4 είναι συνοψίζονται στα τρία βασικά πλεονεκτήματα:

- Δυνατότητα επαναδιαμόρφωσης – reconfigurability: Μπορούν οι προγραμματιστές να αλλάζουν τον τρόπο με τον οποίο τα πακέτα επεξεργάζονται.
- Protocol independence: Τα switches είναι ανεξάρτητα – ελεύθερα από ανεπιθύμητες ενσωματώσεις πρωτοκόλλων.

- Target independence: το λογισμικό μπορεί να τρέξει σε οποιοδήποτε υλικό χωρίς κάποιο πρόβλημα.

Η P4 χρησιμοποιείται συνήθως με το SDN, μπορεί επίσης να χρησιμοποιηθεί αυτόνομα – standalone mode, όπου και τα δύο επίπεδα δεδομένων και ελέγχου είναι στο ίδιο σύστημα. Η γλώσσα είναι πολύ ευέλικτη και μπορεί να προσαρμοστεί στα περισσότερα πρωτόκολλα δικτύου συμπεριλαμβανομένου και του OpenFlow. Ωστόσο, για τον έλεγχο προσαρμοσμένων πρωτοκόλλων που έχουν προγραμματιστεί στην P4 ένα νέο πρωτόκολλο επικοινωνίας χρειάζεται να κατασκευαστεί.

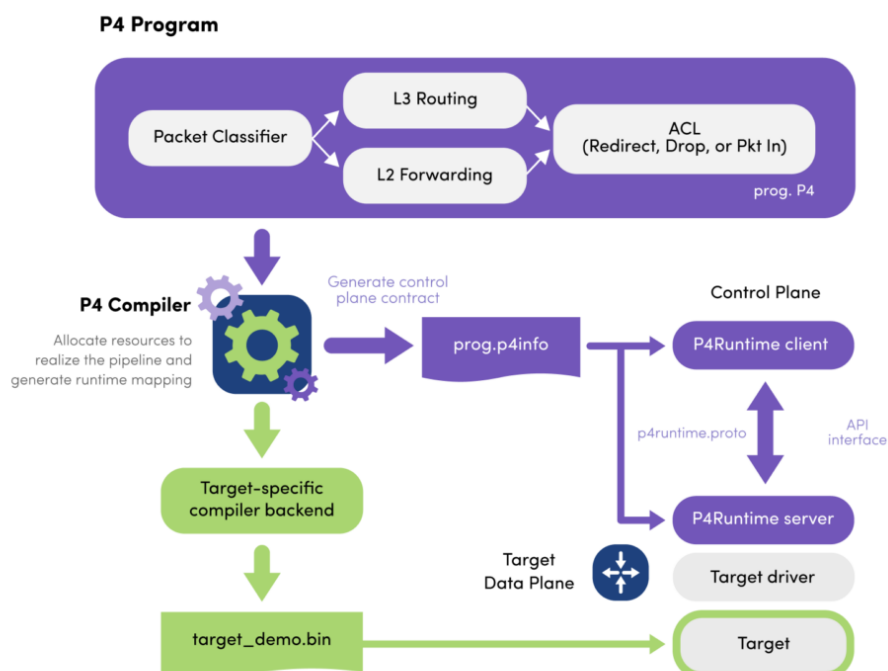
Η ροή εργασιών της P4.

Τα προγράμματα P4 και οι μεταγλωττιστές είναι target – specific. Ο στόχος μπορεί να είναι βασισμένος σε υλικό (FPGA, Programmable ASICs) ή λογισμικό (running on x86).

Ένα P4 πρόγραμμα ταξινομεί τα πακέτα με βάση τον header τους και με τις ενέργειες που πρέπει να γίνουν στα εισερχόμενα πακέτα (e.g., forward, drop).

Ο P4 μεταγλωττιστής δημιουργεί τα runtime mapping metadata για να επιτρέψει στα επίπεδα ελέγχου και δεδομένων να επικοινωνούν μεταξύ τους χρησιμοποιώντας το P4Runtime.

Ο μεταγλωττιστής P4 δημιουργεί επίσης ένα εκτελέσιμο αρχείο για το επίπεδο δεδομένων (target_prog.bin), προσδιορίζοντας τις μορφές του header και τις αντίστοιχες ενέργειες για την συσκευή προορισμού [21].



Εικόνα 2-7: Το P4 πρόγραμμα

2.2 Τα κέντρα δεδομένων – data center

Τα κέντρα δεδομένων είναι μία εγκατάσταση που στην οποία στεγάζονται υποδομές πληροφορικής και εξοπλισμός για την παροχή υπηρεσιών και εφαρμογών. Αυτές οι υποδομές συνεργάζονται με σκοπό για την επεξεργασία, την αποθήκευση, την διαχείριση και την διάδοση δεδομένων. Στην αρχή τα κέντρα δεδομένων ξεκίνησαν ως ιδιόκτητα. Η κάθε εταιρία είχε το δικό της χώρο εντός των εγκαταστάσεων της με την απαραίτητη υποδομή για την παροχή υπηρεσίας. Πρόσφατα, έχουν εξελιχθεί σε απομακρυσμένες εγκαταστάσεις ή δίκτυα τέτοιων εγκαταστάσεων που ανήκουν σε εταιρίες που παρέχουν τέτοιου είδους υπηρεσίες. Αυτές οι υπηρεσίες ονομάζονται υπηρεσίες cloud, είναι οι cloud service providers. Αυτά τα κέντρα δεδομένων CSP στεγάζουν εικονική υποδομή πληροφορικής για την κοινή χρήση πολλών εταιρειών και πελατών. Μία μελέτη από την McKinsey & Company προβλέπει ότι ο κλάδος θα αναπτυχθεί κατά 10% ετησίως. Ένας άλλος όρος που χρησιμοποιείται είναι ο point-of-presence (PoP), δηλαδή σημείο παρουσίας. Αυτό αναφέρεται σε μία εταιρία που έχει παρουσία ενός διακομιστή σε μία τοποθεσία που παρέχει σύνδεση στο διαδίκτυο, ενώ το κέντρο δεδομένων αναφέρεται σε μία τοποθεσία που φιλοξενεί πολλούς διακομιστές. Η κύρια χρησιμότητα του DC είναι να στεγάζει τον μεγαλύτερο αριθμό υπολογιστών και κόμβων αποθήκευσης, που διασυνδέονται από το ειδικά σχεδιασμένο δίκτυο, το δίκτυο κέντρων δεδομένων, το DCN – data center network. Αυτό το δίκτυο χρησιμεύει για την επικοινωνία και παίζει καθοριστικό ρόλο για την βελτιστοποίηση των λειτουργιών του δικτύου. Οι μοναδικές απαιτήσεις στο DCN θέτουν σημαντικές προκλήσεις στην υποδομή και της λειτουργίες του. Κάποιες από τις προκλήσεις είναι:

- Υψηλή επεκτασιμότητα: Είναι σύνηθες για ένα σύγχρονο κέντρο δεδομένων να περιέχει εκατοντάδες χιλιάδες διακομιστές ενώ η κλίμακα των σύγχρονων DC αυξάνεται ραγδαία με εκθετικό ρυθμό. Για παράδειγμα, η Microsoft φιλοξενεί πάνω από 1 εκατομμύριο διακομιστές σε περισσότερα από 100 κέντρα δεδομένων παγκοσμίως. Μια τέτοια τεράστια κλίμακα συστήματος αυξάνει περαιτέρω τις προκλήσεις στο σχεδιασμό του δικτύου όσον αφορά τη διασύνδεση, το κόστος και την ευρωστία.
- Μεγάλη ποικιλία εφαρμογών: Τα σύγχρονα κέντρα δεδομένων φιλοξενούν διάφορες υπηρεσίες και εφαρμογές, συμπεριλαμβανομένων όχι μόνο διαδικτυακών εφαρμογών όπως αναζήτησης στο Web, αλληλογραφίας στο Web και διαδραστικών παιχνιδιών, αλλά και υπηρεσιών υποδομής όπως κατανομημένα συστήματα αρχείων και κατανομημένες μηχανές εκτέλεσης. Οι διαφοροποιημένες υπηρεσίες και εφαρμογές στα κέντρα δεδομένων ορίζουν μια ποικιλία διαφορετικών χαρακτηριστικών κυκλοφορίας. Τα αποτελέσματα των μετρήσεων έχουν δείξει αντίθετα χαρακτηριστικά κυκλοφορίας όταν φιλοξενούνται διαφορετικές εφαρμογές.
- Υψηλή πυκνότητα ισχύος: Η ετήσια κατανάλωση ενέργειας του κέντρου δεδομένων στις ΗΠΑ εκτιμήθηκε ότι ήταν πάνω από 100 δισεκατομμύρια kWh το 2011, αντιπροσωπεύοντας ετήσιο κόστος ηλεκτρικής ενέργειας 7,4 δισεκατομμύρια δολάρια. Ενώ η κατανάλωση ενέργειας των συσκευών δικτύωσης αντιπροσωπεύει

μόνο περίπου το 10% της συνολικής κατανάλωσης ενέργειας ΤΠΕ, το ποσοστό κατανάλωσης ενέργειας από το δίκτυο μπορεί ακόμη και να αυξηθεί στο 50% στο μέλλον, όταν οι ίδιοι οι διακομιστές γίνουν πιο ενεργειακά ανάλογοι με τον φόρτο εργασίας τους.

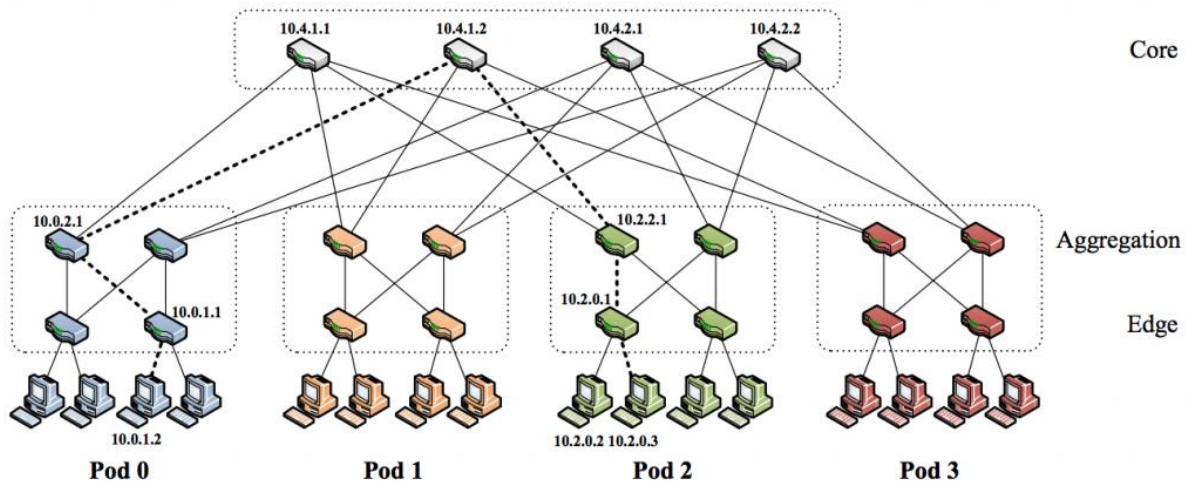
- Αυστηρές απαιτήσεις: Βασική προϋπόθεση για τη φιλοξενία υπηρεσιών σε κέντρα δεδομένων είναι η παροχή υπηρεσιών 24/7/365, κάτι που απαιτεί υψηλή ευρωστία του συστήματος. Ωστόσο, οι αστοχίες δικτύου από υλικό, λογισμικό και ανθρώπινα σφάλματα μπορεί να είναι αναπόφευκτες, απαιτείται συνεχής παρακολούθηση και ευέλικτη αποκατάσταση αστοχιών.

Αντιμέτωποι με αυτές τις σημαντικές τεχνικές προκλήσεις, τα DCN έχουν λάβει πρόσφατα σημαντική προσοχή τόσο από τον ακαδημαϊκό κόσμο όσο και από τη βιομηχανία. Υπάρχουν μερικές έρευνες για συγκεκριμένες πτυχές του DCN, π.χ., τοπολογία δικτύου, πράσινη δικτύωση, οπτική δικτύωση, ασύρματη δικτύωση και εικονικοποίηση δικτύου. Η πολυπλοκότητα του DCN απαιτεί μια συστηματική άποψη τόσο των αρχιτεκτονικών όσο και των λειτουργικών αρχών του. Η υποδομή του κέντρου δεδομένων μπορεί να κατηγοριοποιηθεί βάση δύο τομείς την τεχνολογία μετάδοσης και την σύνδεση. Οι τεχνολογίες μετάδοσης είναι οι ακόλουθες: καλώδια χαλκού, οπτικές ίνες – optical fiber (οπτική ίνα), ασύρματες επικοινωνίες – wireless (IEEE 802.11). Όσον αφορά τη σύνδεση είναι διαχωρισμένη σε δύο: intra data center networks και inter data center networks. Το intra data center αναφέρετε για την σύνδεση εσωτερικά στο data center ενώ το inter data center αναφέρετε για την σύνδεση μεταξύ δύο ή και περισσότερων data center. Στα συμβατικά δίκτυα οι αρχιτεκτονικές οι οποίες χρησιμοποιούνται είναι: two-tier, three-tier και fat-tree. Οι αρχιτεκτονικές three-tier και fat-tree χρησιμοποιούνται επίσης και για τα μεγάλης κλίμακας δίκτυα δεδομένων. Όσον αφορά τα inter data center οι τοπολογίες περιορίζονται σε μεγάλο βαθμό από την επιλογή της τοποθεσίας του κέντρου δεδομένων και από την συνδεσιμότητα δικτύου της τοποθεσίας. Η επιλογή των τοποθεσιών των κέντρων δεδομένων επηρεάζονται από πολλούς παράγοντες όπως: η γεωγραφία, ο ηλεκτρισμός, η συνδεσιμότητα.[28,29]

2.2.1 Η Τοπολογία Fat-Tree

Η τοπολογία Fat-Tree είναι μία ειδική έκδοση της τοπολογίας τύπου Clos. Η συγκεκριμένη τοπολογία χρησιμοποιεί μία αρχιτεκτονική 3 επιπέδων η οποία περιλαμβάνει τα core switches, aggregation switches και τα edge switches. Ένα δίκτυο k-ary fat-tree μπορεί να υποστηρίξει $k^3/4$ hosts χρησιμοποιώντας $5 \cdot k^2/4$ switches. Η βασική ιδέα είναι να υπάρχουν περισσότεροι σύνδεσμοι πιο κοντά στους διακομιστές έτσι ώστε να μειωθεί η συμφόρηση – bandwidth μέσα στο δίκτυο. Τα core switches είναι στην κορυφή του δικτύου, τα aggregation είναι στην μέση και συνδέονται με τα core και τα edge switches. Και τα edge switches βρίσκονται στο κάτω μέρος του δικτύου και συνδέονται με τα aggregation switches. Οι συνδέσεις μεταξύ των επιπέδων switch, core – aggregation και aggregation – edge είναι ίσου κόστους “equal – cost”. Έτσι δημιουργούνται πολλαπλές διαδρομές μεταξύ των διακομιστών με αποτέλεσμα να υπάρχει μεγαλύτερο συνολικό εύρος ζώνης και μεγαλύτερη

ανοχή σε τυχόν σφάλματα ή αστοχίες σε σχέση με ένα απλό δίκτυο. Το βασικό πλεονέκτημα αυτής της τοπολογίας είναι ότι παρέχει υψηλό εύρος ζώνης και μπορεί να έχει πολύ καλή εξισορρόπηση φορτίου. Επίσης είναι πολύ εύκολα επεκτάσιμη και υπάρχει ανοχή σε τυχόν σφάλματα λόγω των πολλαπλών διαδρομών που δημιουργούνται μεταξύ των διακομιστών.

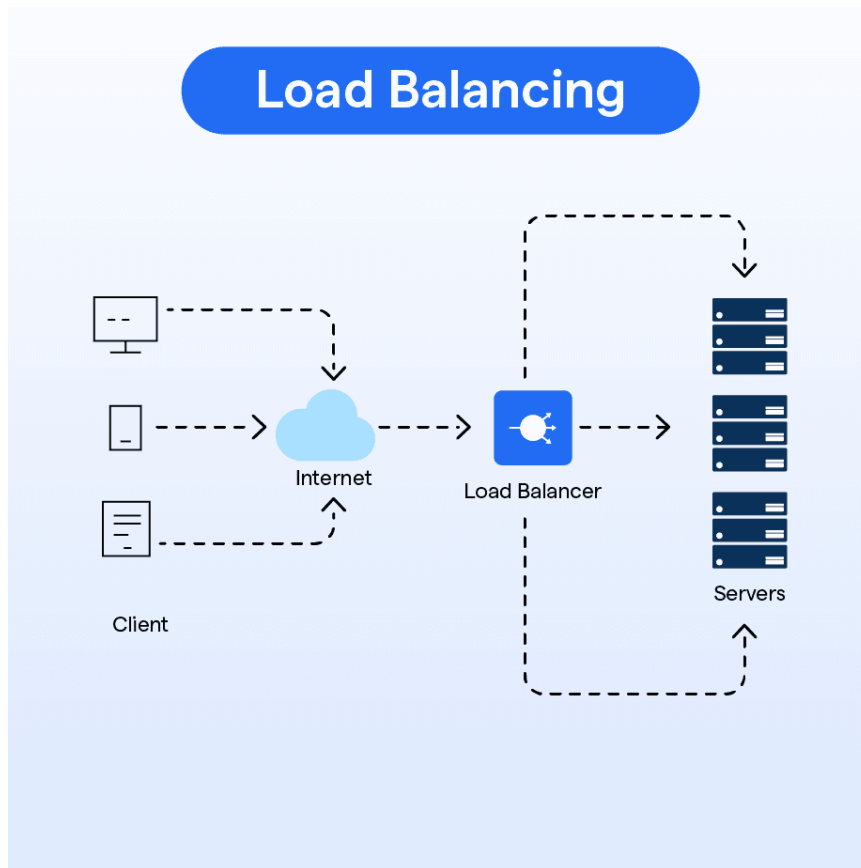


Εικόνα 2-8: τοπολογία Fat-Tree

2.3 Load Balancing – Εξισορρόπηση Φορτίου

2.3.1 Τί είναι η εξισορρόπηση φορτίου;

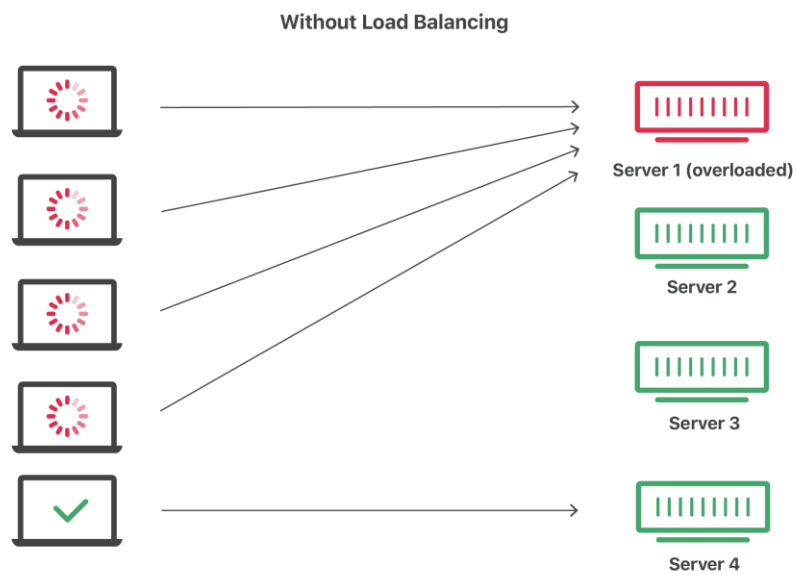
Η εξισορρόπηση φορτίου είναι η διαδικασία με την οποία διανέμεται η κυκλοφορία του δικτύου ανάμεσα στους διακομιστές με τέτοιο τρόπο έτσι ώστε να παρέχει το δίκτυο την βέλτιστη απόδοση και ο χρήστης να μην έχει προβλήματα κατά την χρήση των εφαρμογών που βρίσκονται στους διακομιστές του. Ο σκοπός της διαδικασίας είναι να γίνει το δίκτυο πιο αποδοτικό. Η συσκευή αυτή τοποθετείται ανάμεσα από τον πελάτη και τους διακομιστές. Ο κύριος λόγος ύπαρξης είναι να λαμβάνει τα εισερχόμενα αιτήματα των πελατών και να τα διανέμει στον οποιονδήποτε διακομιστή που είναι ικανός να τα εκπληρώσει πιο γρήγορα ανάλογα με το τρέχω φορτίο του.



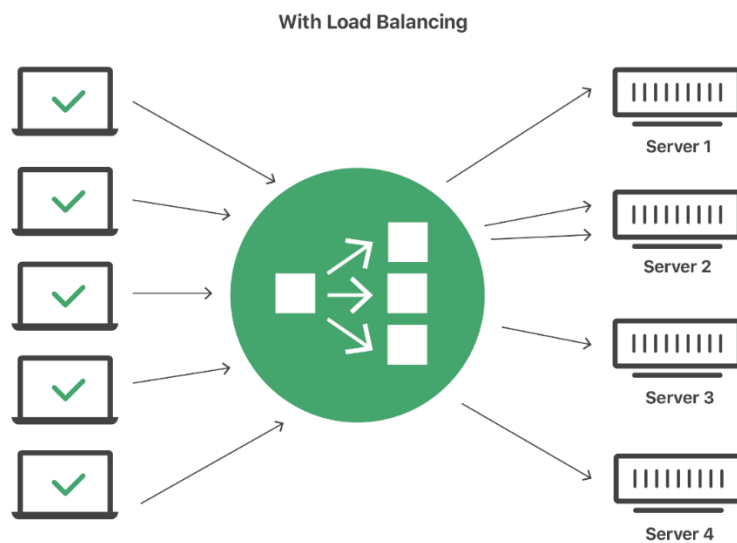
Εικόνα 2-9: Εξισορρόπηση Φορτίου

Γιατί χρειάζεται η εξισορρόπηση φορτίου;

Η εξισορρόπηση φορτίου χρειάζεται για την βελτιστοποίηση της απόδοσης και της αξιοπιστίας της υπηρεσίας στον τελικό χρήστη. Στην περίπτωση που υπάρχει ένας μεμονωμένος διακομιστής ο οποίος έχει την δυνατότητα να ανταποκριθεί σε ένα περιορισμένο αριθμό αιτημάτων, θα μπορούσε να προκαλέσει σφάλματα ή ακόμα και διακοπές λειτουργίας σε κάποια αιτήματα για να ανταποκριθεί σε κάποια άλλα. Η βασική χρησιμότητα του είναι η αποτελεσματική κατανομή της κυκλοφορίας σε πολλούς διακομιστές με τέτοιο τρόπο να μην υπάρχει υπερφόρτωση και να διασφαλιστεί η συνεχής εξυπηρέτηση στον τελικό χρήστη. Με αυτόν τον τρόπο ο χρήστης έχει καλύτερη εμπειρία γιατί έχει καλύτερους χρόνους απόκρισης μεταξύ αυτού και του διακομιστή. Βοηθάει επίσης στην διατήρηση του χρόνου λειτουργίας, σε περίπτωση που ένας διακομιστής τεθεί σε μη λειτουργικός οι άλλοι λαμβάνουν τα αιτήματα του.



Εικόνα 2-10: Αρχιτεκτονική χωρίς εξισορρόπηση φορτίου



Εικόνα 2-11: Αρχιτεκτονική με εξισορρόπηση φορτίου

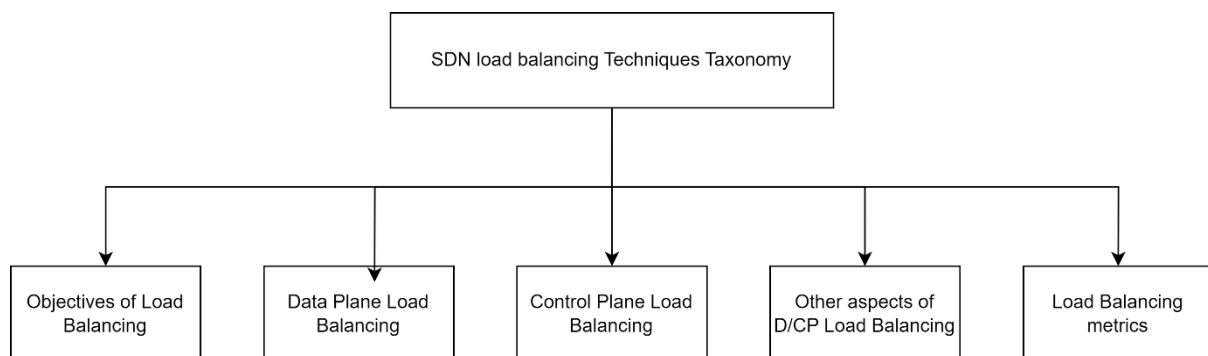
Πλεονεκτήματα της εξισορρόπησης φορτίου

- Επεκτασιμότητα – scalability
- Ευελιξία – flexibility
- Διαθεσιμότητα – availability
- Απόδοση – performance

Πώς δουλεύει η εξισορρόπηση φορτίου;

Εξισορρόπηση φορτίου γίνεται από μία φυσική συσκευή “physical appliance” ή από λογισμικό “software” σε έναν διακομιστή “server” ή ακόμα και σε έναν δρομολογητή. Στη φυσική συσκευή αυτή έχει εγκατασταθεί πάνω της το απαιτούμενο λογισμικό και εγκαθίστανται στις εγκαταστάσεις του πελάτη. Από πλευράς λογισμικού, είναι εφαρμογές “application” οι οποίες είναι εγκατεστημένες σε διακομιστές οι οποίοι είναι ιδιόκτητοι ή φιλοξενούνται στο cloud. Η χρησιμότητα της συσκευής αυτής είναι να λαμβάνει πρώτη τα αιτήματα των χρηστών να τα αξιολογεί και να καθορίζει ποιος διακομιστής θα πρέπει να διαχειριστεί το κάθε ένα βασισμένη σε κάποιους αλγόριθμους. Μόλις ο διακομιστής παραλάβει την αίτηση του πελάτη, στέλνει απάντηση στον πελάτη διαμέσου του εξισορροπιστή και ανοίγει μία συνεδρία μεταξύ πελάτη και διακομιστή, αυτό γίνεται μόνο στους στατικούς αλγορίθμους. Με αυτόν τον τρόπο ο εξισορροπιστής ενώνει την IP address του πελάτη με την IP address του διακομιστή και συνεχίζεται η αλληλεπίδραση πελάτη με διακομιστή μέχρι να εκτελεστούν όλες οι εργασίες και να τερματιστεί η συνεδρία.

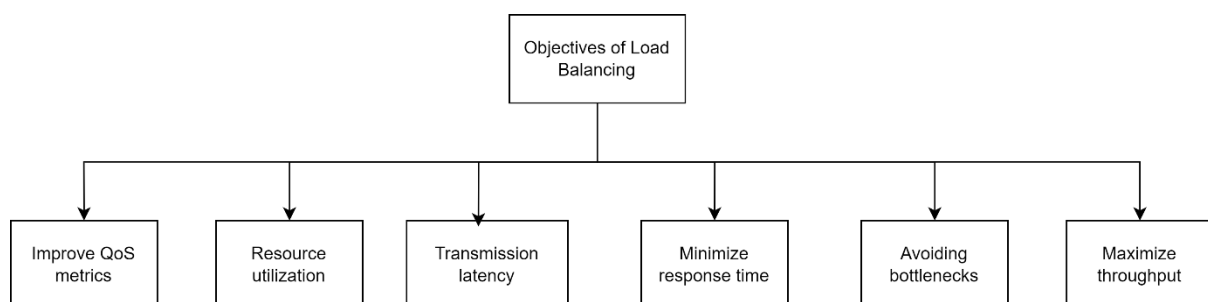
Η ταξινόμηση των τεχνικών load balancing για τα SDN



Εικόνα 2-10: Η ταξινόμηση των τεχνικών της εξισορρόπησης φορτίου

Στην παραπάνω εικόνα βλέπουμε την βασική ταξινόμηση των τεχνικών load balancing στα SDN. Οι τεχνικές έχουν χωριστεί σε: στόχους, επίπεδο δεδομένων, επίπεδο ελέγχου, άλλοι τομείς και μετρήσεις.

Οι στόχοι της εξισορρόπησης φορτίου στα SDN

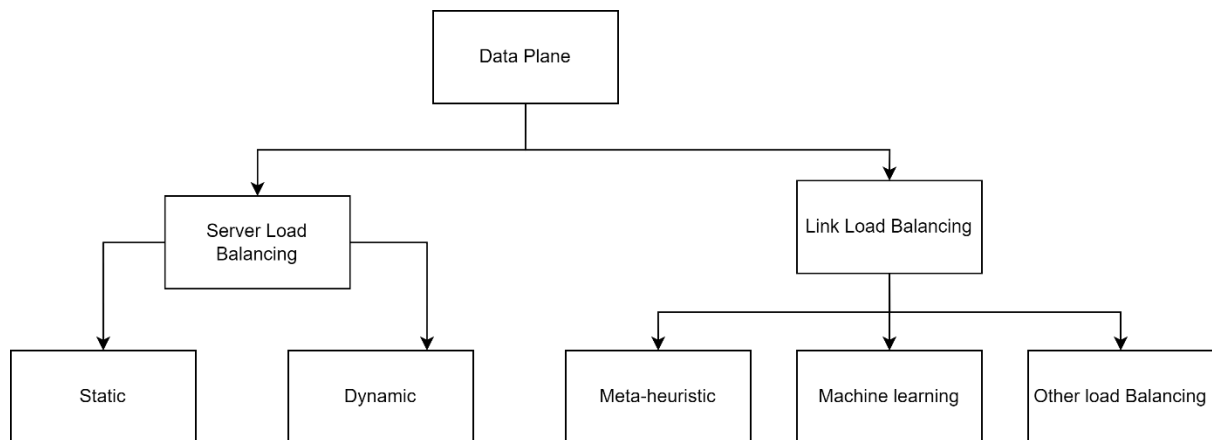


Εικόνα 2-11: Οι στόχοι της εξισορρόπησης φορτίου

Όπως φαίνεται στην εικόνα οι στόχοι είναι οι εξής:

- Improve end-to-end QoS metrics – βελτίωση των μετρήσεων QoS: Ο πρωταρχικός στόχος των LB είναι να εξασφαλίσουν από άκρο σε άκρο QoS για το δίκτυο SDN. Ως εκ τούτου, βελτιώνεται η απόδοση και η συνολική απόδοση του συστήματος. Το QoS στοχεύει να παρέχει καλύτερη εμπειρία χρήστη αποτρέποντας την υπερβολική καθυστέρηση της συσκευής, βελτιστοποιώντας την απόδοση και μειώνοντας τον χρόνο απόκρισης.
- Optimise resource utilization – βελτιστοποίηση της χρήσης των πόρων: Αυτός είναι ένας από τους πρωταρχικούς στόχους του load balancing, καθώς η σωστή χρήση των πόρων είναι σημαντική για την αποτελεσματικότητα του load balancing μοντέλου. Ως εκ τούτου, υπάρχει ένας βαθμός στον οποίο πρέπει να χρησιμοποιηθούν πόροι δικτύου όπως σύνδεση, εύρος ζώνης, επεξεργαστής και χρήση μνήμης. Οι κατάλληλοι αλγόριθμοι παροχής πόρων διασφαλίζουν τη μέγιστη χρήση των πόρων για το load balancing.
- Reduce transmission latency – μείωση της καθυστέρησης μετάδοσης: Η καθυστέρηση μετάδοσης αναφέρεται στο χρόνο που χρειάζεται ο κεντρικός διακόπτης για τη μετάδοση δεδομένων. Αυτό εξαρτάται από πολλούς παράγοντες που περιλαμβάνουν την απόδοση των switch, εάν υπάρχει συμφόρηση στην ουρά μετάδοσης και το μέγεθος των πακέτων δεδομένων. Η καθυστέρηση μετάδοσης δείχνει συμφόρηση στη ζεύξη, και με κάποιο άλλο τρόπο, χρησιμεύει επίσης ως η κατάσταση φορτίου του διακόπτη. Έτσι, ο ελεγκτής SDN πρέπει να συλλέξει τα byte που μεταδίδονται μέσα σε μια συγκεκριμένη περίοδο και τον ρυθμό μετάδοσης. Αυτή η παράμετρος πρέπει να μειωθεί
- Minimize response time – ελαχιστοποίηση του χρόνου απόκρισης: Αυτό καθορίζεται από το χρονικό διάστημα μεταξύ του χρόνου λήψης ενός αιτήματος διακομιστή και του χρόνου απάντησης ή της αποστολής. Επομένως, σε ένα καταναμημένο δίκτυο SDN, είναι καιρός να χρειαστεί ένας συγκεκριμένος αλγόριθμος LB για να αντιδράσει. Αυτή η παράμετρος πρέπει να ελαχιστοποιηθεί
- Avoiding bottlenecks – αποφυγή συμφόρησης: Για να αποφευχθεί τυχόν συμφόρηση ή συμφόρηση στη ρύθμιση του δικτύου SDN, απαιτούνται μέθοδοι load balancing για την ίση κατανομή του φορτίου μεταξύ διαφορετικών διακοπών/ελεγκτών, έτσι ώστε κανένας διακόπτης/ελεγκτής να μην υπερφορτώνεται (δηλαδή, μεταξύ των switches συμφόρησης κάθε συνδέσμου, ο καλύτερος είναι αυτός με τη χαμηλότερη χωρητικότητα). Η σωστή LB μπορεί να μειώσει την κατανάλωση πόρων μέσω της αποτελεσματικής χρήσης των διαθέσιμων πόρων. Επιπλέον, επιβάλλει το failover, επιτρέπει επεκτασιμότητα, αποτρέπει τα σημεία συμφόρησης και μειώνει τον χρόνο απόκρισης.
- Maximize throughput – μεγιστοποίηση του ρυθμού αποστολής ενός μηνύματος: Η υψηλή απόδοση είναι ένα επιθυμητό χαρακτηριστικό που απαιτείται για ένα δίκτυο υψηλής απόδοσης που είναι εφικτό μόνο εάν ο φόρτος εργασίας και οι πόροι καταναμηθούν εξίσου στους διάφορους κόμβους. Αυτό είναι το μέγεθος των δεδομένων που μεταφέρονται σωστά για μια συγκεκριμένη χρονική περίοδο από τη μια τοποθεσία στην άλλη.

Σε αυτήν την εικόνα βλέπουμε τις τεχνικές που χρησιμοποιούνται στο επίπεδο δεδομένων και τις υποκατηγορίες του.

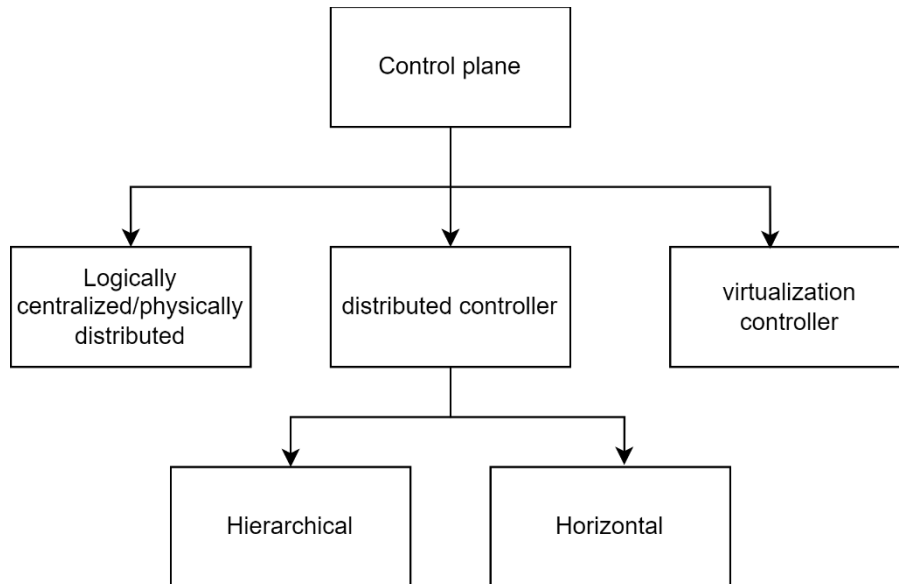


Εικόνα 2-12: Οι τεχνικές στο επίπεδο δεδομένων.

Όπως φαίνεται στην εικόνα μπορεί να γίνει load balancing στον server και στο link. Στον server υπάρχει στατικό – static και δυναμικό – dynamic load balancing. Οι στατικοί αλγόριθμοι είναι πιο απλοί από τους δυναμικούς αλλά είναι δαπανηροί. Είναι κατάλληλοι για ομοιογενείς server, αλλά ο τρόπος που είναι φτιαγμένοι τους καθιστά ακατάλληλους για δυναμικές αλλαγές. Οι δυναμικοί αλγόριθμοι κατανέμουν το φορτίο σύμφωνα με την τρέχουσα κατάσταση του κόμβου του δικτύου. Ο αλγόριθμος πραγματοποιεί έναν έλεγχο ρουτίνας σχετικά με την χωρητικότητα της σύνδεσης και το φορτίο του server κατά το χρόνο εκτέλεσης. Στους δυναμικούς αλγορίθμους οι περισσότερες τεχνικές προορίζονται για την βελτιστοποίηση του φορτίου διαδρομής και την επιλογή της λιγότερο φορτωμένης διαδρομής για τα εισερχόμενα αιτήματα δεδομένων για να ξεπεραστεί η συμφόρηση στο επίπεδο δεδομένων. Επιπλέον αυτή η τεχνική κατηγοριοποιείται σε τρία μέρη: μετα-ευρετικούς αλγόριθμους, αλγόριθμους μηχανικής μάθησης, άλλους αλγόριθμους. Το SDN προσφέρει μια τέλεια αρχιτεκτονική για βελτιστοποίηση σε όλο το δίκτυο χρησιμοποιώντας τεχνητή νοημοσύνη (AI) για τη δημιουργία ενός ανοιχτού, επεκτάσιμου, προγραμματίσιμου και διαχειρίσιμου δικτύου. Επίσης, οι προκλήσεις διαχείρισης και βελτιστοποίησης σε όλο το δίκτυο απαιτούν συνήθως τεράστιο χώρο για λύσεις, μεγάλο αριθμό μεταβλητών και αρκετούς στόχους. Οι ευρετικοί αλγόριθμοι μπορούν να λύσουν αυτά τα προβλήματα σε εύλογο χρόνο, αλλά γενικά περιορίζονται σε συγκεκριμένες συνθήκες του προβλήματος. Αρκετές μελέτες έχουν προτείνει τη χρήση αλγορίθμων μηχανικής μάθησης (ML) σε συνδυασμό με την αρχιτεκτονική SDN για βελτιωμένη απόδοση δρομολόγησης. Στο SDN, οι χειριστές δικτύου μπορούν να τοποθετήσουν κάποια κεντρική λογική στο επίπεδο ελέγχου, η οποία διευκολύνει τη λήψη μιας συνολικής εικόνας του δικτύου των αλγορίθμων ML. Η μέθοδος εξισορρόπησης φορτίου στο επίπεδο δεδομένων χρησιμοποιείται για τη βελτίωση της απόδοσης του δικτύου και τη βελτίωση της ικανότητάς του να ανταποκρίνεται στις ανάγκες QoS. Αυτή η προσέγγιση συνέκρινε τρεις διαφορετικούς τύπους ελεγκτών, τον ελεγκτή που βασίζεται στο spanning tree protocol (STP), τον ελεγκτή που βασίζεται σε load

balancing και τον ελεγκτή hub-like, για λίγο αξιολογώντας την επίδραση του load balancing σε δεδομένους δείκτες απόδοσης δικτύου. Τα πειραματικά ευρήματα έδειξαν μείωση της συμφόρησης του δικτύου ενώ η απόδοση του δικτύου γενικά αυξήθηκε.

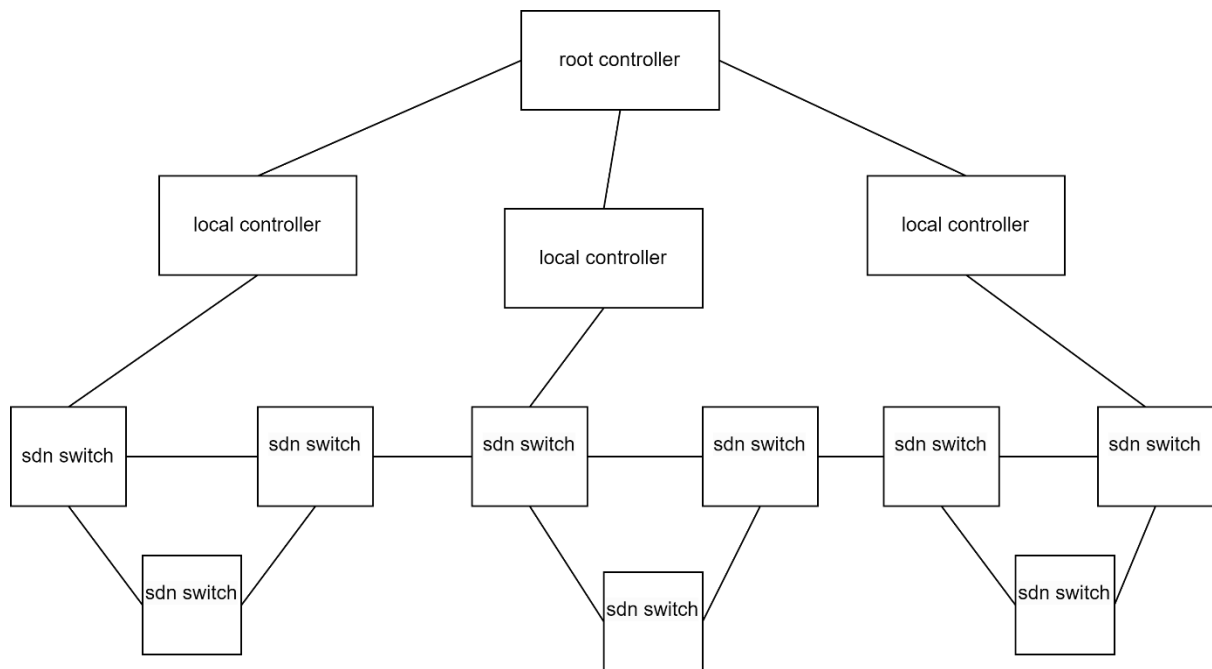
Σε αυτήν την εικόνα βλέπουμε τις τεχνικές που χρησιμοποιούνται στο επίπεδο ελέγχου.



Εικόνα 2-13: Οι τεχνικές στο επίπεδο ελέγχου

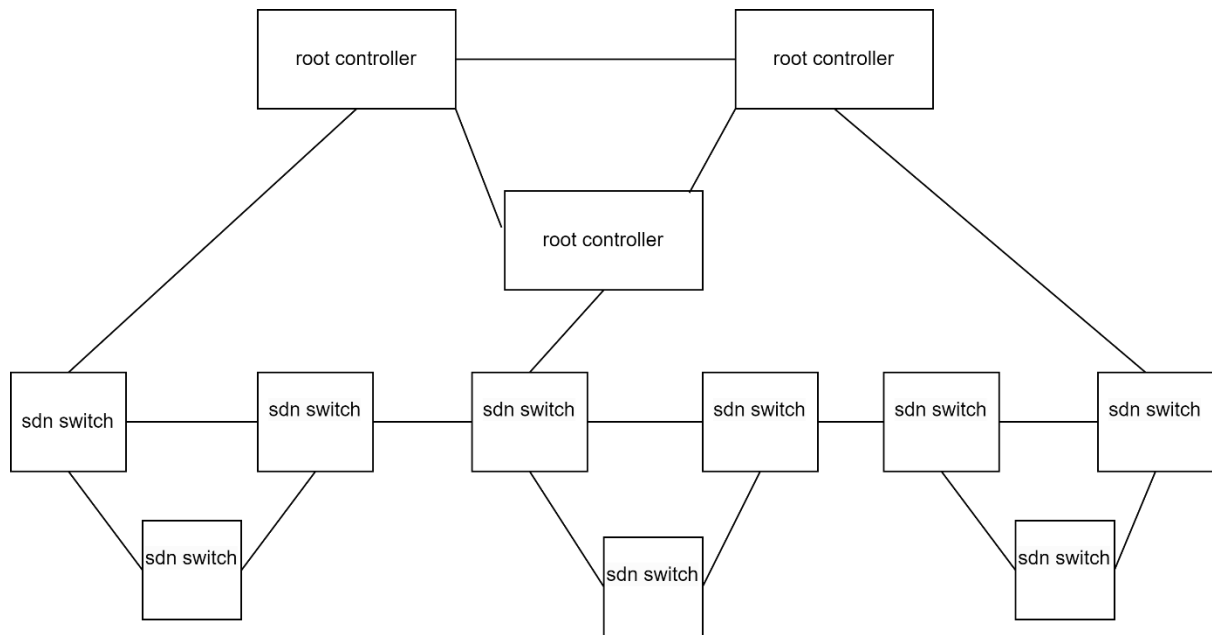
Το επίπεδο ελέγχου load balancing εξισορροπεί τα φορτία σε κατανεμημένους ελεγκτές για να αποφύγει τα σημεία συμφόρησης που σχετίζονται με το τεράστιο δίκτυο SDN σε έναν κεντρικό ελεγκτή. Με βάση μελέτες που έχουν διεξαχθεί σε πολλαπλούς ελεγκτές, μπορεί να κατηγοριοποιηθεί σε λογικά συγκεντρωτικά/φυσικά κατανεμημένα, κατανεμημένοι load balancers, τα οποία χωρίζονται σε ιεραρχικά, και οριζόντια και ελεγκτές εικονικοποίησης – virtualization load balancing. Στην κατηγορία logically centralized/physically distributed οι ελεγκτές είναι λογικά κεντριοποιημένοι και φυσικά σε μία κατανεμημένη αρχιτεκτονική. Κάθε ελεγκτής μπορεί να επικοινωνήσει ή να αποφασίσει για τον τομέα για τον οποίο είναι υπεύθυνος. Η διαφορά σε σχέση με τον λογικά συγκεντρωτικό σχεδιασμό είναι ότι ο κάθε ελεγκτής βλέπει όλο το δίκτυο για να πάρει απόφαση. Οι αρχιτεκτονικές load balancing κατανεμημένου ελεγκτή συνεπάγονται τη χρήση ενός ή περισσότερων ελεγκτών σε δίκτυα για την αντιμετώπιση προκλήσεων που αντιμετωπίζει ένα μεμονωμένο δίκτυο ελεγκτών. Το σκεπτικό γι' αυτό είναι να διασφαλιστεί ότι διαμορφώνονται ελεγκτές που μπορούν να επιτρέψουν την εξίσου κατανομή του φορτίου στο δίκτυο. Επιπλέον, ένας ελεγκτής μπορεί να αναλάβει τη διαχείριση από έναν άλλο ελεγκτή σε περίπτωση κατάρρευσης. Οι περισσότερες από αυτές τις περίπλοκες διαδικασίες απαιτούν περιορισμένες τεχνολογικές υλοποιήσεις από κατανεμημένα συστήματα. Αυτός ο τύπος δικτύου είναι φυσικά κατανεμημένος και διαφορετικός με βάση τη θέση των ελεγκτών και τον τύπο επικοινωνίας. Το κύριο πρόβλημα με τις λύσεις κατανεμημένων ελεγκτών σχετίζεται με τη διατήρηση της συνέπειας σε διάφορους κατανεμημένους ελεγκτές. Σε αντίθεση με τις κεντρικές αποφάσεις, οι κατανεμημένες αποφάσεις έχουν προτιμώμενη απόδοση που είναι κατάλληλη από την

άποψη της ανταπόκρισης, της αξιοπιστίας και της επεκτασιμότητας ως αποτέλεσμα της φυσικής κατανομής και της λογικής συγκέντρωσης. Από την άλλη πλευρά, οι αρχιτεκτονικές εξισορρόπησης φορτίου κατανεμημένου ελεγκτή χωρίζονται για να οργανώσουν την εξισορρόπηση φορτίου του ελεγκτή ως ιεραρχική και οριζόντια εξισορρόπηση φορτίου. Στην ιεραρχική δομή ο κάθε ελεγκτής τοποθετείται σε διαφορετικό επίπεδο. Ωστόσο, έχει καταστεί αδύνατο για τους ελεγκτές εντός των επιπέδων να επικοινωνούν μεταξύ τους μέσα στο σύστημα, και έτσι περιορίζει τη χρήση των υπηρεσιών δεύτερης βαθμίδας που απαιτούνται για μια συνολική προβολή του δικτύου.



Εικόνα 2-14: Ιεραρχική δομή με ελεγκτές sdn

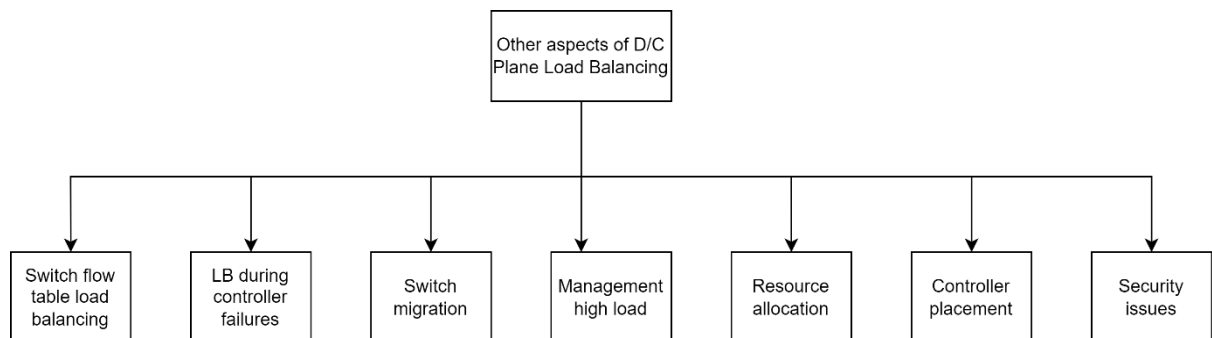
Στην οριζόντια δομή όλοι οι ελεγκτές βρίσκονται στο ίδιο επίπεδο. Ωστόσο αυτή η δομή αναφέρεται και σαν επίπεδος ελεγκτής. Αν και κάθε ελεγκτής βρίσκει διαφορετικούς τομείς, όλοι οι ελεγκτές θεωρούνται ως ελεγκτές ρίζας που είναι εξοικειωμένοι με ολόκληρη την κατάσταση του δικτύου. Χρησιμοποιούνται τα east – west interface για την επικοινωνία μεταξύ των ελεγκτών και μοιράζονται τις πληροφορίες των δικτύων, διότι ο κάθε ελεγκτής ελέγχει ένα συγκεκριμένο τομέα.



Εικόνα 2-15: οριζόντια δομή με ελεγκτές sdn

Ένας άλλος τύπος εξισορρόπησης φορτίου ελέγχου χρησιμοποιώντας την τεχνική slices βασίζεται στην εικονικοποίηση δικτύου SDN. Σε αυτή την προσέγγιση, η φυσική υποδομή δικτύου έχει ένα εικονικό επίπεδο που τοποθετείται από πάνω της. Με τον έλεγχο της δρομολόγησης πακέτων και της ισορροπίας φορτίου, μπορεί να επιτευχθεί ένα εικονικοποιημένο δίκτυο διεγείροντας τα καλώδια αδράνειας ώστε να μοιραστούν το φορτίο τους. Η εικονικοποίηση δικτύου παρέχεται σε αυτό το επίπεδο μέσω της κατασκευής

Σε αυτήν την εικόνα βλέπουμε άλλες πτυχές χρήσης του load balancer στα SDN δίκτυα.



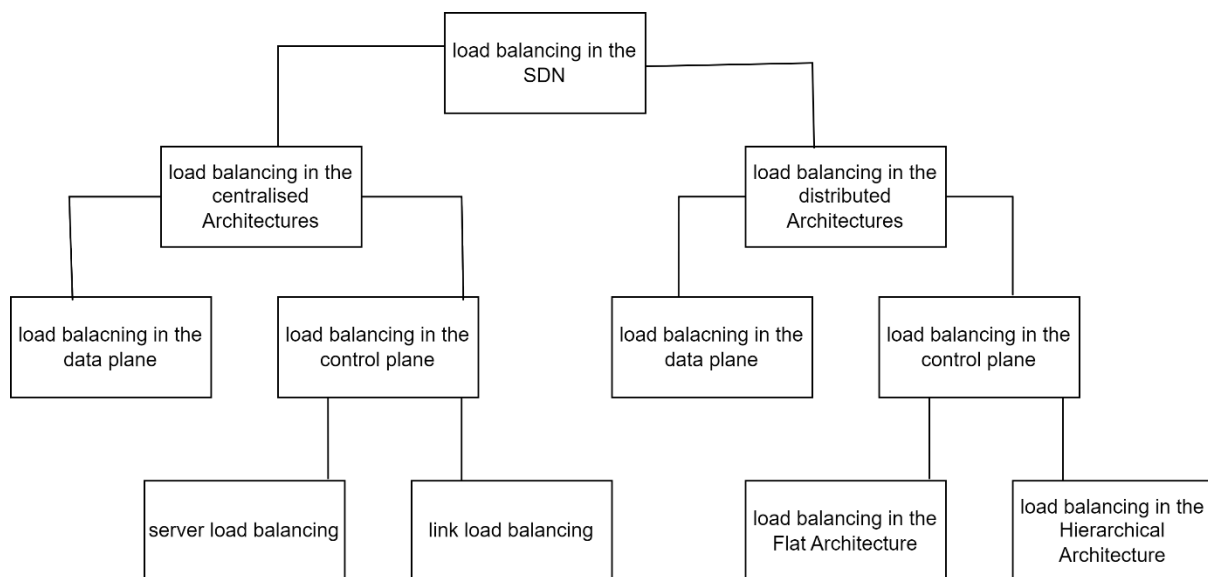
Εικόνα 2-16: Άλλες περιπτώσεις χρήσης των ελεγκτών sdn

Ως αναδυόμενη τεχνολογία, το SDN εξασφαλίζει ευελιξία στον έλεγχο της ροής του δικτύου. Σε αντίθεση με το συμβατικό δίκτυο IP, όπου οι εργασίες δρομολόγησης εκτελούνται χρησιμοποιώντας παραδοσιακά σχήματα δρομολόγησης (όπως ECMP και OSPF) με μη εποπτευόμενες συσκευές προώθησης (όπως routers και switches) διασκορπισμένες σε όλο το δίκτυο, ένας κεντρικός ελεγκτής με τη δυνατότητα καθολικής προβολής του δικτύου χρησιμοποιείται στο SDN για τη δρομολόγηση ροών και προσαρμόζεται ανάλογα με τη διακύμανση της κατάστασης εντός του δικτύου. Οι βασικές λειτουργίες του δικτύου (όπως η προώθηση στο επίπεδο 2, η δρομολόγηση στο επίπεδο 3 και η ταξινόμηση των πακέτων στο

επίπεδο-2-4) αντιμετωπίζονται/υποτίθεται ότι είναι καταχωρήσεις ροής στις προδιαγραφές OpenFlow. Ωστόσο, λόγω της υψηλής κατανάλωσης ενέργειας και του κόστους των σιπ, υπάρχει περιορισμένος χώρος στον πίνακα ροής στους περισσότερους διακόπτες που είναι συμβατοί με το OpenFlow. Ως εκ τούτου, καθώς ο αριθμός των ταυτόχρονων ροών γίνεται μεγάλος στο δίκτυο, οι πίνακες ροής είναι ανεπαρκείς. Αυτή η ανεπάρκεια αυξάνει την πιθανότητα υπερχειλίσης του πίνακα ροής. Για να αντιμετωπιστεί αυτό το μειονέκτημα, οι καταχωρήσεις ροής που χρησιμοποιούνται συχνά αφαιρούνται για να δημιουργηθεί χώρος για νέες ροές. Επιπλέον, η υπερχειλίση του πίνακα ροής υποδηλώνει ότι ο ελεγκτής πρέπει να ενημερώνει κατά διαστήματα τον πίνακα ροής με αποτέλεσμα την υποβάθμιση της εξόδου του TCP λόγω διακοπής των ροών ρεύματος. Αποτυχία λογισμικού ή υλικού που οδηγεί σε αστοχία του κύριου ελεγκτή σε έναν κατανεμημένο SDN πολλαπλών τομέων και πολλαπλών ελεγκτών SDN LB εκκινεί την άμεση κατανομή προκαθορισμένων υποτελών ελεγκτών στους διακόπτες που ελέγχονταν από τον αποτυχημένο κύριο ελεγκτή. Αν και αυτή η εφεδρική προσέγγιση βελτιώνει την αξιοπιστία καθώς και την επεκτασιμότητα του δικτύου SDN, ελάχιστη ή καθόλου έρευνα υπάρχει σχετικά με τον τρόπο σχεδιασμού της ανάθεσης slave ελεγκτή μετά από αποτυχία του κύριου ελεγκτή στην τρέχουσα εργασία. Γενικά, η παρούσα προσέγγιση της εκχώρησης εξαρτημένων ελεγκτών είναι άκαμπτη και εκχωρεί τον πλησιέστερο ελεγκτή στον διακόπτη ως υποτελή, όπως έχει προκαθοριστεί. Ένα σημαντικό ελάττωμα σε αυτό το σχήμα εκχώρησης αποκαλύπτεται όταν υπάρχει αστοχία της αλυσίδας ελεγκτή. Ο φόρτος κυκλοφορίας θα πρέπει να μεταφέρεται στους άλλους ελεγκτές μέσω του ελεγκτή υπερφόρτωσης που είναι υπεύθυνος για τη ροή του δικτύου για να βελτιωθεί η ικανότητα του επιπέδου ελέγχου να αυξάνει την απόκριση σε αιτήματα μεταγωγής. Αυτή η έννοια ονομάζεται μηχανισμός μετανάστευσης – migration mechanism. Αυτός ο μηχανισμός βελτιώνει επίσης τη διαθεσιμότητα του ελεγκτή. Η αρχιτεκτονική του ελεγκτή SDN περιλαμβάνει έναν ή περισσότερους ελεγκτές χρήσιμους για τη διαχείριση του φορτίου SDN (δηλαδή επεκτασιμότητα, αξιοπιστία, διαθεσιμότητα) χρησιμοποιώντας κατανεμημένους ελεγκτές SDN. Η ιδέα βασίζεται στην προϋπόθεση ότι καθώς σχηματίζονται πολλαπλοί ελεγκτές, μπορούν να μοιράζονται το φορτίο εντός του δικτύου. Ομοίως, ένας ελεγκτής μπορεί να εκτελέσει τη δουλειά ενός άλλου ελεγκτή, ειδικά όταν υπάρχει σύγκρουση. Το πιο σημαντικό, αυτές οι διαδικασίες μπορεί να είναι πιο περίπλοκες, έτσι ώστε να απαιτείται η υιοθέτηση μερικών μεθόδων από κατανεμημένα συστήματα ελεγκτών SDN. Με την εμφάνιση καινοτόμων υπηρεσιών και εφαρμογών, η εξέλιξη των τεχνολογιών επικοινωνίας έχει γίνει ένα κρίσιμο καθήκον για τις αναπτυσσόμενες προσεγγίσεις για τη διαχείριση των πόρων των δικτύων με την ελάχιστη ανθρώπινη παρέμβαση. Για να καθιερωθεί η διαχείριση και ο έλεγχος με αποτελεσματικό και προσαρμοστικό τρόπο, υπάρχει ανάγκη να εξεταστούν τόσο στατικά όσο και δυναμικά σενάρια όπου απαιτείται μια προσέγγιση LB κατανεμημένου ελεγκτή. Οι αρχιτεκτονικές κατανεμημένων ελεγκτών χρησιμοποιούν πολλαπλούς ελεγκτές για την επίλυση των προβλημάτων που είναι γνωστά με την τοποθέτηση ενός ελεγκτή SDN, π.χ. διαθεσιμότητα. Οι πολλαπλοί ελεγκτές εξασφαλίζουν μειωμένη καθυστέρηση, βελτιωμένη επεκτασιμότητα καθώς και ανοχή σε ένα σφάλμα. Ωστόσο, η επιβάρυνση αναζήτησης για επικοινωνία μεταξύ διακοπών και πολλών ελεγκτών αυξάνεται όταν χρησιμοποιείται αυτή η αρχιτεκτονική. Ένα πιθανό μειονέκτημα αυτής της προσέγγισης είναι η διασφάλιση ότι ολόκληρο το κατανεμημένο δίκτυο διατηρείται με συνέπεια. Η εφαρμογή δικτύου τείνει να ενεργεί ακατάλληλα, ιδιαίτερα όταν η άποψη της κατάστασης του

παγκόσμιου δικτύου είναι ασυνάρτητη. Η έννοια της ασφάλειας και της εξισορρόπησης φορτίου είναι κρίσιμα ζητήματα που προκύπτουν στην επιχειρηματική αγορά SDN. Η ανάπτυξη και η διαχείριση λύσεων ασφαλείας είναι δύσκολη σε ένα μεγάλο περιβάλλον πολλών προμηθευτών. Από την άποψη της ασφάλειας, το SDN παρακολουθεί πολλαπλά επίπεδα πακέτων σε όλο το δίκτυο. Η ανάπτυξη συσκευών ασφαλείας, π.χ. τείχη προστασίας, είναι πολύ δύσκολη στα δίκτυα SDN, επειδή το SDN μπορεί να χρησιμοποιήσει ένα κεντρικό τείχος προστασίας για τη δρομολόγηση όλης της κυκλοφορίας. Η χρήση ενός μόνο σημείου για τη ροή της κυκλοφορίας είναι ευάλωτη, αλλά είναι εύκολο να καταγράψετε και να αναλύσετε IDS και IPS σε πραγματικό χρόνο. Υπάρχουν πολλές απειλές ασφαλείας σε ένα δίκτυο SDN, όπως η δημιουργία εμπιστοσύνης μεταξύ του switch και του ελεγκτή, η ασφάλιση και η προστασία του ελεγκτή SDN, DoS/DDoS, μεγάλες ουρές αναμονής, εγκληματολογία και αποκατάσταση και η ανάπτυξη ενός ισχυρού πλαισίου πολιτικής.

Οι κατηγορίες της εξισορρόπησης φορτίου.



Εικόνα 2-17: κατηγορίες εξισορρόπησης φορτίου

Όπως φαίνεται από το σχήμα η εξισορρόπηση φορτίου στα SDN χωρίζεται σε δύο μεγάλες κατηγορίες: το κεντρικές αρχιτεκτονικές – ένας μονός ελεγκτής και στις κατανεμημένες αρχιτεκτονικές – πολλαπλοί ελεγκτές. Μετά κάνουμε διαχωρισμό σε πιο επίπεδο του ελεγκτή θα κάνουμε εξισορρόπηση φορτίου: επίπεδο δεδομένων – data plane, επίπεδο ελέγχου – control plane. Το πιο κοινό επίπεδο που γίνεται εξισορρόπηση είναι το control plane. Στις κατανεμημένες αρχιτεκτονικές υπάρχουν δυο τεχνικές που χρησιμοποιούνται συνήθως: επίπεδες αρχιτεκτονικές – flat architecture, ιεραρχικές αρχιτεκτονικές – hierarchical architecture. Όπως και στις κατανεμημένες αρχιτεκτονικές έτσι και στις κεντρικές αρχιτεκτονικές υπάρχουν δυο τεχνικές που χρησιμοποιούνται κατά κύριο λόγο: διακομιστής – server load balancing, σύνδεση – link load balancing.

2.4 Ταξινόμηση αλγορίθμων εξισορρόπησης φορτίου στα SDN

2.4.1 Κατάταξη με βάση το επίπεδο εφαρμογής

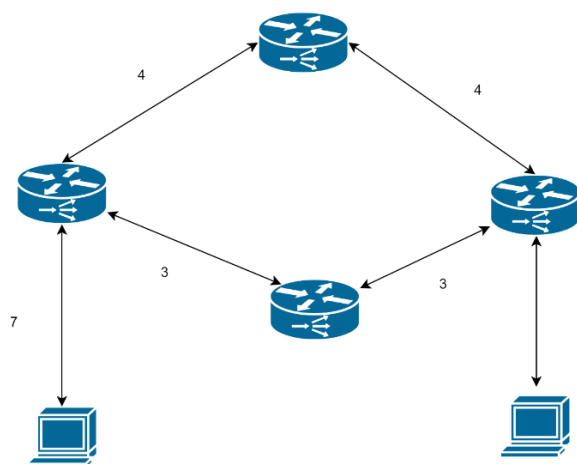
Η τεχνική εξισορρόπησης φορτίου είναι ένας σημαντικός τρόπος για να αποδοθούν πόροι αποτελεσματικά σε ένα δίκτυο, και να βελτιστοποιηθεί η απόδοση και η ποιότητα υπηρεσίας του. Η αρχιτεκτονική των δικτύων καθοριζόμενα από λογισμικό είναι υψηλότερα σε απόδοση και αποτελεσματικότητα σε σύγκριση με τα παραδοσιακά δίκτυα. Στην κεντρική αρχιτεκτονική γίνεται εξισορρόπηση φορτίου στο επίπεδο δεδομένων και συμπεριλαμβάνει εξισορρόπηση σε διακομιστή – server και εξισορρόπηση σε σύνδεση – link. Ένας πολύ κοινός αλγόριθμος ο οποίος χρησιμοποιείται στην δεύτερη περίπτωση είναι ο ECMP – Equal Cost Multi Path. Σε πολλαπλούς ελεγκτές που η εξισορρόπηση συμβαίνει στο επίπεδο ελέγχου μπορεί να εφαρμοστεί σε επίπεδες αρχιτεκτονικές και ιεραρχικές αρχιτεκτονικές.

Η εξισορρόπηση φορτίου στα κέντρα δεδομένων διαχωρίζεται σε δύο κατηγορίες. Η μία κατηγορία κάνει εξισορρόπηση φορτίου εργασίας (working load) στους servers, ενώ η άλλη κατηγορία κάνει εξισορρόπηση κίνησης φόρτου (traffic load) στο δίκτυο ή στους συνδέσμους (link load).

- Εξισορρόπηση φορτίου στους servers (server load balancing): Τα κέντρα δεδομένων αποτελούνται κατά κύριο λόγο από πανομοιότυπους διακομιστές (server). Η κύρια εργασία του εξισορροπιστή (load balancer) είναι να κατανέμει τις διεργασίες που έρχονται μέσα στο κέντρο δεδομένων μεταξύ των server για να αποφευχθεί η επιβάρυνση ορισμένων server με μεγάλα φορτία. Οι μέθοδοι της εξισορρόπησης φορτίου μπορεί να είναι είτε στατικές είτε δυναμικές. Οι στατικές μέθοδοι χρησιμοποιούν κανόνες, και με βάση αυτούς τους προκαθορισμένους κανόνες εκχωρούν τις εργασίες στους διάφορους servers. Από την άλλη μεριά οι δυναμικές μέθοδοι λαμβάνουν υπόψιν τον φόρτο εργασίας στον εκάστοτε server, αυτό γίνεται με κάποιον μηχανισμό ενημέρωσης ή στέλνουν feedback στον load balancer, και μπορούν να κάνουν καλύτερη εξισορρόπηση φορτίου. Οι τρεις κοινοί μέθοδοι εξισορρόπησης φορτίου σε αυτήν την κατηγορία είναι: Round Robin, Least connections, Shortest expected delay.
 - Ο Round Robin είναι μία από τις απλούστερες μεθόδους για την διανομή αιτημάτων client σε μία ομάδα διακομιστών. Ο load balancer έχει μία λίστα με τους διακομιστές και κάθε φορά που έρχεται ένα αίτημα από πελάτη το προωθεί σε κάθε έναν διακομιστή. Αυτή η διαδικασία γίνεται κυκλικά. Όταν φτάσει στο τέλος της λίστας ο balancer ξεκινάει ξανά από την αρχή την λίστα με τους server. Το κύριο πλεονέκτημα αυτού του αλγορίθμου είναι ότι είναι απλό στην εφαρμογή του. Ωστόσο δεν έχει ως αποτέλεσμα την ακριβή και αποδοτικότερη διανομή της κίνησης. Υπάρχουν κάποιες παραλλαγές αυτού του αλγορίθμου που λαμβάνουν υπόψιν πρόσθετους παράγοντες και μπορούν να έχουν καλύτερο load balancing. Αυτοί είναι: weighted round robin, dynamic round robin. Στον Weighted round robin ένα βάρος “weight” εκχωρείται σε κάθε διακομιστή με βάση τα κριτήρια που επιλέγει ο διαχειριστής. Το σύνηθες κριτήριο είναι ικανότητα διαχείρισης κίνησης – traffic-handling capacity. Όσο μεγαλύτερο το βάρος τόσο μεγαλύτερο το

ποσοστό των αιτημάτων από πελάτες που μπορεί να λάβει ο εκάστοτε διακομιστής. Στον Dynamic round robin ένα βάρος εκχωρείται και εδώ όπως και πριν απλά εδώ γίνεται με βάση δεδομένων σε πραγματικό χρόνο σχετικά με το τρέχων φορτίο – current load και την χωρητικότητα αδράνειας του διακομιστή.

- Ο Least Connection είναι ένας δυναμικός αλγόριθμος όπου τα αιτήματα των client διανέμονται στους διακομιστές με βάση τον λιγότερο αριθμό ενεργών συνδέσεων τη στιγμή που λαμβάνεται το αίτημα του πελάτη. Αυτός ο αλγόριθμος λαμβάνει υπόψιν το ενεργό φορτίο σύνδεσης. Σε περιπτώσεις που οι διακομιστές έχουν παρόμοιες προδιαγραφές, κάποιος διακομιστής μπορεί να υπερφορτωθεί λόγω των συνδέσεων μεγάλης διάρκειας. Αυτή η τεχνική είναι κατάλληλη για εισερχόμενα αιτήματα που έχουν διαφορετικούς χρόνους σύνδεσης.
- Ο Shorted Expected Delay είναι ένας δυναμικός αλγόριθμος όπου τα αιτήματα των client διανέμονται στους διακομιστές με βάση την μικρότερη αναμενόμενη καθυστέρηση. Η αναμενόμενη καθυστέρηση που θα αντιμετωπίσει η εργασία είναι $(C_i + 1) / U_i$, U_i εάν σταλεί στον i th διακομιστή, στον οποίο C_i είναι ο αριθμός των εργασιών στον i th διακομιστή και U_i είναι ο σταθερός ρυθμός εξυπηρέτησης (βάρος) του i th διακομιστή .
- Εξισορρόπηση φορτίου στο δίκτυο (network load balancing) ή αλλιώς στους συνδέσμους (link load balancing): Όταν αντιστοιχίζονται πολλαπλές ροές στον ίδιο σύνδεσμο, ορισμένες από τις ροές δεδομένων ενδέχεται να υποφέρουν από ορισμένα προβλήματα, όπως ο αυξημένος χρόνος αναμονής, η μεγάλη καθυστέρηση μετάδοσης και ούτω καθεξής. Χρησιμοποιώντας μια συγκεκριμένη στρατηγική προγραμματισμού για τη διανομή των ροών δεδομένων σε διαφορετικές διαδρομές για τη βελτίωση της χρήσης της σύνδεσης. Αυτή η αρχιτεκτονική διαχειρίζεται το φορτίο μεταξύ πολλαπλών συνδέσεων διασφαλίζοντας πλεονασμό και υψηλή απόδοση.



Εικόνα 2-18: Εξισορρόπηση φορτίου στους συνδέσμους, όταν υπάρχουν εναλλακτικές διαδρομές.

Μία τεχνική [38] η οποία μπορεί να εφαρμοστεί σε κέντρα δεδομένων για την βελτίωση του throughput και μείωσης της καθυστέρησης – latency είναι η explicit congestion notification (ECN). Σε αυτήν την τεχνική [38] υπάρχουν πολλαπλές ουρές

υπηρεσιών που σχετίζονται με κάθε θύρα για την απομόνωση διαφορετικών κατηγοριών κυκλοφορίας [25]. Μία άλλη μελέτη [39] στοχεύει να υποστηρίξει την ισορροπία μικρό-φορτίου για DCN αφήνοντας κάθε switch να λάβει την απόφασή του χωρίς έναν κεντρικό ελεγκτή. Για να γίνει αυτό, θα πρέπει να προ-εγκατασταθεί στον πίνακα προώθησης σε κάθε switch ένα σύνολο υποψήφιων διαδρομών (next-hop) για κάθε προορισμό. Στη συνέχεια, ένας switch αποφασίζει τοπικά πού θα δρομολογήσει κάθε πακέτο και επιδιώκει να εξισορροπήσει τα φορτία των θυρών του [25]. Σε μία άλλη μελέτη [40] εξετάζεται ένα DCN μικρής κλίμακας, όπου κάθε switch Top-of-Rack (ToR) συνδέεται με τα υπόλοιπα στοιχεία στο DCN (π.χ. aggregation switches ή servers). Για να εξισορροπήσει τα φορτία κυκλοφορίας, διαμορφώνει εκ νέου την τοπολογία του δικτύου για να επιτρέψει στους ToR switches με βαριά φορτία να χρησιμοποιούν ανεπαρκώς χρησιμοποιούμενες uplinks από τους γείτονές τους. Οι πιο πάνω τεχνικές δεν χρησιμοποιούν τα Software Define Networking για να εφαρμόσουν load balancing on Network. Έχουν γίνει όμως κάποιες ερευνητικές προσπάθειες που χρησιμοποιούν – υιοθετούν το SDN για τη διαχείριση των ροών κυκλοφορίας στα DCN. Στην προκειμένη προσπάθεια, υποθέτοντας ότι τα switches σχηματίζουν μια τοπολογία πλέγματος, εκχωρείται ένα κόστος για κάθε σύνδεση για τη δρομολόγηση της κυκλοφορίας. Όταν ένας σύνδεσμος έχει χαμηλότερη χρήση, οι ροές VoIP δίνονται με χαμηλότερο κόστος σε αυτόν τον σύνδεσμο. Στη συνέχεια, ο αλγόριθμος του Dijkstra χρησιμοποιείται για την εύρεση της συντομότερης διαδρομής για κάθε ροή. Έτσι, οι ροές VoIP μπορεί να έχουν προτεραιότητα έναντι άλλων για να αποκτήσουν εύρος ζώνης δικτύου. Μία άλλη εργασία [42] χρησιμοποιεί την τεχνική depth-first αναζήτηση βάθους, για να βρει πολλαπλές διαδρομές από την πηγή στον προορισμό σε ένα DCN με την τοπολογία Abilene [41]. Στη συνέχεια, επιλέγεται η διαδρομή που προσφέρει το μέγιστο εύρος ζώνης για αποστολή δεδομένων. Επιπλέον, όταν η συνολική χρήση του εύρους ζώνης είναι πάνω από 80%, νέα αιτήματα κίνησης θα απορρίπτονται για να αποφευχθεί η συμφόρηση του δικτύου. [25]

2.4.2 Κατάταξη με βάση τη συχνότητα αλλαγής αποφάσεων για εξισορρόπηση φορτίου

Μία άλλη κατηγοριοποίηση αλγορίθμων για την εξισορρόπηση φορτίου είναι η παρακάτω:

- οι στατικοί αλγόριθμοι: Οι στατικοί αλγόριθμοι διανέμουν τα αιτήματα χωρίς να λαμβάνουν υπόψη την τρέχουσα κατάσταση του συστήματος. Δεν λαμβάνουν κάποια ανατροφοδότηση από τους διακομιστές. Η ανάθεση των εργασιών γίνεται με βάση ένα προκαθορισμένο σχέδιο. Οι αλγόριθμοι αυτοί είναι εύκολη στην ρύθμιση τους αλλά όταν υπάρχει μεγάλος φόρτος αιτημάτων είναι αναποτελεσματικοί. Κάποιοι αλγόριθμοι είναι: Round Robin, weighted Round Robin, IP Hash.
- οι δυναμικοί αλγόριθμοι: Οι δυναμικοί αλγόριθμοι λαμβάνουν την τρέχων διαθεσιμότητα, τον φόρτο και την υγεία των διακομιστών του συστήματος. Οι διεργασίες κινούνται δυναμικά από τον ένα διακομιστή στον άλλον με σκοπό την γρήγορη επεξεργασία των εργασιών στο σύστημα. Αυτοί οι αλγόριθμοι

αποδίδουν καλύτερα αποτελέσματα όταν ο χρόνος εκτέλεσης διαφέρει από μία εργασία σε μία άλλη. Κάποιοι αλγόριθμοι είναι: Least connections, Least response time, Resource based.

2.4.3 Reactive vs Proactive Load Balancing

Μία άλλη μεγάλη κατηγοριοποίηση μεθόδων load balancing είναι για το αν ο αλγόριθμος είναι reactive ή proactive. Η κύρια διαφορά σε αυτού του είδους την κατηγοριοποίηση είναι για το αν προϋπάρχουν κανόνες στον ελεγκτή για το πού θα γίνει προώθηση του πακέτου. Στους proactive αλγόριθμους, η προώθηση των πακέτων γίνεται με βάση κανόνες οι οποίοι έχουν προκαθοριστεί από τον χρήστη. Στους reactive αλγόριθμους το πακέτο εισέρχεται στον ελεγκτή, γίνεται η επεξεργασία του και μετά γίνεται η προώθηση του. Στο Reactive Load Balancing, το πρώτο πακέτο ροής αποστέλλεται στον ελεγκτή για επεξεργασία. Ο ελεγκτής αξιολογεί το πακέτο και στέλνει μήνυμα ροής για αλλαγή εισαγωγής μιας καταχώρησης πίνακα ροής για το πακέτο. Τα επόμενα πακέτα (σε ροή) ταιριάζουν με την καταχώρηση μέχρι την αδράνεια ή το δύσκολο χρονικό όριο. Το Reactive Load Balancing έχει το πρόβλημα της προέλευσης του DOS – Denial Of Service στο switch (συμβάν εισόδου πακέτου). Η έντονη αποστολή πακέτων – busty behaviour προς τον ελεγκτή μπορεί να δημιουργήσει παροδική συμφόρηση στην οποία πρέπει να αντιδράσουμε πριν από την υπερχειλίση των buffer του switch για να αποφευχθεί η απώλεια. Αυτή η απαίτηση καθιστά αναποτελεσματικά τα περισσότερα από τα κεντρικά αντιδραστικά σχήματα, καθώς είναι συχνά πολύ αργά για να αντιδράσουν σε οποιαδήποτε εκτός από τα μεγαλύτερα συμβάντα δικτύου, π.χ., αποτυχίες σύνδεσης. Επιπλέον, τα κεντρικά σχήματα μπορούν να υποβαθμίσουν την απόδοση κατά την αναδρομολόγηση ροών χρησιμοποιώντας παλιές πληροφορίες. Υπάρχει μεγάλη πιθανότητα τα switch να στέλνουν ταυτόχρονα και με μεγάλη ταχύτητα πακέτα για επεξεργασία στον ελεγκτή και ο ελεγκτής να μην μπορεί να ανταπεξέλθει. Τα κατανεμημένα αντιδραστικά σχήματα όπως το CONGA μπορούν να ανταποκριθούν στη συμφόρηση σε ταχύτερα χρονοδιαγράμματα, αλλά έχουν υψηλό εμπόδιο στην ανάπτυξη. Η Presto [43] τεχνική ακολουθεί μια προληπτική, σωστή προσέγγιση για τη διαχείριση της συμφόρησης. Δηλαδή, εάν μικρά, σχεδόν ομοιόμορφα τμήματα κίνησης είναι εξίσου ισορροπημένα σε μια συμμετρική τοπολογία δικτύου, τότε η εξισορρόπηση φορτίου μπορεί να παραμείνει αγνωστική στη συμφόρηση και να αφήσει τον έλεγχο συμφόρησης στα υψηλότερα επίπεδα της στοίβας δικτύωσης. Στο Proactive Load Balancing, οι Ελεγκτές συμπληρώνουν εκ των προτέρων στοιχεία ροής στο switch πριν φτάσουν τα πακέτα, έτσι ώστε οι proactive ροές εξαλείφουν κάθε λανθάνουσα κατάσταση που προκαλείται από έναν ελεγκτή σε κάθε ροή. Υπάρχει μηδενικός χρόνος ρύθμισης για νέες ροές. Η απώλεια σύνδεσης μεταξύ switch και ελεγκτή δεν διακόπτει την κυκλοφορία του δικτύου. Το proactive load balancing απαιτεί πιο έξυπνες προσεγγίσεις από την απλή αντίδραση σε συμβάντα δικτύου (γνώση όλου του δικτύου, ανακάλυψη, ενημερώσεις κ.λπ.)[16]

2.5 Μετρήσεις απόδοσης για τις τεχνικές εξισορρόπησης φορτίου.

Παρακάτω θα αναφερθούν και θα αναλυθούν οι μετρικές, οι οποίες χρησιμοποιούνται στην αξιολόγηση της επίδοσης των αλγορίθμων εξισορρόπησης φορτίου. Οι βασικές μετρικές είναι οι ακόλουθες:

- Packet loss rate ($P_r(\text{Loss})$) – ρυθμός απώλειας πακέτων: Ο ρυθμός απώλειας πακέτων είναι ένα κοινό φαινόμενο που εμφανίζεται συχνά σε συσκευές μετάδοσης δεδομένων. Όταν οι switches στο δίκτυο είναι απασχολημένοι με την επεξεργασία των εισερχόμενων πακέτων, είναι πιθανό να απορρίψουν μερικά από τα εισερχόμενα πακέτα. Οι switches είναι απασχολημένοι ως αποτέλεσμα του ρυθμού απώλειας πακέτων καθώς και της κατάστασης φορτίου της διαδρομής. Ο ελεγκτής SDN θα είναι σε θέση να διασφαλίσει ότι ο συσσωρευμένος αριθμός των μεταδιδόμενων πακέτων Tx_{packet} συλλέγονται στις θύρες των OpenFlow switches, καθώς και ο αθροιστικός αριθμός του πακέτου που είναι ως λαμβανόμενο Rx_{packet} στις αντίστοιχες OpenFlow θύρες των switches. Επομένως, ο ρυθμός απώλειας πακέτων μπορεί να υπολογιστεί χρησιμοποιώντας την εξίσωση:
$$P_r(\text{Loss}) = \frac{Tx_{packet} - Rx_{packet}}{Tx_{packet}}$$
- Throughput ή network throughput: Αυτή η μέτρηση υπολογίζει το άθροισμα των εργασιών που ολοκληρώθηκαν σε μία μονάδα χρόνου μετά την εκτέλεση load balancing. Καθορίζει τον ρυθμό με τον οποίο γίνεται η υπολογιστική εργασία χρησιμοποιώντας μια τεχνική load balancing. Ο αλγόριθμος load balancing στοχεύει στην επίτευξη μεγαλύτερης αποτελεσματικότητας.
- Average response time – μέσος χρόνος απόκρισης: Μία από τις κύριες μετρήσεις των τεχνικών load balancing είναι ο μέσος χρόνος απόκρισης. Ορίζεται ως ο χρόνος που χρειάζεται για να ανακτήσει ο χρήστης τα αποτελέσματα ενός αιτήματος. Οι διάφορες μεταβλητές που επηρεάζουν τον χρόνο απόκρισης περιλαμβάνουν το εύρος ζώνης, τον αριθμό των χρηστών που έχουν πρόσβαση στο δίκτυο ταυτόχρονα, τον αριθμό των αιτημάτων και τον μέσο χρόνο επεξεργασίας. Ένας μεγάλος αριθμός αιτημάτων ανά δευτερόλεπτο πρέπει να υποβληθεί σε επεξεργασία για να λάβετε πιο γρήγορες απαντήσεις.
- Average end-to-end delay – μέσος χρόνος καθυστέρησης από άκρο σε άκρο: Η καθυστέρηση από άκρο σε άκρο μιας ροής είναι ο χρόνος που χρειάζεται για να φτάσουν όλα τα πακέτα στον κόμβο προορισμού. Αυτό υπολογίζεται με βάση τον τύπο:
$$T_{E2E} = T_{i\text{ End}} - T_{i\text{ Start}}$$
- Για το δίκτυο, η μέση καθυστέρηση από άκρο σε άκρο είναι το άθροισμα όλων των καθυστερήσεων από άκρο σε άκρο διαιρούμενο με τον συνολικό αριθμό ροών N_F .
$$Avg(T_{E2E}) = \sum_{i=1}^N T_{E2Ei} / N_F$$
- Resource utilization ratio (RU ration) – αναλογία χρήσης πόρων: Υπολογίζεται ως η αποτελεσματική χρήση του πόρου στην επεξεργασία του αιτήματος και ο στόχος είναι να βελτιστοποιηθεί ο πιο κρίσιμος πόρος για την επεξεργασία του αιτήματος. Οι πόροι μπορεί να είναι ο λόγος χρήσης εύρους ζώνης, η σύνδεση, η χρήση μνήμης και ο επεξεργαστής. Επιπλέον, η αναλογία χρήσης πόρων αντιπροσωπεύει ένα ποσοστό της συνολικής χωρητικότητας σύνδεσης.
- Overhead: Οποιοσδήποτε συνδυασμός πρόσθετου χρόνου “additional time”, εύρους ζώνης “bandwidth” ή οποιωνδήποτε άλλων απαιτούμενων παραγόντων για την ολοκλήρωση μιας δεδομένης εργασίας αναφέρεται ως γενικά έξοδα. Εάν τα πακέτα λαμβάνονται από έναν κόμβο πάνω από την χωρητικότά του Ex_{packet} , δημιουργείται γενική επιβάρυνση στον κόμβο. Διαφορετικά, εάν τα παρεληφθέντα

πακέτα βρίσκονται εντός εμβέλειας, δεν δημιουργείται overhead. Το overhead σε έναν κόμβο μπορεί να μετρηθεί χρησιμοποιώντας εξισώσεις:

$$1. Ex_{packet} = Rx_{packet} - Tx_{packet}$$

$$2. Ov_{packet} = \frac{Ex_{packet}}{Packet\ threshold}$$

- Transmission hop count (hop count) – ο αριθμός μεταπηδήσεων από έναν κόμβο σε έναν άλλον: Το πλήθος αναπηδήσεων μετάδοσης θεωρείται ένας από τους σημαντικούς παράγοντες για την προσέγγιση δρομολόγησης. Η πιθανότητα συμφόρησης – congestion μπορεί να αυξηθεί με μεγάλο αριθμό άλματος μετάδοσης. Αντίθετα, η πιθανότητα απώλειας πακέτων, packet loss, μπορεί να μειωθεί καθώς και η καθυστέρηση μετάδοσης σε παρόμοια κατάσταση δικτύου εάν το πακέτο που μεταδίδεται κατά μήκος της διαδρομής γίνεται με λιγότερες αναπηδήσεις. Ως αποτέλεσμα της προστασίας της παγκόσμιας τοπολογίας δικτύου του ελεγκτή SDN στη βάση δεδομένων, υπάρχει πρόβλεψη για πολλές μεταπηδήσεις μεταξύ ενός ζεύγους switch.
- Root Mean Square Error (RMSE) στους Servers: Είναι μια μετρική για τη μέτρηση της αποτελεσματικότητας στις τεχνικές load balancing. Ένα μικρότερο RMSE έχει καλύτερη απόδοση. Το RMSE θα είναι 0 εάν όλοι οι Servers είναι εξίσου φορτωμένοι.
- Flow completion time – ο χρόνος ολοκλήρωσης της ροής: Είναι η κύρια μέτρηση που χρησιμοποιείται για τον προσδιορισμό της απόδοσης μεταφοράς ροής στην περίπτωση μεγάλης ροής σε δίκτυα κέντρων δεδομένων. Μπορεί να οριστεί ως η διάρκεια που χρειάζεται για να ολοκληρωθεί μια μεταφορά αρχείου σε μια ροή. Οι προδιαγραφές QoS απαιτούν μια μεγάλη σειρά πόρων δικτύου για την ολοκλήρωση του χρόνου ροής. Το FCT μπορεί να περιλαμβάνει, μεταξύ άλλων, τους χρόνους εκτέλεσης και μετεγκατάστασης (δηλαδή, ο χρόνος που απαιτείται για τη μετακίνηση ενός διακόπτη είναι απαραίτητος για την αποτελεσματική λειτουργία ενός πρωτοκόλλου επιπέδου ελέγχου load balancing). Οι τεχνικές load balancing στοχεύουν στη μείωση του χρόνου ολοκλήρωσης της ροής στο ελάχιστο δυνατό.
- Types of traffic (EF, MF) – τύπος ροής κυκλοφορίας: Αυτό περιλαμβάνει τη διαφοροποίηση του τύπου της εισερχόμενης ροής κυκλοφορίας ως προς το εάν η μεγάλη ροή μακράς διάρκειας (δηλαδή, EF) ή η βραχεία ροή (δηλ., MF). Με βάση προηγούμενες μετρήσεις στα δίκτυα των κέντρων δεδομένων, βρέθηκε ότι το 80% των ροών (δηλαδή, τα MF) είναι κάτω από 10 KB και διήρκεσαν για λίγα χιλιοστά του δευτερολέπτου. Αντίθετα, το κορυφαίο 10% των μεγάλων ροών (δηλαδή, EF) αντιπροσωπεύει το μεγαλύτερο μέρος της κίνησης. Οποιαδήποτε επισκεψιμότητα υπερβαίνει ένα συγκεκριμένο όριο ανά μονάδα χρόνου (π.χ. 1 MBps) ταξινομείται ως EF. Δεδομένου του υψηλού επιπέδου EF στην κίνηση του δικτύου, ο αποτελεσματικός έλεγχος και η επαναδρομολόγηση τους μπορεί να οδηγήσει σε αυξημένη απόδοση του δικτύου SDN. Από την άλλη πλευρά, ο ανταγωνισμός μεταξύ MF και EF για περιορισμένους πόρους δικτύου δείχνει ότι τα MF δεν έχουν αρκετό εύρος ζώνης.

2.6 Σχετικές εργασίες

Σε μία εργασία εξισορρόπησης φορτίου στους συνδέσμους οι ερευνητές χρησιμοποίησαν ένα νευρωνικό δίκτυο για την εκμάθηση των δεδομένων. Αυτό έγινε διότι η κίνηση στο

δίκτυο είναι δυναμική και οι ερευνητές έκριναν απαραίτητο να πραγματοποιηθεί μία βαθιά εκμάθηση των μακροπρόθεσμων και βραχυπρόθεσμων δεδομένων. Έτσι προτείνανε έναν σενάριο εξισορρόπησης φορτίου που περιλάμβανε εξισορρόπηση φορτίου στους συνδέσμους, στον διακομιστή και ταξινόμηση κυκλοφορίας. Αυτή η μέθοδος παρέχει βελτιωμένη απόδοση, ελαχιστοποιεί τον χρόνο ολοκλήρωσης της ροής και ελαχιστοποιεί την απώλεια δεδομένων [30]. Σε μία άλλη εργασία κεντριοποιημένης αρχιτεκτονικής SDN οι ερευνητές ασχολήθηκαν στον τομέα της μηχανικής κυκλοφορίας – traffic engineering. Σε αυτήν την εργασία ο ελεγκτής παρακολουθεί περιοδικά τα στατιστικά των ροών κυκλοφορίας και την χρησιμοποίηση των συνδέσμων και όταν κάποιος σύνδεσμος είναι υπερχρησιμοποιημένος – over-utilised link και προκαλεί συμφόρηση – congestion και απώλεια πακέτων προσδιορίζεται ως σύνδεσμος συμφόρησης – bottleneck. Η επαναδρομολόγηση γίνεται μέσω της εναλλακτικής διαδρομής η οποία είναι ελαφρώς φορτωμένη. Το προτεινόμενο σχήμα μοντελοποιεί ένα Bayesian δίκτυο χρησιμοποιώντας την παρατηρούμενη χρήση θύρας και το υπολειπόμενο εύρος ζώνης για να αποφασίσει εάν η νέα εναλλακτική διαδρομή που υπολογίστηκε μπορεί να χειριστεί το νέο φορτίο ροής πριν από την εισαγωγή ροής, το οποίο με τη σειρά του μειώνει τη διάδοση της συμφόρησης. Τα αποτελέσματα της προσομοίωσης δείχνουν ότι όταν η κίνηση του δικτύου αυξάνεται η προτεινόμενη μέθοδος επαναδρομολογεί τις ροές και εξισορροπεί το φορτίο του δικτύου, που βελτιώνει σημαντικά την απόδοση του δικτύου και την ποιότητα υπηρεσίας – QoS [31]. Σε μία άλλη εργασία σχετική με το traffic engineering, οι ερευνητές προτείνανε μία μέθοδο με ετικέτες odd, και το ονομάσανε Oddlab. Αυτή η τεχνική δημιουργεί ένα heuristic – ευρετικό μοντέλο για αποτελεσματικό προγραμματισμό ροής και ανίχνευσης ελαττωματικών συνδέσεων χρησιμοποιώντας αποκλειστικά τα συγκεντρωμένα στατιστικά στοιχεία από το επίπεδο δεδομένων DCN, όπως το υπολειπόμενο εύρος ζώνης και τον αριθμό των εγκατεστημένων ροών ελεφάντων. Το Oddlab μπορεί να ελαχιστοποιήσει τον χρόνο ολοκλήρωσης της ροής, να μεγιστοποιήσει το εύρος ζώνης διχοτόμησης, να βελτιώσει τη χρήση του δικτύου και να αναγνωρίσει ελαττωματικές συνδέσεις με επαρκή ακρίβεια για τη βελτίωση της παραγωγικότητας DC [32].

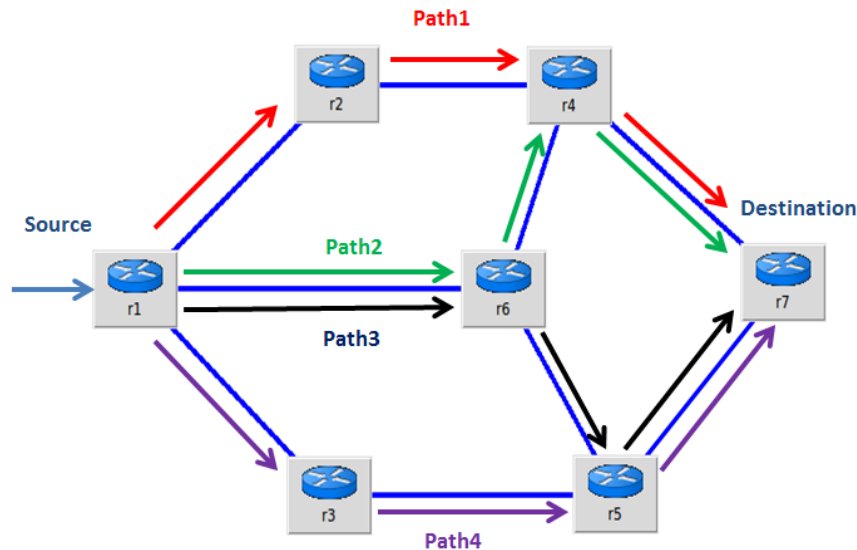
3 Αλγόριθμοι προληπτικής (proactive) εξισορρόπησης φορτίου

Στην ενότητα αυτή θα παρουσιαστούν τέσσερις βασικοί αλγόριθμοι που προτάθηκαν για εξισορρόπηση φορτίου σε δίκτυα SDN. Όλοι οι αλγόριθμοι λειτουργούν προληπτικά (proactive) κατανέμοντας την κυκλοφορία των δεδομένων σε μονοπάτια ίσου κόστους.

3.1 Ο αλγόριθμος ECMP

Η τεχνική ECMP (Equal Cost Multi Path) είναι μία από τις πιο κοινά χρησιμοποιημένες τεχνικές για την εξισορρόπηση φορτίου στους συνδέσμους – link load balancing. Τα περισσότερα από τα σημερινά κέντρα δεδομένων χρησιμοποιούν Spanning Tree ή ECMP για δρομολόγηση. Στο Spanning Tree, είναι βελτιστοποιημένο να επιλέγει μια μεμονωμένη διαδρομή για κάθε ζεύγος πηγής-προορισμού, όπως όλες οι επισκέψεις διασχίζουν σε ένα μόνο δέντρο που οδηγεί σε πολλούς αχρησιμοποίητους συνδέσμους. Αυτός ο αλγόριθμος δεν λειτουργεί καλά με κέντρα δεδομένων με δέντρο πολλαπλών ριζών. Για την καλύτερη αξιοποίηση αυτών των χαρακτηριστικών τοπολογιών και την πλήρη χρήση των πολλαπλών διαδρομών, υιοθετείται το ECMP. Το ECMP δρομολογεί τυχαία ροές μέσω των διαθέσιμων μονοπατιών ίσου κόστους με βάση την τιμή κατακερματισμού ορισμένων πεδίων ενός πακέτου. Το μειονέκτημα στον ECMP, είναι ότι ο hash αλγόριθμος λαμβάνει υπόψη την ισοδυναμία της διαδρομής και δεν μπορεί να προσαρμοστεί δυναμικά σύμφωνα με την κίνηση στην εκάστοτε διαδρομή. Αυτό το είδος κατανομής κίνησης στο ECMP επιχειρεί να εξισορροπήσει το φορτίο ως προς τον αριθμό των ροών σε διαφορετικές διαδρομές και όχι στον ρυθμό μετάδοσης bit. Καθώς υπάρχουν διαφορετικά μεγέθη ροών στο δίκτυο του κέντρου δεδομένων, με την χρησιμοποίηση του ECMP, αυτή η ομοιόμορφη κατανομή σύμφωνα με τον αριθμό των ροών, μπορεί να οδηγήσει σε συμφόρηση σε ορισμένες συνδέσεις, ενώ άλλες παραμένουν ανεπαρκώς αξιοποιημένες. Ο ECMP αλγόριθμος είναι ένας μηχανισμός προώθησης για την δρομολόγηση πακέτων σε πολλαπλές διαδρομές που έχουν ίσο κόστος για να γίνει η κατανομή του φόρτου ίση μέσα στο δίκτυο. Ο ECMP αλγόριθμος βασίζεται σε συναρτήσεις κατακερματισμού ή αλλιώς hash functions οι οποίες είναι stateless. Για να την δημιουργία hash επιλέγεται ένα σύνολο πεδίων συνήθως τα 5 κοινά είναι (source IP, destination IP, IP protocol, TCP/UDP source port, TCP/UDP destination port and IPv6 flow label), όλα τα πακέτα που ανήκουν στην ίδια TCP ροή κινούνται προς το ίδιο μονοπάτι – path, αυτό γίνεται για να αποφευχθεί η επανατοποθέτηση – reordering των πακέτων. Για κάθε πακέτο που προωθείται το switch παίρνει τις τιμές τις συνδυάζει και δημιουργεί το hash το οποίο το χρησιμοποιεί για να υπολογίσει το επόμενο βήμα. Για να υπολογίσει το switch το επόμενο βήμα ή next_hop διαιρεί το hash με τον αριθμό των διαδρομών που υπάρχουν στον εκάστοτε δρομολογητή. Για παράδειγμα $h \% n$, όπου n είναι ο αριθμός των εξόδων L διαδρομών στο ECMP αλγόριθμο. Αυτή η τιμή είναι μεταξύ 0 και $n - 1$, και αντιστοιχεί στο πλήθος των συνδέσμων στους οποίους θα γίνει η κατανομή του φόρτου – load balancing. Αυτή η τιμή είναι η επιλεγμένη σύνδεση ή διαδρομή για το εκάστοτε πακέτο. Ο αλγόριθμος ECMP κατακερματίζει την ροή – flow που εισέρχεται στο switch και την προωθεί στα μονοπάτια που ακολουθούν. Όταν φθάνει ένα πακέτο με πολλαπλές υποψήφιες διαδρομές, προωθείται σε αυτό που αντιστοιχεί σε έναν κατακερματισμό επιλεγμένων πεδίων “headers” του πακέτου ανάλογα με τον αριθμό των διαδρομών. Σε περίπτωση που η ροή είναι αρκετά μεγάλη και γίνεται κατακερματισμός υπάρχει πιθανότητα να υπάρχει μεγάλη σημαντική ανισορροπία στις ροές. Στον ECMP

αλγόριθμο παίρνει απόφαση το τοπικό switch και η απόφαση λαμβάνεται ανεξάρτητα, χωρίς να γνωρίζει για πιθανή συμφόρηση στα άλλα μελλοντικά switch μέσα στο δίκτυο. Η λειτουργία του αλγορίθμου ECMP παρουσιάζεται στην Εικόνα 3.1

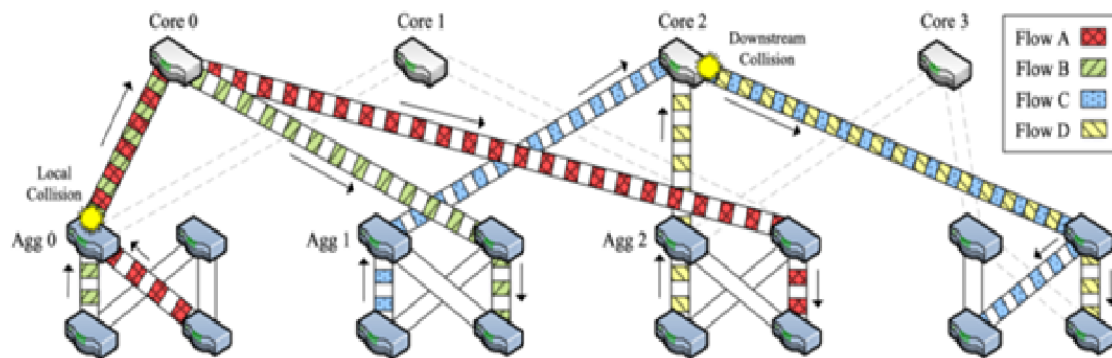


Εικόνα 3-1: Η λειτουργία του αλγορίθμου ECMP

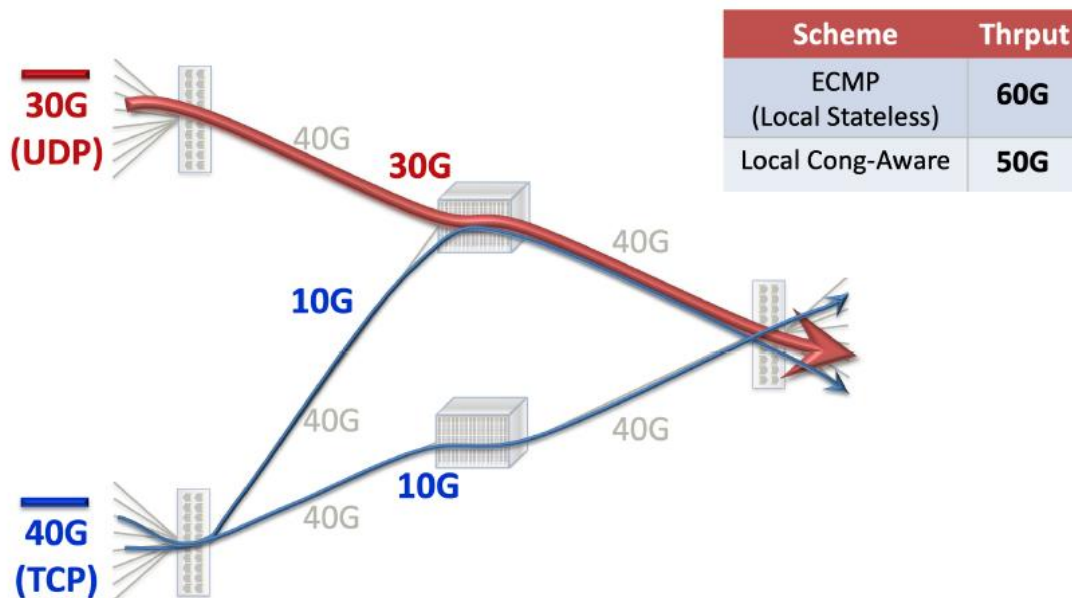
3.1.1 Προβλήματα του ECMP

Τα δυο κύρια προβλήματα του ECMP είναι οι συγκρούσεις ως αποτέλεσμα των συναρτήσεων hash, και η ασυμμετρία. Οι συναρτήσεις hash στέλνονται ανά ροές μεταξύ των switch. Ο περιορισμός του ECMP είναι όταν δύο ή περισσότερες ροές hash συγκρουστούνε μεταξύ τους στην είσοδο του switch και καταλήξουν μαζί στην ίδια έξοδο. Εκεί θα υπάρχει μία σύγκρουση, συμφόρηση - bottleneck μεταξύ των ροών. Αυτό αντικατοπτρίζεται στην Εικόνα 3-2. Μεταξύ του switch Agg0 και Core0 υπάρχει συμφόρηση των ροών A και B, όπως επίσης μεταξύ Core2 και Agg3 για τις ροές C και D. Αυτές οι συγκρούσεις θα μπορούσαν να είχαν αποφευχθεί αν υπήρχε ενημέρωση της συμφόρησης ή απλά πρόβλεψη. Το άλλο σημαντικό πρόβλημα του ECMP είναι η ασυμμετρία. Αυτό συμβαίνει κατά κύριο λόγο όταν υπάρχει αποτυχία – failure στις συνδέσεις μεταξύ των switch. Σε αυτήν την περίπτωση όλος ο όγκος

της ροής θα αναδιανεμηθεί στις εναπομείναντες συνδέσεις όπου και θα υπάρξει πιθανή συμφόρηση.



Εικόνα 3-2: Το πρόβλημα των συγκρούσεων.



Εικόνα 3-3: Το πρόβλημα της ασυμμετρίας του ECMP.

3.2 Ο αλγόριθμος Hedera

Ο αλγόριθμος Hedera είναι ένας αλγόριθμος δυναμικού προγραμματισμού της εισερχόμενης ροής στο κέντρο δεδομένων ο οποίος προτάθηκε και στόχευε να επιλύσει στο φαινόμενο της ασυμμετρίας του φορτίου που προκαλείται από τις πολλαπλές ροές που χαρτογραφούνται στην ίδια διαδρομή. Το Hedera συλλέγει πληροφορίες ροής, υπολογίζει της διαδρομές που δεν θα γίνουν συγκρούσεις και με βάση τους υπολογισμούς δίνει εντολές στα switches να επαναδρομολογούν την κυκλοφορία ανάλογα. Το Hedera δρομολογεί βραχύβιες ροές, short-lived flows, με ECMP ενώ ροές μεγάλης διάρκειας, long-lived flows, με δυναμική προσέγγιση. Όταν ανιχνεύει μια μακρόβια ροή, το Hedera εκτιμά τη φυσική ζήτηση της ροής και υπολογίζει μια κατάλληλη διαδρομή για τη ροή. Ο στόχος του αλγορίθμου είναι να μεγιστοποιήσει την συνολική χρήση του δικτύου με τέτοιο τρόπο που να μην υπάρχει

αντίκτυπο στις ενεργές ροές. Το δυναμικό και κεντρικό σχήμα προγραμματισμού είναι μόνο για μακροχρόνια ροή. Το Hedera μπορεί να επιτύχει τον στόχο της μεγιστοποίησης της συνολικής χρήσης του δικτύου προγραμματίζοντας δυναμικά τις μακροχρόνιες ροές αλλά αγνοώντας τη ζήτηση των βραχύβιων ροών. Για παράδειγμα, μια βραχύβια ροή μπορεί να δρομολογηθεί σε μια πλήρως φορτωμένη ζεύξη, και έτσι προκαλείται υπερβολική καθυστέρηση. Το Hedera επίσης έχει συνολική εικόνα της δρομολόγησης και των απαιτήσεων κυκλοφορίας στο δίκτυο, και με αυτόν τον τρόπο το σύστημα προγραμματισμού βλέπει τα σημεία συμφόρησης που δεν μπορούν να αλλάξουν οι τοπικοί ελεγκτές. Το Hedera προσφέρει αυτές τις βελτιώσεις εύρους ζώνης σε ένα δίκτυο με μέτρια επιβάρυνση ελέγχου και υπολογισμού. Η τεχνική αυτή είναι κατάλληλη μόνο για συγκεκριμένες τοπολογίες και έχει έλλειψη από ευελιξία και επεκτασιμότητα.[22][26]

3.3 Ο αλγόριθμος PureSDN

Ο αλγόριθμος PureSDN είναι μία τεχνική δρομολόγησης η οποία δημιουργήθηκε από τον H. Machi και τον L.Cheng[45]. Ο αλγόριθμος PureSDN περιλαμβάνει δύο μοντέλα, το network awareness μοντέλο και το network monitor μοντέλο. Το μοντέλο network awareness χρησιμοποιείται για την ανακάλυψη πληροφορίας του δικτύου, ενώ το network monitor χρησιμοποιείται για να συλλέγει πληροφορίες κίνησης όπως εύρος ζώνης των συνδέσμων. Ο PureSDN βασίζεται στο bandwidth “εύρος ζώνης” ή στο hop “αναπήδηση” για να επιτύχει τον αλγόριθμο μικρότερης διαδρομής – shortest path forwarding. Ο αλγόριθμος PureSDN χρησιμοποιεί επίσης και ένα άλλο μοντέλο το οποίο λέγεται Networkx. Αυτό το μοντέλο το χρησιμοποιεί ο αλγόριθμος όταν θέλει να υπολογίσει την μικρότερη διαδρομή με βάση την αναπήδηση. Το Networkx δημιουργεί έναν γράφο G με τις διαδρομές και τις έχει ταξινομημένες από τη συντομότερη στην μακρινότερη. Όταν ο αλγόριθμος θέλει να υπολογίσει τη διαδρομή με βάση το εύρος ζώνης θα πάρει στατιστικά πληροφορίες από τις πόρτες μέσω του πρωτοκόλλου OpenFlow και υπολογίζει την ταχύτητα της εκάστοτε πόρτας.

3.4 Ο αλγόριθμος Conga

Η τεχνική CONGA που το όνομα της προέρχεται από τα αρχικά της φράσης Congestion Aware Balancing είναι μία τεχνική που στην οποία ο διακομιστής είναι ενήμερος για το αν υπάρχει συμφόρηση ή όχι στο κάθε switch. Το κάθε switch ανατροφοδοτεί τον διακομιστή με την τρέχουσα κατάσταση. Η τεχνική χρησιμοποιεί τον αλγόριθμο DRE (Discount Rate Estimator) για να εκτιμήσει τον φόρτο σε κάθε σύνδεσμο – link, και μετά ο κύριος κόμβος κάνει εξισορρόπηση του φόρτου στο δίκτυο σύμφωνα με την συμφόρηση στους συνδέσμους. Για να πετύχει βέλτιστη κατανομή ροής σε μια ασύμμετρη τοπολογία, χρησιμοποιεί τα flowlets (λεπτομέρεια σε επίπεδο πακέτου) και στη συλλογή της ανατροφοδότησης συμφόρησης από τους μεταγωγείς. Το flowlet είναι ένα μικρό κομμάτι μιας ροής που αποστέλλεται δυναμικά στις θύρες του μεταγωγέα ανάλογα με την ανατροφοδότηση συμφόρησης. Αυτή η διαδικασία επηρεάζει το FCT (flow completion time) των βραχύβιων ροών και την αναδιάταξη των πακέτων στην αναδρομολόγηση flowlet στην ανίχνευση συμφόρησης ή αποτυχίας. Το Conga έχει γρήγορη αντίδραση όταν υπάρχει πολύ υψηλή κυκλοφορία στα κέντρα δεδομένων, αλλά δεν μπορεί να γίνει εύκολα επεκτάσιμη.

3.5 Σύνοψη

Ο ECMP αλγόριθμος είναι ένας link load balancing αλγόριθμος. Ο ECMP δρομολογεί τυχαία ροές μέσω των διαθέσιμων μονοπατιών ίσου κόστους με βάση την τιμή της συνάρτησης κατακερματισμού των πεδίων ενός πακέτου. Τα πεδία που χρησιμοποιούνται για την συνάρτηση κατακερματισμού συνήθως είναι: source IP, destination IP, IP protocol, TCP/UDP source port, TCP/UDP destination port, IPv6 flow label. Τα κύρια προβλήματα του ECMP είναι: η ασυμμετρία, και οι συγκρούσεις των συναρτήσεων κατακερματισμού. Ο αλγόριθμος Hedera συλλέγει πληροφορίες ροής, υπολογίζει της διαδρομές που δεν θα γίνουν συγκρούσεις και προωθεί τις ροές ανάλογα. Ο στόχος του είναι να μεγιστοποιήσει την συνολική χρήση του δικτύου με τρόπο που να μην υπάρχει αντίκτυπο στις ενεργές ροές. Ο PureSDN είναι βασισμένος στο εύρος ζώνης – bandwidth και στην αναπήδηση – hop για να πετύχει την μικρότερη διαδρομή. Ο αλγόριθμος δημιουργεί έναν γράφο G, με τις διαδρομές και τις έχει ταξινομημένες από την συντομότερη στην μακρινότερη. Στον αλγόριθμο Conga ανατροφοδοτείτε ο διακομιστής με την τρέχουσα κατάσταση του κάθε switch και ο αλγόριθμος δρομολογεί τις ροές στο switch με την ελάχιστη συμφόρηση. Τα χαρακτηριστικά των αλγορίθμων συνοψίζονται στον Πίνακα 3-1.

Πίνακας 3-1: Σύγκριση βασικών αλγορίθμων

Αλγόριθμος	Μονάδα πληροφορίας	Μηχανισμός
ECMP	Flow	Use hash function to select a random path over multiple “best paths”
Hedera	Flows	Detects large flows and compute good paths for them
PureSDN	Paths	Network awareness
Conga	Flowlet	Distributed congestion-aware load balancing

4 Πειραματική μελέτη

Για την πειραματική αξιολόγηση των αλγορίθμων χρησιμοποιήθηκε το περιβάλλον εξομοίωσης Mininet. Το Mininet είναι ένας εξομοιωτής δικτύου που δημιουργεί δίκτυα από εικονικούς κόμβους, switches, ελεγκτές και συνδέσμους. Τα εικονικά switches που δημιουργούνται υποστηρίζονται από το OpenFlow, κάτι που επιτρέπει την εξομοίωση δικτύων SDN. Η δυνατότητα που παρέχει το Mininet να εκτελεί πραγματικό κώδικα σε εικονικό δίκτυο που τρέχει σε έναν απλό υπολογιστή το καθιστά ιδανικό για ερευνητικούς και πειραματικούς σκοπούς. Για την δοκιμή του Mininet και του Ryu χρησιμοποιήσαμε virtual machine με λειτουργικό σύστημα Ubuntu 20.05. Εκτελέσαμε την εικονική μηχανή στο VMWare Workstation 16 player. Στην εικονική μηχανή παρέχονται 6.1 GB μνήμης και 4 επεξεργαστές. Ακολουθεί μια σύντομη παρουσίαση του Mininet, του ελεγκτή Ryu, και λεπτομερής περιγραφή του πλαισίου αξιολόγησης των αλγορίθμων.

4.1 Το Mininet

Το Mininet είναι ένα εργαλείο εξομοίωσης δικτύων ανοιχτού κώδικα. Έχει αναπτυχθεί αποκλειστικά για να υποστηρίζει τα SDN δίκτυα, επίσης επιτρέπει την γρήγορη πρωτοτυποποίηση μεγάλων δικτύων σε έναν απλό υπολογιστή. Μπορεί να δημιουργήσει επεκτάσιμα δίκτυα SDN χωρίς να χρησιμοποιεί μεγάλες απαιτήσεις σε μηχανισμούς εικονικοποίησης, όπως διαδικασίες και χώρους ονομάτων δικτύων. Αυτές οι δυνατότητες του επιτρέπουν να δημιουργεί, να αλληλοεπιδρά, να προσαρμόζει και να μοιράζεται τα πρωτότυπα με μεγάλη ταχύτητα. Τα προγράμματα που εκτελούνται μπορούν να στείλουν πακέτα μέσω μιας πραγματικής διεπαφής Ethernet, με δεδομένη ταχύτητα και καθυστέρηση σύνδεσης. Τα πλεονεκτήματα του Mininet είναι:

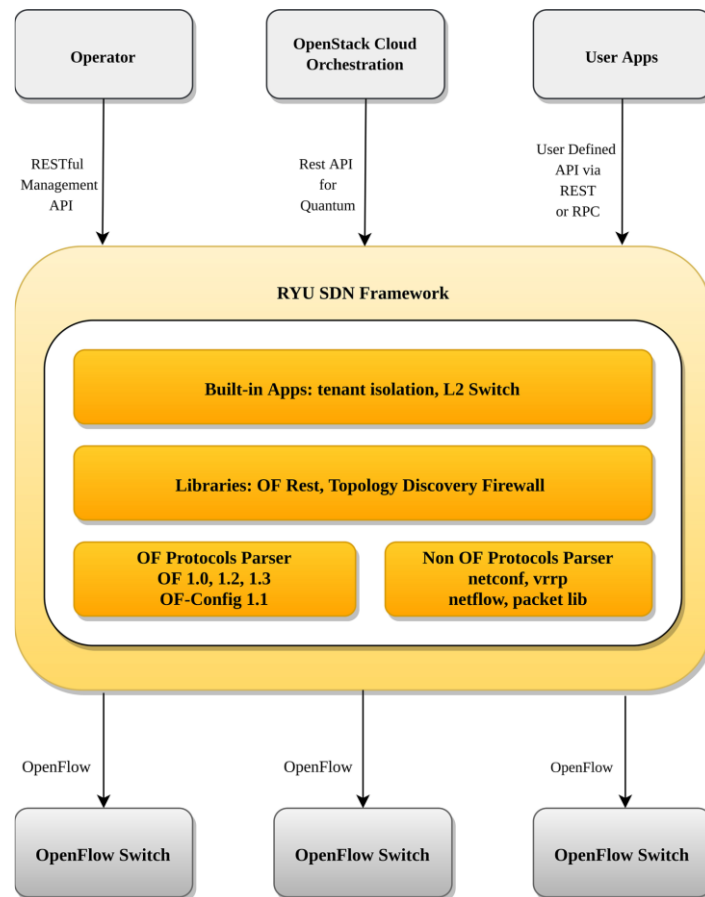
- Είναι γρήγορο - η εκκίνηση ενός απλού δικτύου διαρκεί μόλις λίγα δευτερόλεπτα. Αυτό σημαίνει ότι ο βρόχος run-edit-debug μπορεί να είναι πολύ γρήγορος.
- Ο κάθε χρήστης έχει την δυνατότητα να δημιουργήσει τις δικές του τοπολογίες και να τις προσομοιώσει.
- Μπορεί να εκτελέσει όλα τα προγράμματα που είναι στα Linux.
- Μπορεί να προσαρμόσει την προώθηση πακέτων: Οι διακόπτες του Mininet μπορούν να προγραμματιστούν χρησιμοποιώντας το πρωτόκολλο OpenFlow
- Μπορεί να εγκατασταθεί σε όλες τις πλατφόρμες, υπολογιστές και virtual machines.
- Είναι εύχρηστο καθώς ο χρήστης μπορεί να τρέξει τον κώδικα στο Mininet την ώρα που θα τον κατασκευάσει καθώς επίσης και να τον μοιραστεί.
- Μπορείτε να το χρησιμοποιήσετε εύκολα: μπορείτε να δημιουργήσετε και να εκτελέσετε πειράματα Mininet γράφοντας απλά (ή σύνθετα εάν χρειάζεται) σενάρια Python.
- Το Mininet είναι ένα έργο ανοιχτού κώδικα και βρίσκεται υπό ενεργό ανάπτυξη.

4.2 Ο ελεγκτής Ryu

Υπάρχει μία πληθώρα από ελεγκτές στη βιομηχανία με διαφορετικά χαρακτηριστικά. Κατά καιρούς, έχουν υλοποιηθεί διάφοροι για γενικότερη χρήση ή για συγκεκριμένες δικτυακές υλοποιήσεις, αλλά οι περισσότεροι είτε δεν υποστηρίζονται πλέον, είτε η εξέλιξη τους επικεντρώθηκε μόνο σε περιορισμένες λειτουργίες για συγκεκριμένες ανάγκες μέσα στο

δίκτυο. Η επιλογή ενός ελεγκτή ως του καλύτερου είναι αδύνατο να γίνει διότι γι' αυτό παίζουν ρόλο οι ανάγκες μας και η υλοποίηση του δικτύου. Οι πιο γνωστοί ελεγκτές ανοικτού κώδικα είναι οι: ο NOX, ο POX, ο Ryu, ο Floodlight, ο ONOS, ο OpenDaylight [34][35]. Στα πλαίσια αυτής της εργασίας επιλέχθηκε ο ελεγκτής Ryu καθώς διαθέτει δομικά στοιχεία λογισμικού, διεπαφές προγραμματισμού εφαρμογών (API), που διευκολύνουν τους προγραμματιστές να δημιουργήσουν τις δικές τους εφαρμογές. Ο RYU είναι ένα λογισμικό ελεγκτή SDN που βασίζεται σε εξαρτήματα ανοικτού κώδικα και αναπτύχθηκε από τη Nippon Telegraph and Telephone (NTT). Το RYU είναι ένας ελεγκτής SDN πλήρως υλοποιημένος στην Python. Το όνομα RYU προέρχεται από μια ιαπωνική λέξη που σημαίνει «ροή», που είναι ένας ιαπωνικός δράκος, ένας από τους θεούς του νερού. Διαχειρίζεται τον έλεγχο «ροής» για να ενεργοποιήσει την έξυπνη δικτύωση. Ο ελεγκτής RYU παρέχει στοιχεία λογισμικού με καλά καθορισμένες διεπαφές προγραμμάτων εφαρμογών (API) που διευκολύνουν τους προγραμματιστές να δημιουργήσουν νέες εφαρμογές διαχείρισης και ελέγχου δικτύου. Το RYU υποστηρίζει διάφορα πρωτόκολλα για τη διαχείριση συσκευών δικτύου, όπως OpenFlow, Netconf, OF-config, κ.λπ. Σχετικά με το OpenFlow, το RYU υποστηρίζει πλήρως τις επεκτάσεις 1.0, 1.2, 1.3, 1.4, 1.5 και Nicira.

Η αρχιτεκτονική του ελεγκτή RYU SDN που φαίνεται πιο κάτω στην Εικόνα 4-1. Ο ελεγκτής RYU SDN έχει τρία επίπεδα. Το ανώτερο επίπεδο αποτελείται από επιχειρηματικές και δικτυακές εφαρμογές λογικής γνωστές ως επίπεδο εφαρμογής. Το μεσαίο επίπεδο αποτελείται από υπηρεσίες δικτύου γνωστές ως επίπεδο ελέγχου ή πλαίσιο SDN και το κάτω επίπεδο αποτελείται από φυσικές και εικονικές συσκευές που είναι γνωστές ως επίπεδο υποδομής. Το μεσαίο στρώμα φιλοξενεί API με βόρεια και νότια API. Ο ελεγκτής εκθέτει ανοιχτά API με κατεύθυνση προς τον βορρά, όπως ένα Restful management API, REST, API για Quantum, API καθορισμένο από το χρήστη μέσω REST ή RPC, που χρησιμοποιούνται από εφαρμογές. Το RYU παρέχει μια ομάδα στοιχείων όπως OpenStack Quantum, Firewall, OFREST κ.λπ. χρήσιμα για εφαρμογές SDN. Ο στόχος αυτών των εφαρμογών είναι να συλλέξουν ευφυΐα δικτύου χρησιμοποιώντας έναν ελεγκτή, να εκτελέσουν αναλυτικά στοιχεία εκτελώντας αλγόριθμους και στη συνέχεια να ενορχηστρώσουν τους νέους κανόνες χρησιμοποιώντας τον ελεγκτή. Η διεπαφή νότιας κατεύθυνσης είναι ικανή να υποστηρίξει πολλαπλά πρωτόκολλα όπως OpenFlow, Netconf, OF-config, κ.λπ. Το RYU χρησιμοποιεί το OpenFlow για να αλληλοεπιδρά με το επίπεδο προώθησης (διακόπτες και δρομολογητές) για να τροποποιήσει τον τρόπο με τον οποίο το δίκτυο θα χειρίζεται τις ροές κυκλοφορίας. Έχει δοκιμαστεί και πιστοποιηθεί για να λειτουργεί με πολλούς διακόπτες OpenFlow, συμπεριλαμβανομένου του OpenvSwitch και προτάσεων από τις εταιρίες Centec, Hewlett Packard, IBM και NEC [33].



Εικόνα 4-1: Η αρχιτεκτονική του ελεγκτή Ryu

4.3 Το πλαίσιο αξιολόγησης των αλγορίθμων εξισορρόπησης φορτίου

Η υλοποίηση βασίστηκε στον κώδικα του Huangmachi από το repository [45].

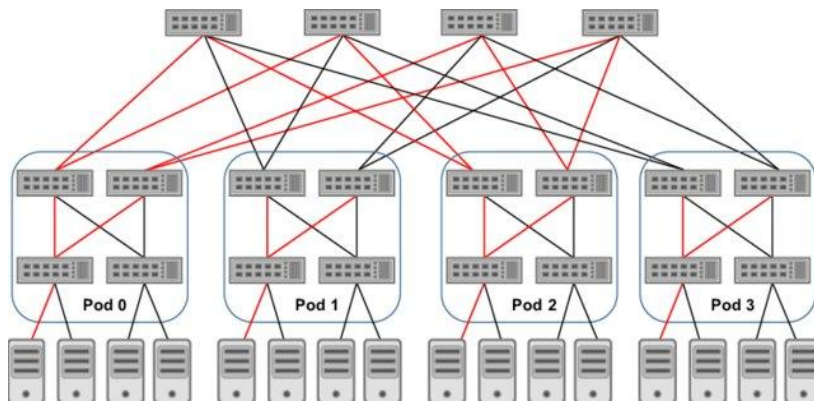
4.3.1 Εργαλεία λήψης μετρήσεων (Iperf -ping)

Το Iperf είναι ένα εργαλείο για την μέτρηση του εύρους ζώνης μεταξύ των κόμβων σε ένα δίκτυο. Για την μέτρηση χρησιμοποιείται ένας server και ένας client. Αυτοί οι δύο δημιουργούν ένα μία σύνδεση μεταξύ τους και στέλνουν μεταξύ τους δεδομένα για να γίνει η μέτρηση. Και τα δύο μέρη της εφαρμογής βρίσκονται στο ίδιο πακέτο και η επιλογή γίνεται κατά την εκκίνηση. Το πρόγραμμα διαθέτει πολλές παραμέτρους σχετικά με τον χρονισμό των συνδέσεων και υποστηρίζει τα πρωτοκόλλα TCP, UDP σε IPv4 και IPv6.[36]

4.3.2 Τοπολογίες και μοτίβα κυκλοφορίας

Το δίκτυο το οποίο μελετήσαμε, βασίστηκε σε τοπολογίες FatTree που παρουσιάστηκαν στην ενότητα 2.2.1. Συγκεκριμένα χρησιμοποιήθηκαν οι τοπολογίες 4-ary FatTree και το 8-ary FatTree, ο αλγόριθμος είναι ο ECMP. Με την εντολή Iperf έχει την δυνατότητα ο χρήστης να μετρήσει την απόδοση – “throughput” μεταξύ των κόμβων σε ένα δίκτυο. Για να γίνει η μέτρηση χρησιμοποιείται ένας server και ένας client. Αυτοί δημιουργούν μία σύνδεση μεταξύ τους και ο ένας στέλνει δεδομένα στον άλλον για να πραγματοποιηθεί η μέτρηση. Μπορεί να υποστηρίξει τα πρωτόκολλα TCP, UDP. Το 4-ary FatTree, όπως φαίνεται και στην Εικόνα 4-2, είναι μία τοπολογία η οποία είναι δομημένη με τέτοιο τρόπο να ελαχιστοποιεί την

καθυστέρηση – “latency” και να μεγιστοποιεί το εύρος ζώνης – “bandwidth” στα κέντρα δεδομένων. Ο αριθμός 4 ή 6 ή 8 είναι πόσα pods ή πόσο πλατύ – “Fat”, θα είναι το δίκτυο.



Εικόνα 4-2: Το δίκτυο FatTree για $k=4$

Μοτίβα κυκλοφορίας: Για τη δημιουργία κυκλοφορίας χρησιμοποιείται το εργαλείο iperf (ή ring) και παράγονται οι παρακάτω τέσσερις τύποι κυκλοφορίας. Το stag_0.2_0.3 σημαίνει 20% κυκλοφορία στο ίδιο edge switch, 30% στο διαφορετικό switch στο ίδιο pod, και το 50% μεταξύ των διαφορετικών pod. Το stag_0.3_0.3 σημαίνει 30% κυκλοφορία στο ίδιο edge switch, 30% στο διαφορετικό switch στο ίδιο pod, και το 40% μεταξύ των διαφορετικών pod. Το stag_0.4_0.3 σημαίνει 40% κυκλοφορία στο ίδιο edge switch, 30% στο διαφορετικό switch στο ίδιο pod, και το 30% μεταξύ των διαφορετικών pod. Και τέλος το stag_0.5_0.3 σημαίνει 50% κυκλοφορία στο ίδιο edge switch, 30% στο διαφορετικό switch στο ίδιο pod, και το 20% μεταξύ των διαφορετικών pod.

4.3.3 Σενάρια, Μετρικές, Παράμετροι

Αφού ορίσουμε την τοπολογία του δικτύου και τον τύπο της κυκλοφορίας καθορίζουμε τους αλγορίθμους που θα χρησιμοποιηθούν για την εξισορρόπηση φορτίου. Συγκεκριμένα επιλέχθηκαν οι αλγόριθμοι ECMP, Hedera, PureSDN. Για την συγκριτική τους αξιολόγηση επιλέξαμε ως μετρικές:

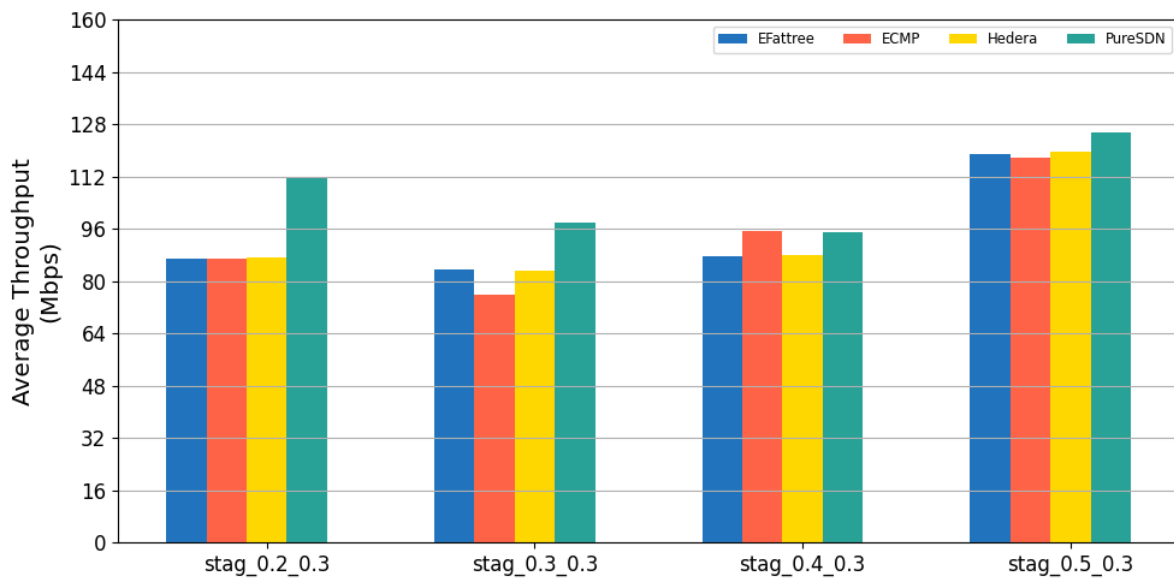
- Ρυθμοαπόδοση (throughput)
- First packet round trip delay
- Μέση διακύμανση (mean deviation)
- Απώλεια πακέτων (packet loss)
- Ο χρόνος καθυστέρησης μετ' επιστροφή (average packet round-trip delay)
- Το κάθε πείραμα (τοπολογία - αλγόριθμος) είχε διάρκεια 600 sec και λήφθηκε ο μέσος όρος για την κάθε μετρική.

4.4 Αποτελέσματα

4.4.1 Ρυθμοαπόδοση (Throughput)

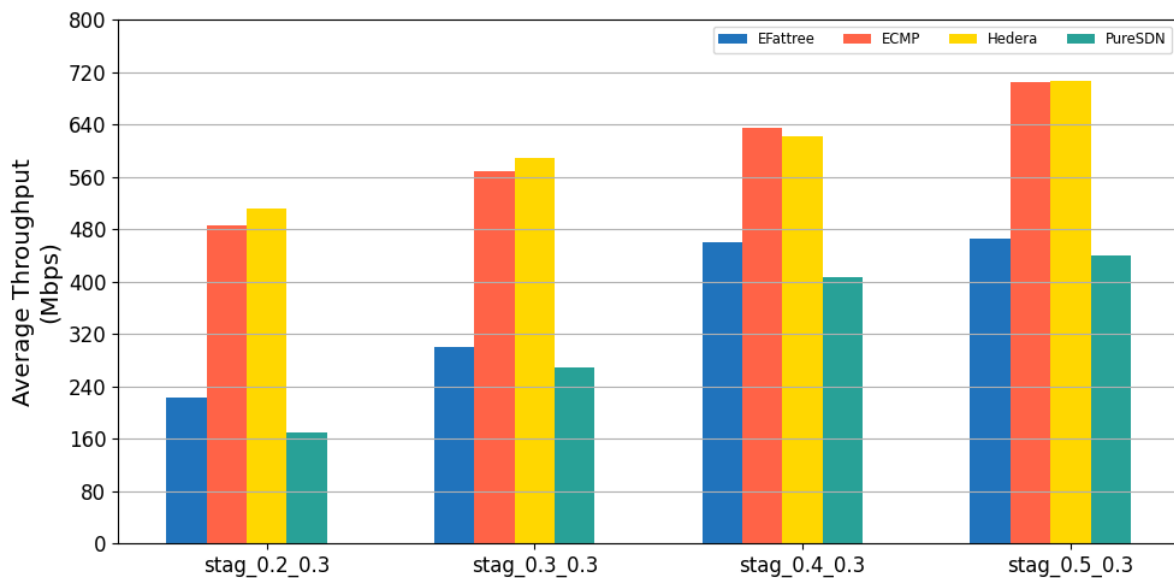
Το throughput είναι η πραγματική ποσότητα δεδομένων που μεταδίδονται με επιτυχία μέσω μίας σύνδεσης στο δίκτυο σε μία δεδομένη χρονική περίοδο. Είναι ο πραγματικός ρυθμός μετάδοσης και μετριέται σε bits per second (bps)[44]. Στις γραφικές παραστάσεις που

ακολουθούν βλέπουμε τα τέσσερα διαφορετικά μοτίβα κυκλοφορίας της πληροφορίας.



Εικόνα 4-3: Average throughput σε τοπολογία FatTree k = 4

Στην Εικόνα 4-3 βλέπουμε το average throughput για $k = 4$. Αυτό που μπορούμε να παρατηρήσουμε είναι ο αλγόριθμος PureSDN υπερσχύει σε όλα τα μοτίβα κυκλοφορίας και έχει την μέγιστη τιμή στο μοτίβο 0.5_0.3. Επίσης όσο αυξάνεται η κυκλοφορία στα switch η απόδοση αυξάνεται σε όλα τα μοτίβα. Την μικρότερη τιμή την έχει ο ECMP για το μοτίβο 0.3_0.3



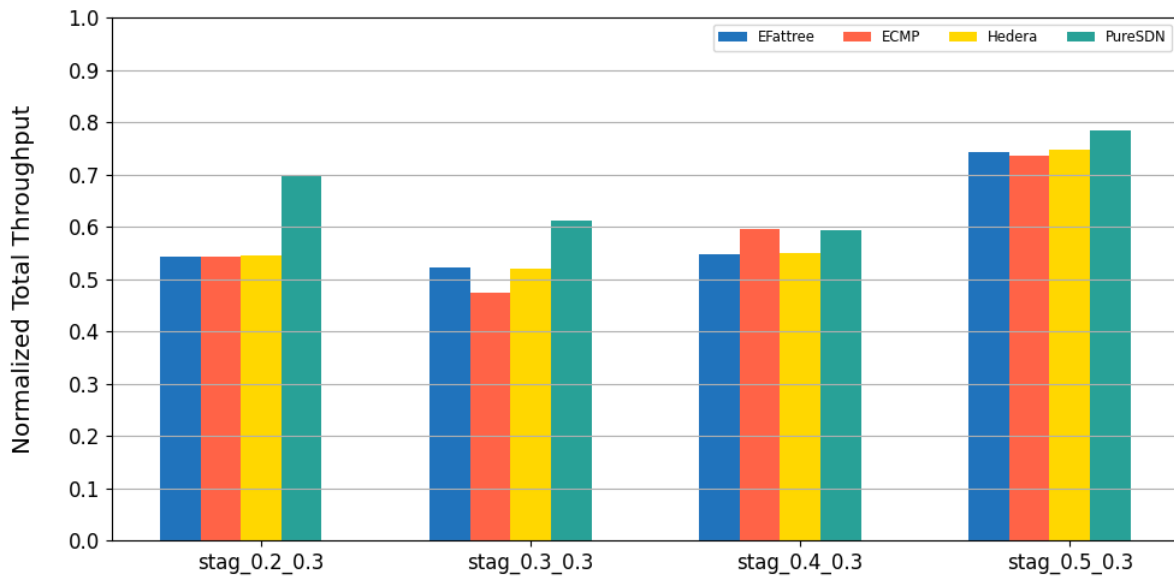
Εικόνα 4-4: Average throughput σε τοπολογία FatTree k = 8

Στην Εικόνα 4-4 βλέπουμε το average throughput για $k = 8$. Αυτό που παρατηρούμε είναι ότι το throughput αυξάνεται όσο αυξάνεται η κίνηση στα edge switches τόσο αυξάνεται και η απόδοση όπως για $k = 4$. Για τις 2 πρώτες περιπτώσεις κυκλοφορίας την καλύτερη απόδοση παρουσιάζει ο αλγόριθμος Hedera, μετά το ECMP, τρίτο το EFatTree και τελευταίος ο PureSDN. Στην τρίτη περίπτωση καλύτερος είναι το ECMP και μετά ακολουθεί το Hedera.

Στην τελευταία καλύτερη απόδοση έχει ο Hedera. Την μικρότερη απόδοση την έχει ο PureSDN στην πρώτη περίπτωση και την μεγαλύτερη την έχει ο PureSDN στην τελευταία.

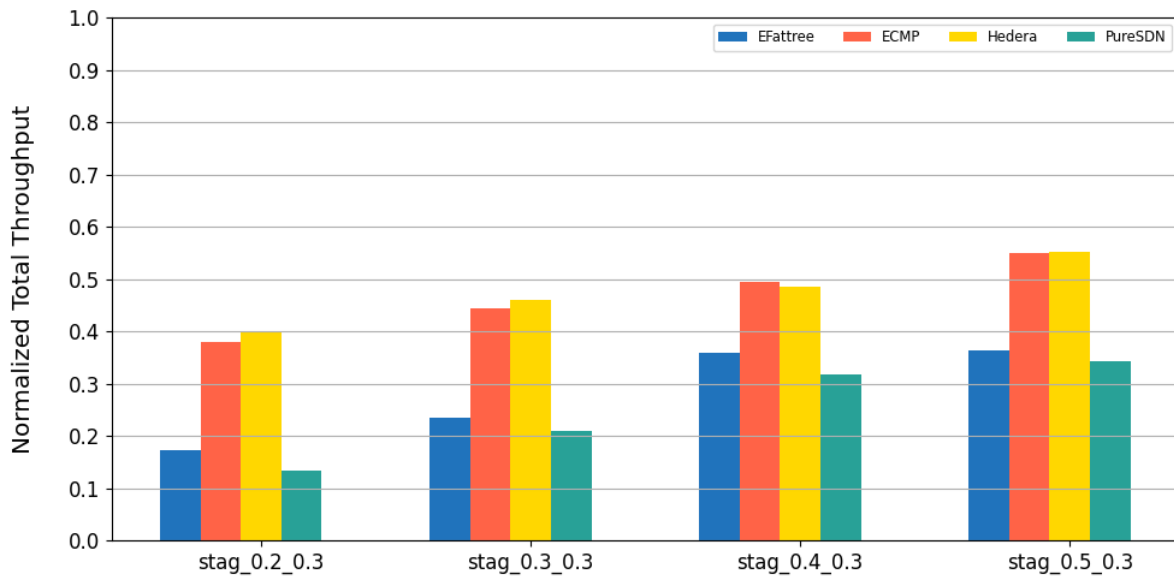
4.4.2 Normalised total throughput

Η κανονικοποιημένη απόδοση – normalised total throughput ορίζεται ως ο λόγος της πραγματικής απόδοσης ενός δικτύου προς τη μέγιστη δυνατή απόδοση, λαμβάνοντας υπόψη παράγοντες όπως το μέγεθος του δικτύου, τα μεγέθη buffer και το προσφερόμενο φορτίο. Βοηθά στη σύγκριση της απόδοσης διαφορετικών διαμορφώσεων δικτύου λαμβάνοντας υπόψη αυτές τις μεταβλητές.



Εικόνα 4-5: Normalised total throughput $k = 4$

Στην εικόνα 4-5 βλέπουμε την κανονικοποιημένη απόδοση για $k=4$. Αυτό που παρατηρούμε είναι ότι η απόδοση κυμαίνεται συνολικά σε όλες τις περιπτώσεις από 10% έως 80%. Την μέγιστη απόδοση την έχουμε για τον αλγόριθμο PureSDN για την περίπτωση stag_0.5_0.3. Την δεύτερη μεγαλύτερη απόδοση την έχουμε επίσης για τον PureSDN για την περίπτωση stag_0.2_0.3. Για τις περιπτώσεις stag_0.3_0.3 και stag_0.4_0.3 έχουμε επίσης μεγαλύτερη απόδοση με τον αλγόριθμο PureSDN αλλά η τιμή του είναι μικρότερη. Επίσης όσο αυξάνεται η κυκλοφορία στο ίδιο switch τόσο αυξάνεται και η απόδοση.

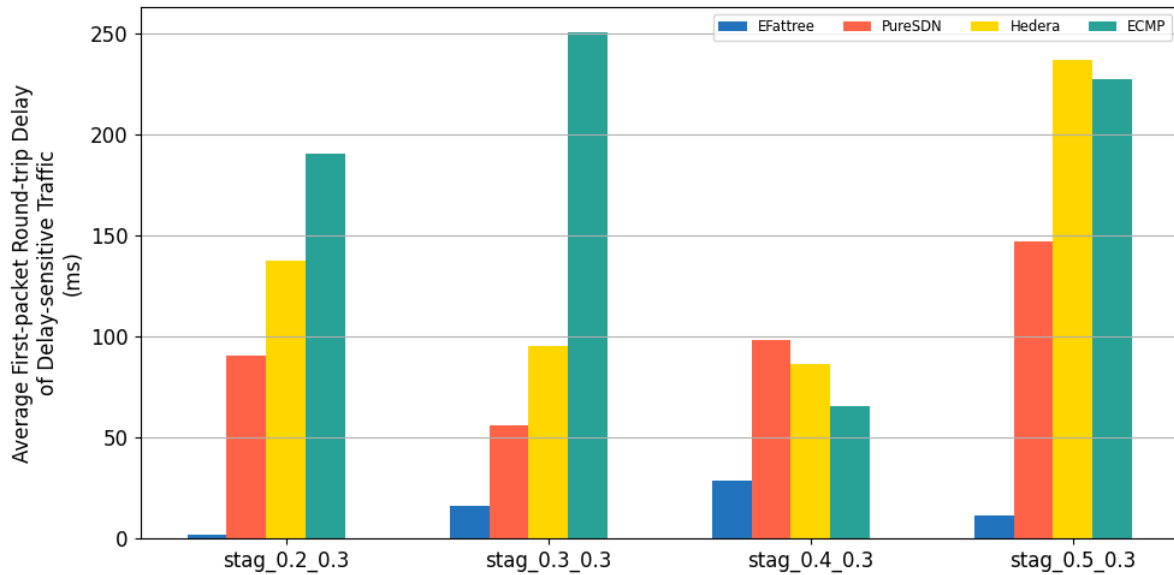


Εικόνα 4-6: Normalised total throughput $k = 8$

Στην εικόνα 4-6 παρατηρούμε το ίδιο μοτίβο όπως πριν για $k=4$. Όσο αυξάνεται η κυκλοφορία στο ίδιο switch τόσο αυξάνεται και το throughput. Το throughput κυμαίνεται από 10% έως 60% για όλες τις περιπτώσεις. Την μεγαλύτερη τιμή την βρίσκουμε για την περίπτωση stag_0.5_0.3 για τους αλγόριθμους ECMP και Hedera. Την μικρότερη τιμή την βρίσκουμε για τον αλγόριθμο PureSDN στην περίπτωση stag_0.2_0.3 και επίσης παρατηρούμε ότι ο PureSDN έχει σε όλες τις περιπτώσεις την μικρότερη τιμή.

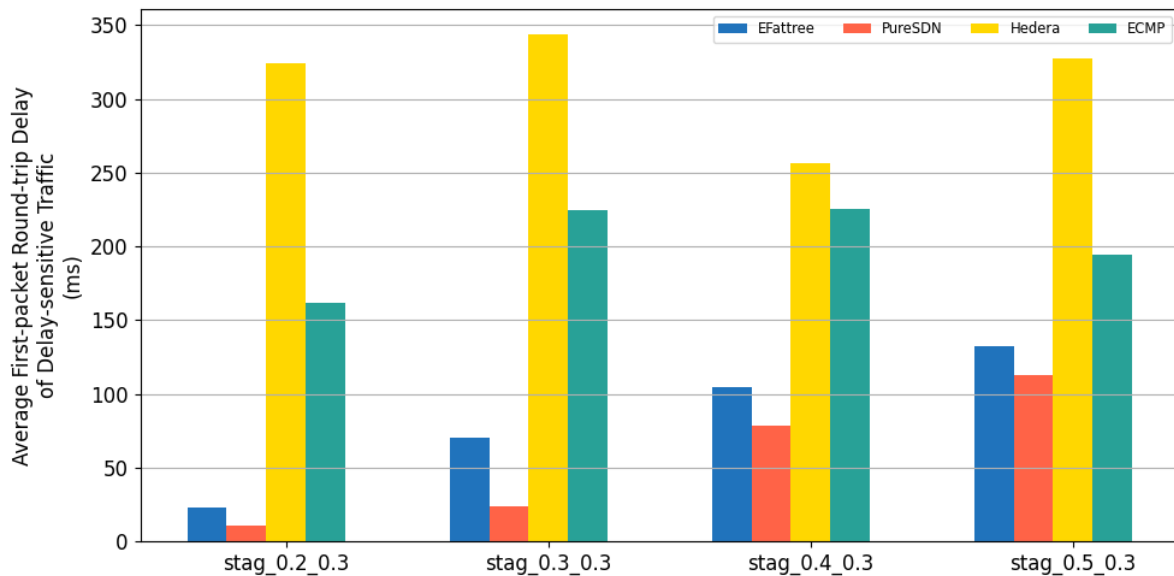
4.4.3 First packet RTT

Η μέτρηση first packet RTT είναι ο χρόνος που χρειάζεται το πρώτο πακέτο να φτάσει στον προορισμό του μέσα από το δίκτυο και μετά να στείλει πίσω στον αποστολέα το πακέτο επιβεβαίωσης «acknowledgment». Η μέτρηση συμπεριλαμβάνει όλες τις καθυστερήσεις που μπορεί να προκληθούν από την δρομολόγηση και την κίνηση δικτύου. Το πακέτο επιβεβαίωσης δεν χρειάζεται να ακολουθήσει την ίδια διαδρομή που ακολούθησε το πακέτο αποστολής.



Εικόνα 4-7 First packet round trip delay $k=4$

Στην Εικόνα 4-7 βλέπουμε την μέση καθυστέρηση του πρώτου πακέτου μετ' επιστροφή για $k = 4$. Αυτό που παρατηρούμε είναι ότι την μεγαλύτερη τιμή έχει το ECMP για την περίπτωση stag_0.3_0.3. Επίσης έχει την μεγαλύτερη τιμή και για την περίπτωση stag_0.2_0.3. Για τις περιπτώσεις stag_0.4_0.3 και stag_0.5_0.3 την μέγιστη τιμή την έχει ο PureSDN και ο Hedera αντίστοιχα. Αντιθέτως την μικρότερη τιμή την έχει ο EFatTree σε όλες τις περιπτώσεις.

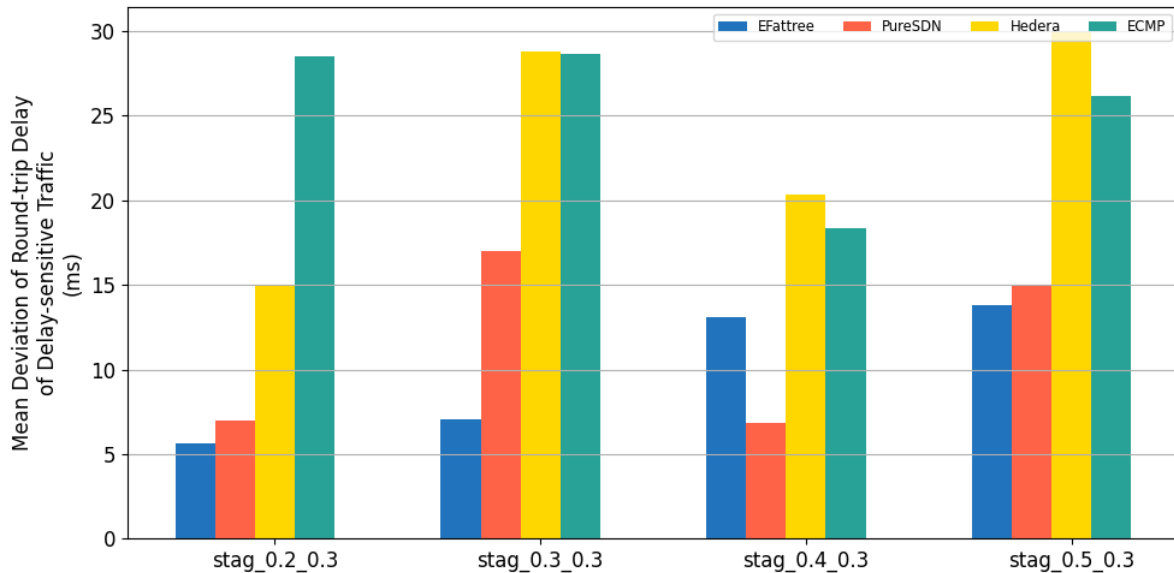


Εικόνα 4-8: First packet round trip delay $k=8$

Στην Εικόνα 4-8 βλέπουμε την καθυστέρηση για το πρώτο πακέτο. Αυτό που μπορούμε να παρατηρήσουμε είναι ότι ο Hedera έχει την μεγαλύτερη τιμή σε όλες τις περιπτώσεις και δεύτερος είναι ο ECMP, μετά είναι ο ECMP και τελευταίος ο PureSDN. Αυτή η σειρά είναι η ίδια σε όλες τις περιπτώσεις. Την μεγαλύτερη τιμή την έχει ο Hedera για την περίπτωση stag_0.2_0.3.

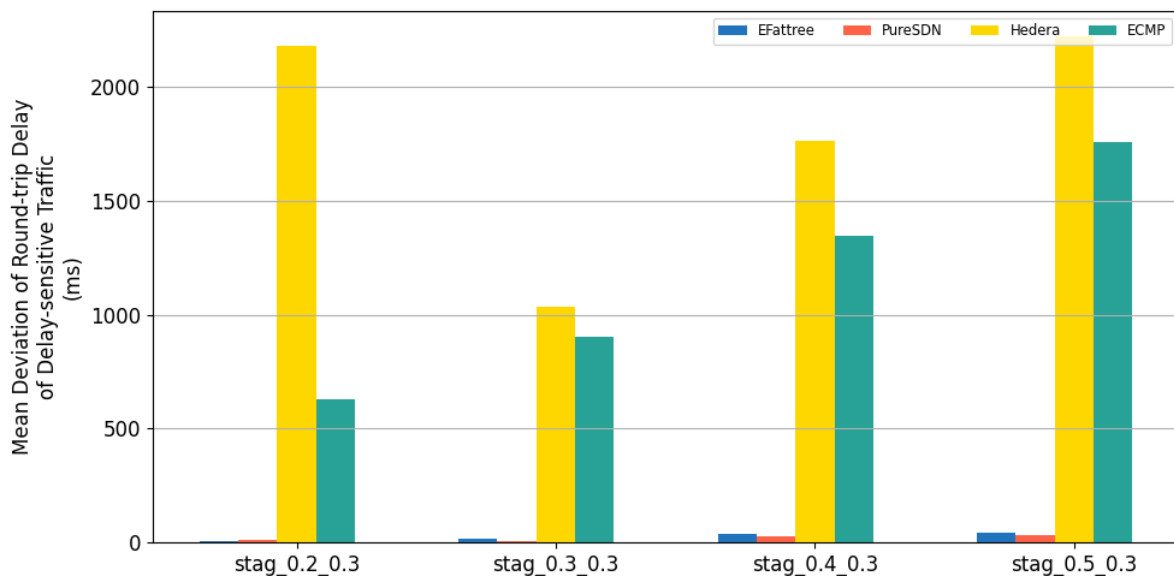
4.4.4 Μέση διακύμανση RTT

Η μέση διακύμανση ή μέση τιμή απόκλισης καθυστέρησης του χρόνου μετ' επιστροφής είναι ένα μέτρο για το πόσο οι μεμονωμένες μετρήσεις RTT αποκλίνουν από την μέση τιμή του RTT.



Εικόνα 4-9: Mean deviation of round trip delay $k=4$

Στην Εικόνα 4-9 βλέπουμε την μέση τιμή απόκλισης καθυστέρησης για $k=4$. Η μέση τιμή καθυστέρησης έχει την μεγαλύτερη τιμή στην περίπτωση stag_0.5_0.3 και είναι μεγαλύτερη από 30ms. Την μικρότερη τιμή την έχει ο EFatTree στην περίπτωση stag_0.2_0.3 και είναι λίγο πάνω από 5ms. Επίσης οι τιμές του ECMP είναι υψηλές πάνω από τα 20ms. Ενώ όσο αυξάνεται η κίνηση αυξάνεται και η τιμή του EFatTree.

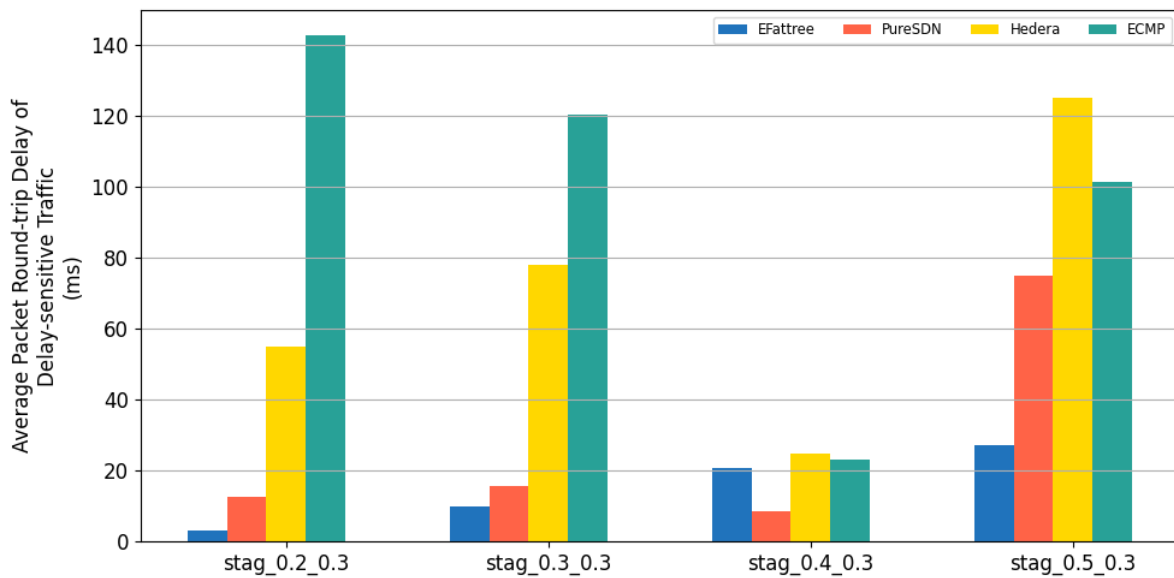


Εικόνα 4-10: Mean deviation of round trip $k=8$

Στην Εικόνα 4-10, βλέπουμε ότι η μέγιστη μέση απόκλιση είναι ελάχιστη για τους αλγορίθμους EFatTree και PureSDN. Την μέγιστη τιμή την έχει ο Hedera στην περίπτωση stag_0.5_0.3. Την μικρότερη τιμή την έχει ο ECMP στην περίπτωση stag_0.2_0.3. Όσο αυξάνεται η κίνηση αυξάνεται και η τιμή για τον αλγόριθμο ECMP.

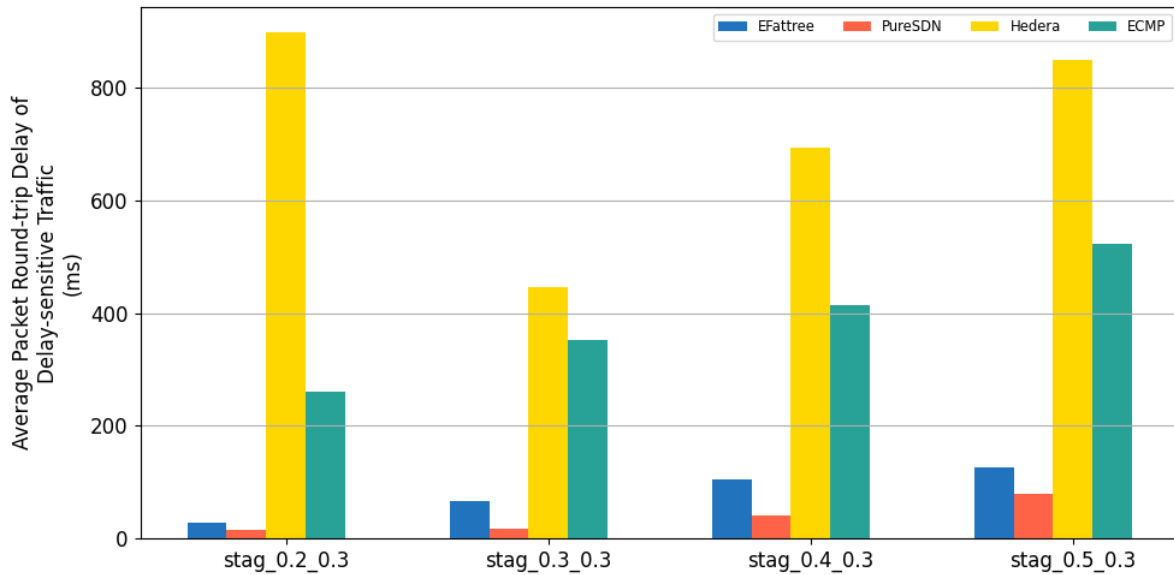
4.4.5 Average packet round-trip delay

Το round-trip delay ή το round-trip time είναι μία σημαντική μέτρηση η οποία υποδεικνύει την ποιότητα των επικοινωνιών μεταξύ 2 σημείων. Ορίζεται ως μία μετρική που μετρά σε χιλιοστά του δευτερολέπτου τον χρόνο που χρειάζεται για την αποστολή ενός πακέτου δεδομένων συν τον χρόνο που χρειάζεται για την επιβεβαίωση (acknowledgement) της λήψης αυτού του σήματος. Αυτή η χρονική καθυστέρηση περιλαμβάνει τους χρόνους διάδοσης για τις διαδρομές μεταξύ των δύο τελικών σημείων επικοινωνίας.



Εικόνα 4-11: Round trip delay για $k = 4$

Στην εικόνα 4-11 βλέπουμε ότι την μέγιστη τιμή για το round trip time την έχει ο αλγόριθμος ECMP. Ο ECMP έχει την μεγαλύτερη τιμή και για την περίπτωση stag_0.3_0.3. Η περίπτωση stag_0.4_0.3 έχει τις μικρότερες τιμές για όλους τους αλγορίθμους. Στις περιπτώσεις stag_0.4_0.3 και stag_0.5_0.3 ο αλγόριθμος Hedera έχει τις μεγαλύτερες τιμές.

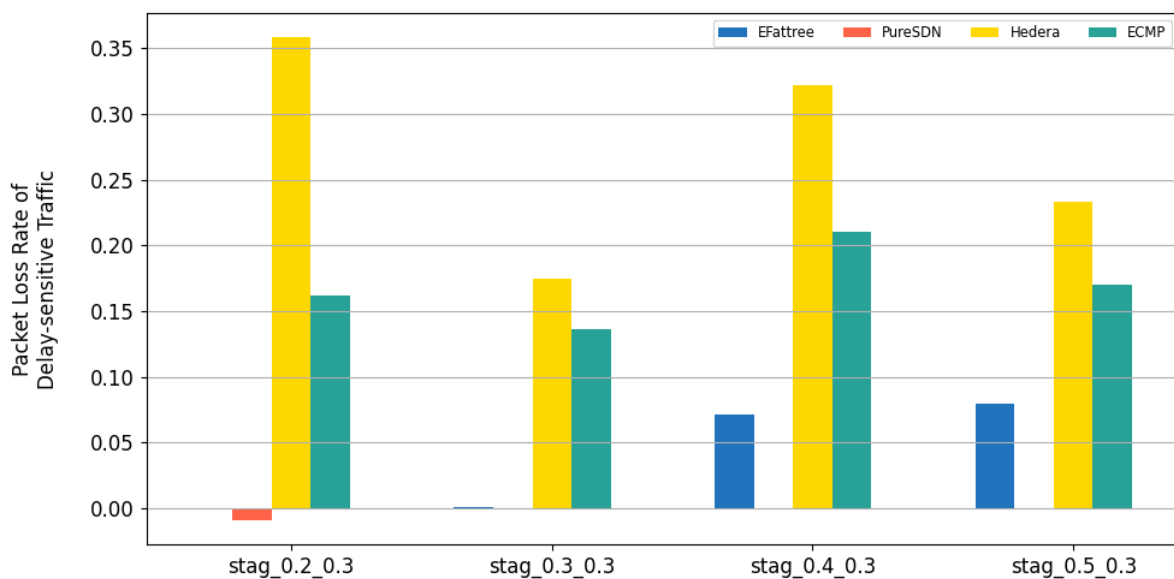


Εικόνα 4-12: Round trip delay για $k = 8$

Στην εικόνα 4-12 βλέπουμε ότι την μεγαλύτερη τιμή την έχει ο αλγόριθμος Hedera για την περίπτωση stag_0.2_0.3. Όσο αυξάνεται η κίνηση αυξάνονται και οι τιμές για τους αλγόριθμους PureSDN, EFatTree, ECMP. Τις μικρότερες τιμές των αλγορίθμων τις βρίσκουμε στην περίπτωση stag_0.2_0.3.

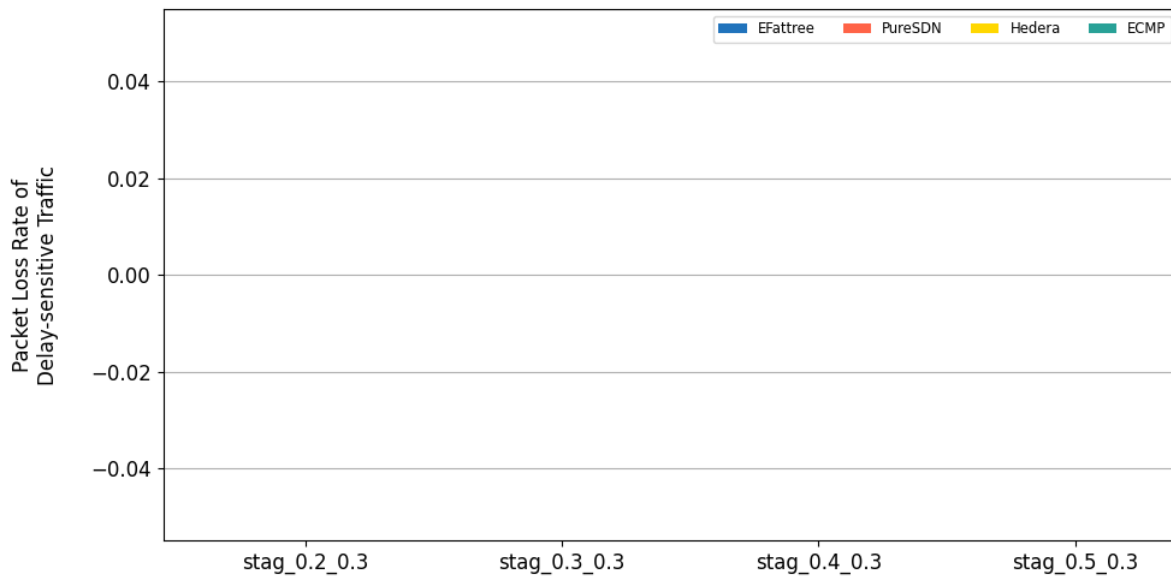
4.4.6 Packet Loss – απώλεια πακέτων

Η απώλεια πακέτων συμβαίνει όταν τα πακέτα αποτυγχάνουν να φτάσουν στον προορισμό τους. Συνήθως η υψηλή απώλεια πακέτων οδηγεί σε υψηλή καθυστέρηση καθώς ο αποστολέας χρειάζεται να ξαναστείλει τα χαμένα πακέτα, και αυτό ομοίως προκαλεί επιπλέον καθυστερήσεις στην μετάδοση δεδομένων.



Εικόνα 4-13: Ρυθμός απώλειας πακέτων $k = 8$

Στην Εικόνα 4-13 βλέπουμε την απώλεια πακέτων για $k = 8$ για τους τέσσερις αλγόριθμους. Την μεγαλύτερη απώλεια την έχει ο αλγόριθμος Hedera στην περίπτωση stag_0.2_0.3. Επίσης στην ίδια περίπτωση το EFatTree δεν έχει απώλειες και το PureSDN έχει αρνητικές. Στην περίπτωση stag_0.3_0.3 τις περισσότερες απώλειες της έχει ο Hedera ενώ το PureSDN δεν έχει καθόλου και ο EFatTree ελάχιστες. Στις άλλες δύο περιπτώσεις βλέπουμε ότι σταθερές απώλειες έχει ο EFatTree ενώ για την περίπτωση stag_0.5_0.3 οι απώλειες μειώνονται σε σύγκριση με το stag_0.4_0.3, όσο αυξάνεται η κυκλοφορία.



Εικόνα 4-14: Ρυθμός απώλειας πακέτων $k = 4$

Στην Εικόνα 4-14 βλέπουμε την απώλεια πακέτων $k = 4$. Όπως φαίνεται από την εικόνα δεν έχουμε απώλεια πακέτων στην περίπτωση $k = 4$.

4.5 Σύνοψη

Για $k = 4$ στη μέση απόδοση και την κανονικοποιημένη συνολική απόδοση των αλγορίθμων ο PureSDN αλγόριθμος υπερσχύει σε όλες τις περιπτώσεις. Την μέγιστη τιμή την έχει ο αλγόριθμος PureSDN για το stag_0.5_0.3 και η τιμή είναι 120 Mbps. Την ελάχιστη τιμή την έχει ο αλγόριθμος ECMP για το stag_0.3_0.3 και η τιμή είναι 74 Mbps. Η κανονικοποιημένη απόδοση κυμαίνεται από 45% έως 80%. Σχετικά με το χρόνο εγκατάστασης της ροής (average first-packet round trip delay) διαπιστώνουμε ότι την μικρότερη τιμή την έχει για όλες τις περιπτώσεις ο EFatTree και οι τιμές του είναι μέχρι 25ms. Επίσης την μέγιστη τιμή την έχει ο ECMP για την περίπτωση stag_0.3_0.3 με τιμή 250ms. Το εύρος των τιμών του PureSDN είναι από 50ms έως 150ms. Από την γραφική της μέσης καθυστέρησης του πακέτου μετ' επιστροφή βλέπουμε ότι και εδώ την μέγιστη τιμή την έχει ο ECMP με τιμή 145ms στο stag_0.2_0.3 και την δεύτερη μεγαλύτερη τιμή την έχει ο Hedera με τιμή 125ms στο stag_0.5_0.3. Επίσης και εδώ ο EFatTree έχει την μικρότερη τιμή της γραφικής με τιμή 1ms στο stag_0.2_0.3. Για την μέση διακύμανση μπορούμε να παρατηρήσουμε ότι όσο αυξάνεται η κίνηση αυξάνεται και η τιμή του EFatTree και ότι η τιμές του ECMP είναι σταθερά υψηλές, πάνω από 20ms. Την μέγιστη τιμή την έχει ο Hedera στην περίπτωση stag_0.5_0.3

Για $k = 8$ η μέση απόδοση αυξάνεται όσο αυξάνεται η κίνηση εντός του ίδιου switch. Στην 2^η περίπτωση την μέγιστη τιμή την έχει ο Hedera ενώ στην 3^η περίπτωση ο ECMP που είναι κοντά στα 630ms και στην 4^η ο Hedera που έχει την μεγαλύτερη τιμή. Η κανονικοποιημένη συνολική απόδοση αυξάνεται και αυτή όσο αυξάνεται η κίνηση εντός του ίδιου switch. Την μεγαλύτερη τιμή την έχει ο Hedera και είναι κοντά στα 55% για κίνηση 50% στο ίδιο switch, ενώ την δεύτερη μεγαλύτερη τιμή την έχει ο ECMP που είναι και αυτή κοντά στα 50% για κίνηση 40% στο ίδιο switch. Για το average first-packet RTT τις μέγιστες τιμές για όλες τις περιπτώσεις τις έχει ο Hedera και είναι όλες πάνω από τα 250ms. Επίσης της δεύτερες καλύτερες τιμές τις έχει ο ECMP για όλες τις περιπτώσεις και είναι πάνω από τα 150ms. Η γραφική για τους αλγορίθμους EFatTree και PureSDN είναι παράλληλη όσο αυξάνεται η κίνηση αυξάνεται και η γραφική. Για το Round trip delay την μεγαλύτερη τιμή την έχει ο Hedera για όλες τις περιπτώσεις, ενώ ο PureSDN έχει την μικρότερη. Την μεγαλύτερη τιμή την βρίσκουμε στην περίπτωση stag_0.2_0.3 και είναι πάνω από 750ms. Η τιμή του ECMP, του EFatTree, PureSDN αυξάνεται όσο αυξάνεται η κίνηση. Οι τιμές των EFatTree, PureSDN είναι κάτω από τα 200ms. Για την απώλεια πακέτων μπορούμε να παρατηρήσουμε ότι ο Hedera έχει την περισσότερη. Καμία απώλεια δεν έχει ο PureSDN, ενώ ο EFatTree έχει μία μικρή απώλεια έως 0.08%. Η απώλεια του Hedera είναι από 0.15 έως 0.35, ενώ του ECMP από 0.10 έως 0.20. Η μέση διακύμανση Round trip time αυξάνεται όσο αυξάνεται η κίνηση για τους αλγορίθμους Hedera και ECMP. Για τους αλγορίθμους EFatTree και PureSDN υπάρχει μία πάρα πολύ μικρή αύξηση αλλά όχι σημαντική. Επίσης η μεγαλύτερη τιμή είναι στην περίπτωση stag_0.5_0.3 για τον αλγόριθμο Hedera.

5 Συμπεράσματα

Η εξισορρόπηση φορτίου στα δίκτυα καθοριζόμενα από λογισμικό εξακολουθεί να είναι ένα θέμα το οποίο απασχολεί την ακαδημαϊκή και ερευνητική κοινότητα. Η εξισορρόπηση φορτίου είναι ένας τρόπος για να δημιουργήσουμε αποδοτικά το δίκτυο μας. Οι ερευνητές έχουν προτείνει πολλούς αλγορίθμους και τους έχουν εφαρμόσει σε πολλές διαφορετικές τοπολογίες. Μία ευρέως διαδεδομένη τοπολογία που χρησιμοποιείται στα κέντρα δεδομένων είναι το FatTree. Η συγκεκριμένη τοπολογία είναι με τέτοιο τρόπο σχεδιασμένη που μπορεί να παρέχει μεγαλύτερο συνολικό εύρος ζώνης, μεγαλύτερη ανοχή σε τυχόν σφάλματα και είναι εύκολα επεκτάσιμη. Με την εφαρμογή των διαφόρων αλγορίθμων εξισορρόπησης φορτίου στην τοπολογία FatTree μπορούμε να πετύχουμε μείωση της καθυστέρησης στις παρεχόμενες υπηρεσίες. Στην εργασία αυτή εξετάσαμε την αποτελεσματικότητα τριών αλγορίθμων: ECMP, Hedera, PureSDN για την τοπολογία FatTree με την χρήση του Mininet και την υλοποίηση των αλγορίθμων που αναπτύχθηκε από τον (Huangmachi 2017). Οι μετρικές που για τις οποίες πήραμε αποτελέσματα είναι: throughput, first packet round trip delay, mean deviation round trip delay, packet loss. Τις μετρικές τις πήραμε για δύο περιπτώσεις, για $k = 4$ και $k = 8$. Επίσης διαφοροποιήσαμε τα μοτίβα κυκλοφορίας στα edge switches, και συγκεκριμένα αυξάνουμε κατά 10% την κυκλοφορία στο ίδιο edge switch, τροποποιώντας αντίστοιχα την κυκλοφορία στα διαφορετικά switches στο ίδιο pod, και μεταξύ των διαφορετικό pod. Για το throughput μπορούμε να συμπεράνουμε ότι κυμαίνεται από 48% έως 75% για $k = 4$ και 15% έως 55% για $k = 8$. Όσο αυξάνεται η κυκλοφορία εντός του ίδιου switch αυξάνεται και το throughput. Η μέγιστη τιμή για $k = 4$ την έχει ο PureSDN και είναι για κυκλοφορία 75%. Ενώ η μέγιστη τιμή για $k = 8$ την έχει και εκεί ο Hedera για κυκλοφορία 55%, απλά οι τιμές του ECMP με του Hedera είναι πολύ κοντά και έχουν παρόμοια συμπεριφορά. Για το first packet round trip delay παρατηρούμε ότι ο ECMP έχει την μεγαλύτερη τιμή για $k = 4$, ενώ για $k = 8$ την έχει ο Hedera σε όλες τις περιπτώσεις. Την μικρότερη τιμή την έχει ο EFatTree για $k = 4$, ενώ για $k = 8$ την έχει ο PureSDN. Για το average packet round trip delay την μέγιστη τιμή την έχει ο Hedera για $k = 8$, την ελάχιστη ο PureSDN. Όσο αυξάνεται η κίνηση αυξάνεται και η τιμή. Για $k = 4$ την μέγιστη τιμή την έχει ο ECMP, και την μικρότερη ο EFatTree. Δεν έχουμε απώλεια πακέτων για $k = 4$, ενώ για $k = 8$ υπάρχει απώλεια για τον Hedera και για τον ECMP. Την μεγαλύτερη απώλεια την έχει ο Hedera, ενώ την μικρότερη ο EFatTree. Ο PureSDN έχει αρνητική απώλεια και αυτή μόνο σε μία περίπτωση. Οι απώλειες του ECMP είναι μεταξύ 0.12 έως 0.23. Ενώ του Hedera 0.15 έως 0.37. Για την μέση διακύμανση round trip time μπορούμε να παρατηρήσουμε ότι την καλύτερη συμπεριφορά την έχει ο αλγόριθμος Hedera και για τις δύο περιπτώσεις $k = 4$ και 8. Επίσης η μέση διακύμανση είναι αμελητέα για $k = 8$ για τους αλγορίθμους EFatTree και PureSDN. Επίσης όσο αυξάνεται η κίνηση τόσο αυξάνεται και η τιμή για τους αλγορίθμους Hedera, ECMP για $k = 8$. Ενώ για $k = 4$ αυτό δεν γίνεται. Επίσης την μεγαλύτερη τιμή την έχει ο αλγόριθμος Hedera και για τις δύο περιπτώσεις. Από τις γραφικές παραστάσεις μπορούμε να συμπεράνουμε ότι την καλύτερη συμπεριφορά την έχει ο αλγόριθμος Hedera. Αλλά ο αλγόριθμος αυτός δεν είναι γενικής χρήσης. Ο ECMP αλγόριθμος εν αντιθέσει είναι γενικής χρήσης αλλά μπορεί να επιτύχει βέλτιστη ροή κυκλοφορίας για τα κέντρα δεδομένων που έχουν τοπολογία FatTree. Το κύριο πρόβλημα των αλγορίθμων: PureSDN και Hedera είναι ότι δεν είναι γενικής χρήσης όπως είναι ο ECMP,

δεν είναι ευέλικτοι αλγόριθμοι. Αυτοί οι αλγόριθμοι δημιουργήθηκαν για να βοηθήσουν στην επίλυση συγκεκριμένων προβλημάτων και τις περισσότερες φορές σε ένα συγκεκριμένο δίκτυο. Συμπερασματικά ο ECMP είναι η καλύτερη επιλογή αλγορίθμου για την εξισορρόπηση φορτίου σε ένα δίκτυο κέντρων δεδομένων με την τοπολογία FatTree. Ο αλγόριθμος Hedera είναι επίσης μία πάρα πολύ καλή επιλογή υπό την προϋπόθεση να έχει την συνολική γνώση του δικτύου και των ροών.

Βιβλιογραφία

1. Martinez, J. (2025) Software-defined networking explained: 2025 SDN guide, Software-defined Networking Explained | 2025 SDN Guide. Available at: <https://www.strongdm.com/blog/software-defined-networking>
2. Braun, W. and Menth, M. (2014) 'Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices', Future Internet, 6(2), pp. 302–336.
3. Stallings, W. (2015) Foundations of Modern Networking. Addison-Wesley Professional.
4. Yangyang, G. (2021) What Is OpenFlow? How Does It Relate to SDN? - Huawei, info.support.huawei.com. Available at: <https://info.support.huawei.com/info-finder/encyclopedia/en/OpenFlow.html>.
5. Wikipedia Contributors (2019) Software-defined networking, Wikipedia. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Software-defined_networking.
6. VMware (2024) What is Software-Defined Networking (SDN)? | VMware Glossary, www.vmware.com. Available at: <https://www.vmware.com/topics/software-defined-networking>.
7. What is Software-Defined Networking (SDN)? (no date) CBT Nuggets. Available at: <https://www.cbtnuggets.com/blog/technology/networking/software-defined-networking>.
8. Nunes, B.A.A. et al. (2014) 'A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks', IEEE Communications Surveys & Tutorials, 16(3), pp. 1617–1634. Available at: <https://doi.org/10.1109/surv.2014.012214.00180>.
9. Mohsin, M. (2022) SDN Use Cases. Explained. - Mohammed Mohsin CCIE, CISSP, CCSP, CISM - Medium, Medium. Available at: <https://mohsinccie.medium.com/sdn-use-cases-explained-1e135e99b52f> (Accessed: 15 April 2025).
10. Chapter 2: Use Cases — Software-Defined Networks: A Systems Approach Version 2.1-dev documentation (no date) sdn.systemsapproach.org. Available at: <https://sdn.systemsapproach.org/uses.html>.
11. Top Five Apps and Services That Can Benefit from SDN (2016) Datacenterknowledge.com. Available at: <https://www.datacenterknowledge.com/data-center-software/top-five-apps-and-services-that-can-benefit-from-sdn>.
12. IBM (2023) Load Balancing | IBM, www.ibm.com. Available at: <https://www.ibm.com/topics/load-balancing>.
13. Wikipedia Contributors (2019) Load balancing (computing), Wikipedia. Wikimedia Foundation. Available at: [https://en.wikipedia.org/wiki/Load_balancing_\(computing\)](https://en.wikipedia.org/wiki/Load_balancing_(computing)).
14. Pramatarov, M. (2023) What is Load Balancing?, CloudDNS Blog. Available at: <https://www.cloudns.net/blog/load-balancing/>.
15. Kang M., Lee H., Song J., and Kwon Ted "Taekyoung", Proactive SDN-based Load Balancing for Datacenter Network (Poster), The 14th International Conference on Future Internet Technologies (CFI '19), Phuket, Thailand, August 2019
16. Yu, H. et al. (2018) 'Openflow Based Dynamic Flow Scheduling with Multipath for Data Center Networks', Computer Systems Science and Engineering, 33(4), pp. 251–258. Available at: <https://doi.org/10.32604/csse.2018.33.251>.

17. Demystifying DCN Topologies: Clos/Fat Trees - Part2 (no date) packetpushers.net. Available at: <https://packetpushers.net/blog/demystifying-dcn-topologies-clos-fat-trees-part2/>.
18. Alizadeh, M. et al. (2014) 'CONGA', ACM Special Interest Group on Data Communication [Preprint]. Available at: <https://doi.org/10.1145/2619239.2626316>.
19. Team, O. (2024) Link Aggregation and ECMP load-balancing, Ostinato Traffic Generator for Network Engineers. Ostinato. Available at: <https://ostinato.org/blog/network-basics-link-aggregation-ecmp>.
20. The (2021) P4~16~ Language Specification, P4.org. Available at: <https://p4.org/p4-spec/docs/P4-16-v1.2.2.html>.
21. Ke, C. H., & Hsu, S. J. (2020). Load balancing using P4 in software-defined networks. *Journal of Internet Technology*, 21(6), 1671-1679.
22. Al-Fares, M. et al. (2010) 'Hedera: dynamic flow scheduling for data center networks', *Networked Systems Design and Implementation*, pp. 19–19. Available at: <https://doi.org/10.5555/1855711.1855730>.
23. Using Round Robin for Simple Load Balancing (2025) F5, Inc. Available at: <https://www.f5.com/glossary/round-robin-load-balancing>.
24. Load Balancing Algorithms, Types and Techniques - Kemp (no date) kemptechnologies.com. Available at: <https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques>.
25. Wang, Y.-C. and You, S.-Y. (2018) 'An Efficient Route Management Framework for Load Balance and Overhead Reduction in SDN-Based Data Center Networks', *IEEE Transactions on Network and Service Management*, 15(4), pp. 1422–1434. Available at: <https://doi.org/10.1109/tnsm.2018.2872054>.
26. Li, L. and Xu, Q. (2017) 'Load balancing researches in SDN: A survey', pp. 403–408. Available at: <https://doi.org/10.1109/iceiec.2017.8076592>.
27. Hamdan, M. et al. (2021) 'A comprehensive survey of load balancing techniques in software-defined network', *Journal of Network and Computer Applications*, 174, p. 102856. Available at: <https://doi.org/10.1016/j.inca.2020.102856>.
28. IBM (2024) Data centers, ibm.com. Available at: <https://www.ibm.com/think/topics/data-centers>.
29. Xia, W. et al. (2017) 'A Survey on Data Center Networking (DCN): Infrastructure and Operations', *IEEE Communications Surveys & Tutorials*, 19(1), pp. 640–656. Available at: <https://doi.org/10.1109/comst.2016.2626784>.
30. Fancy, C. and Pushpalatha, M. (2021) 'Proactive Load Balancing Strategy Towards Intelligence-Enabled Software-Defined Network', *Arabian Journal for Science and Engineering* [Preprint]. Available at: <https://doi.org/10.1007/s13369-021-05621-8>.
31. C.N, S., E, G.M.K. and K, R. (2018) 'A Proactive Flow Admission and Re-Routing Scheme for Load Balancing and Mitigation of Congestion Propagation in SDN Data Plane', *International journal of Computer Networks & Communications*, 10(6), pp. 117–134. Available at: <https://doi.org/10.5121/ijcnc.2018.10607>.
32. Aymen Hasan Alawadi and Sándor Molnár (2021) 'Oddlab: fault-tolerant aware load-balancing framework for data center networks', *Annals of Telecommunications*, 77(9-10), pp. 641–662. Available at: <https://doi.org/10.1007/s12243-021-00898-0>.

33. Islam, M. T., Islam, N., & Refat, M. A. (2020). Node to node performance evaluation through RYU SDN controller. *Wireless Personal Communications*, 112, 555-570.
34. mininet (2021) Introduction to Mininet, GitHub. Available at: <https://github.com/Mininet/Mininet/wiki/Introduction-to-Mininet#what> (Accessed: 15 April 2025).
35. What Is an SDN Controller? Definition (2023) SDxCentral. Available at: <https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/what-is-sdn-controller/>.
36. GUEANT, V. (2013) iPerf - The TCP, UDP and SCTP network bandwidth measurement tool, Iperf.fr. Available at: <https://iperf.fr/>.
37. Moorthy, Kumar, S.M. and Sivalingam, K.M. (2023) 'ECN based Congestion Control for a Software Defined Network', arXiv (Cornell University) [Preprint]. Available at: <https://doi.org/10.48550/arxiv.2310.17129>.
38. Bai, W. et al. (2016) 'Enabling ECN in multi-service multi-queue data centers', *Networked Systems Design and Implementation*, pp. 537–549.
39. Ghorbani, S. et al. (2017) 'DRILL'. Available at: <https://doi.org/10.1145/3098822.3098839>.
40. Chatzieleftheriou, A., Legtchenko, S., Williams, H., & Rowstron, A. (2018). Larry: Practical network reconfigurability in the data center. In 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18) (pp. 141-156).
41. Laszka, A. and Gueye, A. (2015) 'Network Topology Vulnerability/Cost Trade-Off: Model, Application, and Computational Complexity', *Internet Mathematics*, 11(6), pp. 588–626. Available at: <https://doi.org/10.1080/15427951.2015.1018019>.
42. Maris Fajar Ramdhani, Sofia Naning Hertiana and Burhanudin Dirgantara (2016) 'Multipath routing with load balancing and admission control in Software-Defined Networking (SDN)', *International Conference on Information and Communication Technology* [Preprint]. Available at: <https://doi.org/10.1109/icoict.2016.7571949>.
43. He, K. et al. (2015) 'Presto', *ACM SIGCOMM Computer Communication Review*, 45(4), pp. 465–478. Available at: <https://doi.org/10.1145/2829988.2787507>.
44. Latency vs Throughput vs Bandwidth: Unraveling the Complexities of Network Speed (2023) Kentipedia. Available at: <https://www.kentik.com/kentipedia/latency-vs-throughput-vs-bandwidth/>.
45. Huangmachi (2017) GitHub - Huangmachi/PureSDN: PureSDN routing application for FatTree network., GitHub. Available at: <https://github.com/Huangmachi/PureSDN>.