

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ



## ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ στις ΨΗΦΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ ΚΑΙ ΔΙΚΤΥΑ

### ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΕΛΕΓΚΤΩΝ σε ΔΙΚΤΥΑ ΟΡΙΖΟΜΕΝΑ ΑΠΟ ΛΟΓΙΣΜΙΚΟ

Αρβανιτάκης Δημήτριος

Διπλωματική εργασία

που υποβλήθηκε στο Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στις Ψηφιακές Επικοινωνίες και Δίκτυα

Πειραιάς

Φεβρουάριος 2019

«Το άτομο το οποίο εκπονεί την Διπλωματική Εργασία φέρει ολόκληρη την ευθύνη προσδιορισμού της δίκαιης χρήσης του υλικού, η οποία ορίζεται στην βάση των εξής

παραγόντων: του σκοπού και χαρακτήρα της χρήσης (εμπορικός, μη κερδοσκοπικός ή εκπαιδευτικός), της φύσης του υλικού, που χρησιμοποιεί (τμήμα του κειμένου, πίνακες, σχήματα, εικόνες ή χάρτες), του ποσοστού και της σημαντικότητας του τμήματος, που χρησιμοποιεί σε σχέση με το όλο κείμενο υπό copyright, και των πιθανών συνεπειών της χρήσης αυτής στην αγορά ή στη γενικότερη αξία του υπό copyright κειμένου».

«Η παρούσα Διπλωματική Εργασία εγκρίθηκε ομόφωνα από την Τριμελή Εξεταστική Επιτροπή του Τμήματος Ψηφιακών Συστημάτων Πανεπιστημίου Πειραιώς σύμφωνα με τον Κανονισμό Λειτουργίας του Προγράμματος Μεταπτυχιακών Σπουδών στα Ψηφιακά Συστήματα.

Τα μέλη της Επιτροπής ήταν:

- ΤΣΑΓΚΑΡΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ (Επιβλέπων)
- ΑΛΕΞΙΟΥ ΑΓΓΕΛΙΚΗ
- ΡΟΥΣΚΑΣ ΑΓΓΕΛΟΣ

Η έγκριση της Διπλωματικής Εργασίας από το Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς δεν υποδηλώνει αποδοχή των γνώμων του συγγραφέα.»

## ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εργασία θα ασχοληθούμε με τα δίκτυα διαχειριζόμενα από λειτουργικό (software defined networks). Θα εξηγήσουμε τι είναι, την χρησιμότητα τους και τις εφαρμογές τους. Θα αναφερθούμε στα υπάρχουσα εργαλεία αξιολόγησης των ελεγκτών και σε ποια θα χρησιμοποιήσουμε. Μέσα από αυτά τα εργαλεία θα γίνει προσομοίωση ενός τέτοιου δικτύου και θα επικεντρωθούμε συγκεκριμένα στους ελεγκτές του (controllers). Αφού αναφερθούμε στους ελεγκτές που πρόκειται να χρησιμοποιήσουμε / δοκιμάσουμε θα παρουσιάσουμε τις τελικές μετρήσεις ως προς την αδράνεια τους (latency) και την διεκπεραιωτική τους ικανότητα (throughput). Στο τέλος τις εργασίας θα έχουμε συγκεντρωμένα τα χαρακτηριστικά του κάθε ελεγκτή ώστε να του συγκρίνουμε μεταξύ τους. Έτσι θα πάρουμε ένα δείγμα ως προς την αποτελεσματικότητά τους. Βέβαια η αποτελεσματικότητα του κάθε ελεγκτή έγκειται και σε άλλους παράγοντες στους οποίους θα αναφερθούμε. Στόχος είναι η διεξαγωγή συμπερασμάτων ως προς την λειτουργία των δικτύων διαχειριζόμενων από λειτουργικό με διαφορετικούς ελεγκτές για να καθοριστεί αν είναι ικανοί να πραγματοποιούν συγκεκριμένες εργασίες με επιτυχία, αν κάποιος ελεγκτής είναι καλύτερος σε σύγκριση με κάποιον άλλον, τα προτερήματά τους και τις αδυναμίες τους.

## Πίνακας περιεχομένων

<b>1 ΕΙΣΑΓΩΓΗ</b> .....	<b>1</b>
<b>2 ΔΙΚΤΥΑ ΔΙΑΧΕΙΡΙΖΟΜΕΝΑ ΑΠΟ ΛΟΓΙΣΜΙΚΟ</b> .....	<b>1</b>
2.1 ΠΑΡΑΔΟΣΙΑΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΙΚΤΥΩΝ .....	4
2.1.1 ΔΟΜΗ ΠΑΡΑΔΟΣΙΑΚΩΝ ΔΙΚΤΥΩΝ .....	4
2.1.2 ΠΕΡΙΟΡΙΣΜΟΙ ΠΑΡΑΔΟΣΙΑΚΩΝ ΔΙΚΤΥΩΝ .....	6
2.2 Η ΔΟΜΗ ΤΩΝ SDN .....	9
2.2.1 ΟΦΕΛΗ ΤΩΝ SDN .....	10
2.2.2 ΕΦΑΡΜΟΓΕΣ ΤΩΝ SDN .....	13
<b>3 OPENFLOW</b> .....	<b>17</b>
3.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ OPENFLOW .....	18
3.2 OPENFLOW ΔΙΑΚΟΠΤΕΣ .....	20
3.2.1 ΠΕΔΙΑ ΠΙΝΑΚΩΝ ΡΟΗΣ .....	21
3.2.2 ΤΥΠΟΙ ΡΟΩΝ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ .....	23
3.1 ΕΠΙΣΚΟΠΗΣΗ OPENFLOW ΠΡΩΤΟΚΟΛΛΟΥ .....	24
3.3.1 ΜΗΝΥΜΑΤΑ ΕΛΕΓΚΤΗ - ΔΙΑΚΟΠΤΗ .....	24
3.3.2 ΑΣΥΓΧΡΟΝΑ ΜΗΝΥΜΑΤΑ .....	25
3.3.3 ΣΥΜΜΕΤΡΙΚΑ ΜΗΝΥΜΑΤΑ .....	26
3.4 OPENFLOW ΕΛΕΓΚΤΕΣ .....	27
3.4.1 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ OPENFLOW ΕΛΕΓΚΤΩΝ .....	29
3.5 ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΔΕΣΗΣ .....	31
3.5.1 ΔΙΑΔΙΚΑΣΙΑ ΕΥΡΕΣΗΣ ΤΟΠΟΛΟΓΙΑΣ .....	34
<b>4 ΕΡΓΑΛΕΙΑ - ΜΕΤΡΗΣΕΙΣ</b> .....	<b>36</b>
4.1 ΕΡΓΑΛΕΙΑ ΑΞΙΟΛΟΓΗΣΗΣ ΕΛΕΓΚΤΩΝ .....	36
4.2 ΕΡΓΑΛΕΙΑ ΠΡΟΣΟΜΕΙΩΣΗΣ .....	38
4.3 ΜΕΘΟΔΟΛΟΓΙΑ .....	40
4.4 ΜΕΤΡΗΣΕΙΣ .....	41
4.4.1 ΡΟΧ .....	41
4.4.2 ΝΟΧ .....	42
4.4.3 RYU .....	43
4.4.4 MAESTRO .....	45
4.4.5 BEACON .....	47
4.4.6 OPENMUL .....	48

4.4.7 LIBFLUID BASE .....	51
4.4.8 OPENIRIS .....	42
4.4.9 ONOS .....	53
<b>5 ΣΥΜΠΕΡΑΣΜΑΤΑ</b> .....	<b>55</b>
5.1 ΑΝΑΛΥΣΗ ΜΕΤΡΗΣΕΩΝ.....	55
5.2 ΕΠΙΛΟΓΟΣ.....	58
ΠΗΓΕΣ.....	59

## ΚΕΦΑΛΑΙΟ 1

### 1 ΕΙΣΑΓΩΓΗ

Οι πρόσφατες τεχνολογικές εξελίξεις αλλάζουν τον τρόπο που γίνεται η διαχείριση και η διακίνηση δεδομένων. Συγκεκριμένα η εικονικοποίηση διακομιστών (server virtualization) και ο υπολογισμός του "σύννεφου" (cloud computing) εισχωρούν ολοένα και περισσότερο στις υποδομές. Το server virtualization παρέχει ευελιξία και αποδοτικότερη χρήση των πόρων και το cloud computing επεκτείνει περαιτέρω την ευελιξία με το να δίνει την δυνατότητα στην αγορά να επιλέγει κλιμακούμενα μοντέλα σχεδιασμένα για τις εκάστοτε ανάγκες τους χωρίς να απαιτούν την κυριότητα από μέρους τους ή την παροχή αποκλειστικών πόρων.

Ωστόσο αν και η ανάπτυξη των δύο παραπάνω τεχνολογιών είναι ραγδαία οι υπάρχουσες τεχνολογίες δικτύωσης αδυνατούν να εξυπηρετήσουν την ομαλή λειτουργία τους καθώς δεν σχεδιάστηκαν με αυτά τα πρότυπα. Οι υπάρχουσες τεχνολογίες δικτύωσης καθορίζονται από την στατικότητα τους. Το ότι δηλαδή κάθε αλλαγή χρειάζεται να γίνει χειροκίνητα και σε φυσικό επίπεδο.



Σχήμα 1: SDN

Εδώ έρχεται η τεχνολογία δικτύωσης που καθορίζεται από το λογισμικό (SDN). Η τεχνολογία δικτύωσης που καθορίζεται από το λογισμικό (SDN) είναι μια προσέγγιση στον

υπολογισμό του cloud που διευκολύνει τη διαχείριση δικτύου και επιτρέπει τον προγραμματισμό για την αποδοτική διαμόρφωση δικτύου, προκειμένου να βελτιωθεί η απόδοση και η παρακολούθηση του δικτύου. [1] Το SDN αποσκοπεί να αντιμετωπίσει το γεγονός ότι η στατική αρχιτεκτονική των παραδοσιακών δικτύων είναι αποκεντρωμένη και πολύπλοκη, ενώ τα τρέχοντα δίκτυα απαιτούν μεγαλύτερη ευελιξία και εύκολη αντιμετώπιση προβλημάτων. Το SDN προτείνει να συγκεντρωθεί η νοημοσύνη δικτύου σε ένα στοιχείο δικτύου αποσυνδέοντας τη διαδικασία προώθησης πακέτων δικτύου (επίπεδο δεδομένων) από τη διαδικασία δρομολόγησης (επίπεδο ελέγχου). Το επίπεδο ελέγχου αποτελείται από έναν ή περισσότερους ελεγκτές (controllers) οι οποίοι θεωρούνται ως ο εγκέφαλος του δικτύου SDN όπου ενσωματώνεται ολόκληρη η νοημοσύνη. Ο ελεγκτής είναι υπεύθυνος για τις αποφάσεις προώθησης που τοποθετούνται στους διακόπτες μέσω πρότυπων πρωτοκόλλων, όπως το OpenFlow.

Το πρωτόκολλο OpenFlow [2] εισάγεται για να ενοποιήσει της διεπαφής μεταξύ του αντικαθιστούμενου υλικού (εικονικό hardware) και του απομακρυσμένου ελεγκτή στο πρότυπο SDN. Αυτό το πρωτόκολλο παρέχει στον ελεγκτή μια δυνατότητα για να ανακαλύψει τους συμβατούς με το OpenFlow διακόπτες (switches), καθορίζει κανόνες προώθησης για το αντικαθιστούμενο υλικό και συλλέγει στατιστικά στοιχεία από τα switches. Το πρότυπο OpenFlow δημιουργήθηκε στο Πανεπιστήμιο του Στάνφορντ και τώρα διατηρείται από τον μη κερδοσκοπικό οργανισμό Open Networking Foundation [2] (ONF) που ήταν που ιδρύθηκε το Μάρτιο του 2011 από την Deutsche Telekom, το Facebook, το Google, τη Microsoft, τη Verizon, και το Yahoo.

Το SDN από την εμφάνισή του το 2011 συσχετίστηκε με το πρωτόκολλο OpenFlow (για απομακρυσμένη επικοινωνία με στοιχεία επιπέδου δικτύου για τον προσδιορισμό της διαδρομής των πακέτων δικτύου σε όλους τους διακόπτες δικτύου). Ωστόσο, από το 2012 [3] [4] το OpenFlow για πολλές εταιρείες δεν αποτελεί πλέον αποκλειστική λύση καθώς προσέθεσαν ιδιόκτητες τεχνικές, όπως για παράδειγμα το περιβάλλον ανοιχτού δικτύου της Cisco (Cisco Systems Open Network Environment) και η πλατφόρμα εικονικοποίησης δικτύων της Nicira (Nicira's network virtualization platform).



Ωστόσο, η συγκέντρωση πληροφοριών έχει τα δικά της μειονεκτήματα όταν πρόκειται για την ασφάλεια, [5] την επεκτασιμότητα και την ελαστικότητα [6] και αυτό είναι το κύριο θέμα του SDN.

## ΚΕΦΑΛΑΙΟ 2

### 2 ΔΙΚΤΥΑ ΔΙΑΧΕΙΡΙΖΟΜΕΝΑ ΑΠΟ ΛΟΓΙΣΜΙΚΟ

Δίκτυα που ορίζονται από το λογισμικό: πρόκειται για μια νέα αρχιτεκτονική δικτύου που περιλαμβάνει διαφορετικές τεχνολογίες δικτύων για να δομηθεί ένα ευέλικτο, κλιμακούμενο, ευκίνητο και εύκολα διαχειρίσιμο δίκτυο. Αυτό γίνεται εφικτό διαχωρίζοντας τα επίπεδα ελέγχου (control) και προώθησης (forwarding).

#### 2.1 ΠΑΡΑΔΟΣΙΑΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΙΚΤΥΩΝ

Τα δίκτυα που γνωρίζουμε σήμερα είναι απομονωμένα και χωρισμένα φυσικά για να καλύψουν τις ανάγκες της βιομηχανίας, των παρόχων υπηρεσιών, των οργανισμών και ακόμη και των τελικών χρηστών. Ο σχεδιασμός τους προβλέπει το επίπεδο ελέγχου όσο και το επίπεδο προώθησης δεδομένων να βρίσκεται στην ίδια συσκευή. Αν και αυτό το είδος δικτυακής αρχιτεκτονικής έχει δουλέψει αρκετά καλά στο παρελθόν, όπως αναφέραμε και παραπάνω με τις αυξανόμενες ανάγκες εικονικοποίησης η παρούσα δικτυακή αρχιτεκτονική θα είναι δύσκολο αν όχι αδύνατο να ανταποκριθεί στις νέες εικονικές απαιτήσεις. Στο παραπάνω θα πρέπει να συνυπολογίσουμε ότι τα τεχνολογικά τμήματα των εταιριών επιδιώκουν να εικονικοποιήσουν τους servers τους και λόγω κόστους.

##### 2.1.1 ΔΟΜΗ ΠΑΡΑΔΟΣΙΑΚΩΝ ΔΙΚΤΥΩΝ

Για να κατανοήσουμε λίγο καλύτερα την υπάρχουσα αρχιτεκτονική αξίζει να αναφερθούμε λίγο στη δομή της. Ένα τηλεπικοινωνιακό δίκτυο είναι μια συλλογή τερματικών κόμβων (terminal nodes) [7] οι οποίοι είναι συνδεδεμένοι έτσι ώστε να επιτρέπουν την επικοινωνία μεταξύ των τερματικών (terminals). Οι κόμβοι χρησιμοποιούν εναλλαγή κυκλωμάτων, μεταγωγή μηνυμάτων ή μεταγωγή πακέτων για να περάσουν το σήμα μέσω των σωστών συνδέσεων και κόμβων για να φτάσουν στο σωστό τερματικό προορισμό.

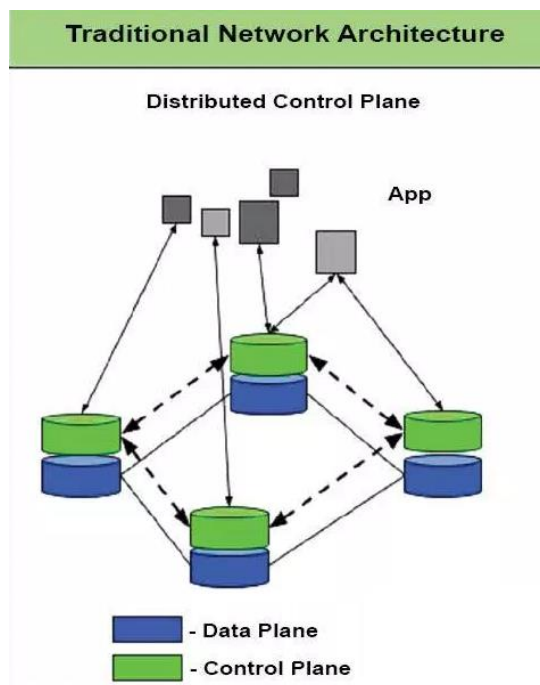
Κάθε τερματικό στο δίκτυο έχει συνήθως μια μοναδική διεύθυνση, ώστε τα μηνύματα ή οι συνδέσεις να μπορούν να δρομολογηθούν στους σωστούς παραλήπτες. Η συλλογή διευθύνσεων στο δίκτυο ονομάζεται χώρος διευθύνσεων. Παραδείγματα τηλεπικοινωνιακών

δικτύων είναι: τα δίκτυα υπολογιστών, το διαδίκτυο, το τηλεφωνικό δίκτυο, το παγκόσμιο δίκτυο Telex, το αεροναυτικό δίκτυο ACARS.

Γενικότερα κάθε δίκτυο τηλεπικοινωνιών αποτελείται εννοιολογικά από τρία μέρη ή επίπεδα (planes). Το επίπεδο δεδομένων (data plane, επίσης επίπεδο χρήστη, επίπεδο φορέα ή επίπεδο προώθησης): περιέχει την κίνηση των χρηστών του δικτύου, το πραγματικό ωφέλιμο φορτίο, το επίπεδο ελέγχου (control plane): περιέχει πληροφορίες ελέγχου και το επίπεδο διαχείρισης (management plane): περιέχει τις λειτουργίες και την διαχείριση της κυκλοφορίας που απαιτείται για τη διαχείριση του δικτύου. Το επίπεδο διαχείρισης μερικές φορές θεωρείται ως μέρος του επιπέδου ελέγχου.

Τα δίκτυα δεδομένων (data networks) χρησιμοποιούνται εκτενώς σε όλο τον κόσμο για να συνδέσουν άτομα και οργανισμούς. Τα δίκτυα δεδομένων μπορούν να συνδεθούν για να επιτρέψουν στους χρήστες απρόσκοπτη πρόσβαση σε πόρους που φιλοξενούνται εκτός του συγκεκριμένου φορέα στον οποίο είναι συνδεδεμένοι. Οι τερματικοί σταθμοί που είναι συνδεδεμένοι με δίκτυα TCP / IP απευθύνονται χρησιμοποιώντας διευθύνσεις IP. Υπάρχουν διαφορετικοί τύποι διεύθυνσης IP, αλλά η πιο συνηθισμένη είναι η IPv4 και προσφάτως η IPv6. Κάθε μοναδική διεύθυνση αποτελείται από 4 ακέραιους αριθμούς μεταξύ 0 και 255, οι οποίοι διαχωρίζονται συνήθως με τελείες όταν καταγράφονται προς τα κάτω, π.χ. 82.131.34.56. Τα TCP / IP είναι τα θεμελιώδη πρωτόκολλα που παρέχουν τον έλεγχο και τη δρομολόγηση των μηνυμάτων σε όλο το δίκτυο δεδομένων. Υπάρχουν πολλές διαφορετικές δομές δικτύου που μπορούν να χρησιμοποιηθούν για την αποτελεσματική δρομολόγηση μηνυμάτων TCP / IP, για παράδειγμα: δίκτυα ευρείας περιοχής (WAN), δίκτυα μητροπολιτικών περιοχών (MAN), τοπικά δίκτυα (LAN), δίκτυα περιοχής Internet (IAN), δίκτυα πανεπιστημιούπολεων (CAN), εικονικά ιδιωτικά δίκτυα (VPN).

Τα Datacenter βασίζονται επίσης σε TCP / IP για επικοινωνία μεταξύ των μηχανών. Συνδέουν χιλιάδες διακομιστές, έχουν σχεδιαστεί για να είναι ιδιαίτερα ανθεκτικά, να παρέχουν χαμηλή καθυστέρηση (latency), η οποία είναι συνήθως μέχρι και εκατοντάδες μικροδευτερόλεπτα (μs), και υψηλό εύρος ζώνης (bandwidth). Η τοπολογία του δικτύου των Datacenter διαδραματίζει σημαντικό ρόλο στον καθορισμό του επιπέδου της αντοχής στις αστοχίες, της ευκολίας της σταδιακής επέκτασης, του εύρους ζώνης της επικοινωνίας και της καθυστέρησης [8].



Σχήμα 2.1: Παραδοσιακή αρχιτεκτονική δικτύων

### 2.1.2 ΠΕΡΙΟΡΙΣΜΟΙ ΠΑΡΑΔΟΣΙΑΚΩΝ ΔΙΚΤΥΩΝ

Οι υπάρχουσες παραδοσιακές αρχιτεκτονικές δικτύου δεν έχουν κατασκευαστεί ώστε να ανταποκρίνονται στις σημερινές απαιτήσεις για τους τελικούς χρήστες, τους παρόχους υπηρεσιών και τις επιχειρήσεις. Η αρχιτεκτονική και η υποδομή που έχει χρησιμοποιηθεί για την οικοδόμηση δικτύων παλαιού τύπου δεν έχει αλλάξει εδώ και σχεδόν τρεις δεκαετίες. Πολλοί περιορισμοί πρέπει να ξεπεραστούν εάν οι οργανισμοί πρόκειται να κάνουν τη μετάβαση σε ένα νέο δίκτυο IP. Αυτοί οι περιορισμοί περιλαμβάνουν [9]:

- Άκαμπτη αρχιτεκτονική και τοπολογία  
 Η αρχιτεκτονική που χρησιμοποιήθηκε για την κατασκευή παραδοσιακών δικτύων σχεδιάστηκε για μια εποχή όπου οι πόροι πληροφορικής, όπως η υπολογιστική υποδομή και η υποδομή εφαρμογών, ήταν αρκετά στατικές. Επομένως, και το δίκτυο μπορεί να είναι άκαμπτο. Όταν συμβαίνει κάτι που απαιτεί την αλλαγή του δικτύου, όπως η προσθήκη ή η διαγραφή μιας νέας εφαρμογής, συχνά χρειάζονται μεγάλοι χρόνοι για την πραγματοποίηση ακόμη και απλών αλλαγών.

Στη σημερινή εποχή που οι απαιτήσεις για ευελιξία είναι αυξημένες, αυτό δεν είναι αποδεκτό, επειδή η υποδομή πληροφορικής μπορεί να είναι τόσο ευέλικτη όσο η λιγότερο ευέλικτη συνιστώσα της (που είναι συχνά το δίκτυο). Πολύ συχνά, η ακαμψία της υποδομής των παραδοσιακών δικτύων είναι ένα τεράστιο εμπόδιο για τις επιχειρήσεις που γίνονται ευέλικτες επιχειρήσεις.

- Τα παραδοσιακά δίκτυα είναι βασισμένα σε κλειστά, ιδιόκτητα πρωτόκολλα. Οι κορυφαίοι προμηθευτές της αγοράς χρησιμοποιούν κλειστές, αποκλειστικές τεχνολογίες ως έναν τρόπο για την ταχύτερη παροχή χαρακτηριστικών (features), αντί να περιμένουν να αναπτυχθούν τα πρότυπα της βιομηχανίας. Συχνά, αυτό οδηγεί σε "κλειδώμα του πωλητή" (vendor lock) και εμποδίζει τους πελάτες να εφαρμόσουν την καλύτερη τεχνολογία.

Παρά τους περιορισμούς αυτούς, τα κλειστά, ιδιόκτητα δίκτυα ήταν επαρκή όταν το δίκτυο υπήρχε τοπικά σε επίπεδο εταιρίας ή οργανισμού. Το cloud οδήγησε στην ανάγκη για μεγαλύτερη ενοποίηση μεταξύ των εφαρμογών καθώς και της υποδομής των υπολογισμών και των δικτύων. Οι κλειστές και αποκλειστικές τεχνολογίες μπορούν να παρεμποδίσουν τα οικοσυστήματα του cloud και να περιορίσουν την καινοτομία.

- Η ανεπαρκής χρήση των πόρων του δικτύου

Τα παραδοσιακά δίκτυα πολλαπλών επιπέδων χρησιμοποιούν το πρωτόκολλο Spanning Tree Protocol (STP) για προστασία από βρόχους δρομολόγησης και καταιγίδες εκπομπής. Όταν υπάρχουν πολλαπλές διαδρομές μεταξύ δύο σημείων, το STP θα ενεργοποιήσει την ταχύτερη διαδρομή, θα απενεργοποιήσει την εναλλακτική διαδρομή και θα την ενεργοποιήσει μόνο όταν αποτύχει η κύρια σύνδεση.

Το STP ήταν ένα μεγάλο άλμα προς τα εμπρός για τη δικτύωση πριν από μερικές δεκαετίες. Ωστόσο, καθώς τα δίκτυα έχουν αυξηθεί, η χρήση του STP έχει προκαλέσει τις οργανώσεις να εγκαθιστούν παραπάνω εξοπλισμό από όσο πραγματικά χρησιμοποιούν. Σήμερα, ως αποτέλεσμα του STP, μέχρι το μισό των θυρών του δικτύου είναι ανενεργές, επειδή είναι παθητικοί σύνδεσμοι.

Αποτέλεσμα, η λιγότερο από βέλτιστη αξιοποίηση του δικτύου της τάξης του 30%, σύμφωνα με την ZK Research.

- Υλικό-κεντρικό

Με δίκτυα παλαιού τύπου, κάθε είδους κλιμάκωση για επιπλέον χωρητικότητα ή νέες υπηρεσίες πρέπει να γίνει μέσω της προσθήκης νέου υλικού. Αυτό συνήθως απαιτεί μεγάλες αναβαθμίσεις της υπάρχουσας τεχνολογίας για μεγάλο χρονικό διάστημα, με αποτέλεσμα τη μείωση της διαθεσιμότητας των εφαρμογών και των δεδομένων.

Επιπλέον, ο χαρακτήρας των παραδοσιακών δικτύων τα οποία εξαρτώνται από το υλικό, καθιστούν το δίκτυο πολύ δαπανηρό, ιδίως για τους παρόχους υπηρεσιών και τις παγκόσμιες επιχειρήσεις, οι οποίες συχνά χρειάζεται να προσθέσουν γρήγορα χωρητικότητα ή οι υπηρεσίες.

- Χειροκίνητες διαδικασίες διαμόρφωσης

Οι παραδοσιακές συσκευές δικτύου, όπως διακόπτες και δρομολογητές, σχεδιάζονται με ενσωματωμένα επίπεδα ελέγχου και δεδομένων. Ως εκ τούτου, η πλειονότητα των ρυθμίσεων και της διαχείρισης πρέπει να γίνεται με από συσκευή σε συσκευή. Αυτό οδηγεί σε εκτεταμένες περιόδους διαχείρισης και σε υψηλό ποσοστό ανθρώπινου σφάλματος.

Η έρευνα της ZK 2014 Network Purchase Intention Study αποκαλύπτει ότι το ανθρώπινο λάθος είναι η πρώτη αιτία της διακοπής του δικτύου σήμερα. Για να ενεργοποιηθούν οι γρήγορες και ακριβείς αλλαγές, η αυτοματοποίηση έχει καταστεί κορυφαία προτεραιότητα, με τους ηγέτες της πληροφορικής και τα στελέχη των επιχειρήσεων να θέλουν να αυτοματοποιήσουν τις αλλαγές στην υποδομή υπολογισμών και εφαρμογών τους βάσει των επιχειρηματικών πολιτικών. Η επικεντρωμένη στη συσκευή φύση της διαχείρισης των δικτυακών συσκευών καθιστά δύσκολη, αν όχι αδύνατη, την αυτοματοποίηση στο δίκτυο.

- Έλλειψη αναπροσαρμογής / προγραμματισμού

Είναι σημαντικό το δίκτυο να μπορεί να διασυνδέεται με ένα ευρύ φάσμα εφαρμογών και υποδομών για τη βελτιστοποίηση της απόδοσης. Για παράδειγμα, όταν γίνεται εκκίνηση μιας κλήσης βίντεο, η εφαρμογή πρέπει να κατευθύνει το δίκτυο για να διατηρήσει ένα συγκεκριμένο εύρος ζώνης για τη φωνή και τις εικόνες.

Αυτή η αναπροσαρμογή είναι δύσκολη σήμερα, επειδή η υποδομή παλαιού τύπου δεν διαθέτει προγραμματιζόμενες διεπαφές, πράγμα το οποίο απαγορεύει στους

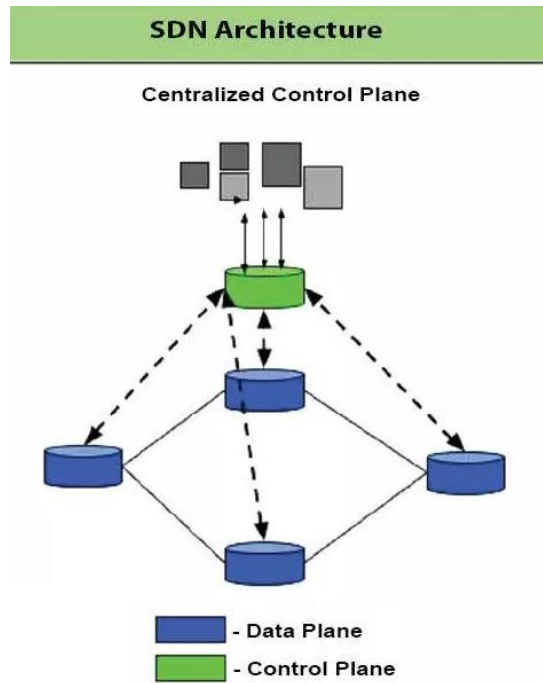
προγραμματιστές εφαρμογών να προγραμματίζουν το δίκτυο για αυτόματη προσαρμογή αυτών των τύπων τυχαίων μεταβλητών.

- Σχεδιασμένο για παλαιότερες εφαρμογές

Τα παραδοσιακά δίκτυα είναι βελτιστοποιημένα για παλαιότερες εφαρμογές όπως e-mail, φωνή, CRM και άλλες στατικές εφαρμογές. Οι εφαρμογές του σήμερα είναι σχεδιασμένες για το cloud και περιλαμβάνουν εφαρμογές για κινητά και μεγάλες αναλύσεις δεδομένων. Αυτές οι εφαρμογές έχουν σημαντικά διαφορετικές απαιτήσεις δικτύου από τις παλιότερες αντίστοιχες εφαρμογές, αν και υπάρχει ακόμα ανάγκη να υποστηριχθούν ταυτόχρονα και οι δύο. Η στροφή προς το cloud computing οδηγεί σε μια νέα εποχή στη δικτύωση. Ένα νέο δίκτυο IP που είναι βελτιστοποιημένο για το cloud είναι πλέον απαραίτητο.

## 2.2 Η ΔΟΜΗ ΤΩΝ SDN

Η βασική ιδέα του SDN είναι σχετικά απλή. Οι δρομολογητές, οι διακόπτες ή οποιαδήποτε άλλη συσκευή του δικτύου έχει δύο επίπεδα, το πρώτο επίπεδο είναι το επίπεδο προώθησης που μπορεί να ανακληθεί η διαβίβαση των δεδομένων. επομένως ονομάζεται επίπεδο δεδομένων η μεταφοράς. Το 2<sup>ο</sup> επίπεδο είναι το επίπεδο ελέγχου, το οποίο είναι υπεύθυνο για όλες τις πληροφορίες του δικτύου και τη λήψη αποφάσεων σχετικά με το πού να κατευθύνει την κυκλοφορία. Η ιδέα του SDN είναι η αποσύνδεση αυτών των δύο επιπέδων και η μετατροπή του παραδοσιακού στατικού δικτύου σε ένα ανταποκρινόμενο, προγραμματιζόμενο και έξυπνό δίκτυο που μπορεί να είναι κεντρικά ελεγχόμενο. Όλα αυτά μπορούν να γίνουν με λογικούς τρόπους που θα ανταποκρίνονται ανάλογα με τα μοντέλα κυκλοφορίας, τους τύπους ή ακόμα και τις καταστάσεις έκτακτης ανάγκης.



Σχήμα 2.2: Αρχιτεκτονική δικτύων που καθορίζονται από λογισμικό

### 2.2.1 ΟΦΕΛΗ ΤΩΝ SDN

Τα δίκτυα διαχειριζόμενα από λογισμικό αλλάζουν το τρόπο με τον οποίο οι μηχανικοί και οι σχεδιαστές των δικτύων δημιουργούν και διαχειρίζονται τα δίκτυα. Με τα SDN τα δίκτυα γίνονται εύκολα στη διαχείριση και τον προγραμματισμό, δεν είναι πιά αποκλειστικά και χρησιμοποιούν ανοικτά πρότυπα. Οι επιχειρήσεις και οι φορείς αποκτούν μεγαλύτερο έλεγχο στα δίκτυα τους και μπορούν πιά να τα προσαρμόσουν και να τα βελτιστοποιήσουν για τη μείωση του κόστους του. Κάποια από τα οφέλη των SDN αναφέρονται παρακάτω [10]:

- Παροχή κεντρικού δικτύου.

Τα δίκτυα που έχουν καθοριστεί από το λογισμικό παρέχουν μια κεντρική εικόνα ολόκληρου του δικτύου, καθιστώντας ευκολότερη την κεντρική διαχείριση και την παροχή υπηρεσιών. Για παράδειγμα, περισσότερα VLANs γίνονται μέρος των φυσικών LAN, δημιουργώντας ένα "γόνδιο δεσμό" συνδέσεων και εξαρτήσεων. Διαχωρίζοντας τα επίπεδα ελέγχου και δεδομένων, τα SDN μπορούν να επιταχύνουν την παροχή υπηρεσιών και να παρέχουν μεγαλύτερη ευελιξία στην παροχή τόσο των εικονικών όσο και των φυσικών συσκευών δικτύου από μια κεντρική τοποθεσία.



- Ολιστική διαχείριση επιχειρήσεων.

Τα εταιρικά δίκτυα πρέπει να δημιουργήσουν νέες εφαρμογές και εικονικές μηχανές για να ικανοποιήσουν νέες ανάγκες, όπως αυτές για μεγάλα δεδομένα. Το SDN επιτρέπει στους διαχειριστές IT να πειραματιστούν με τη διαμόρφωση του δικτύου χωρίς να επηρεάζουν το δίκτυο. Το SDN υποστηρίζει επίσης τη διαχείριση των φυσικών και των εικονικών διακοπών και των συσκευών δικτύου από έναν κεντρικό ελεγκτή, κάτι που δεν μπορούμε να κάνουμε με το SNMP. Το SDN παρέχει ένα ενιαίο σύνολο API's για τη δημιουργία μιας ενιαίας κονσόλας διαχείρισης για φυσικές και εικονικές συσκευές.

- Πιο συμπαγής ασφάλεια.

Ένα από τα πλεονεκτήματα των SDN που απευθύνεται περισσότερο στους διαχειριστές πληροφορικής είναι η κεντρική ασφάλεια. Η εικονικοποίηση έχει κάνει τη διαχείριση του δικτύου πιο δύσκολη. Με τις εικονικές μηχανές που έρχονται και πηγαινούν ως μέρος των φυσικών συστημάτων, είναι πιο δύσκολη η εφαρμογή ενός τείχους προστασίας με συνέπεια και το φιλτράρισμα του περιεχομένου σύμφωνα με την πολιτική μας. Αν προσθέσουμε και επιπλέον πολυπλοκότητα όπως η διασφάλιση συσκευών BYOD (συσκευές που συνδέονται στο δίκτυο από τους χρήστες, χωρίς να είναι στον αρχικό σχεδιασμό), το πρόβλημα ασφαλείας επιδεινώνεται.

Ο ελεγκτής SDN παρέχει ένα κεντρικό σημείο ελέγχου για τη συνεπή διανομή των πληροφοριών ασφαλείας και πολιτικής σε όλη την επιχείρηση. Συγκεντρώνοντας όμως τον κεντρικό έλεγχο ασφαλείας σε μια οντότητα (όπως τον ελεγκτή SDN) έχει το μειονέκτημα να δημιουργήσει ένα κεντρικό σημείο επίθεσης. Αν όμως το SDN σχεδιαστεί και εφαρμοστεί με ασφάλεια και σωστά μπορεί να χρησιμοποιηθεί αποτελεσματικά για τη διαχείριση της ασφαλείας σε ολόκληρη την επιχείρηση.

- Χαμηλότερο λειτουργικό κόστος.

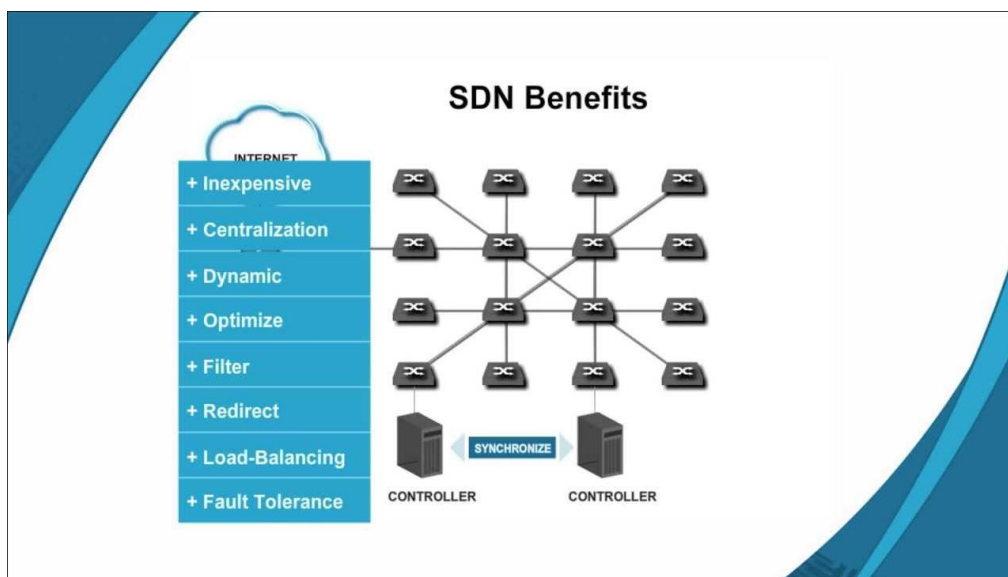
Η αποτελεσματικότερη διαχείριση, η βελτιστοποίηση στη χρήση του server, ο καλύτερος έλεγχος της εικονικοποίησης και άλλα οφέλη θα πρέπει να έχουν ως

αποτέλεσμα λειτουργικές εξοικονομήσεις. Παρόλο που είναι νωρίς ακόμα για να πούμε πόσα τελικά μπορούμε εξοικονομήσουμε, το SDN θα πρέπει να μειώσει το συνολικό λειτουργικό κόστος και να οδηγήσει σε εξοικονόμηση πόρων, καθώς πολλά από τα θέματα ρουτίνας της διαχείρισης του δικτύου μπορούν να συγκεντρωθούν και να αυτοματοποιηθούν.

- Εξοικονόμηση υλικού και μειωμένες κεφαλαιουχικές δαπάνες.  
Η υιοθέτηση του SDN δίνει νέα ζωή σε υπάρχουσες συσκευές δικτύου. Το SDN διευκολύνει τη βελτιστοποίηση του προμηθευόμενου hardware. Το υπάρχον hardware μπορεί να επανατοποθετηθεί χρησιμοποιώντας τις οδηγίες από τον ελεγκτή SDN και το λιγότερο δαπανηρό υλικό μπορεί να αξιοποιηθεί σε μεγαλύτερο βαθμό, καθώς οι νέες συσκευές ουσιαστικά γίνονται "λευκά κουτιά / white boxes" με όλες τις εντολές να έρχονται από τον ελεγκτή SDN.
- Αφαίρεση πόρων από το Cloud.  
Το cloud είναι εδώ για να παραμείνει και εξελίσσεται σε μια ενοποιημένη υποδομή. Αφαιρώντας κάποιους πόρους του cloud χρησιμοποιώντας τα δίκτυα που ορίζονται από λογισμικό, είναι ευκολότερο να ενοποιήσουμε τους υπόλοιπους πόρους. Τα στοιχεία του δικτύου που αποτελούν τις μαζικές πλατφόρμες των data centers μπορούν να διαχειρίζονται από τον ελεγκτή SDN.
- Εγγυημένη παράδοση περιεχομένου.  
Η δυνατότητα διαμόρφωσης και ελέγχου της κυκλοφορίας δεδομένων είναι ένα από τα κύρια πλεονεκτήματα της δικτύωσης που καθορίζεται από το λογισμικό. Η ικανότητά της να κατευθύνει και να αυτοματοποιεί την κυκλοφορία δεδομένων καθιστά ευκολότερη την εφαρμογή της ποιότητας των υπηρεσιών (QoS) για τη μετάδοση φωνής μέσω IP και πολυμέσων. Η ροή βίντεο υψηλής ποιότητας είναι ευκολότερη, επειδή το SDN βελτιώνει την ανταπόκριση του δικτύου για να εξασφαλίσει μια άψογη εμπειρία χρήστη.

Τα συγκεκριμένα πλεονεκτήματα της δικτύωσης που καθορίζεται από το λογισμικό θα διαφέρουν από δίκτυο σε δίκτυο, αλλά υπάρχουν οφέλη από τον διαχωρισμό του δικτύου και την ευκινησία που προσφέρει για τη διαχείριση και την αυτοματοποίηση. Ο καλύτερος

τρόπος για την αξιοποίηση στο έπακρο του SDN είναι η αξιολόγηση των στοιχείων του δικτύου και της υποδομής για να διαπιστώσουμε εάν το SDN μπορεί να βοηθήσει στην αντιμετώπιση θεμάτων όπως η διαθεσιμότητα πόρων, η εικονικοποίηση και η ασφάλεια του δικτύου. Η δικτύωση που καθορίζεται από το λογισμικό δεν είναι η σωστή προσέγγιση για κάθε περιβάλλον δικτύου, αλλά όταν υπάρχουν σαφή οφέλη, το SDN μπορεί να είναι η καλύτερη λύση για την βελτιστοποίηση του δικτύου μας.



Σχήμα 2.3: Οφέλη δικτύων που καθορίζονται από λογισμικό

## 2.2.2 ΕΦΑΡΜΟΓΕΣ ΤΩΝ SDN

Έρευνα Internet: Δεδομένου ότι το Διαδίκτυο είναι ένα ζωντανό δίκτυο και χρησιμοποιείται συνεχώς, θα είναι δύσκολο να γίνουν ενημερώσεις ή δοκιμές για νέες ιδέες που μπορεί να λύσουν τα ζητήματα ή τα προβλήματα που αντιμετωπίζει η υπάρχουσα υποδομή διαδικτύου. Με το SDN έχουμε περισσότερο έλεγχο δεδομένου ότι το τμήμα ελέγχου του δικτύου και η κίνηση δεδομένων διαχωρίζονται, ή με άλλα λόγια διαχωρίζεται το hardware από το software. Αυτός ο διαχωρισμός επιτρέπει τη δοκιμή νέων ιδεών για τη μελλοντική αρχιτεκτονική του Διαδικτύου πριν την ενσωμάτωση του στο ζωντανό δίκτυο. [11]

Εξισορρόπηση φορτίου για διακομιστές εφαρμογών: Η εξισορρόπηση φορτίου είναι απαραίτητη προϋπόθεση για τα δίκτυα επιχειρήσεων, ώστε να παρέχεται υψηλή

διαθεσιμότητα και δυνατότητα κλιμάκωσης για τα αιτήματα μιας συγκεκριμένης υπηρεσίας. Κανονικά, αυτή η λειτουργία εξισορρόπησης φορτίων μεταξύ διαφόρων διακομιστών υλοποιείται από μια ειδική συσκευή που είναι ενσωματωμένη στο δίκτυο. Με το SDN ένας OpenFlow switch μπορεί να εκτελέσει αυτή τη λειτουργία αυτόματα και θα διανείμει την κίνηση σε διαφορετικούς διακομιστές. Είναι δυνατόν να γραφτεί μια εφαρμογή που να λειτουργεί πάνω από τον ελεγκτή και να μπορεί να προσφέρει μια κλιμακούμενη και αποδοτική εξισορρόπηση φορτίου [12]. Με μια τέτοια εφαρμογή η ανάγκη για μια ειδική συσκευή στο δίκτυο θα εξαλειφθεί.

Αναβάθμιση των κέντρων δεδομένων: Τα κέντρα δεδομένων είναι ένα ουσιαστικό εσωτερικό κομμάτι πολλών εταιρειών μεγάλης κλίμακας. Για παράδειγμα, η Google, το Facebook, το Amazon και το Yahoo έχουν μεγάλο αριθμό data centers για να καλύψουν τον τεράστιο αριθμό αιτήσεων και να ανταποκρίνονται γρήγορα. Αυτά τα κέντρα δεδομένων είναι εξαιρετικά ακριβά και περίπλοκα στη διατήρηση και την λειτουργία τους, το SDN και το OpenFlow επιτρέπουν στις εταιρείες να μειώσουν το κόστος της εγκατάστασης και της διαμόρφωσης του κέντρου δεδομένων, δεδομένου ότι τα κομμάτια που προωθούν τα δεδομένα του δικτύου μπορούν να διαχειριστούν από μια κεντρική τοποθεσία [13].

Υπηρεσίες ασφαλείας: Το σύγχρονο οικοσύστημα εικονικοποίησης υποστηρίζει συγκεκριμένες εικονικές υπηρεσίες που εκτελούνται στο επίπεδο του δικτύου. Αυτό σημαίνει την ενσωμάτωση λειτουργιών όπως το NFV σε πλατφόρμες SDN. Αυτός ο τύπος ασφάλειας δικτύου δημιουργεί ένα πραγματικά ενεργό περιβάλλον ικανό να μειώσει τον κίνδυνο και να ανταποκριθεί σε συμβάντα πολύ πιο γρήγορα. Όταν συμβαίνει παραβίαση, κάθε δευτερόλεπτο είναι κρίσιμο για τη διακοπή της επίθεσης. Επίσης σημαντική είναι η δυνατότητα εντοπισμού της επίθεσης και η διασφάλιση ότι τα υπόλοιπα στοιχεία του δικτύου είναι ασφαλή. Καθώς το επίπεδο του δικτύου καθίσταται ακόμα πιο κρίσιμο - και καθώς η σύγχρονη οργάνωση γίνεται ακόμη πιο ψηφιοποιημένη - θα δούμε περισσότερες επιθέσεις και πιο εξελιγμένες προχωρημένες απειλές. Με την ενσωμάτωση ισχυρών υπηρεσιών ασφαλείας στο επίπεδο SDN, δημιουργείται ένα πιο προνοητικό περιβάλλον που είναι σε θέση να ανταποκριθεί στην αλλαγή [14].

Δίκτυο πληροφοριών και παρακολούθησης: Οι σύγχρονες τεχνολογίες SDN βοηθούν στην αφαίρεση ενός από το πιο κρίσιμου επιπέδου των data centers: του δικτύου. Οι

αρχιτεκτονικές δικτύων είναι πολύ πιο σύνθετες και πρέπει να χειρίζονται περισσότερα δεδομένα από ποτέ. Αυτό σημαίνει ότι το να γνωρίζουμε τι περνά μέσα από το περιβάλλον μας είναι κρίσιμο. Υπάρχει καθυστέρηση σε μια θύρα; Τι συμβαίνει αν χρησιμοποιούμε μια ετερογενή αρχιτεκτονική δικτύου; Όλες αυτές οι προκλήσεις εξαφανίζονται όταν έχουμε μια σταθερή νοημοσύνη δικτύου και ένα επίπεδο παρακολούθησης. Ωστόσο, αποκτάμε πραγματική εικόνα και όφελος ενσωματώνοντας αυτές τις τεχνολογίες στην αρχιτεκτονική SDN. Η ροή της κυκλοφορίας, οι διαμορφώσεις των θυρών, η ενσωμάτωση του hypervisor, η προειδοποιήσεις ή ακόμα και η βελτιστοποίηση μπορούν να ενσωματωθούν στις τεχνολογίες δικτύων πληροφοριών και παρακολούθησης. Πάνω απ' όλα, αυτοί οι τύποι ευκίνητων συστημάτων θα μας βοηθήσουν να παρακολουθήσουμε περαιτέρω την κίνηση του δικτύου μεταξύ του κέντρου δεδομένων μας και του cloud [14].

Συμμόρφωση και εφαρμογές που συνδέονται με τον κανονισμό: Οι κύριοι προμηθευτές cloud προσφέρουν τη δυνατότητα αποθήκευσης και εργασίας σε σχέση με τον εκάστοτε κανονισμό. Τώρα, οι οργανώσεις έχουν την επιλογή να επεκτείνουν αρχιτεκτονικές που αρχικά ήταν πολύ περιορισμένες λόγω κανονισμών σε κατανεμημένα περιβάλλοντα και το cloud. Αλλά πώς τμηματοποιείται η κυκλοφορία; Πώς διασφαλίζεις ότι τα πακέτα που υπόκεινται στη συμμόρφωση (compliance) και την κανονιστική ρύθμιση (regulation) είναι συνεχώς ασφαλή και επιβλέπονται; Αυτό είναι κάτι που το SDN μπορεί να βοηθήσει. Η κίνηση του δικτύου που μεταφέρεται μεταξύ διακόπτων (switch), σημείων δικτύου ή ακόμη και υπερμεταφορέων (hypervisor) μπορεί να ελεγχθεί σε μια αρχιτεκτονική SDN. Αρκεί να θυμηθούμε ότι αυτό το επίπεδο αφαιρεί εικονικές λειτουργίες και στοιχεία ελέγχου του hardware. Αυτό το ισχυρό στρώμα μπορεί να εκτείνεται σε διάφορες τοποθεσίες, σημεία virtualization, ακόμα και τοποθεσίες cloud [14].

Εφαρμογές υψηλής απόδοσης. Βλέπουμε μια ανάπτυξη σε νέους τύπους εφαρμογών τεχνολογίας. Η εικονικοποίηση επέτρεψε την παράδοση πλούσιων εφαρμογών όπως το GIS, CAD, λογισμικό μηχανικής και γραφικών. Παραδοσιακά, αυτοί οι φόρτοι εργασίας απαιτούσαν bare metal αρχιτεκτονικές (αρχιτεκτονική που το VM είναι εγκατεστημένο στο hardware και όχι λειτουργικό σύστημα) με τη δική τους σύνδεση. Ωστόσο, με το virtualization, οι εφαρμογές πλέον μεταδίδονται ως stream και το VDI (virtual desktop infrastructure) μπορεί να βοηθήσει στη δημιουργία ισχυρών εμπειριών επιφάνειας εργασίας. Ωστόσο, στο επίπεδο του δικτύου βλέπουμε επίσης την ενσωμάτωση του SDN στον έλεγχο

εφαρμογών. Δημιουργία ισχυρών πολιτικών QoS, εξασφάλιση εμπιστευτικών δεδομένων, κατάτμηση βαρειάς κυκλοφορίας, ακόμη και δημιουργία ειδοποιήσεων κατωφλίου γύρω από σημεία συμφόρησης. Όλες αυτές οι λειτουργίες εντός του SDN συμβάλλουν στην υποστήριξη πλούσιων εφαρμογών υψηλής απόδοσης που παρέχονται μέσω virtualization [14].

Καταναμημένος έλεγχος εφαρμογών και ενσωματοποίηση του Cloud: Ένα από τα μεγαλύτερα πλεονεκτήματα του SDN είναι η ικανότητά του να επεκτείνεται σε ολόκληρο το κέντρο δεδομένων. Αυτός ο τύπος ευκινησίας ενσωματώνει καταναμημένες τοποθεσίες, το cloud και ολόκληρο τον οργανισμό . Το SDN επιτρέπει τη διέλευση κρίσιμης κυκλοφορίας δικτύου μεταξύ διαφόρων τοποθεσιών, ανεξάρτητα από τον τύπο της υποκείμενης αρχιτεκτονικής δικτύου. Αν αφαιρέσουμε κρίσιμα στοιχεία ελέγχου δικτύου, μπορούμε να διευκολύνουμε την κίνηση δεδομένων μεταξύ του κέντρου δεδομένων και των τοποθεσιών του cloud. Επειδή το SDN είναι μια μορφή εικονικοποίησης δικτύου, μπορούμε να χρησιμοποιήσουμε ισχυρά API's για να ενσωματώσουμε όχι μόνο έναν πάροχο cloud, αλλά επίσης να ελέγξουμε συγκεκριμένες υπηρεσίες δικτύου. Αυτό μας επιτρέπει να διαχειριζόμαστε λεπτομερώς τους φόρτους εργασίας μας διατηρώντας παράλληλα την επιχείρησή μας ευκίνητη [14].

## ΚΕΦΑΛΑΙΟ 3

### 3 OPENFLOW

Στην υπάρχουσα υποδομή, τα καθήκοντα που αποτελούν τη συνολική λειτουργικότητα του δικτύου όπως η δρομολόγηση (routing), η μεταγωγή (switching) ή οι αποφάσεις πρόσβασης στο δίκτυο αντιμετωπίζονται από συσκευές δικτύου που είναι από διάφορους διαφορετικούς προμηθευτές που έχουν διαφορετικό firmware. Αυτό δεν δίνει αρκετό χώρο για νέες ερευνητικές ιδέες. Νέοι αλγόριθμοι δρομολόγησης δεν μπορούν να δοκιμαστούν σε μεγάλης κλίμακας πραγματικά δίκτυα. Επιπλέον, κάθε προσπάθεια για πειραματικές ιδέες πάνω σε κριτικά σημεία του δικτύου μπορεί να οδηγήσουν σε κατάρρευση του ίδιου του δικτύου. Τα παραπάνω έχουν οδηγήσει στη στασιμότητα της υποδομής του δικτύου και έλλειψη προσαρμοστικότητας του, λόγος ο οποίος είναι ικανός για να μην υλοποιούνται σημαντικές καινοτομίες [15].

Το OpenFlow είναι μια προσέγγιση για την αντιμετώπιση αυτού του προβλήματος. Επειδή οι φορείς εκμετάλλευσης δικτύων μπορούν να εφαρμόζουν και να ελέγχουν τα χαρακτηριστικά που θέλουν στο λογισμικό, αντί να περιμένουν για έναν πωλητή να το σχεδιάσει στα ιδιόκτητα προϊόντα του. Επιτρέπει επίσης στους προμηθευτές να προσφέρουν στους ερευνητές πρόσβαση στον εξοπλισμό τους με ενιαίο τρόπο ώστε οι ερευνητές μπορούν να διεξάγουν πειράματα με νέα πρωτόκολλα σε ένα δίκτυο πραγματικού κόσμου χωρίς να επηρεάζεται η παραγόμενη κίνηση.

Το OpenFlow χρησιμοποιεί πίνακες ροής που είναι παρόμοιοι με τους πίνακες αναζήτησης στους σύγχρονους Ethernet διακόπτες και δρομολογητές. Αυτοί οι πίνακες ροής μπορούν να ενσωματώσουν τείχη προστασίας, NAT, QoS ή να συλλέγουν στατιστικά στοιχεία για τη διαχείριση δικτύου χωρίς να ασχολούνται με τους προμηθευτές του εξοπλισμού. Αυτοί οι πίνακες ροής περιέχουν κανόνες αντιστοίχισης / δράσης που μπορούν να δημιουργηθούν και τροποποιηθούν από έναν κεντρικό ελεγκτή. Ο ελεγκτής προσφέρει ένα προγραμματιστικό έλεγχο των ροών, ώστε ο διαχειριστής του δικτύου να μπορεί να ορίσει μια συγκεκριμένη διαδρομή από την πηγή στον προορισμό χρησιμοποιώντας την επεξεργασία των διαβιβαζόμενων πακέτων με βάση τη ροή. Επιπλέον μειώνεται η κατανάλωση ενέργειας και του κόστους διαχείρισης του δικτύου.

Το OpenFlow είναι ένα πρωτόκολλο που επιτρέπει στο διακομιστή να λέει στους διακόπτες του δικτύου το πού να σταλθούν τα πακέτα. Σε ένα συμβατικό δίκτυο, κάθε διακόπτης διαθέτει ιδιόκτητο λογισμικό που του λέει τι πρέπει να κάνει. Με το OpenFlow, οι αποφάσεις μεταφοράς πακέτων είναι συγκεντρωμένες, έτσι ώστε το δίκτυο να μπορεί να προγραμματιστεί ανεξάρτητα από τους μεμονωμένους διακόπτες και το κέντρο δεδομένων.

Σε ένα συμβατικό διακόπτη, η διαβίβαση πακέτων (η διαδρομή δεδομένων) και η δρομολόγηση υψηλού επιπέδου (η διαδρομή ελέγχου) εμφανίζονται στην ίδια συσκευή. Ένας διακόπτης OpenFlow διαχωρίζει τη διαδρομή δεδομένων από τη διαδρομή ελέγχου. Το τμήμα διαδρομής δεδομένων βρίσκεται στον διακόπτη ενώ ένας ξεχωριστός ελεγκτής παίρνει τις αποφάσεις δρομολόγησης υψηλού επιπέδου. Ο διακόπτης και ο ελεγκτής επικοινωνούν μέσω του πρωτοκόλλου OpenFlow. Αυτή η μεθοδολογία, γνωστή ως δικτύωση καθορισμένη από λογισμικό (SDN), επιτρέπει αποτελεσματικότερη χρήση των πόρων του δικτύου σε σχέση με τα παραδοσιακά δίκτυα. Το OpenFlow έχει κερδίσει προβάδισμα σε VM (εικονική μηχανή) εφαρμογές, σε δίκτυα κρίσιμων αποφάσεων και τα κινητά δίκτυα επόμενης γενιάς IP.

Αρκετές αναγνωρισμένες εταιρείες, όπως η IBM, η Google και η HP, είτε χρησιμοποίησαν πλήρως είτε ανακοίνωσαν την πρόθεσή τους να υποστηρίξουν το πρότυπο OpenFlow. Η Big Switch Networks, μια εταιρία SDN που εδρεύει στο Palo Alto της Καλιφόρνιας, έχει υλοποιήσει δίκτυα OpenFlow που λειτουργούν πάνω από τα παραδοσιακά δίκτυα και καθιστώντας δυνατή την τοποθέτηση εικονικών μηχανών οπουδήποτε σε ένα κέντρο δεδομένων για την πλήρη αξιοποίηση της υπάρχουσας υπολογιστικής ικανότητας. Μέχρι τις αρχές του 2012, το εσωτερικό δίκτυο της Google έτρεξε εξ ολοκλήρου στο OpenFlow.

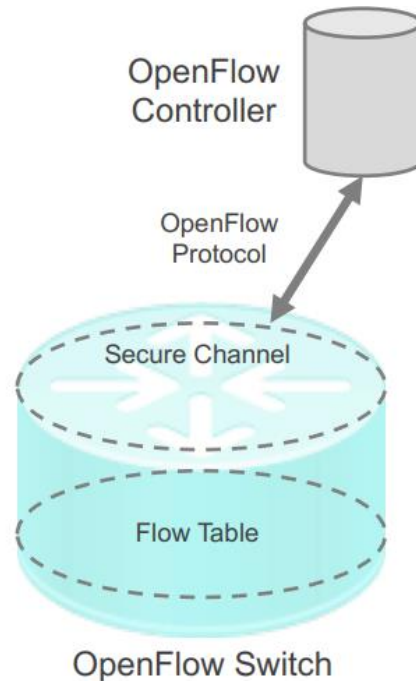
### 3.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ OPENFLOW

Η αρχιτεκτονική του δικτύου OpenFlow αποτελείται από τρεις βασικές έννοιες:

1. Διακόπτες συμβατοί με το OpenFlow που συνθέτουν το επίπεδο δεδομένων.
2. Το επίπεδο ελέγχου αποτελείται από έναν ή περισσότερους OpenFlow ελεγκτές
3. Ένα ασφαλές κανάλι ελέγχου συνδέει τους διακόπτες με το επίπεδο ελέγχου.



Οι διακόπτες επικοινωνούν με τους κεντρικούς υπολογιστές και μεταξύ τους χρησιμοποιώντας τη διαδρομή δεδομένων που παρέχει το λογισμικό και ο ελεγκτής επικοινωνεί με τους διακόπτες χρησιμοποιώντας τη διαδρομή δεδομένων όπως φαίνεται στο σχήμα 3.1.



Σχήμα 3.1: Αρχιτεκτονική OpenFlow

Η σύνδεση μεταξύ του ελεγκτή OpenFlow και του διακόπτη είναι ασφαλής χρησιμοποιώντας SSL ή TLS κρυπτογραφικά πρωτόκολλα, στα οποία ο διακόπτης και ο ελεγκτής πιστοποιούν αμοιβαία την γνησιότητα με το να ανταλλάζουν πιστοποιητικά που υπογράφονται από το ιδιωτικό κλειδί των δύο πλευρών. Αν και αυτός είναι ένας πολύ ισχυρός αλγόριθμος ασφαλείας, ο ελεγκτής μπορεί να είναι ευάλωτος σε επιθέσεις άρνησης υπηρεσίας (DoS), ή σε επιθέσεις που αναφέρονται ως Man in the middle, επομένως επιπλέον πρακτικές πρέπει να εφαρμοστούν για την πρόληψη τέτοιων επιθέσεων.

### 3.2 OPENFLOW ΔΙΑΚΟΠΤΕΣ

Ένας διακόπτης OpenFlow είναι ένα πρόγραμμα λογισμικού ή ένα hardware που προωθεί τα πακέτα σε ένα περιβάλλον δικτύου (SDN) καθορισμένο από το λογισμικό. Οι διακόπτες OpenFlow βασίζονται είτε στο πρωτόκολλο OpenFlow είτε είναι συμβατοί με αυτό. Ένας διακόπτης OpenFlow (σχήμα 3.1) αποτελείται από:

- Πίνακα ροής:

- σύνολο καταχωρήσεων ροής που καθορίζουν τις συνθήκες αντιστοίχισης πακέτων και τις επακόλουθες ενέργειες

Ένας διακόπτης στο δίκτυο OpenFlow έχει έναν (ή περισσότερους) πίνακες ροής που περιέχουν ένα σύνολο καταχωρήσεων, κάθε μία από τις οποίες αποτελείται από πεδία (σχήμα.3.2) όπως: κριτήρια αντιστοίχισης πακέτων (π.χ. πεδία κεφαλίδων (header fields) για αντιστοίχιση με πακέτα), μηδενικές ή περισσότερες ενέργειες (actions) για την εφαρμογή πακέτων που ταιριάζουν με κάποια αντιστοίχιση, μετρητές δραστηριότητας (counters) που ενημερώνονται για τα αντίστοιχα πακέτα

Όλα τα πακέτα που επεξεργάζεται ο διακόπτης συγκρίνονται με τον πίνακα ροής.

Header Fields	Actions	Counters
Flow entry 1	Forward to port 1/1	
Flow entry 2	Drop	
Flow entry $n$	Send to controller	

Σχήμα 3.2 Πεδία πίνακα ροής

- Ασφαλές κανάλι:

- κανάλι σε έναν εξωτερικό ελεγκτή ο οποίος διαχειρίζεται τον διακόπτη χρησιμοποιώντας το πρωτόκολλο OpenFlow

Το ασφαλές κανάλι (secure channel - SC) είναι η διεπαφή που συνδέει κάθε διακόπτη OpenFlow με τον ελεγκτή. Ένας ελεγκτής ρυθμίζει και διαχειρίζεται το διακόπτη μέσω αυτής της διασύνδεσης, λαμβάνει γεγονότα από το διακόπτη και στέλνει πακέτα από το διακόπτη.

Το ασφαλές κανάλι εγκαθιστά και τερματίζει τη σύνδεση μεταξύ του διακόπτη OpenFlow και του ελεγκτή χρησιμοποιώντας τις διαδικασίες: Ρύθμιση σύνδεσης, Διακοπή σύνδεσης. Τέλος σύνδεση SC είναι μια σύνδεση TLS. Ο διακόπτης και ο ελεγκτής αλληλεξαρτώνται με την ανταλλαγή πιστοποιητικών που υπογράφονται από έναν συγκεκριμένο ιδιωτικό κλειδί.

### 3.2.1 ΠΕΔΙΑ ΠΙΝΑΚΩΝ ΡΟΗΣ

Το πεδίο της κεφαλίδας του πίνακα ροής αποτελείται από επιμέρους πεδία στα οποία τα εισερχόμενα πακέτα συγκρίνονται με:

- Θύρα εισερχόμενου διακόπτη
- IEEE 802.3 Ethernet πηγή και διεύθυνση προορισμού
- IEEE 802.3 τύπος Ethernet
- IEEE 802.1Q VLAN ID και προτεραιότητα
- Πηγή προέλευσης IP και διεύθυνση προορισμού
- Πεδίο πρωτοκόλλου IP
- Τύπο υπηρεσίας IP (TOS)
- TCP / UDP πηγές και θύρες προορισμού

Εάν μια κεφαλίδα ενός πακέτου ταιριάζει με μια καταχώρηση ροής, εκτελείται μια ενέργεια για αυτήν την καταχώρηση στο πακέτο (π.χ., η ενέργεια ενδέχεται να είναι η προώθηση ενός πακέτου από μια συγκεκριμένη θύρα). Εάν δεν βρεθεί μια αντιστοίχιση, το πακέτο μέσω ενός ασφαλούς καναλιού προωθείται στον ελεγκτή.

Τα εισερχόμενα πακέτα μπορούν να συνδυαστούν με διάφορα πεδία σε κάθε στρώμα του μοντέλου OSI για ένα πακέτο, που κυμαίνεται από το σύνδεσμο δεδομένων έως το στρώμα μεταφοράς καθώς και την θύρα του διακόπτη. Προκειμένου να ταιριάξουν όλα τα πεδία της επικεφαλίδας η ειδική τιμή ANY μπορεί να χρησιμοποιηθεί στον πίνακα ροής.

Εάν ένα πακέτο εισόδου αντιστοιχεί σε ένα από τα πεδία αντιστοίχισης στον πίνακα ροής, κάποια ενέργεια θα πρέπει να γίνει σε αυτό το πακέτο. Η προώθηση των πακέτων σε κάθε φυσική θύρα πρέπει να υποστηρίζεται από το διακόπτη OpenFlow. Επιπλέον, υπάρχουν εικονικές θύρες που ορίζονται από το πρότυπο OpenFlow ως ειδικοί στόχοι που μπορούν να

μεταφερθούν τα πακέτα. Αυτές οι ενέργειες στις εικονικές θύρες ορίζονται ως "απαιτούμενες" ενέργειες και "προαιρετικές" ενέργειες. Οι απαιτούμενες ενέργειες πρέπει να υποστηρίζονται από όλους τους διακόπτες ώστε να είναι συμβατοί με το OpenFlow και οι προαιρετικές ενέργειες έχουν αποδειχθεί ότι είναι χρήσιμες, αλλά δεν υποστηρίζονται αναγκαία από έναν συμβατό με το OpenFlow διακόπτη [16].

Απαιτούμενες ενέργειες είναι οι εξής:

ALL: Προώθηση του πακέτου σε όλες τις θύρες εκτός από την θύρα που λήφθηκε

CONTROLLER: Συμπύκνωση του πακέτου και αποστολή του στον ελεγκτή

TABLE: Εκτέλεση ενεργειών που περιγράφονται στον πίνακα ροής. Μόνο για μηνύματα εξερχομένων πακέτων

LOCAL: Αποστολή του πακέτου στην τοπική στοίβα δικτύου για το διακόπτη

IN\_PORT: Αποστολή του πακέτου στην λαμβανόμενη θύρα

Προαιρετικές ενέργειες είναι οι εξής:

NORMAL: Προώθηση του πακέτου χρησιμοποιώντας το παραδοσιακές μεθόδους προώθησης, δηλ. τα παραδοσιακά L2, VLAN και L3 επεξεργασία

FLOOD: Αποστολή του πακέτου κατά το ελάχιστο spanning tree, χωρίς να συμπεριλαμβάνεται η εισερχόμενη διεπαφή. Κάθε θύρα των διακοπών που είναι ενεργοποιημένα από το OpenFlow έχουν ένα NO\_FLOOD bit, το οποίο δείχνει ότι η θύρα δεν ανήκει στο ελάχιστο spanning tree. Τα πακέτα που ταιριάζουν σε αυτή την εγγραφή της ροής προωθούνται στις θύρες που έχουν NO\_FLOOD bit

Εκτός από τις ενέργειες προώθησης, υπάρχουν και άλλες ενέργειες στον πίνακα ροής.

Drop: Μια απαιτούμενη ενέργεια που υποδεικνύεται από μια κενή λίστα ενεργειών. Όλα τα πακέτα που ταιριάζουν με μια κενή λίστα ενεργειών απορρίπτονται

Enqueue: Αυτή η προαιρετική ενέργεια μπορεί να χρησιμοποιηθεί για να θέσει τα πακέτα σε μια ουρά που συσχετίζεται με μια θύρα προκειμένου να διατεθεί κάποια ποιότητα υπηρεσίας (QoS)

Modify-field: Αυτή η προαιρετική ενέργεια χρησιμοποιείται για την τροποποίηση μιας συγκεκριμένης κεφαλίδας για το εισερχόμενο πακέτο

Τα πρότυπα OpenFlow επιτρέπουν στο διακόπτη να εκθέσει στατιστικά στοιχεία μέσω μετρητών. Οι μετρητές αποτελούνται από πολλές μεταβλητές ανά πίνακα, ροή, θύρα, ουρά. Οι μετρητές προορίζονται για τη συλλογή στατιστικών στοιχείων σχετικά με τις ροές. Αποθηκεύουν τον αριθμό των λαμβανόμενων πακέτων και bytes, καθώς και τη διάρκεια της ροής.

### 3.2.2 ΤΥΠΟΙ ΡΟΩΝ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ

Τα 2 υπάρχουσα μοντέλα ροής είναι η δρομολόγηση ροής (flow routing) και η συσσώρευση (aggregation). Και τα 2 μοντέλα είναι εφικτά με το OpenFlow ωστόσο, οι συγκεντρωτικοί κανόνες είναι απαραίτητοι για την αντιμετώπιση του ορίου που έχουμε από το hardware για τον αριθμό των καταχωρίσεων που επιβάλλονται από τα τρέχοντα σήματα TCAM [16].

Δρομολόγηση με βάση την ροή: Κάθε ροή δημιουργείται ξεχωριστά από τον ελεγκτή και υπάρχει ακριβής αντιστοιχία για να υπάρξει κάποια ενέργεια. Ο πίνακας ροής περιέχει μόνο μια εγγραφή ανά ροή. Χρησιμοποιείται κυρίως σε συστήματα μικροδιαχείρισης όπως π.χ. δίκτυα πανεπιστημιούπολεων.

Συσσωρευμένη δρομολόγηση: Κάθε ροή καλύπτει μια μεγάλη ομάδα ροών και η αντιστοιχία είναι μεγαλύτερου φάσματος. Ο πίνακας ροής περιέχει μια εγγραφή ανά κατηγορία ροών. Χρησιμοποιείται κυρίως σε μεγάλο αριθμό ροών όπως π.χ. στον κορμό του δικτύου.

Η διαχείριση κατηγοριοποιείται και αυτή σε 2 μοντέλα, στα αντιδραστικά (reactive) και τα προληπτικά (proactive) [17].

Reactive: Σε αυτό το μοντέλο ο ελεγκτής είναι αδρανής μέχρι να λάβει το πρώτο πακέτο από το διακόπτη. Ο ελεγκτής αναλύει το εισερχόμενο πακέτο και μετά εισάγει μια νέα καταχώρηση ροής στον πίνακα ροής του διακόπτη. Σε αυτόν τον τύπο κάθε νέα ροή χρειάζεται ένα μικρό επιπλέον χρόνο ρύθμισης. Εάν η σύνδεση μεταξύ του διακόπτη και του ελεγκτή αποτύχει και ο διακόπτης δεν έχει τη δυνατότητα να προωθήσει το πακέτο ως παραδοσιακός διακόπτης, τότε το πακέτο δεν θα μπορέσει να προωθηθεί στους κεντρικούς υπολογιστές.

Proactive: Σε αυτό το μοντέλο ο ελεγκτής προ-εγκαθιστά τις καταχωρήσεις ροής στον πίνακα ροής του διακόπτη χωρίς να απαιτείται η λήψη του πρώτου πακέτου της ροής. Δεν απαιτείται πρόσθετος χρόνος για τη ρύθμιση ροής και σε περίπτωση αποτυχίας σύνδεσης μεταξύ του ελεγκτή και του διακόπτη, η κυκλοφορία δεν θα διαταράσσεται. Ουσιαστικά όμως απαιτούνται συσσωρευμένες αντιστοιχίες μεγαλύτερου φάσματος ώστε να είναι λειτουργικό αυτό το μοντέλο.

### 3.3 ΕΠΙΣΚΟΠΗΣΗ OPENFLOW ΠΡΩΤΟΚΟΛΛΟΥ

Το πρωτόκολλο OpenFlow υποστηρίζει τρεις τύπους μηνυμάτων: ελεγκτή - διακόπτη, ασύγχρονα και συμμετρικά (τον καθένα με πολλαπλούς επιμέρους τύπους) . Τα μηνύματα ελεγκτή - διακόπτη ξεκινούν από τον ελεγκτή και χρησιμοποιούνται για την άμεση διαχείριση ή επιθεώρηση της κατάστασης του διακόπτη. Τα ασύγχρονα μηνύματα ξεκινούν από το διακόπτη και χρησιμοποιούνται για την ενημέρωση του ελεγκτή για τα συμβάντα του δικτύου και τις αλλαγές στην κατάσταση του διακόπτη. Τα συμμετρικά μηνύματα ξεκινούν είτε από το διακόπτη είτε από τον ελεγκτή και αποστέλλονται χωρίς παράκληση. Οι τύποι μηνυμάτων που χρησιμοποιούνται από το OpenFlow περιγράφονται παρακάτω.

#### 3.3.1 ΜΗΝΥΜΑΤΑ ΕΛΕΓΚΤΗ - ΔΙΑΚΟΠΤΗ

Τα μηνύματα του ελεγκτή - διακόπτη ξεκινούν από τον ελεγκτή και μπορεί ή όχι να απαιτούν απάντηση από το διακόπτη [18].

Μήνυμα χαρακτηριστικών: Κατά την δημιουργία της συνεδρίας Transport Layer Security (TLS), ο ελεγκτής στέλνει ένα μήνυμα αίτησης λειτουργιών στον διακόπτη. Ο διακόπτης πρέπει να ανταποκριθεί με μια απάντηση χαρακτηριστικών που καθορίζει τις δυνατότητες που υποστηρίζει ο διακόπτης.

Μήνυμα διαμόρφωσης: Ο ελεγκτής είναι σε θέση να ρυθμίσει και να ζητήσει παραμέτρους διαμόρφωσης στον διακόπτη. Ο διακόπτης ανταποκρίνεται μόνο σε κάποιο ερώτημα από τον ελεγκτή.

Μήνυμα κατάστασης τροποποίησης: Τα μηνύματα κατάστασης τροποποίησης αποστέλλονται από τον ελεγκτή για τη διαχείριση της κατάστασης των διακοπών. Ο κύριος

σκοπός τους είναι να προσθέσουν / διαγράψουν και να τροποποιήσουν τις ροές στους των πινάκων ροής και για να ορίσουν τις ιδιότητες των θυρών διακόπτη.

Μήνυμα κατάστασης ανάγνωσης: Τα μηνύματα κατάστασης ανάγνωσης χρησιμοποιούνται από τον ελεγκτή για τη συλλογή στατιστικών στοιχείων από τους πίνακες ροής των διακοπών, τις θύρες και τις καταχωρήσεις μεμονωμένων ροών.

Μήνυμα αποστολής πακέτων: Αυτά χρησιμοποιούνται από τον ελεγκτή για την αποστολή πακέτων από μια καθορισμένη θύρα του διακόπτη.

Μήνυμα φράγματος: Τα μηνύματα αιτήματος / απάντησης φράγματος χρησιμοποιούνται από τον ελεγκτή για να εξασφαλίσει ότι πληρούνται τα dependencies των μηνυμάτων ή για να λαμβάνει ειδοποιήσεις για ολοκληρωμένες λειτουργίες.

### 3.3.2 ΑΣΥΓΧΡΟΝΑ ΜΗΝΥΜΑΤΑ

Τα ασύγχρονα μηνύματα αποστέλλονται χωρίς να τα ζητάει ο ελεγκτής από ένα διακόπτη. Οι διακόπτες στέλνουν ασύγχρονα μηνύματα στον ελεγκτή για να δηλώσουν την άφιξη πακέτων, την αλλαγή αλλαγής κατάστασης ή το σφάλμα. Οι τέσσερις κύριοι τύποι ασύγχρονου μηνύματος περιγράφονται παρακάτω:

Packet-in: Για όλα τα πακέτα που δεν έχουν αντίστοιχη καταχώρηση ροής, ένα packet-in συμβάν αποστέλλεται στον ελεγκτή (ή εάν ένα πακέτο ταιριάζει σε μια καταχώρηση με ενέργεια "στείλε σε ελεγκτή"). Εάν ο διακόπτης διαθέτει επαρκή μνήμη για την μετάβαση των πακέτων που αποστέλλονται στον ελεγκτή, τα packet-in συμβάντα περιέχουν κάποιο κομμάτι της κεφαλίδας του πακέτου (από προεπιλογή 128 bytes) και ένα αναγνωριστικό buffer που θα χρησιμοποιηθεί από το ελεγκτή όταν είναι έτοιμος ο διακόπτης να προωθήσει το πακέτο. Οι διακόπτες που δεν υποστηρίζουν την εσωτερικό buffering (ή έχουν εξαντλήσει το εσωτερικό τους buffering) πρέπει να στείλουν το πλήρες πακέτο στον ελεγκτή ως μέρος του συμβάντος.

Flow-removed: Όταν μια καταχώρηση ροής προστίθεται στον διακόπτη με ένα μήνυμα τροποποίησης ροής, μια τιμή χρονικού ορίου αδράνειας υποδεικνύει τότε πρέπει να καταργηθεί η καταχώρηση λόγω έλλειψης δραστηριότητας. Υπάρχει επίσης μια προκαθορισμένη τιμή χρονικής υπέρβασης που δείχνει τότε η καταχώριση θα πρέπει να

καταργηθεί, ανεξάρτητα από τη δραστηριότητα. Το μήνυμα τροποποίησης ροής καθορίζει επίσης αν ο διακόπτης πρέπει να στείλει ένα μήνυμα αφαίρεσης ροής στον ελεγκτή όταν η ροή λήγει. Μηνύματα τροποποίησης ροής που διαγράφουν ροές μπορούν να προκαλέσουν μηνύματα αφαίρεσης ροής.

Port-status: Ο διακόπτης αναμένεται να στείλει μηνύματα κατάστασης της θύρας στον ελεγκτή ως αλλαγές κατάστασης διαμόρφωσης θύρας. Αυτά τα συμβάντα περιλαμβάνουν αλλαγή στη κατάσταση της θύρας (για παράδειγμα, εάν καταργήθηκε απευθείας από έναν χρήστη).

Error: Ο διακόπτης μπορεί να ειδοποιήσει τον ελεγκτή για προβλήματα χρησιμοποιώντας μηνύματα σφάλματος.

### 3.3.3 ΣΥΜΜΕΤΡΙΚΑ ΜΗΝΥΜΑΤΑ

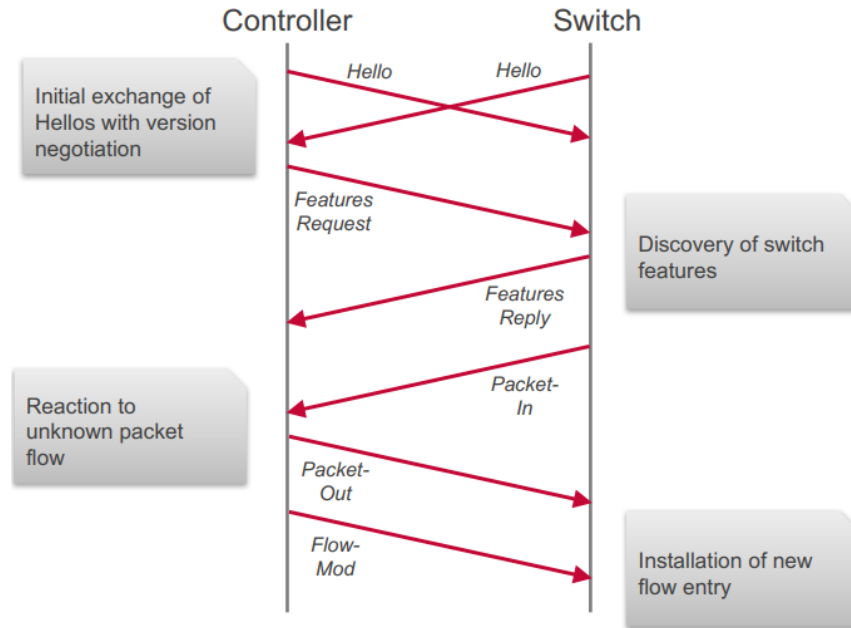
Τα συμμετρικά μηνύματα αποστέλλονται χωρίς πρόσκληση, προς οποιαδήποτε κατεύθυνση.

Hello: Μηνύματα hello ανταλλάσσονται μεταξύ του διακόπτη και του ελεγκτή κατά την έναρξη της σύνδεσης.

Echo: Τα μηνύματα αίτησης / απάντησης Echo μπορούν να σταλθούν είτε από το διακόπτη είτε από τον ελεγκτή και πρέπει να επιστρέψουν μια απάντηση echo. Μπορούν να χρησιμοποιηθούν για να υποδείξουν την καθυστέρηση, το εύρος ζώνης ή την ύπαρξη μιας σύνδεσης ελεγκτή-διακόπτη.

Vendor: Τα μηνύματα προμηθευτών παρέχουν έναν τυπικό τρόπο για τους διακόπτες OpenFlow να προσφέρουν επιπλέον λειτουργικότητα στο σύστημα μηνυμάτων του OpenFlow. Αυτή είναι μια πρόχειρη περιοχή για χαρακτηριστικά που προορίζονται για μελλοντικές αναθεωρήσεις του OpenFlow.





Σχήμα 3.3: Παράδειγμα μηνυμάτων ροής

### 3.4 OPENFLOW ΕΛΕΓΚΤΕΣ

Ένας ελεγκτής σε μια αρχιτεκτονική δικτύωσης (SDN) που καθορίζετε από λογισμικό είναι μια εφαρμογή που διαχειρίζεται τον έλεγχο ροής για βελτιωμένη διαχείριση δικτύου και απόδοση εφαρμογών. Ο ελεγκτής εκτελείται συνήθως σε κάποιον διακομιστή (server) και χρησιμοποιεί πρωτόκολλα για να ειδοποιεί τους διακόπτες (switches) που πρέπει να στέλνουν πακέτα. Οι ελεγκτές SDN κατευθύνουν την κυκλοφορία σύμφωνα με τις πολιτικές προώθησης που εφαρμόζει ένας διαχειριστής δικτύου, ελαχιστοποιώντας έτσι τις χειροκίνητες ρυθμίσεις για μεμονωμένες συσκευές δικτύου. Απομακρύνοντας το επίπεδο ελέγχου από το hardware του δικτύου και εκτελώντας το ως λογισμικό, ο κεντρικός ελεγκτής διευκολύνει την αυτοματοποιημένη διαχείριση του δικτύου και διευκολύνει την ενοποίηση και τη διαχείριση επιχειρηματικών εφαρμογών. Στην πραγματικότητα, ο ελεγκτής SDN λειτουργεί ως ένα είδος λειτουργικού συστήματος (OS) για το δίκτυο.

Ο ελεγκτής είναι ο πυρήνας ενός δικτύου καθοριζομένου από λογισμικό. Βρίσκεται μεταξύ των συσκευών του δικτύου και των εφαρμογών του. Οποιαδήποτε επικοινωνία μεταξύ εφαρμογών και συσκευών δικτύου πρέπει να περάσει από τον ελεγκτή. Επικοινωνεί με εφαρμογές - όπως τείχη προστασίας ή εξισορρόπησης φορτίου - μέσω northbound διεπαφών.

Το Ίδρυμα Ανοιχτής Διασύνδεσης (ONF) δημιούργησε μια ομάδα εργασίας το 2013 επικεντρωμένη ειδικά στα northbound API's και στην ανάπτυξή τους. Ο χώρος δεν έχει κατασταλάξει ακόμα σε ένα τυποποιημένο σύνολο, κυρίως επειδή οι απαιτήσεις των εφαρμογών ποικίλλουν ευρέως. Ο ελεγκτής μιλάει με επιμέρους συσκευές δικτύου χρησιμοποιώντας southbound διεπαφές, όπως το πρωτόκολλο OpenFlow. Αυτά τα southbound πρωτόκολλα επιτρέπουν στον ελεγκτή να διαμορφώνει συσκευές δικτύου και να επιλέγει τη βέλτιστη διαδρομή δικτύου για την κίνηση των εφαρμογών.

Ένα σημαντικό πλεονέκτημα των ελεγκτών SDN είναι ότι ο κεντρικός ελεγκτής γνωρίζει όλα τα διαθέσιμα μονοπάτια του δικτύου και μπορεί να κατευθύνει τα πακέτα με βάση τις απαιτήσεις της κυκλοφορίας. Μπορεί να τροποποιήσει αυτόματα τις ροές κυκλοφορίας και να ειδοποιήσει τους φορείς εκμετάλλευσης δικτύου σχετικά με τους συμφορημένους συνδέσμους. Οι εταιρείες μπορούν - και πρέπει - να χρησιμοποιούν περισσότερους από έναν ελεγκτές, προσθέτοντας έναν backup για εφεδρεία. Τρεις φαίνεται να είναι ένας κοινός αριθμός τόσο μεταξύ εμπορικών SDN όσο και ανοικτών πηγών SDN . Αυτή η εφεδρεία θα επιτρέψει στο δίκτυο να συνεχίσει να λειτουργεί σε περίπτωση απώλειας συνδεσιμότητας ή ευπάθειας του ελεγκτή. Το ότι ο ελεγκτής είναι το κεντρικό σημείο του δικτύου, σημαίνει ότι η ασφάλειά του είναι ζωτικής σημασίας για οποιοδήποτε δίκτυο που έχει καθοριστεί από το λογισμικό. Οποιος κατέχει τον ελεγκτή έχει πρόσβαση σε ολόκληρο το δίκτυο. Αυτό σημαίνει ότι οι φορείς εκμετάλλευσης δικτύων πρέπει να δημιουργήσουν πολιτικές ασφάλειας και ελέγχου ταυτότητας για να εξασφαλίσουν ότι έχουν πρόσβαση μόνο οι κατάλληλοι άνθρωποι.

Προμηθευτές που προσφέρουν SDN ελεγκτές είναι οι: Big Switch Networks, Cisco, Cumulus Networks, Hewlett Packard Enterprise, Juniper Networks, Nuage Networks, Pluribus Networks, VMware και οι Open Source ελεγκτές που θα μελετήσουμε εμείς είναι οι: Nox, Pox, Ryu, OpenMul, Libfluid\_Raw, Beacon, Iris, Maestro, Onos.

### 3.4.1 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ OPENFLOW ΕΛΕΓΚΤΩΝ

#### 1) Γλώσσα προγραμματισμού

Η πρόσβαση σε πολλές πλατφόρμες, η επιλογή του multithreading, η ευκολία μάθησης, η γρήγορη πρόσβαση στη μνήμη και η καλή διαχείριση μνήμης είναι βασικά χαρακτηριστικά των γλωσσών προγραμματισμού. Όταν επιλέγουμε ένα συγκεκριμένο ελεγκτή, πρέπει να λάβουμε υπόψη αυτούς τους παράγοντες επειδή επηρεάζουν την απόδοση του ελεγκτή και την ταχύτητα ανάπτυξης. Η Python, η C ++ και η Java είναι οι περισσότερο χρησιμοποιούμενες γλώσσες για προγραμματισμό ελεγκτών SDN. Γενικότερα, οι ελεγκτές που είναι γραμμένοι σε Java έχουν το χαρακτηριστικό ότι μπορούν να τρέξουν σε πολλές πλατφόρμες και να παρουσιάζουν καλό modularity, οι ελεγκτές που είναι γραμμένοι σε C παρέχουν υψηλή απόδοση αλλά στερούνται υψηλού επιπέδου modularity, έχουν καλή διαχείριση μνήμης και καλό GUI και οι ελεγκτές που είναι γραμμένοι σε Python στερούνται το multithreading.

#### 2) Υποστήριξη OpenFlow

Το πρωτόκολλο OpenFlow αποτελεί βασικό στοιχείο για τα δίκτυα καθοριζόμενα από το λογισμικό. Ήταν το πρώτο τυποποιημένο southbound interface . Επιτρέπει τον άμεσο χειρισμό του επίπεδου προώθησης των OpenFlow διακοπών [19]. Όταν επιλέγουμε ένα ελεγκτή OpenFlow, πρέπει να γνωρίζουμε το OpenFlow που υποστηρίζει ο ελεγκτής καθώς και το αν πρόκειται να αναβαθμιστεί σε μια νεότερη έκδοση, όπως το Openflow v1.3 ή v1.4. Ένας λόγος για να λαμβάνουμε αυτό υπόψη είναι ότι σημαντικές λειτουργίες όπως για παράδειγμα η υποστήριξη IPv6, δεν αποτελεί μέρος του OpenFlow v1.0 αλλά είναι μέρος του OpenFlow v1.3.

#### 3) Προγραμματισμός δικτύου

Η προγραμματισιμότητα του δικτύου είναι το πιο σημαντικό πλεονέκτημα των SDN στην αντιμετώπιση της σημερινής πολυπλοκότητας διαχείρισης του δικτύου (τεράστιος αριθμός συνδεδεμένων συσκευών, ανάπτυξη νέων υπηρεσιών). Χρησιμοποιώντας το μοντέλο συσκευής ανά συσκευή, η διαχείριση μεγάλης κλίμακας των μελλοντικών δικτύων δεν θα είναι εφικτή. Ο παλιός στατικός τρόπος της διαχείρισης των συσκευών του δικτύου είναι χρονοβόρος, επιρρεπής σε σφάλματα και οδηγεί σε ασυνέπειες. Το παράδειγμα της

διαχείρισης από λογισμικό έρχεται να κρύψει αυτές τις δυσκολίες διαχείρισης με την εισαγωγή αυτοματισμού και δυναμικής στην όλη διαδικασία. Αυτοματοποιημένα script μπορούν να εκτελεστούν μέσω CLI και οι εφαρμογές μπορούν να αναπτυχθούν στην πλατφόρμα του ελεγκτή για την εκτέλεση προκαθορισμένων εργασιών και λειτουργιών διαχείρισης. Η υποστήριξη του ελεγκτή στην προγραμματισιμότητα του δικτύου ουσιαστικά βασίζεται στην ενσωμάτωση ενός μεγάλου αριθμού northbound διεπαφών, σε ένα καλό γραφικό περιβάλλον για τον χρήστη και στη γραμμή εντολών (CLI) [20].

#### 4) Αποτελεσματικότητα (απόδοση, αξιοπιστία, δυνατότητα κλιμάκωσης και Ασφάλεια)

Η αποτελεσματικότητα του ελεγκτή είναι ένας όρος που χρησιμοποιείται για να αναφερθούμε στις διάφορες παραμέτρους - απόδοση, επεκτασιμότητα, αξιοπιστία και ασφάλεια. Διάφορες μετρήσεις, όπως ως αριθμός διεπαφών που μπορεί να χειριστεί ένας ελεγκτής, η καθυστέρηση, η απόδοση, κλπ. ορίζουν αυτό που ονομάζουμε απόδοση. Ομοίως, υπάρχουν διάφορες μετρήσεις που ορίζουν την επεκτασιμότητα, αξιοπιστία και ασφάλεια. Η συγκέντρωση του ελέγχου στο σύστημα SDN θα αποτελέσει μια σοβαρή πρόκληση από πλευράς αξιοπιστίας και τις προοπτικών απόδοσης. Το σύστημα κατανομής, υποστηριζόμενο από ορισμένους ελεγκτές, έχει ως στόχο να αντιμετωπίσει αυτό το θέμα [21].

#### 5) Διεπαφές Southbound

Τα southbound APIs επιτρέπουν τον αποτελεσματικό έλεγχο του δικτύου. Αυτά τα API χρησιμοποιούνται από τον ελεγκτή για να κάνει δυναμικές αλλαγές στους κανόνες προώθησης που είναι εγκατεστημένοι σε συσκευές στο επίπεδο δεδομένων όπως οι: διακόπτες, δρομολογητές, κ.λπ. Ενώ το OpenFlow είναι το πιο γνωστό από τα πρωτόκολλα SDN για τα southbound APIs, δεν είναι το μόνο διαθέσιμο ή σε εξέλιξη. NETCONF (τυποποιημένο από το IETF), OF-Config (υποστηρίζεται από το Open Network Foundation (ONF)), Orflex (υποστηρίζεται από τη Cisco) και άλλα παραδείγματα διεπαφών southbound χρησιμοποιούνται για τη διαχείριση συσκευών δικτύου. Επιπλέον, μερικά από τα πρωτόκολλα δρομολόγησης όπως IS-IS, OSPF, BGP έχουν αναπτυχθεί επίσης ως southbound διεπαφές σε ορισμένους ελεγκτές με στόχο την υποστήριξη υβριδικών δικτύων (SDN και μη SDN) ή για να εφαρμοστούν στην παραδοσιακή δικτύωση με ως μια προσέγγιση πιο πολύ βασισμένη σε λογισμικό [22].

## 6) Διεπαφές Northbound

Τα northbound APIs χρησιμοποιούνται από το επίπεδο των εφαρμογών για να επικοινωνούν με τον ελεγκτή. Είναι το πιο κρίσιμο μέρος στην αρχιτεκτονική του ελεγκτή SDN, δεδομένου η αξία του SDN συνδέεται με τις καινοτόμες εφαρμογές που μπορεί ενδεχομένως να υποστηρίξει και να ενεργοποιήσει. Το ότι είναι τόσο κρίσιμο σημαίνει ότι πρέπει να υποστηρίζουν μια ευρεία γκάμα εφαρμογών. Αυτά τα APIs επιτρέπουν επίσης τη σύνδεση με αυτοματοποιημένες στοίβες όπως το OpenStack ή το CloudStack που χρησιμοποιούνται για τη διαχείριση του Cloud. Το πρωτόκολλο μεταφοράς (REST) φαίνεται να είναι η χρησιμοποιημένη northbound διεπαφή και οι περισσότεροι ελεγκτές την έχουν ενσωματώσει.

## 7) Partnership

Μια καλή συνεργασία αυξάνει τις πιθανότητες για έναν SDN ελεγκτή να διατηρηθεί και να βελτιωθεί για μεγάλο χρονικό διάστημα [23]. Η εμπειρία στον τομέα των δικτύων και των υπολογιστών και οι οικονομικές δυνατότητες της συνεργαζόμενης εταιρίας είναι τα βασικά κριτήρια που προσθέτουν εμπιστοσύνη και χρησιμότητα στα προϊόντα. Η Cisco, το Ίδρυμα Linux, η Intel, η IBM, η Juniper κ.λπ. είναι παραδείγματα αξιόπιστων οργανώσεων που εισέρχονται στην αγορά SDN και που συμμετέχουν στην ανάπτυξη ελεγκτών.

Name	Programming Language	Architecture	Northbound API	Southbound API	EastWestbound API	Supported Platform	Interface	License	Multithreading	Modularity	Consistency	Documentation
Beacon	Java	Centralized	ad-hoc	OpenFlow 1.0	-	Linux, MacOS, Windows	CLI, Web UI	GPL 2.0	Yes	Fair	No	Fair
Maestro	Java	Centralized	ad-hoc	OpenFlow 1.0	-	Linux, MacOS, Windows	Web UI	LGPL 2.1	Yes	Fair	No	Limited
NOX	C++	Centralized	ad-hoc	OpenFlow 1.0	-	Linux	CLI, Web UI	GPL 3.0	Yes (Nox-MT)	Low	No	Limited
ONOS	Java	Distributed Flat	REST, Neutron	OpenFlow 1.0, 1.3	Raft	Linux, MacOS, Windows	CLI, Web UI	Apache 2.0	Yes	High	Yes	Good
OpenIRIS	Java	Distributed Flat	REST	OpenFlow 1.0-1.3	Custom Protocol	Linux	CLI, Web UI	Apache 2.0	Yes	Fair	No	Limited
POX	Python	Centralized	ad-hoc	OpenFlow 1.0	-	Linux, MacOS, Windows	CLI, GUI	Apache 2.0	No	Low	No	Limited
Ryu	Python	Centralized	REST	OpenFlow 1.0-1.5	-	Linux, MacOS	CLI	Apache 2.0	Yes	Fair	Yes	Good
OpenMul	C	Centralized	REST	OpenFlow 1.0, 1.3, OVSD, Netconf	-	Linux	CLI	GPL 2.0	Yes	High	No	Good

Σχήμα 3.4 Χαρακτηριστικά ελεγκτών

## 3.5 ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΔΕΣΗΣ

Ο διακόπτης πρέπει να είναι σε θέση να εγκαθιδρύσει την επικοινωνία σε μια παραμετροποιήσιμη από το χρήστη (αλλιώς σταθερή) διεύθυνση IP, χρησιμοποιώντας μια θύρα που καθορίζεται από το χρήστη. Η κυκλοφορία από και προς το ασφαλές κανάλι δεν

ελέγχεται σε σχέση με τον πίνακα ροής. Επομένως, ο διακόπτης πρέπει να αναγνωρίσει την εισερχόμενη κίνηση ως τοπική πριν την ελέγξει σε σχέση με τον πίνακα ροής. Οι μελλοντικές εκδόσεις της προδιαγραφής πρωτοκόλλου θα περιγράψουν έναν δυναμικό πρωτόκολλο ανακάλυψης του ελεγκτή στο οποίο η διεύθυνση IP και η θύρα για την επικοινωνία με τον ελεγκτή καθορίζεται κατά το χρόνο εκτέλεσης.

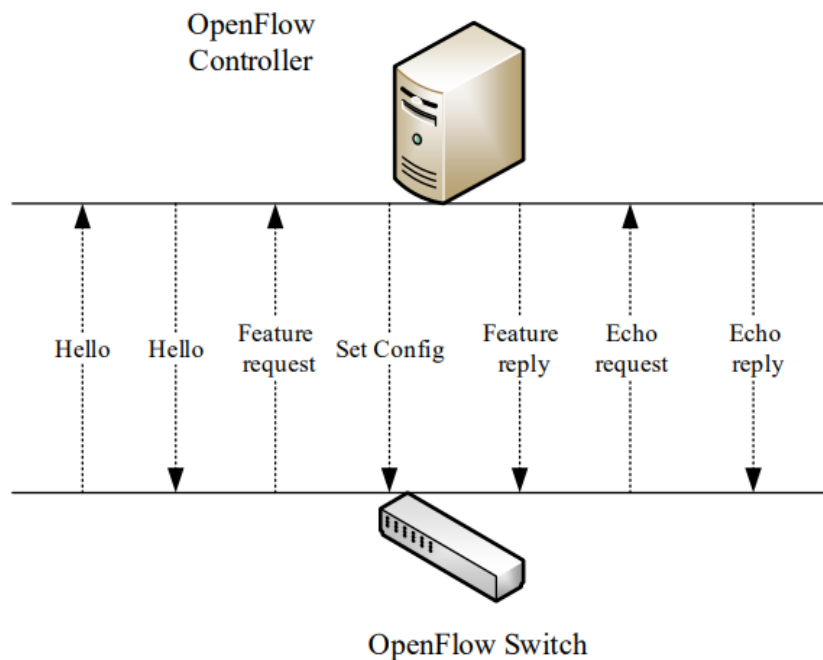
Όταν εγκαθίσταται μια OpenFlow σύνδεση, κάθε πλευρά της σύνδεσης πρέπει να στείλει αμέσως ένα μήνυμα OFPT\_HELLO με το πεδίο της έκδοσης ορισμένο στην υψηλότερη έκδοση πρωτοκόλλου OpenFlow που υποστηρίζεται από τον αποστολέα. Κατά την παραλαβή αυτού του μηνύματος, ο παραλήπτης μπορεί να υπολογίσει την έκδοση του OpenFlow πρωτοκόλλου που θα χρησιμοποιηθεί ως το μικρότερο αριθμό της έκδοσης που έχει σταλθεί και εκείνης που έχει ληφθεί.

Εάν η διαπραγματευόμενη έκδοση υποστηρίζεται από τον παραλήπτη, τότε η σύνδεση συνεχίζεται. Διαφορετικά, ο παραλήπτης πρέπει να απαντήσει με ένα μήνυμα OFPT\_ERROR με τύπο πεδίου OFPET\_HELLO\_FAILED, πεδίο κώδικα OFPHFC\_COMPATIBLE και προαιρετικά μια συμβολοσειρά ASCII που εξηγεί την κατάσταση στα δεδομένα και έπειτα η σύνδεση να τερματιστεί.

Σε περίπτωση που ένας διακόπτης χάσει την επαφή με τον ελεγκτή, ως αποτέλεσμα λήξης ενός echo αιτήματος, λήξης της TLS συνεδρίας ή κάποιας άλλης αποσύνδεσης, θα πρέπει να επιχειρήσει να επικοινωνήσει με έναν ή περισσότερους backup ελεγκτές. Η παραγγελία των IP διευθύνσεων του ελεγκτή δεν καθορίζεται από το πρωτόκολλο. Εάν αποτύχει κάποιος αριθμός προσπαθειών επικοινωνίας με έναν ελεγκτή (μηδέν ή περισσότερες), ο διακόπτης πρέπει να τεθεί σε κατάσταση "έκτακτης ανάγκης" και να επαναρυθμίσει αμέσως την τρέχουσα TCP σύνδεση. Σε κατάσταση έκτακτης ανάγκης, η διαδικασία αντιστοίχισης υπαγορεύεται από τις καταχωρήσεις του πίνακα ροής έκτακτης ανάγκης (εκείνες που σημειώνονται με το bit έκτακτης ανάγκης όταν προστίθενται στο διακόπτη). Όλες οι κανονικές καταχωρήσεις διαγράφονται όταν ο διακόπτης εισέρχεται σε λειτουργία έκτακτης ανάγκης. Με την επανασύνδεση του ελεγκτή, οι καταχωρήσεις ροής έκτακτης ανάγκης παραμένουν, ο ελεγκτής στη συνέχεια έχει την επιλογή να διαγράψει όλες τις καταχωρήσεις ροής, εάν είναι επιθυμητό. Την πρώτη φορά που ξεκινά ένας διακόπτης, θεωρείται ότι

βρίσκεται σε κατάσταση έκτακτης ανάγκης. Η διαμόρφωση του προεπιλεγμένου συνόλου καταχωρήσεων ροής βρίσκεται εκτός του πλαισίου του πρωτοκόλλου OpenFlow.

Όταν ένας διακόπτης συνδέεται σε ένα δίκτυο OpenFlow, δημιουργεί μια χειραψία TCP (handshake) με τη διεύθυνση IP του ελεγκτή (διασύνδεση Loopback 127.0.0.1) και μια προεπιλεγμένη θύρα 6633. Μετά από αυτή τη διαδικασία, και οι δύο πλευρές αρχίζουν να ανταλλάσσουν μηνύματα Hello και αποφασίζεται η έκδοση. Μετά από αυτό στέλνεται το μήνυμα αίτησης χαρακτηριστικών από τον ελεγκτή για να δουν ποιες θύρες είναι διαθέσιμες στον διακόπτη, ο οποίος με τη σειρά του απαντάει με το μήνυμα χαρακτηριστικών που περιέχει μια λίστα με τις θύρες, την ταχύτητα των θυρών και τους υποστηριζόμενους πίνακες και ενέργειες. Το μήνυμα set config αποστέλλεται στη συνέχεια από τον ελεγκτή για να το ζητήσει από τον διακόπτη να στείλει τον τερματισμό ροής. Τέλος, αιτήματα echo και απαντήσεις echo αποστέλλονται συχνά μεταξύ του διακόπτη και του ελεγκτή για την ανταλλαγή πληροφοριών σχετικά με το εύρος ζώνης, την καθυστέρηση και τη ζωή του της σύνδεσής τους (Σχήμα 3.4).

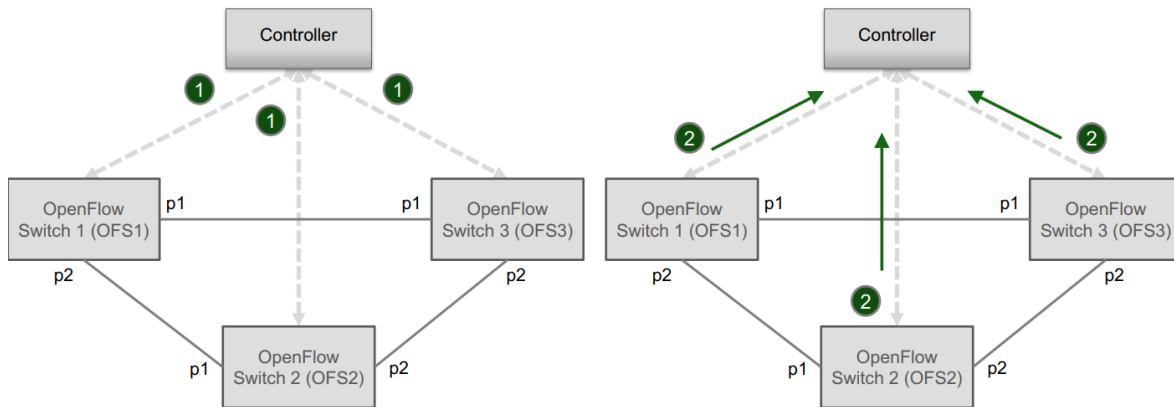


Σχήμα 3.5 Μηνύματα μεταξύ OpenFlow ελεγκτή και διακόπτη

### 3.5.1 ΔΙΑΔΙΚΑΣΙΑ ΕΥΡΕΣΗΣ ΤΟΠΟΛΟΓΙΑΣ

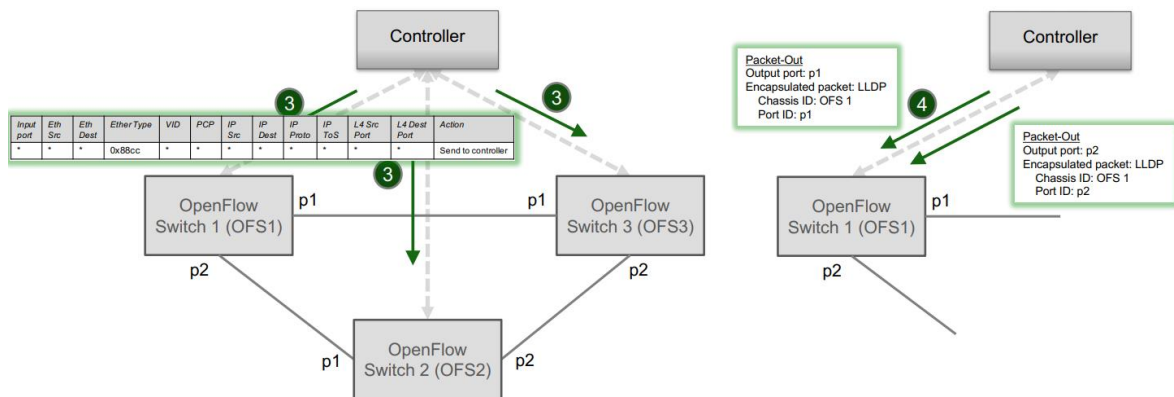
Παρακάτω θα δείξουμε σε 6 βήματα πως βρίσκεται η τοπολογία μεταξύ ενός OpenFlow ελεγκτή και των διακοπών του [23]:

- 1) Οι διακόπτες εγκαθιδρύουν ένα OpenFlow κανάλι με τον ελεγκτή
- 2) Ο ελεγκτής μαθαίνει τις ενεργές θύρες σε όλους του διακόπτες μέσω ενός μηνύματος Feature reply



Σχήμα 3.5.1: Δημιουργία καναλιού και αναγνώριση θυρών

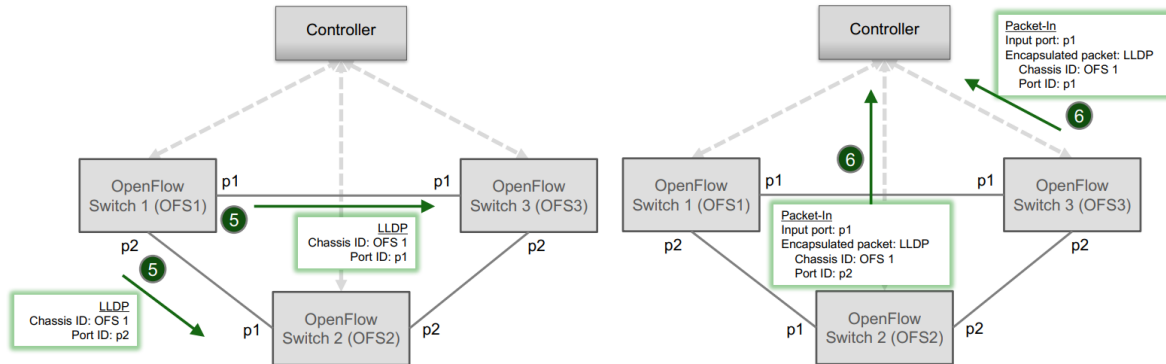
- 3) Δημιουργείται η εγγραφή ροής για την προώθηση όλων των LLDP πακέτων στον ελεγκτή
- 4) Ο ελεγκτής παράγει ένα Packet-out μήνυμα (που εμπεριέχει ένα LLDP πακέτο) για κάθε ενεργή θύρα, κάθε διακόπτη



Σχήμα 3.5.2: Δημιουργία εγγραφής ροής και Packet-out μηνύματος



- 5) Ο διακόπτης στέλνει το LLDP πακέτο από κάθε ενεργή του θύρα
- 6) Οι διακόπτες OFS2 και OFS3 προωθούν το λαμβανόμενο LLDP πακέτο στον ελεγκτή μέσω Packet-in μηνύματος.



Σχήμα 3.5.3: Αποστολή του LLDP πακέτου από τους διακόπτες και λήψη από τον ελεγκτή

Στο τέλος της παραπάνω διαδικασίας ο ελεγκτής μαθαίνει ότι:

- Η θύρα p1 του OFS1 συνδέεται απευθείας με τη θύρα p1 του OFS3
- Η θύρα p2 του OFS1 συνδέεται απευθείας με τη θύρα p1 του OFS2
- Η θύρα p2 του OFS2 είναι απευθείας συνδεδεμένη με τη θύρα p2 του OFS3
- Η θύρα p1 του OFS3 συνδέεται απευθείας με τη θύρα p1 του OFS1
- Η θύρα p1 του OFS2 είναι απευθείας συνδεδεμένη με τη θύρα p2 του OFS1
- Η θύρα p2 του OFS3 συνδέεται απευθείας με τη θύρα p2 του OFS2

## ΚΕΦΑΛΑΙΟ 4

### 4 ΕΡΓΑΛΕΙΑ – ΜΕΤΡΗΣΕΙΣ

Στο παρόν κεφάλαιο θα παρουσιάσουμε την μεθοδολογία, τα εργαλεία και τους ελεγκτές που χρησιμοποιήσαμε. Θα καταγράψουμε τις μετρήσεις μας και τέλος τα συμπεράσματα μας που αποπνέουν από αυτές.

#### 4.1 ΕΡΓΑΛΕΙΑ ΑΞΙΟΛΟΓΗΣΗΣ ΕΛΕΓΚΤΩΝ

CBench: Το CBench [24] είναι ένα από τα θεμελιώδη εργαλεία συγκριτικής αξιολόγησης με άδεια ανοιχτού κώδικα. Έχει σχεδιαστεί ειδικά για την αξιολόγηση της απόδοσης ελεγκτών OpenFlow SDN που υποστηρίζουν OpenFlow 1.0 και 1.3. Ωστόσο, λόγω περιορισμού συμβατότητας, οι ελεγκτές με το OpenFlow 1.3 μπορεί έχουν θέματα επιδόσεων. Υπάρχουν δύο βασικές μετρήσεις αξιολόγησης στο CBench, καθυστέρησης και διέλευσης. Για να μετρήσουμε την καθυστέρηση, ο vSwitch προωθεί ένα μήνυμα πακέτου προς τον ελεγκτή και περιμένει μια απάντηση. Οι δοκιμές μπορούν να επαναληφθούν πολλές φορές για να επιτευχθεί η μέση απόδοση. Ο συνολικός αριθμός των αναγνωρίσεων που λαμβάνονται σε μια δοκιμαστική περίοδο χρησιμοποιείται για να υπολογίσουμε τη μέση λανθάνουσα κατάσταση. Όσον αφορά τη μέτρηση της απόδοσης, κάθε vSwitch αποστέλλει συνεχώς τόσα πακέτα σε μηνύματα όσο είναι δυνατόν, για την εκτίμηση της ικανότητας του ελεγκτή.

HCprobe: Το HCprobe [25] είναι μια επέκταση ανοιχτού κώδικα του CBench, που αναπτύχθηκε με το συνδυασμό Python και Shell scripts για την παροχή πρόσθετων δυνατοτήτων αξιολόγησης της απόδοσης, όπως η αξιοπιστία και η επεκτασιμότητα. Ο εξομοιωμένος διακόπτης μπορεί να στείλει ευάλωτα μηνύματα OpenFlow στους ελεγκτές για έλεγχο ευελιξίας και αξιοπιστίας. Εκτός αυτού, η μηχανή δοκιμών χρησιμοποιεί Linux πυρήνα, ο οποίος επιτρέπει προσαρμόσιμο και επεκτάσιμο συντονισμό των threads της CPU. Αυτό επιτρέπει στον ελεγκτή να αποκτήσει περισσότερα ακριβή στατιστικά στοιχεία απόδοσης ενός ελεγκτή SDN.

WCBench: Το WCBench [26] είναι άλλη μια άλλη παραλλαγή του CBench γραμμένο σε Python και χρησιμοποιεί την βασική βιβλιοθήκη του CBench. Σε σύγκριση με τον CBench, το σύνολο χαρακτηριστικών αυτού του εργαλείου υπερβαίνει την καθυστέρηση και τη διακίνηση, και προσφέρει επιπλέον πτυχές αυτοματοποιημένης αξιολόγησης με λεπτομερείς και γραφικές στατιστικές. Αν και επεκτείνει την υποστήριξη του OpenFlow στην έκδοση 1.3, η συμβατότητα του WCBench εξακολουθεί να είναι περιορισμένη σε συγκεκριμένες εκδόσεις.

OFCBenchmark: Το OFCBenchmark [27] είναι κατασκευασμένο με τη χρήση της βιβλιοθήκης C ++ και Boost για να αντιμετωπίσει ορισμένους από τους περιορισμούς της CBench. Τα στοιχεία αυτού του εργαλείου συγκριτικής αξιολόγησης περιλαμβάνουν ένα γραφικό ταμπλό (φτιαγμένο με Delphi), εικονικοποιημένο scalable vSwitch που είναι το βασικό module και περιλαμβάνει έναν client η που μπορεί να διαχειριστεί δοκιμές αξιολόγησης. Το εργαλείο προσφέρει κατανεμημένη συγκριτική αξιολόγηση με το να επιτρέπει clients να τρέχουν σε πολλαπλές περιπτώσεις, και προσφέρει εκτεταμένη συγκριτική αξιολόγηση, όπως Round Trip Time(RTT), ρυθμός εγκατάστασης ροής, utilization της CPU κ.λπ.

OFCProbe: Το OFCProbe [28] είναι μια αναβαθμισμένη έκδοση του OFCBenchmark το οποίο επικεντρώνεται στη μεγιστοποίηση της ευελιξίας των SDN ελεγκτών εξομοιώνοντας ένα σηματικό ποσό των OpenFlow διακοπών σε περιβάλλον μεγάλης κλίμακας. Είναι επανασχεδιασμένο χρησιμοποιώντας Java ώστε να γίνει ένα εργαλείο ανεξαρτήτου πλατφόρμας και επίσης να ξεπεράσει το overhead της εικονικοποίησης που προκαλείται από την εξομοίωση SDN σε εργαλεία όπως το Mininet [29]. Ο βασικός σκοπός αυτού του εργαλείου είναι να αναλύσει τις επιπτώσεις της τοπολογίας του δικτύου κατά τη διάρκεια της αξιολόγησης που εκτελείται στο κομμάτι του client.

PktBlaster: Το PktBlaster [30] είναι μια ενοποιημένη λύση δοκιμών που εξομοιώνει μεγάλα δίκτυα SDN, συμπεριλαμβανομένης της υποδομής δικτύου και των στρωμάτων των ελεγκτών SDN. Η δωρεάν έκδοση περιορισμένων δυνατοτήτων προσφέρει χαρακτηριστικά όπως μέτρηση απόδοσης και καθυστέρησης με διαφορετικά προφίλ δοκιμών, δηλ. TCP, UDP, ARP request και ARP reply. Μία δοκιμή απόδοσης καθορίζει το ρυθμό με τον οποίο ο ελεγκτής ρυθμίζει τη ροή στους διακόπτες. Η δοκιμή καθυστέρησης δίνει τον ακριβή χρόνο

(σε χιλιοστά του δευτερολέπτου) που ο χρειάζεται ο ελεγκτής για να επεξεργαστεί μια ροή σε ένα διακόπτη. Αν και η δωρεάν έκδοση περιορίζεται σε 16 διακόπτες και 64 διευθύνσεις MAC, προσφέρει επιπλέον επιλογές όπως πίνακες ροής, ομαδικούς πίνακες, πίνακες μετρητών, το μέγεθος του buffer και μέγιστες καταχωρήσεις ανά πίνακα ροής.

OFNet: Το OFNet [31] είναι μια συνδυαστική προσέγγιση για την ενσωμάτωση εξομοίωσης δικτύου OpenFlow με παρακολούθηση επιδόσεων και οπτικό debugging ελεγκτών SDN. Το OFNet μπορεί να αναπτυχθεί σε ένα σύστημα για τη δημιουργία διαφορετικών τύπων τοπολογιών. Η ενσωματωμένη γεννήτρια κυκλοφορίας παράγει διαφορετικούς τύπους κίνησης δικτύου. Είναι ικανό να μετρήσει τα χαρακτηριστικά απόδοσης του ελεγκτή, όπως δημιουργίες ροής, βλάβες ροής, χρήση της CPU, καταχωρήσεις στον πίνακα ροής, μέση τιμή RTT, καθυστέρηση ρύθμισης ροής κ.λπ.

Στην παρούσα εργασία το εργαλείο για την αξιολόγηση των ελεγκτών που χρησιμοποιήσαμε είναι το CBench λόγω της συμβατότητας του με τους ελεγκτές και της προσβασιμότητας του.

#### 4.2 ΕΡΓΑΛΕΙΑ ΠΡΟΣΟΜΕΙΩΣΗΣ

Virtual box: Το Oracle VM VirtualBox [32] (πρώην Sun VirtualBox, Sun xVM VirtualBox και Innotek VirtualBox) είναι ένας ελεύθερος και ανοιχτού κώδικα, hypervisor για υπολογιστές x86 που αναπτύσσονται επί του παρόντος από την Oracle Corporation. Αναπτύχθηκε αρχικά από την Innotek GmbH, αποκτήθηκε από την Sun Microsystems το 2008, η οποία με τη σειρά της αποκτήθηκε από την Oracle το 2010. Το VirtualBox μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα φιλοξενίας, όπως: Linux, macOS, Windows, Solaris και OpenSolaris. Υπάρχουν επίσης θύρες στο FreeBSD και το Genode. Υποστηρίζει τη δημιουργία και τη διαχείριση εικονικών μηχανών guest που εκτελούν εκδόσεις και παραλλαγές των Windows, Linux, BSD, OS / 2, Solaris, Haiku, OSx86 και άλλων και περιορισμένη εικονικοποίηση των χρηστών MacOS στο υλικό της Apple.

Mininet: Το Mininet είναι ένας εξομοιωτής δικτύου ή ίσως, ένα σύστημα ενορχήστρωσης δικτύου. Διαχειρίζεται μια συλλογή τελικών κεντρικών υπολογιστών, διακοπών, δρομολογητών και συνδέσμων σε έναν ενιαίο πυρήνα Linux. Χρησιμοποιεί ελαφριά εικονικοποίηση για να κάνει ένα ενιαίο σύστημα να μοιάζει με ένα πλήρες δίκτυο, τρέχοντας

τον ίδιο πυρήνα, σύστημα και κωδικό χρήστη. Ένας Mininet host συμπεριφέρεται σαν μια πραγματική μηχανή, μπορούμε να το χρησιμοποιήσουμε και να εκτελέσουμε προγράμματα (συμπεριλαμβανομένων οποιωνδήποτε εγκαταστάσεων στο υποκείμενο σύστημα Linux). Τα προγράμματα που εκτελούμε μπορούν να στείλουν πακέτα μέσω μιας πραγματικής Ethernet διεπαφής, με δεδομένη ταχύτητα σύνδεσης και καθυστέρηση. Τα πακέτα επεξεργάζονται με αυτό που μοιάζει με έναν πραγματικό μεταγωγέα Ethernet, δρομολογητή ή middlebox, με ένα δεδομένο ποσό αναμονής. Όταν δύο προγράμματα, όπως ένας iperf server και client επικοινωνούν μέσω του Mininet, η μετρηθείσα απόδοση πρέπει να μοιάζει με αυτή των κανονικών μηχανών. Οι εικονικοί κεντρικοί υπολογιστές του Mininet, οι διακόπτες, οι σύνδεσμοι και οι ελεγκτές – που δημιουργούνται απλώς χρησιμοποιώντας λογισμικό και όχι hardware – έχουν παρόμοια συμπεριφορά με τα φυσικά hardware κομμάτια. Συνήθως είναι δυνατόν να δημιουργηθεί ένα δίκτυο Mininet που μοιάζει με hardware δίκτυο ή ένα hardware δίκτυο που μοιάζει με ένα δίκτυο Mininet και να εκτελούν τον ίδιο δυαδικό κώδικα και εφαρμογές σε οποιαδήποτε πλατφόρμα.

Xming: Το Xming [33] είναι ένας διακομιστής προβολής X11 για λειτουργικά συστήματα Microsoft Windows, συμπεριλαμβανομένων των Windows XP και νεότερων. Ο διακομιστής Xming X βασίζεται στο Cygwin / X. Είναι cross compiled σε Linux με το μεταγλωττιστή MinGW και τη multithreaded βιβλιοθήκη Pthreads-Win32. Το Xming τρέχει εγγενώς στα Windows και δεν χρειάζεται third party λογισμικό εξομοίωσης. Το Xming μπορεί να χρησιμοποιηθεί με εφαρμογές του Secure Shell (SSH) για την ασφαλή προώθηση X11 συνεδριών από άλλους υπολογιστές. Υποστηρίζει PuTTY και ssh.exe και συνοδεύεται από μια έκδοση του plink.exe του PuTTY. Το έργο Xming προσφέρει επίσης μια φορητή έκδοση του PuTTY. Όταν δεν χρησιμοποιείται η προώθηση SSH, το τοπικό αρχείο Xn.hosts πρέπει να ενημερωθεί με το όνομα του κεντρικού υπολογιστή ή τη διεύθυνση IP του απομακρυσμένου μηχανήματος όπου ξεκινά η εφαρμογή GUI. Το λογισμικό έχει προταθεί από συγγραφείς βιβλίων για το ελεύθερο λογισμικό, όταν απαιτείται ένας δωρεάν διακομιστής X, και περιγράφεται ως απλός και πιο εύκολο να εγκατασταθεί, αν και είναι λιγότερο προσαρμόσιμο από άλλες δημοφιλείς ελεύθερες επιλογές όπως το Cygwin / X.

Putty: Το PuTTY [34] είναι ένας ελεύθερος και ανοικτού κώδικα τερματικός εξομοιωτής, σειριακή κονσόλα και εφαρμογή μεταφοράς αρχείων δικτύου. Υποστηρίζει αρκετά πρωτόκολλα δικτύου, συμπεριλαμβανομένων των συνδέσεων SCP, SSH, Telnet, rlogin και

raw socket. Μπορεί επίσης να συνδεθεί σε μια σειριακή θύρα. Το όνομα "PuTTY" δεν έχει επίσημη σημασία. Το PuTTY γράφτηκε αρχικά για τα Microsoft Windows, αλλά έχει μεταφερθεί σε διάφορα άλλα λειτουργικά συστήματα. Επίσημες θύρες είναι διαθέσιμες για ορισμένες πλατφόρμες τύπου Unix και ανεπίσημες θύρες έχουν συμβάλει σε πλατφόρμες όπως το Symbian, το Windows Mobile και το Windows Phone. Το PuTTY γράφτηκε και διατηρείται κυρίως από τον Simon Tatham.

Cbench: Το Cbench [35] μιμείται μια δέσμη διακοπών που συνδέονται με έναν ελεγκτή, στέλνουν μηνύματα packet-in και παρακολουθούν ροές [24]. Το cbench είναι ένα benchmarking εργαλείο για ελεγκτές, ο αλγόριθμος που χρησιμοποιεί είναι ο παρακάτω:

Algorithm:

```
pretend to be n switches (n=16 is default)
create n openflow sessions to the controller
if latency mode (default):
  for each session:
    1) send up a packet in
    2) wait for a matching flow mod to come back
    3) repeat
    4) count how many times #1-3 happen per sec
else in throughput mode (i.e., with '-t'):
  for each session:
    while buffer not full:
      queue packet_in's
      count flow_mod's as they come back
```

### 4.3 ΜΕΘΟΔΟΛΟΓΙΑ

Ο υπολογιστής που χρησιμοποιήσαμε για το πείραμα είναι ένας Intel Core i7-4765T 4άρων πυρήνων και 8 threads, με 16 GB μνήμης. Ο υπολογιστής έχει εγκατεστημένα Windows. Η προσομοίωση μας χρειάζονται LINUX περιβάλλον, χρησιμοποιήσαμε το Putty και το Xming (που αναφέραμε πιο πάνω) για να μπορέσουν να δύο λειτουργικά να επικοινωνήσουν. Χρησιμοποιήθηκε η έκδοση 2.2.2 Mininet σε Ubuntu 14.04 LTS που έτρεξε σε εικονικό περιβάλλον VirtualBox.

Για τις δοκιμές παραμετροποιήθηκε ο κάθε ελεγκτής ξεχωριστά σύμφωνα με τις οδηγίες (αφού εγκαταστήσαμε κάθε dependency που ήταν απαραίτητο) και ύστερα τρέξαμε το cbench [24] σε 2 διαφορετικά modes. Σε throughput mode και σε latency mode. Σε κάθε ελεγκτή έγιναν δοκιμές με 8, 16, 32, 64 και 128 διακόπτες. Κάθε δοκιμή επαναλήφθηκε 5 φορές, με 14 κύκλους διάρκειας 10 δευτερολέπτων ο καθένας. Η εντολή για την διεξαγωγή των δοκιμών είναι η : /cbench -c localhost -p 6633 -l 14 -m 10000 -M 1000 -s 8 -t.

## 4.4 ΜΕΤΡΗΣΕΙΣ

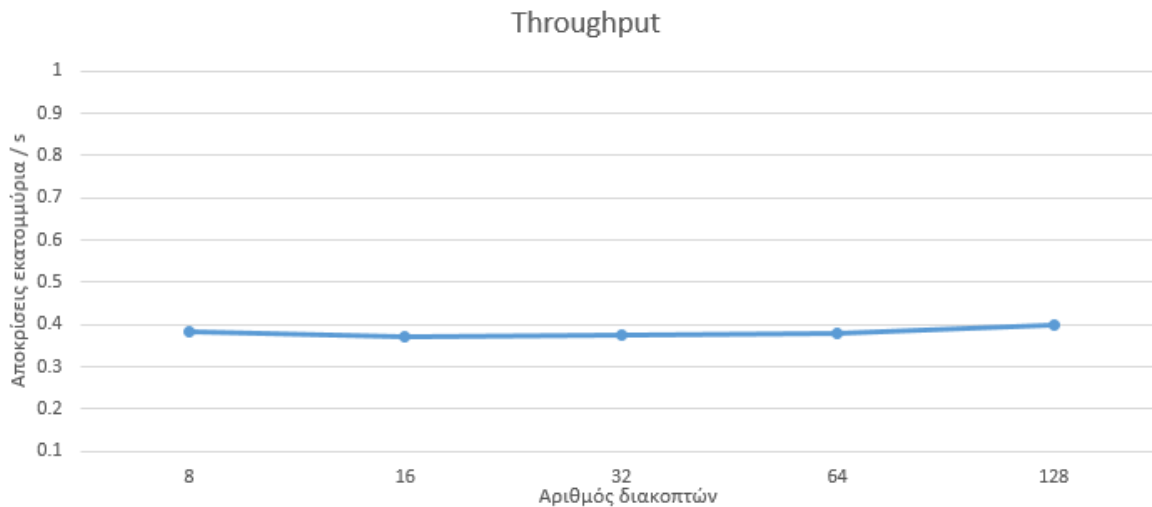
### 4.4.1 POX

Ο POX είναι μια πλατφόρμα ανάπτυξης ανοιχτού κώδικα (βασισμένη στην Python) για SDN εφαρμογές ελέγχου όπως οι ελεγκτές OpenFlow SDN [36]. Ο POX, επιτρέπει την ταχεία ανάπτυξη και τη δημιουργία πρωτοτύπων και η χρήση του γίνεται όλο και πιο συνηθισμένη σε σχέση με τον NOX. Οι λειτουργίες του POX είναι:

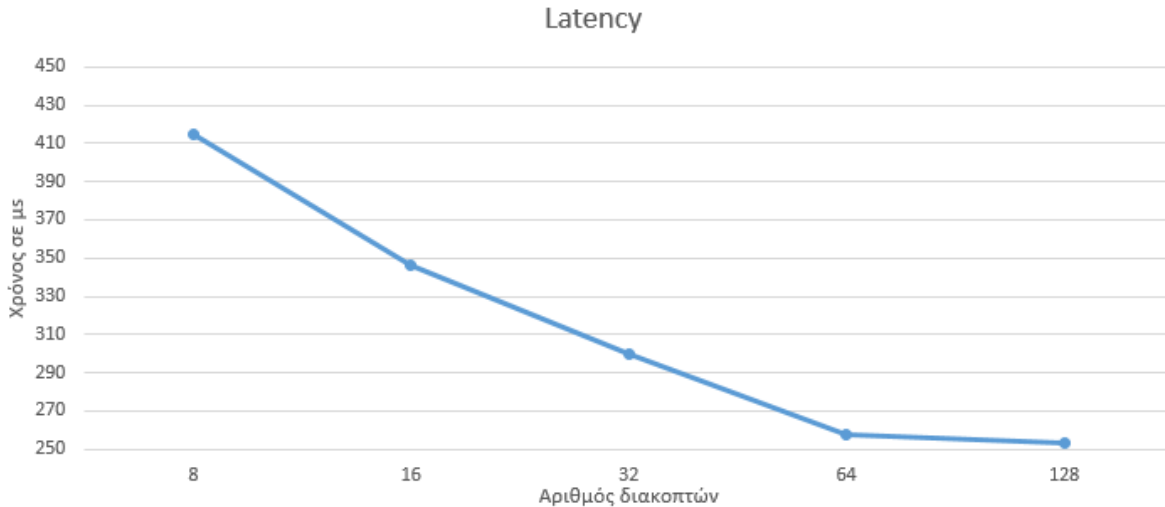
- OpenFlow διεπαφή βασισμένη στην Python.
- Επαναχρησιμοποιήσιμα στοιχεία δείγματος για επιλογή διαδρομής, ανακάλυψη τοπολογίας κλπ.
- Ειδικότερα στοχεύει στο Linux, στο Mac OS και στα Windows.
- Ανακάλυψη τοπολογίας.
- Υποστηρίζει τα ίδια GUI και εργαλεία οπτικοποίησης όπως ο NOX.
- Λειτουργεί καλά σε σύγκριση με τις εφαρμογές του NOX που είναι γραμμένες σε Python.

Ο δικτυακός τόπος του project λέει ότι ο απώτερος στόχος του POX είναι η χρησιμοποίηση του για τη δημιουργία ενός "αρχετυπικού, σύγχρονου ελεγκτή SDN".

Παρακάτω οι μετρήσεις που πήραμε χρησιμοποιώντας το Cbench:



Σχήμα 4.1.1 POX throughput



Σχήμα 4.1.2 POX latency

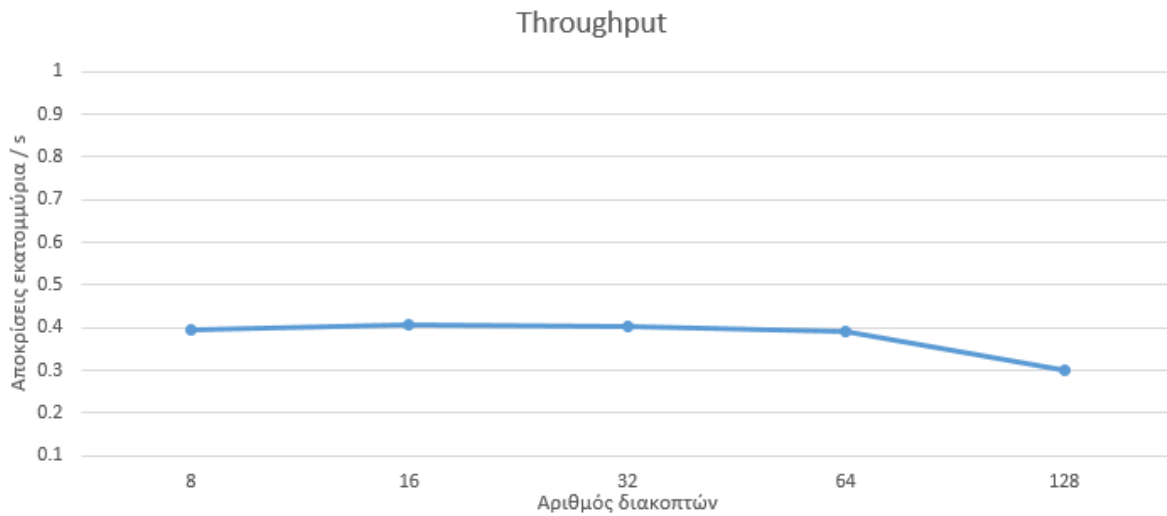
#### 4.4.2 NOX

Ο NOX είναι μια πλατφόρμα ανάπτυξης ανοιχτού κώδικα (βασισμένη στην C++) για SDN εφαρμογές [37]. Αρχικά αναπτύχθηκε στα Nicira Networks μαζί με το OpenFlow. Ο NOX ήταν ο πρώτος ελεγκτής OpenFlow και δωρίστηκε στην ερευνητική κοινότητα το 2008. Οι παλαιότερες εκδόσεις του NOX, γνωστές ως NOX Classic, υποστηρίζουν Python και C++ αλλά δεν υποστηρίζονται πλέον ενεργά. Οι λειτουργίες του NOX είναι:

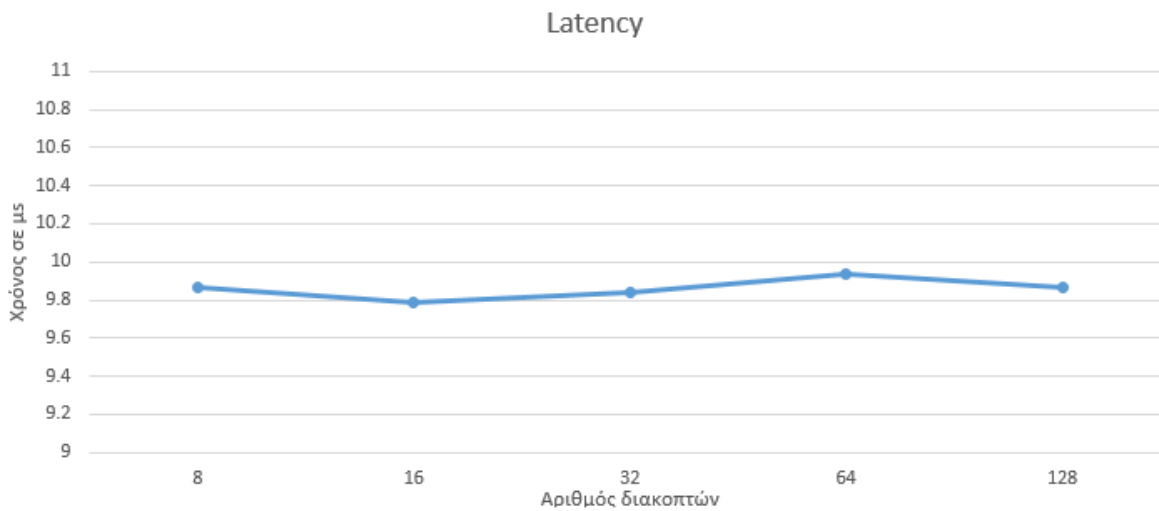
- Παρέχει C++ OpenFlow 1.0 API.
- Παρέχει γρήγορο, ασύγχρονο IO.
- Περιλαμβάνει δείγματα για: ανακάλυψη τοπολογίας, διακόπτη εκμάθησης, διακόπτη σε επίπεδο δικτύου.



Παρακάτω οι μετρήσεις που πήραμε χρησιμοποιώντας το Cbench:



Σχήμα 4.2.1 NOX throughput



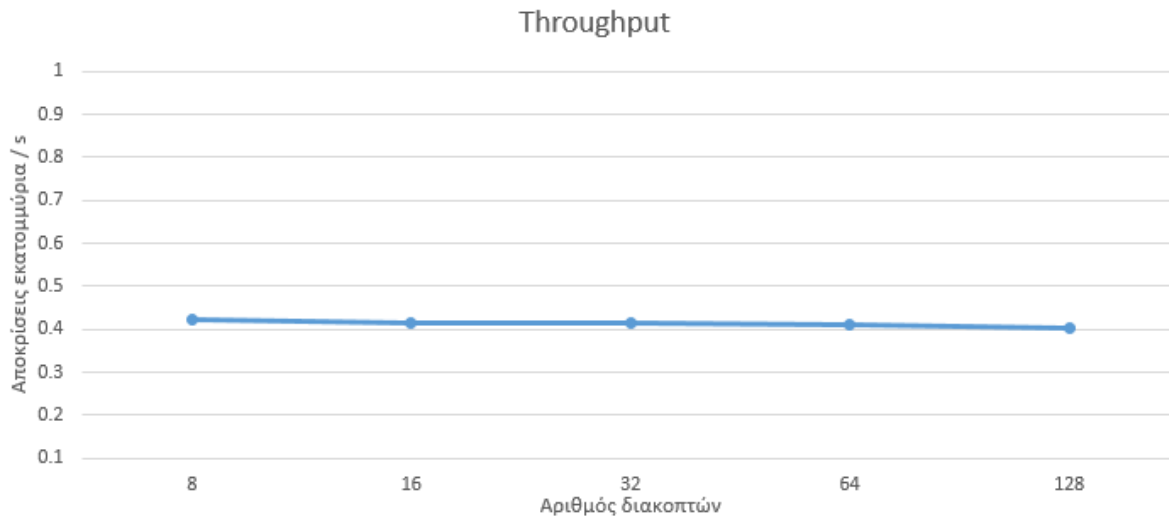
Σχήμα 4.2.2 NOX latency

#### 4.4.3 RYU

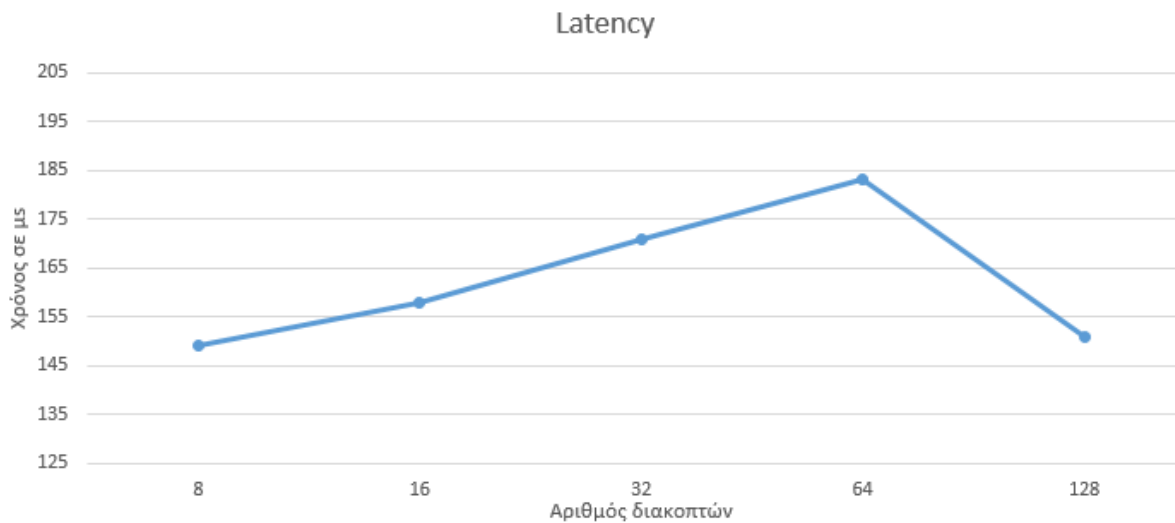
Ο Ryu [38] είναι ένα SDN framework. Ο Ryu παρέχει κομμάτια λογισμικού με καθορισμένο API που διευκολύνουν τους προγραμματιστές να δημιουργήσουν νέες εφαρμογές διαχείρισης και ελέγχου δικτύου. Ο Ryu υποστηρίζει διάφορα πρωτόκολλα για τη διαχείριση συσκευών δικτύου, όπως το OpenFlow, το Netconf, το OF-config κλπ. Σχετικά με το

OpenFlow, ο Ryu υποστηρίζει πλήρως τις επεκτάσεις 1.0, 1.2, 1.3, 1.4, 1.5 και Nicira. Όλος ο κώδικας είναι ελεύθερα διαθέσιμος υπό την άδεια Apache 2.0. Ryu σημαίνει "ροή" στα ιαπωνικά. Ο Ryu προφέρεται "ree-yooh".

Παρακάτω οι μετρήσεις που πήραμε χρησιμοποιώντας το Cbench:



Σχήμα 4.3.1 RYU throughput



Σχήμα 4.3.2 RYU latency

#### 4.4.4 MAESTRO

Το Maestro είναι ένα "λειτουργικό σύστημα" για την ενορχήστρωση εφαρμογών ελέγχου δικτύου. Το Maestro παρέχει διασυνδέσεις για την εφαρμογή αρθρωτών εφαρμογών ελέγχου δικτύου για την πρόσβαση και την τροποποίηση της κατάστασης του δικτύου και για τον συντονισμό των αλληλεπιδράσεών τους. [39] Το Maestro είναι μια πλατφόρμα για την επίτευξη αυτόματων και προγραμματικών λειτουργιών ελέγχου δικτύου χρησιμοποιώντας αυτές τις διαμορφωμένες εφαρμογές. Αν και το έργο αυτό επικεντρώνεται στην οικοδόμηση ενός ελεγκτή OpenFlow χρησιμοποιώντας το Maestro, το Maestro δεν περιορίζεται μόνο στα δίκτυα OpenFlow. Το πλαίσιο προγραμματισμού του Maestro παρέχει διεπαφές για:

- Την εισαγωγή νέων προσαρμοσμένων λειτουργιών ελέγχου με την προσθήκη αρθρωμένων εξαρτημάτων ελέγχου.
- Την διατήρηση της κατάστασης δικτύου για λογαριασμό των στοιχείων ελέγχου.
- Την σύνθεση στοιχείων ελέγχου με το να καθορίζει την αλληλουχία εκτέλεσης και την κατάσταση κοινής χρήσης των στοιχείων.

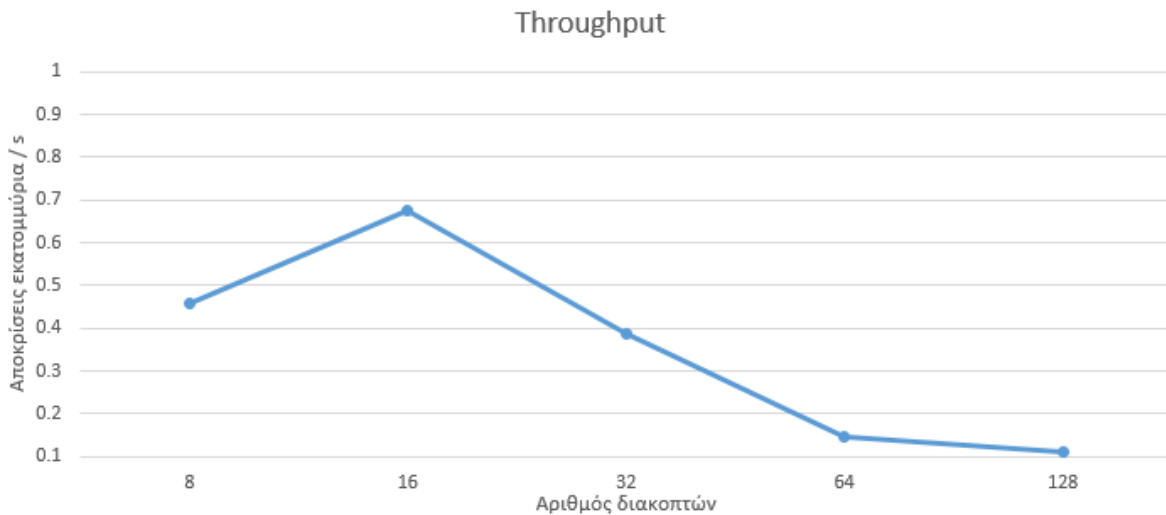
Επιπλέον, το Maestro προσπαθεί να εκμεταλλευτεί τον παραλληλισμό μέσα σε ένα μόνο μηχάνημα για να βελτιώσει την απόδοση του συστήματος. Το βασικό χαρακτηριστικό ενός δικτύου OpenFlow είναι ότι ο ελεγκτής είναι υπεύθυνος για την αρχική εγκατάσταση κάθε ροής επικοινωνώντας με τους αντίστοιχους διακόπτες. Έτσι, η απόδοση του ελεγκτή θα μπορούσε να μειώνει τη δυνατότητα του συστήματος. Κατά το σχεδιασμό του Maestro έγινε προσπάθεια να απαιτείται όσο το δυνατόν λιγότερη προσπάθεια από τους προγραμματιστές για την διαχείριση του παραλληλισμού. Το Maestro χειρίζεται το μεγαλύτερο μέρος της κουραστικής και πολύπλοκη δουλειά της διαχείρισης της διανομής φορτίου εργασίας. Από τη σχεδίαση του, το Maestro είναι φορητό και κλιμακωτό:

- Έχει αναπτυχθεί σε Java (τόσο στην πλατφόρμα όσο και στα εξαρτήματα) - Υψηλή φορητότητα σε διάφορα λειτουργικά συστήματα και αρχιτεκτονικές.
- Είναι multi-threaded - Αξιοποιεί πλήρως τους επεξεργαστές πολλαπλών πυρήνων.

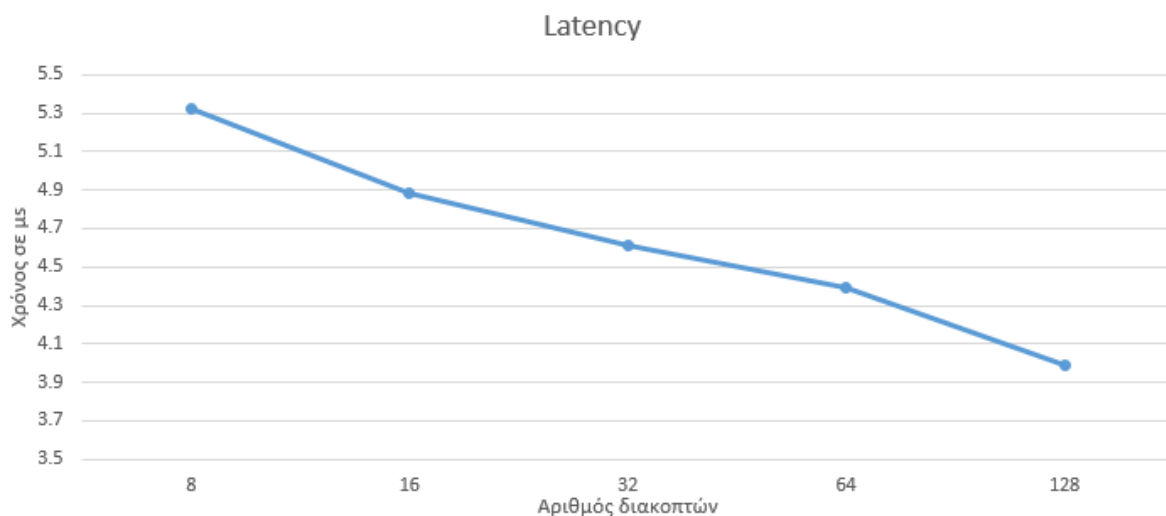
Το Maestro παρέχει τα στοιχεία ελέγχου για την πραγματοποίηση είτε ενός δικτύου διακοπών που μπορούν να "μαθαίνουν", είτε ενός δρομολογημένου δικτύου

χρησιμοποιώντας διακόπτες OpenFlow. Ορισμένα στοιχεία, όπως η κονσόλα γραμμής εντολών, κ.λπ., εξακολουθούν να μην είναι πλήρως ανεπτυγμένα. Σχεδιάζεται να εμπλουτιστεί και να βελτιωθεί περαιτέρω η λειτουργικότητα του Maestro σε μελλοντικές κυκλοφορίες. Το Maestro έχει άδεια χρήσης σύμφωνα με την έκδοση 2.1 του GNU Lesser General Public License.

Παρακάτω οι μετρήσεις που πήραμε χρησιμοποιώντας το Cbench:



Σχήμα 4.4.1 MAESTRO throughput



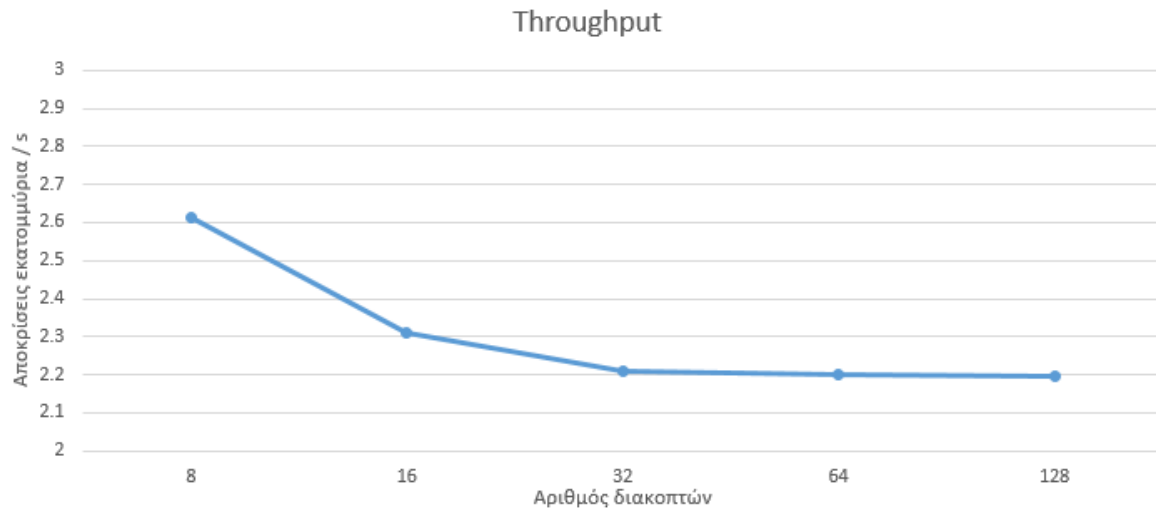
Σχήμα 4.4.2 MAESTRO latency

#### 4.4.5 BEACON

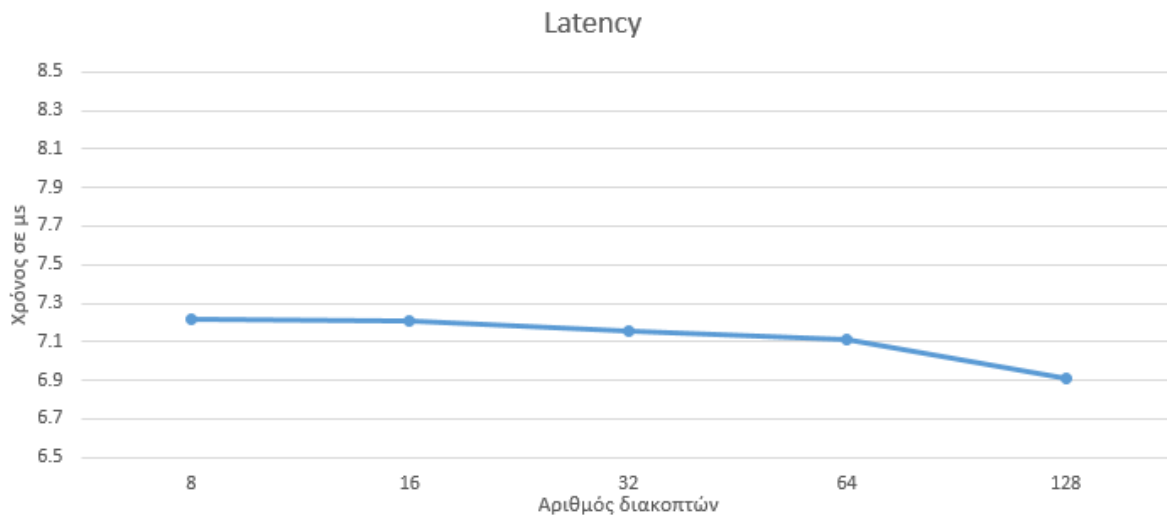
Ο beacon [40] είναι ένας open source OpenFlow ελεγκτής που βασίζεται στην Java, δημιουργήθηκε το 2010. Έχει χρησιμοποιηθεί ευρέως για τη διδασκαλία, την έρευνα, και ως τη βάση του Floodlight. Βασικά χαρακτηριστικά του:

- Σταθερός: Ο Beacon βρίσκεται σε εξέλιξη από τις αρχές του 2010 και έχει χρησιμοποιηθεί σε πολλά ερευνητικά έργα, σε μαθήματα δικτύωσης και δοκιμαστικές εφαρμογές. Ο Beacon εξουσιοδοτεί επί του παρόντος ένα πειραματικό κέντρο δεδομένων με φυσική εναλλαγή 100 πινάκων και έχει τρέξει για μήνες χωρίς διακοπή λειτουργίας.
- Cross-platform: Ο beacon είναι γραμμένος σε Java και τρέχει σε πολλές πλατφόρμες, από διακομιστές Linux πολλαπλών πυρήνων υψηλής τεχνολογίας μέχρι κινητά τηλέφωνα Android.
- Open source: Ο Beacon έχει άδεια χρήσης με συνδυασμό της άδειας GPL v2 και της εξαίρεσης άδειας χρήσης FOSS του Πανεπιστημίου Stanford v1.0.
- Δυναμικός - Οι δέσμες κωδικών στο beacon μπορούν να ξεκινήσουν / σταματήσουν / ανανεωθούν / εγκατασταθούν κατά τη διάρκεια της εκτέλεσης, χωρίς να διακόψουν άλλες μη εξαρτώμενες δέσμες (δηλαδή να αντικαταστήσουν την τρέχουσα εφαρμογή Learning Switch χωρίς να αποσυνδέσουν τους διακόπτες).
- Ταχέως αναπτυσσόμενος: Ο Beacon είναι εύκολο να εγκατασταθεί και να λειτουργήσει. Η Java και η Eclipse απλοποιούν την ανάπτυξη και το debugging των εφαρμογών.
- Γρήγορο: Ο Beacon είναι multi threaded.
- Web UI: Ο Beacon προαιρετικά ενσωματώνει τον εταιρικό web server Jetty και ένα προσαρμοσμένο πλαίσιο επεκτάσιμων UI.
- Frameworks: Ο Beacon βασίζεται σε ώριμα Java frameworks όπως το Spring and Equinox (OSGi).

Παρακάτω οι μετρήσεις που πήραμε χρησιμοποιώντας το Cbench:



Σχήμα 4.5.1 BEACON throughput



Σχήμα 4.5.2 BEACON latency

#### 4.4.6 OPENMUL

Ο OpenMUL Controller παρέχει μια πλατφόρμα ελεγκτή βάσεων για όλα τα SDN / Openflow. Πρόκειται για έναν ελαφρύ ελεγκτή SDN / Openflow γραμμένο σχεδόν εξ ολοκλήρου στη C και παρέχει μια πολύ σταθερή πλατφόρμα με κορυφαία απόδοση από άποψη της διαχείρισης της ροής (ρυθμός λήψης και latency) [41]. Έχει σχεδιαστεί για απόδοση, αξιοπιστία και άμεση διαθεσιμότητα. Είναι επίσης εξαιρετικά ευέλικτος,

αρθρωτός και εύκολος στην εκμάθηση. Ο Mul σημαίνει "βάση ή ρίζα" στο σανσκριτικό. Ο Mul είναι προφέρεται ως "mool" όπως στο "school".

Ο OpenMUL παρέχει πολυεπίπεδα και πολυγλωσσικά API για την υποστήριξη διαφορετικών αναγκών των εφαρμογών. Το ότι είναι γραμμένος στη C σημαίνει ότι μπορεί να χρησιμοποιηθεί για εφαρμογές ευαίσθητες στην απόδοση και το latency, ενώ οι εφαρμογές RESTful API μπορούν να χρησιμοποιηθούν από εφαρμογές ιστού. Μπορεί να ελέγχει συσκευές δικτύου που υποστηρίζουν Openflow, OVSDB καθώς και Netconf. Ακολουθούν τα κύρια στοιχεία του ελεγκτή OpenMUL:

#### 1. Ο πυρήνας του Mul / διαχειριστής του Mul

- Κύριο κομμάτι του Mul.
- Χειρίζεται όλες τις συνδέσεις χαμηλού επιπέδου του διακόπτη και κάνει την Openflow επεξεργασία.
- Παρέχει interface προγραμματισμού εφαρμογών με τη μορφή MLAPIS (API μεσαίου επιπέδου)
- Παρέχει πρόσβαση σε συσκευές Openflow (ή οποιοδήποτε south bound πρωτόκολλο).
- Υποστηρίζει εργαλεία για υψηλής ταχύτητας με χαμηλή καθυστέρηση υποδομή.
- Διασφαλίζει ότι όλες οι ροές, οι ομάδες, οι μετρητές και άλλες οντότητες που αφορούν συγκεκριμένους διακόπτες διατηρούνται σε συγχρονισμό σε όλους τους διακόπτες, τις επανεκκινήσεις / αποτυχίες του ελεγκτή.

#### 2. Υπηρεσίες υποδομής

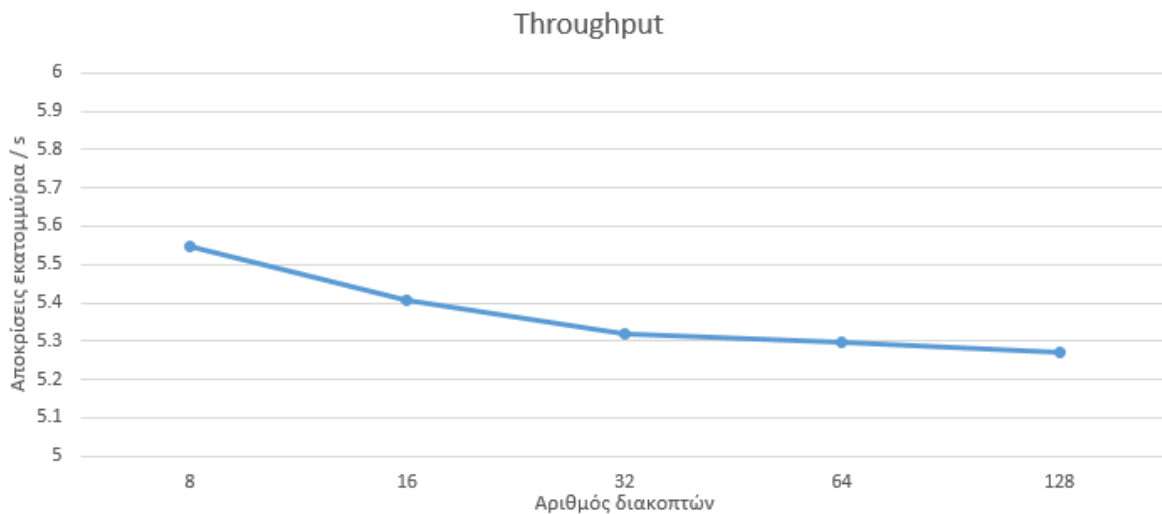
- Παρέχουν βασικές υπηρεσίες υπέρυθρων που είναι χτισμένες πάνω στον πυρήνα του mul.
- Επί του παρόντος διαθέσιμες: υπηρεσία εντοπισμού τοπολογίας, υπηρεσία εύρεσης διαδρομής, υπηρεσία σύνδεσης διαδρομής.

#### 3. Εφαρμογές συστήματος MuL

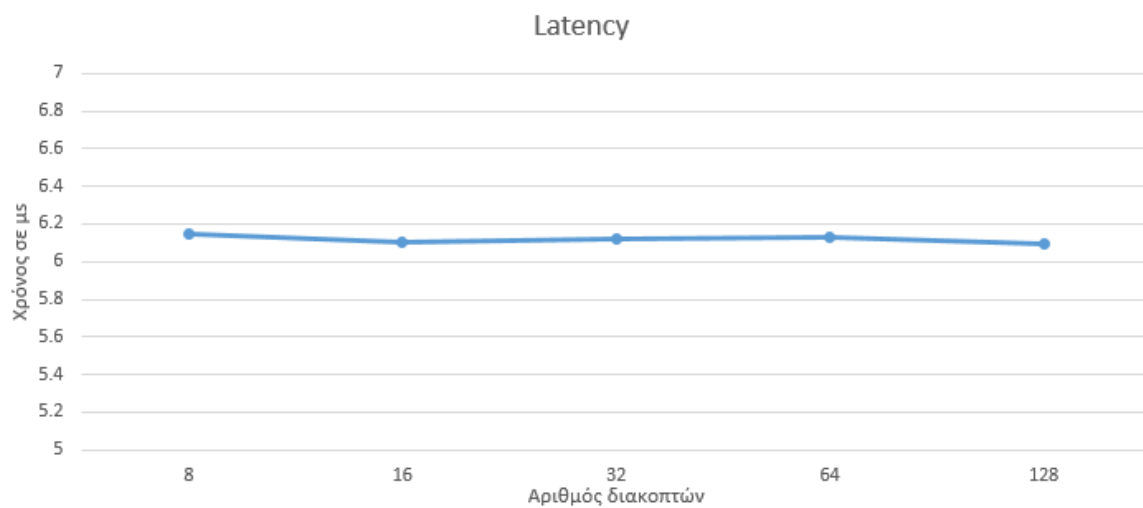
- Οι εφαρμογές του συστήματος κατασκευάζονται χρησιμοποιώντας ένα κοινό api που παρέχεται από το διαχειριστή του mul και τις υπηρεσίες του mul.

- Αυτές δεν γνωρίζουν σχεδόν καθόλου το Openflow και συνεπώς έχουν σχεδιαστεί για να λειτουργούν σε διαφορετικές εκδόσεις openflow με την προϋπόθεση ότι οι διακόπτες υποστηρίζουν την κοινές απαιτήσεις αυτών των εφαρμογών.
- Διατίθεται προς το παρόν: L2switch, CLI app, webserver NBAPI.

Παρακάτω οι μετρήσεις που πήραμε χρησιμοποιώντας το Cbench:



Σχήμα 4.6.1 OPENMUL throughput



Σχήμα 4.6.2 OPENMUL latency

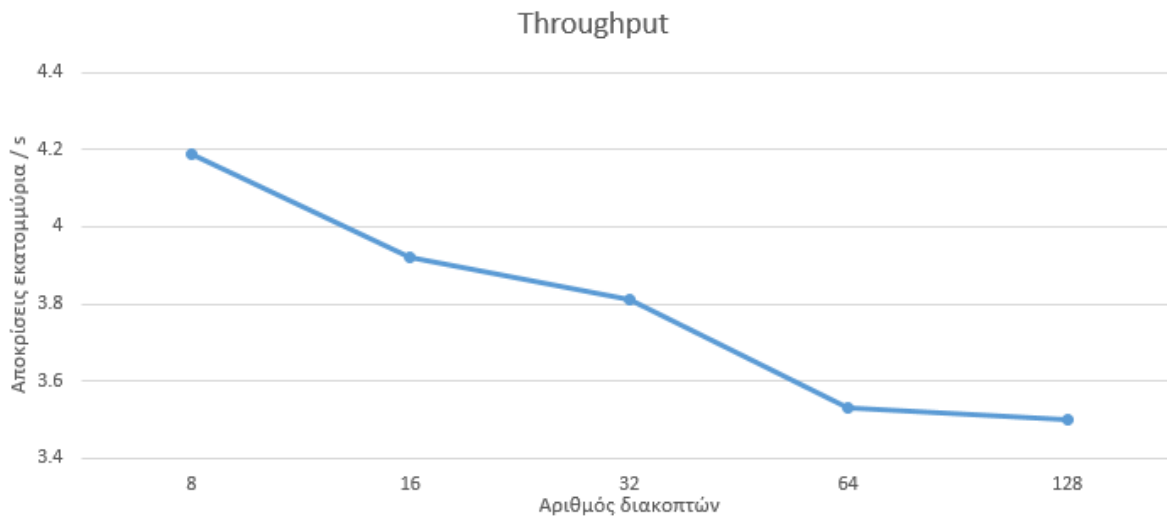


#### 4.4.7 LIBFLUID BASE

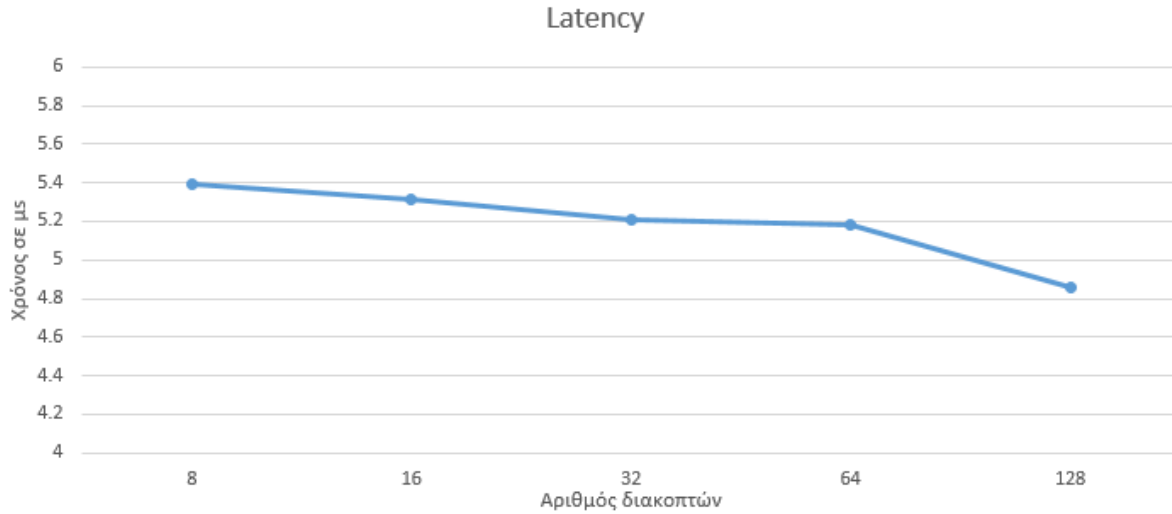
Είναι γραμμένος στην C++ και μπορεί να χρησιμοποιηθεί με οποιαδήποτε άλλη βιβλιοθήκη μηνυμάτων OpenFlow. Έχει άδεια χρήσης με την άδεια Apache, έκδοση 2.0. Το libfluid επιλέχθηκε ως ο νικητής του διαγωνισμού OpenFlow Driver με τη χορηγία του ONF [42].

Ορίζει μια αρχιτεκτονική πελάτη-διακομιστή, στην οποία ο ελεγκτής είναι ένας διακομιστής και ο διακόπτης είναι πελάτης. Παρέχει μια βασική κλάση με την οποία μπορούμε να δημιουργήσουμε τον ελεγκτή μας: `fluid_base :: OFServer`. Πρέπει να εφαρμοστούν οι επανάκλησεις για συμβάντα σύνδεσης και μηνυμάτων, τα οποία θα καλούνται αυτόματα. Οι συνδέσεις αντιπροσωπεύονται από τα αντικείμενα `fluid_base :: OFConnection`.

Παρακάτω οι μετρήσεις που πήραμε χρησιμοποιώντας το Cbench:



Σχήμα 4.7.1 LIBFLUID base throughput



Σχήμα 4.7.2 LIBFLUID base throughput

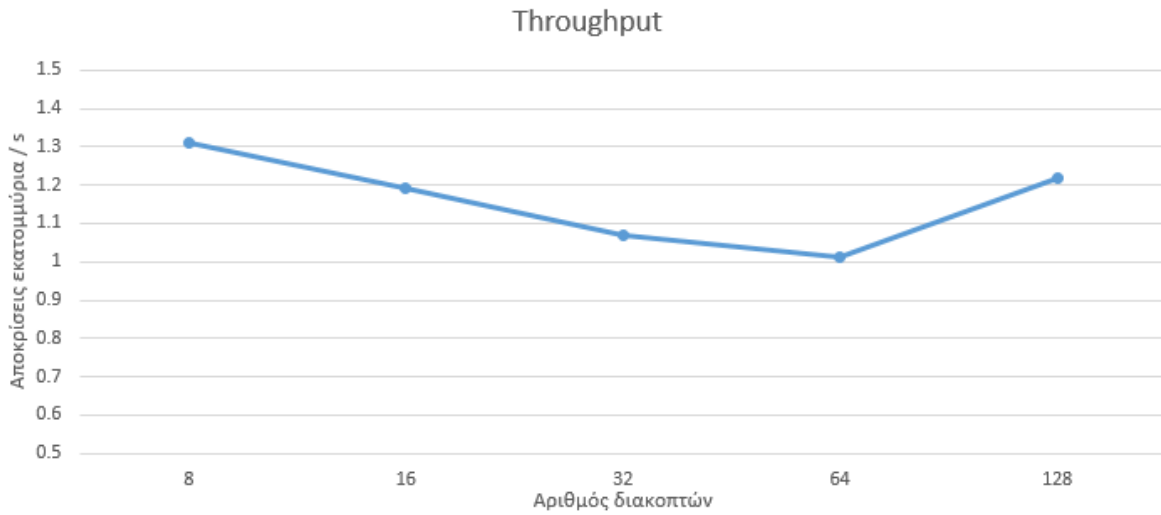
#### 4.4.8 OPENIRIS

Το OpenIRIS είναι μια έκδοση ανοιχτού κώδικα του IRIS. Το IRIS είναι ένας ελεγκτής SDN που βασίζεται στο Openflow, ο οποίος έχει σχεδιαστεί για να επιλύει τα ζητήματα κλιμάκωσης και διαθεσιμότητας του SDN [43].

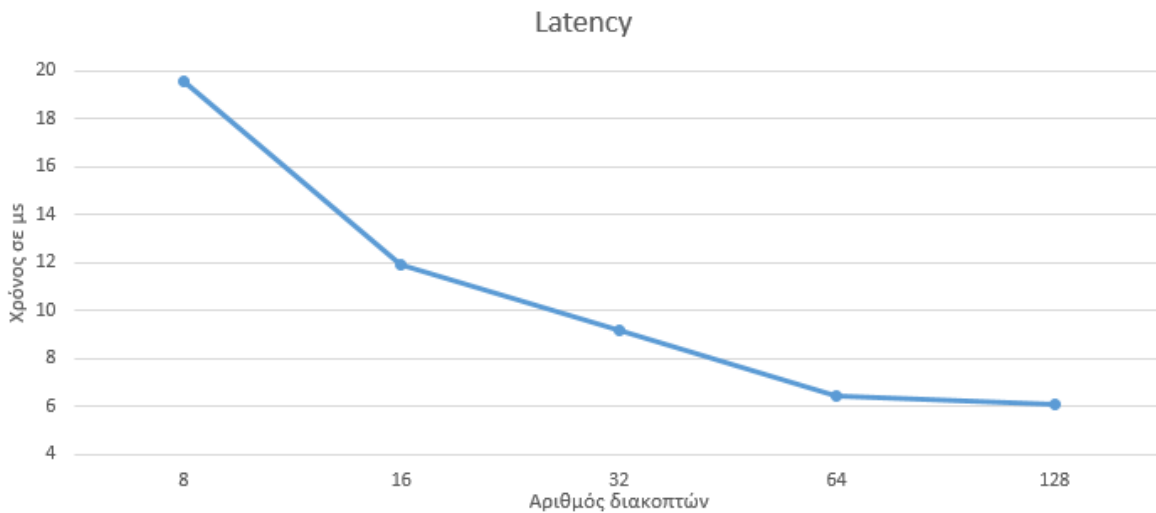
Βασικά χαρακτηριστικά του OpenIris:

- Υποστηρίζει Openflow 1.0.1 ~ 1.3.2.
- Το API Openflow του έχει βάση το Loxigen.
- Καθαρή υλοποίηση που βασίζεται στη Java.
- Υποστηρίζει περίπου 500 ταυτόχρονες συνδέσεις με διακόπτες με το Commodity HW
- Ακολουθεί την ίδια πολιτική αδειοδότησης με το Floodlight (άδεια Apache-2.0)
- Παρέχει learning mac, ανακάλυψη συνδέσμου, διαχείριση τοπολογίας, προώθηση, διαχειριστή συσκευών, τείχος προστασίας, net failover, static flow pusher και modules διαχείρισης κατάστασης.
- Περιλαμβάνει τρεις υλοποιήσεις ελεγκτή, οι οποίες είναι πλήρως λειτουργικές.

Παρακάτω οι μετρήσεις που πήραμε χρησιμοποιώντας το Cbench:



Σχήμα 4.8.1 OPENIRIS throughput



Σχήμα 4.8.2 OPENIRIS latency

#### 4.4.9 ONOS

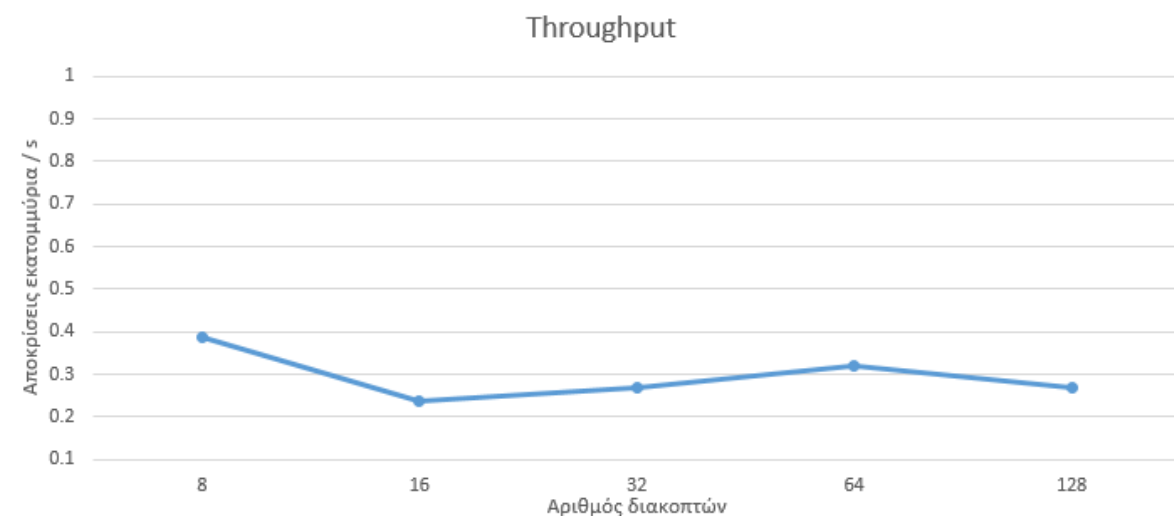
Το ONOS σημαίνει Open Network Operating System. Το ONOS παρέχει το επίπεδο ελέγχου για ένα δίκτυο διαχειριζόμενο από λειτουργικό (SDN), διαχειρίζεται τα στοιχεία του δικτύου (όπως διακόπτες και συνδέσμους) και εκτελεί προγράμματα ή modules για την παροχή υπηρεσιών επικοινωνίας στους τελικούς κεντρικούς υπολογιστές και τα γειτονικά δίκτυα [44].

Ο kernel του ONOS, οι βασικές υπηρεσίες και οι εφαρμογές του, είναι γραμμένες σε Java ως bundles που φορτώνονται στο Karaf OSGi. Το OSGi είναι ένα σύστημα στοιχείων για την Java που επιτρέπει στις λειτουργικές μονάδες να εγκαθίστανται και να εκτελούνται δυναμικά σε ένα ενιαίο JVM. Από τη στιγμή που το ONOS τρέχει στο JVM, μπορεί να τρέξει σε διάφορες πλατφόρμες λειτουργικών συστημάτων.

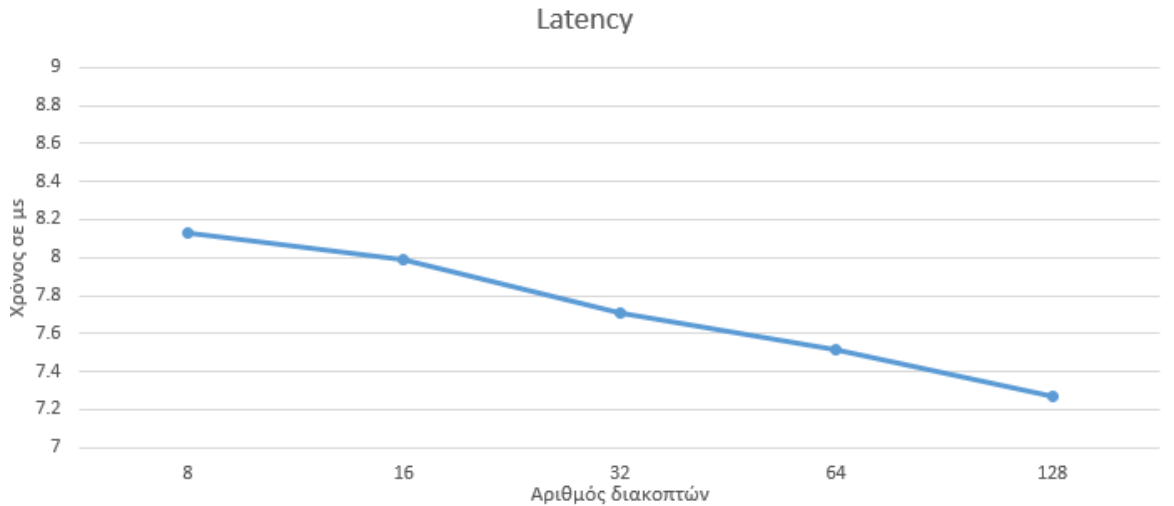
Χαρακτηριστικά του Onos είναι:

- Ευελιξία: Προσφέρει εικονικά απεριόριστη αναπαραγωγή για το scaling της χωρητικότητας του επιπέδου ελέγχου.
- Υψηλή απόδοση: Λειτουργεί στις ακριβείς προδιαγραφές των operators των μεγάλων δικτύων.
- Ανθεκτικότητα: Παρέχει τη διαθεσιμότητα που απαιτείται για λειτουργίες κρίσιμης σημασίας στο δίκτυο.
- Υποστήριξη συσκευών παλαιού τύπου. Κάνει εύκολη την προσθήκη ή τη διαμόρφωση παραδοσιακών συσκευών και υπηρεσιών.
- Υποστήριξη συσκευών επόμενης γενιάς: Παρέχει έλεγχο σε πραγματικό χρόνο για φυσικές συσκευές δεδομένων SDN με OpenFlow.

Παρακάτω οι μετρήσεις που πήραμε χρησιμοποιώντας το Cbench:



Σχήμα 4.9.1 ONOS throughput



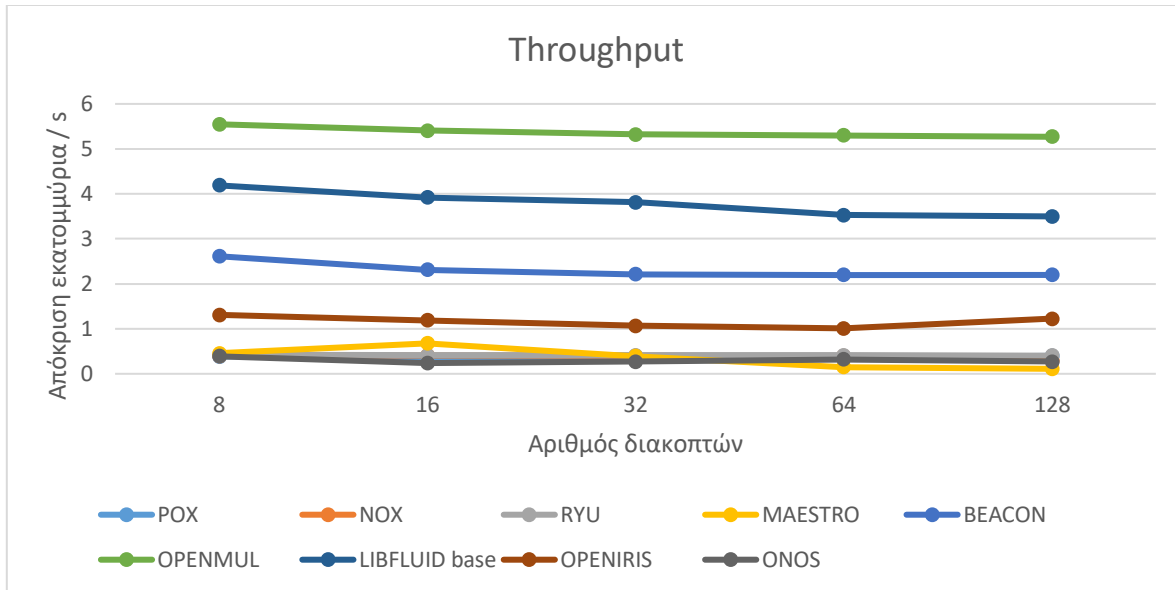
Σχήμα 4.9.2 ONOS latency

## ΚΕΦΑΛΑΙΟ 5

### 5 ΣΥΜΠΕΡΑΣΜΑΤΑ

#### 5.1 ΑΝΑΛΥΣΗ ΜΕΤΡΗΣΕΩΝ

Στην 1<sup>η</sup> δοκιμή που ήταν σχετική με το throughput τα αποτελέσματα δείχνουν ότι οι ελεγκτές που κωδικοποιούνται στη γλώσσα C έδωσαν την υψηλότερη απόδοση. Συγκεκριμένα πρώτος ήταν ο OpenMUI και δεύτερος ο Libfluid. Αμέσως μετά ήρθαν οι ελεγκτές που κωδικοποιούνται στην γλώσσα Java. Δηλαδή στην γενική κατάταξη τρίτος ο Beacon και τέταρτος ο Iris (σχήμα 5.1). Ο Maestro ενώ μέχρι και τα 16 switches είχε καλή απόδοση, στα 32 έπεσε και από εκεί και πέρα υπολειτουργούσε. Η συμπεριφορά του ίσως οφείλεται στο ότι χρησιμοποιεί round robin αλγόριθμο για την διαχείριση των εισερχομένων πακέτων ενώ οι υπόλοιποι ελεγκτές αφιερώνουν τους διαθέσιμους πόρους τους στις συνδέσεις τους με τους διακόπτες.

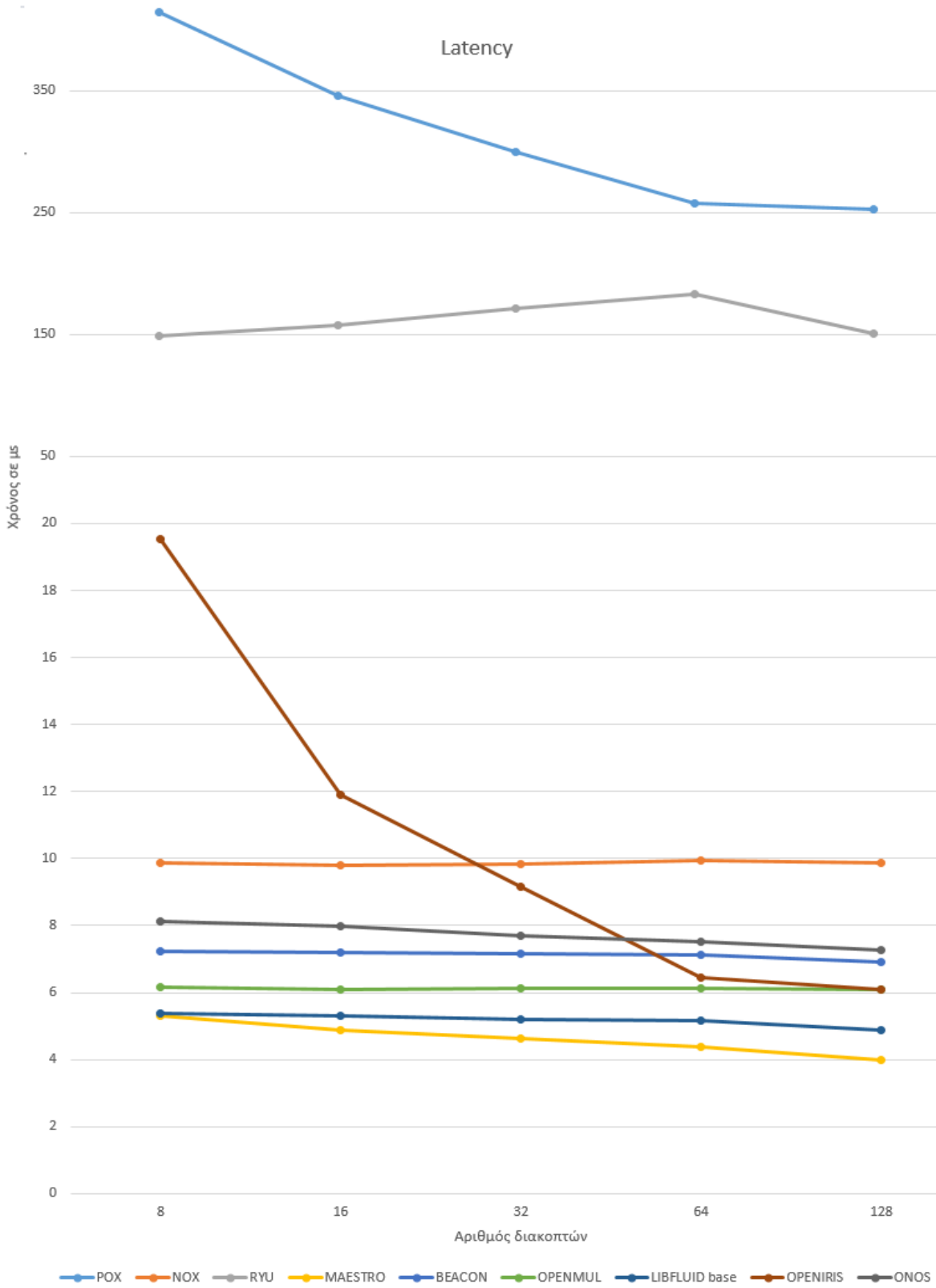


Σχήμα 5.1 Controllers throughput

Στη 2<sup>η</sup> δοκιμή σχετική με το latency ο Maestro χρησιμοποιώντας την προσαρμοστική λειτουργία δοσοληψίας (adaptive batching mode), έδωσε την καλύτερη μέτρηση (σχήμα 5.2), με δεύτερο τον libfluid.

Παρατηρούμε ότι overall καλύτεροι ελεγκτές, δηλαδή με το καλύτερο throughput και καλό latency είναι και πάλι αυτοί που είναι γραμμένοι στην C.

Αξίζει να σημειωθεί ότι ελεγκτές που κωδικοποιούνται στη C και τη java προσφέρουν υψηλότερη απόδοση όταν αλλάζει ο αριθμός των thread [45]. Ωστόσο, ο POX που κωδικοποιείται στην Python δεν παρουσιάζει καμία σημαντική διαφορά όταν χρησιμοποιείτε multithreading επειδή η υποστήριξη της Python για το multithreading δεν είναι πολύ αποτελεσματικός.



Σχήμα 5.2 Controllers latency

## 5.2 ΕΠΙΛΟΓΟΣ

Παραπάνω είδαμε την συμπεριφορά των ελεγκτών σε 2 από τα στοιχεία που καθορίζουν την αποτελεσματικότητά τους. Αν και κάποιοι ελεγκτές ξεχωρίζουν θα πρέπει να τονίσουμε η επιλογή ενός ελεγκτή καθορίζεται από πολλούς παράγοντες καθώς επίσης και ότι θα πρέπει να επιλέξουμε τον ελεγκτή που χρειαζόμαστε ανάλογα με τις ανάγκες μας.

Οι παραπάνω μετρήσεις έχουν ληφθεί σε ιδανικές συνθήκες, το latency σε πραγματικές συνθήκες είναι σίγουρα μεγαλύτερο καθώς στον κανονικό κόσμο υπάρχουν περιορισμοί.

Τέλος όπως βλέπουμε και στο σχήμα 3.4 στην ενότητα τα χαρακτηριστικά των ελεγκτών, κάποια χαρακτηριστικά μπορεί να υποσκιάσουν την απόδοση ή και το αντίστροφο. Για παράδειγμα το υψηλό modularity του ONOS που είναι πολύ σημαντικός παράγοντας για την ανάπτυξη εφαρμογών.



## Πηγες:

- [1] Benzekki Kamal et al. Software-defined networking (SDN): Security and Communication Networks 9, no. 18 (2016): 5803-5833.
- [2] Open Networking Foundation (2014, May 20). "Open networking foundation specifications". [Online]. Available at: <https://www.opennetworking.org/sdnresources/onf-specifications>.
- [3] "Software-defined networking is not OpenFlow, companies proclaim". [Online]. Available at: [searchsdn.techtarget.com](http://searchsdn.techtarget.com).
- [4] "InCNTRE's OpenFlow SDN testing lab works toward certified SDN product"
- [5] Benzekki Kamal et al. Devolving IEEE 802.1 X authentication capability to data plane in software-defined networking (SDN) architecture., Security and Communication Networks 9.17 (2016): 4369-4377.
- [6] Benzekki Kamal et al. Software-defined networking (SDN): a survey., Security and Communication Networks 9, no. 18 (2016): 5803-5833.
- [7] "Design Elements - Telecommunication networks". [Online]. Available at: [conceptdraw.com](http://conceptdraw.com)
- [8] Noormohammadpour, Mohammad; Raghavendra, Cauligi (28 July 2018). "Datacenter Traffic Control: Understanding Techniques and Tradeoffs". Communications Surveys & Tutorials, IEEE. 20 (2): 1492-1525.
- [9] ZK Research Whitepaper Compute Transitions Drive the Need for the New IP Network White Paper, OCT. 2014.
- [10] 7 Advantages of Software Defined Networking by Ingram. [Online]. Available at: [imagineNEXT.ingrammicro.com/trends/august-2017/7-advantages-of-software-defined-networking](http://imagineNEXT.ingrammicro.com/trends/august-2017/7-advantages-of-software-defined-networking)
- [11] Y. Hu, W. Wendong, X. Gong, X. Que and C. Shiduan "Reliability-aware controller placement for Software-Defined Networks". In 2013 IFIP/IEEE

International Symposium on Integrated Network Management (IM 2013), pp. 672-675, May 2013.

[12] R. Wang, D. Butnariu and J. Rexford (2011). "OpenFlow-based server load balancing gone wild". In Proceeding of the USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE), Boston, MA, USA, Mar. 2011.

[13] F. Hu, Q. Hao, and K. Bao. "A Survey on Software Defined Networking (SDN) and OpenFlow: From Concept to Implementation." IEEE Communications Surveys & Tutorials, no. 99, pp. 1-1, May 2014.

[14] Top Five Apps and Services That Can Benefit from SDN. [Online]. Available at: <https://www.datacenterknowledge.com/archives/2016/03/31/top-five-apps-and-services-that-can-benefit-from-sdn>

[15] G. Kontesidou and Z. Kyriakos, "Open flow Virtual Networking: A Flow-Based Network VirtualizationArchitecture." Royal Institute of Technology(KTH), 2009.

[16] Cloud and Datacenter Networking Course – Prof. Roberto Canonico – Università degli Studi di Napoli Federico II Available at: <http://wpage.unina.it/rcanonic/didattica/dcn/lucidi/DCN-L10-b-OpenFlow.pdf>

[17] Open networking summit, (2012. April) "SDN Tutorial for Engineers," open networking summit" [Online]. Available at: <http://www.opennetsummit.org/archives/apr12/heller-mon-intro.pdf>.

[18] OpenFlow version 1.0 (ONF). Available at: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf>

[19] Feng Wang, Heyu Wang, Baohua Lei and Wenting Ma, "A Research on High-Performance SDN Controller," Cloud Computing and Big Data (CCBD), 2014 International Conference on, pp. 168-174

[20] O.N. Foundation, "Software-defined networking: The new norm for networks," ONF White Paper

- [21] Ashton, Metzler. "Ten Things to Look for in an SDN Controller." white paper [online]. Available at: [www.necam.com/Docs](http://www.necam.com/Docs) (2013)
- [22] Sridhar Rao, "SDN Series Part Eight: Comparison of Open Source SDN Controllers" [online]. Available at: <http://thenewstack.io/sdn-series-part-eight-comparison-ofopen-source-sdn-controllers/>
- [23] SDN Workshop by apnic [Online]. Available at: [https://wiki.apnictraining.net/\\_media/apnic44/track4agenda/wsdn01\\_-\\_module\\_2\\_-\\_openflow\\_-\\_v1.0.pdf](https://wiki.apnictraining.net/_media/apnic44/track4agenda/wsdn01_-_module_2_-_openflow_-_v1.0.pdf)
- [24] R. Sherwood and K.-K. Yap, "Cbench Controller Benchmark," 2011. [Online]. Available: <https://github.com/andi-bigswitch/oflops/tree/master/cbench>
- [25] A. Shalimov, D. Zuikov, and D. a. Zimarina, "Advanced Study of SDN/OpenFlow Controllers," in Proceedings of the Central Eastern European Software Engineering Conference. ACM, 2013, pp. 1–6.
- [26] GitHub, "WCBench:CBench, Wrapped in stuff that makes it Useful." [Online]. Available: <https://github.com/dfarrell07/wcbench>
- [27] M. Jarschel, F. Lehrieder, Z. Magyari, and R. Pries, "A Flexible OpenFlow-Controller Benchmark," in 2012 European Workshop on Software Defined Networking, 2012, pp. 48–53
- [28] "OFCProbe: A Platform Independent Tool for OpenFlow Controller Analysis." [Online]. Available: <https://www3.informatik.uni-wuerzburg.de/research/ngn/ofcprobe/ofcprobe.shtml>
- [29] M. Team, "Mininet: An Instant Virtual Network on your Laptop (or other PC)." [Online]. Available: <https://mininet.org/>
- [30] "PktBlaster SDN Datasheet," Veryx Technologies, Tech. Rep., 2016. [Online]. Available: <http://www.veryxtech.com/wp-content/uploads/2015/10/Datasheet-PktBlaster-SDN-Con>

- [31] G. H. Shankar, "OFNet Quick User Guide." [Online]. Available: <http://sdninsights.org/>
- [32] Virtual Box website: <https://www.virtualbox.org/>
- [33] Xming information: <https://en.wikipedia.org/wiki/Xming>
- [34] Putty information: <https://en.wikipedia.org/wiki/PuTTY>
- [35] Cbench tool <https://github.com/mininet/oflops/tree/master/cbench>
- [36] POX tutorial Available at: <http://www.brianlinkletter.com/using-the-pox-sdn-controller/>
- [37] NOX controller Available at: <https://github.com/noxrepo/nox>
- [38] Ryu controller Available at: <https://osrg.github.io/ryu/>
- [39] Maestro controller Available at: <http://zhengcai.github.io/maestro-platform/>
- [40] Beacon controller Available at: <https://openflow.stanford.edu/display/Beacon/Home.html>
- [41] OpenMul controller Available at: <https://www.sdxcentral.com/projects/openmul-controller/>
- [42] LibFluid Base documentation Available at: [http://opennetworkingfoundation.github.io/libfluid/md\\_libfluid\\_base\\_doc\\_Intro.html](http://opennetworkingfoundation.github.io/libfluid/md_libfluid_base_doc_Intro.html)
- [43] OpenIris Available at: <http://openiris.etri.re.kr/>
- [44] ONOS Available at: <https://wiki.onosproject.org/display/ONOS/Wiki+Home>
- [45] Ola Salman, Ayman Kayss, Imad H. Elhadj, Ali Chehab: SDN controllers: A comparative study [https://www.researchgate.net/publication/304457462\\_SDN\\_controllers\\_A\\_comparative\\_study](https://www.researchgate.net/publication/304457462_SDN_controllers_A_comparative_study)