

**Τμήμα
Μηχανικών
Πληροφορικής τ.ε.**

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Software Defined Networking

- *Μελέτη, ανάλυση και υλοποίηση* -

Λούνγκου Βασίλε

Επιβλέπων καθηγητής Β.Ταμπακάς

Οκτώβριος 2016

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

- 1.
- 2.
- 3.

Περίληψη

Όντας σε επαφή συγκεκριμένα με τον χώρο των δικτύων και κινούμενοι στον χώρο των υπολογιστών γενικότερα μπορούμε εύκολα να αντιληφθούμε την τάση που υπάρχει προς την ανάπτυξη των τεχνολογιών του virtualizing και του cloud. Πρόκειται για τεχνολογίες που ήδη μέσα από διάφορα εργαλεία και υπηρεσίες βρίσκονται στην καθημερινότητα των περισσότερων χρηστών Η/Υ. Καθώς όμως οι έννοιες αυτές γνωρίζουν όλο και μεγαλύτερη διεύρυνση και τείνουν να γίνουν το νέο πρότυπο στην παροχή διαδικτυακών υπηρεσιών τίθεται επιτακτικά το θέμα της παραμετροποίησης των δικτύων ώστε να ανταποκριθούν στην αύξηση της κίνησης στα τοπικά εταιρικά/πανεπιστημιακά και άλλα δίκτυα.

Σε μερίδα μηχανικών δικτύων υπάρχει η αντίληψη πως αυτές οι εξελίξεις δεν θα έχουν μεγάλο αντίκτυπο στον τομέα μας και πως οι όποιες επιπλέον απαιτήσεις δημιουργηθούν από την αύξηση της κίνησης θα λυθούν με αντίστοιχη επέκταση του δικτυακού εξοπλισμού και την αύξηση του εύρους ζώνης (bandwidth) των δικτυακών μέσων. Η αντίληψη αυτή είναι βιώσιμη ως ένα σημείο, όμως υπάρχει αντίστοιχα η άποψη που θέτει επί τάπητος τολμηρότερες αλλαγές στον σχεδιασμό και διαχείριση των δικτύων ώστε να ανταποκρίνονται αποδοτικότερα στις νέες συνθήκες. Η άποψη αυτή όμως δεν είναι πλέον μια υπόθεση εργασίας, παίρνει μορφή μέσα από διάφορες νέες αρχιτεκτονικές, πρωτόκολλα και εργαλεία που επιτρέπουν την αναμόρφωση του δικτυακού τοπίου και προσθέτει μια σειρά δυνατοτήτων και πλεονεκτημάτων συγκριτικά με την υπάρχουσα μορφή των δικτύων.

Το SDN, Software Defined Networking αναπτύχθηκε τροφοδοτούμενο από μια σειρά λόγων και καλύπτοντας μια σειρά αναγκών που βρίσκονται μέσα στα όρια της κατάστασης που προαναφέραμε, προσφέροντας μια σημαντική καινοτομία στον χώρο των δικτύων.

Εκτός λοιπόν από μια ενδιαφέρουσα εξέλιξη επί του επιστημονικού μου πεδίου πιστεύω πως το SDN είναι κάτι που ανοίγει μια σειρά δυνατοτήτων, θεώρησα έτσι ενδιαφέρουσα την εμβάθυνση στην νέα αυτή αρχιτεκτονική και την έρευνα του τρόπου λειτουργίας καθώς και των καινοτομιών που κομίζει στον τομέα μας.

ABSTRACT

As clouding and virtualization tools and services become a growing industry and a big part of computing today, we, the people involved in IT and Networking become more aware of the importance that comes with those advancements. Those are not just trends but are already the new standard build into our everyday interaction and user experience with several internet and social networks services. From a networking perspective, we can surely say that we need to reconfigure the way networks work, to create bigger infrastructure and more bandwidth in order to accommodate the new needs.

The typical answer when dealing with these kind of problems does not seem enough for some networking specialists that suggest that we don't only need more infrastructure and bandwidth but also need to rethink of how we perceive and configure networks. To come up with new solutions, created with out of the box thinking, to change many things in the status quo of the networking structure that has served us well for so many years, to reshape the form of existing networks with new protocols, architectures and networking implementation in general.

So, with new challenges new answers and ways of doing thing are required, such as Software Defined Networking, a new networking architecture that changes a lot in the way we think about networks, how we implement and manage them. It offers a better and more convenient way to do work with networks already deployed as well and does not require extended hardware changes in order to make it work. It's a brave new world with SDN and it offers a new filed of interesting new technologies ant tools to experiment with and explore the barriers of the new era networking. As such, SDN is an architecture/technology that is quite interesting and worth investing time and resources to learn and understand.

So, my thesis is about Software Defined Networking, its capabilities, implementations and way of working.

Περιεχόμενα

ΠΡΟΛΟΓΟΣ	ΣΦΑΛΜΑ! ΔΕΝ ΕΧΕΙ ΟΡΙΣΤΕΙ ΣΕΛΙΔΟΔΕΙΚΤΗΣ.
1. ΕΙΣΑΓΩΓΗ.....	11
1.1 Virtualization.....	11
1.2 Το virtualization στην καθημερινότητα.....	11
1.3 Network virtualization	11
1.4 Network virtualization μέσω SDN	12
2. SOFTWARE DEFINED NETWORKING	12
2.1 Ιστορικές αναφορές.....	12
2.2 Ορισμός.....	12
2.3 Χαρακτηριστικά SDN.....	13
2.4 SDN Επίπεδα.....	15
2.5 Προβληματισμοί/Ανησυχίες	16
2.5.1 Ασφάλεια.....	16
2.5.2 Προσωπικό.....	18
3. ΥΠΑΡΧΟΥΣΕΣ SDN ΥΛΟΠΟΙΗΣΕΙΣ	18
3.1 Εφαρμοσμένη αρχιτεκτονική	18
3.1.1 Google	19
3.1.2 NSA.....	19
3.1.3 FACEBOOK	19
3.1.4 MICROSOFT	20
3.2 Controllers – Υλοποιήσεις.....	20
3.3 Δραστηριότητα στον χώρο του SDN	23
3.4 White box Switches.....	24
3.5 Startups.....	25
3.6 SDN συμβατό Hardware-Switches	26
4. SDN LAYERS.....	29

4.1	Επίπεδο Υλικού	29
4.1.1	Δίκτυο	29
4.1.2	OpenFlow	33
4.1.3	Pipeline processing.....	34
4.1.4	Flow Tables	35
4.1.5	Flow Entries.....	35
4.1.6	Group table	37
4.1.7	Meter table	37
4.1.8	Control Channel	37
4.2	Επίπεδο Ελέγχου.....	41
4.2.1	Controllers - Λειτουργία	41
4.2.2	Northbound	42
4.2.3	Southband.....	42
4.2.4	East-West	42
4.2.5	Single - Multicontroller architectures	43
4.2.6	Πρόληψη δυσλειτουργιών	44
4.3	Επίπεδο Εφαρμογών	45
4.3.1	Εφαρμογές Ασφάλειας	47
4.3.2	Εφαρμογές Διαχείρισης	47
5.	ΥΛΟΠΟΙΗΣΗ.....	48
5.1	Εισαγωγικά.....	48
5.2	Επιλογή δικτύου, controller και applications.....	49
5.3	VirtualBox	50
5.3.1	VirtualBox Guest Additions	51
5.3.2	VirtualBox Shared folder	52
5.4	Εγκατάσταση Mininet	53
5.5	Εγκατάσταση OpenDaylight	53
5.6	Εγκατάσταση HP VAN SDN Controller	55
5.7	Χρήση Mininet.....	58
5.8	Χρήση OpenDaylight.....	60
5.8.1	Είσοδος.....	61
5.8.2	Περιβάλλον	62
5.8.3	Τοπολογία	62
5.8.4	Nodes – στοιχεία δικτύου.....	65
5.8.5	Yang UI	67
5.8.6	Wireshark – OpenFlow Captures	69
5.9	Χρήση HP VAN SDN Controller	75

5.9.1	Εισαγωγή	75
5.9.2	Είσοδος	76
5.9.3	Περιβάλλον	76
5.9.4	Τοπολογία	77
5.9.5	Monitor	78
5.9.6	Trace	79
5.9.7	Applications.....	82
5.9.8	Flow Maker	83
6.	ΣΥΜΠΕΡΑΣΜΑ	87
6.1	Τι πραγματικά είναι το SDN	87
6.2	Συμπεράσματα από την υλοποίηση	87
6.3	Το παρόν και το μέλλον του SDN	88
6.3.1	Επί του παρόντος	88
6.3.2	Επί του μέλλοντος	89
7.	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	90

Κατάλογος Εικόνων

Εικόνα 1 Τα τρία επίπεδα λειτουργίας της SDN λειτουργίας.....	15
Εικόνα 2 Εφαρμογή ασφάλειας της HP για SDN δίκτυα	17
Εικόνα 3 Wedge και brack Switches του Facebook	20
Εικόνα 4 Εταιρίες με συμμετοχή στην ανάπτυξη του OpenDaylight	22
Εικόνα 5 Οι εταιρικές συμμετοχές ανά τομέα του SDN οικοσυστήματος.....	24
Εικόνα 6 White-Box Switch της Pica8 και η Nexus 9000 σειρά της Cisco	27
Εικόνα 7 Κανονική λειτουργία Switch	31
Εικόνα 8 Switch ελεγχόμενο από εξωτερικό Controller	32
Εικόνα 9 Η OpenFlow εγκατάσταση εσωτερικά του Switch	33
Εικόνα 10 Η ροή των πακέτων Pipeline Processing μέσω των Flow Tables.....	34
Εικόνα 11 Αναλυτική λειτουργία Flow Table.....	35
Εικόνα 12 Παράδειγμα Flow Entry.....	36
Εικόνα 13 Οι κατηγορίες των Tables και παράδειγμα Flow Table	37
Εικόνα 14 Διάγραμμα του ONE, Controller της Cisco	41
Εικόνα 15 Το SDN app store της Hewlett Packard.....	46
Εικόνα 16 Το VirtualBox με το mininet και τους Controllers εγκατεστημένους	50
Εικόνα 17 Παράδειγμα του κειμένου στον /etc/network/interfaces	51
Εικόνα 18 Η δημιουργία εικονικού δικτύου σε περιβάλλον Mininet	58
Εικόνα 19 Απλή τοπολογία δικτύου στον Opendaylight Controller παραγόμενη μέσω Mininet.....	62
Εικόνα 20 Torus τοπολογία με τέσσερα κανάλια επικοινωνίας	63
Εικόνα 21 Παράδειγμα διαχείρισης τοπολογίας 121 Switches	64
Εικόνα 22 Πληροφορίες σχετικά με τα Nodes όπως εμφανίζονται σε OpenDaylight και Mininet.....	65
Εικόνα 23 Στατιστικά πακέτων και Flow Entries όπως αυτά εμφανίζονται σε OpenDaylight και Mininet.....	66

Εικόνα 24 Πίνακας πληροφοριών σχετικά με Switch	68
Εικόνα 25 Πεδίο εισαγωγής Flow Entries στο Yang UI	69
Εικόνα 26 Τα πρώτα μηνύματα που αποστέλλονται μεταξύ Switches-Controllers	71
Εικόνα 27 Ένα αναλυτικό Features_Reply μήνυμα	73
Εικόνα 28 Ping πακέτα μεταξύ των Hosts/Υπολογιστών του δικτύου	74
Εικόνα 29 Τοπολογία Mininet σε HP VAN Controller	77
Εικόνα 30 Τα FlowTables όπως εμφανίζονται στον HP VAN Controller και στο Mininet αντίστοιχα	79
Εικόνα 31 Καταγραφή των πακέτων που ανταλλάσσονται μεταξύ Controller-Switches στα πρώτα δευτερόλεπτα σύνδεσης	80
Εικόνα 32 Η ανταλλαγή πακέτων που εκτελείται λόγω pingall	81
Εικόνα 33 Κατηγορίες προγραμμάτων στο AppStore της HP	82
Εικόνα 34 Εγκατάσταση εφαρμογής.....	83

Ευχαριστίες

Το τέλος της παρούσας πτυχιακής εργασίας σηματοδοτεί επίσης το τέλος των προπτυχιακών μου σπουδών. Σπουδές που μου επέτρεψαν να μαθητεύσω πάνω σε ένα αντικείμενο ιδιαίτερου ενδιαφέροντος και μεγάλης σημασίας στον σύγχρονο τεχνολογικό πολιτισμό μας.

Σε αυτό το σημείο αισθάνομαι την ανάγκη να απευθύνω τις ευχαριστίες μου στα άτομα που βοήθησαν στην εκπόνηση της παρούσας πτυχιακής και που χωρίς την συνδρομή τους η παρούσα εργασία θα ήταν φτωχότερη.

Κατά πρώτον ευχαριστώ θερμά τον καθηγητή μου κ. Γιάννη Τζήμα (Καθηγητή τμήματος Μηχανικών Πληροφορικής ΤΕΙ Δυτικής Ελλάδας) για την επίβλεψη της εργασίας, τις πολύτιμες συμβουλές του, την υπομονή του κάποιες φορές, και την συμπαράσταση του για την ολοκλήρωση της παρούσας πτυχιακής.

Ευχαριστώ το ίδιο θερμά τον κ. Βασίλη Ταμπακά (Καθηγητή τμήματος Μηχανικών Πληροφορικής ΤΕΙ Δυτικής Ελλάδας) για την ανάληψη και επίβλεψη της συνέχειας και έως την ολοκλήρωσης της πτυχιακής αυτής εργασίας.

Ιδιαίτερες ευχαριστίες θέλω να απευθύνω και στην κοινότητα Ελευθέρου Λογισμικού του ΤΕΙ Δυτικής Ελλάδας για την υποστήριξη και την βοήθεια τους σε τεχνικά ζητήματα που ξεπεράστηκαν χάρη στις γνώσεις και το μεράκι τους.

Κλείνοντας θα ήθελα να ευχαριστήσω εκ βάθους καρδιάς την οικογένεια μου που χωρίς την διαχρονική στήριξη που μου προσέφερε και την ψυχική και υλική βοήθεια δεν θα είχα καταφέρει να ολοκληρώσω τίποτε τις σπουδές μου, πόσο μάλλον την παρούσα πτυχιακή εργασία.

1. ΕΙΣΑΓΩΓΗ

1.1 Virtualization

Virtualisation, ως έννοια ξεκίνησε το 1960 και περιέγραφε την μέθοδο διαμοιρασμού των πόρων στα mainframes της εποχής, όμως μαζί με τον μετασχηματισμό και την πρόοδο της τεχνολογίας H/Y ο όρος άλλαξε και πλέον περιλαμβάνει ένα εύρος εφαρμογών στο πεδίο των λειτουργικών συστημάτων, αποθηκευτικού χώρου, υπολογιστικού hardware και δικτύων. Ένας συντομευμένος ορισμός του virtualizing είναι η δημιουργία εικονικών (αντί πραγματικών) εκδόσεων ενός προγράμματος ή ενός μηχανήματος. Πιο πρακτικά μπορούμε να πούμε πως είναι η δυνατότητα παράλληλης λειτουργίας πολλαπλών επιπέδων λογισμικού/υλικού στο ίδιο σύστημα ως προσομοιώσεις πραγματικών συστημάτων.

1.2 Το virtualization στην καθημερινότητα

Τα εικονικά μηχανήματα φαντάζουν μια αόριστη ιδέα την οποία πολλοί από μας είδαν στην πράξη πειραματιζόμενοι με διάφορα λειτουργικά συστήματα που εγκαθιστούσαμε μέσω VirtualBox και VMWare. Αμέσως ξεχώρισε η ευκολία με την οποία μπορούσαμε να έχουμε οποιοδήποτε software στο σύστημα μας ανεξαρτήτως λειτουργικού συστήματος, σε ένα Linux σύστημα μπορούμε να λύσουμε τις ασυμβατότητες υλικού τρέχοντας ένα Windows εικονικό μηχανήμα. Εξαφανιζόταν ο περιορισμός των στατικών partitions, με την ευελιξία του virtualisation μπορούσαμε να αξιοποιήσουμε σε πραγματικό χρόνο το σύνολο του hardware για πολυεπίπεδες διεργασίες και όταν πια δεν χρειαζόταν το εικονικό μηχανήμα απλά γινόταν να διαγραφεί με ελάχιστον footprint και χωρίς format.

1.3 Network virtualization

Τα πλεονεκτήματα του virtualization να είναι εμφανή και σε επίπεδο server όπου η παραμετροποιήσεις πραγματικού χρόνου χωρίς διακοπές στην παροχή υπηρεσιών και η αποτελεσματική αξιοποίηση του υπάρχοντος hardware είναι κομβικά για την αξιοπιστία και την οικονομική σταθερότητα των εταιριών που τα χειρίζονται. Το νέο status quo που διαμορφώνεται στο διαδίκτυο με την άνοδο των multimedia content και social networking τάσεων δημιουργεί έναν όλο και μεγαλύτερο όγκο ροής δεδομένων που χρειάζεται μια καινούργια αντίληψη για την αποτελεσματική διαχείριση του. Απάντηση

σε αυτό είναι οι δυναμική διαχείριση των πόρων του δικτύου, δυναμικότητα που επιτυγχάνεται μέσω ενός πολυεπίπεδου virtualization με κεντρικοποιημένες δομές που βλέπουν και διαχειρίζονται το σύνολο των πόρων ως μια οντότητα.

1.4 Network virtualization μέσω SDN

Ακρογωνιαίο στοιχείο στην μετάβαση και εφαρμογή του virtualisation σε επίπεδο μεγάλων δικτύων/datacenters και δικτύων συνολικότερα είναι η Software Defined Networking αρχιτεκτονική, που προσφέρει μια σειρά από πλεονεκτήματα συγκριτικά με τα κανονικά. Επιτρέπει στο δίκτυο να είναι άμεσα προγραμματιζόμενο, ευέλικτο, γρήγορο, ασφαλές και είναι βασισμένο κυρίως σε πρότυπα ανοιχτού κώδικα. Ο συνδυασμός cloud, SDN, και άλλων networking/virtualisation εργαλείων έχει ήδη ξεκινήσει να αναμορφώνει τον τρόπο που λειτουργούν τα μεγάλα δίκτυα και είναι ένα ζήτημα που παρουσιάζει ιδιαίτερο ενδιαφέρον και προοπτικές όσον αφορά την επιστήμη μας.

2. Software Defined Networking

2.1 Ιστορικές αναφορές

Η ιδέα για δίκτυα που μπορούν να προγραμματιστούν και να ξεπεραστούν οι περιορισμοί του υλικού υπήρχε αρκετό καιρό και έγιναν διάφορες κινήσεις εταιριών και πανεπιστημίων προς την κατεύθυνση αυτή. Ως αρχιτεκτονική το Software Defined Networking βρήκε έναν σταθερό βηματισμό μέσα από την ερευνητική σύμπραξη των πανεπιστημίων του Berkeley και του Stanford την περίοδο του 2008 και πήρε την συγκεκριμένη μορφή που έχει σήμερα μέσα από το Open Networking Foundation που ιδρύθηκε το 2011 και έθετε το πλαίσιο πάνω στο οποίο θα αναπτυσσόταν αυτή η νέα τεχνολογία.

2.2 Ορισμός

Ως προς το τι ακριβώς είναι το Software Defined Networking θα επικαλεστούμε τον ορισμό που δίνει το Open Networking Foundation που χαρακτηρίζει το SDN ως τον

φυσικό διαχωρισμό του επιπέδου ελέγχου (Control Plane) από το επίπεδο προώθησης δεδομένων (forwarding-data plane), όπου το επίπεδο ελέγχου μπορεί να ελέγχει πολλαπλές συσκευές. (1.ONF) Θα πρέπει ωστόσο να ληφθεί υπόψη πως καθώς πρόκειται για μια αρχιτεκτονική που βρίσκεται σε εξέλιξη και ανάλογα με την ομάδα ανάπτυξης κάποιον πανεπιστημίων ο ορισμός, οι στόχοι και ο τρόπος υλοποίηση του Software Defined Networking μπορεί να έχουν μικρές διαφοροποιήσεις.

2.3 Χαρακτηριστικά SDN

Ευελιξία Η μεταφορά του Control Plane σε εξωτερικό περιβάλλον επιτρέπει μέσω ενός Controller την απευθείας πρόσβαση στις λειτουργίες πολλαπλών Switches που ως δυνατότητα κάνει την διαχείριση και την αλλαγή παραμέτρων σε αυτά πολύ ευκολότερη από το log in και τον προγραμματισμό κάθε μηχανήματος ξεχωριστά.

Γραφικό περιβάλλον Σε επίπεδο Controller προσφέρεται γραφικό περιβάλλον (GUI) το οποίο προσθέτει σε λειτουργικότητα και αμεσότητα και κάνει πιο εύκολη αλληλεπίδραση με τις λειτουργίες του δικτύου. Έτσι ελαττώνεται η χρήση CLI (διασύνδεση γραμμής εντολών) που απαιτεί εξειδίκευση και γνώσεις εντολών (κάποιες φορές διαφορετικές ανά OS κατασκευαστή) χωρίς μάλιστα επιβάρυνση με κίνηση στο δίκτυο, καθώς το GUI είναι λειτουργία που τρέχει αποκλειστικά στον Controller.

Τοπολογία Μέσω του Controller μπορούμε να έχουμε ολοκληρωμένη εικόνα του δικτύου με απεικόνιση της τοπολογίας των δικτυακών στοιχείων σε γραφικό περιβάλλον και δυνατότητα αλληλεπίδρασης με την παραγόμενη απεικόνιση της τοπολογίας για την λήψη περισσότερων πληροφοριών.

Ασφάλεια Μέσω από το SDN δίνεται η δυνατότητα για τοποθέτηση Firewalls λογισμικού στα εξωτερικά σημεία του δικτύου καθώς και δρομολόγησης της κίνησης ευαίσθητων δεδομένων από ασφαλείς κόμβους αντί σημείων που θεωρούνται ευάλωτα. Μπορούν να επιβληθούν εύκολα κανόνες σχετικά με τα πακέτα που κινούνται στο δίκτυο και να αποκλειστούν αυτά που θεωρούνται ύποπτα.

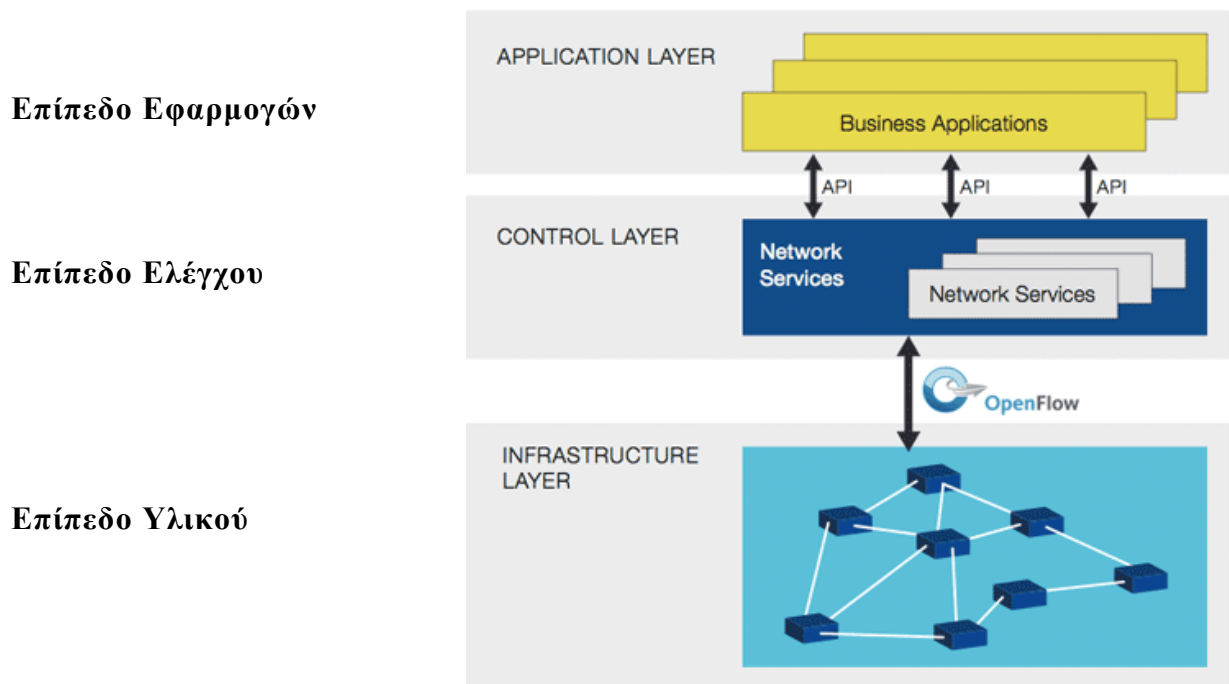
Έλεγχος Η κεντροποιημένη δομή με τον Controller στην κορυφή επιτρέπει ολική θέαση του δικτύου και πλήρη έλεγχο στις λειτουργίες κάθε δικτυακού στοιχείου καθώς και της δικτυακής κίνησης. Οι αλλαγές στις ροές του δικτύου γίνεται άμεσα μέσω του

Controller ο οποίος συντονίζει τα Switches ώστε να επιβληθούν οι κανόνες που θέτονται στο δίκτυο.

Οικονομία Η SDN αρχιτεκτονική επιτρέπει πολύ καλύτερη αξιοποίηση του εξοπλισμού καθώς και υψηλές διαθεσιμότητες στα σε όλο το εύρος του δικτύου. Παρέχεται η δυνατότητα για στοχευμένη χρήση των μηχανημάτων καθώς μέσω των SDN Applications γίνεται εύκολη η εγκατάσταση και λειτουργία δυναμικών Firewalls, FTP Servers και άλλων υπηρεσιών χωρίς να δεσμεύονται οι πόροι μιας ολοκληρωμένης εγκατάστασης κάποιας υπηρεσίας. Επίσης υφίσταται το πλεονέκτημα της αποδέσμευσης από κάποιον συγκεκριμένο προμηθευτή δικτυακού εξοπλισμού (vendor agnostic hardware) καθώς το SDN δίνει το περιθώριο συνύπαρξης μηχανημάτων πολλαπλών εταιριών χωρίς ασυμβατότητες.

2.4 SDN Επίπεδα

Η υλοποίηση της SDN αρχιτεκτονικής παράγει έναν διαχωρισμό 3 επιπέδων στο δίκτυο



Εικόνα 1 Τα τρία επίπεδα λειτουργίας της SDN λειτουργίας

Ο διαχωρισμός των επιπέδων αυτών αποσκοπεί στην αντικατάσταση των λειτουργιών του επιπέδου ελέγχου με έναν ελεγκτή (Controller) ο οποίος θα ελέγχει πολλαπλά δικτυακά μηχανήματα και θα είναι προσβάσιμος μέσα από ένα γραφικό περιβάλλον καθώς και θα έχει δυνατότητες προσθήκης λειτουργιών μέσα από εφαρμογές με την χρήση APIs.

Ερευνώντας τον τρόπο λειτουργίας του SDN θα γίνουν εμφανή τα πλεονεκτήματα της αρχιτεκτονικής αυτής έναντι του συμβατικού τρόπου λειτουργίας των δικτύων. Για να γίνει πλήρως κατανοητό τι ακριβώς είναι το Software Defined Networking και ποιοι οι στόχοι που προσπαθεί να πετύχει θα πρέπει πρώτα να υπάρξει μια πιο συγκεκριμένη ιδέα για το πως λειτουργούν τα δίκτυα και κυρίως τα Switches που είναι και το βασικό πεδίο εφαρμογής του SDN.

2.5 Προβληματισμοί/Ανησυχίες

Η κάθε εξέλιξη στον χώρο της τεχνολογίας, της πληροφορικής και κατά προέκτασης των δικτύων εκτός από τα πλεονεκτήματα και τις διευκολύνσεις που φέρει μαζί της δημιουργεί επίσης και κάποιες ανησυχίες για τις αλλαγές που θα επιφέρει στον τομέα της. Οι προβληματισμοί που παράγονται είναι επιθυμητοί καθώς δημιουργούν ρεαλιστικές προσδοκίες και βοηθούν στην ωρίμανση των τεχνολογιών.

2.5.1 Ασφάλεια

Καθώς πρόκειται για μια καινούργια αρχιτεκτονική, κάθε πτυχή της SDN υλοποίησης είναι και μια πρόκληση, με την ασφάλεια να είναι από τα πρώτα ζητήματα που χρίζουν προσοχής λόγω της σημαντικότητας της. Καθώς πρόκειται για δίκτυα μεγάλων εταιριών τα δεδομένα που ανταλλάσσονται μεταξύ των Switches πολλές φορές έχουν ευαίσθητο περιεχόμενο του οποίου η ασφάλεια πρέπει να εξασφαλιστεί στον μέγιστο δυνατό βαθμό.

Το SDN εκτιμάται πως θα είναι ένα ελκυστικό πεδίο για κακόβουλους χρήστες και βιομηχανικούς κατασκόπους καθώς λόγω της κεντρικοποιημένης δομής του, αν υπάρξει παραβίαση του Controller δίνονται στον hacker μια πληθώρα επιβλαβών δράσεων που μπορεί να αναλάβει, όπως επιθέσεις τύπου man in the middle, εισαγωγή malware, monitor trafficiking, αλλαγή στα flow tables, έλεγχος της κίνησης και άλλα. (Hinden. R, 2014)

Το κύριο ζήτημα που ανακύπτει είναι με ποιον μηχανισμό θα γίνεται αντιληπτή η παραβίαση του Controller από τους διαχειριστές του συστήματος, καθώς δεν υπάρχει πλήρης και διαφανής μηχανισμός ώστε να γίνεται εξακρίβωση των δράσεων του Controller στο δίκτυο.

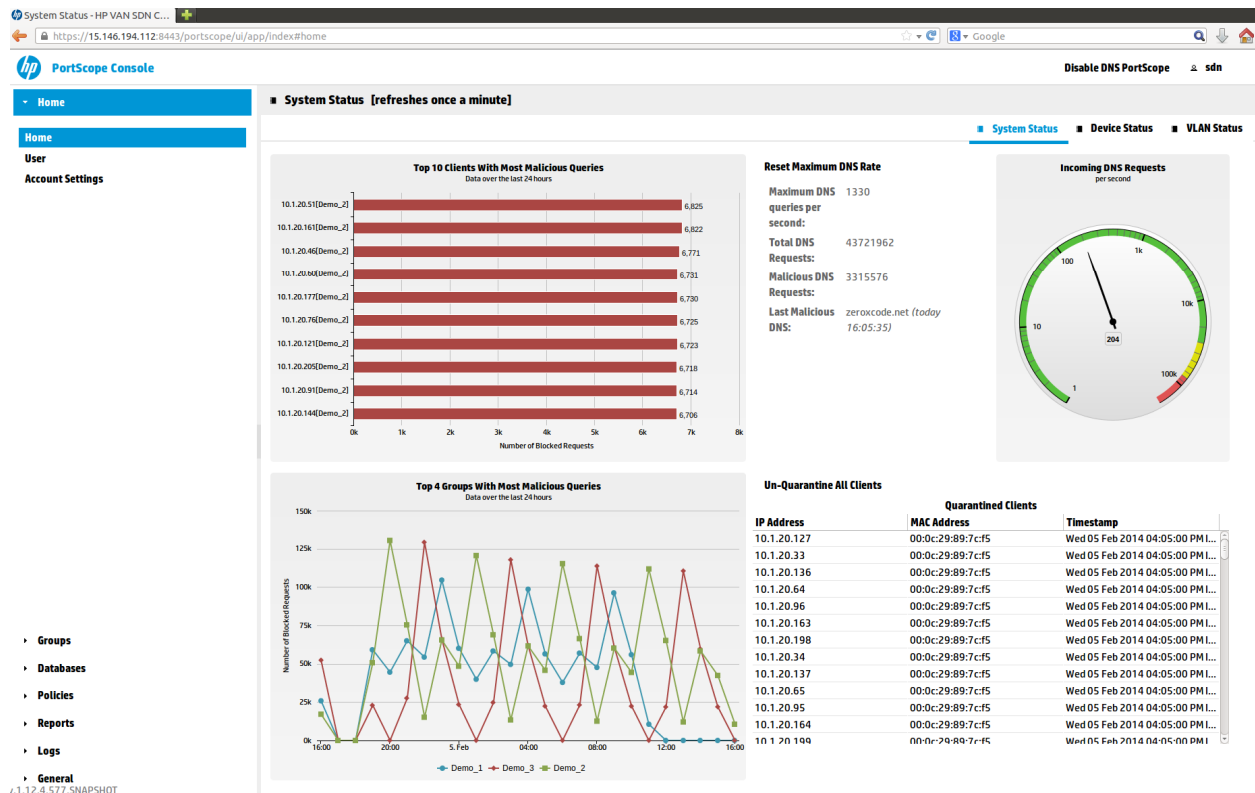
Οι ανησυχίες δεν σταματούν στους κακόβουλους χρήστες αλλά επεκτείνονται και στις Υπηρεσίες Συλλογής Πληροφοριών (πχ NSA) που εκμεταλλευόμενες διάφορα άγνωστα κενά (loopholes) στην ασφάλεια των Controllers θα μπορούν αν έχουν εξαιρετικά εύκολη πρόσβαση στην ροή της πληροφορίας σε ένα δίκτυο, κάνοντας το έργο της κατασκοπείας του εξαιρετικά εύκολο και απλούστερο.

Έως τώρα δεν έχει γίνει γνωστή κάποια μέθοδος παραβίασης του ίδιου του Controller παρά μόνο τρόποι μη πιστοποιημένης παρέμβασης Flow Tables και παραβίαση SDN πολιτικών όσον αφορά κάποιες απαγορεύσεις στο δίκτυο, συγκεκριμένα στην πρώτη

έκδοση του OpenDaylight Controller. Οι αδυναμίες αυτές έχουν διορθωθεί στις καινούργιες εκδόσεις

Η μεγαλύτερη αλλαγή στην ασφάλεια είναι η αλλαγή νοοτροπίας, όπου οι πολιτικές ασφαλείας δεν επιβάλλονται πλέον μέσα από φυσικές συσκευές αλλά σε επίπεδο λογισμικού, κάτι που απαιτεί πλήρη εμπιστοσύνη στον κώδικα αυτού του λογισμικού ώστε να μην υπάρξουν δυσάρεστες συνέπειες. Απαραίτητο κρίνεται επίσης να υπάρχουν ανεξάρτητοι μηχανισμοί που θα πιστοποιούν την σωστή λειτουργία του Controller καθώς και θα μπορούν να παρέμβουν σε περίπτωση που ο controller παραβιαστεί από κακόβουλους χρήστες.

Οι ανησυχίες αυτές ωστόσο είναι σχετικά εύκολο να ξεπεραστούν μέσα από διάφορους τρόπους, όπως με την χρήση εφαρμογής ασφαλείας που θα επιτηρεί το δίκτυο, την λειτουργία του Controller και την σωστή εφαρμογή των SDN πολιτικών στο δίκτυο. Ολοκληρωμένες εφαρμογές τέτοιου είδους ήδη διατίθενται από εταιρίες που αναπτύσσουν Software Defined Networking πλατφόρμες-οικοσυστήματα.



Εικόνα 2 Εφαρμογή ασφάλειας της HP για SDN δίκτυα

Για να ενισχυθεί το επίπεδο ασφάλειας οι εφαρμογές ασφαλείας μπορούν να παρέχουν

σειρά λειτουργιών και δυνατοτήτων όπως αντιμετώπιση παραβιασμένων ή μολυσμένων από ιούς μηχανημάτων, αντιμετώπιση DdoS attacks κ.α.

2.5.2 Προσωπικό

Αλλαγές αναμένεται να συμβούν και σε επίπεδο προσωπικού καθώς σε ένα SDN περιβάλλον αλλάζει ο παραδοσιακός ρόλος των μηχανικών/τεχνικών δικτύων που θα απαιτηθεί εκτός της δικτυακής του εξειδίκευσης να έχουν ένα εύρος προγραμματιστικών γνώσεων. Επίσης θα χρειαστεί μεγαλύτερη συνεργασία μεταξύ των ομάδων που διαχειρίζονται το δίκτυο, των ομάδων που αναπτύσσουν και λειτουργούν τις εταιρικές εφαρμογές καθώς και της ομάδας που είναι υπεύθυνη για την ασφάλεια του δικτύου, όταν αυτή υπάρχει ως διακριτή ομάδα από αυτήν των δικτύων. Η SDN αρχιτεκτονική δημιουργεί ανησυχία όσον αφορά την μακροπρόθεσμες αλλαγές που θα φέρει στον τομέα απασχόλησης των επαγγελματιών στο πεδίο του Information Technology και των δικτύων καθώς διαφαίνεται στον ορίζοντα ο περιορισμός των αρμοδιοτήτων αυτών των τομέων και άρα την λιγότερη απασχόληση προσωπικού σε αυτά. Απάντηση σε αυτό είναι η επιμόρφωση του προσωπικού στα καινούργια δεδομένα, καθώς όπως έχει δείξει η εμπειρία, η πορεία της τεχνολογίας είναι συνεχής και εξελικτική και απαιτεί την ανάλογη ευελιξία και πρόοδο από τους επαγγελματίες του χώρου.

3. Υπάρχουσες SDN υλοποιήσεις

3.1 Εφαρμοσμένη αρχιτεκτονική

Υλοποιήσεις στον τομέα του Software Defined Networking υπάρχουν ήδη από το 2010 και ως τεχνολογία έχει εφαρμοστεί από κολοσσούς στον χώρο της πληροφορικής και του διαδικτύου για την αποτελεσματικότερη διαχείριση και την επέκταση των δικτύων τους. Επειδή πρόκειται για μια αρχιτεκτονική που απευθύνεται κυρίως σε δίκτυα μεγάλου και μεσαίου μέγεθος προς το παρόν έχει υλοποιηθεί σε μεγάλες εταιρίες, καθώς απαιτείται (όπως κάθε νέα τεχνολογία) ένα μεγάλο οικονομικό κεφάλαιο και άτομα με τεχνογνωσία για να εφαρμοστεί σε πραγματικό περιβάλλον.

Ανάμεσα στις εταιρίες που έχουν υλοποιήσει/συμβάλλει στο Software Defined Networking εντοπίζουμε τις Google, Microsoft, NSA, Facebook, VMware, Deutsche Bank και άλλους.

Παρατίθενται κάποιες υλοποιήσεις αναλυτικότερα.

3.1.1 Google

Ο γίγαντας του διαδικτύου βρισκόταν από νωρίς στην εξέλιξη του Software Defined Networking και ήδη από 2012 η google είχε ολοκληρωμένες υλοποιήσεις στο που επέτρεπαν να έχει πολύ υψηλά ποσοστά αξιοποίησης του δικτύου της. Η εμπλοκή της στην ανάπτυξη του SDN τροφοδοτήθηκε εκτός από τα υπόλοιπα οφέλη και από οικονομικά κριτήρια καθώς στα δίκτυα δεν ισχύει ότι και στην επεξεργαστική ισχύ και μεγέθη αποθήκευσης, όπου η αγορά μεγάλων ποσοτήτων μειώνει την τιμή. Τα κόστος στον εξοπλισμό δικτύου παραμένει σταθερό και έτσι οι εταιρίες με μεγάλα και επεκτεινόμενα data centers αναζητούν πρωτοποριακές λύσεις για να χαμηλώσουν το κόστος στην ανάπτυξη τους. (Salisbury, 2013)

3.1.2 NSA

Η μεγαλύτερη κρατική υπηρεσία ελέγχου του διαδικτύου αξιοποίησε την SDN αρχιτεκτονική για την αποδοτικότερη διαχείριση του δικτύου με βάση τον RYU Controller. Έχοντας να αντιμετωπίσει τις ίδιες προκλήσεις που έχει κάθε μεγάλη εταιρία που διαχειρίζεται και αποθηκεύει μεγάλο όγκο δεδομένων η NSA βρήκε στο OpenFlow SDN την λύση που θα της επέτρεπε να κρατάει πιο εύκολα αρχείο των εκατομμυρίων κλήσεων, emails, μηνυμάτων και λοιπών δεδομένων που υποκλέπεται κάθε μέρα. (Duffy, 2015)

3.1.3 FACEBOOK

Όταν μιλάμε για την μεγαλύτερη εταιρία στον χώρο της κοινωνικής δικτύωσης με 1.7 δισεκατομμύρια χρήστες και με τα καθημερινά uploads σε βίντεο να ξεπερνούν ακόμα και αυτά του YouTube είναι πολύ εύκολα αντιληπτό γιατί το FB χρειάζεται μεγάλη ευελιξία στο δίκτυο του. Όμως εκτός της εφαρμογής SDN αρχιτεκτονικής το facebook πήγε ένα βήμα παρακάτω προχωρώντας σε hardware υλοποιήσεις open hardware, αρχικά το wedge Switch και έπειτα το 6-pack Switch που είχε το προηγούμενο ως βάση. Αυτά τα Switch τρέχουν ένα υβριδικό μοντέλο SDN όπου κάθε μηχανήμα έχει ένα τοπικό επίπεδο ελέγχου σε έναν microserver που βρίσκεται σε επικοινωνία με τον κεντρικό Controller, παρέχοντας ευκολία στην διαχείριση και λειτουργία του δικτύου όπως επίσης ασφάλεια και υψηλές διαθεσιμότητες (code.facebook.com).



Εικόνα 3 Wedge και 6pack Switches του Facebook

3.1.4 MICROSOFT

Ως εταιρία είναι γνωστό πως ασχολείται περισσότερο με λογισμικό παρά με big data, όμως πίσω από τα προγράμματα και τα λειτουργικά που παρέχει υπάρχει μια μεγάλη cloud υποδομή (Office 365, Bing, Xbox, OneDrive κλπ) που συνεχώς επεκτείνεται. Για να ανταπεξέλθει στις προκλήσεις των μεγεθών που καλείται να διαχειριστεί η Microsoft ανέπτυξε την Azure, πλατφόρμα cloud που σε επίπεδο δικτύου κάνει χρήση Software Defined αρχιτεκτονικής. Παράλληλα συνεργάζεται με την Cisco για την δημιουργία εξειδικευμένων SDN APIs υλοποιήσεων. (Hubbard, 2015)

3.2 Controllers – Υλοποιήσεις

Ως τώρα υπάρχουν πολλές υλοποιήσεις στο επίπεδο του Controller από διάφορους οργανισμούς και εταιρίες, συγκεκριμένα υπάρχουν οι εξής:

- NOX, ο πρώτος Controller, αναπτύχθηκε από την Nicira Networks, θυγατρική της VMWare και ήταν ανοιχτού κώδικα και βάση για αρκετούς μετέπειτα Controllers
- ONIX, συνέχεια του NOX με την συνεργασία Nicira, NTT και Google, αρχικά είχε ανακοινωθεί πως ο κώδικα θα άνοιγε, αλλά αποφασίστηκε από τις εταιρίες να μείνει κλειστού κώδικα, φημολογείται πως είναι η βάση του Google WAN Controller
- Beacon, αναπτύχθηκε από το πανεπιστήμιο του Stanford στις αρχές του 2010
- NSX, ο Controller που αναπτύχθηκε από την Nicira/Vmware και συνέχεια του ONIX

- Google WAN Controller, αν και η google είχε εξαρχής εμπλοκή με τον SDN δεν έχει αποκαλύψει λεπτομέρειες για τον Controller της, εξάλλου είναι ένα εσωτερικό project της για να εξυπηρετήσει τις δικές τις ανάγκες
- OpenDaylight, ο κυρίαρχος αυτήν την στιγμή Open Source Controller και βάση για μια σειρά Controllers ακόμα
- HP VAN SDN Controller, εμπορικός Controller βασισμένος στον OpenDaylight με πολλές δυνατότητες και το πρώτο λειτουργικό App Store
- Contrail, Controller-SDN πλατφόρμα της Juniper
- Cisco Open SDN Controller, επίσης εμπορικός, βασισμένος στον OpenDaylight, προσαρμοσμένος για πιο αποτελεσματική λειτουργία μέσα στο οικοσύστημα συσκευών της Cisco
- Floodlight, συνέχεια του Beacon
- POX Controller
- Cognos Controller
- RYI είναι γνωστός κυρίως λόγω της χρήσης του από την NSA
- Ericsson SDN Controller

Βλέποντας τον πλουραλισμό που έχει αναπτυχθεί στους Controllers μπορούμε με σιγουριά να εξάγουμε συμπεράσματα για το μέλλον του SDN το οποίο ως τεχνολογία αναμένεται να έχει αυξητικές τάσεις και με όλες τις μεγάλες εταιρίες στον χώρο των δικτύων να έχουν επενδύσει μεγάλα ποσά σε ανάπτυξη Controllers. (SDXcentral)

Όμως δεν είναι μόνο οι εταιρίες που έχουν “κλείσει” την συγκεκριμένη τεχνολογία καθώς υπάρχουν πολλές επιλογές σε Controllers ανοιχτού κώδικα για χρήστες που επιθυμούν να έχουν πλήρη εικόνα του κώδικα ή δεν θέλουν να επωμιστούν τα έξοδα της αγοράς ενός Controller μιας εταιρίας δικτύων. Οι open source Controllers δεν πρέπει να απαξιωθούν ούτε στο ελάχιστο βάση της της λανθασμένης αντίληψης που βλέπει τις υλοποιήσεις ανοιχτού κώδικα ως κατώτερες των εταιρικών και πρέπει να ληφθεί υπόψιν πως οι υλοποιήσεις των μεγάλων εταιριών έχουν ως βάση τους Beacon και έπειτα του νεότερου Opendaylight, αμφότεροι open source Controllers.

OpenDaylight Founding Members

Platinum Members



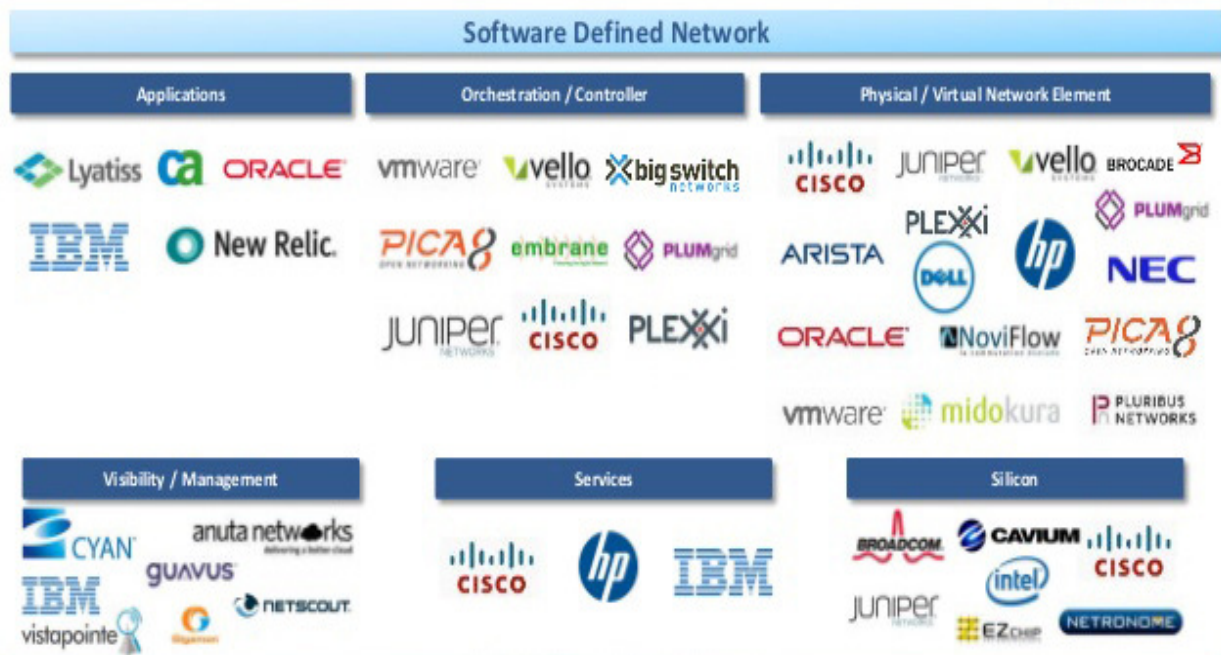
Εικόνα 4 Εταιρίες με συμμετοχή στην ανάπτυξη του OpenDaylight

Προφανώς οι γίγαντες στον χώρο των δικτύων προσπαθούν να προωθήσουν την δική τους τεχνολογία, σε λογισμικό και hardware και για αυτό προσφέρουν πλατφόρμες που δεν περιορίζονται στην προσφορά ενός Controller αλλά ενός ολόκληρου οικοσυστήματος γύρω από την SDN τεχνολογία, με παράλληλη υλοποίηση πολλαπλών εικονικών λειτουργιών του δικτύου όπως Dynamic Cloud, SDN Applications, Switch Virtualisation με Open vSwitch και άλλα. Αυτό φαίνεται να αλλοιώνει τον ανοιχτό χαρακτήρα τον οποίον είχε αρχικά η ιδέα του SDN, όμως πρέπει να ληφθεί υπόψιν πως οι πανεπιστημιακές ιδέες είναι ένα θέμα ξεχωριστό από τις απαιτήσεις της αγοράς που διαμορφώνουν εν τέλει την νέα πραγματικότητα του πεδίου μέσω των χρημάτων που αφιερώνουν στην έρευνα και την κατεύθυνση που της δίνεται. Ενώ λοιπόν το Software Defined Networking ξεκίνησε ως μια ιδέα ανοιχτών συμβατοτήτων μεταξύ όλων των επιπέδων hardware και software φαίνεται να δημιουργούνται διάφοροι εταιρικοί πόλοι που γύρω τους συγκεντρώνουν υλοποιήσεις λογισμικού και υλικού. Η συνέχεια και η πορεία που θα ακολουθήσει η SDN αρχιτεκτονική θα φανεί σε βάθος χρόνου, καθώς είναι δύσκολο να προβλεφθούν οι σταθεροί και αστάθμητοι παράγοντες που την επηρεάζουν..

3.3 Δραστηριότητα στον χώρο του SDN

Πέραν των μεγάλων εταιριών που αναφέρθηκαν παραπάνω οι οποίες πρωτοπόρησαν στο SDN καθώς ως γίγαντες της τεχνολογικής βιομηχανίας είχαν την οικονομική δυνατότητα να πειραματιστούν με τολμηρές λύσεις στα δικτυακά τους προβλήματα γίνεται αντιληπτό πως από το 2012 και έπειτα υπάρχει ένα πολύ έντονο ενδιαφέρον στις εταιρίες του χώρου των δικτύων και όχι μόνο, για το Software Defined Networking. Υπάρχει η προσδοκία πως η αρχιτεκτονική αυτή θα είναι σύντομα διαθέσιμη και ανταγωνιστική για εταιρίες με δίκτυα μεσαία κλίμακας οπότε και θα δημιουργηθεί η προοπτική μιας νέας αγοράς προϊόντων και υπηρεσιών. Το εταιρικό ενδιαφέρον για το SDN τροφοδοτείται από το μικρότερο κόστος που θα απαιτεί η λειτουργία των δικτύων μακροπρόθεσμα, απλοποίηση του τρόπου λειτουργίας των δικτύων και άρα ευκολότερο troubleshooting όπως και της αυξημένης ευελιξίας που προσφέρεται, καθώς επιτρέπει σε εταιρίες να έχουν μικρότερους κύκλους ανάπτυξης από ότι τους πενταετείς που ισχύουν τώρα, και αποδεσμευόμενες από τις σφιχτές συμβατότητες θα μπορεί ο κύκλος αυτός να μειώνεται στα δυο με τρία χρόνια. Κάτι που επιτρέπει πολύ πιο γρήγορη εκτέλεση αλλαγών και προσθήκης νέων υπηρεσιών συγκριτικά με τον παραδοσιακό τρόπο λειτουργίας. Αυτό έχουν αντιληφθεί οι μεγάλες εταιρίες στον χώρο των δικτύων όπως Cisco, Juniper, HP και άλλες που δραστηριοποιούνται πλέον δυναμικά στην SDN αρχιτεκτονική προσφέροντας προτάσεις σε επίπεδο υλικού και λογισμικού.

Αναπτύσσεται λοιπόν ήδη ένα ολόκληρο οικοσύστημα γύρω από το SDN, με εταιρίες να προσφέρουν υλοποιήσεις σε επίπεδο υλικού, ελέγχου και εφαρμογών καθώς και σύνδεση της αρχιτεκτονικής με άλλες που κινούνται σε ίδια ή παρόμοια νοοτροπία, όπως το Software Defined Storage, OpenStack και άλλες. Αυτό χρεώνεται στα κατορθώματα του SDN καθώς φαίνεται πως πολύ γρήγορα καθιερώνεται ως ένα σημαντικό κεφάλαιο στο μέλλον των δικτύων δημιουργώντας κοινό έδαφος λειτουργίας με άλλες τεχνολογίες.



Εικόνα 5 Οι εταιρικές συμμετοχές ανά τομέα του SDN οικοσυστήματος

3.4 White box Switches

White box Switches ή λευκά/κενά Switches κουτιά, αναφέρεται στην ικανότητα των Switch να είναι διαθέσιμα χωρίς Data Plane προ εγκατεστημένο και άρα να είναι πιο ανταγωνιστικά όσον αφορά την τιμή και τις δυνατότητες παραμετροποίησης.

Τέτοιου είδους Switches μπορούν να αποκτηθούν είτε χωρίς κανενός είδους OS (Λειτουργικό Σύστημα) το οποίο μπορεί να επιλεγεί και να εγκατασταθεί σε τρίτο χρόνο ή έρχονται με προ εγκατεστημένο κάποιο μη εμπορικό λειτουργικό σύστημα, συνήθως Linux based λόγω του ανοιχτού τους χαρακτήρα αλλά και του πλήθους των εργαλείων που αυτό διαθέτει για την διαχείριση συσκευών δικτύων.

Ένα τέτοιο project για παράδειγμα είναι το OpenSwitch, Linux διανομή που εξελίσσεται μέσα από κοινότητες στον χώρο του λογισμικού, είναι μέρος του Linux Foundation και προσφέρει το σύνολο των λειτουργιών που προσφέρουν και τα εταιρικά Λειτουργικά Συστήματα.

Το πλεονέκτημα των White Boxes είναι η κατά πολύ χαμηλότερη τιμή απόκτησης τους καθώς και η ελευθερία που προσφέρουν στην παραμετροποίηση κάθε πτυχής τους.

Η εξέλιξη των White Boxes δεν συνδέεται άμεσα με το Software Defined Networking όμως προσφέρει μια ακόμα μέθοδο υλοποίησης του, δημιουργώντας ένα πλεονέκτημα κόστους καθώς είναι φτιαγμένα από μικρότερους κατασκευαστές και άρα φθηνότερα και επίσης η ύπαρξη εμπορικού/εταιρικού Λειτουργικού Συστήματος ή η ύπαρξη ενός ανοιχτού κώδικα OS αφαιρεί άλλη μια δικαιολογία που παρουσιάζουν οι μεγάλες εταιρίες για τον ακριβό εξοπλισμό τους.

Τα White Boxes λοιπόν μπορούν να λειτουργήσουν πολύ αποδοτικά μέσα σε SDN περιβάλλον όπως τα Switches μεγάλων εταιριών με εγκατάσταση του OpenFlow πάνω σε Linuxοειδή Λειτουργικά Συστήματα για Switches σε ανταγωνιστική τιμή και με περισσότερες δυνατότητες παραμετροποιήσεων.

3.5 Startups

Παράλληλα με την εταιρική δραστηριότητα παρατηρείται μια μεγάλη ενεργητικότητα όσον αφορά Startup εταιρίες που έχουν ως αντικείμενο πτυχές της υλοποίησης του Software Defined Networking.

Οι Startups είχαν εξ αρχής παρουσιάσει ενδιαφέροντες ιδέες για την ανάπτυξη του SDN, γειωμένες όμως στην πραγματικότητα των περιορισμών της αγοράς και της ζήτησης μπόρεσαν να συνδεθούν αποτελεσματικά με την υλοποίηση της συγκεκριμένης αρχιτεκτονικής και να παρουσιάσουν έως και εμπορικές, κυρίως White-box hardware υλοποιήσεις, όπως το 48 θυρών Switch της Pica8.

Κάποιες από αυτές τις αρχικές Startups έχουν αγοραστεί από μεγάλες εταιρίες για επίσης μεγάλα χρηματικά ποσά, χαρακτηριστικά παραδείγματα αποτελεί η Nicira Networks, πρωτοπόρο εταιρία στον τομέα του SDN που αγοράστηκε από την VMware για 1.25 δισεκατομμύρια δολάρια και η Contrail Systems που αγοράστηκε από την Juniper καταβάλλοντας το ποσό των 176 εκατομμυρίων δολαρίων.

Άλλες startups επιχορηγούνται από μεγάλες εταιρίες, συνδεόμενες έτσι μαζί τους αλληλοτροφοδοτούνται ιδέες και γνώσεις λειτουργώντας βάσει αμοιβαίου συμφέροντος.

3.6 SDN συμβατό Hardware-Switches

Οι συμβατότητες όσον αφορά το Hardware είναι αρκετά ανεπτυγμένες και αφορούν ως τώρα γύρω γύρω στις τριάντα σειρές κυρίως καινούργιων αλλά και παλιών μηχανημάτων που διαθέτουν OpenFlow/SDN δυνατότητες. Συγκεκριμένα πρόκειται για τα εξής Switches:

A10 Networks AX Series

ADVA Optical Networking FSP 150

ADVA Optical Networking FSP 3000

Brocade CER 2000 Series

Brocade CES 2000 Series

Centec Networks - V330 OpenFlow Switch Reference Design

Cisco Catalyst 6k

Cisco Nexus 9000 Series Switches

Extreme Networks BlackDiamond 8000 Series

Extreme Networks BlackDiamond

Extreme Networks Summit X670

HP 2920 Switch Series

HP 3500 Switch Series

HP 3800 Switch Series

HP 5400 Switch Series

IBM RackSwitch G8264T

Infinera - DTN-X

Intune Networks iVX8000

NEC ProgrammableFlow 1/10GbE Hybrid OpenFlow Switch PF5240

NEC ProgrammableFlow 10GbE OpenFlow Switch PF5820

NEC ProgrammableFlow Network Controller PF 6800

NoviFlow NoviKit 100

Pica8 Open Switches

Transmode TM-Series



**Εικόνα 6 White-Box Switch της Pica8 και η Nexus 9000
σειρά της Cisco**

Όπως φαίνεται από την λίστα των Switches υπάρχουν επιλογές που ποικίλουν και καλύπτουν όλο το φάσμα των εταιριών δικτύων, με προτάσεις σε όλα τα επίπεδα, με White Boxes και υβριδικά Switches, από μεμονωμένα Switches έως σασί πολλαπλών μηχανημάτων, από προτάσεις μικρών και νέων εταιριών (Cumulus, Big Switch, Pica8)

έως προτάσεις μεγάλων και καθιερωμένων προμηθευτών hardware του χώρου (Cisco, Juniper).

Αυτός ο πλουραλισμός στις διαθέσιμες επιλογές και στις δυνατότητες είναι η υλική απόδειξη πως η Software Defined Networking αρχιτεκτονική οδεύει σταθερά προς καθιέρωση της στον χώρο των δικτύων όχι μόνο εντός των μεγάλων εταιρικών data centers αλλά και στο επίπεδο των επιχειρήσεων με μεσαίου μεγέθους δίκτυα.
(SDxCentral 2015)

4. SDN Layers

4.1 Επίπεδο Υλικού

4.1.1 Δίκτυο

4.1.1.1 Περιορισμοί φυσικού δικτύου

Για να δοκιμαστεί η αποτελεσματικότητα ενός εργαλείου πρέπει να υπάρχει και το κατάλληλο πεδίο πάνω στο οποίο θα αξιοποιηθούν πλήρως τις δυνατότητες του, η SDN τεχνολογία απευθύνεται στα δίκτυα και ως εκ τούτου αυτό είναι και το πεδίο δράσης και υλοποίησης της. Καθώς όμως τα δικτυακά στοιχεία (elements) είναι ένα σύνολο συσκευών άκρως συγκεκριμένου χαρακτήρα δεν είναι οικονομικά βιώσιμο για κάποιον ιδιώτη να έχει ένα δίκτυο, ακόμα και μικρών ποσοτικών και ποιοτικών χαρακτηριστικών μόνο για πειραματισμούς. Επίσης σε πανεπιστήμια και επιχειρήσεις αντίστοιχα τα μηχανήματα που διατίθενται για πειραματισμό, εκμάθηση και ανάπτυξη είναι κατά κανόνα περιορισμένα καθώς αυτά αξιοποιούνται στο πρακτικό πεδίο για την μεταφορά της κανονικής κίνησης του δικτύου. Για τους παραπάνω λόγους ή έρευνα, ο πειραματισμός καθώς και η παρούσα πτυχιακή γίνονται κυρίως μέσω προγραμμάτων προσομοίωσης όσον αφορά το επίπεδο υλικού.

4.1.1.2 Προσομοιωτές

Έτσι για να λόγους οικονομικούς αλλά και πρακτικούς αναπτύχθηκαν οι προσομοιωτές δικτύων όπως ο Packet Tracer και ο GNS3 οι οποίοι προσφέρουν μεγάλη ευκολία στην δημιουργία και διαχείριση εικονικών δικτύων και διευκολύνουν τον φοιτητή/τεχνικό που θέλει να πειραματιστεί και να μάθει χωρίς να μπει στην διαδικασία στησίματος και λειτουργίας ενός μικρού εργαστηρίου δικτύων. Ο καθένας από αυτούς τους δυο προσομοιωτές δικτύων έχουν ένα σημείο που ξεχωρίζουν, ο ένας είναι καταλληλότερος για switching και ο άλλος για routing και hub and spoke πλαίσια.

4.1.1.3 Mininet

Η SDN ως νέα τεχνολογία με ιδιαίτερες απαιτήσεις έφερε και τον δικό της προσομοιωτή δικτύων ο οποίος σχεδιάστηκε με γνώμονα την συμβατότητα με τα OpenFlow/SDN πρωτόκολλα, τον Mininet. Ο Mininet προσφέρει την δυνατότητα ανάπτυξης ενός πλήρους εικονικού δικτύου από switches και H/Y οι οποίοι τρέχουν

κανονικούς kernels και υπάρχουν ως αυτόνομες διακριτές μονάδες που μπορούν να παραμετροποιηθούν με ευκολία από το CLI. Το Mininet βασίζεται στο Open vSwitch

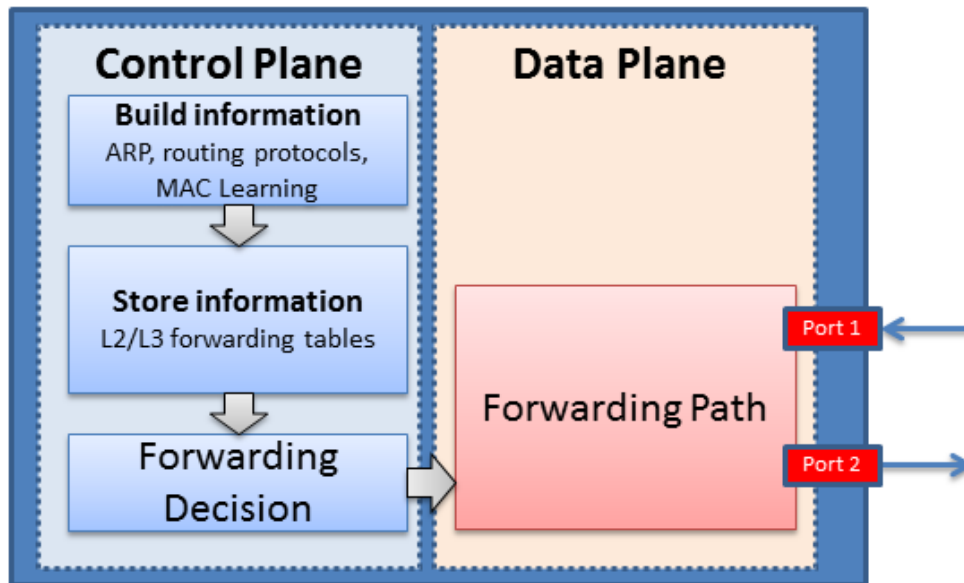
4.1.1.4 Μεταγωγέας (Switch)

Το Switch στην διάσταση που μας ενδιαφέρει ορίζεται ως μια δικτυακή συσκευή της οποίας η λειτουργία είναι η σύνδεση πολλαπλών συσκευών σε ένα δίκτυο συνήθως στο 2ο επίπεδο (data link layer) . Η σύνδεση αυτή πραγματοποιείται μέσα από μια σειρά θυρών (interfaces) με τη χρήση καλωδίων δικτύωσης από υπολογιστές ή άλλες δικτυακές συσκευές. Για να επιτευχθεί η διασυνδεσιμότητα των συσκευών το switch αντιστοιχεί την θύρα του, με την φυσική διεύθυνση (physical address) της συσκευής και συμπληρώνει τον σχετικό πίνακα προώθησης/μεταγωγής (switching table)

Τα Switches για να επιτελέσουν την εργασία τους έχουν 2 επίπεδα που συνυπάρχουν μέσα στην συσκευή:

1. Επίπεδο ελέγχου (Control Plane) – είναι το “μυαλό” της συσκευής, όπου βρίσκονται τα switching tables, και γίνονται οι βασικές διεργασίες που ευθύνονται για την ορθή λειτουργία του data plane . Το συγκεκριμένο πρωτόκολλο τρέχει τοπικά σε κάθε μηχανήμα ώστε να γνωρίζει ποια άλλα μηχανήματα έχει δίπλα του και μαζί σχηματίζουν μια στατική εικόνα του δικτύου σε ένα link state database.

2. Επίπεδο προώθησης δεδομένων (forwarding-data plane) – στην ουσία το επίπεδο υλικού (hardware), που είναι υπεύθυνο για την μεταφορά των δεδομένων μεταξύ των μηχανημάτων υπό τις εντολές του επιπέδου ελέγχου, κάνοντας χρήση των δεδομένων από τα Routing Information Databases και MAC address tables για να προωθήσει τα πακέτα.

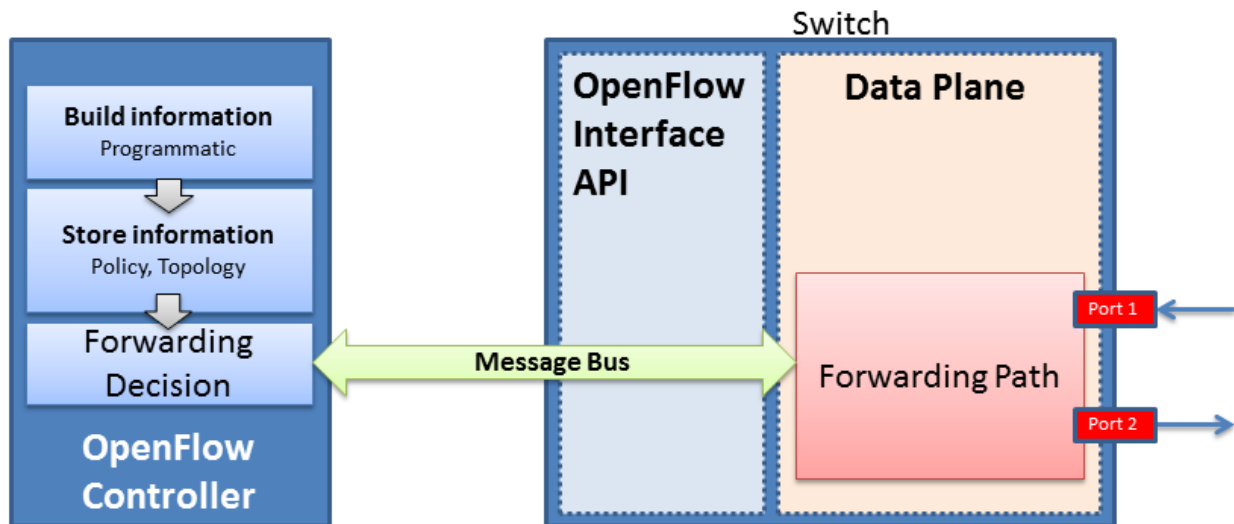


Εικόνα 7 Κανονική λειτουργία Switch

Πρόσβαση στα παραπάνω επίπεδα υπάρχει μόνο μέσω CLI (Command Line Interface) και το περιβάλλον/εντολές εξαρτώνται από τον κατασκευαστή του μηχανήματος. Για παράδειγμα σε ένα Juniper Switch χρησιμοποιούμε την ls εντολή για να δούμε τα περιεχόμενα ενώ σε ένα Cisco Switch την εντολή dir. Οι διαφορές ανά κατασκευαστή γίνονται πολύ εντονότερες όμως σε επίπεδο πρωτοκόλλων όπου υπάρχει μεγάλη πρακτική και αντιληπτική απόσταση στην εφαρμογή της ίδιας διαδικασίας. Αυτό οδηγεί σε μια ανομοιογένεια μεταξύ των μηχανημάτων διαφορετικών εταιριών που κάνουν χρήση ξεχωριστών πρωτοκόλλων και εντολών και μας δημιουργεί δυσκολίες στο να διαχειριστούμε ένα σύνολο διαφορετικών μηχανημάτων.

Επίσης οι περισσότερες εταιρίες “κλειδώνουν” το λογισμικό που τρέχει στα μηχανήματα τους ώστε να μην επιδέχεται αλλαγές και τροποποιήσεις. Αυτό από την μια είναι κατανοητό καθώς πειράζοντας τον πυρήνα το λογισμικού τους μπορούμε να τα καταστήσουμε ασταθή, από την άλλη όμως μπορεί να μας περιορίσει σε διάφορους τομείς, όπως για παράδειγμα στο να εγκαταστήσουμε ένα νέο πρωτόκολλα δρομολόγησης.

Η SDN αρχιτεκτονική ως υλοποίηση “καταργεί” το Control Plane (επίπεδο ελέγχου) της συσκευής τοπικά και αντικαθιστά την λειτουργία του με έναν ελεγκτή (Controller) ο οποίος βρίσκεται εκτός συσκευής. Ως γέφυρα μεταξύ του controller και του Data Plane εγκαθίσταται μια Διεπαφή Προγραμματισμού Επαφών (Application Programming Interface), και συγκεκριμένα στην περίπτωση μας την OpenFlow διεπαφή.



Εικόνα 8 Switch ελεγχόμενο από εξωτερικό Controller

4.1.1.5 White Boxes

White box Switches ή λευκά/κενά Switches κουτιά, αναφέρεται στην ικανότητα των Switch να είναι διαθέσιμα χωρίς Data Plane προ εγκατεστημένο και άρα να είναι πιο ανταγωνιστικά όσον αφορά την τιμή και τις δυνατότητες παραμετροποίησης.

Τέτοιου είδους Switches μπορούν να αποκτηθούν είτε χωρίς κανενός είδους OS (Λειτουργικό Σύστημα) το οποίο μπορεί να επιλεγεί και να εγκατασταθεί σε τρίτο χρόνο ή έρχονται με προ εγκατεστημένο κάποιο μη εμπορικό λειτουργικό σύστημα, συνήθως Linux based λόγω του ανοιχτού τους χαρακτήρα αλλά και του πλήθος των εργαλείων που αυτό διαθέτει για την διαχείριση συσκευών δικτύων. Ένα τέτοιο project για παράδειγμα είναι το OpenSwitch, Linux διανομή που εξελίσσεται μέσα από κοινότητες στον χώρο του λογισμικού, είναι μέρος του Linux Foundation και προσφέρει το σύνολο των λειτουργιών που προσφέρουν και τα εταιρικά Λειτουργικά Συστήματα.

Το πλεονέκτημα των White Boxes είναι η κατά πολύ χαμηλότερη τιμή απόκτησης τους καθώς και η ελευθερία που προσφέρουν στην παραμετροποίηση κάθε πτυχής τους.

Η υλοποίηση των White Boxes έχει σημαντικό αντίκτυπο και στην SDN αρχιτεκτονική, επειδή δεν υπάρχει εγκατεστημένο κανένα λειτουργικό γίνεται εγκατάσταση μόνον του

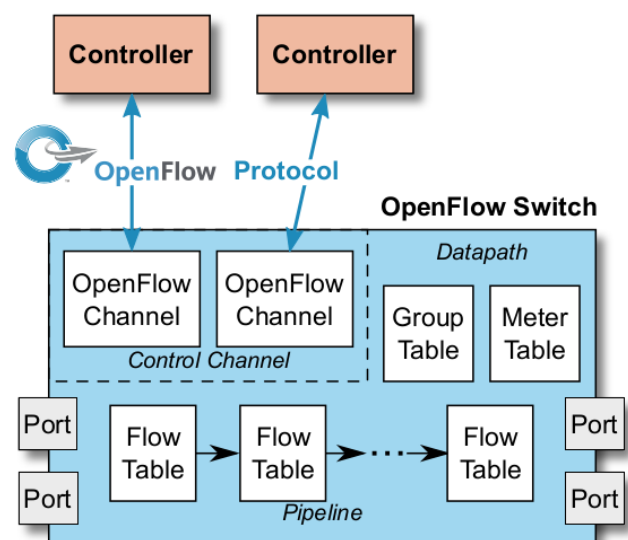
OpenFlow που επιτρέπει την πλήρη μεταφορά του Data Plane στον Controller μέσα από των οποίων εκτελούνται όλες οι λειτουργίες Switching.

Εκτός αυτού μπορεί να υπάρξει και υβριδική υλοποίηση με την εγκατάσταση κάποιου OS που να υποστηρίζει το OpenFlow πρωτόκολλο και να υπάρχει η δυνατότητα λειτουργίας του με τοπικό Data Plane χωρίς Controller σε περίπτωση που υπάρξει κάποια βλάβη σε αυτόν.

4.1.2 OpenFlow

Το OpenFlow έχει κομβικό ρόλο στην Software Defined αρχιτεκτονική καθώς είναι το πρωτόκολλο που γεφυρώνει το επίπεδο υλικού με το επίπεδο ελέγχου και είναι επί της ουσίας ο εφαρμοστής των κανόνων που θεσπίζει ο Controller. Για την εγκατάσταση του σε κάποιο Switch είναι απαραίτητο αυτό να είναι συμβατό (OpenFlow enabled) με το πρωτόκολλο,

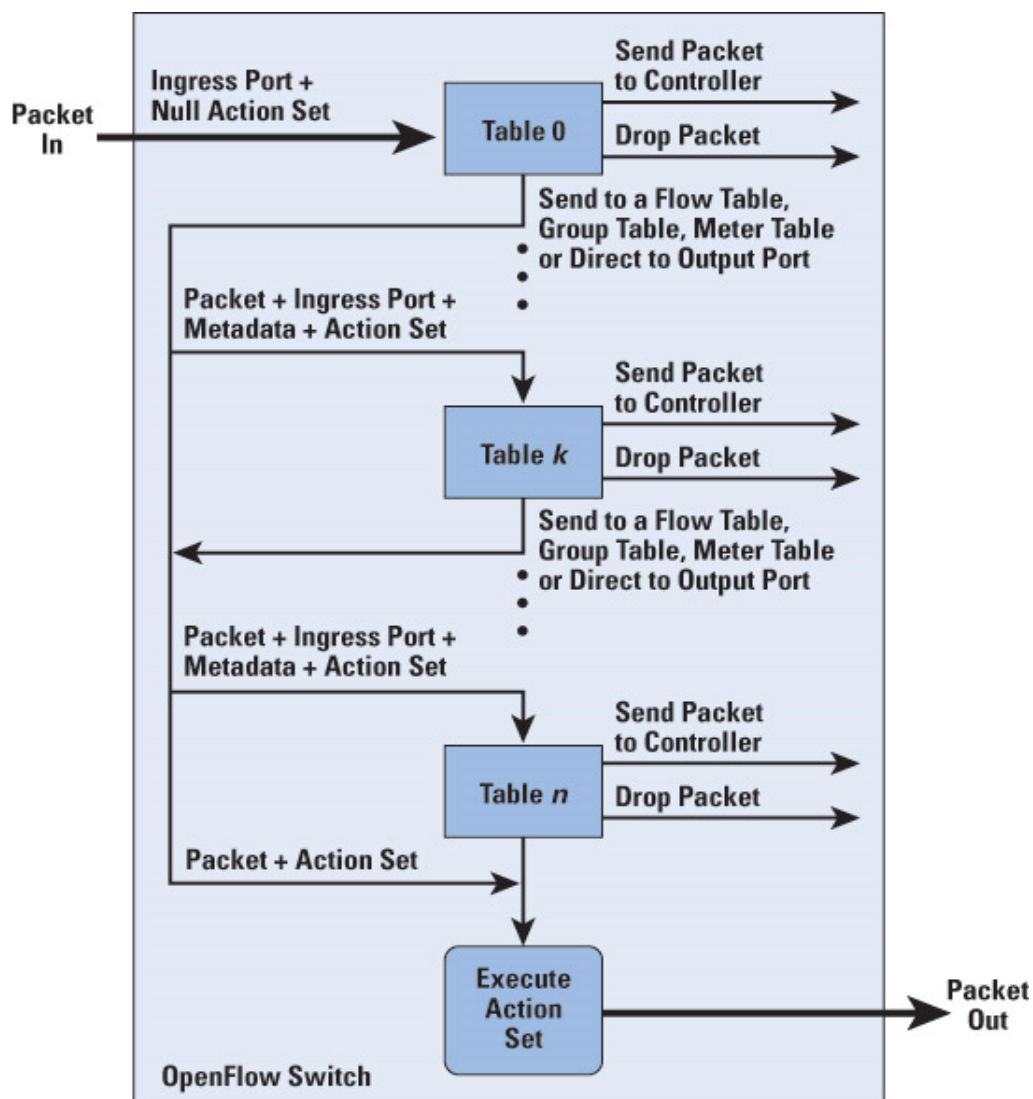
δηλαδή είτε να είναι ένα καινούργιο Switch που υποστηρίζει το OpenFlow ή να είναι κάποιο παλιότερο στο οποίο ο κατασκευαστής μέσω μιας αναβάθμισης λογισμικού έχει προσθέσει την συμβατότητα. Το OpenFlow ενεργοποιείται εύκολα από το περιβάλλον του Switch μέσω των σχετικών εντολών, αναζητεί κάποιον Controller και εγκαθιδρύει μια σύνδεση μαζί του. Η ενεργοποίηση του OpenFlow στο επίπεδο του Switch αδρανοποιεί το τοπικό Control Plane (λειτουργία που σχηματικά μεταβιβάζεται στον Controller) και το στην θέση του λειτουργεί το δικό του περιβάλλον που περιλαμβάνει ως δομικά στοιχεία τα flow tables, group tables, meter tables και control channel. (ONF, 2013)



Εικόνα 9 Η OpenFlow εγκατάσταση εσωτερικά του Switch

4.1.3 Pipeline processing

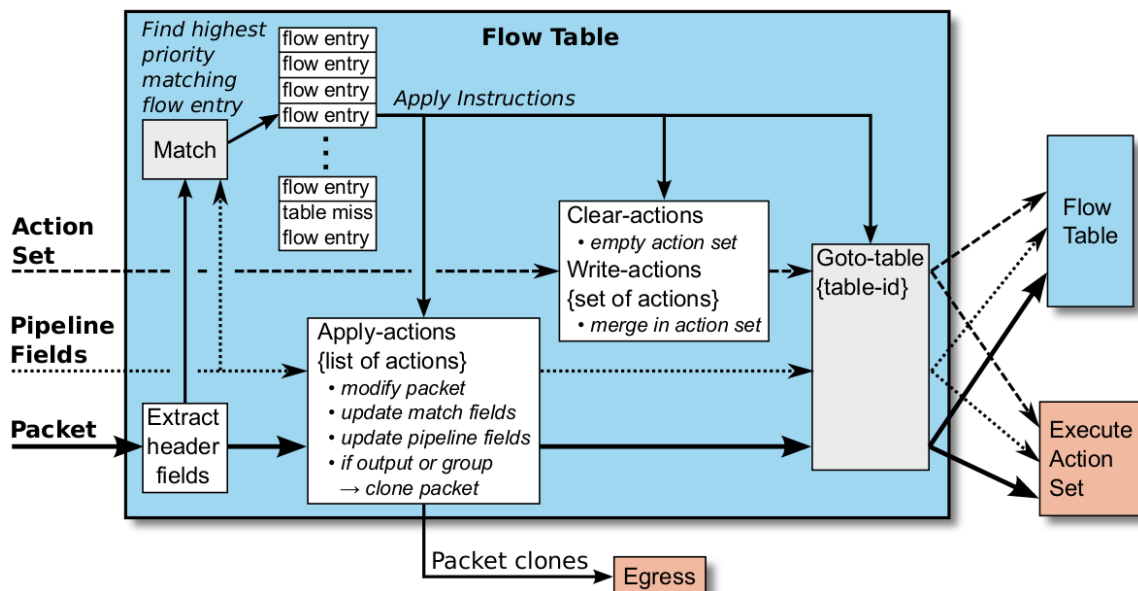
Από την στιγμή που ένα πακέτο εισέρχεται στο δίκτυο γίνεται το υποκείμενο μιας σειράς ενεργειών και ελέγχων που καθορίζουν την μετέπειτα πορεία του. Συγκεκριμένα πληροφορίες σχετικά με το πακέτο εξάγονται από το Header του, όπως η πηγή, ο προορισμό και ο τύπος του που, πληροφορίες που μορφοποιούνται ώστε να είναι πιο εύκολο να βρεθούν αντιστοιχίες μέσα από πολλαπλούς πίνακες που κάνουν ελέγχους κατηγοριοποίησης.



Εικόνα 10 Η ροή των πακέτων Pipeline Processing μέσω των Flow Tables

4.1.4 Flow Tables

Η ραχοκοκαλιά του μηχανισμού επεξεργασίας των πακέτων, οι πίνακες μέσα από τους οποίους τίθενται σε εφαρμογή οι πολιτικές που έχουν οριστεί στο δίκτυο σχετικά με την ροή των πακέτων. Οι πίνακες αυτοί έχουν μια σειρά κανόνων που κατηγοριοποιούν το είδος της κίνησης και καθώς το πακέτο “φιλτράρεται” μέσα από αυτούς τους πίνακες θα πρέπει να πληροί τους κανόνες κάποιος κατηγοριοποίησης (flow entries) ώστε να προχωρήσει. Η ικανοποίηση (ή μη) των κανόνων οδηγεί αντίστοιχα σε κάποιες δράσεις που έχουν προ-προγραμματιστεί και να δράσουν είτε μεταφέροντας το πακέτο σε άλλα Flow Tables, είτε στον Controller, είτε προωθώντας το στον προορισμό του (Packet Out) είτε όποια άλλη δράση επιλέξει ο διαχειριστής του δικτύου.

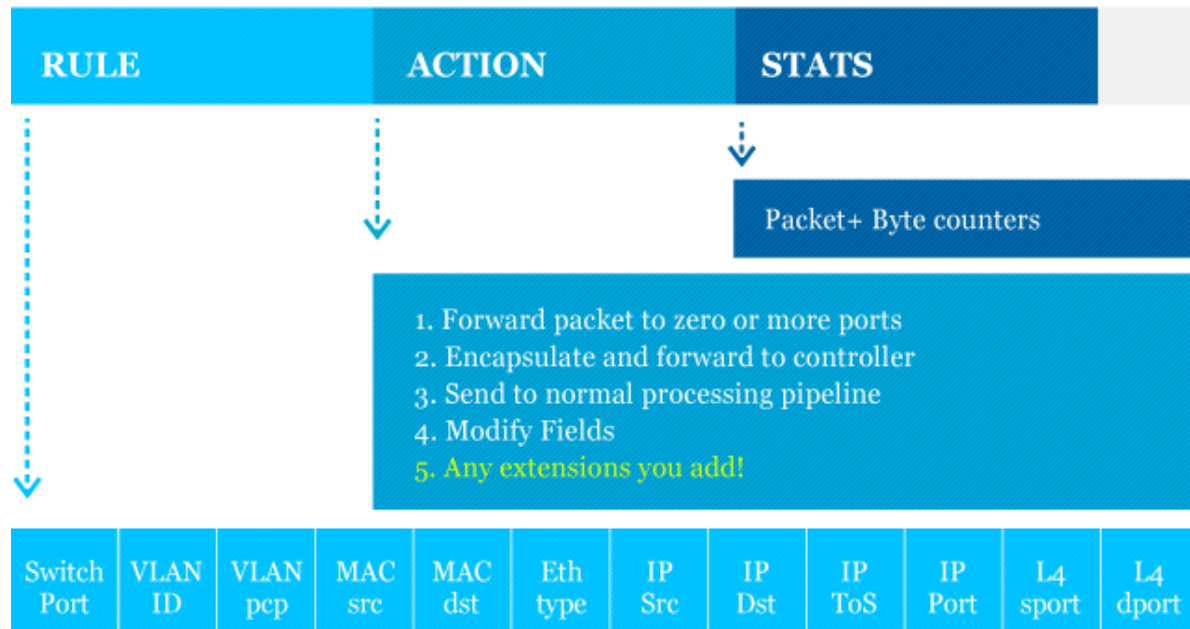


Εικόνα 11 Αναλυτική λειτουργία Flow Table

4.1.5 Flow Entries

Τα Flow Entries είναι τα συστατικά των Flow Tables και αποτελούνται από μια σειρά κατηγοριών/κανόνων που λειτουργούν ως κανόνες κατηγοριοποίησης της κίνησης ανάλογα με την θύρα από την οποία προήλθε το πακέτο, από τον κωδικό ταυτοποίησης του δικτύου, τον αριθμό φυσικής ταυτοποίησης, τύπου διασύνδεσης και άλλων πληροφοριών που το αφορούν. Εφόσον το πακέτο βρίσκει αντιστοίχιση σε κάποια κατηγορία, εκτελείται η δράση που έχει συνδεθεί με αυτήν την αντιστοίχιση και

κρατούνται τα στατιστικά στοιχεία που αφορούν τις παραπάνω πληροφορίες και ενέργειες.



Εικόνα 12 Παράδειγμα Flow Entry

Συνδυασμοί μεταξύ κατηγοριοποιήσεων των Entries μπορούν να προσδώσουν επιπλέον χαρακτηριστικά σε ένα πακέτο, για παράδειγμα να του δοθεί υψηλότερη προτεραιότητα. Σε περίπτωση που δεν υπάρχει αντιστοίχιση σε κανέναν πίνακα (table miss) για το πακέτο η δράση που θα ληφθεί εξαρτάται από την εντολή που έχει συσχετιστεί με το TABLE-MISS FLOW ENTRY η οποία είναι το τελευταίο flow entry σε κάθε flow table.

Σε περίπτωση Table Miss το τελευταίο flow entry μπορεί να προγραμματιστεί προς ανάληψη διαφόρων δράσεων όπως να:

1. ειδοποιείται ο Controller που αναλαμβάνει να δημιουργήσει ένα καινούργιο entry στα flow table που θα αφορά τον τύπο του πακέτου και ορίζει την μελλοντική αντιμετώπιση του
2. αποβάλλει-διαγράφει το πακέτο(drop)
3. το προωθεί σε ένα επόμενο Flow Table
4. να προγραμματιστούν επιπρόσθετες δράσεις

4.1.6 Group table

πίνακας που αποθηκεύει ομάδες θυρών που έχουν δημιουργηθεί (πχ από μια εφαρμογή) και μεταφέρει τις εντολές που έρχονται από τα πάνω επίπεδα σε κάθε μέλος της ομάδας.

4.1.7 Meter table

πίνακας όπου κρατούνται μετρήσεις στα Flows για τα ποσοστά των πακέτων που επεξεργάστηκαν καθώς για συνεργασία με αντίστοιχα Quality Of Service πλαίσια.

Flow Table

Match fields	Priority	Counters	Instructions	Timeout	Cookie
--------------	----------	----------	--------------	---------	--------

Group Table

Group Identifier	Group Type	Counters	Action Buckets
------------------	------------	----------	----------------

Meter Table

Meter Identifier	Meter Bands	Counters
------------------	-------------	----------

MAC src	MAC dst	IP src	IP dst	TCP dport	...	Action	Count
*	10:20:..	*	*	*	*	Port 1	99
*	*	217.99.*	*	*	*	Table 6	18
*	..38:aa:..	*	*	*	*	drop	4
*	*	*	7.7.9.4	25	*	drop	172
*	*	*	*	69	*	local	19
*	*	*	*	*	*	controller	2993

Εικόνα 13 Οι κατηγορίες των Tables και παράδειγμα Flow Table

4.1.8 Control Channel

Το Control Channel είναι η διεπαφή που επιτρέπει στον Controller να παραμετροποιεί και να λειτουργεί το δικτυακό στοιχείο στο οποίο είναι εγκατεστημένο το Open Flow, κυρίως αναφερόμαστε σε Switches όμως υπάρχει η δυνατότητα προέκτασης της συνδεσιμότητας και σε άλλα μηχανήματα όπως Routers και Load Balancers που έχουν συμβατότητα με το πρωτόκολλο.

Σε περιβάλλοντα παραγωγής όπου έχει βαρύνουσα σημασία η προστασία των δεδομένων η επικοινωνία ανάμεσα στα δυο επίπεδα, κρίνεται απαραίτητο η επικοινωνία

να γίνεται κρυπτογραφημένα μέσω TLS (Transport Layer Security) πρωτοκόλλων, όμως υπάρχει η δυνατότητα να λειτουργήσει και μέσω απλού TCP πρωτοκόλλου για λόγους ευκολίας, development και έρευνας.

Καθώς ένα Switch μπορεί να είναι συνδεδεμένο σε πολλαπλούς Controller, ο καθένας τους χρησιμοποιεί έναν παράλληλο δίαυλο επικοινωνίας. Ο αγωγός επικοινωνίας μεταξύ Switch και Controller είναι διαχωρισμένος από τον αγωγό μέσα από τον οποίον περνούν τα πακέτα των δεδομένων του δικτύου.

Για την επικοινωνία του Switch (Control Channel) με το επίπεδο ελέγχου (Controller) γίνεται χρήση μιας σειράς από μηνύματα. Τα μηνύματα αυτά είναι 3 ειδών, Controller to Switch, Asynchronous και Symmetric, όπου ο κάθε τύπος εκτελεί πολλαπλές λειτουργίες.

4.1.8.1 Controller to Switch

Πρόκειται για μηνύματα που αποστέλλονται από τον Controller για να ελέγξει την κατάσταση ενός Switch ή να κάνει αλλαγές πάνω του. Τα μηνύματα αυτά είναι:

Features: όπου ο Controller ζητά από τον Switch την ταυτότητα και τις βασικές δυνατότητες του και η συνήθης χρήση του γίνεται στην αρχική φάση εγκαθίδρυσης ενός OpenFlow διαύλου ανάμεσα στα 2 μηχανήματα.

Configuration: όπου ο Controller ζητά να πληροφορηθεί αλλά και να αλλάξει παραμέτρους διαμόρφωσης του Switch, το οποίο απαντά στην πρώτη περίπτωση μόνο.

Modify-State: μέσω του μηνύματος αυτού ο Controller ενεργεί (προσθέτει, τροποποιεί και διαγράφει) καταχωρήσεις στα OpenFlow tables του Switch καθώς και ρυθμίζει τις ιδιότητες που θα έχουν οι θύρες (ports) του.

Read-State: αυτά τα μηνύματα λειτουργούν για την ενημέρωση του Controller για πληροφορίες των Switches, όπως στατιστικά δεδομένα, ικανότητες και την παρούσα διαμόρφωση του.

Packet-out: τα μηνύματα σε αυτήν την περίπτωση ενημερώνουν για τις αποφάσεις του controller ως προς το ποια θύρα θα χρησιμοποιηθεί για την προώθηση ενός πακέτου και είναι “απαντήσεις” στο “ερώτημα” (Packet-in) που αποστέλλει πρώτα ο switch. Ως

μηνύματα μπορούν να περιέχουν το πλήρες πακέτο ή έναν αριθμό ταυτοποίησης που έχει σχετική αντιστοίχιση στον buffer του Switch.

Barrier: μηνύματα φραγμού που το Controller χρησιμοποιεί για να πληροφορηθεί αν οι διαδικασίες έχουν περαιωθεί σωστά και αν εφαρμόζονται τα ορισμένα κριτήρια.

Role-Request: αυτά τα μηνύματα χρησιμεύουν όταν ένα Switch επικοινωνεί με πολλαπλούς Controllers και χρησιμεύουν στο να καθορίσει ο κάθε Controller τον δίαυλο επικοινωνίας που θα χρησιμοποιηθεί με τον Switch.

Asynchronous-Configuration: ως μηνύματα χρησιμοποιούνται όπως και τα προαναφερθέντα, όταν ένα Switch είναι συνδεδεμένο με πολλούς Controllers, και έχουν χαρακτήρα φίλτρου, όπου ο Controller ορίζει ποια μηνύματα θα λαμβάνει μέσα από την διασύνδεση που έχει οριστεί για την επικοινωνία του με τον Switch.

4.1.8.2 Asynchronous

Αντίστροφα, τα μηνύματα αυτά αποστέλλονται με πρωτοβουλία του Switch με κατεύθυνση τον Controller (και όχι κατά απαίτηση του) και τον ενημερώνουν για συμβάντα στο δίκτυο και αλλαγές στην κατάσταση του μεταγωγέα. Συγκεκριμένα:

Packet-in: με αυτά τα μηνύματα το Switch μεταφέρει τον έλεγχο ενός πακέτου στον Controller ώστε αυτός να αποφασίσει την πορεία του ή όταν ο Controller έχει θέσει ως κανόνα τα συγκεκριμένα πακέτα να περνούν από τον ίδιο πριν προωθηθούν. Είναι τα μηνύματα που προκαλούν την χρήση των Packet-out μηνυμάτων που είδαμε πιο πάνω.

Flow-Removed: μήνυμα πληροφόρησης για την αφαίρεση ενός flow entry από το flow table και είναι απάντηση του Switch στην εντολή του Controller για την σχετική διαγραφή ή όταν έχουμε λήξη κάποιου flow timeout.

Port-status: ενημερώνει τον Controller για αλλαγές στα ports του, δηλαδή σε περίπτωση αλλαγής κατάστασης είτε σε περίπτωση διαμόρφωσης κάποιας θήρας του.

Error: μηνύματα σφάλματος που ειδοποιούν τον Controller σε περίπτωση που εμφανιστούν προβλήματα.

4.1.8.3 Symmetric

Σε αυτήν την περίπτωση τα μηνύματα μπορούν να προέλθουν και από τις 2 κατευθύνσεις έχουν χαρακτήρα αναγνώρισης και επιβεβαίωσης της συνέχειας στην ζεύξη των δυο μηχανημάτων.

Hello: μηνύματα χαιρετισμού που αποστέλλονται μεταξύ Controller και Switch όταν ξεκινά η σύνδεση τους.

Echo: μηνύματα που χρησιμοποιούνται αμφίδρομα για την επιβεβαίωση της σύνδεσης των 2 μηχανημάτων καθώς και για μετρήσεις bandwidth και latency.

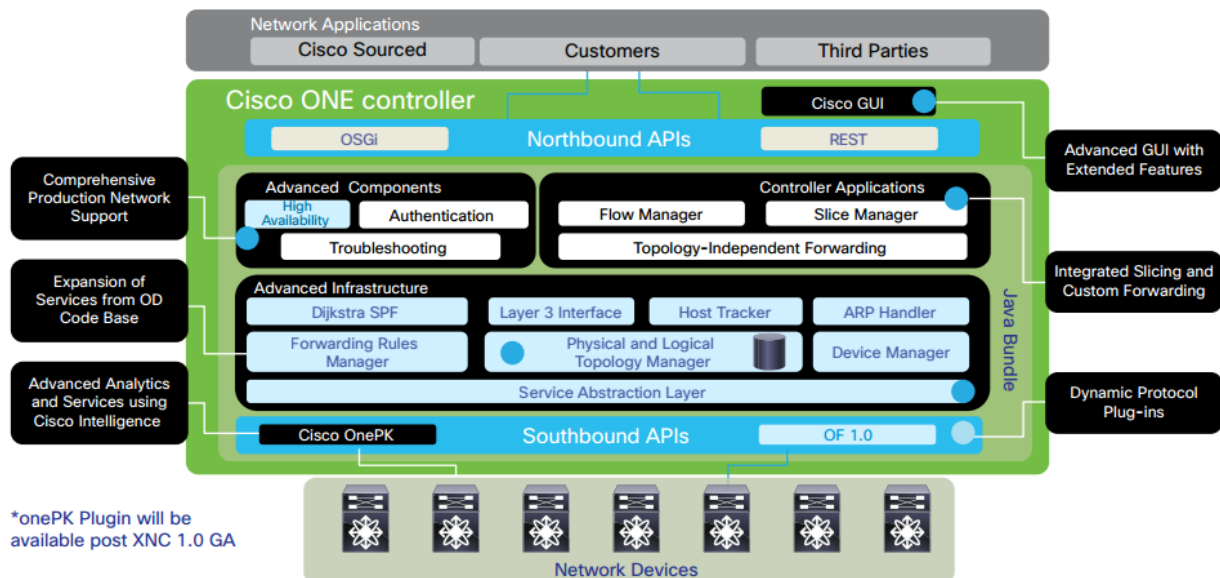
Experimenter: ένας πειραματικός τύπος μηνυμάτων που προορίζεται σε μελλοντικές προσθήκες δυνατοτήτων και ως μηνύματα πειραματισμού για νέα χαρακτηριστικά.

4.2 Επίπεδο Ελέγχου

4.2.1 Controllers - Λειτουργία

Οι Controllers είναι η βασική καινοτομία του Software Defined Networking καθώς αποτελεί το “μυαλό” του δικτύου και η λογικά κεντριοποιημένη δομή που έχει πλήρη εικόνα της κατάστασης καθώς και την δυνατότητα να προβεί σε αλλαγές της ροής κίνησης δεδομένων σε όλο το δίκτυο που ελέγχει. Αποτελείται από μια σειρά εργαλείων που του επιτρέπει λειτουργίες όπως η διαχείριση των Switches, διαχείριση τοπολογίας, διαχείρισης στατιστικών και άλλα.

Εκτός από "μυαλό" ο Controller αποτελεί και "γέφυρα" καθώς είναι το δεύτερο και ενδιάμεσο επίπεδο ανάμεσα στο επίπεδο υλικού και το επίπεδο εφαρμογών τα οποία συνδέει μέσα από μια σειρά εσωτερικών μηχανισμών. Οι μηχανισμοί αυτοί διαφέρουν ανάλογα με την υλοποίηση και τον Controller, όμως κάποια εργαλεία μπορούν να αναγνωριστούν σε όλους τους Controllers, όπως το Service Abstraction Layer. Η λειτουργία επιπέδου αφαίρεσης είναι η "καρδιά" του Controller καθώς του επιτρέπει να υποστηρίξει πολλαπλά επίπεδο επικοινωνίας με το επίπεδο υλικού και να προσφέρει συνεχείς υπηρεσίες σε άλλα εσωτερικά στοιχεία του Controller καθώς και σε εφαρμογές. (Cisco, 2014)



Εικόνα 14 Διάγραμμα του ONE, Controller της Cisco

Οι Controllers είναι στην πλειονότητα τους γραμμένοι σε Java και εγκαθίστανται σε Linux Servers (συνήθως Ubuntu LTS Servers ή Debian servers) μαζί με μια σειρά ακόμα Java, Python και μιας σειράς άλλων προαπαιτούμενων για να λειτουργήσουν σωστά.

Ο Controller όπως είπαμε συνδέεται προς δυο κατευθύνσεις τις οποίες θα δούμε πιο αναλυτικά:

4.2.2 Northbound

Η σύνδεση του με το επίπεδο εφαρμογών (northbound) γίνεται μέσω REST APIs (Διασύνδεση Προγραμματισμού Εφαρμογών), δηλαδή προγραμμάτων που δρουν ως ενδιαμέσοι στην επικοινωνία και εξασφαλίζουν την συμβατότητα μεταξύ των δυο μηχανισμών. Τα REST APIs χρησιμοποιούνται λόγω της χρήσης του Browser ως διεπαφή για την επικοινωνία του διαχειριστή με τον Controller καθώς και για την επικοινωνία των εφαρμογών με τον Controller, ο οποίος ως "εξωγενής" παράγοντας χρειάζεται μια κοινή γλώσσα για να συνεργαστεί μαζί τους, η γλώσσα αυτή είναι τα APIs που είναι εξαιρετικά αποτελεσματικά σε συνδυασμό με http πρωτόκολλα και json εφαρμογές.

4.2.3 Southband

Όσον αφορά στην επικοινωνία με το επίπεδο υλικού αυτή γίνεται κατά βάση μέσω του OpenFlow, το πιο διαδομένο πρωτόκολλο επικοινωνίας controller-switches, όμως μπορούν να χρησιμοποιηθούν μια σειρά ακόμα πρωτοκόλλων ανάλογα με τις ανάγκες για συμβατότητα με επιπέδου υλικού όπως NETCONF, OVSDB, SNMP, BGP/S και άλλων. Για παράδειγμα το OVSDB είναι ιδανικό για επικοινωνία με Open vSwitch πλατφόρμες όπως το Mininet.

4.2.4 East-West

Εκτός της Northbound και Southband επικοινωνίας οι Controllers έχουν την δυνατότητα για East-West ή αλλιώς intercontroller επικοινωνία, δηλαδή την διασύνδεση μεταξύ τους, δυνατότητα που αξιοποιείται σε Multicontroller αρχιτεκτονικές όπου το δίκτυο εργάζεται με άνω του ενός Controller. Η East-West επικοινωνία μπορεί να περιλαμβάνει τον συγχρονισμό των Controllers μεταξύ τους ώστε σε περίπτωση βλάβης στον έναν τον ρόλο της διαχείρισης του δικτύου να αναλάβει ο επόμενος ή για να

υπάρξει συνεργασία όσον αφορά Controllers που βρίσκονται σε διαφορετικά μέρη του δικτύου και λειτουργούν παράλληλα μοιράζοντας μεταξύ τους πληροφορίες για τα δίκτυα τους, για παράδειγμα τοπολογίες και κατάσταση διαφόρων δικτυακών στοιχείων δημιουργώντας έτσι μια πλήρη εικόνα των διασυνδεδεμένων δικτύων-Controllers. (Othmane, 2016)

4.2.5 Single - Multicontroller architectures

Στην SDN αρχιτεκτονική αναφερόμαστε στον Controller όχι ως μια φυσική συσκευή αλλά ως μια λογική οντότητα, κάτι που σημαίνει πως μπορεί να υπάρχουν πολλαπλοί Controllers σε διάφορες διαρρυθμίσεις οι οποίοι όμως γίνονται αντιληπτοί από το δίκτυο ως ένα σύστημα.

Η ανάγκη για πολλαπλούς Controllers στο δίκτυο είναι εμφανής καθώς εάν υπάρχει μόνο ένας προκύπτουν μια σειρά ερωτημάτων όσον αφορά το τι δράσεις μπορούν να ληφθούν αν ο μοναδικός Controller δυσλειτουργήσει, ή προκύψει κάποιο άλλο πρόβλημα.

Έχουν αναπτυχθεί μια σειρά αρχιτεκτονικών που καλύπτουν το εύρος των υλοποιήσεων στο επίπεδο των Controllers:

Φυσικώς κεντριοποιημένη (Physically Centralized): πρόκειται για την αρχιτεκτονική δικτύου που περιλαμβάνει έναν μόνο φυσικό Controller, αρχιτεκτονική που δεν ενδείκνυται καθώς δημιουργεί έναν αδύναμο κρίκο (single point of failure) που σε περίπτωση βλάβης ακυρώνει την λειτουργία του SDN στο δίκτυο.

Φυσικώς κατανεμημένη (Physically Distributed): είναι η αρχιτεκτονική στην οποία γίνεται χρήση περισσότερων από δυο Controllers και η λειτουργία της μπορεί να γίνει μέσω δυο αρχιτεκτονικών:

Λογικά κεντριοποιημένη (Logically Centralized): στην αρχιτεκτονική αυτή λαμβάνουνε υπόψιν την λειτουργία πολλαπλών Controllers όμως το δίκτυο τους αντιλαμβάνεται ως έναν, καθώς όντως μόνο ένας είναι ενεργός σε ότι αφορά το επίπεδο του δικτύου με τους υπόλοιπους να βρίσκονται σε μια συνεχή διαδικασία συγχρονισμού με τον πρώτο. Έτσι σε περίπτωση βλάβης του πρώτου ο δεύτερος να έχει ήδη εικόνα

της τοπολογίας, των Flow Tables και των λοιπών λειτουργιών του δικτύου και να αναλάβει άμεσα καθήκοντα κεντρικού Controller μειώνοντας στο ελάχιστο το downtime στο δίκτυο.

Λογικά κατανεμημένη (Logically Distributed): η συγκεκριμένη αρχιτεκτονική διαχωρίζει σε λογικό και φυσικό επίπεδο τους Controllers, οι οποίοι βρίσκονται σε διάφορα σημεία του δικτύου και χωρίζονται ανά τομέα ευθύνης στον οποίον περιορίζονται οι αρμοδιότητες τους. Τέτοιου είδους αρχιτεκτονικές φαίνεται να αντιτίθενται στην λογική του SDN όμως αποτελούν συγκεκριμένες εξαιρέσεις που εφαρμόζονται σε δίκτυα πολύ μεγάλης έκτασης με μεγάλη απόσταση μεταξύ τους και αντίστοιχα μεγάλα υποδίκτυα, όπως για παράδειγμα το δίκτυο μιας εταιρίας που έχει ένα μέρος του στην Αμερική και ένα άλλο στην Ευρώπη. (Othmane Blial, 2016)

4.2.6 Πρόληψη δυσλειτουργιών

Οι Controllers προσφέρουν μια σειρά χαρακτηριστικών ώστε να μην αφήνεται χώρος για κενά στην ασφάλεια του δικτύου και να επιτρέπουν μια συνεχή κανονική λειτουργία. Όμως παρά τα μέτρα που έχουν ληφθεί σε επίπεδο λειτουργίας Controller έχουν επίσης καταρτιστεί δράσεις που λαμβάνουν υπόψιν και το σενάριο διακοπής της επικοινωνία του Controller με το επίπεδο υλικού (Switches).

Η μη σύνδεση ανάμεσα στα δυο επίπεδα μπορεί να γίνει αντιληπτή από τα Echo μηνύματα που αποστέλλονται τακτικά μεταξύ Switches και Controllers ή από κάποιο TCP timeout. Σε αυτήν την περίπτωση υπάρχει το λεγόμενο Connection Interruption όπου το OpenFlow πρωτόκολλο περιγράφει ως μη δυνατότητα σύνδεσης του Switch με οποιοδήποτε Controller, τότε το Switch εισέρχεται (ανάλογα με τον προγραμματισμό που έχει προϋπάρξει) σε κατάσταση “fail secure mode” ή “fail standalone mode”.

Fail secure mode: λειτουργία για καθαρά OpenFlow Switches που στερούνται εντελώς Data Plane, στην λειτουργία αυτή ο Switch σταματά να στέλνει όσα πακέτα προορίζεται για τον Controller και τα σβήνει μετά από το idle ή hard time-out, και επί της ουσίας σταματά η κίνηση μεταξύ των Switches.

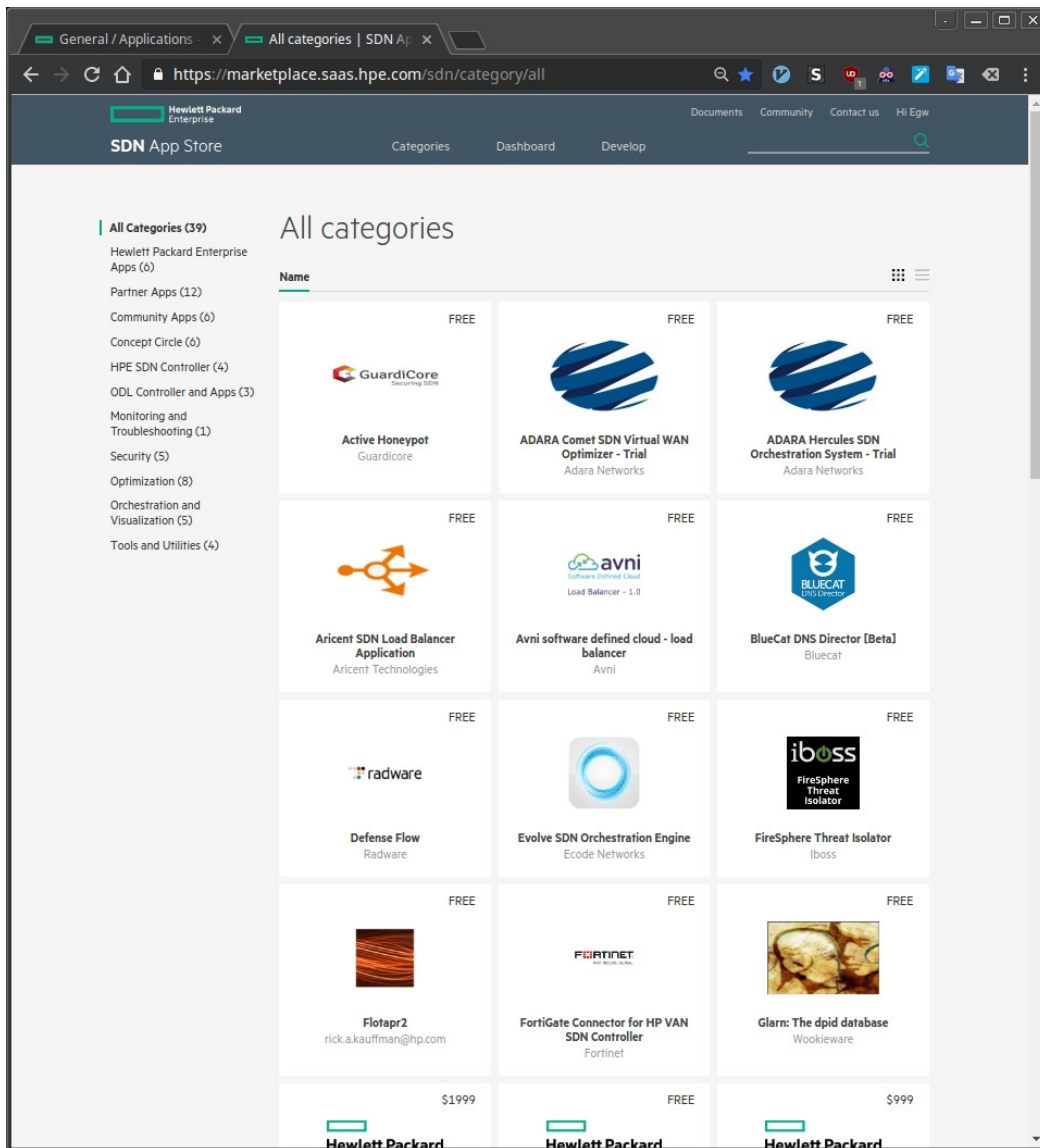
Fail standalone mode: ως δυνατότητα είναι διαθέσιμη κατά βάση στα υβριδικά Switches και επιτρέπει την επεξεργασία των πακέτων μέσω μιας λειτουργίας του

OpenFlow (OFPP_NORMAL) που επιτρέπει την επαναφορά στην παραδοσιακή λειτουργία του Switch εκτός SDN λογικής.

4.3 Επίπεδο Εφαρμογών

Στο υψηλότερο επίπεδο της SDN αρχιτεκτονικής βρίσκονται οι εφαρμογές, που επιτελούν διάφορες συγκεκριμένες εργασίες στο δίκτυο, με κυριότερες την ασφάλεια και την διαχείριση. Οι εφαρμογές βρίσκονται ακόμα σε αρχικό στάδιο εφαρμογής όμως λόγω της ραγδαίας ανάπτυξης που γνωρίζει το Software Defined Networking και ήδη υπάρχουν εφαρμογές που καλύπτουν την λειτουργία σύνθετων μηχανημάτων όπως ενός firewall, load balancer και intrusion detection system. (Paul Goransson)

Οι εφαρμογές είναι κατά κανόνα συνδεδεμένες με τον κατασκευαστή του εκάστοτε Controller που τις αναπτύσσει συνήθως σε συνεργασία με τρίτες εταιρίες με ανάλογες ειδικεύσεις και τις κάνει διαθέσιμες με διάφορους τρόπους. Σε κάποιους Controllers οι εφαρμογές εγκαθίστανται μέσα από την γραμμή εντολών στον μηχανήμα που βρίσκεται ο Controller, σε άλλες περιπτώσεις ο κατασκευαστής δημιουργεί ένα "κατάστημα" προσβάσιμο από το web interface του browser μέσα από τον οποίον ελέγχουμε τον Controller. Μέσα από αυτήν την διεπαφή ο χρήστης μπορεί να κατεβάσει την εφαρμογή την οποία έπειτα κάνει upload στον controller και εγκατάσταση.



Εικόνα 15 Το SDN app store της Hewlett Packard

Εκτός των εφαρμογών που αναπτύσσει η κάθε εταιρία, κάποιες δίνουν την δυνατότητα στα Application Stores τους να φιλοξενηθούν και εφαρμογές τρίτων, συνεργατών και κοινοτήτων, και ορίζει κάποιες προδιαγραφές που πρέπει να πληρούνται για την λειτουργία τους στον Controller και να λάβουν την σχετική ψηφιακή υπογραφή συμβατότητας. Δίδεται η δυνατότητα ωστόσο σε εφαρμογές να τρέξουν χωρίς την ψηφιακή αυτή υπογραφή μέσα από την απενεργοποίηση κάποιων μηχανισμών ασφαλείας και με ευθύνη του χρήστη.

4.3.1 Εφαρμογές Ασφάλειας

Οι εφαρμογές ασφάλειας έχουν κατά κύριο λόγο την λειτουργία ελέγχου και φιλτραρίσματος του δικτύου για ύποπτα πακέτα ή αυξήσεις στην κίνηση χωρίς εμφανή λόγο καθώς μια σειρά άλλων λειτουργιών. (Ericsson, 2015)

Firewalls: δημιουργούν ένα "τείχος" μεταξύ εσωτερικού δικτύου και διαδικτύου ώστε να εμποδίζεται η είσοδος μη εξουσιοδοτημένων χρηστών σε υπηρεσίες εσωτερικών ομάδων και αντίστοιχα να αποτρέπεται η πρόσβαση χρηστών του δικτύου σε μη εγκεκριμένες σελίδες του διαδικτύου.

Honeypots: Είναι τεχνητές "τρύπες" στην ασφάλεια του δικτύου, που εμφανίζονται ως κανονικοί Servers με ευπάθειες ασφάλειας που επιτρέπουν σε εύκολη παραβίαση τους και ως εκ τούτου γίνονται άμεσα στόχος επιθέσεων. Ο επιτιθέμενος καθώς εξασφαλίζει την πρόσβαση του στο δίκτυο-παγίδα τρέφει την ψευδαίσθηση πως έχει εισχωρήσει στο κανονικό δίκτυο και ξεκινά την να κάνει χρήση των γνώσεων και εργαλείων του ώστε να πάρει τον έλεγχο περισσότερων συστημάτων. Το Honeypot είναι δομημένο ώστε να εμφανίζει την εικόνα κανονικού μέρους του δικτύου ενώ στην πραγματικότητα είναι αποκομμένο από αυτό και λειτουργεί μια σειρά εργαλείων που καταγράφουν τις ενέργειες που κάνει ο επιτιθέμενος μέσα σε αυτό. Μέσω του ασφαλούς περιβάλλοντος των Honeypots οι εταιρίες/ερευνητές/κυβερνήσεις μπορούν να είναι ενήμεροι για καινούργιες τακτικές και εργαλεία που αναπτύσσουν οι hackers για τις επιθέσεις τους.

4.3.2 Εφαρμογές Διαχείρισης

Οι εφαρμογές διαχείρισης στοχεύουν στην πιο αποτελεσματική λειτουργία του δικτύου μέσα από μια σειρά εργαλείων και προγραμμάτων που δρουν εποπτικά βοηθώντας τον διαχειριστή του δικτύου να έχει εικόνα των συμβάντων καθώς και να εκτελούν παρεμβατικές ενέργειες που αυξάνουν την λειτουργικότητα του δικτύου (Quality of Service).

Load Balancers: οι συγκεκριμένες εφαρμογές εποπτεύουν την κίνηση στο δίκτυο και ανάλογα με τον το πως έχουν προγραμματιστεί κάνουν έναν ισοκαταμερισμό της κίνησης στο δίκτυο. Λαμβάνοντας υπόψιν την προτεραιότητα των πακέτων που κινούνται στο δίκτυο κάνει αλλαγές στην διαδρομή τους ώστε να μην επιβαρύνονται

συγκεκριμένοι κόμβοι και συνδέσεις ενώ άλλες διαδρομές που μπορούν να εξυπηρετήσουν κίνηση παραμένουν αδρανείς.

Network Visualizers: εφαρμογές που παράγουν εικονικές αναπαραστάσεις διαφόρων στατιστικών δεδομένων και δυναμικών μεταβλητών του συστήματος/δικτύου. Μέσω αυτών επιτυγχάνεται μια πιο εύκολη και γρήγορη κατανόηση της κατάστασης και των καταστάσεων που απαιτούν την προσοχή του διαχειριστή του δικτύου.

5. Υλοποίηση

5.1 Εισαγωγικά

Η υλοποίηση του Software Defined Network για τις ανάγκες της πτυχιακής εργασίας έγινε σε περιβάλλον προσομοίωσης χρησιμοποιώντας παρόμοια ή ίδια εργαλεία με αυτά που λειτουργούν στις εγκαταστάσεις υλικού. Η κύρια διαφορά είναι πως το επίπεδο του υλικού προσομοιώθηκε μέσω Mininet καθώς η εξεύρεση OpenFlow enabled Switches είναι

Αμφότεροι οι Controllers λειτουργούν σε Linux συστήματα, για λόγους ευκολίας και συμβατοτήτων προτιμάται ως βασικό σύστημα κάποιο LTS Ubuntu Server, με την έκδοση 14.04 να είναι η ενδεικνυόμενη. Καθώς η εγκατάσταση θα γίνει σε εικονικό μηχανήμα θα χρειαστεί κάποια πλατφόρμα δημιουργίας/διαχείρισης εικονικών μηχανών, πεδίο στο οποίο υπάρχουν πολλές διαθέσιμες επιλογές, οι δυο γνωστότερες πλατφόρμες είναι το VirtualBox και το VMware. Για την παρούσα εργασία χρησιμοποιήθηκε το VirtualBox λόγω του open source χαρακτήρα του και των δυνατοτήτων που προσφέρει χωρίς της απαίτηση αντιτίμου.

Η εγκατάσταση του VirtualBox δεν απαιτεί εξειδικευμένες γνώσεις, πρέπει ωστόσο να αναφερθούν τα εξής

1. Για να λειτουργήσουν απροβλημάτιστα τα εικονικά μηχανήματα θα απαιτηθεί ένα minimum hardware που περιλαμβάνει κατά ελάχιστο CPU επιπέδου Intel i3, 8GB RAM, 60GB σκληρού δίσκου.
2. Πρέπει να ληφθεί υπόψιν πως η σύνδεση στο διαδίκτυο απαιτείται να γίνει μέσω Ethernet Interface και όχι WiFi, καθώς τα παραπάνω πρωτόκολλα ασφαλείας στην ασύρματη σύνδεση σχεδόν κατά κανόνα δημιουργούν προβλήματα με την Network

Bridged κατάσταση των εικονικών μηχανημάτων. Αν το WiFi είναι η μόνη επιλογή υπάρχει η δυνατότητα επίλυση του θέματος με έναν WiFi extender που να διαθέτει Ethernet έξοδο.

3. Προτιμώνται x64 αρχιτεκτονικές στα προγράμματα που θα εγκατασταθούν για την παρούσα προσομοίωση.

5.2 Επιλογή δικτύου, controller και applications

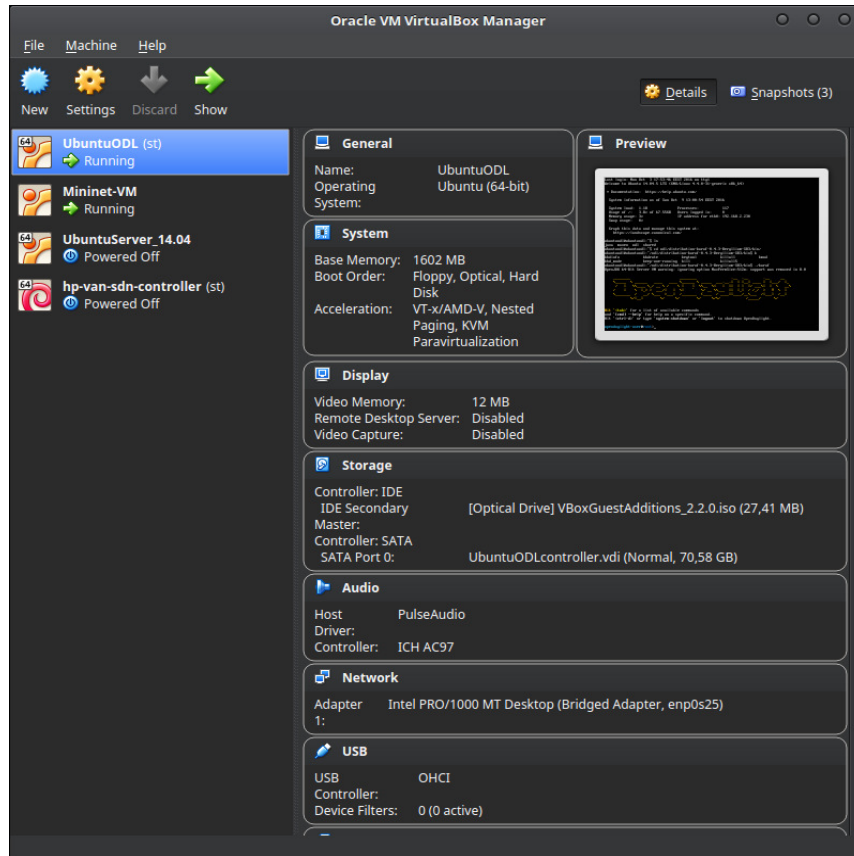
Όπως είδαμε υπάρχει μια πληθώρα διαθέσιμων Controllers για δοκιμή, πειραματισμό και περαιτέρω ανάπτυξη, όμως καθώς δεν θα μπορούσαμε να τους δοκιμάσουμε όλους θεώρησα σκόπιμο να εστιάσω σε δυο Controllers, ο ένας από τον κόσμο του ελεύθερου λογισμικού και βάση των περισσότερων Controllers και ο άλλος κλειστού κώδικα από μια εταιρία που επειδή δεν διαθέτει πρωτοκαθεδρία στον χώρο των δικτύων είναι ανοιχτή σε πειραματισμούς. Επί της παρούσης δηλαδή θα εξετάσουμε τον OpenDaylight και τον HP Van SDN Controller.

Στο κομμάτι της προσομοίωσης θα χρησιμοποιήσουμε το Mininet, το οποίο είναι ένα πρόγραμμα που ανταποκρίνεται πλήρως στις ανάγκες καθώς έχει δημιουργηθεί με γνώμονα την υποστήριξη του Openflow σε εικονικό περιβάλλον. Βασίζεται πάνω στο Open vSwitch.

Σε επίπεδο εφαρμογών θα εγκατασταθούν συγκεκριμένες εφαρμογές από κάθε Controller που επεκτείνουν την λειτουργικότητα του.

5.3 VirtualBox

Εφόσον έχει ολοκληρωθεί η εγκατάσταση του VirtualBox το επόμενο βήμα είναι η εγκατάσταση σε αυτό των 3 εικονικών μηχανημάτων.



Εικόνα 16 Το VirtualBox με το mininet και τους Controllers εγκατεστημένους

Είναι πολύ σημαντικό να επιλεγεί στις ρυθμίσεις του VirtualBox για το κάθε VM, αντί για την NAT επιλογή να επικοινωνεί με το δίκτυο με Bridged Adapter mode ώστε να είναι αναγνωρίσιμο από το δίκτυο ως αυτόνομη οντότητα.

Πολύ σημαντικό τα εικονικά μηχανήματα να έχουν σταθερή IP, λειτουργία που ενεργοποιείται μέσα στο αρχείο:

```
~$ sudo vim /etc/network/interfaces
```

και αλλάζει η τιμή από

```
iface eth0 inet dhcp
```

σε

```
iface eth0 inet static
```

και στην συνέχεια προστίθενται οι σχετικές διευθύνσεις ανάλογα της IP που έχει δοθεί (πχ για το 192.168.2.30)

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.2.30
netmask 255.255.255.0
network 192.168.2.0
broadcast 192.168.2.255
gateway 192.168.2.1
dns-nameservers 8.8.8.8
```

Εικόνα 17 Παράδειγμα του κειμένου στον /etc/network/interfaces

έπειτα σώζουμε το αρχείο και εξερχόμαστε

για να εμφανιστεί η ip του συστήματος χρησιμοποιείται η εντολή

```
~$ ifconfig
```

και για να εξακριβωθεί αν το μηχάνημα έχει απροβλημάτιστη επαφή με το διαδίκτυο εκτελείται ένα ping σε οποιοδήποτε ενεργό σημείο του internet

```
~$ ping www.google.com
```

5.3.1 VirtualBox Guest Additions

Επίσης για να λυθούν τεχνικά θέματα απαιτείται η εγκατάσταση των Guest Additions ώστε να υπάρχει πρόσβαση σε όλες τις δυνατότητες του VirtualBox, όπως οι κοινοί φάκελοι που πιθανόν θα χρειαστούν για να μεταφερθούν αρχεία από το βασικό σύστημα στα εικονικά μηχανήματα.

Η εγκατάσταση γίνεται με την εξής εντολή μέσα στο VM

```
~$ sudo apt-get install virtualbox-guest-additions-iso
```

Είναι πιθανό σε κάποια συστήματα η συγκεκριμένη εντολή να μην λειτουργήσει οπότε και θα χρειαστεί η εγκατάσταση του ISO χειροκίνητα

- download iso και το επιλέγουμε Devices -> Install Guest Additions

έπειτα εισάγονται οι εξής εντολές για να γίνει mount το iso μέσω του εικονικού cdrom, και να εκτελεστεί το αρχείο εγκατάστασης

Αφού ολοκληρωθεί η διαδικασία γίνεται επανεκκίνηση του συστήματος

```
~$ sudo mount /dev/cdrom
```

```
~$ cd /mnt
```

```
~$ sudo ./VboxLinuxAdditions.run
```

```
~$ sudo reboot
```

Για να υπάρξει επιβεβαίωση της εγκατάστασης των GuestAdditions, καθώς και για να εμφανιστούν παραπάνω πληροφορίες σχετικά με την παρούσα έκδοση χρησιμοποιούνται οι κάτωθι εντολές

```
$ dpkg -l | grep virtualbox-guest
```

```
$ lsmod | grep -io vboxguest | xargs modinfo | grep -iw version
```

5.3.2 VirtualBox Shared folder

Για την εύκολη μεταφορά των αρχείων από το σύστημα στα εικονικά μηχανήματα θα χρειαστεί ένας κοινός φάκελος, έχοντας εγκαταστήσει τα Guest Additions είναι εύκολος ο ορισμός ενός κοινού φακέλου.

Δημιουργούμε τον φάκελο και τον ορίζουμε μέσα από το VirtualBox ως default κοινό

Devices -> Shared Folders -> Adds new shared folder

Όπου εισάγεται το path και το όνομα για τον κοινό φάκελο, επίσης γίνεται ενεργοποίηση της επιλογής για το automount

Μετά από αυτό κάνουμε mount τον κοινό φάκελο μέσω της εντολής

```
$ sudo mount -t vboxsf shrdfldrname ~/localfldr
```

όπου shrdfldrname είναι ο τοπικός φάκελος στο μηχάνημα και ο localfldr ο φάκελος στο εικονικό μηχάνημα πάνω στο οποίο γίνεται το mount

5.4 Εγκατάσταση Mininet

Το OVS image είναι διαθέσιμο στην διεύθυνση:

<http://mininet.org/download/>

Γίνεται η εγκατάσταση στο VirtualBox και ορίζεται μια στατική IP

5.5 Εγκατάσταση OpenDaylight

Προχωράμε στην εγκατάσταση ενός τυπικού LTS Ubuntu Server με την SDN συμβατότητα να περιορίζεται έως την έκδοση 14.04

<https://www.ubuntu.com/download/server>

Στην συνέχεια εγκαθιστούμε την νεότερη έκδοση της Java

```
~$ sudo add-apt-repository ppa:openjdk-r/ppa
```

```
~$ sudo apt-get update
```

```
~$ sudo apt-get install openjdk-8-jdk
```

Ορίζεται κύριος φάκελος της Java

```
SET JAVA_HOME
```

```
~$ sudo vim /etc/environment
```

και εισάγεται το εξής κείμενο

```
JAVA_HOME=/usr/lib/jvm/java-8-oracle (ανάλογα τον φάκελο εγκατάστασης της java)
```

```
export JAVA_HOME
```

για να λάβει ισχύ η αλλαγή

```
~$ . /etc/environment
```

αν μετά την επανεκκίνηση του συστήματος χάνεται ο φάκελος και η εντολή

```
~$ echo $JAVA_HOME
```

δεν επιστρέφει κάτι ακολουθείται διαφορετικά προσέγγιση μέσω bash profile

```
~$ vim .bash_profile
```

```
~$ export JAVA_HOME=/usr/lib/jvm/java-8-oracle (ή java-8-openjdk-amd64
```

και ελέγχεται πάλι αν το σύστημα επιστρέφει απάντηση στο

```
~$ echo $JAVA_HOME
```

Εγκατάσταση του OpenDaylight καθεαυτού

επιλογή της νεότερης έκδοσης (Pre-Built Tar File) από την σελίδα

<https://www.opendaylight.org/downloads>

αφού κατέβει και αποσυμπιεστεί το περιεχόμενο και εκτελείται το script εγκατάστασης

```
~$ tar -xvzf && . /distribution-karaf-0.4.2-Beryllium-SR2/bin/karaf
```

εφόσον η εγκατάσταση έχει ολοκληρωθεί

αφού ανοίξει την κονσόλα γίνεται δυνατό να εισάγουμε όποια προγράμματα κρίνονται απαραίτητα και χρήσιμα για την λειτουργία του, αυτό γίνεται με την εντολή feature:install, για παράδειγμα:

```
~$ feature:install odl-dlux-core odl-dlux-all odl-restconf odl-l2switch-switch
```

οι βασικές εγκαταστάσεις είναι οι:

5.6 Εγκατάσταση HP VAN SDN Controller

Καθώς έχουμε να κάνουμε με έναν Controller μία μεγάλης εταιρίας που περιορίζεται σε μια περίοδο δοκιμής 60 ημερών θα μας ζητηθεί να φτιάξουμε λογαριασμό για να κατεβάσουμε τον Controller στο εξής link

<https://marketplace.saas.hpe.com/sdn/content/sdn-controller-free-trial>

Αν δεν επιθυμούμε να μπούμε στην διαδικασία δημιουργίας λογαριασμού μπορούμε να τον κατεβάσουμε από το παρακάτω link όπου υπάρχουν όλες οι εκδόσεις και

<https://h10145.www1.hp.com/downloads/SoftwareReleases.aspx?ProductNumber=J9863AAE&lang=&cc=&prodSeriesId=&SoftwareReleaseUIId=10608&SerialNumber=&PurchaseDate>

Αφού συμφωνήσουμε στους όρους χρήσης της HP και κατεβάσουμε το zip το μεταφέρουμε μέσω του κοινού φακέλου στο VM μας

Πριν τρέξουμε την εγκατάσταση όμως θα πρέπει να κάνουμε σίγουρο πως έχουμε όλα τα στοιχεία που απαιτούνται για να τρέξει ο Controller σωστά, οπότε

1 Εγκατάσταση python

```
~$ sudo apt-get install python-software-properties
```

```
ubuntu-cloud-keyring
```

```
~$ sudo add-apt-repository "deb
```

```
http://ubuntu-cloud.archive.canonical.com/ubuntu
```

```
precise-updates/folsom main"
```

```
~$ sudo apt-get update
```

2. Εγκατάσταση java

```
~$ sudo apt-get install openjdk-7-jre-headless postgresql
```

```
keystone keystone-doc python-keystone iptables unzip
```

3. Έλεγχος της παρούσας έκδοσης της Java την ορισμός της ως προεπιλεγμένη έκδοση

```
~$ sudo update-java-alternatives -l
```

ανάλογα με το ποια έκδοση επιστρέφει η εντολή θα γίνει και η αντίστοιχη εισαγωγή στην στην εντολή

```
~$ sudo update-java-alternatives --jre -s
```

```
java-1.7.0-openjdk-amd64
```

4. Εγκατάσταση του controller

```
~$ sudo dpkg -i hp-sdn-ctl_version_amd64.deb
```

ανάλογα με την έκδοση που υπάρχει

5. Για να βεβαιωθούμε πως ο controller εγκαταστάθηκε επιτυχώς και να ενημερωθούμε για την κατάσταση του μπορούμε να κάνουμε χρήση αυτών των εντολών

```
~$ dpkg -l hp-sdn-ctl
```

```
~$ sudo service sdnc status
```

Για την εγκατάσταση 3ων εφαρμογών χωρίς ψηφιακή υπογραφή από την HP ή κάποιον συνεργάτη της κάνουμε τις εξής ενέργειες

1. Σταματάμε τον controller:

```
$ sudo service sdnc stop
```

2. Τροποποιούμε το αρχείο /opt/sdn/virgo/bin/dmk.sh script to include the line shown below in

boldface type:


```
~$ cd $KERNEL_HOME; exec $JAVA_EXECUTABLE \  
$JAVA_OPTS \  
$DEBUG_OPTS \  
$JMX_OPTS \  
-XX:+HeapDumpOnOutOfMemoryError \  
-XX:ErrorFile=$KERNEL_HOME/serviceability/error.log \  
-XX:HeapDumpPath=$KERNEL_HOME/serviceability/heap_dump.hprof \  
#Προσθέτουμε την εξής γραμμή  
-Dsdn.signedJar=none \  
-Djava.security.auth.login.config=$AUTH_LOGIN \  
-Dorg.eclipse.virgo.kernel.authentication.file=$AUTH_FILE \  

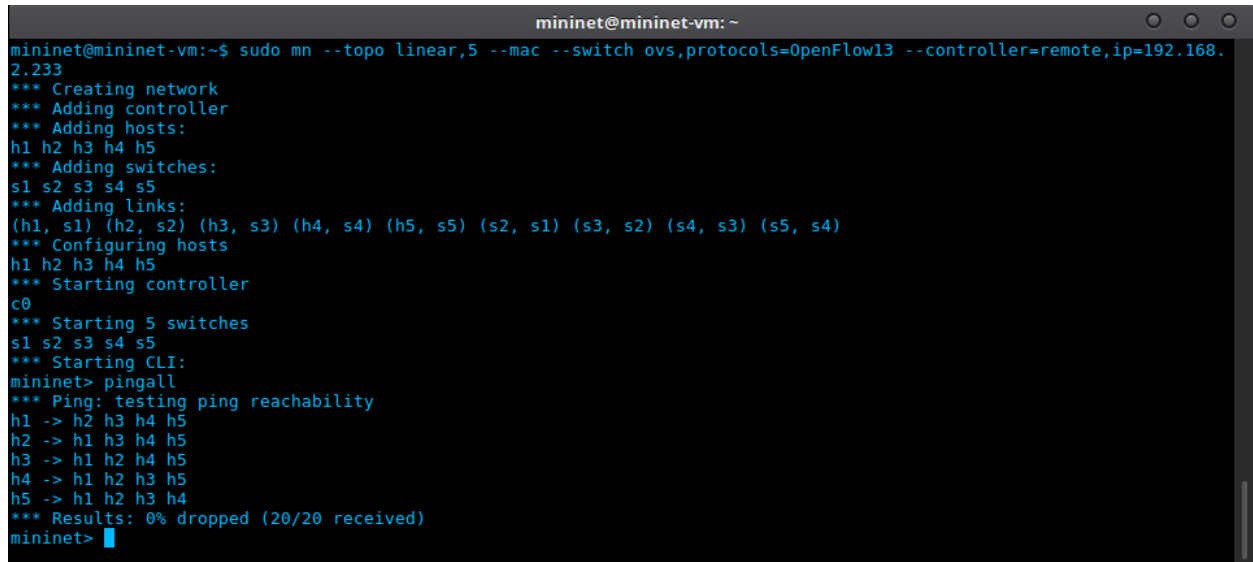
```

3. Ξεκινάμε πάλι τον controller:

```
$ sudo service sdnc start
```

5.7 Χρήση Mininet

Το Mininet προσφέρει έναν ευέλικτο μηχανισμό προσομοίωσης δεικτών μέσω CLI τα οποία αφού επιλέξουμε την τοπολογία, το μέγεθος και άλλες μεταβλητές “στρέφουμε” προς κάποιον SDN Controller από τον οποίο μπορούμε να έχουμε περισσότερες επιλογές.



```
mininet@mininet-vm:~$ sudo mn --topo linear,5 --mac --switch ovs,protocols=OpenFlow13 --controller=remote,ip=192.168.2.233
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5
*** Adding switches:
s1 s2 s3 s4 s5
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (h5, s5) (s2, s1) (s3, s2) (s4, s3) (s5, s4)
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting controller
c0
*** Starting 5 switches
s1 s2 s3 s4 s5
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> h1 h2 h3 h5
h5 -> h1 h2 h3 h4
*** Results: 0% dropped (20/20 received)
mininet>
```

Εικόνα 18 Η δημιουργία εικονικού δικτύου σε περιβάλλον Mininet

Η κύρια εντολή στο Mininet είναι σαφώς αυτή της παραγωγής δικτύου

```
~$ sudo mn --controller=remote,ip=192.168.2.230 --topo=linear,5 --mac
```

Ανάλυση της εντολής:

```
--mac
```

το mac εμφανίζει την φυσική διεύθυνση του μηχανήματος σε αριθμητικά ορισμένη σειρά και όχι στην "κανονική", διευκολύνοντας έτσι την καταγραφή των δικτυακών στοιχείων

```
--controller=remote,ip=192.168.2.***
```

η εντολή αυτή ενημερώνει το Mininet πως το δίκτυο θα ελέγχεται από εξωτερικό Controller και του παρέχει την IP διεύθυνση επικοινωνίας μαζί του

```
--topo=linear,5
```

στην επιλογή topo= συμπληρώνεται ο τύπος της τοπολογίας και ο αριθμός των κόμβων που θέλουμε να εισάγουμε στο δίκτυο

οι διαθέσιμες τοπολογίες είναι οι εξής:

- Linear
- Tree
- Torus
- Single
- Reversed
- Minimal

```
~$ sudo mn -c
```

η συγκεκριμένη εντολή "καθαρίζει" την μνήμη του Mininet, σε περίπτωση που τα κατάλοιπα κάποιου παλιότερου δικτύου παρεμβάλλονται του καινούργιου

Πληροφορίες για θύρες και Flow Entries

```
~$ sudo ovs-vct1 show
```

εμφανίζει πληροφορίες σχετικά με την μορφή του δικτύου

```
~$ sudo ovs-ofctl -O OpenFlow13 dump-ports s1
```

εμφανίζει τις θύρες του Switch 1 που είναι ενεργές μέσα στο OpenFlow πρωτόκολλο

```
~$ sudo ovs-ofctl -O OpenFlow13 dump-flows s1
```

εμφανίζει τα Flow Entries που αφορούν το Switch 1

```
~$ sudo mn -h | more
```

```
# εμφανίζεται η σελίδα με τις διαθέσιμες εντολές σε αναλυτική μορφή
```

```
~$ sudo net
```

```
# εμφανίζονται οι συνδέσεις του δικτύου
```

```
~$ sudo pingall
```

```
# οι Hosts/Υπολογιστές προβαίνουν σε μια σειρά ping ώστε να καθοριστεί η θέση τους
```

```
~$ sudo dump
```

```
# εμφανίζονται γενικές πληροφορίες για το δίκτυο
```

```
~$ sudo link s1 h1 down
```

```
# η συγκεκριμένη εντολή προσομοιάζει την αποσύνδεση της Ethernet σύνδεση μεταξύ των δυο hosts
```

5.8 Χρήση OpenDaylight

Το OpenDaylight είναι ο πιο διαδομένος Controller που αποτελεί την βάση για πολλούς εταιρικούς controllers και αυτό το γνώρισμα του τον διαφοροποιεί σε αρκετές λειτουργίες από τους εμπορικούς Controllers. Καθώς είναι πλατφόρμα δοκιμών και ανάπτυξης δεν έχει πολλές εξειδικευμένες λειτουργίες εξαρχής αλλά αυτές πρέπει να εγκατασταθούν μέσα από το Command Line στον Server της αρχικής εγκατάστασης. Το βάρος της προσαρμογής του έχει δοθεί στο επίπεδο γραμμής εντολών για να είναι πιο απλός, κοντά στον κώδικα, εύκολος στην κατανόηση από τους προγραμματιστές που τον χρησιμοποιούν για την εξέλιξη παράλληλων Controllers.

Πρέπει επίσης να ληφθεί υπόψιν πως ο OpenDaylight Controller έχει περάσει από διάφορα στάδια ανάπτυξης, κάτι που έχει οδηγήσει σε 4 εκδόσεις, κατά χρονολογική σειρά:

- Hydrogen Φεβρουάριος 2014
- Helium Οκτώβριος 2014
- Lithium Ιούνιος 2015
- Beryllium Φεβρουάριος 2016

Κάθε μια από τις παραπάνω εκδόσεις περιλαμβάνει αλλαγές και προσθήκη/αφαίρεση χαρακτηριστικών για την αποτελεσματικότερη λειτουργία του Controller. Αυτό που χρειάζεται να ληφθεί υπόψιν είναι πως μεταξύ των εκδόσεων είναι πολύ πιθανόν να προκύπτουν ασυμβατότητες όσον αφορά τις εφαρμογές και τον τρόπο χρήσης τους.

5.8.1 Είσοδος

Για να ξεκινήσει η λειτουργία του OpenDaylight μεταφερόμαστε στον Linux Server της εγκατάστασης του, εντοπίζεται ο φάκελος στον οποίον είναι εγκατεστημένος και από εκεί ακολουθείται η διαδρομή μέσα από τον /bin φάκελο και στην συνέχεια εκτελείται το karaf (ο container που περιέχει τον Controller). Στην συνέχεια εμφανίζεται το λογότυπο του OpenDaylight και γίνεται προσβάσιμο το πεδίο opendaylight-user@root από όπου εισάγονται οι εντολές που αφορούν τον Controller. Οι εντολές αυτές μπορούν να είναι διαφόρων ειδών, όπως η εγκατάσταση εφαρμογών, η παραμετροποίηση λειτουργιών, το "καθάρισμα" της μνήμης του Controller και άλλα, πλήρης λίστα των εντολών μπορεί να εμφανιστεί με την εντολή "help".

Μετά την εγκατάσταση των λειτουργιών ο Controller μπορεί να είναι προσβάσιμος από οποιονδήποτε Browser εισάγοντας στο url field

<http://192.168.2.localhost:8181/index.html#/login>

αφού αγνοηθεί το μήνυμα της μη ασφαλούς σελίδας εισάγεται admin στο πεδίο του χρήστη και επίσης admin στο πεδίο του κωδικού, credentials τα οποία μπορούν να αλλαχθούν στην συνέχεια αν επιθυμούμε, πράγμα που ενδείκνυται για την δημιουργία ενός πιο ασφαλούς περιβάλλοντος.

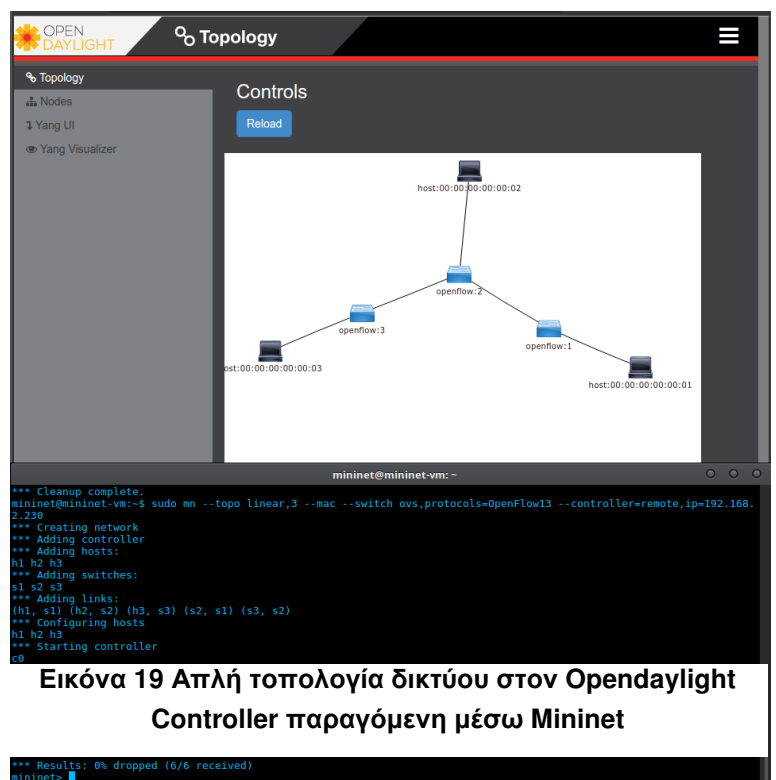
5.8.2 Περιβάλλον

Το περιβάλλον χρήστη είναι αρκετά λιτό με λίγες επιλογές, με τις κύριες να είναι η εμφάνιση της τοπολογίας, των κόμβων (Switches και Υπολογιστών) και τις Yang λειτουργίες που αποτελούν ένα σημαντικό εργαλείο διαχείρισης των λειτουργιών του OpenDaylight. Οι επιλογές αυτές εμπλουτίζονται μέσω της προσθήκης εφαρμογών από την γραμμή εντολών του OpenDaylight, με διάφορες επιλογές να είναι διαθέσιμες, όπως Network Virtualization, Security και Network Optimization εφαρμογές. Στις νεότερες εκδόσεις του OpenDaylight προστέθηκε η δυνατότητα της απόκρυψης των επιλογών όταν η εμφάνιση τους δεν είναι αναγκαία, ώστε να γίνεται καλύτερη διαχείριση του χώρου στην οθόνη.

5.8.3 Τοπολογία

Η τοπολογία εμφανίζεται με δυναμικό τρόπο στον OpenDaylight και έχει διαδραστικά χαρακτηριστικά, εμφανίζοντας για παράδειγμα πληροφορίες με την αιώρηση του δείκτη του ποντικιού πάνω από Switches, Hosts (H/Y) και Links (Συνδέσεων) εμφανίζονται πληροφορίες που αφορούν την κατάσταση και τον ρόλο τους στο δίκτυο. Για την ανανέωση της τοπολογίας πρέπει να γίνεται επιλογή του Reload κουμπιού πάνω από τον πίνακα της τοπολογίας, αυτό δεν γίνεται αυτόματα καθώς η λειτουργία της τοπολογίας δεν έχει monitoring χαρακτηριστικά. Επίσης για να ενημερωθεί ο Controller για τους

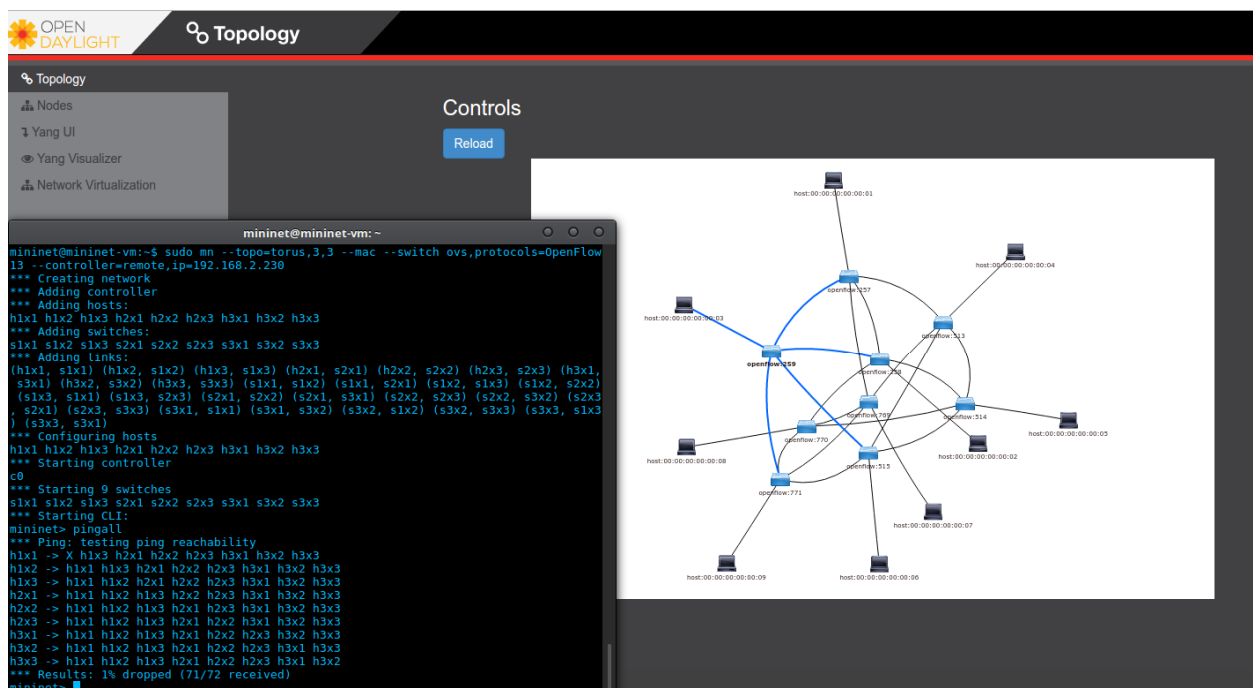
Hosts και να γίνουν εμφανείς αυτοί στην τοπολογία πρέπει να εκτελεστεί η εντολή ringall στο Mininet που παράγει rings από όλους τους Hosts και δημιουργεί την πλήρη εικόνα των στοιχείων του δικτύου. Η λειτουργία του ringall είναι απαραίτητη στις προσομοιώσεις επειδή οι υπολογιστές των προσομοιώσεων δεν παράγουν κίνηση μόνοι τους αλλά μόνο κατόπιν εντολής. Σε πραγματικές συνθήκες οι H/Y παράγουν "μόνοι



Εικόνα 19 Απλή τοπολογία δικτύου στον Opendaylight Controller παραγόμενη μέσω Mininet

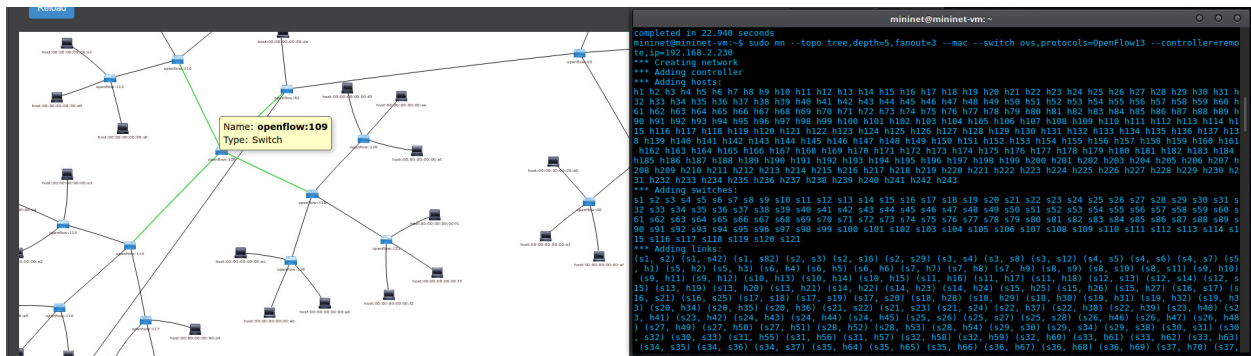
τους" την κίνηση καθώς κάθε φορά που συνδέονται σε ένα δίκτυο μια σειρά προγραμμάτων και εργαλείων του Λειτουργικού Συστήματος στέλνουν πακέτα στο δίκτυο με προορισμό το διαδίκτυο, πακέτα που μέσω των OpenFlow enabled Switches ενημερώνουν τον Controller για την ύπαρξη του H/Y στην συγκεκριμένη τοποθεσία. Επιπλέον μπορεί να εφαρμοστεί μέσω εφαρμογών ή scripts του Controller η αποστολή Broadcast μηνυμάτων για ενεργητική σάρωση του δικτύου και εντοπισμό μηχανημάτων.

Οι τοπολογίες σε ένα δίκτυο μπορούν να είναι διαφορετικών ειδών, ανάλογα με το είδος του και τα κριτήρια που τίθενται και σε αυτόν τον τομέα το Mininet μπορεί να συνεργαστεί άριστα με τον OpenDaylight Controller όσον αφορά την δημιουργία, απεικόνιση και διαχείριση πολυσύνθετων δικτύων που περιλαμβάνουν πολλαπλά κανάλια επικοινωνίας ανάμεσα στα Switches ώστε με αυτόν το τρόπο να ελαττωθεί η πιθανότητα της αποκοπής του ενός μέρους του δικτύου από το άλλο λόγω ενός Single Point of Failure. Για παράδειγμα ένας Torus σχεδιασμός σαν αυτός της εικόνας μπορεί να εξασφαλίσει το μέγιστο επίπεδο ασφάλειας κάνοντας παράλληλα πολύ εύκολη και αποδοτική μια Load Balancing πολιτική με καταμερισμό της κίνησης ανάμεσα σε διαφορετικούς κόμβους και διαδρομές στο δίκτυο.



Εικόνα 20 Torus τοπολογία με τέσσερα κανάλια επικοινωνίας

Πέραν αυτών η τοπολογία όσον αφορά δίκτυα μεγάλων αλλά ακόμα και μεσαίων μεγεθών παίζει πολύ σημαντικό ρόλο στο πως θα γίνει η σχεδίαση και η λειτουργία του από έναν μηχανικό δικτύων. Εκτός των τεχνικών λεπτομερειών τα δίκτυα είναι μια μορφή, ένα σχήμα που αποκτά μορφή μέσα από τα γράμματα και τους αριθμούς του terminal, όμως και πάλι αφήνεται ένα περιθώριο αοριστίας ως προς την εικόνα που μπορεί να έχει ένα δίκτυο. Για έναν μηχανικό δικτύων ή έναν διαχειριστή συστημάτων η εμφάνιση μιας ολοκληρωμένης και δυναμικής εικόνας ενός μεγάλου δικτύου σε διεπαφή γραφικού περιβάλλοντος, παρέχει μια σειρά πλεονεκτημάτων, όπως η άμεση αντίληψη της μορφής του δικτύου, η εύκολη λήψη πληροφοριών μέσω Point and Click στα μηχανήματα της τοπολογίας όπως και η αποτελεσματικότερη διαχείριση βλαβών όπου ο εντοπισμούς του προβληματικού σημείου γίνεται πολύ ευκολότερος καθώς και ο σχεδιασμός εναλλακτικής διαδρομής για την κίνηση.



Εικόνα 21 Παράδειγμα διαχείρισης τοπολογίας 121 Switches

Ο OpenDaylight επιτρέπει επίσης το zoom in/out και ελεύθερη κίνηση στις τοπολογίες ώστε να είναι πιο εύκολη η εις βάθος ανάγνωση και κατανόηση τους, ακόμα και σε δίκτυα που περιλαμβάνουν έναν πολύ μεγάλο αριθμό δικτυακών μηχανημάτων.

5.8.4 Nodes – στοιχεία δικτύου

Στην δεύτερη επιλογή του OpenDaylight Controller, των Nodes, γίνονται προσβάσιμες μια σειρά πληροφοριών σχετικά με τα Switches, όπως ο αριθμός/όνομα τους και μια σειρά πληροφοριών όσον αφορά τα στοιχεία του του δικτύου που είναι συνδεδεμένο με τον κάθε Switch, όπως την θύρα από την οποία συνδέονται, το όνομα τους, ο κωδικό διασύνδεσης τους, η φυσική τους διεύθυνση. Στην εικόνα για παράδειγμα εμφανίζεται η πλήρης εικόνα του δικτύου μέσω Mininet, που περιλαμβάνει 3 Switches όμως στο OpenDaylight εμφανίζονται τέσσερα πεδία,

The screenshot shows the OpenDaylight Nodes page. On the left, there is a sidebar with 'Nodes' selected, showing 'Yang UI' and 'Yang Visualizer'. The main content area is titled 'Node Id - openflow:2' and contains a search bar and a table of Node Connectors. The table has four columns: 'Node Connector Id', 'Name', 'Port Number', and 'Mac Address'. Below the table, there is a terminal window showing the output of the 'show' command in Mininet, displaying details for three bridges (s3, s2, s1) and their connections to controllers.

Node Connector Id	Name	Port Number	Mac Address
openflow:2:1	s2-eth1	1	2A:D7:81:23:1B:E9
openflow:2:LOCAL	s2	LOCAL	EE:01:57:B2:E7:4F
openflow:2:2	s2-eth2	2	16:E4:15:2B:A7:B9
openflow:2:3	s2-eth3	3	42:59:16:97:80:D2

```
mininet@mininet-vm:~$ sudo ovs-vsctl show
8945dad2-35b4-4549-9122-23d5558992e7
Bridge "s3"
  Controller "tcp:192.168.2.230:6633"
  is connected: true
  Controller "ptcp:6636"
  fail mode: secure
  Port "s3-eth1"
  Interface "s3-eth1"
  Port "s3-eth2"
  Interface "s3-eth2"
  Port "s3"
  Interface "s3"
  type: internal
Bridge "s2"
  Controller "ptcp:6635"
  Controller "tcp:192.168.2.230:6633"
  is connected: true
  fail mode: secure
  Port "s2"
  Interface "s2"
  type: internal
  Port "s2-eth2"
  Interface "s2-eth2"
  Port "s2-eth1"
  Interface "s2-eth1"
  Port "s2-eth3"
  Interface "s2-eth3"
Bridge "s1"
  Controller "ptcp:6634"
  Controller "tcp:192.168.2.230:6633"
  is connected: true
  fail mode: secure
  Port "s1-eth1"
  Interface "s1-eth1"
  Port "s1"
  Interface "s1"
  type: internal
  Port "s1-eth2"
  Interface "s1-eth2"
  ovs version: "2.0.2"
mininet@mininet-vm:~$
```

Εικόνα 22 Πληροφορίες σχετικά με τα Nodes όπως εμφανίζονται σε OpenDaylight και Mininet

στο Connector ID, με το επιπλέον να ονομάζεται LOCAL. Το συγκεκριμένο

Node Connector ID, ή Ταυτότητα Διασυνδεδεμένου κόμβου αναφέρεται στην δίοδο σύνδεσης του Switch με τον Controller, καθώς όπως έχει αναφερθεί κάθε Switch επικοινωνεί απευθείας με τον Controller μέσω τους OpenFlow Channel που δεσμεύει μια θύρα σε αυτήν μόνο την λειτουργία.

Πέρα των παραπάνω πληροφοριών πρόσβαση δίνεται και σε στατιστικά στοιχεία (μέσω των Meter Table που αναφέρθηκαν στο προηγούμενο κεφάλαιο) που αφορούν ποιοτικά και ποσοτικά χαρακτηριστικά των πακέτων που πέρασαν από έκαστο Switch.

The screenshot shows the OpenDaylight Nodes interface. The top part displays 'Node Connector Statistics for Node Id - openflow:2' with a table of statistics. Below this is a terminal window showing the execution of 'sudo ovs-ofctl -0 OpenFlow13 dump-ports s2' and 'sudo ovs-ofctl -0 OpenFlow13 dump-flows s2'.

Node Connector Id	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
openflow:2:1	13	97	978	7975	0	0	0	0	0	0	0	0
openflow:2:LOCAL	0	0	0	0	0	0	0	0	0	0	0	0
openflow:2:2	87	96	7275	7905	0	0	0	0	0	0	0	0
openflow:2:3	89	94	7415	7765	0	0	0	0	0	0	0	0

```

mininet@mininet-vm:~$ sudo ovs-ofctl -0 OpenFlow13 dump-ports s2
OFPST_PORT reply (OF1.3) (xid=0x2): 4 ports
  port 3: rx pkts=112, bytes=9370, drop=0, errs=0, frame=0, over=0, crc=0
          tx pkts=117, bytes=9720, drop=0, errs=0, coll=0
          duration=506.268s
  port 1: rx pkts=13, bytes=978, drop=0, errs=0, frame=0, over=0, crc=0
          tx pkts=120, bytes=9930, drop=0, errs=0, coll=0
          duration=506.268s
  port 2: rx pkts=110, bytes=9230, drop=0, errs=0, frame=0, over=0, crc=0
          tx pkts=119, bytes=9860, drop=0, errs=0, coll=0
          duration=506.268s
  port LOCAL: rx pkts=0, bytes=0, drop=0, errs=0, frame=0, over=0, crc=0
              tx pkts=0, bytes=0, drop=0, errs=0, coll=0
              duration=506.215s
mininet@mininet-vm:~$ sudo ovs-ofctl -0 OpenFlow13 dump-flows s2
OFPST_FLOW reply (OF1.3) (xid=0x2):
  cookie=0x2b00000000000010, duration=523.361s, table=0, n_packets=10, n_bytes=700, priority=2,in_port=3 actions=output:1,output:2
  cookie=0x2b0000000000000e, duration=523.361s, table=0, n_packets=8, n_bytes=560, priority=2,in_port=1 actions=output:2,output:3,CONTROLLER:65535
  cookie=0x2b0000000000000f, duration=523.361s, table=0, n_packets=8, n_bytes=560, priority=2,in_port=2 actions=output:1,output:3
  cookie=0x2b00000000000012, duration=528.483s, table=0, n_packets=211, n_bytes=17935, priority=100,dl_type=0x88cc actions=CONTROLLER:65535
  cookie=0x2b00000000000012, duration=528.483s, table=0, n_packets=4, n_bytes=328, priority=0 actions=drop
mininet@mininet-vm:~$

```

Εικόνα 23 Στατιστικά πακέτων και Flow Entries όπως αυτά εμφανίζονται σε OpenDaylight και Mininet

Στην παραπάνω εικόνα γίνονται εμφανή αυτά τα στατιστικά όπως εμφανίζονται μέσω του interface του Controller και όπως εμφανίζονται στο Terminal του Mininet. Εκτός αυτού μέσω της εντολής

```
~$ sudo ovs-ofctl Openflow13 dump-flows s2
```

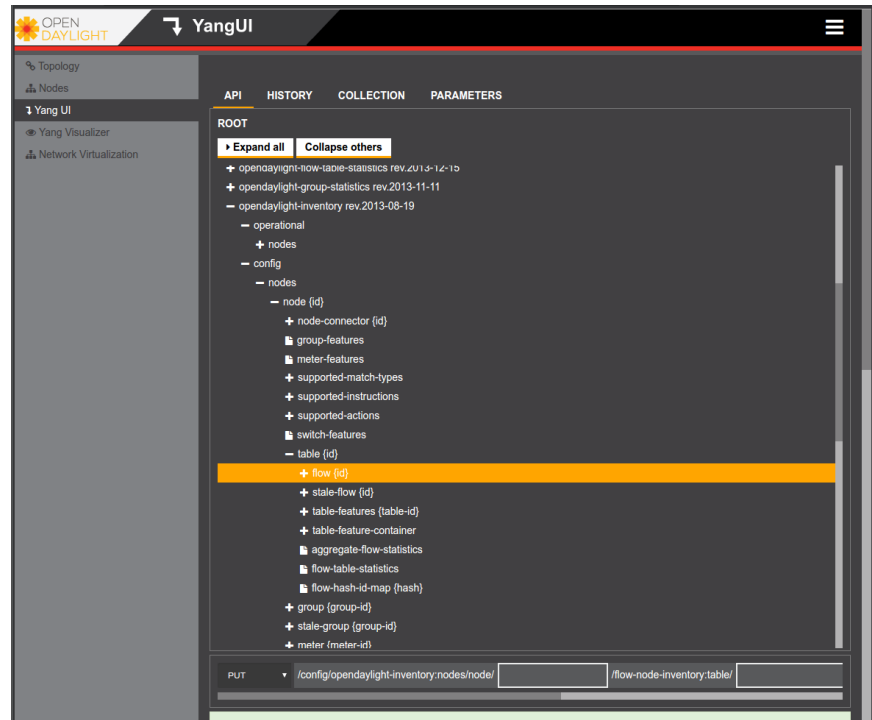
όπου ζητείται από τον Switch 2 να εμφανίσει τα Flow Entries του OpenFlow πρωτοκόλλου

στο Mininet εμφανίζονται τα Flows των οποίων τα στατιστικά δεν εμφανίζονται στον Controller όμως έχουν άμεση επιρροή στην λειτουργία των πακέτων που εμφανίζονται.

5.8.5 Yang UI

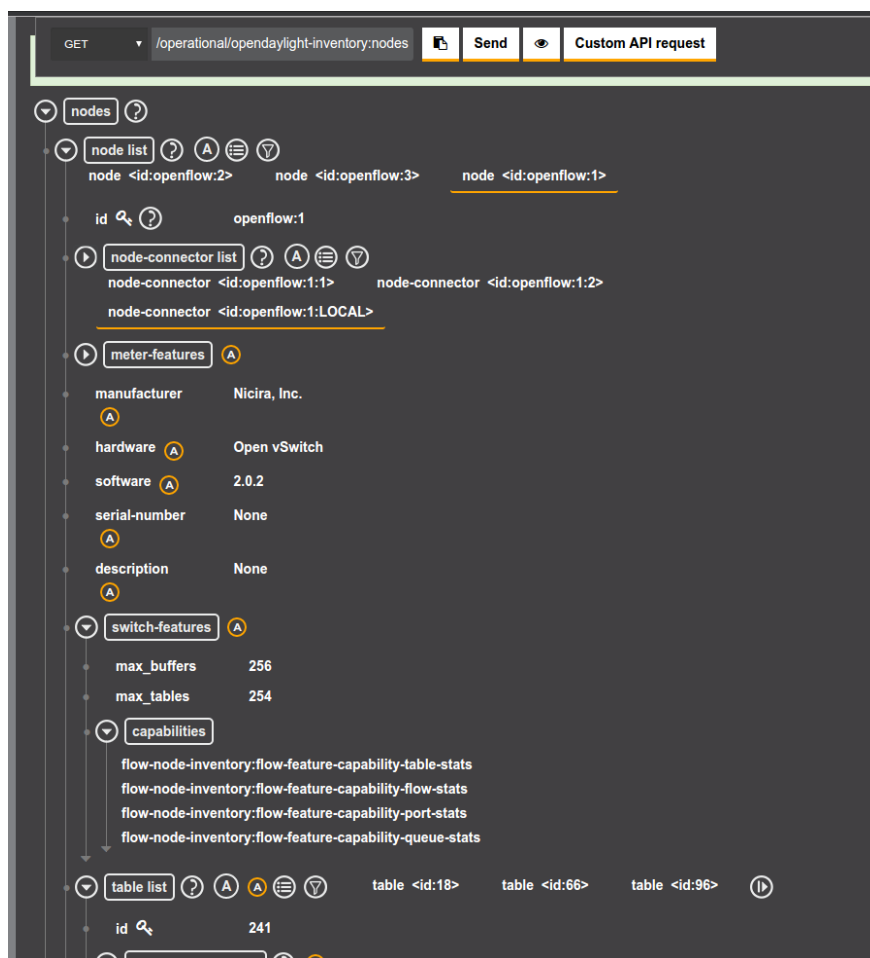
Το Yang User Interface είναι μια εφαρμογή διαχείρισης του OpenDaylight Controller και περιλαμβάνει μία σειρά λειτουργιών που καλύπτουν σχεδόν συνολικά το εύρος των δυνατοτήτων του ODL.

Ουσιαστικά πρόκειται για μια γλώσσα μοντελοποίησης που λειτουργεί εξυπηρετώντας τις λειτουργίες NETCONF πρωτοκόλλων. Είναι συγγενής της UML και της XML, με κύρια σημεία διαφοροποίησης μαζί τους την ευκολία κατανόησης από τους προγραμματιστές και της



παροχή μηχανισμών επικύρωσης στην διαμόρφωση σημασιολογίας και σύνταξης των μοντέλων δεδομένων που παράγει. Όσον αφορά την λειτουργία του στον OpenDaylight Controller έχει την δυνατότητα να αναγνωρίζει, επεξεργάζεται, λειτουργεί και να παρεμβαίνει σε όλα τα πεδία του Controller ενώ το περιβάλλον του είναι πολύ φιλικό σε παραμετροποιήσεις και αλλαγές για να καλυφθεί κάθε απαίτηση. Απευθύνεται κυρίως σε προγραμματιστές και ομάδες που αναπτύσσουν τον OpenDaylight Controller καθώς και εφαρμογές για αυτόν και έχει το πλεονέκτημα πως πάνω στην δομή του μπορούν εύκολα να αναπτυχθούν άλλες εφαρμογές χωρίς να γίνεται χρήση ενός ενδιάμεσου REST API.

Το κεντρικό του μενού περιλαμβάνει ένα πλήθος λειτουργιών που μπορεί να κάνει στον Controller ανά κατηγορία και μέσα στην κάθε κατηγορία τηρείται μια ιεραρχική κλίμακα που διευκολύνει κατά πολύ την ανάπτυξη προγραμματιστικών και αυτοματοποιημένων δομών που θα χρησιμοποιηθούν έπειτα στην ανάπτυξη τρίτων εφαρμογών.



Εικόνα 24 Πίνακας πληροφοριών σχετικά με Switch

Πέραν αυτού όμως το Yang UI είναι ένα πολύ αποτελεσματικό εργαλείο, με λεπτομερέστατη καταγραφή και εμφάνιση των λειτουργιών όχι μόνο του Controller αλλά και των στοιχείων που απαρτίζουν το δίκτυο που ο Controller ελέγχει. Έτσι μέσω αυτού μπορούμε να λαμβάνουμε πληροφορίες για Switches, Nodes,

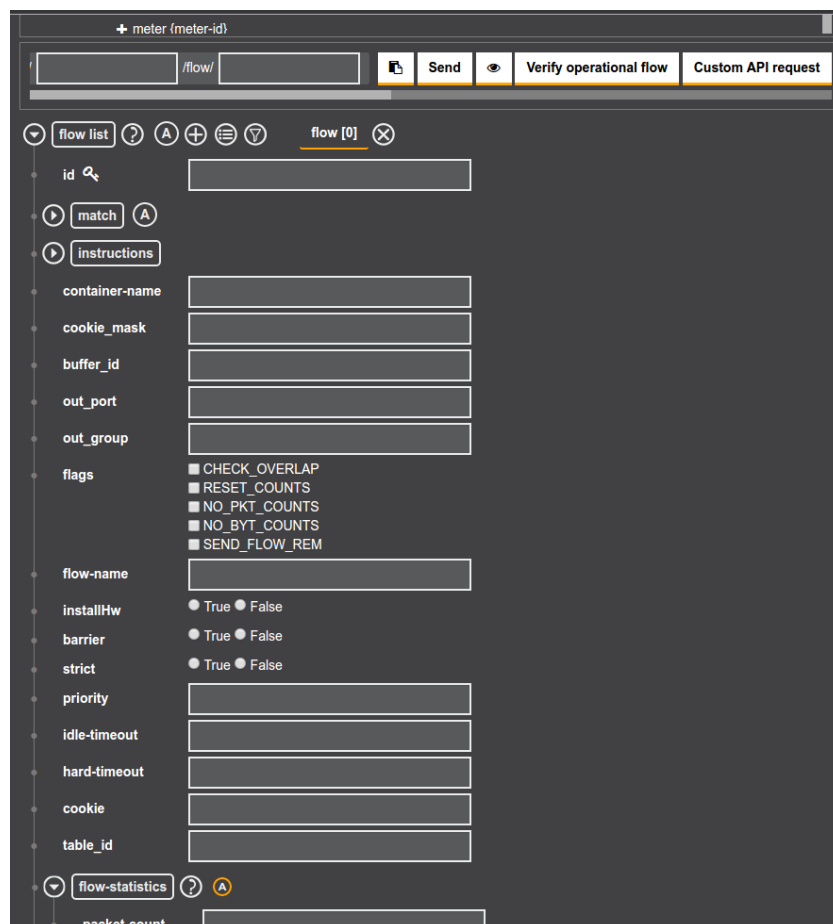
Connections όπως και για εσωτερικές λειτουργίες

τους όπως για παράδειγμα τα Flow Tables και τα Flow Entries τους.

Προσφέρει επίσης πεδίο παρέμβασης και σύνθεσης σε όλες τις λειτουργίες του μέσω της επιλογής Custom API Request που δίνει απευθείας πρόσβαση στον κώδικα που εκτελείται για την περάτωση της όποιας λειτουργίας, και μέσω της επεξεργασίας του κώδικα να κάνει κανείς πιο εξειδικευμένη ή εξατομικευμένη χρήση μιας λειτουργίας.

Δίνονται επίσης δυνατότητες φιλτραρίσματος των αποτελεσμάτων/αναζητήσεων, έως και λειτουργία επεξήγησης του κάθε πεδίου για την ευκολότερη και ακριβέστερη κατανόηση της πληροφορίας που προσφέρεται σε αυτό.

Όσον αφορά τις παρεμβάσεις στον Controller, το Yang UI αποτελεί μια ολοκληρωμένη πρόταση για την εισαγωγή στοιχείων και λειτουργιών, όπως για παράδειγμα η διαχείριση και δημιουργία Flow Entries και Flow Tables. Για παράδειγμα στην δίπλα εικόνα είναι εμφανές το περιβάλλον εισαγωγής Flow Entries και συγκεκριμένα για τις δράσεις που αναλαμβάνει το συγκεκριμένο Flow με δυνατότητα εισαγωγής πολλών στοιχείων, δράσεων και μηχανισμών εξακρίβωσης.



Εικόνα 25 Πεδίο εισαγωγής Flow Entries στο Yang UI

5.8.6 Wireshark – OpenFlow Captures

Το Wireshark είναι ένα πρόγραμμα (ίσως το δημοφιλέστερο) καταγραφής και ανάλυσης κίνησης σε επίπεδο πακέτων. Μέσα από αυτό το πρόγραμμα μπορούμε να παρατηρούμε τι είδους πακέτα/εντολές ανταλλάσσονται στο δίκτυο μεταξύ Controller και Switches.

Αυτό μπορεί να χρησιμεύσει για 3 λόγους:

- 1) Ασφάλεια: Για να εξακριβώνεται το τι ακριβώς πακέτα ανταλλάσσονται μεταξύ επιπέδου ελέγχου και υλικού και αν υπάρχει κάποια παραβίαση των κανόνων λειτουργίας του Controller.
- 2) Κατανόηση: Η εις βάθος μελέτη και κατανόηση του OpenFlow πρωτοκόλλου βοηθάται κατά πολύ από την γνώση των πακέτων και των μηνυμάτων που στέλνονται μέσα από δειγματοληψία κανονικής “κίνησης”.

3) Troubleshooting: Σε περίπτωση που υπάρχει κάποιου προβλήματος σε επίπεδο μεταφοράς πακέτων μπορεί να εντοπισθεί από που προέρχεται αυτή και ποιες οι λεπτομέρειες και τα αίτια του.

Στην παρούσα εργασία το Wireshark εγκαταστάθηκε στον Server που φιλοξενείται ο OpenDaylight Controller, όμως πρέπει να ληφθεί υπόψιν πως αν και η κανονική έκδοση του Wireshark υποστηρίζει κανονικά τα Openflow πρωτόκολλα (openflow_v1, openflow_v4, openflow_v5), η έκδοση που εγκαθίστανται σε Ubuntu Servers από τον package manager δεν υποστηρίζει το OpenFlow. Αυτή η δυσκολία μπορεί να ξεπεραστεί με το κατέβασμα του κώδικα και το compiling του προγράμματος στον Server του Controller που ενέχει επίσης ένα μικρό επίπεδο δυσκολίας για κάποιον μη εξοικειωμένο με το Linux Command Line.

Εφόσον γίνει επιτυχώς η εγκατάσταση του Wireshark γίνεται εκκίνηση του και ορίζεται το φίλτρο στο openflow_v4, καθώς περιλαμβάνει την 1.3 σταθερή έκδοση του OpenFlow. Στην συνέχεια ανοίγοντας το Mininet, δημιουργώντας ένα δίκτυο και συνδέοντας το με την IP του Controller ξεκινούν και καταγράφονται τα πακέτα που ανταλλάσσονται μεταξύ επιπέδου υλικού και επιπέδου ελέγχου.

Για να γίνει πιο εύκολα κατανοητό το περιεχόμενο της εικόνας πρέπει να αναφερθεί πως η IP Controller είναι 192.168.2.230 και η IP του Mininet είναι 192.168.2.229, επίσης υπό κανονικές συνθήκες όπου τα Switches θα ήταν πραγματικές συσκευές και όχι προσομοιώσεις, το καθένα θα είχε την ξεχωριστή του IP. Στην λήψη των πακέτων που εμφανίζονται στην παρούσα εργασία φαίνονται μόνο δυο IPs, αυτή του Controller και αυτή του Mininet. Μέσα στο οποίο περιλαμβάνεται ο όποιος αριθμός Switches τους οποίους όμως ο Controller αντιμετωπίζει ως διαφορετικές και ξεχωριστές οντότητες ταυτοποιώντας τον καθένα ξεχωριστά μέσω της του datapath_id που προσδίδει σε κάθε Switch που επικοινωνεί μαζί του..

The image shows two windows. The top window is Wireshark, displaying a packet capture on interface eth0. The filter is 'openflow_v4'. The packet list shows a series of OpenFlow messages between 192.168.2.229 and 192.168.2.230, including Hello, Features Request/Reply, Barrier Request/Reply, Role Reply, Multipart Request/Reply, and Set Config. The packet details pane shows the selected packet (No. 2928) as an OpenFlow 1.3 Hello message.

The bottom window is a terminal on a mininet VM. It shows the execution of the 'mn' command to create a network topology with 4 hosts (h1-h4) and 4 switches (s1-s4). The terminal output indicates that the network was successfully created and configured. A ping test was performed, showing 8% dropped packets (11/12 received).

Εικόνα 26 Τα πρώτα μηνύματα που αποστέλλονται μεταξύ Switches-Controllers

Ελέγχοντας τα πακέτα διαπιστώνουμε στην πράξη όσα προηγουμένως είχαμε δει θεωρητικά σχετικά με την λειτουργία του OpenFlow. Τα Switches επικοινωνούν πρώτα με τον Controller μέσω **Hello** μηνυμάτων στα οποία ο Controller απαντά επίσης με Hello. Κατά αυτόν τον τρόπο τα Switches ενημερώνουν τον Controller για την ύπαρξη τους καθώς και αναφέρουν ποια είναι η υψηλότερη έκδοση του OpenFlow πακέτου που μπορούν να υποστηρίξουν ώστε ο Controller να χρησιμοποιεί την συμβατή με αυτούς έκδοση.

Στην συνέχεια ο Controller ζητά από τα Switches να ενημερωθεί για τις δυνατότητες τους μέσω ενός **features request** στα οποία τα Switches απαντούν όπως θα δούμε παρακάτω με τις OpenFlow λειτουργίες που μπορούν αν υποστηρίξουν.

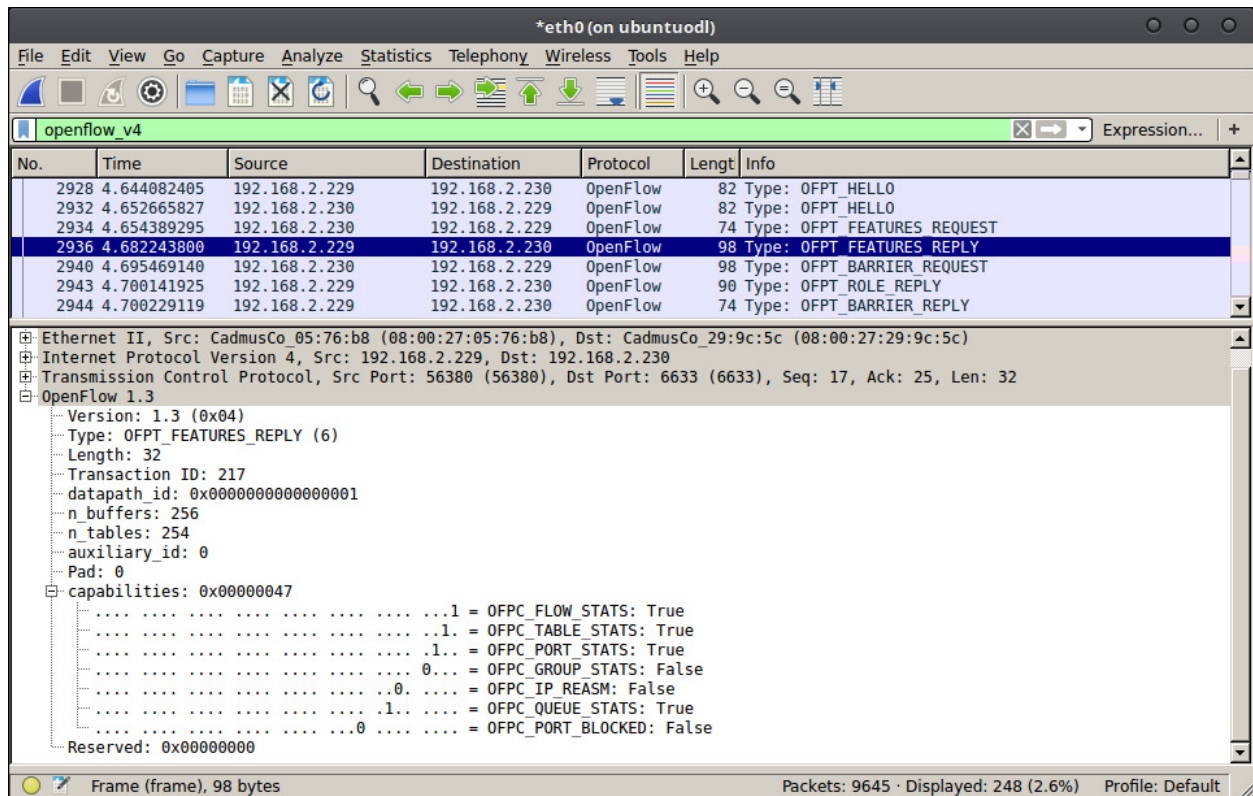
Τα **Barrier** μηνύματα αποστέλλονται από τον Controller ως εντολές συγχρονισμού, όπου όταν το Switch λαμβάνει ένα barrier request πρέπει να εκτελέσει το σύνολο των εντολών/λειτουργιών που του ανατέθηκαν νωρίτερα και όταν τελειώσει απαντά με ένα barrier reply μήνυμα.

Το **Role request/reply** είναι μηνύματα μέσω των οποίων ο Controller ζητά και ενημερώνεται από τα Switches αν είναι συνδεδεμένα ήδη με άλλον/άλλους Controllers και πληροφορίες για αυτούς.

Τα **Multipart** μηνύματα χρησιμοποιούνται κυρίως για την ερώτηση/απάντηση όσον αφορά σύνολα στατιστικών στοιχείων και πληροφοριών όπως ο αριθμός των Flow Tables, ποιοι πίνακες είναι σε χρήση, πόσα Flow Entries διαθέτουν κ

Τα **Port Description** (port_desc) μηνύματα περιέχουν γενικές πληροφορίες για το Hardware και το Software του Switch καθώς και πληροφορίες που αφορούν τις θύρες, ποιες είναι κατειλημμένες, τι λειτουργίες επιτελούν καθώς και τις ταχύτητες που υποστηρίζουν.

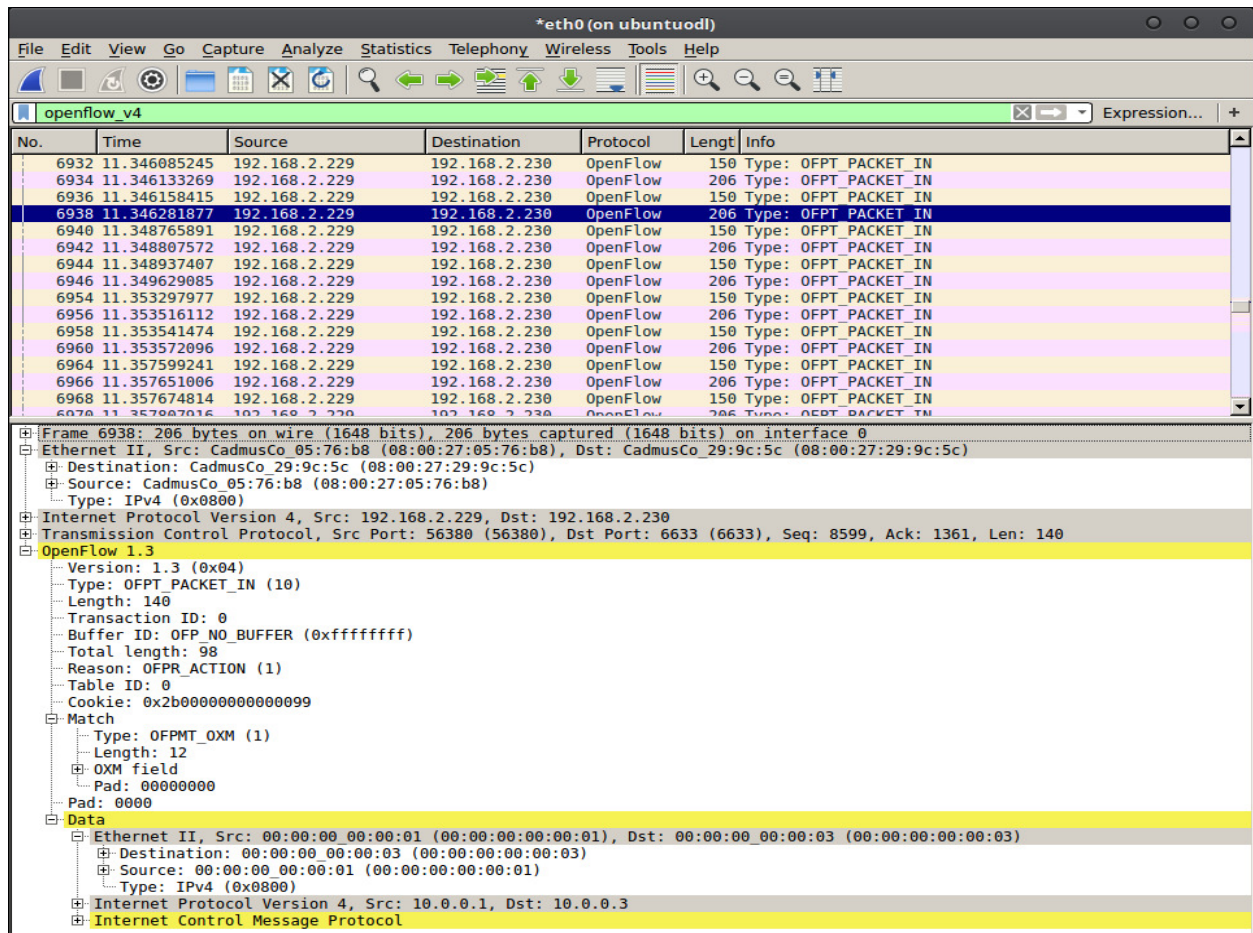
Στην συνέχεια αυτή η διαδικασία επαναλαμβάνεται για όλα τα Switches.



Εικόνα 27 Ένα αναλυτικό Features_Reply μήνυμα

Αν ανοιχθούν τα Feature Reply μηνύματα στο OpenFlow επίπεδο εμφανίζεται ένας πίνακας δυνατοτήτων που ενημερώνει τον Controller για το ποιες δυνατότητες του OpenFlow πακέτου μπορεί να υποστηρίζει. Για παράδειγμα στην πάνω εικόνα μπορούμε να δούμε πως το Switch υποστηρίζει έως 254 Flow Tables ($n_tables=254$) και στις δυνατότητες (capabilities) βλέπουμε πως υποστηρίζεται η τήρηση στατιστικών σχετικά με τα Flow Entries, Flow Tables, με τις θύρες και στατιστικές ουρών που μπορούν να φανούν χρήσιμες για Load Balancing λειτουργίες.

Αντίστοιχα υπάρχει μια σειρά δυνατοτήτων που φαίνεται πως δεν υποστηρίζονται στο συγκεκριμένο Switch.



Εικόνα 28 Ping πακέτα μεταξύ των Hosts/Υπολογιστών του δικτύου

Στην παραπάνω εικόνα η κίνηση σε επίπεδο πακέτων που προκαλείται από την ringall εντολή του Mininet. Συγκεκριμένα μέσω της εντολής αυτής κάθε H/Y κάνει broadcast rings τα οποία περνούν μέσα από τα Switches του Mininet τα οποία με την σειρά τους ενημερώνουν τον Controller για αυτές τις κινήσεις. Όπως φαίνεται το μήνυμα αυτό απεστάλη στον Controller ως **Packet In**, έχει IPv4 μορφή και έχει πηγή προέλευσης τον Host νούμερο 1 με MAC 00:00:00:00:00:01 και προορισμός είναι ο Host νούμερο 3 με MAC address 00:00:00:00:00:03, περνά μέσα από το 1^ο Flow Table (Table ID:0). Οι χρωματισμοί διαφέρουν κάνοντας έτσι πιο εύκολο να καταλάβουμε πως πρόκειται για αλληλουχία πακέτων Ping Request και Ping Reply.

Παρακάτω θα δούμε πως εύκολα με την χρήση εφαρμογών του Controller μπορούμε να απαγορεύσουμε τα rings μεταξύ Hosts του δικτύου επιλεκτικά και συγκεκριμένα χωρίς να διαταραχθεί η λειτουργία του δικτύου συνολικά.

5.9 Χρήση HP VAN SDN Controller

5.9.1 Εισαγωγή

Ο SDN VAN (Virtual Application Networks) Controller είναι ένας εμπορικός Controller της Hewlett Packard, εταιρία γνωστή για την παραγωγή υπολογιστών και περιφερειακών πληροφορικής έχει δραστηριότητα στον χώρο των δικτύων επίσης με μια επαγγελματική σειρά δικτυακού εξοπλισμού. Τα τελευταία χρόνια επιχειρεί με ιδιαίτερη ενεργητικότητα να μπει στον τομέα του Software Defined Networking αναπτύσσοντας και εξελίσσοντας συνεχώς τον Controller της, παρουσιάζοντας μάλιστα το πρώτο App Store στην ιστορία του SDN καθώς και αναπτύσσοντας καινούρια Switches με γνώμονα τις SDN λειτουργίες αλλά ανοίγοντας και OpenFlow συμβατότητες στα παλιά της Switches μέσω ενημερώσεων λογισμικού. (Hewlett-Packard, 2014)

Έως τώρα η HP έχει παρουσιάσει τις εξής βασικές εκδόσεις του Controller της κατά χρονολογική σειρά:

- VAN_SDN_Controller_v2.0 Νοέμβρης 2013
- VAN_SDN_Controller_v2.2 Μάρτιος 2014
- VAN_SDN_Controller_v2.3 Ιούλιος 2014
- VAN_SDN_Controller_v2.4 Νοέμβρης 2014
- VAN_SDN_Controller_v2.5 Μάιος 2015
- VAN_SDN_Controller_v2.6 Νοέμβρης 2015
- VAN_SDN_Controller_v2.7 Μάρτιος 2016

Ο Controller που χρησιμοποιήθηκε για την παρούσα εργασία ήταν αρχικά ο VAN_SDN_Controller_v2.7.16 που διατέθηκε τον Ιούλιο του 2016 και έπειτα ο VAN_SDN_Controller_v2.7.18 που είναι η πιο πρόσφατη προσθήκη στην σειρά της HP καθώς διατέθηκε τον Οκτώβρη του 2016. Στις τελευταίες εκδόσεις έχουν αλλάξει οι χρωματικοί συνδυασμοί, τα σχήματα και ο τρόπος με την οποία γίνεται απεικόνιση της τοπολογίας δικτύων. Ανανέωση έχει υπάρξει επίσης και στον ισότοπο-AppStore στον οποίον διατίθενται οι εφαρμογές του Controller.

Η HP έχει πλαισιώσει την πρόταση της όσον αφορά το SDN σε επίπεδο υλικού, Controller αλλά και Εφαρμογών επαναπρογραμματίζοντας ήδη υπάρχοντα εμπορικά της προγράμματα που λειτουργούν σε περιβάλλοντα παραδοσιακών δικτύων και μεταφέροντας τα SDN App Store της παρουσιάζει ένα ολοκληρωμένο SDN οικοσύστημα, ίσως από τα μεγαλύτερα του χώρου.

Λαμβάνοντας υπόψιν τα παραπάνω μπορούμε να πούμε πως αξίζει να πειραματιστεί αλλά και να ασχοληθεί σοβαρά κάποιος με την πρόταση της HP, καθώς είναι εμφανές πως η παρουσία της θα είναι διαρκής στο Software Defined Networking. Αν και πρόκειται για εμπορικά προγράμματα όπως είπαμε, η HP διαθέτει τον Controller της και τα προγράμματα της με περίοδο δοκιμής δυο μηνών ώστε να ενθαρρύνει προγραμματιστές και καταναλωτές να δοκιμάσουν τις προτάσεις της.

5.9.2 Είσοδος

Μετά την εγκατάσταση του Controller γίνεται εκκίνηση του Server και είσοδος. Ο Controller ξεκινά αυτομάτως την λειτουργία του μετά το Log In όμως είναι δυνατόν να σταματά και να ξεκινά κατόπιν εντολής:

```
~$ sudo service sdn start/stop
```

Και για να εμφανιστεί η κατάσταση του Controller

```
~$ sudo service sdn status
```

Ο Controller είναι προσβάσιμος μέσω Browser από την διεύθυνση

<https://localhost:8443/api>

Ο Browser πιθανότατα θα εμφανίσει ενημέρωση για μη ασφαλή διεύθυνση καθώς ο Controller χρησιμοποιεί μια δική του πιστοποίηση ασφαλείας (self-signed certification).

Το όνομα χρήστη και ο κωδικός είναι εκ προεπιλογής sdn και skyline αντίστοιχα, τα στοιχεία εισόδου είναι ίδια για το web interface αλλά και για την είσοδο στην εγκατάσταση στον Server. Συνίσταται βεβαίως σε περιβάλλοντα επιχειρησιακής χρήσης οι κωδικοί και τα στοιχεία να αλλάζουν για λόγους ασφαλείας.

5.9.3 Περιβάλλον

Στην αρχική σελίδα εμφανίζονται μια σειρά από επιλογές/λειτουργίες του Controller οι οποίες είναι αισθητά περισσότερες συγκριτικά με τον OpenDaylight. Βλέπουμε λοιπόν

τις γενικές λειτουργίες όπως αυτή των ειδοποιήσεων, των εφαρμογών, των ρυθμίσεων, των Logs/αρχείου Εξαρχής λοιπόν υπάρχουν εγκατεστημένες οι εφαρμογές για την εμφάνιση της τοπολογίας του δικτύου, εμφάνισης των στοιχείων του δικτύου και σχετικών πληροφοριών, εφαρμογή αντίστοιχη του Wireshark για την καταγραφή της κίνησης των πακέτων ενημέρωσης μέσω notifications για αλλαγές/δυσλειτουργίες και άλλα. Στην καρτέλα του OpenFlow Monitor είναι διαθέσιμα τα flows των switches μαζί με μια σειρά πληροφοριών όπως το Table στο οποίο βρίσκονται οι προτεραιότητες των πακέτων, στατιστικά στοιχεία, Flow Entries κανόνες με τις αντίστοιχες δράσεις και το ID των Flow ομάδων.

5.9.4 Τοπολογία

Όσον αφορά την τοπολογία αυτή είναι εύκολη στην ανάγνωση και κατανόηση μέσω του Controller που επιτρέπει στην αλλαγή παραμέτρων του σχεδιαγράμματος και την λήψη πληροφοριών μέσω του hover του ποντικιού πάνω από μια συσκευή, Switch ή Host.

The screenshot displays the HP VAN SDN Controller interface. The main window shows the 'General / OpenFlow Topology' with a network diagram. A red arrow points to a host node labeled '10.0.0.3'. A green arrow points to a host node labeled '10.0.0.1'. The diagram shows several switches (green circles) and hosts (red circles) connected in a network topology. A legend on the right identifies 'Switch' (green square), 'Collapsed Switch' (light green square), and 'End Host' (red square). Below the diagram is a 'Switch Details' table with columns for Priority, Bytes, Packets, In P., Out Port, Src. MAC, Dst. MAC, Src. IP, Dst. IP, Action, Eth. Type, Prot., Age, TIO, and Idle TIO. The table contains several rows of data. To the right of the main window is a terminal window showing the output of the 'show' command, displaying details for various bridges (s3, s1, s2, s4, s5) and their controllers, ports, and interfaces.

Priority	Bytes	Packets	In P.	Out Port	Src. MAC	Dst. MAC	Src. IP	Dst. IP	Action	Eth. Type	Prot.	Age	TIO	Idle TIO
60000	154	2	4294967293						8099		429	0	0	
31000	356	8	4294967293						0806		429	0	0	
34000	588	14	4294967293						0806		428	0	0	
35000	0	0	4294967293						0800	17	429	0	0	
35500	0	0	4294967293						0800	17	429	0	0	
0	3108	37	4294967293								429	0	0	

Εικόνα 29 Τοπολογία Mininet σε HP VAN Controller

Όπως και στον OpenDaylight Controller τα στοιχεία του δικτύου μπορούν να μετακινηθούν, να γίνει zoom in/out όμως προσφέρονται μια σειρά λειτουργιών που εξαρχής δεν υπάρχουν στον ODL. Για παράδειγμα ο ορισμός ενός σημείου εκκίνησης

και τερματισμού μεταξύ Hosts μαζί με την διαδρομή που θα ακολουθηθεί για την επικοινωνία των δυο, με την επιλογή ενός Switch δίνεται η δυνατότητα εμφάνισης Flow Entries κατευθείαν στην σελίδα της τοπολογίας μέσω ενός μικρότερου παραθύρου όπως και η λειτουργία αναζήτησης με την οποία μπορεί να γίνει εύρεση διαφόρων παραμέτρων των στοιχείων της τοπολογίας.

5.9.5 Monitor

Στην Monitor επιλογή εμφανίζονται τα Switches που υπάρχουν στο δίκτυο με διάφορες επιλογές που παράγουν αντίστοιχος πίνακες, όπως τις γενικές πληροφορίες μιας συσκευής, την διαρρύθμισης και την κατάσταση των θυρών της, τα Flow Entries καθώς και των ομάδων Switches που μπορούν να οριστούν όπως είδαμε όταν αναφερόμασταν στους Group Tables που φιλοξενούνται στην OpenFlow εγκατάσταση του Switch.

Κάθε κατηγορία έχει λεπτομερειακή καταγραφή των πληροφοριών και είναι βάσιμο να υποστηρίξει κανείς πως και η συγκεκριμένη παράμετρος του HP VAN Controller υπερτερεί έναντι του OpenDaylight.

Τα δεδομένα που αφορούν τα Flow Tables παρουσιάζονται λεπτομερειακά με τον αριθμό του πίνακα στον οποίον φιλοξενείται το Flow, την προτεραιότητα του πακέτου, τον αριθμό των πακέτων που έχουν καταμετρηθεί, το μέγεθος των πακέτων που έχουν περάσει από το συγκεκριμένο Flow σε μέγεθος Bytes και το Flow Class ID.

Σημαντικότερα χαρακτηριστικά ωστόσο είναι η εμφάνιση των Match πεδίων, δηλαδή της αντιστοίχισης σε κάποια κατηγορία που πρέπει να πληροί το πακέτο ώστε να ληφθεί η αντίστοιχη δράση. Η δράση/οδηγία είναι το επόμενο πεδίο αυξημένης σημαντικότητας όπου εμφανίζονται οι δράσεις που αναλαμβάνονται όσον αφορά την πορεία του πακέτου, αν δηλαδή θα σταλεί στον Controller, αν θα προωθηθεί στον επόμενο πίνακα, θα προωθηθεί στον προορισμό του ή θα διαγραφεί.

Όπως φαίνεται και στην εικόνα τα στοιχεία που δίνει ο Controller είναι πλήρως εναρμονισμένα με τα στοιχεία που εμφανίζει ο Mininet όταν του ζητηθεί να εμφανίσει τα Flow Entries για το Switch νούμερο δυο. Υπόψιν πρέπει να ληφθεί πως το cookie που εμφανίζεται πριν τα Flow Entries είναι το ID τους, τα στοιχεία ταυτοποίησης των Entries και δεν εμφανίζονται στον πίνακα του Controller.

The screenshot displays the HP VAN SDN Controller interface. The top navigation bar shows 'HPE VAN SDN Controller' and '40 sdn'. The left sidebar contains various menu items, with 'OpenFlow Monitor' highlighted. The main area shows 'Flows for Data Path ID: 00:00:00:00:00:00:02' with a table of flow entries. Below the table is a terminal window showing Mininet CLI commands and their output.

Table ID	Priority	Packets	Bytes	Match	Actions/Instructions	Flow Class ID
0	60000	4	268	eth_type: bddp	apply_actions: output: CONTR	com.hp.sdn.bddp.st...
0	34000	8	336	in_port: 2 eth_type: arp	apply_actions: output: NORM	com.hp.sdn.arp.filter
0	34000	15	630	in_port: 3 eth_type: arp	apply_actions: output: NORM	com.hp.sdn.arp.filter
0	31500	0	0	eth_type: ipv4 ip_proto: udp udp_src: 68 udp_dst: 67	apply_actions: output: CONTR output: NORM	com.hp.sdn.dhcp.co...
0	31500	0	0	eth_type: ipv4 ip_proto: udp udp_src: 67 udp_dst: 68	apply_actions: output: CONTR output: NORM	com.hp.sdn.dhcp.co...
0	31000	8	336	eth_type: arp	apply_actions: output: CONTR output: NORM	com.hp.sdn.arp.copy
0	0	53	4694		apply_actions: output: NORM	com.hp.sdn.normal

```

mininet@mininet-vm: ~
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> h1 h2 h3 h5
h5 -> h1 h2 h3 h4
*** Results: 0% dropped (20/20 received)
mininet> sh ovs-ofctl -O OpenFlow13 dump-flows s2
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0xffff0000faded000, duration=160.168s, table=0, n_packets=4, n_bytes=268, priority=60000,dl_type=0x8999 actions=CONTROLLER:65535
 cookie=0xfffd0000babadada, duration=160.168s, table=0, n_packets=8, n_bytes=336, priority=31000,arp actions=CONTROLLER:65535,NORMAL
 cookie=0xffff90000000000000, duration=158.283s, table=0, n_packets=8, n_bytes=336, priority=34000,arp,in_port=2 actions=NORMAL
 cookie=0xffff90000000000000, duration=158.244s, table=0, n_packets=15, n_bytes=630, priority=34000,arp,in_port=3 actions=NORMAL
 cookie=0xfffc0000babadada, duration=160.168s, table=0, n_packets=0, n_bytes=0, priority=31500,udp,tp_src=68,tp_dst=67 actions=CONTROLLER:65535,NORMAL
 cookie=0xfffc0000babadada, duration=160.168s, table=0, n_packets=0, n_bytes=0, priority=31500,udp,tp_src=67,tp_dst=68 actions=CONTROLLER:65535,NORMAL
 cookie=0xffff000000000000, duration=160.168s, table=0, n_packets=53, n_bytes=4694, send_flow_rem priority=0 actions=NORMAL
mininet>

```

Εικόνα 30 Τα FlowTables όπως εμφανίζονται στον HP VAN Controller και στο Mininet αντίστοιχα

5.9.6 Trace

Η Trace λειτουργία του HP VAN Controller είναι ένας μηχανισμός καταγραφής και θέασης της κίνησης σε επίπεδο πακέτων και εκτελεί την ίδια λειτουργία που εκτελέστηκε πριν μέσω του Wireshark. Το Trace αντίθετα με το Wireshark είναι ένα πολύ πιο απλουστευμένο πρόγραμμα που προσφέρει πολύ λιγότερες δυνατότητες εμβάθυνσης στο είδος και στις επιμέρους λεπτομέρειες των πακέτων. Επίσης η καταγραφή περιορίζεται χρονικά οπότε δεν είναι δυνατή η καταγραφή μεγάλου όγκου κίνησης για ανάλυση, περιορισμός που δεν υπάρχει σε λιγότερο βαθμό στο Wireshark.

Στον αντίποδα όμως το Trace είναι ένα πρόγραμμα που έρχεται ενσωματωμένο στον Controller και δεν απαιτεί παραπάνω εγκαταστάσεις και compiles όπως το Wireshark ούτε παραμετροποιήσεις και άνοιγμα ξεχωριστών παραθύρων για να εκτελεστεί.

Είναι ένα από πρόγραμμα που κάνει πολύ καλά την δουλειά του όσον αφορά την θέαση της κίνησης στο δίκτυο και την αντιμετώπιση προβλημάτων (troubleshooting). Καθώς είναι ανεπτυγμένο συγκεκριμένα για το OpenFlow πρωτόκολλο παρουσιάζει μια πολύ πιο περιεκτική και στοχευμένη εικόνα της λειτουργίας του δικτύου στο επίπεδο επικοινωνίας του Controller με τα Switches.

The screenshot displays the HPE VAN SDN Controller interface. The main window shows a 'General / OpenFlow Trace' with columns for Time, Event, Data Path ID, and Message. The trace shows the following sequence of events:

- 00:00:00.0000000001**: CkPt to Rx: Datapath (null) -- Connected
- 00:00:00.0000000001**: Rx to CkPt: [ofm(V_1_0)HELLO.8.7]
- 00:00:00.0000000001**: CkPt to Rx: [ofm(V_1_0)FEATURES_REPLY.77a905]
- 00:00:00.0000000001**: Rx to CkPt: [ofm(V_1_0)SET_CONFIG.12.906]
- 00:00:00.0000000001**: CkPt to Rx: [ofm(V_1_0)MULTIPART_REPLY.10a8.907]
- 00:00:00.0000000001**: Rx to CkPt: [ofm(V_1_0)HELLO.8.8]
- 00:00:00.0000000001**: CkPt to Rx: [ofm(V_1_0)FEATURES_REPLY.224.921]
- 00:00:00.0000000001**: Rx to CkPt: [ofm(V_1_0)SET_CONFIG.12.922]
- 00:00:00.0000000001**: CkPt to Rx: [ofm(V_1_0)MULTIPART_REPLY.10a8.923]

The terminal window on the left shows the Mininet CLI commands and their output, including network creation, host addition, and controller startup.

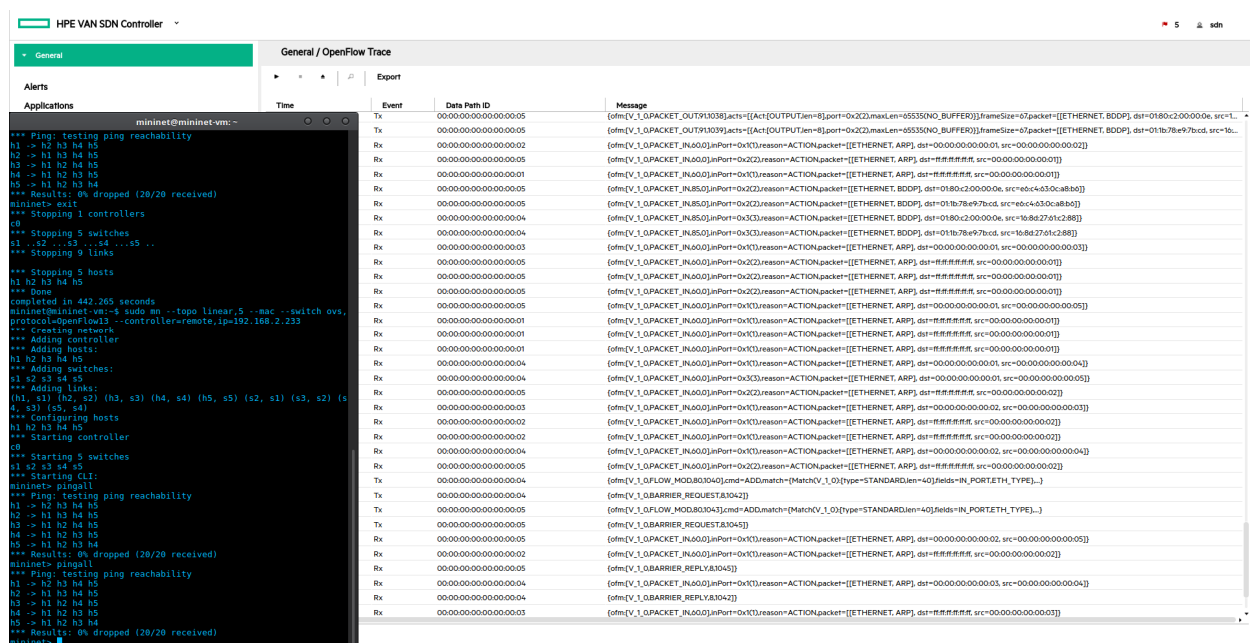
Εικόνα 31 Καταγραφή των πακέτων που ανταλλάσσονται μεταξύ Controller-Switches στα πρώτα δευτερόλεπτα σύνδεσης

Όπως φαίνεται στην εικόνα η στο αρχικό στάδιο της σύνδεσης ο Controller εγκαθιδρύει την επικοινωνία του με το δίκτυο του Mininet στην IP 192.168.2.229 το οποίο στην 3^η γραμμή ενημερώνει για την ύψιστη έκδοση του OpenFlow που υποστηρίζει (1.3) και έπειτα ακολουθεί η σύνδεση με τα Switches που αυτό φιλοξενεί. Στην 5^η σειρά βλέπουμε το πακέτο που στέλνεται από το πρώτο Switch (**Rx**) με τον Controller περιέχοντας την έκδοση του OpenFlow που το Switch υποστηρίζει (**V_1_0**) και του χαιρετισμού του και αντίστοιχα ο Controller (**Tx**) απαντά επίσης με **Hello** και συμφωνεί στην χρήση της έκδοσης 1.0 του OpenFlow για την επικοινωνία μεταξύ τους.

Στην συνέχεια βλέπουμε τις ανταλλαγές πακέτων για την γνωστοποίηση των **Features** του Switch στον Controller και έπειτα μέσω μια **Multipart** ερώτησης ο Controller ζητά

να μάθει περισσότερες πληροφορίες για το Switch και αυτό απαντά με διάφορες πληροφορίες όπως τον κατασκευαστή του, τον τύπο του και άλλα γνωρίσματα του.

Ιδιαίτερο ενδιαφέρον έχει το σημείο μετά την 13^η σειρά όπου βλέπουμε τον Controller να στέλνει τα Flow Entries στον Switch μέσω **FLOW_MOD** μηνυμάτων και έπειτα με ένα **BARRIER_REQUEST** μήνυμα του ζητά να επιβεβαιώσει την εφαρμογή των προηγούμενων εντολών για εγγραφή των Flow Entries μόλις αυτές ολοκληρωθούν. Στη συνέχεια ο Controller προχωρά στα επόμενα Switches με ακολουθώντας την αντίστοιχη σειρά βημάτων ενώ μόλις ολοκληρώσει την εγγραφή των Flow Entries ο πρώτος Switch απαντά με ένα **BARRIER_REPLY** μήνυμα για να ειδοποιήσει τον Controller ότι ολοκλήρωσε επιτυχώς της διαδικασία οπότε και η εσωτερική λειτουργία του Controller θεωρεί τον 1^ο Switch λειτουργικό και έτοιμο για χρήση.



Εικόνα 32 Η ανταλλαγή πακέτων που εκτελείται λόγω pingall

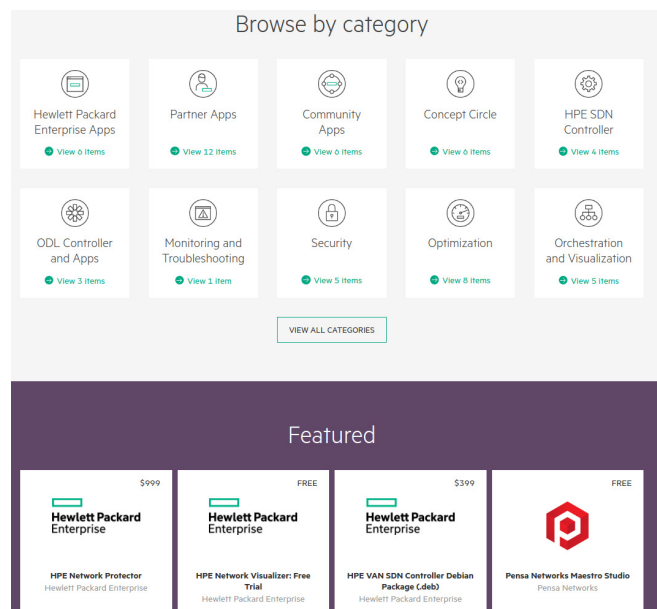
Εκτελώντας την pingall εντολή στο Mininet βλέπουμε πως ο Controller ενημερώνεται συνεχώς για τα πακέτα που ανταλλάσσουν τα Switches μεταξύ τους στην διαδικασία των ping χωρίς να παρεμβαίνει καθώς πρόκειται για κίνηση που υπόκειται σε κανόνες που ήδη έχουν τεθεί. Έχει όμως ενδιαφέρον να παρατηρηθεί πιο λεπτομερικά το περιεχόμενο των πακέτων του ping, βλέπουμε λοιπόν πως πρόκειται για ARP πακέτα που έχουν εκτελούν broadcast λειτουργία (dst ff:ff:ff:ff:ff:ff) και αντίστοιχα την MAC διεύθυνση του Host που τα στέλνει και αντίστοιχα απάντηση όπως και πληροφορίες που

αφορούν τις θύρες του Switch από όπου περνά το πακέτο και άλλες παραμέτρους επίσης.

Εκτός της καταγραφής και της θέασης των πακέτων προσφέρεται και η λειτουργία του Export δηλαδή δυνατότητα εξαγωγής και αποθήκευση της καταγραφής για ανάλυση σε τρίτο χρόνο ή για λόγους ασφάλειας και αρχειοθέτησης. Με την επιλογή της η καταγραφή γίνεται download μέσω το Browser σε zip μορφή.

5.9.7 Applications

Ο HP VAN SDN Controller έχει ένα σύνολο εφαρμογών που εκτείνονται σε 3 κατηγορίες που καλύπτουν κάθε πτυχή της λειτουργία του Controller, από την ασφάλεια, την διαχείριση, την οπτικοποίηση του δικτύου έως εφαρμογές πειραματισμού και εκμάθησης του Controller. Το θετικό με το App Store της HP είναι πως δεν διαθέτει μόνο εταιρικά της προγράμματα στο site της αλλά το έχει "ανοιχτό" σε συνεργάτες της όπως και σε τρίτους προγραμματιστές και κοινότητες που ασχολούνται με το SDN και γράφουν εφαρμογές συμβατές με τον HP Controller. Αν και οι εφαρμογές της ίδιας της HP απαιτούν ένα

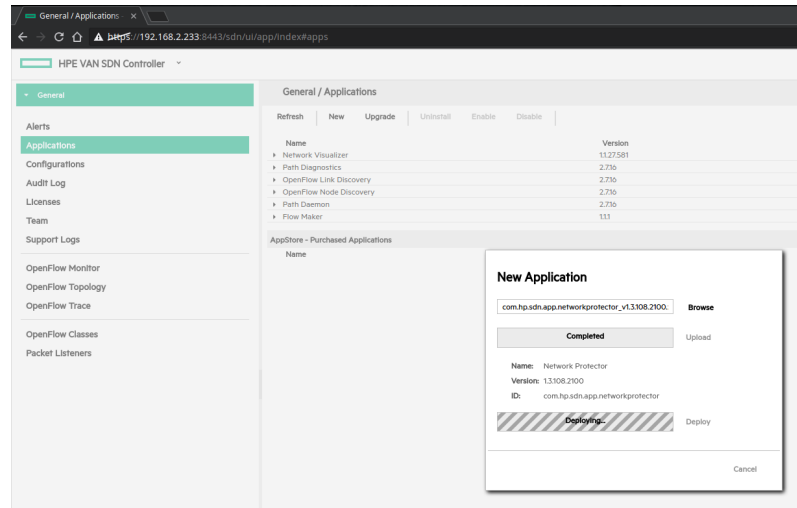


Εικόνα 33 Κατηγορίες προγραμμάτων στο AppStore της HP

χρηματικό αντίτιμο όμως διατίθεται δωρεάν για μια δοκιμαστική περίοδο ώστε να αξιολογηθούν πριν την αγορά τους. Οι εταιρικές εφαρμογές που πουλά η HP είναι καλογραμμένες, με λιγότερη κατανάλωση όσον αφορά την RAM και καλύτερη απόδοση στις εργασίες τους συγκριτικά με τρίτες εφαρμογές πράγμα που τις κάνει ανταγωνιστικές παρά το αντίτιμο που απαιτούν.

Η εγκατάσταση εφαρμογών είναι ιδιαίτερα εύκολη στον Controller, αρχικά εντοπίζεται η εφαρμογή που ενδιαφέρει, επιλέγεται το Download, και κατεβαίνει ως Zip αρχείο, το οποίο αρχείο μετά το unzip παράγει ένα ακόμα zipαρισμένο αρχείο που εγκαθίσταται

μέσω του browser. Πηγαίνοντας στην επιλογή Applications του Controller, επιλέγεται το New και εμφανίζεται ένα παράθυρο στο οποίο δίνεται το path του αρχείου, γίνεται το Upload και έπειτα η εγκατάσταση του στον Controller όπου γίνεται διαθέσιμο για λειτουργία.



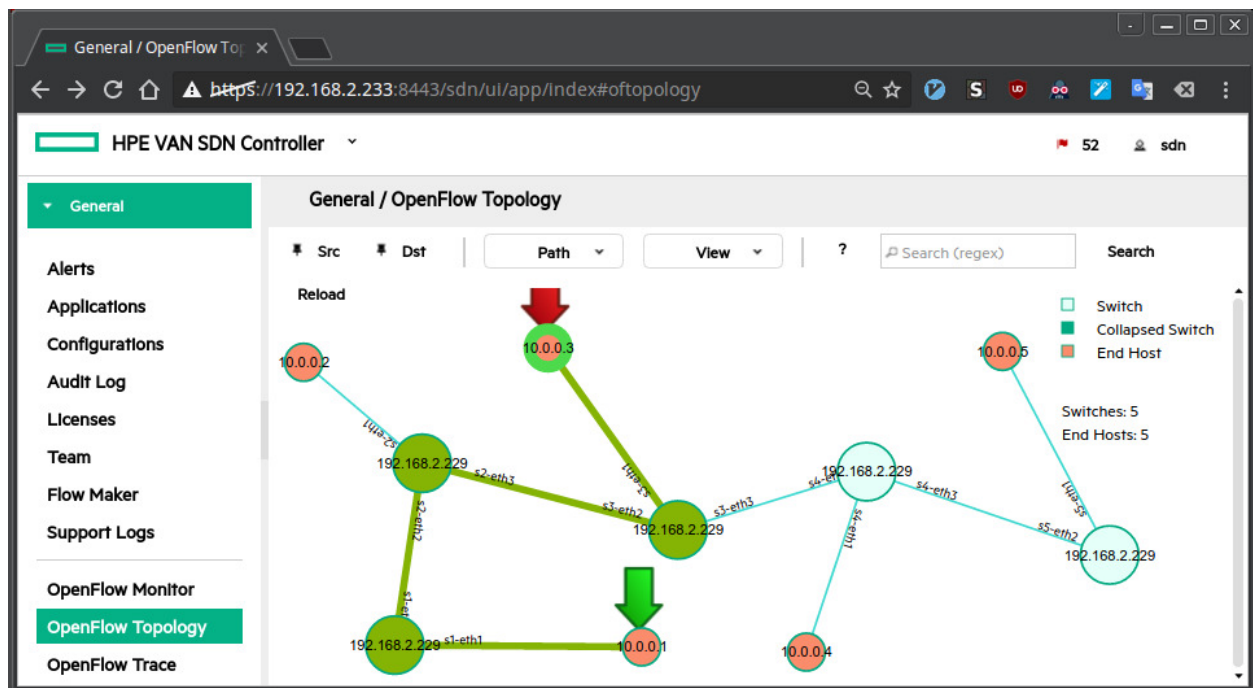
Εικόνα 34 Εγκατάσταση εφαρμογής

5.9.8 Flow Maker

Το Flow Maker είναι μια εφαρμογή που επιτρέπει εύκολες παρεμβάσεις στα Flow Entries δια του περιβάλλοντος του Controller και προσφέρει μια ολοκληρωμένη λύση όσον αφορά το αυτό το κομμάτι της διαχείρισης του δικτύου. Η εφαρμογή αναπτύχθηκε από την εταιρία Northbound Networks αποκλειστικά για τον Controller της HP, και υπήρχε διαθέσιμη στο AppStore της HP όμως αφαιρέθηκε και είναι διαθέσιμη πλέον μόνο από τον ιστότοπο (<https://northboundnetworks.com/collections/flowmaker>) της εταιρίας ανάπτυξης του. Διατίθεται μια δωρεάν έκδοση και αντίστοιχα μια επί πληρωμή που ζητά ένα μικρό αντίτιμο όμως προσφέρει της δυνατότητα εκτός της δημιουργίας Flow Entries, να γίνεται αποθήκευση τους και την αυτόματη επαναφορά τους σε περίπτωση reboot του Controller.

Μέσω της δημιουργία Flow Entries ανοίγεται μια σειρά δυνατοτήτων όσον αφορά την διαχείριση του δικτύου θέτοντας κανόνες με διάφορες παραμέτρους, όπως για παράδειγμα τον περιορισμό της προτεραιότητας στη κίνηση ενός συγκεκριμένου υπολογιστή του οποίου οι λειτουργίες δεν είναι σημαντικές ώστε να εξοικονομηθεί Bandwidth για πιο σημαντικές λειτουργίες ή άλλους υπολογιστές.

Μια άλλη εφαρμογή που θα δοκιμαστεί στην πράξη από το παρόν πείραμα είναι να αποκόψουμε την σύνδεση μεταξύ δυο υπολογιστών συγκεκριμένα χωρίς όμως να περιορίσουμε την επικοινωνία τους από το υπόλοιπο δίκτυο, για παράδειγμα στην παρακάτω τοπολογία έχουν επιλεγθεί δυο υπολογιστές, ο host1 και ο host3 των οποίων θέλουμε να εμποδίσουμε την επικοινωνία μεταξύ τους. Από το σχήμα βλέπουμε πως μεταξύ τους παρεμβάλλονται 3 Switches.



Δοκιμάζουμε να κάνουμε ping τους δυο υπολογιστές στο Mininet και βλέπουμε πως τα αποτελέσματα είναι πολύ καλά και πως υπάρχει άριστη επικοινωνία ανάμεσα τους.

```

mininet@mininet-vm: ~
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.206 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.096 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.065/0.122/0.206/0.061 ms
mininet>

```

Στη συνέχεια επιλέγουμε έναν ενδιαμέσο στους δυο υπολογιστές Switch, στην προκειμένη για παράδειγμα επέλεξα μέσω Flow Maker το 3^ο Switch και στα πεδία εισαγωγής επιλέγω τον πρώτο πίνακα (Flow Table) με τον αριθμό 0, καθώς εκεί ξεκινά η αρίθμηση του, επίσης δίνω μια υψηλή προτεραιότητα στο Flow Entry με μηδενικό Idle και Hard Timeouts ώστε το Entry να μην διαγραφεί μετά από μια περίοδο αδράνειας. Επίσης στο πεδίο των αντιστοιχίσεων και συγκεκριμένα το πεδίο της φυσικής διεύθυνσης αποστολέα, το MAC του πρώτου υπολογιστή, δηλαδή την 00:00:00:00:01 και στο πεδίο του προορισμού την MAC διεύθυνση του άλλου

υπολογιστή, δηλαδή την 00:00:00:00:03. Στην συνέχεια ορίζω τα πεδία των πρωτοκόλλων σε κανονικής μορφής κίνησης και αφήνω κενά τα πεδία των δράσεων και επιλέγω Add (προσθήκη)

HPE VAN SDN Controller

General

Alerts

Applications

Configurations

Audit Log

Licenses

Team

Flow Maker

Support Logs

OpenFlow Monitor

OpenFlow Topology

OpenFlow Trace

OpenFlow Classes

Packet Listeners

Add Flow to Data Path ID: 00:00:00:00:00:00:03

Northbound NETWORKS | Clear | Add

Metadata

Table ID: 0 Idle Timeout: 0

Priority: 10000 Hard Timeout: 0

Match

Source MAC: 00:00:00:00:00:01 Dest. MAC: 00:00:00:00:00:03

Source IP: Dest. IP:

Source Netmask: Dest. Netmask:

Source Port: Dest. Port:

VLAN ID: In Port:

Protocol

IP Protocol: ICMP

Ethernet Type: IPv4

Actions

Action 1: No Action Value:

Action 2: No Action Value:

Αν στα πεδία των δράσεων επέλεγα την αποστολή ερώτησης προς τον Controller ή την προώθηση στο επόμενο Table και άρα την προώθηση του πακέτου η επικοινωνία θα συνεχιζόταν όπως πριν. Μη συμπληρώνοντας τα ο Controller δεν ξέρει τι να κάνει αυτήν την κίνηση και την διαγράφει. Έτσι η επικοινωνία μεταξύ των δυο υπολογιστών διακόπτεται λόγω του Flow Entry στο 3^ο Switch δημιουργώντας με πολύ απλό τρόπο ένα αποτελεσματικό φραγμό.

Πηγαίνοντας στον πίνακα των Flow Entries βλέπουμε προτελευταίο το Flow που δημιουργήσαμε με τις λεπτομέρειες του.

The screenshot displays the HPE VAN SDN Controller interface. The top navigation bar shows 'HPE VAN SDN Controller' and a status indicator '52 sdn'. The left sidebar contains various menu items, with 'Flow Maker' highlighted. The main area is titled 'Flows for Data Path ID: 00:00:00:00:00:00:03' and features a table of flow rules. The table includes columns for Table ID, Priority, Packets, Bytes, Match Fields, Actions, and Flow Class ID. Below the table, a terminal window shows the results of ping commands between hosts h1, h2, h3, and h4.

Table ID	Priority	Packets	Bytes	Match Fields	Actions	Flow Class ID
0	60000	4	268	eth_type: bddp	apply_actions: output: CONTROLLER	com.hp.sdn.bddp.steal
0	34000	39	1638	in_port: 2 eth_type: arp	apply_actions: output: NORMAL	com.hp.sdn.arp.filter
0	34000	16	672	in_port: 3 eth_type: arp	apply_actions: output: NORMAL	com.hp.sdn.arp.filter
0	31500	0	0	eth_type: ipv4 ip_proto: udp udp_src: 68 udp_dst: 67	apply_actions: output: CONTROLLER output: NORMAL	com.hp.sdn.dhcp.copy
0	31500	0	0	eth_type: ipv4 ip_proto: udp udp_src: 67 udp_dst: 68	apply_actions: output: CONTROLLER output: NORMAL	com.hp.sdn.dhcp.copy
0	31000	31	1302	eth_type: arp	apply_actions: output: CONTROLLER output: NORMAL	com.hp.sdn.arp.copy
0	10000	0	0	eth_src: 00:00:00:00:00:01 eth_dst: 00:00:00:00:00:03 eth_type: ipv4		
0	0	133	11746	ip_proto: icmp	apply_actions: output: NORMAL	com.hp.sdn.normal

```

mininet@mininet-vm: ~
mininet> h1 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
^C
--- 10.0.0.3 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5040ms

mininet> h1 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.639 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.080 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.063 ms
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.063/0.260/0.639/0.268 ms

mininet> h3 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.436 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.447 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.066 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms

```

Για να ελέγξουμε πως όντως οι υπολογιστές δεν επικοινωνούν μεταξύ τους εκτελούμε ένα ping πάλι από τον h1 στον h3 και βλέπουμε πως δεν υπάρχει απάντηση. Σταματώντας την εκτέλεση του ping (Ctrl + c) βλέπουμε πως όλα τα πακέτα που έχουν σταλεί έχουν πέσει στο κενό. Αντίθετα όμως τα δυο μηχανήματα επικοινωνούν κανονικά με τα γειτονικά τους χωρίς να προκύπτει κανένα πρόβλημα.

6. Συμπέρασμα

6.1 Τι πραγματικά είναι το SDN

Το Software Defined Networking έχει τραβήξει τα φώτα της δημοσιότητας πάνω του και έχουν δημιουργηθεί μεγάλες προσδοκίες, ένα hype που έχει οδηγήσει ακόμα και σε χρήση χαρακτηρισμών όπως: η επανάσταση στον χώρο των δικτύων. Η πραγματικότητα είναι πως δεν πρόκειται για κάποια ριζική καινοτομία που θα αλλάξει τα πάντα στον χώρο, παρά για μια συλλογή παλιών και νέων εργαλείων μέσα σε ένα καινούργιο πλαίσιο/αρχιτεκτονική που είναι φορέας μιας νέας αντίληψης των δικτύων.

Το πλαίσιο αυτό θα εμπλουτίσει τον τρόπο που οι τεχνικοί σχεδιάζουν, υλοποιούν και διαχειρίζονται τα δίκτυα. Προστίθενται δυνατότητες που δεν υπήρχαν ως τώρα όμως αυτό γίνεται με έναν μη επιθετικό τρόπο καθώς το SDN δεν επιδιώκει την εκδίωξη του "παλιού" και την επικράτηση της δικής του αρχιτεκτονικής. Αντιθέτως, η ενσωμάτωση του Software Defined Networking θα γίνει σταδιακά, με την χρήση και επέκταση των υπαρχόντων υποδομών και σε

6.2 Συμπεράσματα από την υλοποίηση

Η υλοποίηση μιας Software Defined Networking προσομοίωσης παρουσιάζει μια σειρά μέτρων δυσκολιών, απαιτεί κάποιες γνώσεις σε Linux Command Line και γνώσεις λειτουργίας δικτύων. Είναι ωστόσο μια πολύ διδακτική εμπειρία που μέσα από την υπερπήδηση των δυσκολιών προσφέρει ένα εύρος γνώσεων γύρω από Linux συστήματα, Virtualizing.

Το Mininet είναι ένα εύκολο στην χρήση και πολλών δυνατοτήτων πρόγραμμα προσομοίωσης δικτύων το οποίο προσφέρεται για πειραματισμούς και είναι ιδιαίτερα εύχρηστο σε περιβάλλον SDN όπου καλείται να συνεργαστεί με Controllers.

Ο OpenDaylight αποδεικνύεται ένας Controller πιο απαιτητικός σε γνώσεις, που ζητά οικειότητα στο command line και σε Linux εντολές, όμως δίνει μεγάλες δυνατότητες παραμετροποίησης και απευθύνεται κυρίως σε προγραμματιστές αποτελώντας μια πλατφόρμα βάσης πάνω στην οποία ο εκάστοτε οργανισμός/επιχείρηση θα εξελίξει την δική της πρόταση.

Ο Controller της HP είναι πιο εύχρηστος για έναν αρχικό χρήστη, επίσης το περιβάλλον του είναι πολύ φιλικότερος προσφέροντας περισσότερες λειτουργίες από την

εγκατάσταση του. Οι εφαρμογές είναι εύκολα προσβάσιμες, με λεπτομερή περιγραφή των λειτουργιών τους με λίγο πιο περίπλοκη διαδικασία εγκατάστασης όμως. Συνολικά είναι ένας άριστος Controller που καλύπτει αποτελεσματικά το σύνολο μιας SDN υλοποίησης.

Συνολικά ήταν μια ιδιαίτερα διδακτική εμπειρία όσον αφορά την κατανόηση του τρόπου λειτουργίας του Software Defined Networking, των δυνατοτήτων και περιορισμών που αυτή η νέα δικτυακή προσέγγιση φέρνει στα ήρεμα νερά του Level 2/Level 3 των δικτύων.

6.3 Το παρόν και το μέλλον του SDN

6.3.1 Επί του παρόντος

Φαίνεται να επικρατεί μια στασιμότητα όσον αφορά το Software Defined Networking, ως έννοια απόλαυσε της προσοχής του κόσμου των δικτύων στα σχετικά blogs και sites το 2013, δημοσιότητα που συνέχισε και για το 2014. Οι προβλέψεις για τον μετασχηματισμό των δικτύων, αλλά κυρίως για την αγορά που θα δημιουργούσε και για την αξία που θα αποκτούσε ως τεχνολογική καινοτομία κάλπαζαν. Τα 2 προηγούμενα έτη (2015 και 1016) όμως παρατηρείται μια ηρεμία στο ζήτημα, με αισθητά λιγότερα επιστημονικά άρθρα, δημοσιεύσεις εταιριών, και ενθουσιασμό στα forums/sites/blogs.

Αυτό δεν οφείλεται στην εγκατάλειψη του SDN αλλά πρόκειται για το “ξεπέρασμα” των αρχικών διογκωμένων προσδοκιών, παιδική ασθένεια κάθε νέας τεχνολογίας.

Στον παρόντα χρόνο το Software Defined Networking ανασυντάσσεται, ωριμάζει, μαθαίνει να τρέχει αντί να περπατά, βρίσκει τα όρια του και τον τρόπο με τον οποίον θα ενταχθεί στην πραγματικότητα των δικτύων. Οι εταιρίες και οι κοινότητες συνεχίζουν να εξελίσσουν, να μετασχηματίζουν και να εμπλουτίζουν το SDN, όμως όχι κάτω από το φως των προβολέων της δημοσιότητας. Αυτό ερμηνεύεται από κάποιους ειδικούς ως σημάδι σοβαρής προετοιμασίας για την διάθεση ολοκληρωμένων SDN υλοποιήσεων και της συνειδητοποίησης πως αυτό πρέπει να γίνει με έναν τρόπο που δεν θα δημιουργήσει προσδοκίες που θα διαψευστούν ούτε θα δοθεί μια αρχιτεκτονική που δεν έχει αναπτυχθεί πλήρως.

6.3.2 Επί του μέλλοντος

Καθώς το SDN θα σταθεροποιεί την μορφή του μέσα από την μακροχρόνια εφαρμοσμένη λειτουργία του στα Data Centers των μεγάλων εταιριών, υπάρχει η εκτίμηση πως θα δημιουργηθούν διάφορα εταιρικά οικοσυστήματα που θα αλλοτριώσουν ως κάποιον βαθμό την τον ανοιχτό χαρακτήρα του εγχειρήματος. Ο χρόνος πάντως σίγουρα θα εμπλουτίσει τις συμβατότητες του SDN με όλο και περισσότερους κατασκευαστές και δικτυακό εξοπλισμό, κατά συνέπεια η τιμή των σχετικών Switches τους θα πέσει κάνοντας την αρχιτεκτονική πιο προσιτή σε εταιρίες μεσαίου και μικρού μεγέθους. (Ramel. D, 2015)

Δεν αποκλείεται κάποια εταιρία να προσφέρει μια εξαιρετική υλοποίηση που θα της επιτρέψει την επικράτηση της στις πωλήσεις SDN Controllers και Switches, επειδή όμως δεν φαίνεται στον ορίζοντα κάποια ανατροπή και καθώς οι περισσότερες εταιρίες προσφέρουν ήδη ολοκληρωμένες και ανταγωνιστικές υλοποιήσεις μπορεί να υποτεθεί πως η παρούσα ισορροπία θα συνεχιστεί για αρκετά ακόμα. Οι πολλές επιλογές θα ενισχύσουν τον ανταγωνισμό που επίσης θα συντελέσει στην πτώση των τιμών όσον αφορά την αγορά κάποιου Controller και μιας σουίτας εφαρμογών.

Παράλληλα αναμένεται να συνεχιστεί η δράση των κοινοτήτων λογισμικού και της ύπαρξης του ανοιχτού κώδικα στο Software Defined Networking με την ανάπτυξη και διεύρυνση των ήδη υπαρχόντων βάσεων της ως παράλληλες στις εταιρικές υλοποιήσεις. Η διαχρονική ανάμειξη μη κερδοσκοπικών δομών στο SDN θα εγγυηθεί την ύπαρξη των ανοιχτών προτύπων καθώς και διάθεση δωρεάν εργαλείων που θα επιτρέπουν σε μικρές εταιρίες αλλά και μεμονωμένα άτομα να έχουν πρόσβαση στην αρχιτεκτονική ώστε να πειραματιστούν και να κάνουν υλοποιήσεις μικρής κλίμακας.

7. Βιβλιογραφία

- code.facebook.com. (2015). *Introducing “6-pack”: the first open hardware modular switch*. <https://code.facebook.com/posts/717010588413497/introducing-6-pack-the-first-open-hardware-modular-switch/>
- Duffy, J. (2015). *NSA uses OpenFlow for tracking... its network*. <http://www.networkworld.com/article/2937787/sdn/nsa-uses-openflow-for-tracking-its-network.html>
- Hubbard, P. (2015). *Why you shouldn't overlook Microsoft SDN*. <http://searchsdn.techtarget.com/opinion/Why-you-shouldnt-overlook-Microsoft-SDN>
- ONF. (2014). *Software-Defined Networking (SDN) Definition* . <https://www.opennetworking.org/sdn-resources/sdn-definition>
- Paul Goransson, C. B. (2014). *Software Defined Networks: A Comprehensive Approach*. Morgan Kaufmann.
- Salisbury, B. (2013). *Inside Google's Software-Defined Network*. <http://www.networkcomputing.com/networking/inside-googles-software-defined-network/512240144>
- ONF. (2013). *OpenFlow Switch Specification*. <https://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/openflow-spec-v1.3.2.pdf>
- Cisco. (2014). *Cisco Extensible Network Controller Configuration Guide*. http://www.cisco.com/c/en/us/td/docs/net_mgmt/xnc/xnc_config/1-7/b_XNC_Configuration_Guide_17/b_XNC_Configuration_Guide_17_chapter_00.html
- Hinden, R. (2014). *SDN & Security: Why take over the hosts when you can take over the network*. Ανάκτηση από RSA Conference.
- ONF. (2013). *OpenFlow Switch Specification*. <https://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/openflow-spec-v1.3.2.pdf>

- Ramel, D. (2015). *Survey: Not So Fast on this SDN Thing*.
<https://virtualizationreview.com/articles/2015/01/09/piper-jaffray-sdn-survey.aspx>
- Othmane Blial, M. B. (2016). *An Overview on SDN Architectures with Multiple Controllers*. Journal of Computer Networks and Communications.
- Cisco. (2014). *Cisco Extensible Network Controller Configuration Guide*.
http://www.cisco.com/c/en/us/td/docs/net_mgmt/xnc/xnc_config/1-7/b_XNC_Configuration_Guide_17/b_XNC_Configuration_Guide_17_chapter_00.html
- Ericsson. (2015). *Identifying and addressing the vulnerabilities and security issues of SDN*. Stockholm, Sweden: Ericsson AB.
- Hewlett-Packard. (2014). *HP VAN SDN Controller Administrator Guide*. Hewlett-Packard Development Company.
- Hinden, R. (2014). *SDN & Security: Why take over the hosts when you can take over the network*. RSA Conference.
- Hu, F. (2014). *Network Innovation through OpenFlow and SDN*. Auerbach Publications.
- ONF. (2012). *Software-Defined Networking: The New Norm for Networks*. Open Networking Foundation.
- ONF. (2013). *OpenFlow Switch Specification*.
<https://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/openflow-spec-v1.3.2.pdf>
- Ramel, D. (2015). *Survey: Not So Fast on this SDN Thing*.
<https://virtualizationreview.com/articles/2015/01/09/piper-jaffray-sdn-survey.aspx>
- SDxCentral - Cisco. (2015). *SDN Controllers Report*.
https://www.sdxcentral.com/wp-content/uploads/2015/11/2015_SDxCentral_-SDN_Controllers-Report_Cisco_FINAL.pdf