ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΚΑΛΥΨΗ ΤΟΠΟΛΟΓΙΑΣ ΔΙΚΤΥΟΥ ΚΑΘΟΡΙΣΜΕΝΗ ΑΠΟ ΛΟΓΙΣΜΙΚΟ

Χριστιάνα Άγκο

ΑΜ:15250

Επιβλέπουσα καθηγήτρια: Μαργαρίτη Σπυριδούλα

ΑΡΤΑ 2020

NETWORK TOPOLOGY DISCOVERY IN SDN

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής

2. Μέλος επιτροπής

3. Μέλος επιτροπής

Ο/Η Προϊστάμενος/η του Τμήματος

Υπογραφή

Δήλωση μη λογοκλοπής

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

Άγκο Χριστιάνα

Υπογραφή

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά την επιβλέπουσα καθηγήτρια Μαργαρίτη Σπυριδούλα για την ανάθεση της πτυχιακής εργασίας καθώς και για την βοήθεια και καθοδήγηση που έλαβα καθ' όλη την διάρκεια της μελέτης, επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την στήριξη τους όλο αυτό το διάστημα.

ΠΕΡΙΛΗΨΗ

Η συγκεκριμένη πτυχιακή εργασία έχει σκοπό να μελετήσει μια νέα τεχνολογία στα δίκτυα υπολογιστών η οποία έχει την ονομασία δίκτυα που είναι καθοριζόμενα από λογισμικό, σκοπός αυτών των δικτύων είναι να κάνουν το δίκτυο πιο ευέλικτο μέσω του προγραμματισμού. Μια ακόμη σημαντική παράμετρος είναι και η τοπολογία αυτών των δικτύων η οποία είναι σημαντική καθώς την διαχειρίζεται ο κεντρικός ελεγκτής ο οποίος είναι και ο ¨εγκέφαλος¨ ολόκληρού του δικτύου. Επιπλέον αναλύονται γενικά ζητήματα γύρω από το SDN και την τοπολογία του όπως το τι είναι, ιστορική αναδρομή ,αρχιτεκτονική κλπ., καθώς και το βασικό πρωτόκολλο αυτών το δικτυών το OpenFlow όπως επίσης και τα πρωτοκολλά OFDP και LLDP τα οποία χρησιμοποιούνται για την ανίχνευση αυτής της τοπολογίας. Τέλος, κάνουμε προσομοίωση αυτών των δικτύων χρησιμοποιώντας το λογισμικό mininet με τον pox ελεγκτή.

Λέξεις-κλειδιά: SDN, τοπολογία δικτύου, OpenFlow, Mininet

ABSTRACT

This dissertation aims to study a new technology in computer networks called software-defined networking, the purpose of these networks is to make the network more flexible through programming. Another important parameter is the topology of these networks which is important as it is managed by the central controller who is the ¨brain¨ of the entire network. In addition, issues are generally discussed around SDN and its topology such as what it is, historical background, architecture, etc., as well as the basic protocol of these networks the OpenFlow and the OFDP and LLDP protocols which are used to detect this topology. Finally, we simulate these networks using the mininet software with the pox controller.

Keywords: SDN, topology discovery, OpenFlow, Mininet

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

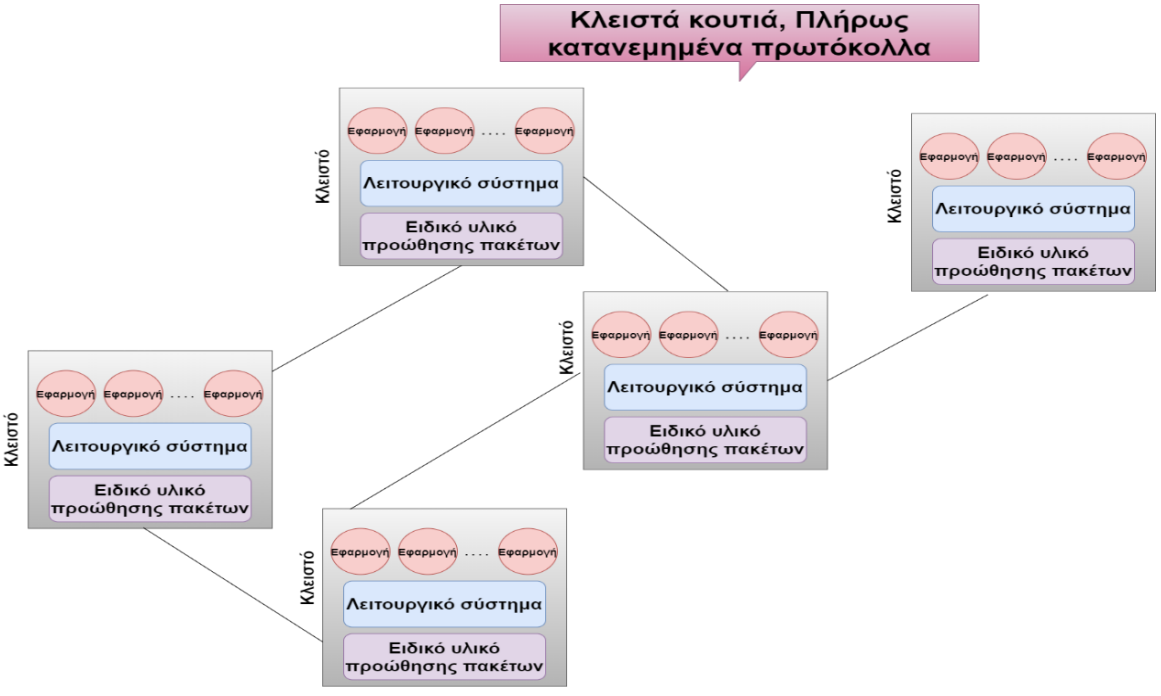
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

Κεφάλαιο 1: Εισαγωγή

Τα δίκτυα που είναι καθοριζόμενα από λογισμικό είναι μια καινούργια τεχνολογία που σκοπό έχει να κάνει το δίκτυο πιο ευέλικτο και προγραμματιζόμενο. Οι διαχειριστές μπορούν να ελέγχουν ολόκληρο το δίκτυο χάρις τον κεντρικό ελεγκτή και την ευφυΐα του, έτσι σε σχέση με τα παραδοσιακά δίκτυα που για να διαμορφώσουμε το δίκτυο μας θα απαιτούσε περισσότερη προσπάθεια πλέον με την ιδιότητα του προγραμματισμού όλα γίνονται ευκολότερα χωρίς να χρειαζόμαστε περεταίρω και επιπλέον υλικό. Η τοπολογία του δικτύου είναι σημαντική καθώς οι πληροφορίες συλλέγονται στον κεντρικό ελεγκτή από τις συσκευές δικτύου από το επίπεδο δεδομένων της αρχιτεκτονικής του, έτσι βοηθούν τον ελεγκτή να έχει μια αφηρημένη εικόνα ολόκληρου του δικτύου επιτρέποντας του την ομαλή και αποτελεσματική λειτουργία.

1.1 Παραδοσιακά δίκτυα – Σύγχρονα δίκτυα



Εικόνα 1: Παραδοσιακά δίκτυα [1]

Τα παραδοσιακά δίκτυα αποτελούνται από ένα σύνολο δικτυακών συσκευών (δρομολογητές, μεταγωγείς, κ.ά.), μέσα μετάδοσης και τελικούς κόμβους (υπολογιστές, εξυπηρετητές, κ.λπ.) με

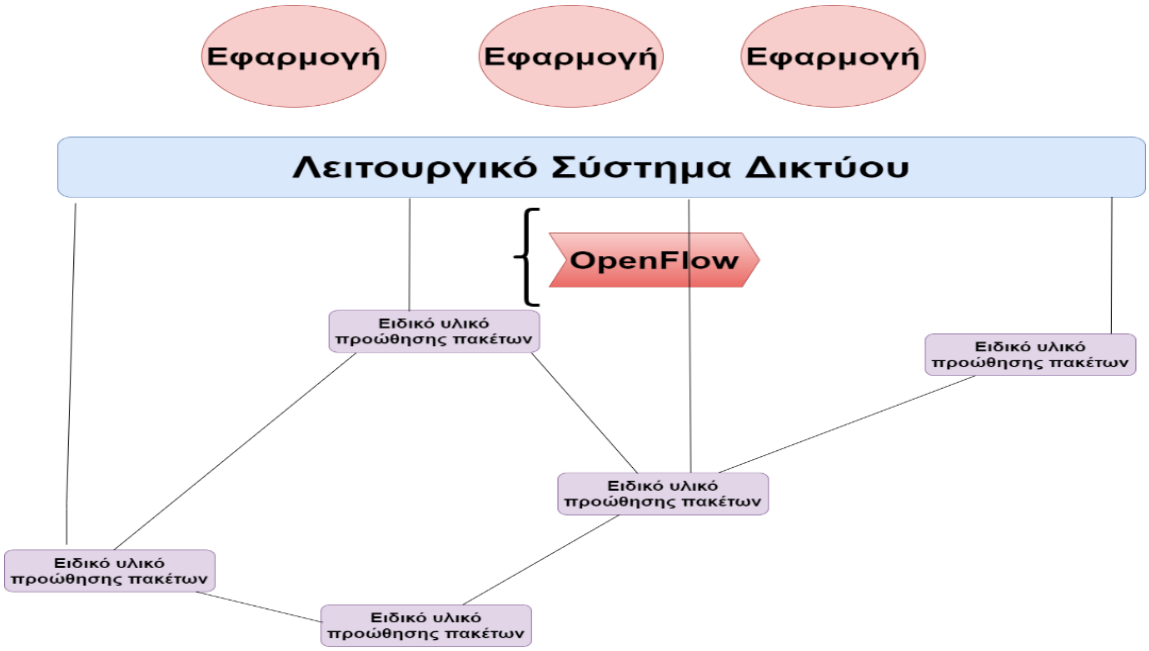
σκοπό την επικοινωνία ή την παροχή υπηρεσιών. Στην Εικόνα 1 βλέπουμε τη μορφή ενός παραδοσιακού δικτύου όπως το χρησιμοποιούμε μέχρι και σήμερα. Οι ενδιάμεσοι κόμβοι του δικτύου, για παράδειγμα οι δρομολογητές, οι μεταγωγείς κ.λπ. για να επιτελέσουν το ρόλο τους στηρίζονται σε τρία επίπεδα: το υλικό, το λειτουργικό σύστημα και τις εφαρμογές που εκτελούνται πάνω από το λειτουργικό σύστημα. Ο ρόλος του δρομολογητή είναι να δρομολογήσει τα δεδομένα μεταξύ των δικτύων (π.χ. μεταξύ δύο τοπικών δικτύων). Αντίστοιχα ο μεταγωγέας αναλαμβάνει την προώθηση των δεδομένων στο ίδιο δίκτυο.

Για την επίτευξη της δικτύωσης και την μετάδοση των δεδομένων, κάθε συσκευή θα πρέπει να ρυθμιστεί ξεχωριστά έτσι ώστε να:

- Επίπεδο δεδομένων: προωθεί τα πακέτα στον προορισμό τους.
- Επίπεδο ελέγχου: δρομολογεί την κίνηση λαμβάνοντας τις κατάλληλες αποφάσεις.
- Επίπεδο διαχείρισης: επιβλέπει και παρακολουθεί την δικτυακή κίνηση.

Πέρα από τη διαμόρφωση της κάθε συσκευής, απαιτείται και η χρήση πολλών και διαφορετικών πρωτοκόλλων καθώς η κάθε συσκευή βασίζεται στο δικό της επίπεδο δεδομένων και επίπεδο ελέγχου. Αυτά τα συστήματα μέχρι και σήμερα είναι κλειστά συστήματα που σημαίνει ότι δεν μπορούμε να επέμβουμε στον κώδικα του συστήματος για να τον τροποποιήσουμε ή για να ενσωματώσουμε δικό μας κώδικα. Για παράδειγμα αν αγοράσουμε έναν δρομολογητή από την cisco και επιθυμούμε να γράψουμε ένα δικό μας πρόγραμμα και να το τρέξουμε μέσα στο λειτουργικό σύστημα αυτού του συστήματος δεν μπορούμε να το κάνουμε διότι είναι κλειστό. Για να απαλειφθούν τα παραπάνω προβλήματα, να μειωθεί η πολυπλοκότητα λειτουργίας τους, αλλά και το κόστος εγκατάστασης και συντήρησης του δικτύου αξιοποιήθηκε η ιδέα της αποσύνδεσης του επιπέδου ελέγχου από το επίπεδο δεδομένων. Το αποτέλεσμα αυτής της ιδέας είναι τα δίκτυα που είναι καθοριζόμενα από το λογισμικό, όπου βασική επιτυχία αυτών των δικτύων είναι η αρχιτεκτονική τους που είναι πλέον ένα ανοιχτό σύστημα. Τα συστήματα αυτά είναι πιο ευκολά προγραμματιζόμενα καθώς μπορούμε να προγραμματίσουμε συνολικά τα δίκτυα υπολογιστών και όχι ανά συσκευή. Ουσιαστικά αυτό που κάνει είναι ότι παίρνει την παραδοσιακή τεχνολογία, βγάζει το λειτουργικό έλεγχο μέσα από τους κόμβους και δημιουργεί ένα ενιαίο σύστημα ελέγχου για όλο το δίκτυο. Επιπλέον, οι εφαρμογές αφαιρούνται από τους κόμβους και τοποθετούνται για να τρέχουν πάνω από το ενιαίο λειτουργικό σύστημα που είναι όλου του δικτύου. Έπειτα σε αυτήν την προσέγγιση έχουμε μια ανοιχτή διεπαφή μέσω της οποίας το λειτουργικό σύστημα του δικτύου

διαχειρίζεται το υλικό και εισάγει κάποιους κανόνες σε αυτό το υλικό για την προώθηση των πακέτων το οποίο ονομάζεται OpenFlow [1].



Εικόνα 2: Σύγχρονα δίκτυα [1]

1.2 Σύγκριση παραδοσιακών δικτυών με το SDN δίκτυο

Η κύρια διαφορά ανάμεσα στα παραδοσιακά δίκτυα και το SDN είναι ότι τα παραδοσιακά δίκτυα βασίζονται σε υλικό ενώ το SDN σε λογισμικό. Επειδή βασίζεται σε λογισμικό το SDN είναι πιο ευέλικτο επιτρέποντας στους χρήστες μεγαλύτερο έλεγχο και ευκολία στη διαχείριση των πόρων σχεδόν σε όλο το επίπεδο ελέγχου. Αντίθετα τα παραδοσιακά δίκτυα χρησιμοποιούν μεταγωγείς, δρομολογητές και άλλες φυσικές υποδομές για τη δημιουργία συνδέσεων και την λειτουργία του δικτύου. Στα SDN δίκτυα η επικοινωνία βασίζεται στα API (Northbound και Southbound) που αλληλεπιδρούν με τους ελεγκτές. Χάρη αυτής της επικοινωνίας, οι προγραμματιστές εφαρμογών μπορούν να προγραμματίσουν απευθείας το δίκτυο, σε αντίθεση με τη χρήση των πρωτοκόλλων που απαιτούνται από την παραδοσιακή δικτύωση. Εν κατακλείδι, για παράδειγμα θα μπορούσαμε να πούμε ότι το ένα απαιτεί περισσότερο εξοπλισμό για επέκταση και το άλλο απαιτεί μόνο να

πληκτρολογήσεις [2]. Ο πίνακας 1 αναφέρει μερικά παραδείγματα με βάση κάποια βασικά κριτήρια ανάμεσα στις διαφορές του παραδοσιακού δικτύου με το SDN:

Κριτήρια	**Παραδοσιακό δίκτυο**	**SDN**
Διαχείριση δικτύου	Δύσκολη διότι οι αλλαγές εφαρμόζονται ξεχωριστά σε κάθε συσκευή	Εύκολη με την βοήθεια του ελεγκτή
Κόστος συντήρησης	Υψηλότερο	Χαμηλότερο
Απαιτούμενος χρόνος για τον χειρισμό ενημερώσεων/σφαλμάτων	Μερικές φορές μπορεί να διαρκέσει μήνες	Εύκολος λόγω του κεντρικού ελεγκτή
Χρήση ελεγκτή	Δεν απαιτείται	Σημαντική
Αξιοποίηση πόρων	Χαμηλή	Υψηλή

Πίνακας 1: Σύγκριση παραδοσιακών δικτύων με το SDN δίκτυο [3]

1.3 Αντικείμενο της εργασίας

Το αντικείμενο της εργασίας είναι να μελετήσουμε και να εξετάσουμε τους αλγορίθμους που εφαρμόζονται για να ανιχνεύεται η τοπολογία των δικτύων που είναι καθοριζόμενα από λογισμικό. Συγκεκριμένα να μελετήσουμε πως αυτά τα δίκτυα μαθαίνουν για την τοπολογία τους.

1.4 Οργάνωση εργασίας

Η συγκεκριμένη εργασία αποτελείται από πέντε κεφάλαια. Στο πρώτο κεφάλαιο γίνεται παρουσίαση των παραδοσιακών δικτύων με τα σύγχρονα δίκτυα και κάνουμε μια σύγκριση μεταξύ τους, έπειτα αναλύεται το αντικείμενο της εργασίας. Το δεύτερο κεφάλαιο αναφέρεται στην τεχνολογία SDN, αναλυτικότερα παρουσιάζεται ο ορισμός, η ιστορική του αναδρομή, η

αρχιτεκτονική του, τα χαρακτηριστικά καθώς και τα πλεονεκτήματα και μειονεκτήματα που έχει. Στην συνέχεια, στο κεφάλαιο τρία βλέπουμε το βασικό πρωτόκολλο που χρησιμοποιείται στα συγκεκριμένα δίκτυα το OpenFlow, συγχρόνως αναλύεται η ιστορική αναδρομή και η αρχιτεκτονική του. Στο τέταρτο κεφάλαιο γίνεται μελέτη στην ανίχνευση της τοπολογίας με τα πρωτόκολλα OFDP και LLDP, ενώ μιλάμε για τα βασικά χαρακτηριστικά της τοπολογίας ενός δικτύου καθώς και για την τοπολογία στα δίκτυα SDN. Τέλος στο πέμπτο κεφάλαιο κάνουμε προσομοίωση χρησιμοποιώντας το mininet και τον pox ελεγκτή, επιπλέον μιλάμε για το λογισμικό mininet, τις προκαθορισμένες τοπολογίες και δημιουργούμε τοπολογίες δικτύων με κάποιες βασικές εντολές.

Κεφάλαιο 2: Δίκτυα καθοριζόμενα από λογισμικό

2.1 Τι είναι το SDN

Το SDN είναι μια δικτύωση η οποία είναι καθορισμένη από λογισμικό και η αρχιτεκτονική του έχει σχεδιαστεί για να κάνει ένα δίκτυο πιο ευέλικτο, προγραμματιζόμενο και ευκολότερο στη διαχείριση του μέσω ενός κεντρικού ελεγκτή. Ουσιαστικά πραγματοποιείται εικονικοποίηση του δικτύου όπου επιδιώκεται να βελτιστοποιηθεί η χρήση των πόρων του δικτύου και να προσαρμόσει γρήγορα τα δίκτυα στις μεταβαλλόμενες επιχειρηματικές ανάγκες [4]. Η βασική θεμελιώδης αρχή του είναι ο διαχωρισμός του επιπέδου ελέγχου από το επίπεδο δεδομένων. Το επίπεδο ελέγχου είναι ο ¨εγκέφαλος¨ του δικτύου, μπορεί να εκτελεστεί ξεχωριστά από τις συσκευές και υπολογίζει τη λογική του τρόπου προώθησης της κίνησης. Το επίπεδο δεδομένων είναι κατανεμημένο στο υλικό και ελέγχεται από το επίπεδο ελέγχου [5].

2.2 Ιστορική αναδρομή στο SDN

Ως πρώτη εφαρμογή που χρησιμοποίησε τη λογική του διαχωρισμού του επιπέδου ελέγχου από το επίπεδο δεδομένων ήταν το δημόσιο τηλεφωνικό δίκτυο ως ένας τρόπος απλοποίησης της παροχής και της διαχείρισης. Εκτός από αυτό η επιχειρησιακή ομάδα τεχνολογίας του διαδικτύου (IETF) άρχισε να εξετάζει με διάφορους τρόπους την λογική της διάκρισης των λειτουργιών. Το 2004 δημοσίευσε ένα προτεινόμενο πρότυπο διεπαφής με την ονομασία " Forwarding and Control Element Separation" (ForCES) [6]. Η ιδέα του SDN προήλθε από το έργο του Πανεπιστημίου του Στάνφορντ, το οποίο πήρε την ονομασία OpenFlow το 2006 από τον Nick Mckeown, ο οποίος είναι καθηγητής στο τμήμα Ηλεκτρολόγων Μηχανικών και Επιστήμης των Υπολογιστών. Ακριβέστερα ο όρος «software-defined networking» χρησιμοποιήθηκε για πρώτη φορά από την Open Networking Foundation (ONF), η οποία είναι μια μη κερδοσκοπική οργάνωση που ιδρύθηκε τον Μάρτιο του 2011 και αποτελείται από 69 μέλη, συμπεριλαμβανομένων των IBM, Cisco, Juniper, HP και Dell, συγκεκριμένα δημιουργήθηκε για να επιταχύνει την ανάπτυξη, την τυποποίηση και την εμπορευματοποίηση του SDN. Ο στόχος του ONF είναι να μετατρέψει τη βιομηχανία δικτύωσης σε βιομηχανία λογισμικού μέσω του SDN και να προωθήσει μια τυπική ανάπτυξη. Τον Δεκέμβριο του 2009 κυκλοφόρησε το OpenFlow 1.0, ενώ σημαντικοί προμηθευτές όπως η IBM, η HP και η Cisco έχουν κυκλοφορήσει προϊόντα που υποστηρίζουν το πρωτόκολλο

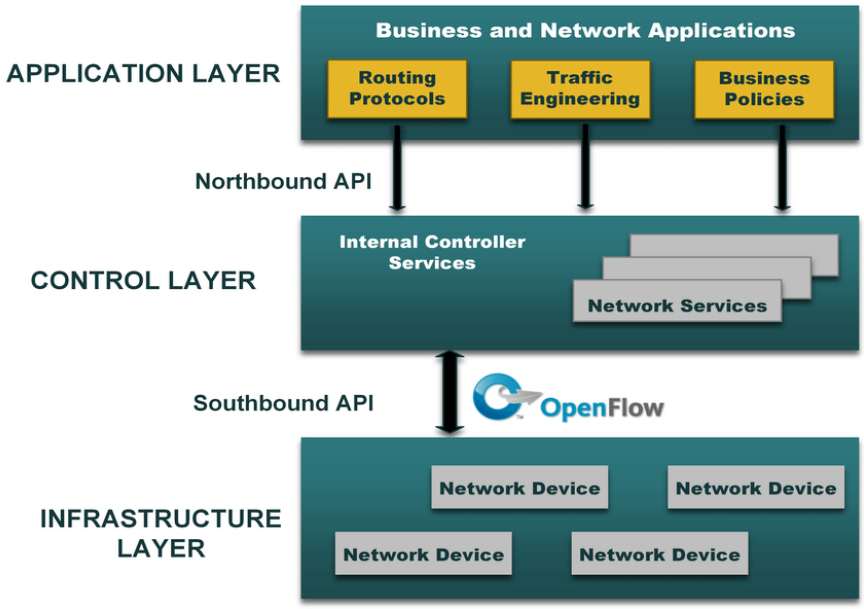
OpenFlow. Μια από τις πιο γνωστές εταιρίες η Google έχει αναπτύξει το SDN στα κέντρα δεδομένων της σε όλο τον κόσμο [7].

2.3 Αρχιτεκτονική

Η αρχιτεκτονική του δικτύου που καθορίζεται από λογισμικό αποτελείται από τρία βασικά επίπεδα το:

- Επίπεδο εφαρμογής (Application layer)
- Επίπεδο ελέγχου (Control layer)
- Επίπεδο υποδομής (Infrastructure layer)

Για να επικοινωνήσουν μεταξύ τους τα επίπεδα, το SDN χρησιμοποιεί διεπαφές προγράμματος εφαρμογών northbound και southbound (APIs) όπου το northbound API επικοινωνεί μεταξύ του επιπέδου ελέγχου και εφαρμογής και το southbound API επικοινωνεί μεταξύ του επιπέδου ελέγχου και υποδομής.



Εικόνα 3: Τα 3 επίπεδα της αρχιτεκτονικής του δικτύου SDN [8]

2.3.1 Επίπεδο εφαρμογής

Το επίπεδο εφαρμογής περιέχει προγράμματα που γνωστοποιούν ρητά και μέσω προγραμματισμού την επιθυμητή συμπεριφορά και τις απαιτήσεις του δικτύου στον ελεγκτή.

2.3.2 Επίπεδο ελέγχου

Στο επίπεδο ελέγχου ο ελεγκτής συνδέει το επίπεδο εφαρμογής και το επίπεδο υποδομής, αυτό το επίπεδο επεξεργάζεται τις οδηγίες και τις απαιτήσεις που αποστέλλονται από το επίπεδο εφαρμογής (μέσω της διεπαφής Northbound) και τις προωθεί στα στοιχεία δικτύωσης (μέσω της διεπαφής Southbound), ανακοινώνει επίσης τις απαραίτητες πληροφορίες που εξήχθησαν από τις συσκευές δικτύωσης στην εφαρμογή για τη βέλτιστη λειτουργία.

2.3.3 Επίπεδο υποδομής

Το επίπεδο υποδομής απαρτίζεται από συσκευές δικτύωσης που ελέγχουν τις δυνατότητες προώθησης και επεξεργασίας δεδομένων για το δίκτυο, πιο συγκεκριμένα οι συσκευές είναι υπεύθυνες για τον χειρισμό πακέτων με βάση τους κανόνες που παρέχει ο ελεγκτής, είναι το φυσικό επίπεδο που είναι υπεύθυνο για τη συλλογή καταστάσεων δικτύου, όπως στατιστικά κίνησης, τοπολογία δικτύου, χρήση δικτύου κ.λπ. καθώς και για την αποστολή τους στο επίπεδο ελέγχου [9].

2.3.4 Northbound interface

Οι εφαρμογές βασίζονται στον ελεγκτή για να τους πει ποια είναι η κατάσταση της υποδομής του δικτύου ώστε να μπορούν να γνωρίζουν ποιοι πόροι είναι διαθέσιμοι, επίσης ο ελεγκτής διασφαλίζει αυτόματα ότι η κυκλοφορία των εφαρμογών δρομολογείται σύμφωνα με τις πολιτικές που καθορίζονται από τους διαχειριστές του δικτύου, συνεπώς οι εφαρμογές επικοινωνούν με το επίπεδο ελέγχου όπου και συντονίζει τον τρόπο με τον οποίο οι εφαρμογές λαμβάνουν τους διαθέσιμους πόρους στο δίκτυο και του αναφέρουν τι πόρους χρειάζονται για τον προορισμό τους.

2.3.5 Southbound interface

Ο ελεγκτής επικοινωνεί με το επίπεδο υποδομής, όπως δρομολογητές και μεταγωγείς μέσω του southbound, επιπλέον η υποδομή ενημερώνεται για ποια διαδρομή πρέπει να ακολουθήσουν τα δεδομένα της εφαρμογής όπως αποφασίστηκε από τον ελεγκτή. Σε πραγματικό χρόνο, ο ελεγκτής μπορεί να αλλάξει τον τρόπο μεταφοράς δεδομένων από τους δρομολογητές και τους μεταγωγείς, όμως τα δεδομένα δεν βασίζονται πλέον στις συσκευές και στους πίνακες δρομολόγησης για να προσδιορίσουν πού πηγαίνουν τα δεδομένα σε αντίθεση η ευφυΐα του ελεγκτή λαμβάνει ενημερωμένες αποφάσεις που βελτιστοποιούν τη διαδρομή των δεδομένων [10].

2.4 Τα χαρακτηριστικά του SDN

Άμεσα προγραμματιζόμενο: Οι διαμορφώσεις ελέγχου δικτύου μπορούν να γραφτούν σε γλώσσα Python ή σε γλώσσα C.

Ευέλικτο: Οι διαχειριστές μπορούν να προσαρμόσουν δυναμικά τη ροή κυκλοφορίας σε όλο το δίκτυο κατά το χρόνο εκτέλεσης.

Κεντρική διαχείριση: Η ευφυΐα και η διαμόρφωση του δικτύου είναι συγκεντρωτικά και εμφανίζονται στις εφαρμογές ως ένας ενιαίος, λογικός μεταγωγέας.

Αυτοματισμός: Τα αυτόματα προγράμματα SDN μπορούν να γραφτούν για να διαμορφώσουν και να διαχειριστούν τους πόρους του δικτύου.

Ανοιχτά πρότυπα και ουδέτερα από τον προμηθευτή: Ο ελεγκτής SDN είναι πολύ πιο εύκολος στην εφαρμογή και τη διαχείριση από πολλές συσκευές και πρωτόκολλα που αφορούν συγκεκριμένους προμηθευτές [11].

2.5 Πλεονεκτήματα και μειονεκτήματα του SDN

Πλεονεκτήματα:

Ορατότητα: Ο διαχωρισμός του επιπέδου δεδομένων από το επιπέδου ελέγχου δίνει μια εναέρια όψη ολόκληρου του δικτύου, αυτό κάνει τις αλλαγές πολύ πιο εύκολες, καθώς μπορούμε να δούμε τις επιδράσεις του δικτύου καθαρά.

Αυξημένη ταχύτητα: Χάρις την ορατότητα, οι αλλαγές μπορούν να γίνουν αυξάνοντας την ταχύτητα ολόκληρου του δικτύου.

Καλύτερη ασφάλεια: Το SDN επιτρέπει καλύτερη ασφάλεια με πολλούς τρόπους, καθώς η ορατότητα βοηθάει στο να εντοπιστούν τυχόν κίνδυνοι ασφαλείας και να εμποδιστεί να εξαπλωθούν, επιπροσθέτως τα άτομα που έχουν πρόσβαση στον ελεγκτή μπορούν να διευκρινίσουν ασφαλείς διαδρομές στο σύστημα χωρίς το τείχος προστασίας για να αποκλείσουν τυχόν μη έγκυρες διαδρομές [12].

Κεντρική παροχή: Το SDN παρέχει την ικανότητα διαχείρισης ενός δικτύου από μια κεντρική προοπτική. Με λίγα λόγια, εικονικοποιεί τόσο τα δεδομένα όσο και τα επίπεδα ελέγχου επιτρέποντας στο χρήστη να παρέχει φυσικά και εικονικά στοιχεία από μία τοποθεσία. Αυτό είναι

εξαιρετικά χρήσιμο καθώς η παραδοσιακή υποδομή μπορεί να είναι δύσκολο να ελεγχθεί ειδικά εάν υπάρχουν πολλά διαφορετικά συστήματα που πρέπει να διαχειριστούν μεμονωμένα.

Επεκτασιμότητα: Χάρις την κεντρική παροχή το SDN δίνει στον χρήστη μεγαλύτερη δυνατότητα κλιμάκωσης, έχοντας τη δυνατότητα παροχής πόρων κατά βούληση, έτσι μπορούμε να αλλάξουμε την υποδομή του δικτύου μας αμέσως. Η διαφορά στην επεκτασιμότητα είναι αξιοσημείωτη σε σύγκριση με εκείνη μιας παραδοσιακής εγκατάστασης δικτύου όπου οι πόροι πρέπει να αγοραστούν και να διαμορφωθούν χειροκίνητα [13].

Μειονεκτήματα:

Ευπάθεια του ελεγκτή: Ο ελεγκτής είναι ο τρόπος με τον οποίο διαχειριζόμαστε το δίκτυο αντί των δρομολογητών και των μεταγωγέων, αυτό σημαίνει ότι ο ελεγκτής πρέπει να είναι απολύτως ασφαλής, επιπλέον πρέπει να παρακολουθείτε προσεκτικά ποιος έχει πρόσβαση στον ελεγκτή για να διατηρηθεί η πρόσβαση ασφαλής.

Κατανεμημένες επιθέσεις άρνησης εξυπηρέτησης: Εάν ένας μεγάλος αριθμός αδήλωτων διαδρομών εισάγεται ταυτόχρονα στο δίκτυο, θα ζητήσει μια συγκεκριμένη διαδρομή. Δυστυχώς, αυτή η εισροή αιτημάτων μπορεί να δυσκολέψει το δίκτυο να ανταποκριθεί σε πραγματικά αιτήματα.

Έλλειψη ασφάλειας υλικού: Δεδομένου ότι καταργείτε η χρήση των φυσικών δρομολογητών και των μεταγωγέων, η ασφάλεια που τα συνοδεύει δεν θα είναι η ίδια. Το βασικό στοιχείο που θα λείπει είναι το τείχος προστασίας, με συνέπεια να αφήσει το δίκτυό μας πιο ευάλωτο εάν δεν είμαστε προσεκτικοί [12].

Καθυστέρηση: Ένα από τα προβλήματα με την εικονικοποίηση οποιασδήποτε υποδομής είναι ο λανθάνων χρόνος που προκύπτει ως αποτέλεσμα, καθώς η ταχύτητα της αλληλεπίδρασής μας με μια συσκευή εξαρτάται από πόσους εικονικοποιημένους πόρους έχουμε στη διάθεσή μας. Κάθε ενεργή συσκευή σε ένα δίκτυο επηρεάζει τη διαθεσιμότητα του δικτύου μας και αυτό θα επιδεινωθεί στο μέλλον καθώς περισσότερες συσκευές διαδικτύου των πραγμάτων (IoT) κυκλοφορούν στην αγορά και αρχίζουν να ενσωματώνονται.

Πιο σύνθετη διαχείριση δικτύου: Αν και τα παραδοσιακά δίκτυα μπορεί να έχουν τους περιορισμούς τους, υπάρχει μια τυποποιημένη συναίνεση σχετικά με τις απειλές και τις διαδικασίες ασφαλείας. Σε αυτό το σημείο, δεν υπάρχει τέτοια συναίνεση για το SDN. Αν και

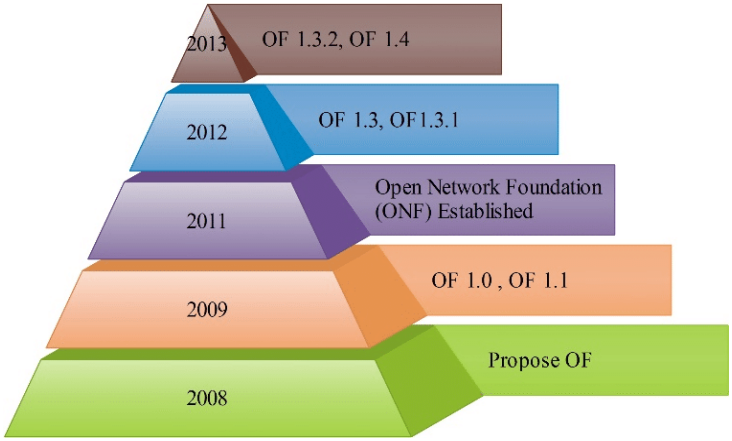
υπάρχουν πολλοί πάροχοι λύσεων SDN, οι ανησυχίες για την ασφάλεια του αποτελούν μη διαγραμμένη περιοχή για πολλούς διαχειριστές. Ως εκ τούτου, μπορεί να είναι πολύ δύσκολο να διατηρηθεί η ακεραιότητα μιας υπηρεσίας SDN έναντι εξωτερικών απειλών, όταν δεν έχουμε τις απαραίτητες γνώσεις για την υπεράσπιση του συστήματος [13].

Κεφάλαιο 3: Το πρωτόκολλο OpenFlow

Το OpenFlow είναι ένα προγραμματιζόμενο πρωτόκολλο δικτύου για το περιβάλλον του SDN, το οποίο χρησιμοποιείται για την επικοινωνία μεταξύ των μεταγωγέων και των ελεγκτών του OpenFlow [14]. Το SDN χρησιμοποιεί το πρωτόκολλο OpenFlow για να διαχωρίσει το επίπεδο δεδομένων από το επίπεδο ελέγχου στο δίκτυο και να τοποθετήσει όλη τη νοημοσύνη σε ένα λογικό στοιχείο που ονομάζεται ελεγκτής. Ο ελεγκτής διαχειρίζεται το δίκτυο και αποφασίζει πού πρέπει να πάει η ροή κυκλοφορίας, προκειμένου να διατηρηθεί η υπηρεσία, καθώς ο ελεγκτής SDN συνεχίζει να ενημερώνει σχετικά με το δίκτυο και την τοπολογία του. Ένα από τα κρίσιμα καθήκοντα για τα οποία είναι υπεύθυνος ο ελεγκτής είναι ότι σε σχεδόν σε πραγματικό χρόνο θα πρέπει να γνωρίζει την τοπολογία του δικτύου. Η διαδικασία αυτή είναι γνωστή ως Topology Discovery. Σε σύγκριση με τα πρωτόκολλα παλαιού τύπου που εξαρτώνται από την κατάσταση σύνδεσης (OSPF)[1], το Topology Discovery στο SDN είναι πιο κρίσιμο. Όλοι οι ελεγκτές δικτύου χρησιμοποιούν το πρωτόκολλο OpenFlow Discovery Protocol (OFDP) για την ανακάλυψη τοπολογίας δικτύου σε ένα δίκτυο SDN χρησιμοποιώντας πλαίσια Link Layer Discovery Protocol (LLDP) σε μορφή μηνυμάτων. Επιπροσθέτως το OpenFlow ανοίγει μια πύλη μεταξύ εφαρμογών δικτύου και προώθησης δικτύου σε περιβάλλον SDN που παραχωρεί την απαραίτητη πρόσβαση για τη διαχείριση και τη διαμόρφωση των μεταγωγέων του δικτύου [15].

[1]Το Open Shortest Path First (OSPF) είναι ένα πρωτόκολλο δρομολόγησης κατάστασης σύνδεσης (link-state) που χρησιμοποιείται για την εύρεση της καλύτερης διαδρομής μεταξύ της πηγής και του δρομολογητή προορισμού χρησιμοποιώντας το δικό του σύντομο μονοπάτι. Το OSPF αναπτύχθηκε από την επιχειρησιακή ομάδα τεχνολογίας του διαδικτύου (IETF) ως ένα από τα πρωτοκολλά εσωτερικών πυλών (IGP), πιο συγκεκριμένα είναι το πρωτόκολλο που στοχεύει στη μετακίνηση του πακέτου σε ένα μεγάλο αυτόνομο σύστημα ή τομέα δρομολόγησης [39].

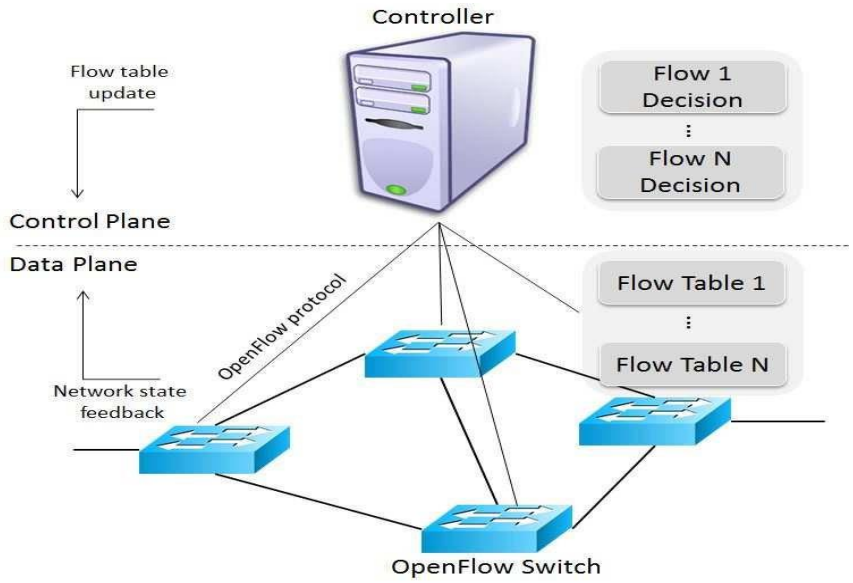
3.1 Ιστορική αναδρομή



Εικόνα 4: Ιστορική αναδρομή του OpenFlow [14]

Ο μη κερδοσκοπικός οργανισμός openflow.org δημιουργήθηκε το 2008 για την προώθηση και υποστήριξη του OpenFlow, αν και το openflow.org υπήρχε επίσημα στο διαδίκτυο, τα πρώτα χρόνια η φυσική του οργάνωση ήταν μια ομάδα ατόμων που συναντήθηκαν στο Πανεπιστήμιο του Στάνφορντ. Η πρώτη έκδοση που κυκλοφόρησε ήταν η 1.0.0 όπου εμφανίστηκε στις 31 Δεκεμβρίου του 2009, αν και υπήρχαν πολλές προ-εκδόσεις πριν από αυτό διατέθηκαν όμως μόνο για πειραματικούς σκοπούς. Σε αυτό το σημείο και συνεχίζοντας μέχρι την κυκλοφορία του 1.1.0, η ανάπτυξη και η διαχείριση των προδιαγραφών πραγματοποιήθηκαν υπό την υποστήριξη του openflow.org. Στις 21 Μαρτίου του 2011, το Open Network Foundation (ONF) δημιουργήθηκε με σκοπό να επιταχύνει την παράδοση και εμπορευματοποίηση του SDN. Για το ONF, το OpenFlow παραμένει ο πυρήνας του οράματος του SDN για το μέλλον, για το λόγο αυτό το ONF έχει γίνει η υπεύθυνη οντότητα για τις εξελισσόμενες προδιαγραφές του OpenFlow. Επομένως ξεκινώντας μετά την κυκλοφορία του V.1.1, έγιναν αναθεωρήσεις στην προδιαγραφή OpenFlow που έχουν κυκλοφορήσει και διαχειριστεί από το ONF [15]. Έπειτα στις 13 Απρίλιου του 2012 κυκλοφόρησε η έκδοση 1.3 οπού η κυριά αλλαγή ήταν η βελτιωμένη περιγραφή των δυνατοτήτων του πίνακα, καθώς οι προηγούμενες εκδόσεις της προδιαγραφής OpenFlow περιλάμβαναν περιορισμένη έκφραση των δυνατοτήτων ενός μεταγωγέα OpenFlow, λίγους μήνες αργότερα τον Αύγουστο του 2012 κυκλοφόρησε και η έκδοση 1.3.1 [16]. Στις 25 Απρίλιου του 2013 κυκλοφόρησε η έκδοση 1.3.2 και τον Αύγουστο του ίδιου χρόνου η έκδοση 1.4.0 [17].

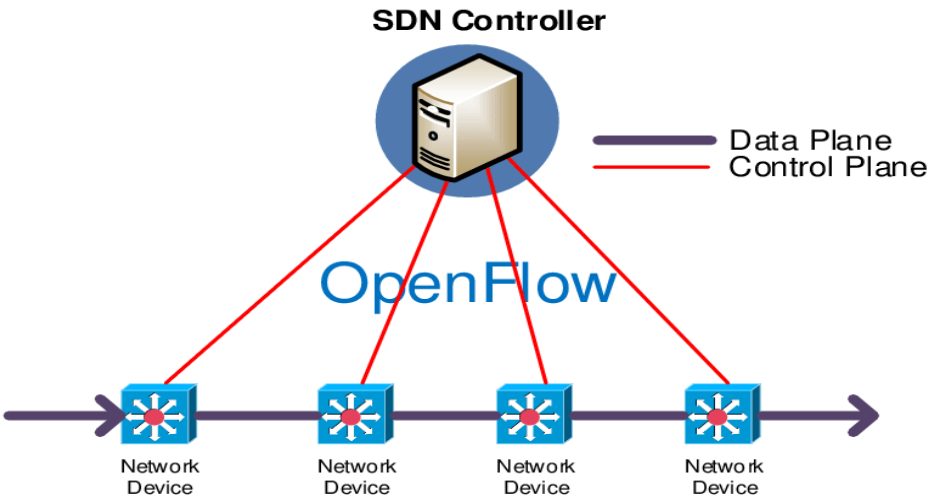
3.2 Αρχιτεκτονική



Εικόνα 5: Αρχιτεκτονική του OpenFlow [18]

Στο SDN, οι συσκευές δικτύου παίζουν έναν απλό ρόλο προώθησης πακέτων, ενώ η νοημοσύνη ή οι λογικές ελέγχου εφαρμόζονται στον ελεγκτή, επομένως, το SDN μπορεί να αυξήσει την ευελιξία της ανάπτυξης του δικτύου και την προγραμματιζόμενη δυνατότητα καθώς και την αξιοπιστία των λειτουργιών και των συσκευών ελέγχου δικτύου. Στην εικόνα 5 βλέπουμε την αρχιτεκτονική του OpenFlow, που περιέχει έναν ελεγκτή OpenFlow για τη λήψη αποφάσεων ροής για μεταγωγείς OpenFlow, ο οποίος διαχειρίζεται πρωτόγονες λειτουργίες δικτύου προγραμματίζοντας τον πίνακα ροής ενός μεταγωγέα OpenFlow, που σύμφωνα με τις πληροφορίες στον πίνακα ροής, κάθε μεταγωγέας OpenFlow μπορεί να προωθήσει εισερχόμενα πακέτα. Ο πίνακας ροής στον μεταγωγέα OpenFlow είναι για να καταγράφει καταχωρήσεις ροής για τις αντίστοιχες εντολές δράσης, οι οποίες μπορούν να προστεθούν και να αφαιρεθούν δυναμικά με βάση την εντολή από τον ελεγκτή OpenFlow. Ως εκ τούτου, η τοπολογία του δικτύου και η κατάσταση ολόκληρου του δικτύου μπορούν να παρακολουθούνται από τον ελεγκτή OpenFlow [18].

3.2.1 OpenFlow ελεγκτής



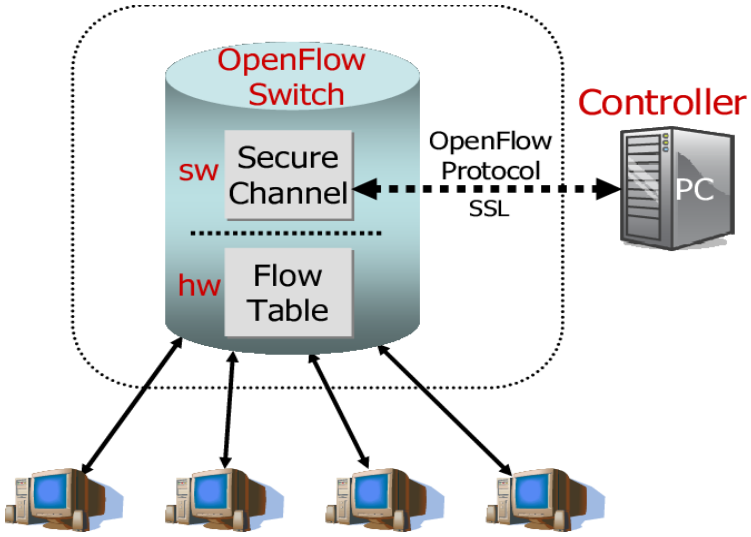
Εικόνα 6: OpenFlow ελεγκτής [19]

Ο ελεγκτής OpenFlow είναι ένας τύπος ελεγκτή SDN που χρησιμοποιεί το πρωτόκολλο OpenFlow, αυτός ο ελεγκτής είναι το στρατηγικό σημείο στο δίκτυο που καθορίζεται από το λογισμικό και χρησιμοποιεί το πρωτόκολλο OpenFlow για την σύνδεση και ρύθμιση παραμέτρων των συσκευών δικτύου όπως (δρομολογητές, μεταγωγείς κ.λπ.) για τον καθορισμό της καλύτερης διαδρομής για την κυκλοφορία των εφαρμογών. Επιπροσθέτως, οι ελεγκτές SDN μπορούν να απλοποιήσουν τη διαχείριση του δικτύου, χειρίζοντας όλες τις επικοινωνίες μεταξύ εφαρμογών και συσκευών για αποτελεσματική διαχείριση και τροποποίηση ροών δικτύου για την κάλυψη μεταβαλλόμενων αναγκών. Όταν το επίπεδο ελέγχου εφαρμόζεται σε λογισμικό, και όχι σε υλικολογισμικό, οι διαχειριστές μπορούν να διαχειρίζονται την κίνηση του δικτύου πιο δυναμικά και σε πιο αναλυτικό επίπεδο, χρειάζεται επίσης να σημειωθεί ότι ένας ελεγκτής SDN μεταδίδει πληροφορίες στους μεταγωγείς / δρομολογητές (μέσω του southbound API) και στις εφαρμογές (μέσω του northbound API). Συγκεκριμένα, οι ελεγκτές OpenFlow δημιουργούν ένα κεντρικό σημείο ελέγχου για να επιβλέπουν μια ποικιλία στοιχείων δικτύου με δυνατότητα OpenFlow και έχουν σχεδιαστεί για να αυξάνουν την ευελιξία εξαλείφοντας ιδιόκτητα πρωτόκολλα από προμηθευτές υλικού [20]. Ο πίνακας 2 παρουσιάζει μερικούς γνωστούς ελεγκτές OpenFlow και τα κύρια χαρακτηριστικά τους:

Ελεγκτές	**Γλώσσα προγραμματισμού**	**Χαρακτηριστικά**
NOX	C++ - Python	Γρήγορη υποστήριξη, ασύγχρονη είσοδος/έξοδος
POX	Python	Έχει καλή απόδοση σε σύγκριση με τις εφαρμογές NOX που είναι γραμμένες στο Python (ειδικά όταν λειτουργεί με το PyPy)
Beacon	Java	Ανεξάρτητο πλατφόρμας, δομοστοιχειωτό, ελεγκτής που βασίζεται σε Java που υποστηρίζει λειτουργίες που βασίζονται σε συμβάντα και σε νήματα
Floodlight	Java	Μπορεί να διαχειριστεί τόσο το OpenFlow όσο και τα μη OpenFlow δίκτυα
Maestro	Java	Παρέχει αφαίρεση προβολής σε ομαδικές καταστάσεις δικτύου σε ένα υποσύνολο
Trema	Ruby/C	Ένα πλαίσιο προγραμματισμού πλήρους στοίβας που επιτρέπει στους χρήστες να αναπτύξουν και να δοκιμάσουν ελεγκτές OpenFlow σε φορητό υπολογιστή
OpenDayLight	Java	Χρησιμοποιεί το πλαίσιο OSGi και παρέχει REST API
Ryu	Python	Μπορεί να ενσωματωθεί στο OpenStack, δημιουργεί ένα εικονικό δίκτυο χωρίς τη χρήση VLAN

Πίνακας 2: Γνωστοί ελεγκτές OpenFlow και τα κύρια χαρακτηριστικά τους [14]

3.2.2 OpenFlow μεταγωγέας



Εικόνα 7: OpenFlow μεταγωγέας [21]

Ο μεταγωγέας OpenFlow είναι ένας μεταγωγέας δεδομένων με δυνατότητα OpenFlow που επικοινωνεί μέσω ενός καναλιού OpenFlow με έναν εξωτερικό ελεγκτή, πέρα από αυτό εκτελεί αναζήτηση πακέτων και προωθεί σύμφωνα με έναν ή περισσότερους πίνακες ροής (flow tables) και έναν πίνακα ομάδας (group table), για την ακρίβεια επικοινωνεί με τον ελεγκτή και ο ελεγκτής διαχειρίζεται τον μεταγωγέα μέσω του πρωτοκόλλου μεταγωγέα OpenFlow. Κατά συνέπεια μπορεί να λειτουργήσει μόνο με τη συνεργαζόμενη εργασία τριών βασικών στοιχείων:

- Τους πίνακες ροής που είναι εγκατεστημένοι στους μεταγωγείς.
- Τον ελεγκτή και ένα ιδιόκτητο πρωτόκολλο OpenFlow για τον ελεγκτή για να μιλά με ασφάλεια με τους μεταγωγείς.

Οι πίνακες ροής έχουν ρυθμιστεί στους μεταγωγείς και οι ελεγκτές μιλούν στους μεταγωγείς μέσω του πρωτοκόλλου OpenFlow και επιβάλλουν πολιτικές στις ροές, επιπλέον ο ελεγκτής μπορεί να ρυθμίσει διαδρομές μέσω του δικτύου που έχουν βελτιστοποιηθεί για συγκεκριμένα χαρακτηριστικά, όπως ταχύτητα, λιγότερος αριθμός hops ή μειωμένη καθυστέρηση [21].

Κεφάλαιο 4: Ανίχνευση τοπολογίας

4.1 Βασικά χαρακτηριστικά τοπολογίας ενός δικτύου

Το βασικό χαρακτηριστικό της τοπολογίας ενός δικτύου είναι το πως είναι συνδεδεμένες οι συσκευές μεταξύ τους καθώς και ο τρόπος μετάδοσης των δεδομένων από τον ένα κόμβο στον άλλο [23]. Ένα δίκτυο μπορεί να έχει μία φυσική τοπολογία και πολλαπλές λογικές τοπολογίες ταυτόχρονα. Η φυσική διάταξη του δικτύου αναφέρεται στις φυσικές συνδέσεις των συσκευών, όπως μέσα μετάδοσης, καλώδια κ.λπ. Εργασίες όπως η παροχή, η ρύθμιση και η συντήρηση απαιτούν πληροφορίες σχετικά με τη φυσική διάταξη. Η λογική διάταξη του δικτύου, αφετέρου, είναι μια εννοιολογική αναπαράσταση του τρόπου λειτουργίας διαφόρων συσκευών σε διαφορετικά επίπεδα αφαίρεσης όπου παρέχει πληροφορίες σχετικά με τη διασύνδεση πολλαπλών κόμβων, τον τρόπο μετάδοσης των δεδομένων και το μέσο μετάδοσης. Οι οργανισμοί ανάλογα με την καταλληλότητα των διαφόρων λειτουργιών, το συνολικό μέγεθος του δικτύου και τους επιχειρηματικούς στόχους μπορούν να επιλέξουν διαφορετικούς τύπους τοπολογιών δικτύου όπως για παράδειγμα: τοπολογία διαύλου, δακτυλίου, αστέρα, πλέγματος, δένδρου και υβριδική [24].

4.2 Η τοπολογία στα δίκτυα SDN

Η διαχείριση της τοπολογίας είναι ένα μοναδικό χαρακτηριστικό του SDN σε σύγκριση με τα παραδοσιακά δίκτυα. Ο διαχωρισμός του επιπέδου ελέγχου από το επίπεδο δεδομένων επιτρέπει στο SDN να έχει έναν λογικά συγκεντρωτικό έλεγχο του δικτύου, παράλληλα για να επιτευχθεί ο κεντρικός έλεγχος, χρειάζεται ένας ελεγκτής (ο οποίος είναι υπεύθυνος για τον κεντρικό έλεγχο του δικτύου) που θα πρέπει να έχει καθολική προβολή του πλήρους δικτύου. Ένας ελεγκτής ενσωματώνει διάφορες βασικές ενότητες που βοηθούν στην εκτέλεση διαφόρων εφαρμογών SDN. Μεταξύ των βασικών ενοτήτων, μια διαχείριση τοπολογίας δημιουργεί μια τοπολογία ολόκληρης της υποδομής SDN. Η τοπολογία όχι μόνο διευκολύνει τον ελεγκτή, αλλά επίσης βοηθά την υπηρεσία επιπέδου εφαρμογής να εκτελεί τη λειτουργία της χρησιμοποιώντας την προγραμματιζόμενη δυνατότητα δικτύου. Η τοπολογία του δικτύου είναι σημαντική τόσο για το επίπεδο ελέγχου όσο και για το επίπεδο εφαρμογής, επειδή παρέχει μια αφηρημένη ορατότητα ολόκληρων των συσκευών δικτύου. Το πρωτόκολλο OpenFlow είναι μια τυπική προσέγγιση που χρησιμοποιείται για την επικοινωνία μεταξύ του ελεγκτή και των μεταγωγέων OpenFlow στη διεπαφή southbound του SDN. Η διεπαφή southbound μεταφέρει αιτήματα και απαντήσεις τόσο

στους ελεγκτές όσο και στους μεταγωγείς OpenFlow. Οι ενημερωμένες πληροφορίες τοπολογίας δικτύου είναι σημαντικές για τον ελεγκτή στην παροχή αποτελεσματικού ελέγχου και διαχείρισης του δικτύου. Ως αποτέλεσμα, η ανακάλυψη της τοπολογίας θεωρείται ένα σημαντικό χαρακτηριστικό για τον ελεγκτή. Στον πίνακα 3 βλέπουμε μερικά παραδείγματα που χρησιμοποιούνται στην ανίχνευση τοπολογίας από τα παραδοσιακά δίκτυα σε αντίθεση με τα δίκτυα SDN [22].

Χαρακτηριστικά	**Ανακάλυψη τοπολογίας σε παραδοσιακά δίκτυα**	**Ανακάλυψη τοπολογίας σε δίκτυα καθορισμένα από λογισμικό**
Host Discovery	NMAP	Packet_In message
Switch Discovery	SNMP	Αρχική διαδικασία χειραψίας
Link Discovery	RIP, OSPF, LSA, OLSR	LLDP
Διαχείριση ελέγχου	Ανεξάρτητο	Ελεγκτής
Επεκτασιμότητα	Αριθμός μεταγωγέων	Αριθμός OF μεταγωγέων
Ενημερώσεις επικοινωνίας	Switch – Switch	Controller-Switch-Controller

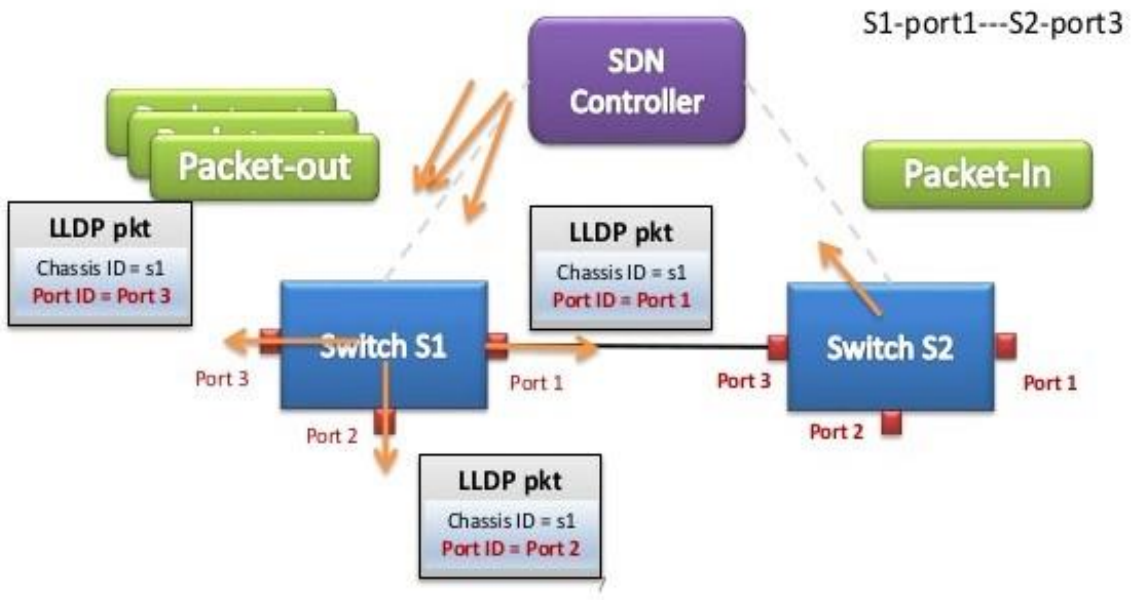
Πίνακας 3: Σύγκριση μεταξύ ενός παραδοσιακού δικτύου με την ανίχνευση της τοπολογίας SDN [22]

4.3 Ανίχνευση τοπολογίας

Η ανίχνευση τοπολογίας δικτύου είναι μια αναπαράσταση του τρόπου με τον οποίο οι συσκευές σε ένα δίκτυο η υποδίκτυο είναι φυσικά συνδεδεμένες [23]. Η αυτόματη δημιουργία μιας προβολής της τοπολογίας του δικτύου είναι μια κρίσιμη υπηρεσία που ο ελεγκτής πρέπει να διασφαλίσει για την ορθή λειτουργία άλλων υπηρεσιών και εφαρμογών δικτύου, καθώς λόγω των διαφορετικών λειτουργικών επιπέδων που μπορούν να μοντελοποιήσουν ένα δίκτυο υπολογιστών, η ανίχνευση της τοπολογίας του δικτύου μπορεί να πραγματοποιηθεί σε διαφορετικά επίπεδα του

ιδίου. Έτσι η φυσική τοπολογία μας παρέχει έναν χάρτη όπου μπορεί κανείς να εκτιμήσει την κατανομή των κόμβων και τις συνδέσεις τους στο δίκτυο, ενώ η λογική τοπολογία δείχνει τη ροή δεδομένων μεταξύ των συσκευών σύμφωνα με τα πρωτόκολλα που χρησιμοποιούνται στα διάφορα λειτουργικά επίπεδα. Ως ανίχνευση τοπολογίας θεωρούμε τις εργασίες που εκτελούνται από τον ελεγκτή για την ανίχνευση των μεταγωγέων (switches), των συνδέσμων (links) και των κεντρικών υπολογιστών (hosts) σε έναν ενιαίο τομέα διαχείρισης OpenFlow [8].

4.3.1 OFDP – OpenFlow Discovery Protocol



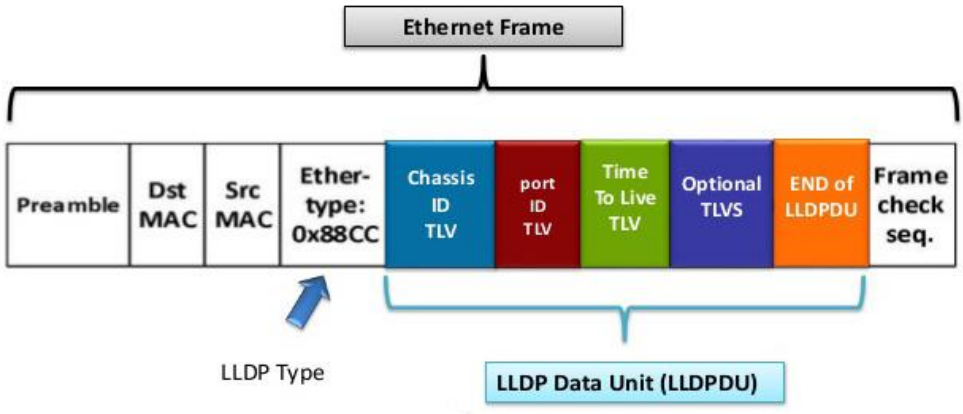
Εικόνα 8: Παράδειγμα βασικού σεναρίου OFDP [24]

Κάθε μήνυμα απασχολεί μια διαφορετική εργασία, ο ελεγκτής ζητά πληροφορίες σχετικά με τη διαμόρφωση από μεταγωγείς όπως λειτουργικές θύρες (operational ports) και τις διευθύνσεις των MAC τους χρησιμοποιώντας το μήνυμα OFPT_FEATURES_REQUEST. Αν και οι μεταγωγείς ανακοινώνουν την ύπαρξή τους και τις λειτουργικές θύρες τους στον ελεγκτή, ωστόσο, ο ελεγκτής δεν έχει ιδέα για τους λειτουργικούς συνδέσμους (operational links) και πώς συνδέονται. Στο τρέχον πρωτόκολλο OFDP, μετά τη δημιουργία σύνδεσης μεταξύ μεταγωγέων και ελεγκτή, ο ελεγκτής στέλνει ένα μήνυμα OFTP_PACKET_OUT σε κάθε θύρα σε κάθε μεταγωγέα. Σε αυτό το παράδειγμα, ο ελεγκτής SDN στέλνει τρία πακέτα LLDP σε κάθε θύρα του μεταγωγέα S1

χρησιμοποιώντας το μήνυμα OFTP_PACKET_OUT. Το μήνυμα που περιλαμβάνεται στο πακέτο LLDP περιέχει οδηγίες για την κατεύθυνση του πακέτου στη σχετική θύρα. Για παράδειγμα, το πακέτο LLDP που έχει θύρα ID 1 αποστέλλεται στη θύρα ID 1, το πακέτο που έχει θύρα ID 2 αποστέλλεται στη θύρα 2 και ούτω καθεξής. Επιπλέον, όλοι οι μεταγωγείς δικτύου περιέχουν έναν κανόνα για την αποστολή κάθε πακέτου LLDP που δεν προέρχεται από τον ελεγκτή, αυτή η διαδικασία γίνεται χρησιμοποιώντας το μήνυμα OFTP_PACKET_IN. Σύμφωνα με τους ενσωματωμένους κανόνες προώθησης στους μεταγωγείς, ο μεταγωγέας S2 στέλνει πακέτο LLDP με μήνυμα OFTP_PACKET_IN στον ελεγκτή. Αυτό το πακέτο περιλαμβάνει το αναγνωριστικό μεταγωγέα (switch ID) και το αναγνωριστικό θύρας (port ID) εισαγωγής από το οποίο λαμβάνεται το πακέτο. Λόγω αυτών των πληροφοριών, ο ελεγκτής μπορεί να καθορίσει ότι ένας μεταγωγέας συνδέσεων καναλιού λειτουργίας (S1, Port1) και (S2, Port3) πως οι πληροφορίες σχετικά με τη σύνδεση αποθηκεύονται σε μια βάση δεδομένων που περιέχει πληροφορίες σχετικά με την τοπολογία. Αυτή η διαδικασία εκτελείται επαναληπτικά για κάθε λειτουργικό μεταγωγέα στο δίκτυο SDN. Προκειμένου να διατηρούνται πάντα ενημερωμένες οι πληροφορίες σχετικά με την τοπολογία, ο ελεγκτής πραγματοποιεί περιοδική ανακάλυψη συνδέσμου σε σταθερά χρονικά διαστήματα. Λόγω του γεγονότος ότι η ανακάλυψη της τοπολογίας του δικτύου SDN είναι μια υπηρεσία που συμβαίνει συνεχώς στο παρασκήνιο έτσι για να προσδιοριστεί το φορτίο του ελεγκτή, είναι απαραίτητο να γνωρίζουμε την ποσότητα των μηνυμάτων OFTP_PACKET_OUT που στέλνονται και απαιτούνται για την επεξεργασία των μηνυμάτων OFTP_PACKET_IN. Ο αριθμός των μηνυμάτων OFTP_PACKET_IN μπορεί να προσδιοριστεί με διπλάσιο αριθμό συνδέσμων στην τοπολογία του δικτύου. Ο αριθμός των μηνυμάτων PACKET_IN μπορεί να υπολογιστεί σύμφωνα με την εξίσωση $P_{in\_OFDP} = 2L$ όπου P_{in_OFDP} είναι ο αριθμός των μηνυμάτων OFTP_PACKET_IN σε κάθε κύκλο ανακάλυψης και L είναι ο αριθμός των λειτουργικών καναλιών δικτύου στο δίκτυο. Ενώ, η εύρεση μηνυμάτων OFTP_PACKET_OUT απαιτεί τον ακόλουθο τύπο: $P_{out\_OFDP} = \sum_{i=1}^{N} Pi$ όπου P_{out_OFDP} είναι τα μηνύματα PACKET_OUT που στέλνει ο ελεγκτής, N είναι ο αριθμός των μεταγωγέων και το Pi είναι ο αριθμός των θυρών σε κάθε μεταγωγέα. Δεν είναι αποτελεσματικό να στείλουμε ένα πακέτο για κάθε θύρα του δικτύου, μια καλύτερη λύση είναι να στείλουμε μόνο ένα πακέτο OFTP_PACKET_OUT ανά μεταγωγέα στο δίκτυο. Επίσης, το OpenFlow μπορεί να υποστηρίξει αυτήν τη δυνατότητα χωρίς καμία τροποποίηση από το ίδιο το πρωτόκολλο [25].

4.3.2 LLDP – Link Layer Discovery Protocol

Το Link Layer Discovery Protocol (LLDP) τυποποιήθηκε για πρώτη φορά το 2005 από το Ινστιτούτο Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών (IEEE), με το όνομα IEEE 802.1AB, όμως το 2009 αντικαταστάθηκε και τυποποιήθηκε επίσημα από μια νέα έκδοση, που αναφέρεται επίσημα από την IEEE ως "Station and Media Access Control Connectivity Discovery" που ορίζεται στα πρότυπα IEEE ως έγγραφο 802.1AB IEEE-2009. Το LLDP είναι ένα γειτονικό πρωτόκολλο ανακάλυψης ενός άλματος, δηλαδή ¨διαφημίζει¨ την ταυτότητα και τις δυνατότητές του και λαμβάνει τις ίδιες πληροφορίες από τους παρακείμενους μεταγωγείς. Επιπλέον είναι ένα ουδέτερο πρωτόκολλο προμηθευτή που λειτουργεί στο επίπεδο 2 του μοντέλου OSI και βασίζεται σε έννοιες πολλών ιδιόκτητων πρωτοκόλλων ανακάλυψης: όπως το Cisco Discovery Protocol (CDP), το Nortel Discovery Protocol (NDP), το Extreme Discovery Protocol (EDP) κλπ. [8].



Εικόνα 9: Δομή του πλαισίου LLDP [24]

Κάθε πλαίσιο LLDP αποτελείται από μια κεφαλίδα (header) και μια μονάδα δεδομένων LLDP ωφέλιμου φορτίου (payload) που ονομάζεται (LLDPDU) όπως φαίνεται στην εικόνα 9. Η προεπιλεγμένη τιμή του πεδίου Ethertype στην κεφαλίδα για κάθε μήνυμα LLDP είναι 0x88cc, αυτές οι πληροφορίες είναι ζωτικής σημασίας για την αποτελεσματική αναγνώριση των πακέτων εντοπισμού σε δίκτυα OpenFlow. Στο πεδίο προορισμού MAC (destination MAC) ορίζεται μια διεύθυνση MAC προορισμού πολλαπλής διανομής, αυτή η διεύθυνση τυποποιείται ως “Διεύθυνση πολλαπλής διανομής LLDP” (LLDP Multicast address) και επιτρέπει στους παραδοσιακούς μεταγωγείς να αναγνωρίζουν τα LLDP πακέτα ανακάλυψης. Το LLDPDU αποτελείται από προαιρετικές και υποχρεωτικές δομές τιμής μήκους τύπου (TLV). Το ωφέλιμο φορτίο ξεκινά με τρία υποχρεωτικά TLV, ακολουθούμενο από έναν αριθμό προαιρετικών TLV και τελειώνει με ένα

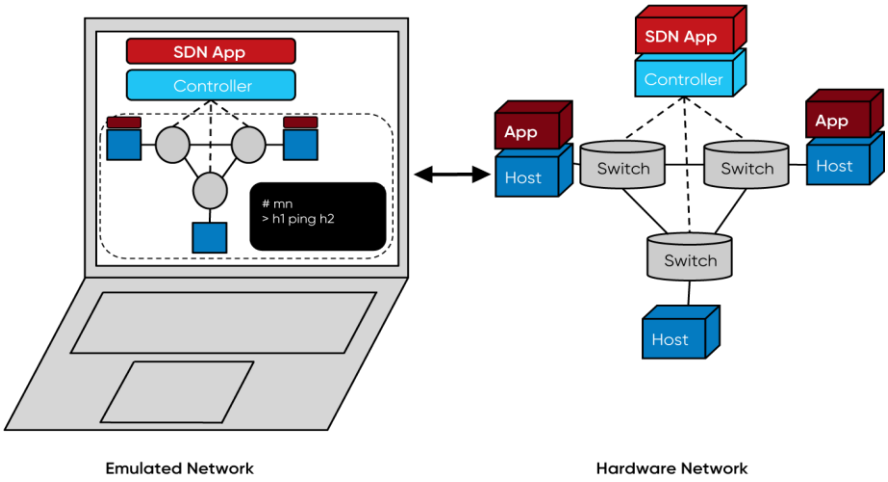
ειδικό υποχρεωτικό TLV στο οποίο τα πεδία τύπου και μήκους είναι μηδέν. Το προαιρετικό TLV περιλαμβάνει το βασικό σύνολο TLV και τα οργανικά ειδικά TLV και μπορεί να χρησιμοποιηθεί για την εισαγωγή νέων χαρακτηριστικών ανακάλυψης μέσω του πρωτοκόλλου LLDP. Τα τέσσερα υποχρεωτικά TLV στο πλαίσιο αποτελούν τον πυρήνα της λειτουργίας του πρωτοκόλλου ανακάλυψης και περιγράφονται εν συντομία παρακάτω:

- Chassis ID (Τύπος 1): Αυτή η δομή TLV περιέχει το αναγνωριστικό του μεταγωγέα που στέλνει το πακέτο LLDP.
- Port ID (Τύπος 2): Αυτή η δομή TLV περιέχει το αναγνωριστικό της θύρας μέσω της οποίας αποστέλλεται το πακέτο LLDP.
- Time to Live (Τύπος 3): Αυτή η δομή TLV περιέχει την τιμή του χρόνου σε δευτερόλεπτα κατά την οποία οι πληροφορίες που λαμβάνονται στο πακέτο LLDP πρόκειται να είναι έγκυρες.
- Τέλος του LLDPDU (Τύπος 4): Ειδική δομή TLV που δείχνει το τέλος του ωφέλιμου φορτίου (payload) στο πλαίσιο LLDP.

Στα παραδοσιακά δίκτυα, κάθε πλαίσιο LLDP αποστέλλεται από τους μεταγωγείς που υποστηρίζουν και έχουν ενεργοποιήσει αυτήν τη λειτουργικότητα σε σταθερά χρονικά διαστήματα, που έχουν διαμορφωθεί από τον διαχειριστή του δικτύου αντίθετα στα δίκτυα που βασίζονται στο OpenFlow, οι μεταγωγείς στέλνουν τα μηνύματα LLDP για να ανακαλύψουν την υποκείμενη τοπολογία κατόπιν αιτήματος του ελεγκτή [8].

Κεφάλαιο 5: Προσομοίωση ενός SDN δικτύου χρησιμοποιώντας το Mininet και τον POX ελεγκτή

5.1 Το λογισμικό Mininet



Εικόνα 10: Το λογισμικό Mininet [26]

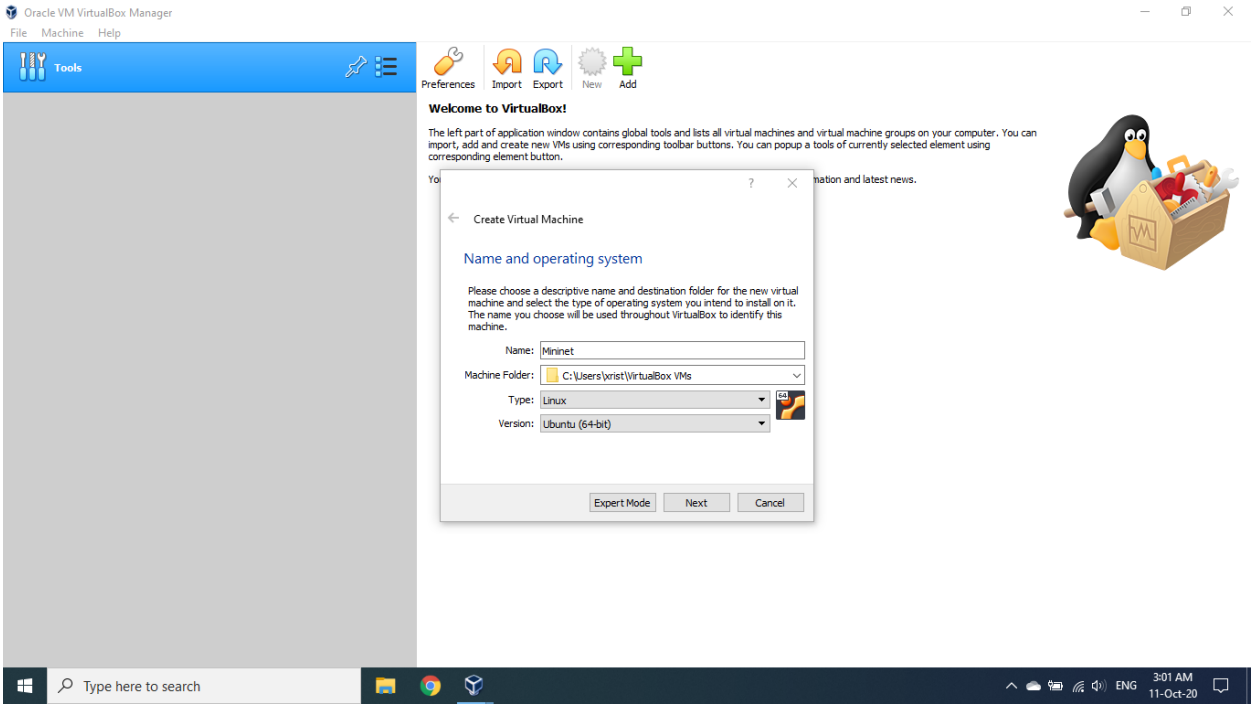
Το Mininet είναι ένας εξομοιωτής δικτύου για τη δημιουργία μεγάλων δικτύων σε έναν κεντρικό υπολογιστή, που έχει γραφτεί σε γλώσσα Python, παρέχει επεκτάσιμο Python API και είναι μια προσπάθεια ανοιχτού κώδικα που ιδρύθηκε από τον Bob Lants, τρέχει σε Linux αλλά μπορεί να χρησιμοποιηθεί και σε άλλα λειτουργικά συστήματα μέσω του Mininet VM. Ακόμη δημιουργεί εικονικούς υπολογιστές, συνδέσμους, μεταγωγείς και ελεγκτές για να μιμηθούν ένα πλήρες δίκτυο. Εκτελεί ελαφριά εικονικοποίηση σε επίπεδο λειτουργικού συστήματος, συμπεριλαμβανομένης της εικονικοποίησης χώρου ονομάτων διεργασιών και δικτύου, επιπλέον μπορεί να συνδεθεί σε ένα πραγματικό δίκτυο και να επιτρέψει στους χρήστες να αλληλοεπιδράσουν με το δίκτυο mininet χρησιμοποιώντας το CLI του. Δεδομένου ότι το mininet εκτελεί πραγματικό κώδικα, εφαρμογές που αναπτύχθηκαν στο mininet μπορούν να εξαχθούν και σε πραγματικά δίκτυα με ελάχιστη αλλαγή. Χρειάζεται επίσης να σημειωθεί ότι μπορεί να χρησιμοποιηθεί για τη δοκιμή τόσο των παραδοσιακών όσο και των δικτύων που είναι καθοριζόμενα από λογισμικό [27].

5.1.1 Εγκατάσταση του Mininet

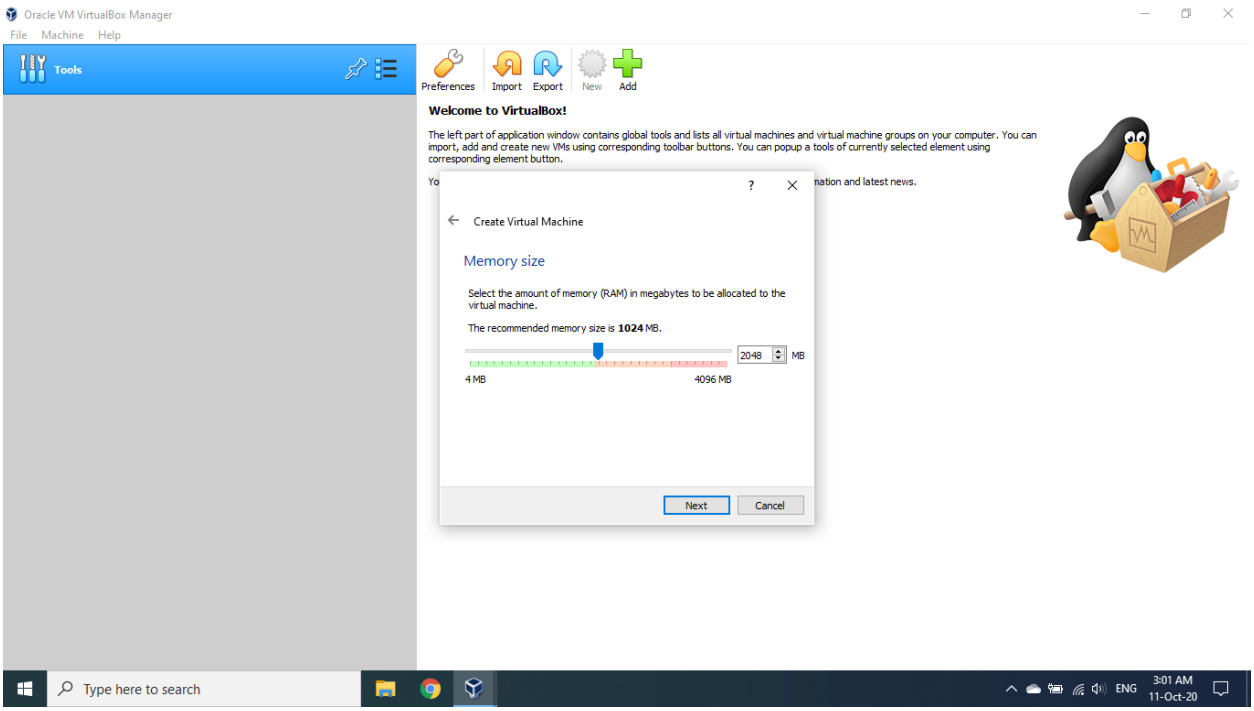
Για να ανιχνεύσουμε την τοπολογία στα δίκτυα που είναι καθοριζόμενα από λογισμικό αρχικά θα πρέπει να εγκαταστήσουμε την εικονική μηχανή (Virtual Box) και έπειτα να προσθέσουμε την εικονική μηχανή mininet σε αυτό.

Επίσης κατεβάζουμε το putty 

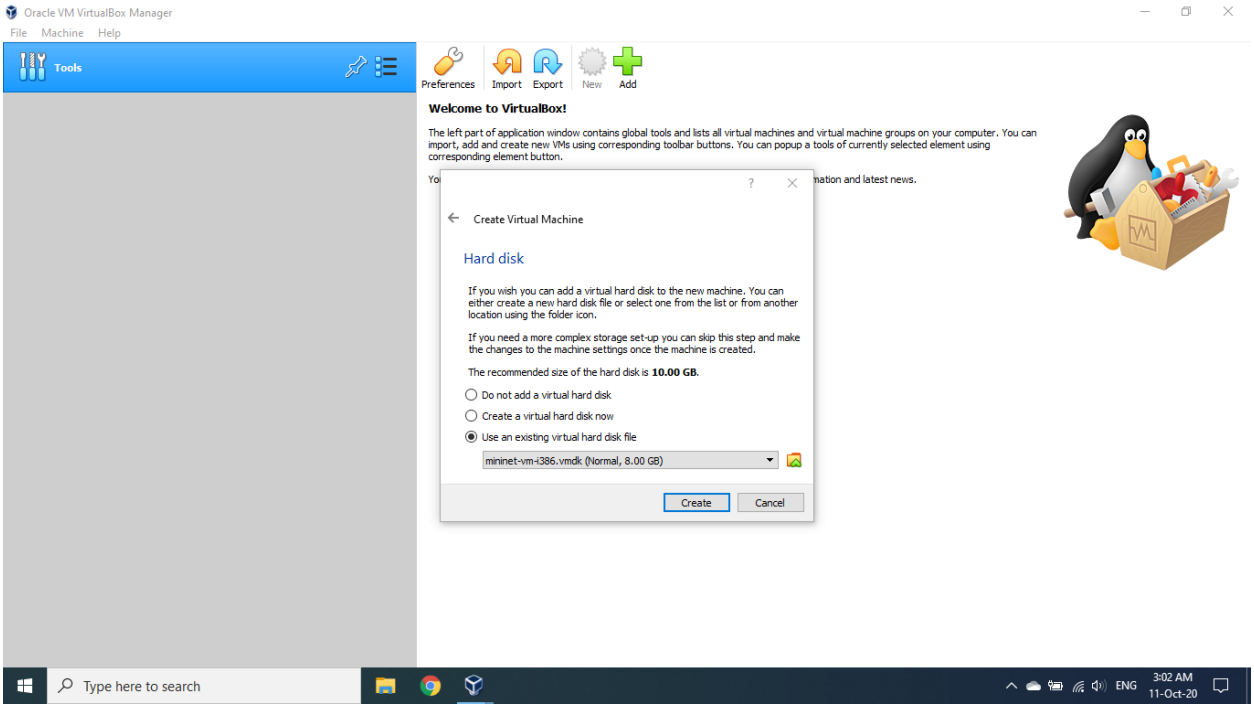
και το Xming 



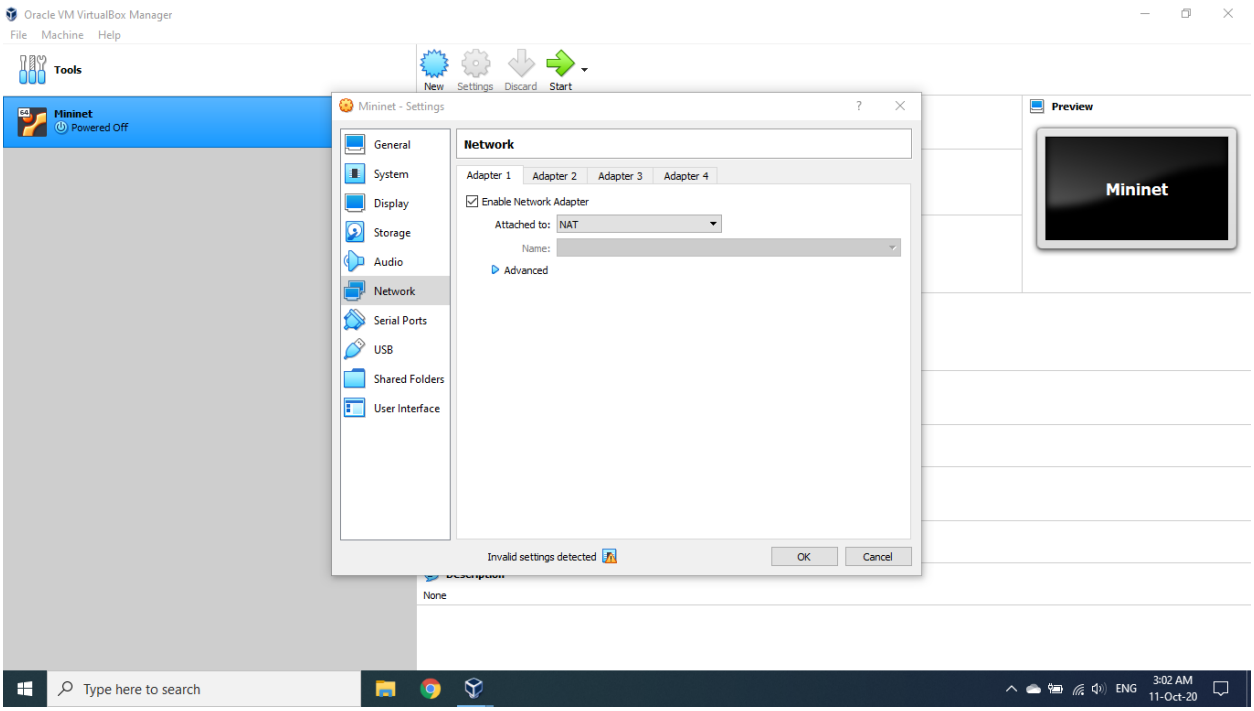
Αρχικά στο Virtual Box πατάμε στο εικονίδιο New για να εισάγουμε την εικονική μηχανή του mininet, έπειτα πληκτρολογούμε το όνομα και επιλέγουμε Type: Linux και Version :Ubuntu(64-bit).



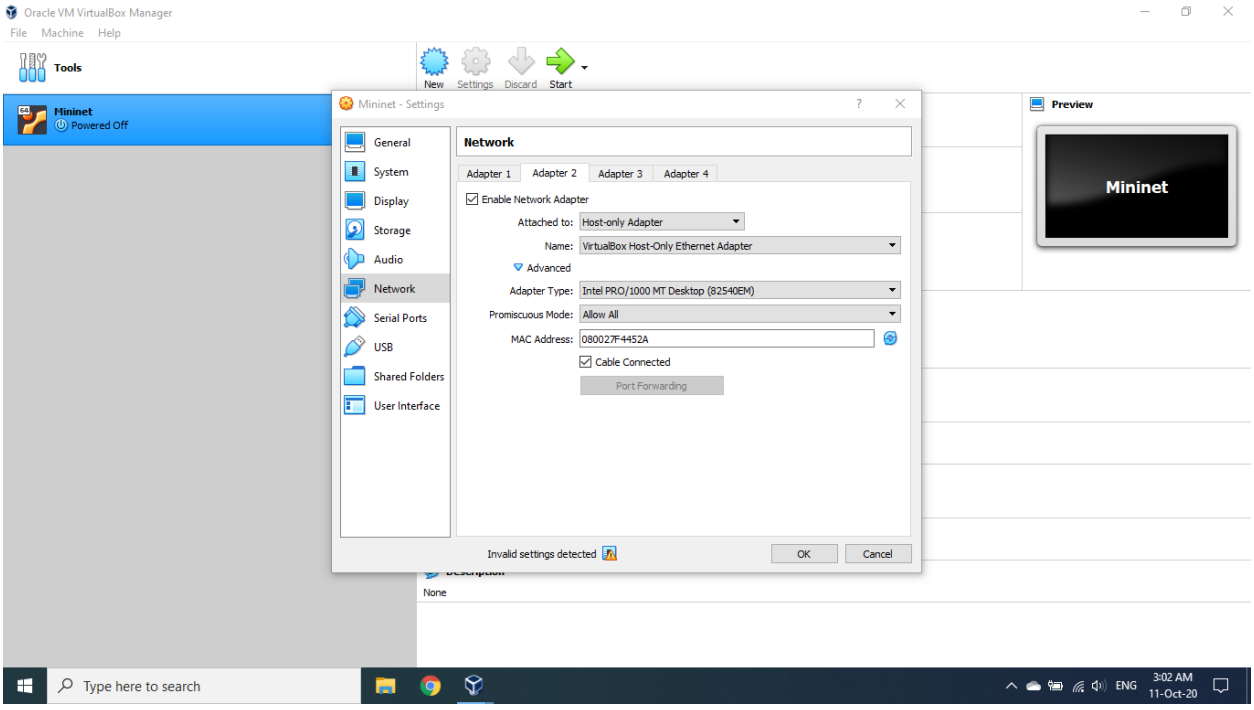
Στην συνέχεια, επιλέγουμε την μνήμη που χρειαζόμαστε στη περίπτωση μας διαλέγουμε 2048MB.



Επιλέγουμε Use an existing virtual hard disk file και εισάγουμε την εικονική μηχανή που ήδη έχουμε εγκαταστήσει στον υπολογιστή μας.

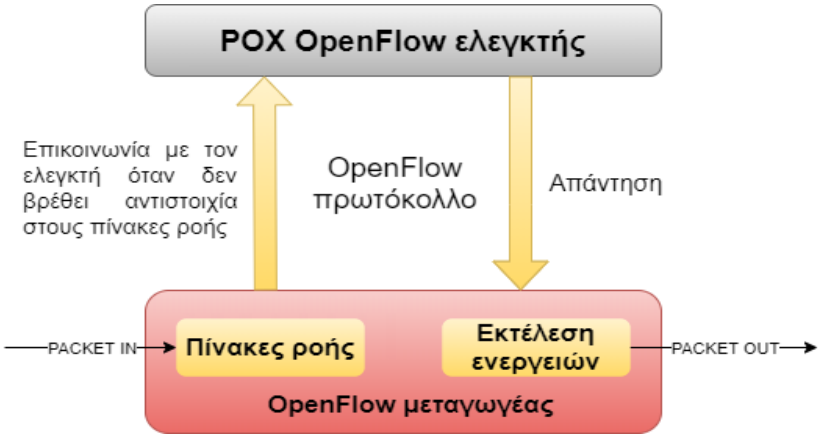


Στα settings ελέγχουμε στο Network το Adapter1 να είναι επιλεγμένο το NAT.



Και στο Adapter2, επιλέγουμε το Enable Network Adapter και διαλέγουμε το Host-only Adapter, ανοίγουμε το Advanced και στο Promiscuous Mode επιλέγουμε το Allow all, τέλος πατάμε το οκ.

5.2 POX ελεγκτής



Εικόνα 11: Αρχιτεκτονική POX ελεγκτή [28]

Ο POX είναι ένας ελεγκτής ανοιχτού κώδικα που διαθέτει υψηλό επίπεδο τοπολογίας με δυνατότητα αναζήτησης και υποστήριξη για εικονικοποίηση. Παρέχει επαναχρησιμοποιήσιμα στοιχεία, επιλογή διαδρομής, ανίχνευση τοπολογίας, εξισορρόπηση φορτίου κ.λπ. Είναι ο προεπιλεγμένος ελεγκτής που χρησιμοποιείται όταν ενεργοποιείται ένας ελεγκτής στο λογισμικό Mininet, επίσης υποστηρίζει γραφικό περιβάλλον εργασίας χρήστη (GUI) και εργαλεία οπτικοποίησης. Η αρχιτεκτονική του που βλέπουμε στην εικόνα 11 είναι απλή σε σύγκριση με άλλους ελεγκτές. Η επικοινωνία μεταξύ του ελεγκτή και των μεταγωγέων χρησιμοποιεί το πρωτόκολλο OpenFlow. Οι μεταγωγείς OpenFlow συμπεριφέρονται σαν συσκευές προώθησης και αποδίδουν μόνο σύμφωνα με τις οδηγίες του ελεγκτή, επιπρόσθετα κάθε μεταγωγέας έχει τον δικό του πίνακα ροής. Κατά την άφιξη ενός πακέτου, ο μεταγωγέας στέλνει ένα μήνυμα Packet_In στον ελεγκτή. Στη συνέχεια, ο ελεγκτής εισάγει ολόκληρη τη ροή στον πίνακα ροής του μεταγωγέα σχετικά με τον τρόπο χειρισμού του πακέτου. Η καταχώρηση ροής έχει 3 μέρη: τον κανόνα, την δράση και τους μετρητές. Έτσι, καθώς κάθε πακέτο περνά, μια καταχώρηση ροής εγκαθίσταται στον μεταγωγέα έτσι ώστε να μπορεί να χειριστεί το πακέτο χωρίς την παρέμβαση του ελεγκτή, εάν οι καταχωρήσεις ροής δεν ταιριάζουν με αυτήν στον ελεγκτή, στέλνει μια απάντηση απόρριψε το πακέτο. Το πλεονέκτημα της χρήσης του POX είναι ότι χρειάζεται λιγότερος χώρος μνήμης για να λειτουργήσει σε αντίθεση με άλλους ελεγκτές και το μειονέκτημα είναι ότι έχει χαμηλότερο ρυθμό απόδοσης [28].

5.3 Δημιουργία δικτύου

Για να δημιουργήσουμε ένα δίκτυο στο mininet χρειάζεται να γνωρίζουμε κάποιες βασικές εντολές οι οποίες αναφέρονται παρακάτω:

Για την σύνταξη των εντολών:

$: προηγείται των εντολών Linux που πρέπει να πληκτρολογούνται στη γραμμή εντολών.

mininet>: προηγείται των εντολών Mininet που πρέπει να πληκτρολογούνται στη γραμμή εντολών.

#: προηγείται των εντολών Linux που έχουν πληκτρολογηθεί με ειδικά δικαιώματα στη γραμμή εντολών [29]**.**

sudo: Το Sudo (superuser do) είναι ένα βοηθητικό πρόγραμμα για συστήματα που βασίζονται σε Unix και Linux, παρέχει έναν αποτελεσματικό τρόπο για να δοθεί σε συγκεκριμένους χρήστες άδεια να χρησιμοποιούν συγκεκριμένες εντολές συστήματος στο ριζικό (πιο ισχυρό) επίπεδο του συστήματος [30].

Γραμμή εντολών mininet:

Με την δημιουργία μιας μικρής τοπολογίας **sudo mn** έχουμε:

help: Εμφάνιση διαθέσιμων εντολών.

```
mininet> help

Documented commands (type help <topic>):
========================================
EOF    gterm  iperfudp  nodes        pingpair      py      switch
dpctl  help   link      noecho       pingpairfull  quit    time
dump   intfs  links     pingall      ports         sh      x
exit   iperf  net       pingallfull  px            source  xterm

You may also send a command to a node using:
  <node> command {args}
For example:
  mininet> h1 ifconfig

The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
  mininet> h2 ping h3
should work.

Some character-oriented interactive commands require
noecho:
  mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
  mininet> xterm h2
```

nodes: Εμφάνιση διαθέσιμων κόμβων.

```
mininet> nodes
available nodes are:
c0 h1 h2 s1
```

links: Εμφανίζει ποιοι υπολογιστές και μεταγωγείς είναι συνδεδεμένοι μεταξύ τους.

```
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
```

net: Είναι παρόμοιο με το link, δείχνει πώς τα στοιχεία δικτύου συνδέονται μεταξύ τους.

```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
```

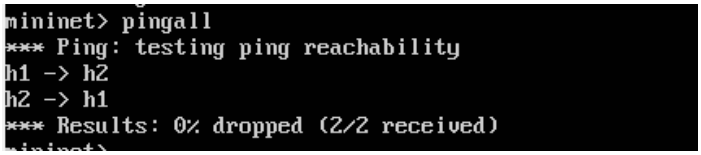
dump: Παραθέτει πληροφορίες σχετικά με τους κόμβους, τους μεταγωγείς και τους ελεγκτές.

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1601>
<Host h2: h2-eth0:10.0.0.2 pid=1603>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=1608>
<Controller c0: 127.0.0.1:6653 pid=1594>
```

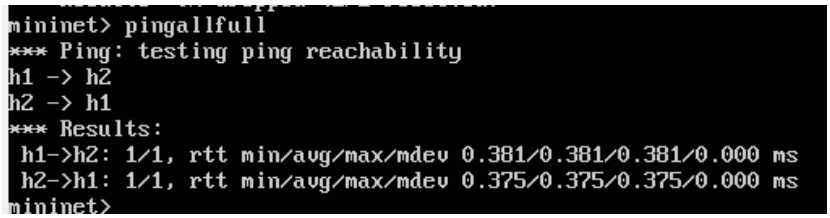
ping: Δείχνει τη συνδεσιμότητα μεταξύ συγκεκριμένων υπολογιστών, σε αυτό το παράδειγμα του h1 με τον h2.

```
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=3.56 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.535 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.509 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.042 ms
^C
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9002ms
rtt min/avg/max/mdev = 0.040/0.492/3.569/1.042 ms
```

pingall: Εμφανίζει τη συνδεσιμότητα μεταξύ όλων των υπολογιστών και μας λέει ποιοι υπολογιστές είναι συνδεδεμένοι μεταξύ τους.

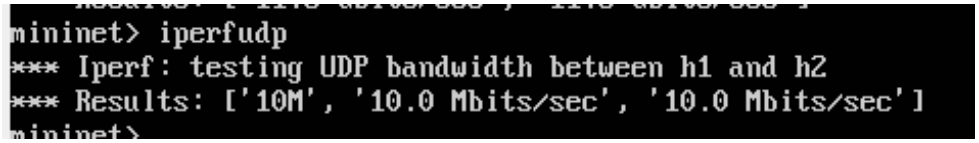


Pingallfull: Δείχνει περισσότερες λεπτομέρειες σχετικά με τον τρόπο σύνδεσης των υπολογιστών και εμφανίζει το ελάχιστο, το μέσο και το μέγιστο χρόνο μεταξύ δύο υπολογιστών σε ms.



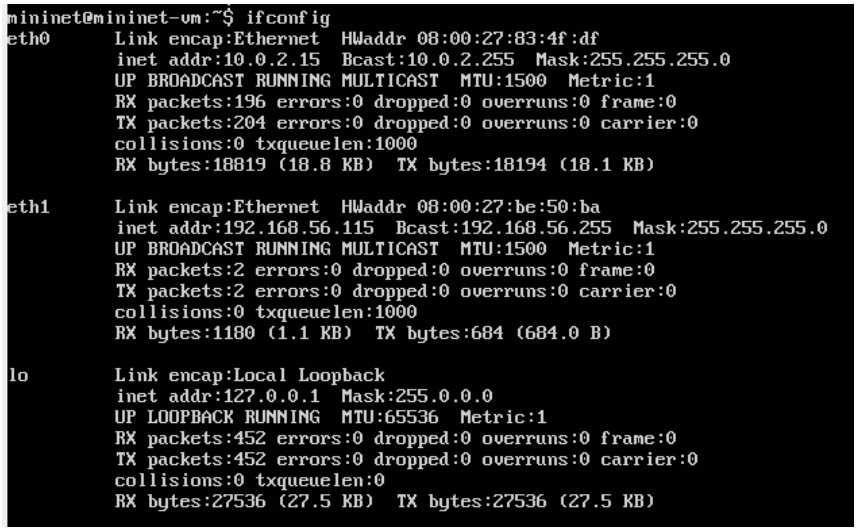
iperf: Χρησιμοποιείται γενικά για σύνδεση tcp όπως επίσης και για τη δοκιμή εύρους ζώνης μεταξύ των υπολογιστών.

iperfudp: Είναι παρόμοιο με το iperf, αλλά χρησιμοποιείται για συνδέσεις udp.

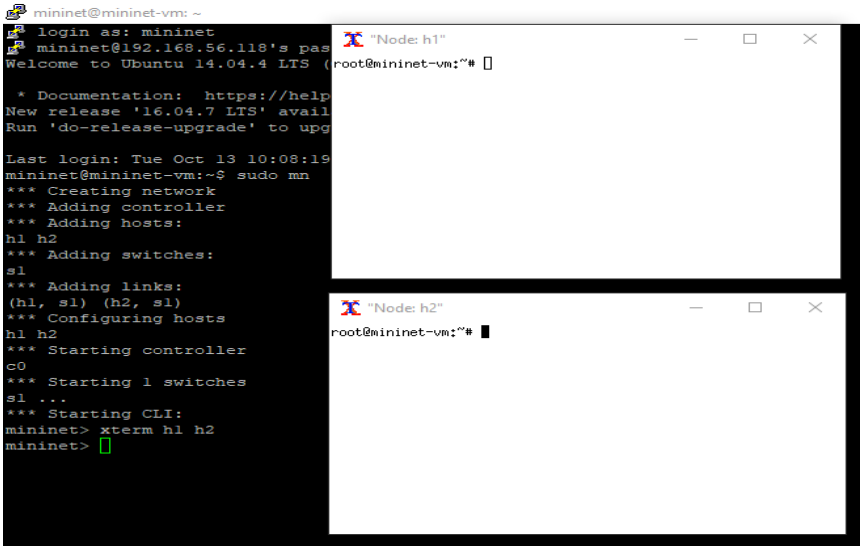


exit: Χρησιμοποιείται για έξοδο από την τοπολογία.

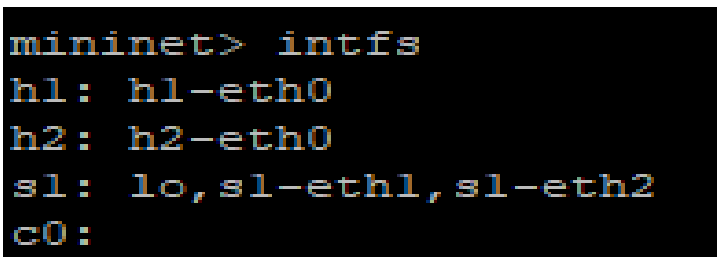
ifconfig: Χρησιμοποιείται για τον έλεγχο της διεύθυνσης IP (192.168.56.115) ενός εικονικού υπολογιστή.



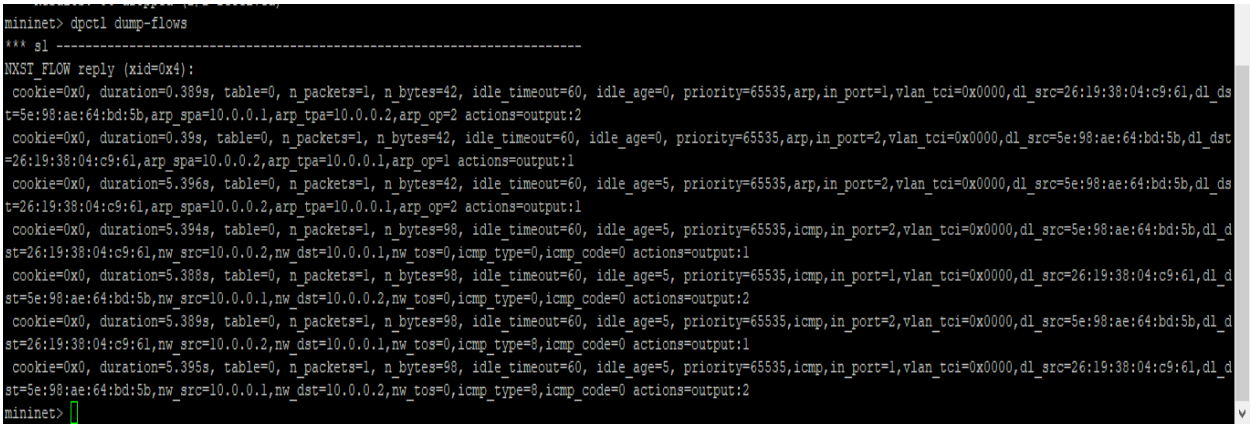
xterm: Χρησιμοποιείται για να ανοίξει ένα νέο παράθυρο συνδεδεμένο σε έναν κόμβο.



intfs: Εμφανίζει τις διασυνδέσεις που είναι συνδεδεμένες.



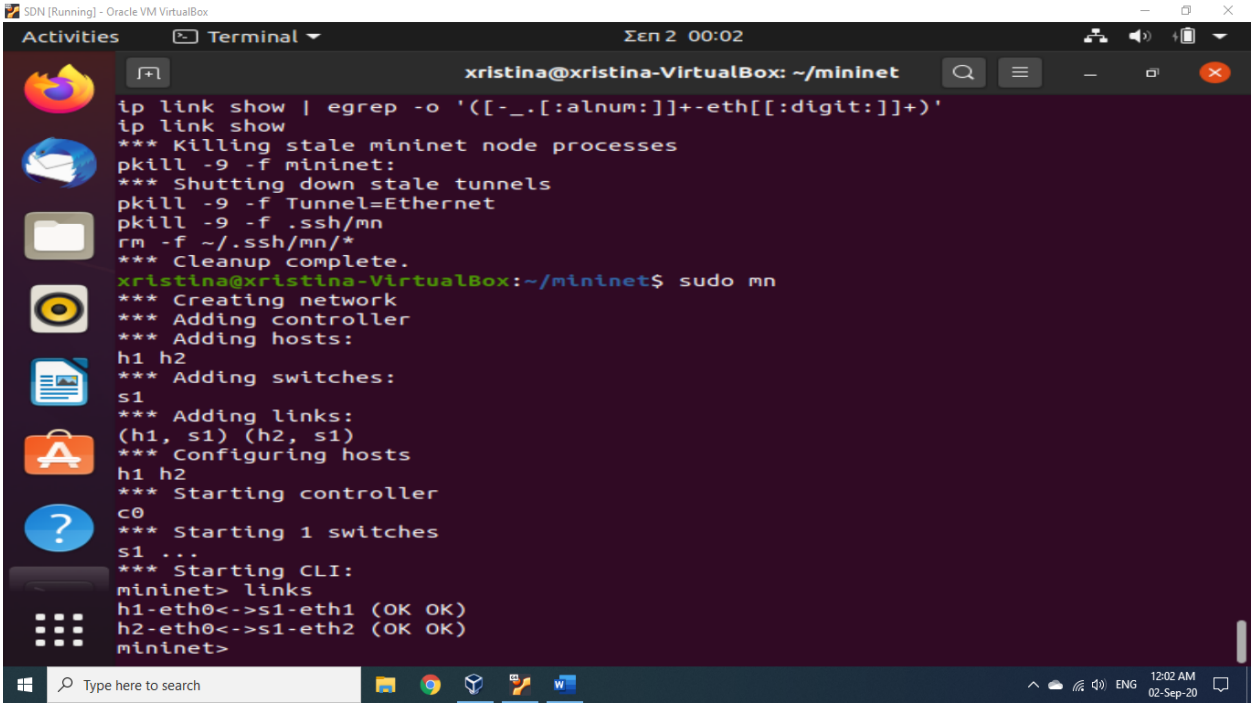
dpctl: Χρησιμοποιείται για την προβολή των ροών στον πίνακα του μεταγωγέα.



Το Mininet περιέχει πέντε προεπιλεγμένες τοπολογίες την: μικρή, απλή, αντεστραμμένη, γραμμική και δέντρο. Ο ελεγκτής του δικτύου δηλώνεται με c0, οι μεταγωγείς συμβολίζονται με s1 έως sn και οι υπολογιστές υποδηλώνονται με h1 έως hn στα σχήματα που χρησιμοποιούνται για την αναπαράσταση αυτών των τοπολογιών. Αυτές οι τοπολογίες εξηγούνται εν συντομία παρακάτω [31].

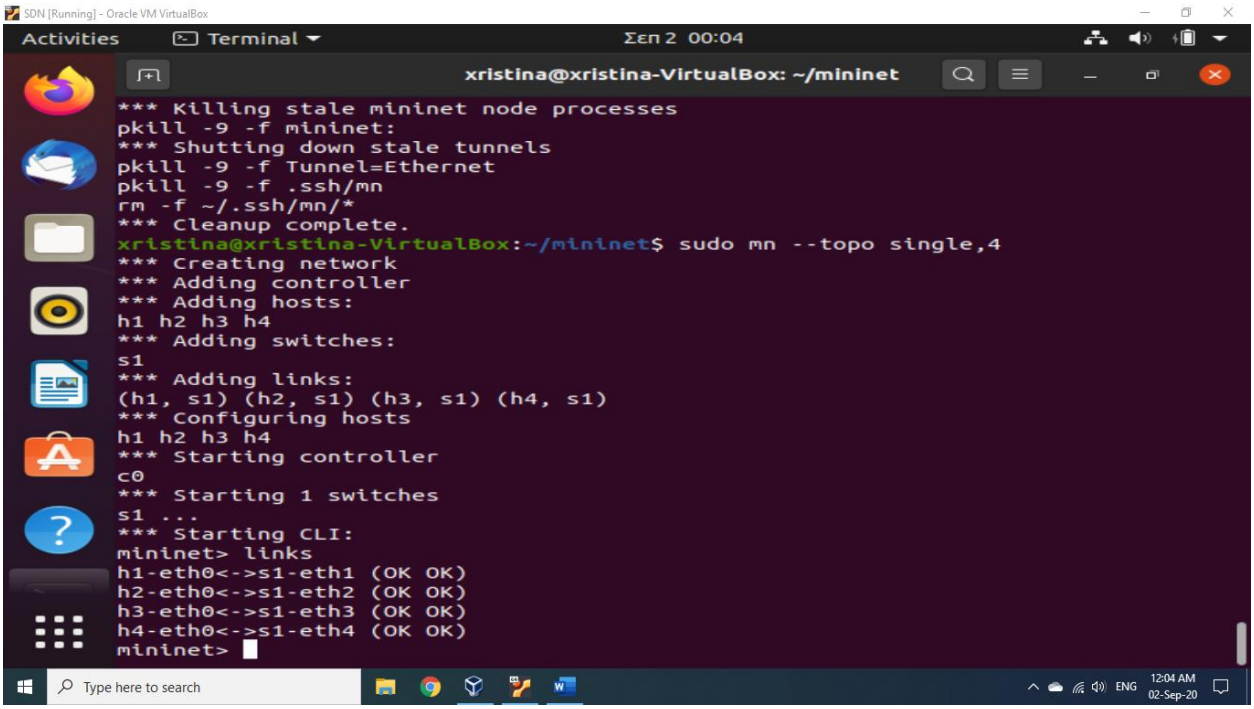
Για να τρέξουμε τις εντολές αρχικά πρέπει να κατεβάσουμε το VirtualBox, και έπειτα για να εγκαταστήσουμε το mininet εισάγουμε στην γραμμή εντολών την εντολή: sudo apt-get install mininet [32]. Καθώς και τις υπόλοιπες εντολές. Αφού ολοκληρωθεί η εγκατάσταση, ελέγχουμε την λειτουργικότητα του mininet με την εντολή: sudo mn - -test pingall [33].

Δημιουργία μικρής τοπολογίας: **sudo mn - -topo minimal ή sudo mn**



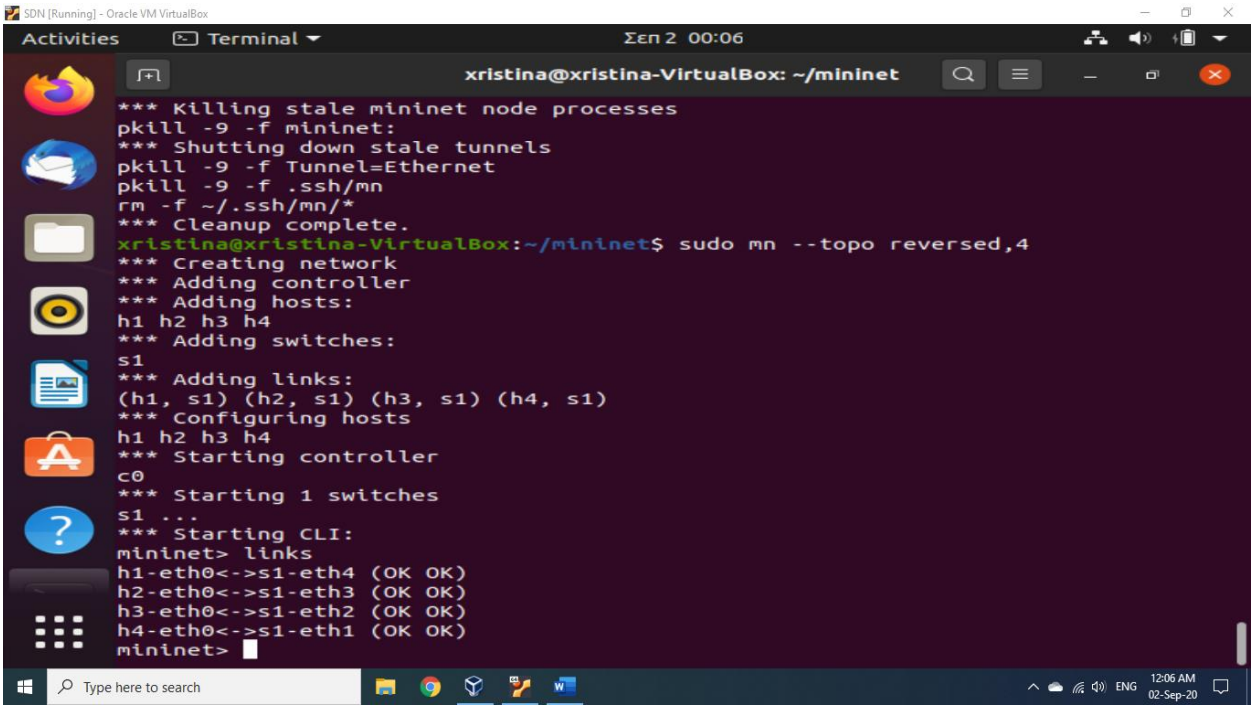
Εικόνα 12: Δημιουργία μικρής τοπολογίας

Δημιουργία απλής τοπολογίας: **sudo mn - - topo single, 4**



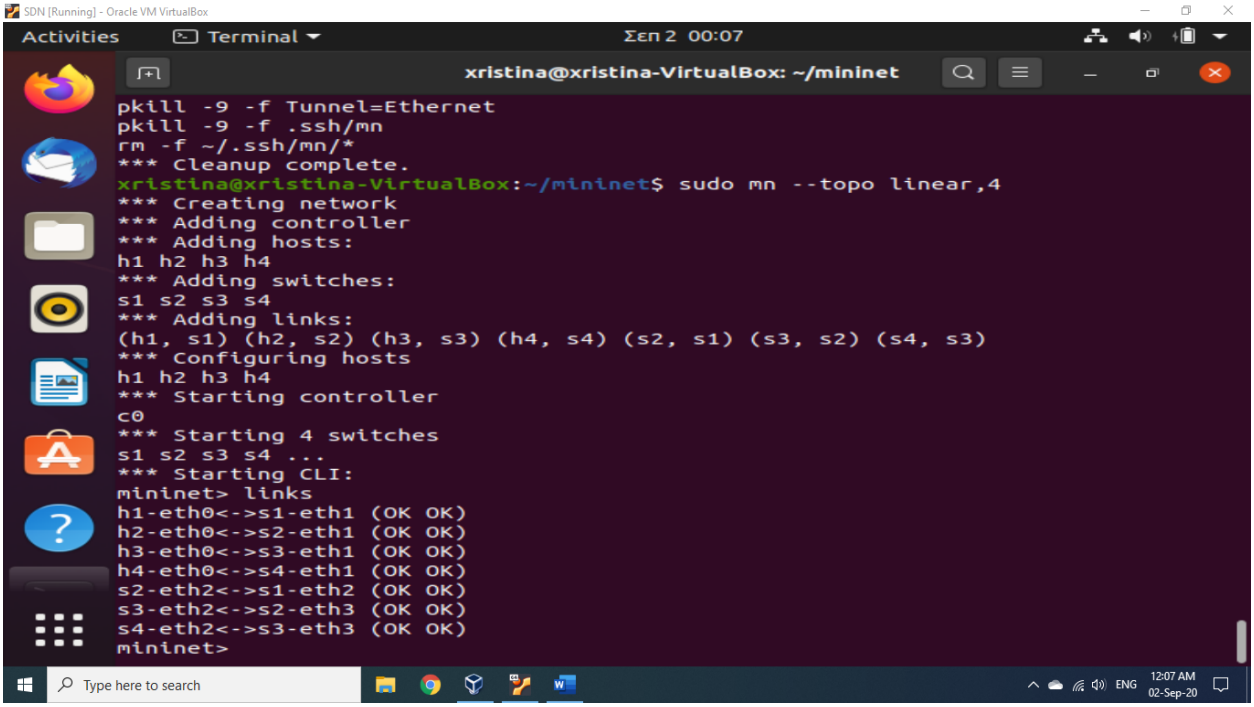
Εικόνα 13: Δημιουργία απλής τοπολογίας

Δημιουργία αντεστραμμένης τοπολογίας: **sudo mn - - topo reversed, 4**



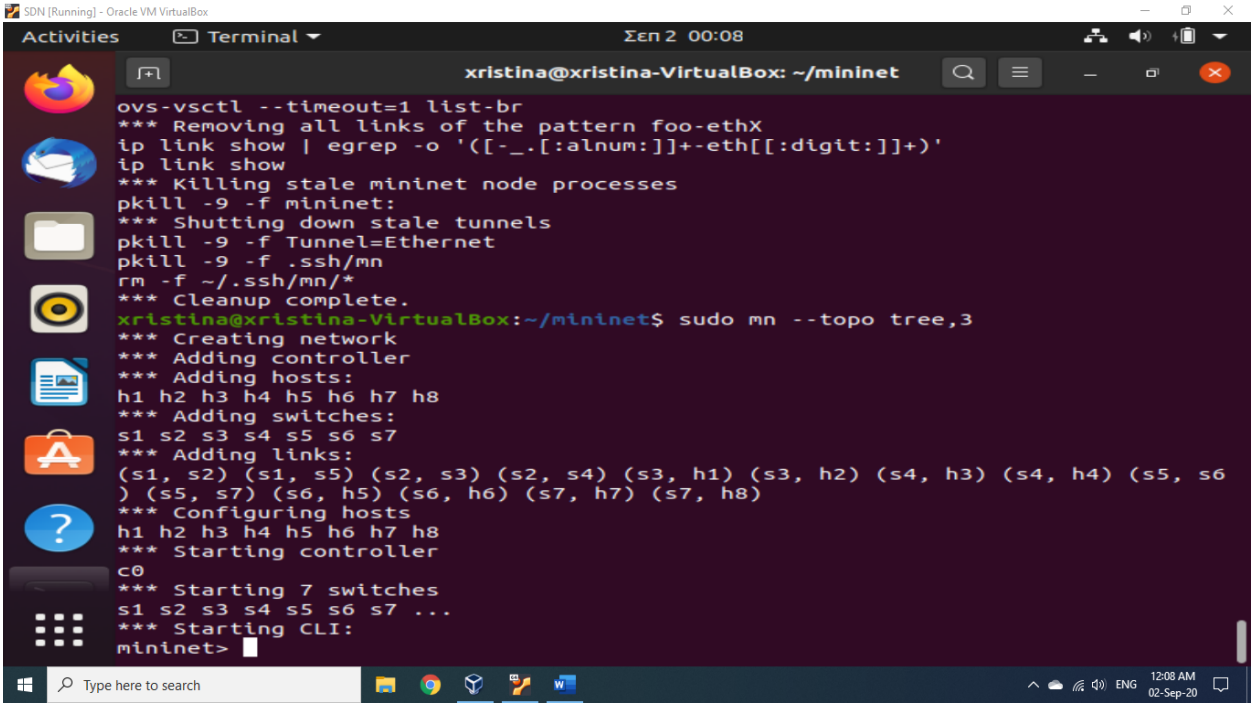
Εικόνα 14: Δημιουργία αντεστραμμένης τοπολογίας

Δημιουργία γραμμικής τοπολογίας: **sudo mn - - topo linear, 4**



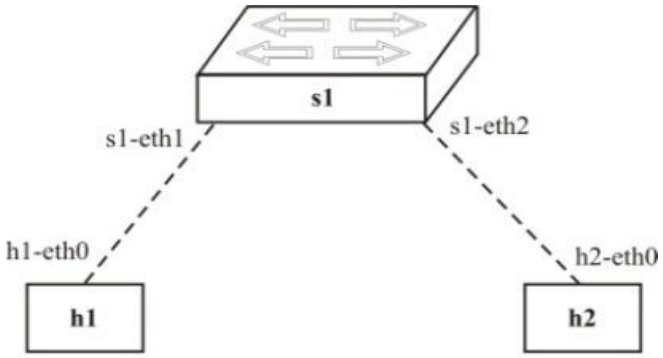
Εικόνα 15: Δημιουργία γραμμικής τοπολογίας

Δημιουργία τοπολογίας δέντρο: **sudo mn - - topo tree, 3**



Εικόνα 16: Δημιουργία τοπολογίας δέντρο

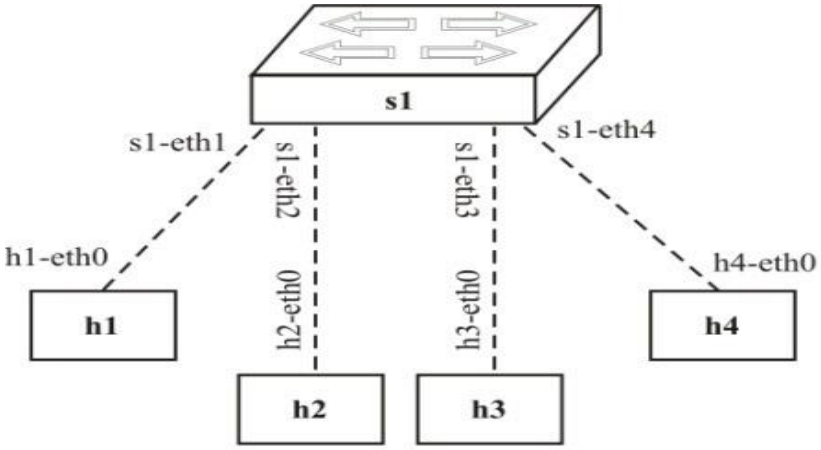
5.3.1 Μικρή τοπολογία - Minimal



Εικόνα 17: Μικρή τοπολογία [34]

Η μικρή τοπολογία περιέχει έναν ελεγκτή, έναν μεταγωγέα OpenFlow (s1) και δυο κεντρικούς υπολογιστές (h1, h2) επιπλέον δημιουργεί συνδέσμους μεταξύ του μεταγωγέα και των δύο κεντρικών υπολογιστών. Με την εντολή sudo mn --topo minimal ή sudo mn δημιουργούμε την παραπάνω τοπολογία.

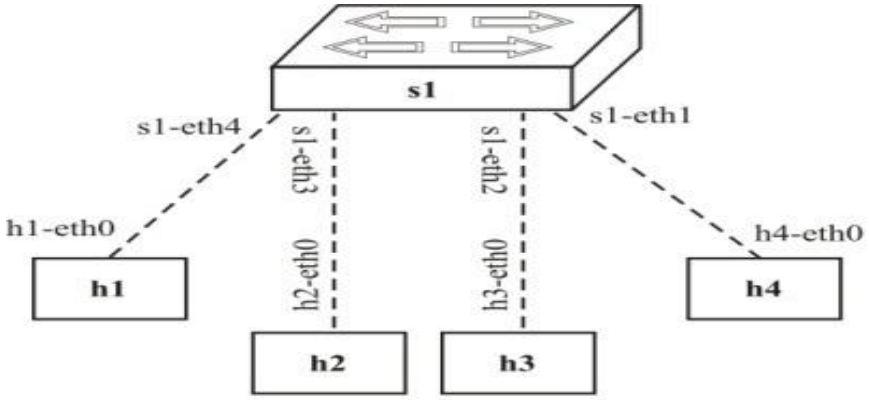
5.3.2 Απλή τοπολογία – Single



Εικόνα 18: Απλή τοπολογία [34]

Είναι μια απλή τοπολογία με έναν ελεγκτή, έναν μεταγωγέα OpenFlow και k κεντρικούς υπολογιστές, δημιουργεί επίσης, μια σύνδεση μεταξύ του μεταγωγέα και των k κεντρικών υπολογιστών. Με την εντολή sudo mn - - topo single,4 δημιουργούμε την παραπάνω τοπολογία.

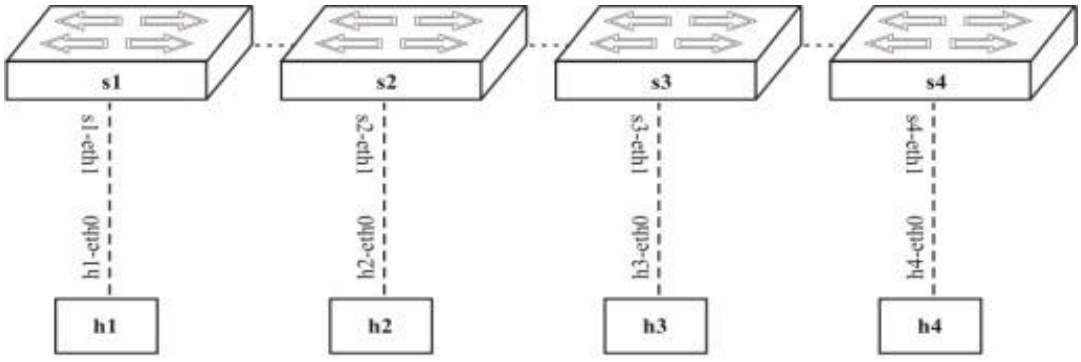
5.3.3 Αντεστραμμένη τοπολογία - Reversed



Εικόνα 19: Αντεστραμμένη τοπολογία [34]

Η αντεστραμμένη τοπολογία είναι παρόμοια με την απλή τοπολογία αλλά η σειρά σύνδεσης είναι αντεστραμμένη. Με την εντολή sudo mn --topo reversed,4 δημιουργούμε την παραπάνω τοπολογία.

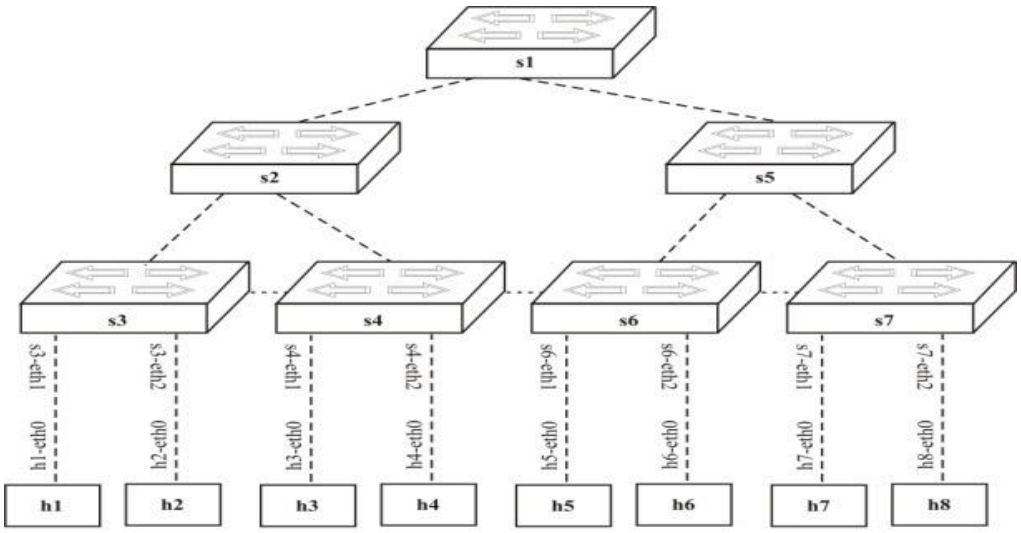
5.3.4 Γραμμική τοπολογία – Linear



Εικόνα 20: Γραμμική τοπολογία [34]

Η γραμμική τοπολογία περιέχει έναν ελεγκτή, k μεταγωγείς και k κεντρικούς υπολογιστές, επιπροσθέτως δημιουργεί μια σύνδεση μεταξύ κάθε μεταγωγέα και κάθε κεντρικού υπολογιστή και μεταξύ των μεταγωγέων. Με την εντολή sudo mn --topo linear,4 δημιουργούμε την παραπάνω τοπολογία.

5.3.5 Τοπολογία δέντρο - Tree



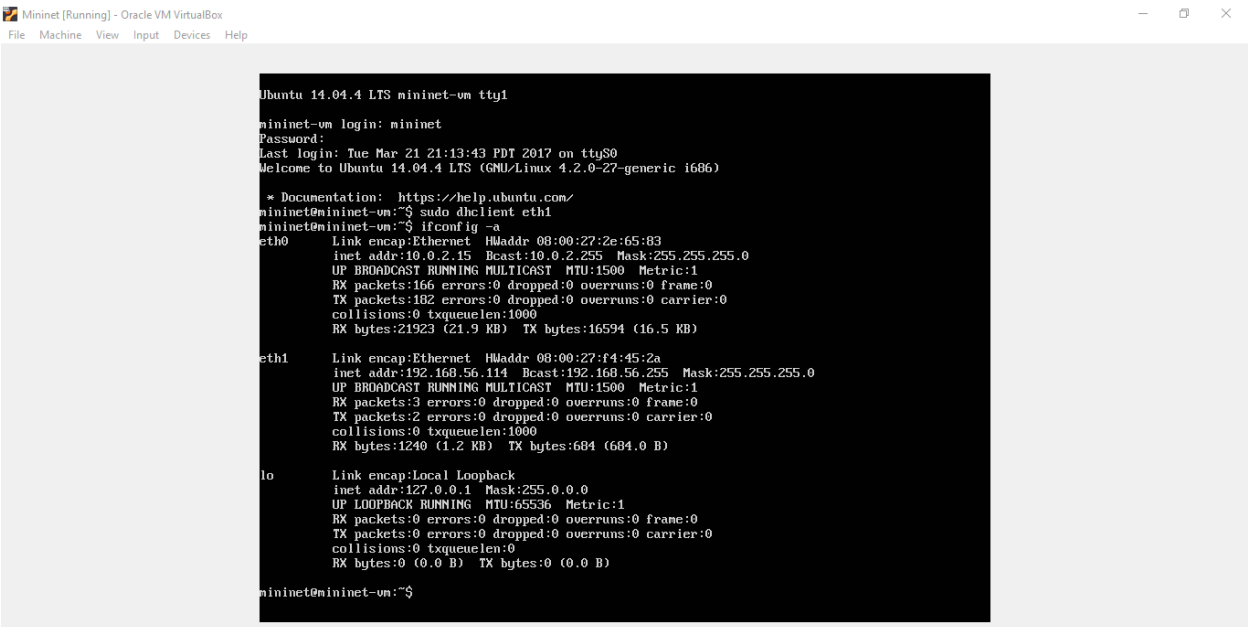
Εικόνα 21: Τοπολογία δέντρο [34]

Η τοπολογία δέντρο περιέχει έναν ελεγκτή, k επίπεδα και δυο κεντρικούς υπολογιστές που είναι συνδεδεμένοι ανά μεταγωγέα. Με την εντολή sudo mn - - topo tree,3 δημιουργούμε την παραπάνω τοπολογία.

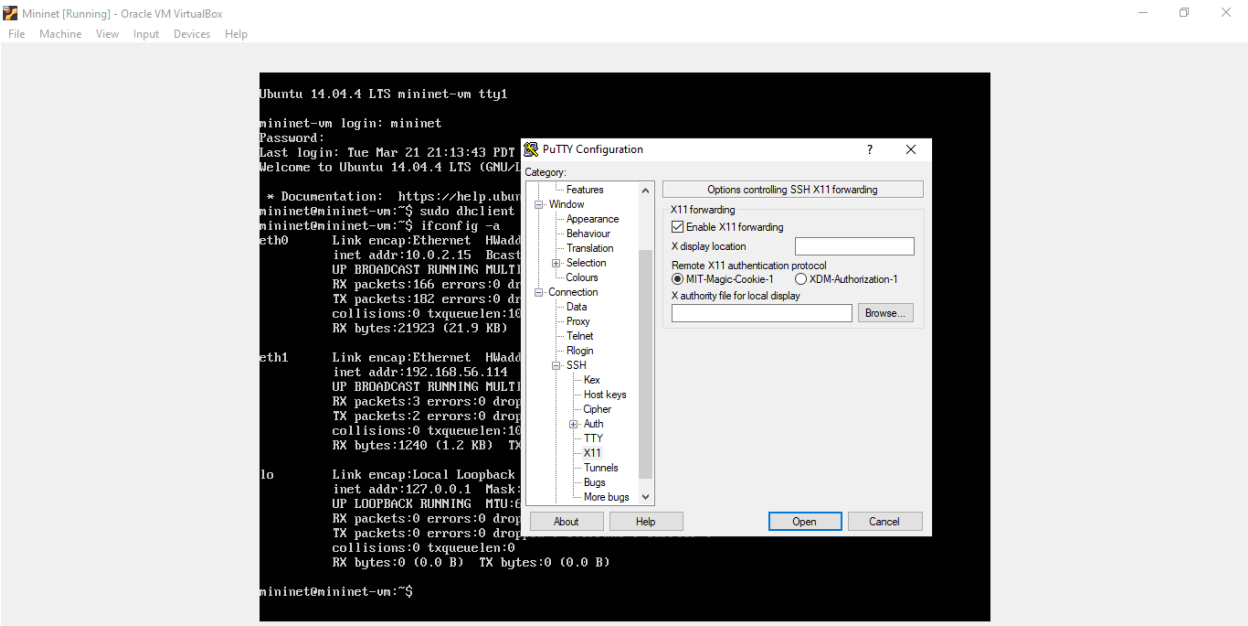
5.3.6 Προσαρμοσμένη τοπολογία – Custom

Μπορούμε επίσης να δημιουργήσουμε την δικιά μας τοπολογία και να την προσαρμόσουμε όπως εμείς θέλουμε, για παράδειγμα αν θέλουμε να φτιάξουμε μια προσαρμοσμένη τοπολογία με 2 μεταγωγείς και 5 κεντρικούς υπολογιστές χρειάζεται απλώς να γράψουμε μερικές γραμμές κώδικα σε γλώσσα Python [34].

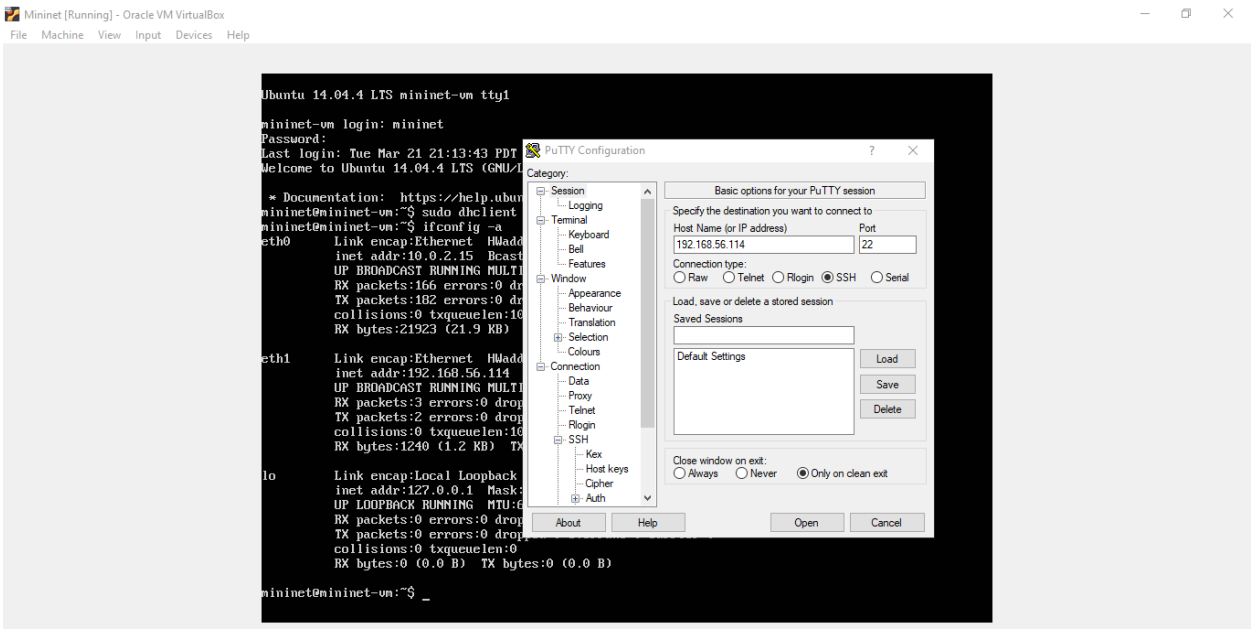
5.4 Ανίχνευση τοπολογίας



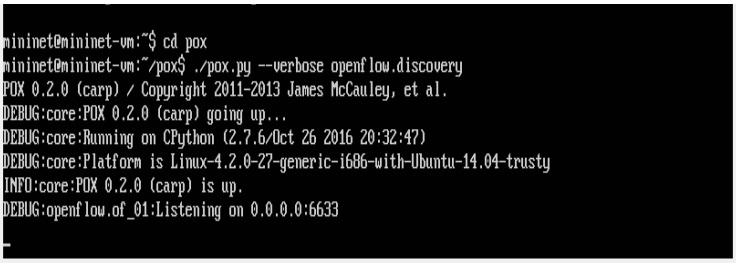
Αφού μπούμε στο mininet πληκτρολογούμε στο login: mininet και password: mininet, τρέχοντας παράλληλα και το Xming, έπειτα για να δούμε την εικονική κάρτα δικτύου του οικοδεσπότη εισάγουμε την εντολή ifconfig -a, παρατηρούμε ότι η διεύθυνση μας είναι η 192.168.56.114.



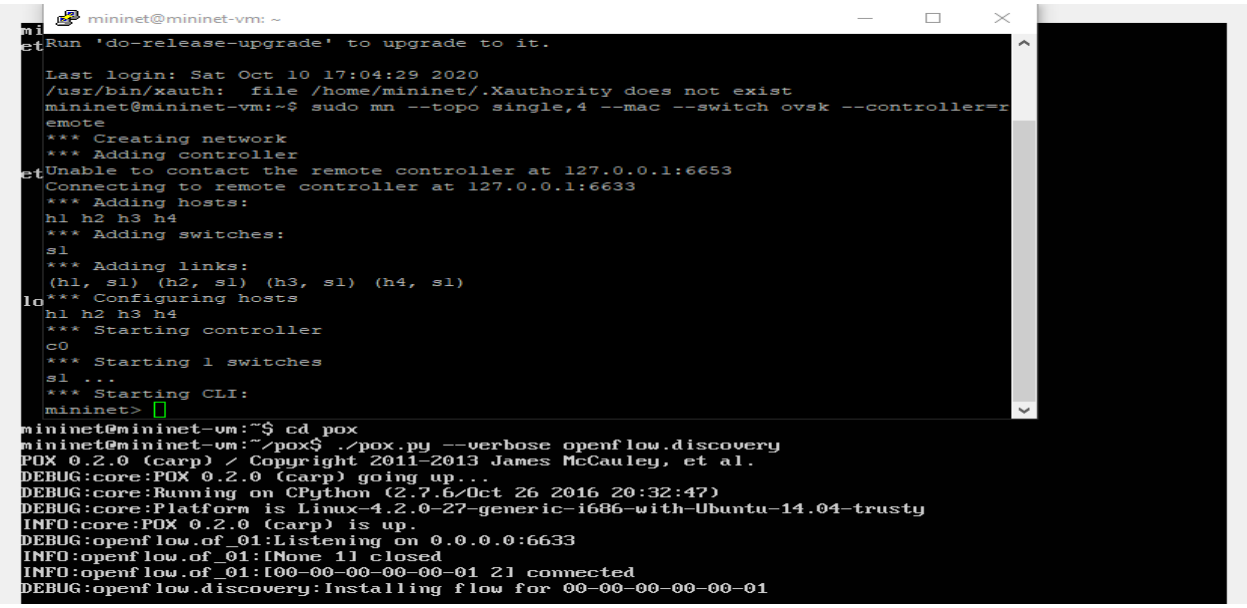
Ανοίγουμε το putty για να συνδεθούμε, στο SSH - X11 επιλέγουμε Enable X11 forwarding.



Στο Session πληκτρολογούμε στο Host Name (or IP address) την IP μας 192.168.56.144 και πατάμε το Open θα μας ανοίξει ένα νέο παράθυρο putty, θα εισάγουμε και στα δυο το κωδικό mininet.



Στο παράθυρο του mininet θα πληκτρολογήσουμε την εντολή cd pox για να μπούμε στον φάκελο του pox, στην συνέχεια θα εισάγουμε την εντολή: ./pox.py - -verbose openflow.discovery, βλέπουμε πως μας εμφανίζει τον pox ελεγκτή μας οπού ακούει στην προεπιλεγμένη θύρα 6633.

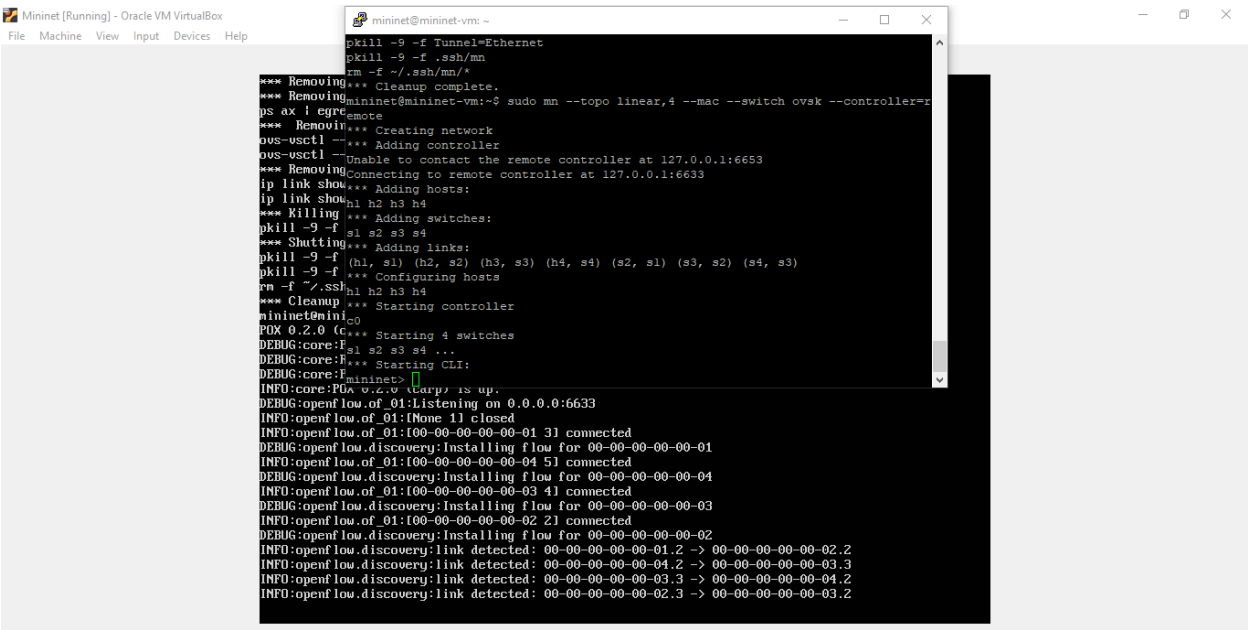


Κατόπιν, πάμε στο putty και γράφουμε την εντολή sudo mn --topo single,4 --mac --switch ovsk --controller=remote, αυτή η εντολή λέει στο Mininet να ξεκινήσει μια απλή τοπολογία δικτύου με:

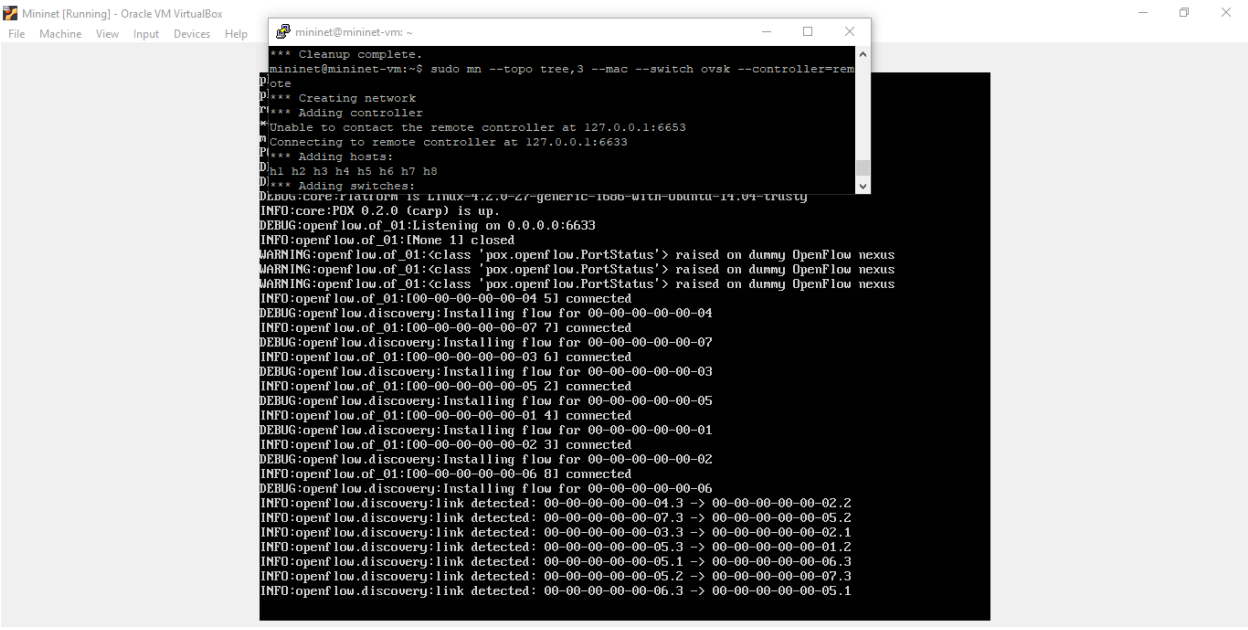
- 4 εικονικούς υπολογιστές, ο καθένας με ξεχωριστή IP διεύθυνση.
- Έναν ενιαίο μεταγωγέα λογισμικού OpenFlow (OpenvSwitch[2]) με 4 θύρες, που τρέχει στον πυρήνα του Mininet.
- Συνδέεται κάθε εικονικός κεντρικός υπολογιστής στο μεταγωγέα με ένα εικονικό καλώδιο ethernet.
- Ορίζει τη διεύθυνση MAC κάθε κεντρικού υπολογιστή ίση με την διεύθυνση IP του.
- Διαμορφώνεται ο μεταγωγέας OpenFlow για απομακρυσμένη σύνδεση με τον ελεγκτή (που από προεπιλογή εκτελείται στο localhost) [35].

Στο παράθυρο που έχουμε τρέξει το openflow.discovery βλέπουμε ότι έχουν προστεθεί δυο καινούργιες γραμμές, η πρώτη μας λέει ότι έχει συνδεθεί και η δεύτερη γραμμή ότι το πρωτόκολλο openflow.discovery εγκατέστησε μια ροή στην MAC address 00-00-00-00-00-01.

[2] Το OpenvSwitch, μερικές φορές συντομευμένο ως OVS, είναι μια υλοποίηση ανοιχτού κώδικα ενός διανεμημένου εικονικού μεταγωγέα πολλαπλών επιπέδων.

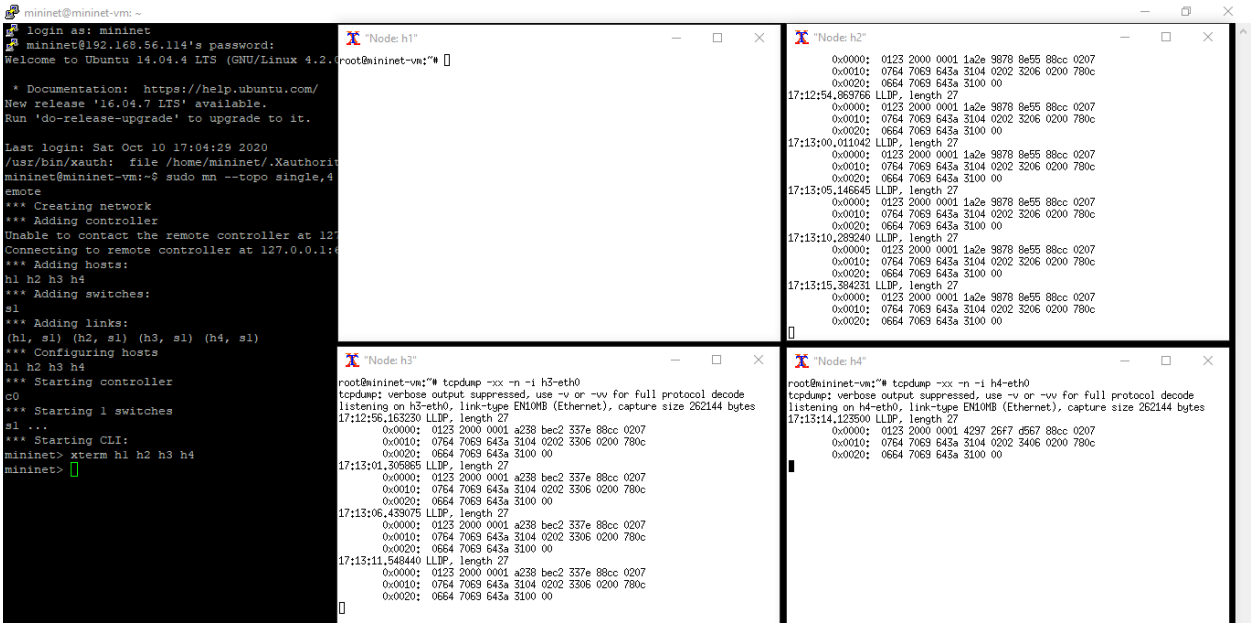


Δοκιμάζουμε και την γραμμική τοπολογία sudo mn --topo linear,4 --mac --switch ovsk --controller=remote, βλέπουμε ότι όλοι οι μεταγωγείς έχουν συνδεθεί και έχει εγκαταστήσει μια ροή MAC address για το καθένα, επιπλέον έχει εντοπίσει αυτήν την τοπολογία και μας λέει ότι το 00-00-00-00-00-01.2 συνδέεται με το 00-00-00-00-00-02.2 και αντίστοιχα και για τα υπόλοιπά.



Επίσης, δοκιμάζουμε και την τοπολογία δέντρο sudo mn --topo tree,3 --mac --switch ovsk --controller=remote, βλέπουμε ότι όλοι οι μεταγωγείς έχουν συνδεθεί και έχει εγκαταστήσει μια ροή

MAC address για το καθένα, επιπλέον έχει εντοπίσει αυτήν την τοπολογία και μας λέει ότι το 00-00-00-00-00-04.3 συνδέεται με το 00-00-00-00-00-02.2 και αντίστοιχα και για τα υπόλοιπά.



Υστέρα, για να επαληθεύσουμε ότι οι κεντρικοί υπολογιστές στην τοπολογία του δικτύου μας μπορούν να επικοινωνήσουν μεταξύ τους (χρησιμοποιώντας την εντολή ping) και ότι η εφαρμογή hub στον ελεγκτή λειτουργεί σωστά, ελέγχοντας ότι όλα τα host μπορούν να δουν την ίδια κίνηση. Για να γίνει αυτό, θα εκτελέσουμε συνεδρίες xterm για κάθε κεντρικό υπολογιστή και θα παρακολουθήσουμε την κίνηση που μπορεί να δει ο καθένας [35]. Εισάγουμε την εντολή xterm h1 h2 h3 h4 οπού θα μας ανοίξει 4 νέα παράθυρα των υπολογιστών. Στο κάθε ένα από αυτά εκτός από το h1 γράφουμε την παρακάτω εντολή: tcpdump -xx -n -i h2-eth0 όπου h2 είναι ο πρώτος υπολογιστής και κάνουμε το ίδιο και για τους υπολοίπους (h2,h3) το αποτέλεσμα αυτών των ενεργειών είναι το παραπάνω που βλέπουμε στην εικόνα.

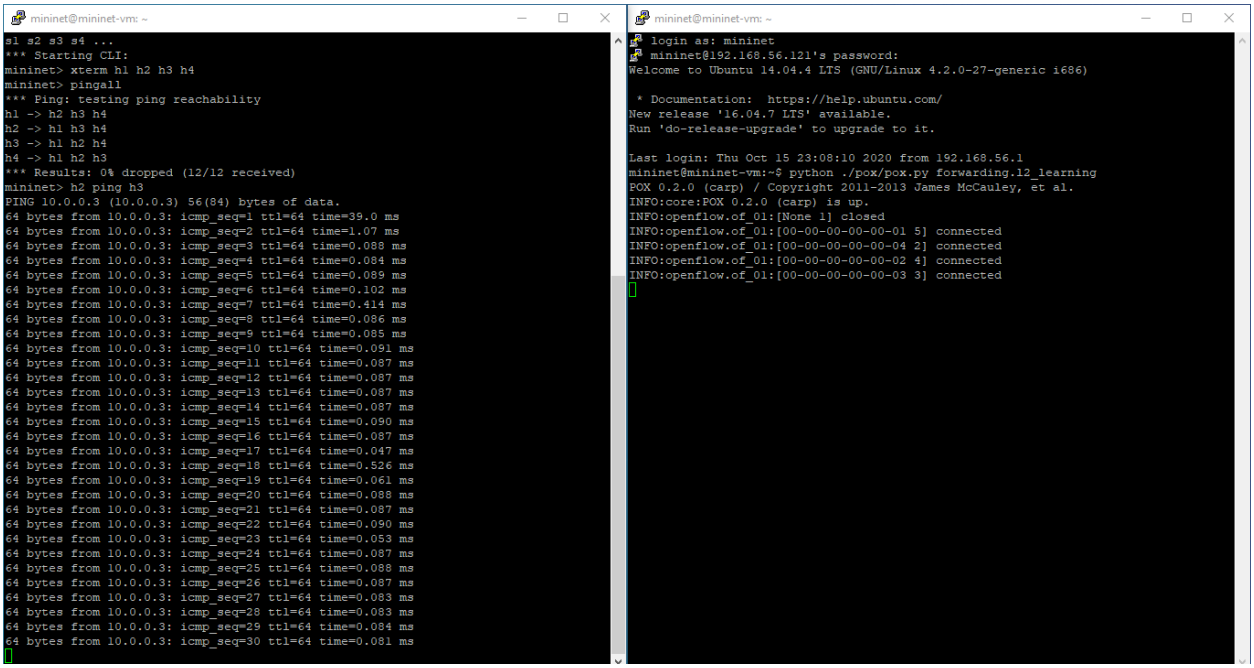
Η εντολή **tcpdump** είναι ένα ισχυρό και ευρέως χρησιμοποιούμενο εργαλείο εντοπισμού πακέτων γραμμής εντολών ή εργαλείων ανάλυσης πακέτων που χρησιμοποιείται για τη λήψη ή φιλτράρισμα πακέτων TCP/IP που έλαβαν ή μεταφέρθηκαν μέσω του δικτύου σε μια συγκεκριμένη διεπαφή [36].

-xx: Καταγράφει τα δεδομένα κάθε πακέτου, συμπεριλαμβανομένης της κεφαλίδας του επιπέδου σύνδεσης σε μορφή δεκαεξαδικού (HEX) και ASCII.

-n: Καταγραφή πακέτων διεύθυνσης IP.

-i: Καταγραφή πακέτων από συγκεκριμένη διεπαφή.

h2-eth0: Επιλογή του host h2 με την διεπαφή eth0, το ίδιο και για τα h3,h4.



Τέλος με την εντολή ping ελέγχουμε ότι οι υπολογιστές στην τοπολογία του δικτύου μας μπορούν να επικοινωνήσουν μεταξύ τους.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Ξ. Δημητρόπουλος, «opencourses.uoc.,» 3 24 2015. [Ηλεκτρονικό]. Available: https://opencourses.uoc.gr/courses/mod/page/view.php?id=6881. [Πρόσβαση 8 2020].

[2] IBM Services, "IBM," 9 8 2019. [Online]. Available: https://www.ibm.com/services/network/sdn-versus-traditional-networking. [Accessed 2020].

[3] A. K. Singh, «researchgate,» 3 2018. [Ηλεκτρονικό]. Available: https://www.researchgate.net/figure/Software-defined-networking-SDN-vs-traditional-network_tbl1_323974224. [Πρόσβαση 8 2020].

[4] Juniper Networks, "Juniper Networks," [Online]. Available: https://www.juniper.net/us/en/solutions/sdn/what-is-sdn/.

[5] T. D. Nadeau and K. Gray, SDN: Software Defined Networks, First Edition ed., O'Reilly Media, Inc., 2013, p. 9.

[6] «wikipedia,» 11 6 2020. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Software-defined_networking. [Πρόσβαση 2020].

[7] M. N. O. Sadiku, Emerging Internet-Based, CRC Press, 2019, p. 98.

[8] L. Ochoa-Aday, C. Cervello-Pastor και . A. Fernandez-Fern ´ andez, «Current Trends of Topology Discovery in OpenFlow-based Software Defined Networks,» 2015.

[9] B. Curtis, "YourTechDiet," [Online]. Available: https://www.yourtechdiet.com/blogs/software-defined-networking-sdn/.

[10] sdxcentral, "sdxcentral," 26 8 2016. [Online]. Available: https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/. [Accessed 2020].

[11] adool999, «NETWORK SIMULATION WITH MININET,» 9 7 2015. [Ηλεκτρονικό]. Available: https://projectmincpm.wordpress.com/2015/07/09/openflow-and-software-defined-networking/#more-49. [Πρόσβαση 2020].

[12] Technology Blog, «Pros And Cons Of Software-Defined Networking (SDN),» 21 1 2018. [Ηλεκτρονικό]. Available: http://comsoft-bh.com/2018/01/21/pros-and-cons-of-software-defined-networking-sdn/. [Πρόσβαση 15 8 2020].

[13] T. keary, «comparitech,» 13 7 2020. [Ηλεκτρονικό]. Available: https://www.comparitech.com/net-admin/software-defined-networking/. [Πρόσβαση 15 8 2020].

[14] A. Abdelaziz, T. F. Ang, M. Sookhak, S. Khan, A. Vasilakos, C. S. Liew και A. Akhunzada, «Survey on Network Virtualization Using OpenFlow: Taxonomy, Opportunities,and Open Issues,» 2016.

[15] P. Göransson and C. Black, Software Defined Networks A Comprehensive Approach, Waltham: Elsevier Inc, 2014, pp. 82-83.

[16] Open Networking Foundation, "opennetworking.org," 6 9 2012. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf.

[17] Open Networking Foundation, «opennetworking.org,» 14 10 2013. [Ηλεκτρονικό]. Available: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.4.0.pdf. [Πρόσβαση 2020].

[18] Y.-J. Chen , L.-C. Wang, . F.-Y. Lin και B.-S. Paul Lin, «Deterministic Quality of Service Guarantee for Dynamic Service Chaining in Software Defined Networking,» IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, 2017.

[19] S. Shuruthi, S. Kannan και .. Sanjayprasad, «Implementation of Failure Link Recovery through Spanning Tree Protocol in SDN,» National engineering college,, kovilpatti, 2012.

[20] SDxCentral Staff, «SDxCentral,» 23 6 2016. [Ηλεκτρονικό]. Available: https://www.sdxcentral.com/networking/sdn/definitions/openflow-controller/. [Πρόσβαση 2020].

[21] A. Zhu., «Network Switches,» 27 7 2018. [Ηλεκτρονικό]. Available: http://www.cables-solutions.com/whats-openflow-switch-how-it-works.html. [Πρόσβαση 2020].

[22] S. Khan, A. Gani, A. W. A. Wahab, M. Guizani και M. K. Khan, «Topology Discovery in Software Defined Networks:Threats, Taxonomy, and State-of-the-art,» IEEE, 2016.

[23] L. Ochoa-Aday, C. Cervelló-Pastor and A. Fernández-Fernández, "core.ac.uk," [Online]. Available: https://core.ac.uk/download/pdf/81578389.pdf. [Accessed 2020].

[24] F. Pakzad, «Efficient Topology Discovery in Software Defined Networks,» Australia, 2014.

[25] D. Hasan και M. Othman, «Efficient Topology Discovery in Software Defined Networks:,» *ScienceDirect,* pp. 2-9, 13-14 10 2017.

[26] J. GASCON και juangascon, «mininet,» [Ηλεκτρονικό]. Available: https://atomstore.pluginthefuture.eu/?order_by=last_release&ordering=desc. [Πρόσβαση 2020].

[27] R. R. Brooks και I. Ozcelik, Distributed Denial of Service Attacks Real-world Detection and Mitigation, India: CRC Press, 2020, p. 123.

[28] N. Saritakumar, V. Adarsh , E. Srinivasan , S. Albert και R. Subha , «Performance Evaluation of Pox Controller for Software Defined Networks,» IJITEE, 2019.

[29] M. Team, «mininet.org,» 2018. [Ηλεκτρονικό]. Available: http://mininet.org/walkthrough/. [Πρόσβαση 8 2020].

[30] M. Rouse, «TechTarget,» 8 2005. [Ηλεκτρονικό]. Available: https://searchsecurity.techtarget.com/definition/sudo-superuser-do. [Πρόσβαση 10 2020].

[31] K. K. Sharma και M. Sood, «Mininet as a Container Based Emulator for Software Defined Networks,» India, 2014.

[32] M. Team, «mininet.org,» 2018 . [Ηλεκτρονικό]. Available: http://mininet.org/download/. [Πρόσβαση 8 2020].

[33] A. Bhadange, 1 12 2017. [Ηλεκτρονικό]. Available: https://www.youtube.com/watch?v=ZUzHKDIUFh4. [Πρόσβαση 8 2020].

[34] K. Kaur, J. Singh και N. S. Ghumman, «Mininet as Software Defined Networking Testing Platform,» Shaheed Bhagat Singh State Technical Campus, India, 2014.

[35] P. Bernhard , P. Georgopoulos, B. Ager, M. Happe, K. Humme και V. Kotronis, «Advanced Topics in Communication Networks,» Zurich, 2014.

[36] N. Shrestha, «Tecmint,» 3 7 2020. [Ηλεκτρονικό]. Available: https://www.tecmint.com/12-tcpdump-commands-a-network-sniffer-tool/. [Πρόσβαση 10 2020].

[37] T. Keary, «comparitech,» 4 6 2020. [Ηλεκτρονικό]. Available: https://www.comparitech.com/net-admin/network-topologies-advantages-disadvantages/. [Πρόσβαση 8 2020].

[38] TEK-TOOLS, 13 4 2020. [Ηλεκτρονικό]. Available: https://www.tek-tools.com/network/best-network-topology-software. [Πρόσβαση 8 2020].

[39] saurabhsharma, «geeksforgeeks,» 14 5 2020. [Ηλεκτρονικό]. Available: https://www.geeksforgeeks.org/open-shortest-path-first-ospf-protocol-states/. [Πρόσβαση 9 2020].