



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ**

**ΤΜΗΜΑ: ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΥΛΟΠΟΙΗΣΗ ΔΙΚΤΥΩΝ ΟΡΙΖΟΜΕΝΩΝ ΑΠΟ ΛΟΓΙΣΜΙΚΟ – SDN**

**ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ: ΝΙΚΟΛΑΟΣ ΟΙΚΟΝΟΜΟΥ**

**ΕΠΙΒΛΕΠΟΥΣΑ: ΣΠΥΡΙΔΟΥΛΑ ΜΑΡΓΑΡΙΤΗ**

**ΑΡΤΑ 2021**



**UNIVERSITY OF IOANNINA**

**DEPARTMENT: INFORMATICS AND TELECOMMUNICATIONS**

**THESIS**

**IMPLEMENTATION OF SOFTWARE DEFINED NETWORKS – SDN**

**STUDENT: NIKOLAOS OIKONOMOU**

**SUPERVISOR: SPIRIDOULA MARGARITI**

**ARTA 2021**

### Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιακή εργασία είναι εξολοκλήρου 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.

**ΟΙΚΟΝΟΜΟΥ ΝΙΚΟΛΑΟΣ**

## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει ως θέμα την ανάλυση των δικτύων οριζόμενων από λογισμικό (SDN). Αρχικά παρουσιάζεται η δικτύωση SDN ως προς τη δομή και τη λειτουργία του. Στη συνέχεια θα χρησιμοποιηθεί ο εξομοιωτής Mininet συνοδευόμενος και από άλλα λογισμικά προκειμένου να υλοποιηθούν οι πιο ευρέως διαδεδομένες τοπολογίες SDN. Από τις εξομοιώσεις αυτές θα συλλεχθούν στοιχεία που αποσκοπούν στην αξιολόγηση της αποδοτικότητας αυτών των δικτύων καθώς και στη διερεύνηση των απαιτήσεων τους σχετικά το πόσους πόρους καταναλώνονται από το ένα σύστημα για την εκτέλεσή τους. Τέλος θα εξαχθούν συμπεράσματα για το πόσο συμφέρουν τέτοιου είδους εξομοιώσεις και για το αν συμφέρει γενικότερα η εξέλιξη δικτύων οριζόμενων από λογισμικό (SDN).

## **ABSTRACT**

The present dissertation is on the analysis of software defined networks (SDN). In the first phase, the SDN will be analyzed in terms of its structure and operation. Then the Mininet emulator will be used accompanied by other software to implement the more widespread SDN topologies. These simulations will collect data that will test the efficiency of their execution as well as what is needed to execute them and how many resources are consumed by a system. Finally, conclusions will be drawn as to how beneficial such simulations are and whether the development of software-defined networks (SDNs) is generally beneficial.

## Περιεχόμενα

Περιεχόμενα.....	6
1.Εισαγωγή.....	9
1.1 Αντικείμενο και στόχος της πτυχιακής εργασίας.....	9
1.2 Δομή πτυχιακής εργασίας.....	10
2. SDN .....	11
2.1 Ορισμός .....	11
2.2 Εξέλιξη SDN .....	12
2.3 Αρχιτεκτονική SDN .....	12
2.3.1 Επίπεδο εφαρμογών .....	13
2.3.2 Επίπεδο ελέγχου .....	13
2.3.3 Επίπεδο προώθησης δεδομένων.....	14
2.4 Χαρακτηριστικά SDN .....	14
2.5 Εφαρμογές SDN.....	15
2.6 Πλεονεκτήματα χρήσης SDN.....	16
2.7 Μειονεκτήματα χρήσης SDN.....	16
3. Η λειτουργία SDN .....	18
3.1 Βασικά λειτουργικά στοιχεία .....	18
3.2 Ελεγκτές SDN .....	18
3.2.1 Είδη ελεγκτών.....	18
3.2.2 Ο ελεγκτής POX .....	22
3.3 Πρωτόκολλα επικοινωνίας.....	23
3.3.1 Το πρωτόκολλο Openflow.....	23
3.4 Τοπολογίες SDN.....	29
3.4.1 Είδη τοπολογιών SDN.....	29
3.5 Παράδειγμα εφαρμογής SDN .....	35
4. Περιβάλλον εξομίωσης SDN .....	41

4.1 Mininet OS.....	41
4.1.1 Στόχος Mininet OS.....	42
4.1.2 Όρια λειτουργίας Mininet OS.....	43
4.2 VirtualBox.....	43
4.3 X-Ming.....	43
4.4 WinSCP.....	44
4.5 PuTTY.....	44
4.6 GERPHI.....	45
4.7 Τεχνικά χαρακτηριστικά.....	46
4.8 Εγκατάσταση λογισμικού.....	46
4.8.1 Εγκατάσταση VirtualBox.....	46
4.8.1.1 Περιβάλλον εργασίας VirtualBox.....	48
4.8.1.2 CPU Virtualization.....	49
4.8.2 Εγκατάσταση Mininet.....	56
4.8.2.1 Περιβάλλον εργασίας Mininet OS.....	57
4.8.3 Εγκατάσταση X-Ming.....	61
4.8.3.1 Περιβάλλον εργασίας X-Ming.....	61
4.8.4 Εγκατάσταση WinSCP.....	61
4.8.4.1 Περιβάλλον εργασίας WinSCP.....	62
4.8.5 Εγκατάσταση PuTTY.....	63
4.8.5.1 Περιβάλλον εργασίας PuTTY.....	63
4.8.6 Εγκατάσταση Gerphi.....	63
4.8.6.1 Περιβάλλον εργασίας Gerphi.....	64
5. Υλοποίηση εξομίωσης SDN.....	66
5.1 Τοπολογίες εξομίωσης.....	66
5.2 Μετρικές επιδόσεις.....	67
5.3 Ρύθμιση Προγραμμάτων.....	68

5.3.1 Ρύθμιση VirtualBox .....	68
5.3.2 Ρύθμιση Mininet .....	73
5.3.3 Ρύθμιση X-Ming.....	81
5.3.4 Ρύθμιση WinSCP.....	81
5.3.5 Ρύθμιση PuTTY .....	84
5.3.6 Ρύθμιση Gephi.....	87
5.4 Εισαγωγή δεδομένων.....	94
5.4.1 Εξομοίωση τοπολογίας Tree .....	94
5.4.2 Εξομοίωση τοπολογίας Linear .....	99
5.4.3 Εξομοίωση τοπολογίας Erdos Renyi.....	100
5.4.4 Γραφική απεικόνιση τοπολογιών μέσω Gephi .....	100
5.4.4.1 Γραφική απεικόνιση Tree.....	101
5.4.4.2 Γραφική απεικόνιση Linear .....	102
5.4.4.3 Γραφική απεικόνιση Erdos Renyi .....	104
5.5 Συλλογή αποτελεσμάτων .....	105
5.6 Ανάλυση αποτελεσμάτων .....	108
5.6.1 CPU Usage .....	108
5.6.2 Memory Usage .....	111
5.6.3 Setup & Tear Time.....	113
5.6.4 Bandwidth .....	115
5.6.5 Latency .....	117
6.Συμπεράσματα.....	122
7.Βιβλιογραφικές αναφορές .....	123
7.1 Παράρτημα.....	126



## **1.Εισαγωγή**

Τα ταχύτατα αναπτυσσόμενα σύγχρονα δίκτυα παρουσιάζουν αυξανόμενες ανάγκες σε χωρητικότητα και ταχύτητα με την πάροδο των χρόνων διότι οι εταιρείες δικτυακών παροχών έχουν αυξημένες δυνατότητες που απαιτούν περισσότερους πόρους ενώ ταυτόχρονα έχουν αυξηθεί κατακόρυφα οι χρήστες και οι απαιτήσεις τους. Αυτό έχει ως αποτέλεσμα να φανούν οι αδυναμίες των παραδοσιακών δικτύων και σιγά σιγά να δώσουν τη θέση τους στις σύγχρονες τεχνολογίες δικτύων οριζόμενων από λογισμικό (SDN). Ένα παραδοσιακό δίκτυο απαιτείται από φυσικές υποδομές που δεν μπορούν να διαμορφωθούν εύκολα και χάνονται αναπόφευκτα από το προσκήνιο. Οι τεχνολογίες SDN σε αντίθεση μπορούν εικονικά μέσω λογισμικού να αντικαταστήσουν τις ξεπερασμένες πλέον φυσικές-υλικές υποδομές των παραδοσιακών δικτύων και να ανταποκριθούν στη σύγχρονη πραγματικότητα παρέχοντας αυξημένες δυνατότητες.

### **1.1 Αντικείμενο και στόχος της πτυχιακής εργασίας**

Στην παρούσα πτυχιακή εργασία θα παρουσιάσουμε την σύγχρονη και ταχύτατα αναπτυσσόμενη αρχιτεκτονική δικτύων SDN που ήρθε να αντικαταστήσει τα παραδοσιακά δίκτυα και θα μελετήσουμε τη δομή και τη λειτουργία τους εστιάζοντας στον τρόπο δόμησης, τις δυνατότητες, τις εφαρμογές τους και τις χαρακτηριστικές ιδιότητες. Προκειμένου να κατανοηθεί η λειτουργία τους και να εξεταστούν τα πλεονεκτήματα και τα μειονεκτήματά τους θα πραγματοποιηθεί εξομοίωση με τη βοήθεια του εργαλείου Mininet. Η προσομοίωση επιτρέπει την ανάπτυξη δικτύων SDN, την παρατήρηση της συμπεριφοράς τους και τη συλλογή αποτελεσμάτων που μας αποκαλύπτουν τον τρόπο λειτουργίας του συστήματος, την αποδοτικότητά και τις απαιτήσεις του. Τέλος θα εξαχθούν συμπεράσματα γύρω από τη λειτουργία των SDN καθώς και τον σκοπό και την αποτελεσματικότητα των εξομοιώσεων.

## **1.2 Δομή πτυχιακής εργασίας**

Το 1<sup>ο</sup> είναι το εισαγωγικό κεφάλαιο της πτυχιακής εργασίας. Στο 2<sup>ο</sup> κεφάλαιο γίνεται αναφορά στον ορισμό του SDN, στη δομή του, τα χαρακτηριστικά καθώς και το σκοπό της δημιουργίας και της χρήσης του. Στο 3<sup>ο</sup> κεφάλαιο αναλύεται η λειτουργία του SDN, οι ελεγκτές, τα πρωτόκολλα καθώς και οι τοπολογίες του. Στο 4<sup>ο</sup> κεφάλαιο αναλύονται το εργαλείο Mininet που χρησιμοποιήθηκε στην παρούσα πτυχιακή καθώς και τα υπόλοιπα λογισμικά που απαιτούνται για την επιτυχία της εξομοίωσης. Συμπεριλαμβάνονται και αναλυτικές οδηγίες εγκατάστασης καθώς και χρήσης τους. Στο 5<sup>ο</sup> κεφάλαιο περιλαμβάνονται οι τοπολογίες που θα εκτελεστούν στην εξομοίωση και οι πλήρως αναλυτικές οδηγίες ρύθμισης των λογισμικών και παρουσίαση της εξομοίωσης. Εξάγονται στατιστικά αποτελέσματα ως προς την επιβάρυνση του συστήματος από την υλοποίηση τέτοιων εξομοιώσεων και αναλύονται. Στο 6<sup>ο</sup> κεφάλαιο παρουσιάζονται τα συμπεράσματα που προκύπτουν από την παρούσα πτυχιακή εργασία.

## 2. SDN

### 2.1 Ορισμός

Ο όρος Software Defined Networking (SDN) μεταφράζεται ως δίκτυα οριζόμενα από λογισμικό. Το SDN είναι μια εκκολαπτόμενη αρχιτεκτονική δικτύων η οποία διαχωρίζει το επίπεδο ελέγχου (Control Plane) ενός δικτύου από το επίπεδο προώθησης δεδομένων (Forwarding/Data Plane) και παρέχει τη δυνατότητα στο επίπεδο ελέγχου να διαχειριστεί πολλαπλές συσκευές [\[38\]](#). Με τον όρο επίπεδο ελέγχου περιγράφεται το κομμάτι ενός δικτύου που καθορίζει τη δρομολόγηση των δεδομένων μέσα σε αυτό. Επιπλέον είναι υπεύθυνο για:

- Τη διαχείριση του πίνακα δρομολογήσεων.
- Τη διαχείριση του πίνακα προωθήσεων .
- Τον σχεδιασμό των τοπολογιών ενός δικτύου.
- Την εξουσιοδότηση του δρομολογητή στο πως να προωθούνται τα δεδομένα.

Ο όρος επίπεδο προώθησης δεδομένων περιγράφει όλες τις διαδικασίες που γίνονται σε ένα δίκτυο για την προώθηση δεδομένων μεταξύ διεπαφών και βασίζεται στο λειτουργία του επιπέδου ελέγχου. Κυριότερες διαδικασίες του επιπέδου προώθησης δεδομένων αποτελούν οι :

- Πίνακες Δρομολόγησης
- Πίνακες Προώθησης

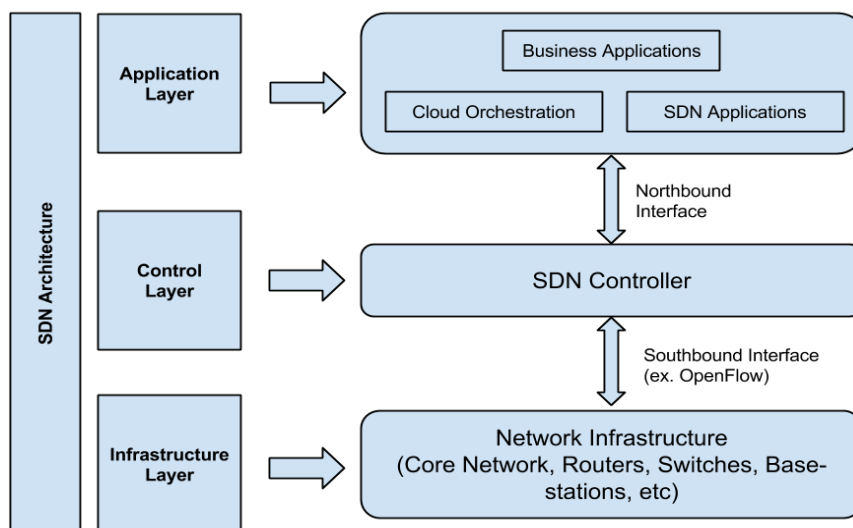
## 2.2 Εξέλιξη SDN

Αρχικά το SDN δοκιμαζόταν σε δημόσια τηλεφωνικά δίκτυα για να παρέχει καλύτερη ανταπόκριση μέσω του διαχωρισμού επιπέδου ελέγχου με επίπεδο προώθησης δεδομένων. Το 2004 γίνανε τα πρώτα βήματα εφαρμογής του SDN αλλά ιστορικά αποδεικνύεται ότι δεν είχαν επιτυχία λόγω του φόβου απώλειας δεδομένων. Το 2008 άρχισε να παίρνει τη μορφή που ξέρουμε σήμερα μέσω τις συνεργασίας των πανεπιστημίων Berkeley και Stanford ως που ήρθε και η επίσημη δημιουργία αυτής της τεχνολογίας το 2011 από τον οργανισμό Open Networking Foundation (ONF). Το SDN αποτελεί μια πολύ σύγχρονη και συνεχώς αναπτυσσόμενη τεχνολογία η οποία άρχισε από το 2016 και έπειτα να διαδίδεται και να χρησιμοποιείται και μέσα στην επόμενη δεκαετία πιστεύεται ότι θα εδραιωθεί ως η κύρια αρχιτεκτονική δικτύου.

## 2.3 Αρχιτεκτονική SDN

Το SDN αρχιτεκτονικά αποτελείται από 3 επίπεδα:

- Επίπεδο Εφαρμογών (Application Layer)
- Επίπεδο Ελέγχου (Control Layer)
- Επίπεδο Προώθησης Δεδομένων (Data/Forwarding/Infrastructure Layer)



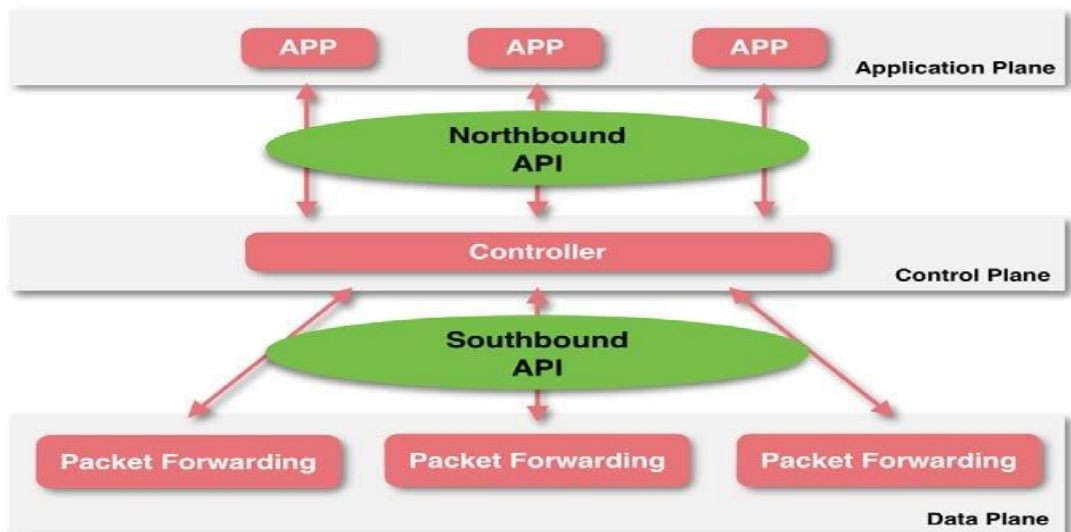
Εικόνα 2.1: Τα επίπεδα αρχιτεκτονικής του δικτύου SDN. Προσαρμογή από [33].

### 2.3.1 Επίπεδο εφαρμογών

Το επίπεδο εφαρμογής αποτελείται από προγράμματα που επικοινωνούν μέσω προγραμματισμού και ρυθμίζουν την συμπεριφορά και τις απαιτήσεις του δικτύου σε σχέση με τον ελεγκτή SDN είτε αυτόματα είτε με τη βοήθεια του χρήστη. Αυτή τη ρύθμιση κατά κύριο λόγο την κάνουν εφαρμογές τύπου Orchestrator. Αυτές οι εφαρμογές περιλαμβάνουν όλους τους κανόνες λειτουργίας και συνεργασίας για ένα δίκτυο.

### 2.3.2 Επίπεδο ελέγχου

Ο ελεγκτής SDN αποτελεί σχηματικά το μεσαίο στρώμα που συνδέει το επίπεδο εφαρμογών και το επίπεδο προώθησης δεδομένων. Η βόρεια διεπαφή (Northbound) αποτελεί τη σύνδεση μεταξύ του ελεγκτή και των εφαρμογών και στην ουσία είναι αυτή που πραγματοποιεί την επαφή μεταξύ κάποιου χρήστη και του επιτρέπει να ρυθμίσει το δίκτυο. Η νότια διεπαφή (Southbound) αποτελεί τη σύνδεση μεταξύ του ελεγκτή και του επιπέδου προώθησης δεδομένων και την παραμετροποίηση των συσκευών που βρίσκονται σε αυτό με τη χρήση πρωτοκόλλων. Αυτό το επίπεδο επεξεργάζεται τις οδηγίες και τις απαιτήσεις που αποστέλλονται από το επίπεδο εφαρμογής μέσω της νότιας διεπαφής και τις προωθεί στα στοιχεία δικτύωσης μέσω της βόρειας διεπαφής. Ανακοινώνει επίσης τις απαραίτητες πληροφορίες που εξήχθησαν από τις συσκευές δικτύωσης στην εφαρμογή για να λειτουργήσει βέλτιστα.



Εικόνα 2.2: Η λειτουργία του επιπέδου ελέγχου μέσω Northbound και Southbound. Προσαρμογή από [52].

### 2.3.3 Επίπεδο προώθησης δεδομένων

Το επίπεδο προώθησης δεδομένων αποτελείται από συσκευές δικτύωσης που ελέγχουν τις δυνατότητες προώθησης και επεξεργασίας δεδομένων στο δίκτυο. Οι συσκευές είναι υπεύθυνες για το χειρισμό πακέτων με βάση τους κανόνες που παρέχει ένας ελεγκτής. Είναι το φυσικό επίπεδο που είναι υπεύθυνο για τη συλλογή των καταστάσεων δικτύου, όπως στατιστικά κίνησης, τοπολογία δικτύου, χρήση δικτύου κ.λπ. καθώς και την αποστολή τους στο επίπεδο ελέγχου. Στο επίπεδο ελέγχου βρίσκεται το Compute. Το Compute ανήκει στο επίπεδο προώθησης δεδομένων και είναι αυτό που επιτρέπει τη δημιουργία και των ελεγχών των προδιαγραφών των εικονικών δικτύων [42].

## 2.4 Χαρακτηριστικά SDN

Τα βασικά χαρακτηριστικά του SDN παρουσιάζονται ως εξής:

- Άμεσα Προγραμματισίμο: Ο έλεγχος δικτύου είναι άμεσα προγραμματιζόμενος, επειδή αποσυνδέεται από τις λειτουργίες προώθησης.
- Ευέλικτο: Η διάσπαση ελέγχου και προώθησης επιτρέπει στους διαχειριστές να προσαρμόζουν δυναμικά τη ροή της κυκλοφορίας σε όλο το δίκτυο για να καλύψουν τις μεταβαλλόμενες ανάγκες.
- Κεντρική διαχείριση: Η αντίληψη δικτύου (λογικά) συγκεντρώνεται σε ελεγκτές SDN που βασίζονται σε λογισμικό και διατηρούν μια καθολική άποψη του δικτύου, η οποία εμφανίζεται σε εφαρμογές και μηχανισμούς πολιτικής ως ενιαίος λογικός switch.
- Δυνατότητα διαμόρφωσης μέσω προγραμματισμού: Το SDN επιτρέπει στους διαχειριστές δικτύου να διαμορφώνουν, να διαχειρίζονται, να ασφαλίζουν και να βελτιστοποιούν τους πόρους του δικτύου πολύ γρήγορα μέσω δυναμικών, αυτοματοποιημένων προγραμμάτων SDN, τα οποία μπορούν να γράψουν οι ίδιοι επειδή τα προγράμματα δεν εξαρτώνται από ιδιόκτητο λογισμικό..
- Υλοποίηση ανοιχτού κώδικα: Το SDN λόγω του ότι αποτελεί υλοποίηση τύπου ανοιχτού κώδικα απλοποιεί το σχεδιασμό και τη λειτουργία του δικτύου, επειδή οι οδηγίες παρέχονται από ελεγκτές SDN αντί για πολλαπλές συσκευές και πρωτόκολλα για προμηθευτές.

## 2.5 Εφαρμογές SDN

Το SDN βρίσκει εφαρμογές σε οργανισμούς και επιχειρήσεις όπου:

- Έχουν εφαρμογές Cloud δεδομένων.
- Η ψηφιακή ασφάλεια είναι κύρια προτεραιότητα.
- Υπάρχει υψηλή κίνηση δεδομένων.
- Η ταχύτητα επεξεργασίας και μεταφοράς δεδομένων είναι κύρια προτεραιότητα [\[22\]](#).

Παραδείγματα γνωστών οργανισμών που κάνουν εφαρμογή της τεχνολογίας SDN στη δικτυακή διαχείρισή τους είναι μεταξύ άλλων οι εξής [\[64\]](#):

- Microsoft
- VMware
- HP
- Dell, Inc.
- Cisco
- Citrix
- Kaspersky Lab
- Fortinet
- Riverbed Technology, Inc.
- Universidad De Granada
- Vodafone Group plc
- Google
- Yahoo
- Facebook
- Twitter
- NSA

## 2.6 Πλεονεκτήματα χρήσης SDN

- Ένα από τα κύρια πλεονεκτήματα του SDN είναι ότι έχει τη δυνατότητα άμεσης εφαρμογής σε εφαρμογές Virtualization. Η χρήση εικονικών μηχανών (VM) είναι άρρηκτα συνδεδεμένη με μεγάλες κινήσεις δεδομένων και το SDN παρέχει τα μέσα για τη διαχείριση VM και αυτών των μεταφορών δεδομένων.

- Μείωση λειτουργικού κόστους επιχειρήσεων και οργανισμών:
  - Λιγότερες αλλαγές εξοπλισμού και συντήρησης
  - Ευκολότεροι πειραματισμοί χωρίς υλικές ζημιές σε περίπτωση λάθους
  - Μείωση λαθών
  - Αποτελεσματικότερη διαχείριση ανθρωπίνου δυναμικού

- Σταθερή γρήγορη και ακριβής μεταφορά δεδομένων
- Καλύτερη διαχείριση Cloud εφαρμογών
- Αυξημένη ασφάλεια δικτύου λόγω χρήσης VM.
- Ολιστική διαχείριση επιχειρήσεων και οργανισμών
- Αυξημένες δυνατότητες εκπαιδευτικού χαρακτήρα σε πανεπιστήμια καθώς μηδενίζει το κόστος εξοπλισμού για εκμάθηση δικτύων.

## 2.7 Μειονεκτήματα χρήσης SDN

- Το κυριότερο μειονέκτημα του SDN είναι ότι παρότι είναι μια ταχύτατα αναπτυσσόμενη τεχνολογία παραμένει ακόμα αρκετά καινούργια χωρίς να έχει επιβεβαιωθεί σε βάθος χρόνου η αποτελεσματικότητα και η αξιοπιστία της.

- Βασίζει το μεγαλύτερο μέρος της ασφάλειάς της στον Controller πράγμα που σημαίνει ότι ο ελεγκτής(controller) πρέπει πάντα να διαχειρίζεται προσεκτικά από συγκεκριμένα άτομα. Αυτό έχει ως αποτέλεσμα την ολική κατάρρευση του SDN σε περίπτωση λάθους στον Controller.

- Όλη η λειτουργία της αρχιτεκτονικής του SDN βασίζεται σε ψηφιακή και όχι φυσική λειτουργία και αυτό έχει ως αποτέλεσμα να μην υπάρχει κάποια φυσική προστασία όπως Firewall που διαθέτουν φυσικά και όχι εικονικά Switch.



- Μπορεί να μπερδευτεί εύκολα εάν δεχθεί επιθέσεις τύπου DOS.
- Παρότι η χρήση του μειώνει αποδεδειγμένα τα έξοδα ενός οργανισμού με το να ξεκινήσει να χρησιμοποιείται η αρχιτεκτονική SDN πρέπει να γίνουν δαπάνες στην αλλαγή όλου του τύπου δικτύου και στήσιμο του από την αρχή.
- Επιπλέον εκπαίδευση προσωπικού για τη διαχείριση δικτύων SDN

## 3. Η λειτουργία SDN

### 3.1 Βασικά λειτουργικά στοιχεία

Το SDN όπως αναφέρθηκε και στο κεφάλαιο 2 αποτελείται δομικά από 3 επίπεδα

- Επίπεδο Εφαρμογών
- Επίπεδο Ελέγχου
- Επίπεδο Προώθησης Δεδομένων

Από τα οποία το επίπεδο ελέγχου είναι το κύριο επίπεδο που επικοινωνεί με τα άλλα δύο.

### 3.2 Ελεγκτές SDN

Ο ελεγκτής σε ένα δίκτυο SDN είναι ο «εγκέφαλος» του δικτύου και από αυτόν καθορίζονται όλες οι παράμετροι και οι κινήσεις του δικτύου καθώς και είναι το βασικό δομικό στοιχείο του επιπέδου ελέγχου. Οι ελεγκτές λειτουργούν μέσω του επιπέδου ελέγχου όπως ακριβώς αναλύθηκε στο κεφάλαιο 2.3.2 . Οι ελεγκτές κατά κύριο λόγο είναι γραμμένοι σε γλώσσα Java και λειτουργούν και διανομές και συστήματα βασισμένα σε Linux.

#### 3.2.1 Είδη ελεγκτών

Οι ελεγκτές χωρίζονται σε ανοιχτού κώδικα οι οποίοι είναι άμεσα προσβάσιμο και επεξεργάσιμο από τον χρήστη και αναπτύσσονται συνεχώς ταχύτητα από ερευνητές σε όλο τον κόσμο και υπάρχουν και οι εμπορικοί ελεγκτές μεγάλων εταιρειών.

Λίστα ελεγκτών ανοιχτού κώδικα:

- Beacon(ανενεργός)
  - Είναι ένας γρήγορος, cross-platform ελεγκτής OpenFlow που βασίζεται σε Java. Το Beacon είναι μια εφαρμογή ανοιχτού κώδικα και ισχυρίζεται ότι έχει χρησιμοποιηθεί σε διάφορα ερευνητικά έργα, τάξεις δικτύωσης και δοκιμαστικές εφαρμογές. Το Beacon για τα έτη 2019-2020 λειτουργούσε ένα πειραματικό κέντρο 100 εικονικών switch και 20 φυσικών και να λειτουργεί για μήνες χωρίς διακοπή λειτουργίας.

- Faucet

- Το Faucet είναι ένας πρακτικός ελεγκτής OpenFlow ανοιχτού κώδικα, ο οποίος επιτρέπει στους χειριστές δικτύου να εκτελούν τα δίκτυά τους με τον ίδιο τρόπο που κάνουν οι συστάδες διακομιστών. Το Faucet μεταφέρει λειτουργίες ελέγχου δικτύου (όπως πρωτόκολλα δρομολόγησης, ανακάλυψη γειτονιάς και αλγόριθμους εναλλαγής) σε ανεξάρτητο προμηθευτή λογισμικό που βασίζεται σε διακομιστές, όπου αυτές οι λειτουργίες είναι εύκολες στη διαχείριση, δοκιμή και επέκταση με σύγχρονες βέλτιστες πρακτικές και εργαλεία διαχείρισης συστημάτων. Το Faucet ελέγχει το πρωτόκολλο OpenFlow 1.3.

- Floodlight(ανενεργός)

- Floodlight Controller είναι ένας ελεγκτής SDN που αναπτύχθηκε από μια ανοιχτή κοινότητα προγραμματιστών, πολλοί από τους οποίους από τα Big Switch Networks, που χρησιμοποιούν με το πρωτόκολλο OpenFlow για να ενορχηστρώνουν ροές κυκλοφορίας σε περιβάλλον δικτύωσης SDN. Το OpenFlow είναι ένα από τα πρώτα και ευρύτερα χρησιμοποιούμενα πρότυπα SDN. ορίζει το ανοιχτό πρωτόκολλο επικοινωνιών σε περιβάλλον SDN που επιτρέπει στον Ελεγκτή SDN (εγκέφαλοι του δικτύου) να μιλά στο επίπεδο προώθησης (switches, δρομολογητές κ.λπ.) για να κάνει αλλαγές στο δίκτυο.

- Lumina(ανενεργός)

- Το Lumina SDN Controller είναι ένας από τους δημοφιλείς ελεγκτές SDN με βάση το OpenDaylight (sprun-off από την Brocade). Δημιουργεί ένα κοινό επίπεδο ελέγχου σε πολλούς τομείς έτσι ώστε οι υπηρεσίες να μπορούν να αναπτυχθούν χρησιμοποιώντας ένα μόνο σύνολο εφαρμογών. Η ίδια υπηρεσία μπορεί να δημιουργηθεί ταυτόχρονα μέσω ενός δρομολογημένου τομέα, ενός εναλλασσόμενου τομέα και πολλών προμηθευτών ταυτόχρονα. Το Network Convergence είναι ένας βασικός διαφοροποιητής για το Lumina SDN Controller.

- Maestro

- Είναι μια επεκτάσιμη πλατφόρμα ελέγχου γραμμένη σε Java, η οποία υποστηρίζει switch πρωτοκόλλου OpenFlow. Διανέμεται ως λογισμικό ανοιχτού κώδικα. Το Maestro παρέχει διεπαφές για την εφαρμογή εφαρμογών αρθρωτού ελέγχου δικτύου. Το Maestro είναι μια πλατφόρμα για την επίτευξη αυτόματων και προγραμματικών λειτουργιών ελέγχου δικτύου.

- NEC

- Ο ελεγκτής NEC έχει σχεδιαστεί για να λειτουργεί με εξοπλισμό δικτύου Open Compute Project (OCP) χαμηλού κόστους. Υποστηρίζει μια ικανότητα αγωγού επεξεργασίας πακέτων για να εξασφαλίσει μια αποτελεσματική, υψηλή απόδοση και ευέλικτη αρχιτεκτονική δικτύου. Υποστηρίζει έως και 10.000 switch.

- NOX/POX

- NOX:Ο πρώτος ελεγκτής SDN που υπήρξε

- POX:Η σημερινή του εξέλιξη.

- Ryu

- Το Ryu είναι ένα λειτουργικό σύστημα για δίκτυα καθορισμένα από λογισμικό. Η Ryu στοχεύει στην παροχή ενός λογικά συγκεντρωτικού ελέγχου και ενός καλά καθορισμένου API που διευκολύνει τους χειριστές να δημιουργήσουν νέες εφαρμογές διαχείρισης και ελέγχου δικτύου. Αναπτύσσεται και συντηρείται από εργαστήρια NTT.

Οι υπόλοιποι ελεγκτές ανοιχτού κώδικα είναι επιγραμματικά οι εξής:

- The Fast Data Project
- Trema
- Vneio/Sdnc
- Lighty-core
- Loom
- Cherry
- ONOS
- Open vSwitch
- OpenContrail
- OpenKilda
- Project Calico
- Floodlight
- RUNOS
- NodeFlow(ανενεργός)

#### Λίστα εμπορικών ελεγκτών:

- OpenDaylight (Ο πιο δημοφιλής εμπορικός ελεγκτής καθώς διατίθεται σε διάφορες εκδοχές του από τις παρακάτω εταιρείες):

- Cisco
- Ericsson
- Lumina Networks

- Big Network Controller(ανενεργός)

- Το Big Network Controller είναι ένας εμπορικός ελεγκτής OpenFlow® που βασίζεται στο Project Floodlight και είναι το θεμέλιο για Open Net Defined Networks (SDN), παρέχοντας ενοποιημένη ευφυΐα δικτύου, επεκτασιμότητα επιχειρήσεων και υψηλή διαθεσιμότητα. Το Big Network Controller χρησιμοποιεί πρωτόκολλα βιομηχανικού προτύπου, όπως το OpenFlow, για να δημιουργήσει ένα κοινό μοντέλο αφαίρεσης και καθολικών δεδομένων για τα υποκείμενα στοιχεία του επιπέδου δεδομένων δικτύου. Όταν συνδυάζεται με ανοιχτές και δημοσιευμένες διεπαφές προγραμματισμού εφαρμογών (API), το Big Network Controller παρέχει μια ευέλικτη και επεκτάσιμη πλατφόρμα για την ανάπτυξη εφαρμογών δικτύου.

- HP

- Το HP Virtual Application Networks SDN Controller είναι το επίκεντρο της αρχιτεκτονικής SDN. Ο ελεγκτής προσφέρει ένα δυναμικό επίπεδο ελέγχου με την ευφυΐα να αυτοματοποιεί και να προγραμματίζει το δίκτυο για να επιτρέψει την ευελιξία του δικτύου. Έχει πλήρη υποστήριξη για το πρωτόκολλο OpenFlow και έχει Open APIs για να επιτρέψει την ανάπτυξη εφαρμογών SDN τρίτων.

- Cisco

- Το Cisco Open SDN Controller είναι μια εμπορική διανομή του OpenDaylight που παρέχει επιχειρηματική ευελιξία μέσω αυτοματοποίησης της υποδομής δικτύου που βασίζεται σε πρότυπα. Αφαιρεί την πολυπλοκότητα της διαχείρισης ετερογενών δικτύων δικτύου για τη βελτίωση της παροχής υπηρεσιών και τη μείωση του λειτουργικού κόστους.

- Ericsson

- Το Ericsson Cloud SDN Controller παρέχει απρόσκοπτη σύνδεση εντός και μεταξύ κέντρων δεδομένων για εικονικούς και φυσικούς και βασισμένους φόρτους εργασίας. Συνδυάζει μοναδικά έναν βιομηχανικό ελεγκτή OpenDaylight με προηγμένες δυνατότητες

δρομολόγησης. Παραδίδεται ως μέρος της λύσης NFVI (NFV Infrastructure) που έχει επαληθευτεί από το σύστημα και παρέχει υπηρεσίες αυτοματισμού δικτύου σε φόρτους εργασίας NFV(Εικονικές λειτουργίες δικτύων).

- IBM

➤ Το IBM Unified Controller παρέχει ένα ύφασμα δικτύου βασισμένο σε OpenFlow με κεντρικό έλεγχο ροών δικτύου και απεριόριστη κινητικότητα εικονικής μηχανής που εφαρμόζεται σε λογισμικό εταιρικής κλάσης. Το λογισμικό ελεγκτή ανακαλύπτει αυτόματα και συνεχώς την τοπολογία του δικτύου OpenFlow και χαρτογραφεί φυσικές και εικονικές ροές κίνησης σε οποιοδήποτε περιβάλλον δικτύου κέντρων δεδομένων που βασίζεται στο OpenFlow. Η IBM δεν είναι ακόμη καθιερωμένη εταιρεία στον κλάδο για ελεγκτές SDN.

Οι υπόλοιποι εμπορικοί ελεγκτές είναι επιγραμματικά οι εξής [\[46\]](#):

- VMware
- Brocade
- Nokia
- Avaya
- Huawei

### 3.2.2 Ο ελεγκτής POX

Ο ελεγκτής POX είναι η εξέλιξη του ελεγκτή NOX ο οποίος ήταν ο πρώτος ελεγκτής που δημιουργήθηκε για SDN από την Nicira η οποία τον έκανε δωρεά στην παγκόσμια ερευνητική κοινότητα με αποτέλεσμα το 2008 να γίνει ανοιχτού κώδικα. Ο POX γράφτηκε σε γλώσσα Python και η χρήση του έχει πολλά πλεονεκτήματα μεταξύ των οποίων:

- Υποστήριξη του πρωτοκόλλου OpenFlow
- Σχεδιασμένος ειδικά για εφαρμογές Virtualization
- Δειγματοληπτικό έλεγχο για την καλύτερη διαχείριση τοπολογιών
- Μπορεί να τρέξει όλες τις τοπολογίες σε συνεργασία με βιβλιοθήκες Python
- Τρέχει στα κυριότερα λειτουργικά συστήματα:
  - Windows

- Linux
- MacOS

### 3.3 Πρωτόκολλα επικοινωνίας

Οι εντολές και οι κανόνες που περιλαμβάνει ένας ελεγκτής προκειμένου να δρομολογήσει δεδομένα προέρχονται από το πρωτόκολλο επικοινωνίας που χρησιμοποιεί. Στην ουσία ένα πρωτόκολλο επικοινωνίας μέσα σε ένα δίκτυο καθορίζει:

- Κανόνες
- Γλώσσα
- Σύνταξη
- Σημασιολογία
- Ρυθμό
- Μεθόδους αντιμετώπισης σφαλμάτων

Όλα τα παραπάνω μπορούν να εφαρμοστούν από:

- Φυσικές συσκευές(Hardware)
- Εικονικές συσκευές ή λογισμικό(Software)
- Συνδυασμό των παραπάνω

#### 3.3.1 Το πρωτόκολλο Openflow

Το πρωτόκολλο OpenFlow είναι το πρώτο και το κύριο πρωτόκολλο εφαρμογών SDN που δημιουργήθηκε από το πανεπιστήμιο του Stanford. Όλα τα υπόλοιπα πρωτόκολλα που έχουν κυκλοφορήσει έπειτα βασίζονται πάνω στο OpenFlow είτε έχοντας επιπλέον δυνατότητες είτε φέροντας κάποιες αλλαγές. Το OpenFlow παρέχει στους ελεγκτές δικτύου τη δυνατότητα να καθορίζουν τη διαδρομή πακέτων και δεδομένων μέσα σε ένα δίκτυο. Διαχωρίζοντας το επίπεδο ελέγχου από το επίπεδο προώθησης δεδομένων πετυχαίνεται η βέλτιστη διαχείριση της κυκλοφορίας χρησιμοποιώντας λίστες ελέγχου πρόσβασης (ACL) , πρωτόκολλα δρομολόγησης και κάνοντας αντιστοίχιση, κατηγοριοποίηση πακέτων δίνοντάς τους προτεραιότητες .Επιπλέον, το Openflow επιτρέπει τη συνεργασία και τον έλεγχο Switches διαφορετικών κατασκευαστών αποστέλλοντας έτσι αυτόματα ένα μεγάλο

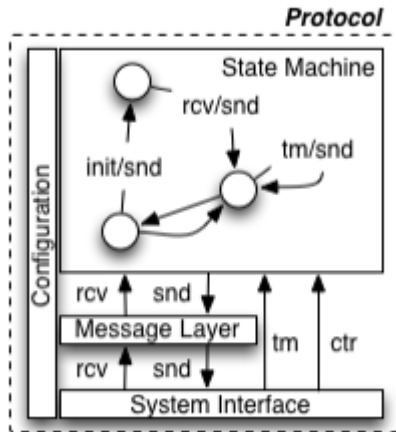
πλεονέκτημα και διευρύνοντας την γκάμα υλικού που μπορεί να χρησιμοποιηθεί σε ένα δίκτυο(Τις περισσότερες φορές ο κάθε κατασκευαστής Switch έχει ξεχωριστές γλώσσες επικοινωνίας μέσω διαφορετικών πρωτοκόλλων και πριν την έλευση του πρωτόκολλου OpenFlow υπήρχε σημαντικό πρόβλημα σε κάθε προσπάθεια επικοινωνίας).Με αυτόν τον τρόπο τα switch που υποστηρίζουν πρωτόκολλο Openflow έχουν ξεκάθαρο πλεονέκτημα λειτουργίας απέναντι στα παραδοσιακά. Η χρήση του πρωτοκόλλου Openflow αποφέρει τα εξής πλεονεκτήματα:

- Καθοδήγηση πληροφοριών χαμηλής προτεραιότητας μέσω διαδρομών χαμηλής χρήσης για να αξιοποιηθούν και αυτές καλύτερα
- Ενίσχυση αποδοτικότητας σε απαιτητικές μεταφορές δεδομένων με υψηλούς ρυθμούς μετάδοσης μέσω τις διευθυνσιοδότησης με στόχο τη σταθερή ροή δεδομένων
- Εύκολος διαχωρισμός και κατηγοριοποίηση δεδομένων
- Ευκολότερο στήσιμο δικτύων
- Ευκολότερη αναβάθμιση δικτύων
- Ευκολότερη κλωνοποίηση δικτύων
- Αυξημένη δυνατότητα βελτίωσης με την πάροδο του χρόνου λόγω της συνεχούς εξέλιξης

Τέλος οι ελεγκτές θα πρέπει να χρησιμοποιούν τη θύρα TCP 6653 για την επιτυχή δημιουργία δημιουργήσουν μιας σύνδεσης. (Παλαιότερες εκδόσεις του πρωτοκόλλου Openflow χρησιμοποίησαν ανεπίσημα τη θύρα 6633). Ενώ ιδιαίτερης μνείας χρήζει η λειτουργία του Openflow που πραγματοποιείται πάντα σε ασφαλές κανάλι [\[65\]](#).

Το πρωτόκολλο Openflow μπορεί να χωριστεί σε τέσσερα στοιχεία: επίπεδο μηνυμάτων(Message Layer),μηχανή κατάστασης(State Machine),διεπαφή συστήματος(System Interface),διαμόρφωση(Configuration) και μοντέλο δεδομένων(Data Model).Η παρακάτω εικόνα και η λίστα απεικονίζουν αυτά τα στοιχεία, την αλληλεπίδραση τους και περιγράφουν τη λειτουργία τους.





Εικόνα 3.1: Λειτουργική δομή πρωτοκόλλου Openflow. (Πηγή: [15]).

- Επίπεδο Μηνυμάτων
  - Το επίπεδο μηνυμάτων είναι ο πυρήνας της στοίβας πρωτοκόλλου. Ορίζει έγκυρη δομή και σημασιολογία για όλα τα μηνύματα. Ένα τυπικό επίπεδο μηνυμάτων υποστηρίζει τη δυνατότητα δημιουργίας, αντιγραφής, σύγκρισης, εκτύπωσης και χειρισμού μηνυμάτων.
- Μηχανή Κατάστασης
  - Η μηχανή κατάστασης καθορίζει τη βασική συμπεριφορά χαμηλού επιπέδου του πρωτοκόλλου. Συνήθως, χρησιμοποιείται για την περιγραφή ενεργειών όπως: διαπραγμάτευση, ανακάλυψη δυνατοτήτων, έλεγχος ροής, παράδοση κ.λπ.
- Διεπαφή συστήματος
  - Η διεπαφή συστήματος καθορίζει τον τρόπο με τον οποίο το πρωτόκολλο αλληλοεπιδρά με τον έξω κόσμο. Προσδιορίζει συνήθως τις απαραίτητες και προαιρετικές διεπαφές μαζί με την προβλεπόμενη χρήση τους, όπως TLS και TCP ως κανάλια μεταφοράς.
- Διαμόρφωση
  - Σχεδόν όλα τα σημεία του πρωτοκόλλου έχουν διαμορφώσεις ή αρχικές τιμές. Η διαμόρφωση μπορεί να καλύπτει οτιδήποτε, από προεπιλεγμένα μεγέθη buffer και διαστήματα απάντησης έως πιστοποιητικά X.509.

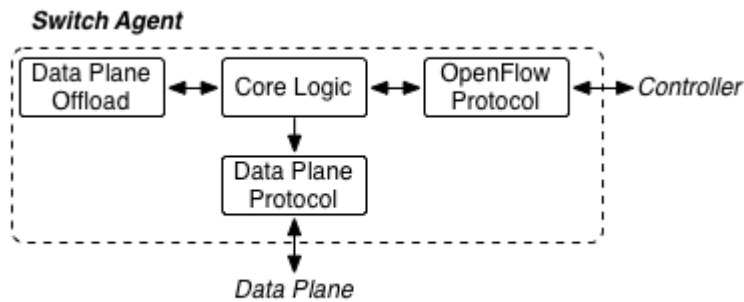
- Μοντέλο δεδομένων

➤ Ένας άλλος τρόπος προσέγγισης του πρωτοκόλλου OpenFlow είναι να κατανοηθεί το υποκείμενο μοντέλο δεδομένων του. Κάθε switch διατηρεί ένα σχεσιακό μοντέλο δεδομένων που περιέχει τα χαρακτηριστικά για κάθε αφαίρεση OpenFlow. Αυτά τα χαρακτηριστικά είτε περιγράφουν μια ικανότητα αφαίρεσης, την κατάσταση διαμόρφωσής της, είτε κάποιο σύνολο τρεχόντων στατιστικών.

Ένα switch που λειτουργεί βάση πρωτοκόλλου Openflow μπορεί να χωριστεί σε δύο στοιχεία : τον μεταγωγέα (switch agent) και το επίπεδο προώθησης δεδομένων (data plane). Ο μεταγωγέας μεταφέρει τις λειτουργίες του πρωτοκόλλου OpenFlow στους ελεγκτές (έναν ή περισσότερους). Επιπλέον επικοινωνεί με το επίπεδο προώθησης δεδομένων κάνοντας χρήση του απαραίτητου εσωτερικού πρωτοκόλλου. Ο μεταγωγέας θα πραγματοποιήσει μετάφραση των εντολών από τον ελεγκτή στο κατάλληλο επίπεδο έτσι ώστε να προωθηθούν στο επίπεδο προώθησης δεδομένων και ταυτόχρονα θα μεταφράσει τις εσωτερικές ρυθμίσεις του επιπέδου προώθησης δεδομένων και θα τις προωθήσει στον ελεγκτή. Το επίπεδο προώθησης δεδομένων εκτελεί όλη την προώθηση πακέτων και τον χειρισμό τους. Ωστόσο, με βάση τη διαμόρφωσή του μερικές φορές θα στείλει πακέτα πίσω στον μεταγωγέα για περαιτέρω χειρισμό.

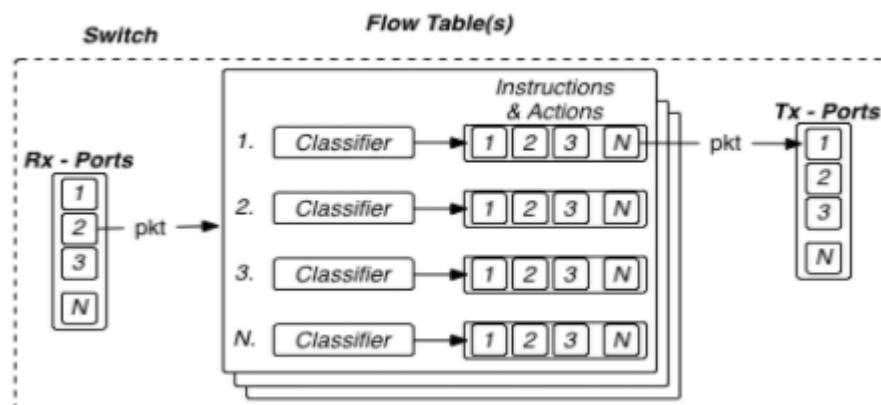
Δομικά ο μεταγωγέας αποτελείται από:

- Πρωτόκολλο Openflow
- Οι δευτερεύουσες λειτουργίες του πρωτοκόλλου Openflow
- Λογική πυρήνα (Core Logic)
- Εδώ διαχειρίζεται το switch και εκτελούνται εντολές σχετικές με το επίπεδο προώθησης δεδομένων βάση των απαιτήσεων π.χ. (Αποφόρτιση επιπέδου δεδομένων , ρύθμιση εύρους κ.α.).
- Μεταφορά επιπέδου προώθησης δεδομένων (Data Plane Offload)
- Σε αυτό το επίπεδο γίνεται εκφόρτωση κάποιας λειτουργίας που υπάρχει στο Openflow αλλά δεν παρέχεται από την υπάρχουσα εφαρμογή του επιπέδου δεδομένων.
- Πρωτόκολλο επιπέδου προώθησης δεδομένων (Data Plane Protocol)
- Το εσωτερικό πρωτόκολλο που χρησιμοποιείται για τη διαμόρφωση της κατάστασης του επιπέδου προώθησης δεδομένων.



Εικόνα 3.2:Λειτουργική δομή του μεταγωγέα. (Πηγή: [15]).

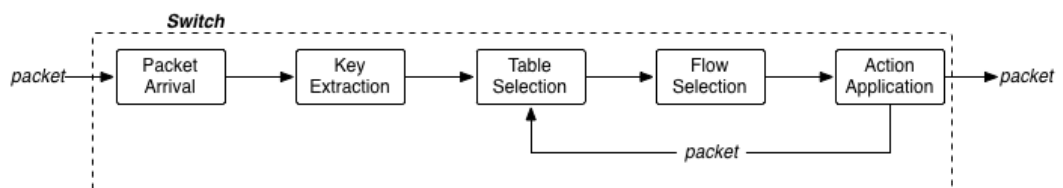
Το επίπεδο προώθησης δεδομένων ενός switch αποτελείται από μερικά απλά πράγματα: θύρες(ports), πίνακες ροής(flow tables), ροές(flows), ταξινομητές(classifiers) και ενέργειες(actions). Τα πακέτα φτάνουν και αφήνουν το σύστημα μέσω των θυρών. Διαφορετικές εκδόσεις του πρωτοκόλλου υποστηρίζουν διαφορετικούς τύπους θυρών, ιδιοτήτων και διαμορφώσεων. Τα πακέτα αντιστοιχίζονται με τις κατάλληλες ροές σε πίνακες ροής χρησιμοποιώντας τους ταξινομητές. Οι ροές περιέχουν σύνολα ενεργειών που εφαρμόζονται σε κάθε πακέτο που αντιστοιχίζεται.



Εικόνα 3.3:Δομή του επιπέδου προώθησης δεδομένων. (Πηγή: [15]).

Κάθε πακέτο βιώνει την ίδια συμπεριφορά στην πορεία του στο επίπεδο προώθησης δεδομένων του switch. Όταν φτάνει ένα πακέτο, δημιουργείται ένα κλειδί που περιέχει και τις πληροφορίες από το πακέτο, όπως τις τιμές ορισμένων πεδίων πρωτοκόλλου, καθώς και τα μεταδεδομένα(metadata) που συλλέγονται για το πακέτο, όπως θύρα άφιξης, ώρα άφιξης κ.λπ. Το κλειδί χρησιμοποιείται για την επιλογή μιας συγκεκριμένης ροής σε έναν πίνακα ροής και οι σχετικές ενέργειες εφαρμόζονται στο πακέτο. Ένα σύνολο ενεργειών μπορεί να ρίξει, να μεταλλάξει, να τοποθετήσει σε ουρά, να προωθήσει ή να κατευθύνει αυτό το πακέτο σε έναν νέο πίνακα ροής. Η διαδρομή του πακέτου είναι η εξής:

- Άφιξη πακέτου
  - Τα πακέτα φτάνουν στις θύρες, όπου αυτές μπορεί να είναι φυσικές ή εικονικές. Είναι σημαντικό να συγκρατούνται οι πληροφορίες σχετικά με τις θύρες άφιξης για επεξεργασία βάσει πηγής αργότερα κατά τη διαδρομή των δεδομένων στον αγωγό.
- Δημιουργία κλειδιού
  - Κάθε πακέτο μετά την άφιξή του περιέχει ένα μικρό μέρος μεταδεδομένων, που ονομάζεται κλειδί. Το κλειδί περιλαμβάνει πολλά πεδία από το πακέτο, καθώς και δευτερεύοντες πληροφορίες όπως: θέση του πακέτου με ρυθμιστή(buffer), τιμές κεφαλίδας, θύρα άφιξης, ρολόι άφιξης κ.λπ. Το κλειδί είναι αυτό που τροφοδοτείται στον υπόλοιπο αγωγό.
- Επιλογή πίνακα
  - Το κλειδί χρησιμοποιείται για αναζήτηση πίνακα. Ωστόσο, ενδέχεται να υπάρχουν πολλοί πίνακες. Σε αυτήν την περίπτωση πρέπει πρώτα να επιλεγεί ένας πίνακας. Όταν ένα πακέτο πραγματοποιεί διαδρομή μέσω του αγωγού για πρώτη φορά, ο πρώτος πίνακας επιλέγεται από προεπιλογή. οι επόμενοι πίνακες μπορούν να επιλεγούν μέσω ενεργειών ή λόγω ελλείψεων θέσεων.
- Επιλογή ροής
  - Το κλειδί χρησιμοποιείται για την επιλογή μιας συγκεκριμένης ροής σε έναν πίνακα. Η πρώτη ροή ενός πίνακα όπου ο ταξινομητής καταλαμβάνει το κλειδί γίνεται η επιλεγμένη ροή.
- Ενέργειες εφαρμογής
  - Κάθε σειρά περιέχει ένα σύνολο ενεργειών για εφαρμογή σε όλα τα πακέτα που έχουν περάσει. Όταν επιλέγεται ροή, οι ενέργειές της εφαρμόζονται στο αντίστοιχο πακέτο. Οι ενέργειες ενδέχεται να τροποποιήσουν την κατάσταση των πακέτων ή / και να επηρεάσουν τη μεταχείριση του πακέτου.



Εικόνα 3.4:Ο κύκλος ζωής των πακέτων στο επίπεδο προώθησης δεδομένων. (Πηγή: [15]).

### 3.4 Τοπολογίες SDN

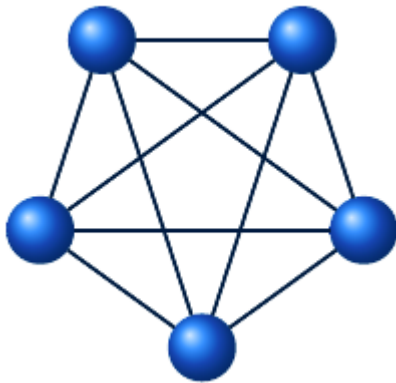
Με τον όρο τοπολογία ορίζουμε τη γεωμετρική αναπαράσταση της σύνδεσης συσκευών σε ένα δίκτυο.

#### 3.4.1 Είδη τοπολογιών SDN

Οι βασικές τοπολογίες δικτύων είναι:

- Mesh (Πλέγματος)

Στην τοπολογία πλέγματος οι συσκευές ενώνονται μεταξύ τους με σύνδεση τύπου point-to-point δηλαδή από τη μια άκρη στην άλλη μιας συσκευής μεταφέρονται δεδομένα που αφορούν μόνο αυτές τις δύο συσκευές και όχι κάποια άλλη . Αυτό έχει σαν αποτέλεσμα όταν θέλουμε να στήσουμε ένα δίκτυο χρησιμοποιώντας την τοπολογία πλέγματος το οποίο αποτελείται από  $n$  αριθμό συσκευών ,κάθε συσκευή στο τέλος πρέπει να είναι συνδεδεμένη με  $(n-1)$  συσκευές στο δίκτυο και ο αριθμός των συνδέσμων μεταξύ συσκευών θα πρέπει να είναι  $n(n-1)/2$ .



Εικόνα 3.5 Απεικόνιση τοπολογίας πλέγματος

Πλεονεκτήματα τοπολογίας πλέγματος:

- Δεν υπάρχουν προβλήματα μετάδοσης δεδομένων καθώς δεν παρεμβαίνει κάτι στην επικοινωνία μεταξύ 2 συσκευών.
- Αξιόπιστη διότι εάν για κάποιο λόγο χαλάσει ένας σύνδεσμος δεν επηρεάζονται οι υπόλοιπες συσκευές.
- Ασφαλής διότι λόγω του ότι δεν υπάρχει κάτι μεταξύ 2 συσκευών πέρα από τον σύνδεσμο δεν μπορεί κάποιος επιπλέον να αποκτήσει ανεπιθύμητη πρόσβαση.

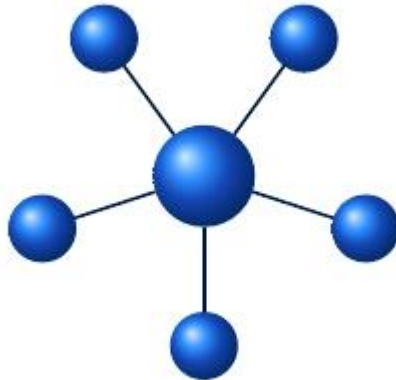
- Εύκολη διάγνωση λαθών.

Μειονεκτήματα τοπολογίας πλέγματος:

- Πληθώρα καλωδίων μεταξύ συσκευών
- Επειδή κάθε συσκευή πρέπει να είναι συνδεδεμένη με τις υπόλοιπες απαιτείται τεράστιος αριθμός θυρών σε κάθε συσκευή.
- Δεν μπορεί να δημιουργηθεί δίκτυο με πολύ μεγάλο αριθμό συσκευών.

- Star (Αστέρα)

Στην τοπολογία αστέρα όλες οι συσκευές είναι συνδεδεμένες σε έναν κόμβο(hub) που βρίσκεται στο κέντρο του δικτύου. Η τοπολογία αστέρα δεν επιτρέπει την επικοινωνία μεταξύ συσκευών καθώς κάθε συσκευή πρέπει πρώτα να επικοινωνήσει με τον κόμβο και μετά ο κόμβος να επικοινωνήσει με την άλλη συσκευή.



Εικόνα 3.6:Απεικόνιση τοπολογίας αστέρα

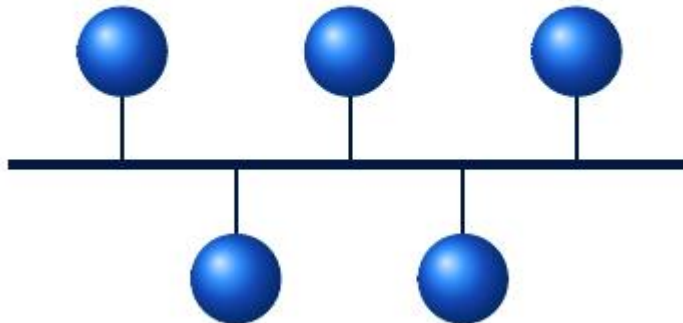
Πλεονεκτήματα τοπολογίας αστέρα:

- Χαμηλό κόστος καθώς σε κάθε συσκευή απαιτείται η χρήση μόνο μιας θύρας και μόνο ενός συνδέσμου.
- Εύκολη εγκατάσταση και δημιουργία.
- Μικρό πλήθος καλωδίων και συνδέσεων.
- Σε περίπτωση ελαττώματος σε έναν κόμβο οι υπόλοιποι δεν παθαίνουν κάτι.
- Εύκολη εύρεση λαθών.

Μειονεκτήματα τοπολογίας αστέρα:

- Σε περίπτωση βλάβης στον κόμβο καταρρέει ολόκληρη η τοπολογία.
- Ο κόμβος έχει αυξημένη ανάγκη συντήρησης και καταναλώνει περισσότερους πόρους.

- Bus (Διαύλου, επίσης συναντάται και με την ονομασία Linear που με αυτή την ονομασία αναγνωρίζεται από τον εξομοιωτή Mininet και θα χρησιμοποιηθεί στη συνέχεια της εργασίας). Στην τοπολογία διαύλου υπάρχει ένα κύριο καλώδιο και όλες οι συσκευές συνδέονται σε αυτό.



Εικόνα 3.7:Απεικόνιση τοπολογίας διαύλου

Πλεονεκτήματα τοπολογίας διαύλου:

- Εύκολη εγκατάσταση.
- Ακόμα λιγότερα καλώδια και σύνδεσμοι από ότι στην τοπολογία αστέρα και πλέγματος.

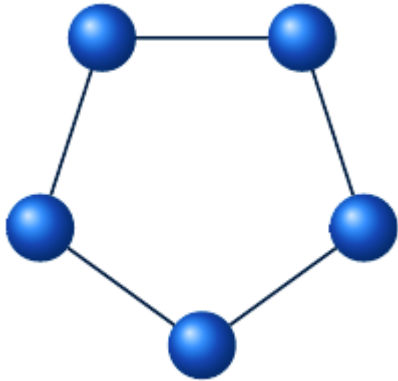
Μειονεκτήματα τοπολογίας διαύλου:

- Δυσκολία ελέγχου λαθών
- Μεγάλος περιορισμός στο μήκος του καλωδίου και στον αριθμό των κόμβων που συνδέονται πάνω σε αυτό.

- Ring (Δακτυλίου)

Στην τοπολογία δακτυλίου οι συσκευές συνδέονται μεταξύ τους κυκλικά και κάθε συσκευή συνδέεται με άλλες 2 στις 2 πλευρές της μέσω συνδέσμων point-to-point. Εάν μια συσκευή θέλει να στείλει δεδομένα σε άλλη συσκευή, τότε στέλνει τα δεδομένα σε μία

κατεύθυνση, κάθε συσκευή σε τοπολογία δακτυλίου έχει ένα repeater(επαναλήπτη), εάν τα ληφθέντα δεδομένα προορίζονται για άλλη συσκευή, τότε το repeater προωθεί αυτά τα δεδομένα έως ότου τα λάβει η προβλεπόμενη συσκευή.



Εικόνα 3.8:Απεικόνιση τοπολογίας δακτυλίου

Πλεονεκτήματα τοπολογίας δακτυλίου:

- Εύκολη εγκατάσταση
- Υπερβολικά εύκολη αντικατάσταση συσκευής σε περίπτωση ελαττώματος καθότι για να αλλαχθεί μια συσκευή χρειάζεται μόνο να βγουν 2 σύνδεσμοι.

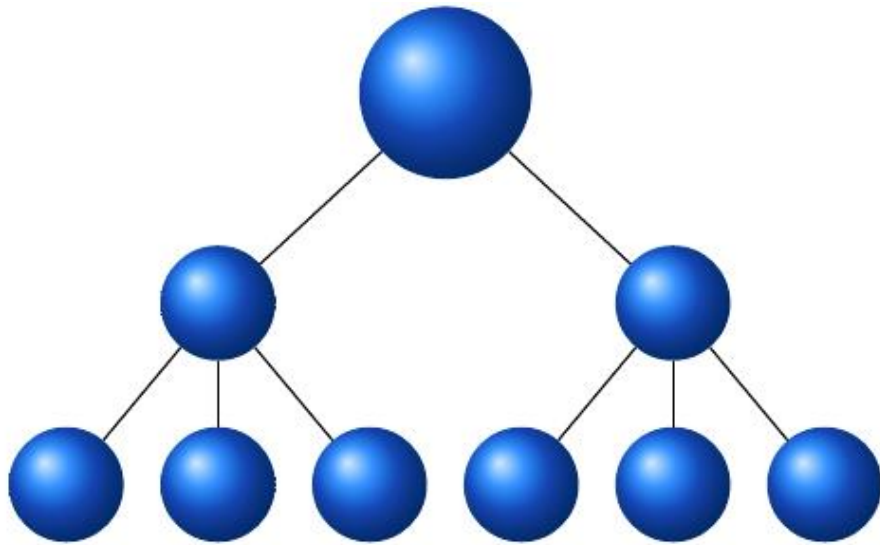
Μειονεκτήματα τοπολογίας δακτυλίου:

- Η βλάβη σε έναν σύνδεσμο μπορεί να προκαλέσει ολική αποτυχία στο δίκτυο.
- Επειδή όλα τα δεδομένα δρομολογούνται μέσα στον δακτύλιο υπάρχει πρόβλημα δρομολόγησης δεδομένων.

- Hybrid (Υβριδική)

Ο συνδυασμός οποιονδήποτε από τις 2 παραπάνω τοπολογίες αποτελεί μια υβριδική τοπολογία. Ένα παράδειγμα υβριδικής τοπολογίας είναι η τοπολογία TREE(δέντρο) η οποία συνδυάζει την τοπολογία διαύλου και αστέρα. Στην πράξη δίκτυα τύπου αστέρα ενώνονται μεταξύ τους μέσω δικτύων διαύλου.





Εικόνα 3.9:Απεικόνιση υβριδικής τοπολογίας .

Πλεονεκτήματα υβριδικών τοπολογιών:

- Μεγάλη προσαρμοστικότητα στη δημιουργία υβριδικού δικτύου με βάση τις απαιτήσεις του.
- Μπορούμε να προσθέσουμε επιπλέον δίκτυα σε ήδη υπάρχοντα.

Μειονεκτήματα υβριδικών τοπολογιών:

- Δύσκολη εύρεση λαθών.
- Δύσκολη εγκατάσταση.
- Περίπλοκη σχεδίαση
- Υψηλό κόστος συντήρησης.

- Custom (Προσαρμοσμένων χαρακτηριστικών)

Στον εξομοιωτή Mininet που θα χρησιμοποιηθεί παρακάτω στην εργασία για την υλοποίηση τοπολογιών δίνεται η δυνατότητα μέσω προγραμματισμού PYTHON για δημιουργία τοπολογιών προσαρμοσμένων χαρακτηριστικών που προσαρμόζονται σε όλες τις ανάγκες.

- Erdos Renyi

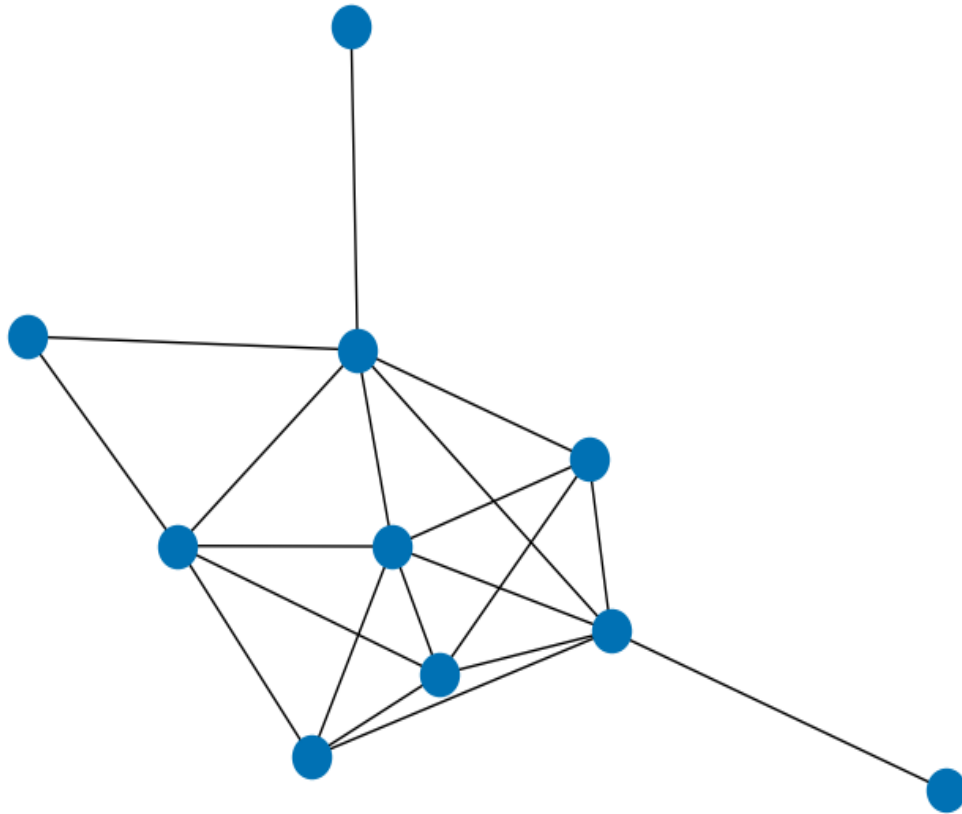
Στη θεωρία γραφημάτων που αποτελεί κλάδο των μαθηματικών, το μοντέλο Erdos Renyi είναι ένα από τα δύο μοντέλα δημιουργίας τυχαίων γραφημάτων. Οφείλει το όνομά του στους Ούγγρους μαθηματικούς Paul Erdos και Alfred Renyi, οι οποίοι εισήγαγαν για πρώτη φορά ένα από τα μοντέλα το 1959, ενώ ο Edgar Gilbert παρουσίασε το άλλο μοντέλο ταυτόχρονα και ανεξάρτητα από τους Erdos και Renyi. Στο μοντέλο των Erdos και Renyi, όλα τα γραφήματα με συγκεκριμένο αριθμό κορυφών είναι εξίσου πιθανά να δημιουργηθούν. Στο μοντέλο που εισήγαγε ο Gilbert, κάθε άκρη έχει μια σταθερή πιθανότητα να υπάρχει ή να απουσιάζει, ανεξάρτητα από τα άλλα άκρα. Αυτά τα μοντέλα μπορούν να χρησιμοποιηθούν στην πιθανολογική μέθοδο για να αποδείξουν την ύπαρξη γραφημάτων που ικανοποιούν διάφορες ιδιότητες ή για να παρέχουν έναν αυστηρό ορισμό για μια ιδιότητα που εμφανίζεται για σχεδόν όλα τα γραφήματα.

➤ Ορισμός:  $G(n, p)$  είναι ένα τυχαίο γράφημα με  $n$  κορυφές όπου κάθε πιθανό ακμή έχει πιθανότητα  $p$  ύπαρξης. Ο αριθμός των ακμών σε ένα γράφημα

$G(n, p)$  είναι μια τυχαία μεταβλητή με την αναμενόμενη τιμή.

$$\binom{n}{2}p.$$

Ένα στενά συνδεδεμένο μοντέλο έχει τη μορφή  $GE(n, e)$ . Από όλα τα πιθανά γραφήματα με  $n$  κορυφές και ακριβώς  $e$  ακμές, επιλέγεται ένα τυχαία. Τα δύο μοντέλα έχουν πολύ παρόμοιες ιδιότητες αλλά θα επικεντρωθούμε κυρίως στο μοντέλο Erdos Renyi  $G(n, p)$  όπου θα ορίζουμε ως  $n$  τις κορυφές (κόμβους του δικτύου) και ως  $p$  την πιθανότητα σύνδεσης μεταξύ κόμβων. Αξίζει να σημειωθεί ότι βάση λειτουργίας τα δίκτυα Erdos Renyi λειτουργούν με βάση τα πραγματικά δίκτυα γιατί πάντα υπάρχει η πιθανότητα μη επικοινωνίας ή μη επιτυχούς σύνδεσης και για αυτό επιλέχθηκε αυτός ο τύπος τοπολογίας από τις custom τοπολογίες ακριβώς γιατί κάθε φορά το γράφημα που υλοποιείται είναι τυχαίο.



Εικόνα 3.10: Τυχαίο γράφημα Erdos Renyi 10 κόμβων

### 3.5 Παράδειγμα εφαρμογής SDN

Ένας από τους πιο γνωστούς οργανισμούς και κάνει χρήση δικτύου SDN και τα εξελίσσει είναι η Google. Τα τελευταία χρόνια η Google έχει κάνει μεγάλη εξέλιξη σε μεταφορές δεδομένων μέσω δικτύων SDN και στο πως τα διαχειρίζεται το επίπεδο ελέγχου. Η Google επέλεξε να χρησιμοποιήσει αρχιτεκτονική SDN διότι διαθέτει τη μεγαλύτερη υπηρεσία cloud δεδομένων αυτή τη στιγμή και ήθελε αξιοπιστία και ταχύτητα. Το δίκτυο cloud της εκτείνεται και στις 5 Ηπείρους. Αυτή η εφαρμογή SDN Cloud (Cloud CDN) λειτουργεί ως εξής, χρησιμοποιεί HTTP(S) εξισορροπητές φόρτου (load balancing) ως προέλευση για περιεχόμενο με δυνατότητα προσωρινής αποθήκευσης. Το ασφαλές πρωτόκολλο μεταφοράς υπερκειμένου (HTTPS) είναι η ασφαλής έκδοση του HTTP, το οποίο είναι το κύριο πρωτόκολλο που χρησιμοποιείται για την αποστολή δεδομένων μεταξύ ενός προγράμματος περιήγησης ιστού και ενός ιστότοπου. Το HTTPS είναι κρυπτογραφημένο προκειμένου να αυξηθεί η ασφάλεια της μεταφοράς δεδομένων. Αυτό είναι ιδιαίτερα σημαντικό όταν οι χρήστες μεταδίδουν ευαίσθητα δεδομένα, όπως όταν συνδέονται σε τραπεζικό λογαριασμό, υπηρεσία email ή πάροχο ασφάλισης υγείας. Οποιοσδήποτε ιστότοπος, ειδικά εκείνοι που

απαιτούν διαπιστευτήρια σύνδεσης, πρέπει να χρησιμοποιούν HTTPS. Στα σύγχρονα προγράμματα περιήγησης ιστού όπως το Chrome, ιστότοποι που δεν χρησιμοποιούν HTTPS επισημαίνονται διαφορετικά από αυτούς που είναι. Αναζητήστε ένα πράσινο λουκέτο στη γραμμή διευθύνσεων URL για να δηλώσετε ότι η ιστοσελίδα είναι ασφαλής. Τα προγράμματα περιήγησης στο Web λαμβάνουν σοβαρά υπόψη το HTTPS. Ένας εξωτερικός εξισορροπητής φόρτωσης HTTP(S) μπορεί να προσφέρει ένα συνδυασμό στατικού και δυναμικά δημιουργημένου περιεχομένου στους χρήστες μέσω μιας καθολικής διεύθυνσης IP από ομάδες παρουσιών εικονικής μηχανής Compute Engine (VM), κάδους Cloud Storage ή και τα δύο. Με απλά λόγια μέσω ενός εικονικού δικτύου μπορείς να παραμετροποιήσεις ή να προβάλλεις το περιεχόμενό του οπουδήποτε στον κόσμο. Στην πράξη εισάγεται η πληροφορία μέσω του επιπέδου εφαρμογών, το επίπεδο ελέγχου εξισορροπεί τον φόρτο του δικτύου και μέσω του επιπέδου προώθησης δεδομένων χρησιμοποιούνται τα κατάλληλα μέσα όπου ανάλογα την περίπτωση αλλάζουν και οι προδιαγραφές των μέσων (VoIP τηλεφωνία, video streaming/conferencing κ.α.) και γίνεται πάλι εξισορρόπηση του φόρτου του δικτύου προκειμένου όλοι οι χρήστες να έχουν τη μέγιστη δυνατή ταχύτητα και ποιότητα. Οι κανόνες προώθησης πακέτων ελέγχουν την κίνηση βάση θυρών(ports), πρωτοκόλλων και διεύθυνσης IP πριν πραγματοποιήσουν τη δρομολόγηση.



Εικόνα 3.11: Παγκόσμιο κάλυψη από το δίκτυο Google CDN. (Πηγή: [20]).

Πέρα από τις εφαρμογές τύπου Cloud η Google χρησιμοποιεί τεχνολογίες SDN και στα δίκτυα ευρείας περιοχής που έχει.(Wide Area Network-WAN). Τα σύγχρονα δίκτυα

ευρείας περιοχής έχουν σημαντικό μερίδιο ευθύνης στην απόδοση και την αξιοπιστία του Διαδικτύου, παρέχοντας terabits/sec συνολικού εύρους ζώνης σε χιλιάδες μεμονωμένους συνδέσμους. Οι σύνδεσμοι WAN είναι ακριβοί επειδή η απώλεια δεδομένων σε δρομολογητές(router) WAN αγγίζουν το 100% αποτελεσματικότητας μετάδοσης πληροφορίας και έτσι αποτελούνται από εξειδικευμένου επαγγελματικό εξοπλισμό υψηλής διαθεσιμότητας. Εν κατακλείδι τα WAN αντιμετωπίζουν όλα τα κομμάτια(bits) πληροφορίας το ίδιο και ενώ αυτό έχει πολλά οφέλη, όταν η αναπόφευκτη αποτυχία συμβαίνει, όλες οι εφαρμογές χρίζουν ισάξιο ποσοστό διαχείρισης, παρά την εξαιρετικά ευαίσθητη μεταβλητότητά τους σε θέματα χωρητικότητας. Δεδομένων αυτών των εκτιμήσεων, συνήθως σύνδεσμοι WAN που φτάνουν έως 30-40% μέση χρήση προκειμένου να υπάρχουν οι διαθέσιμοι πόροι να καλύψουν οποιαδήποτε λάθη προκύψουν στην πορεία. Αυτό σημαίνει ότι χάνεται ένα 60-70% πιθανής επιπλέον χρήσης για χάρη της αξιοπιστίας. Αντιμετωπίσαμε αυτά τα γενικά έξοδα για τη δημιουργία ενός WAN που συνδέει πολλά κέντρα δεδομένων με σημαντικές απαιτήσεις εύρους ζώνης. Ωστόσο το σύγχρονο δίκτυο της Google τύπου WAN έχει επιπλέον χαρακτηριστικά λειτουργίας που ξεπερνάει ότι μειονεκτήματα υπήρχαν μέχρι σήμερα με τη χρήση αρχιτεκτονικής SDN. Ο σχεδιασμός του δικτύου επικεντρώνεται:

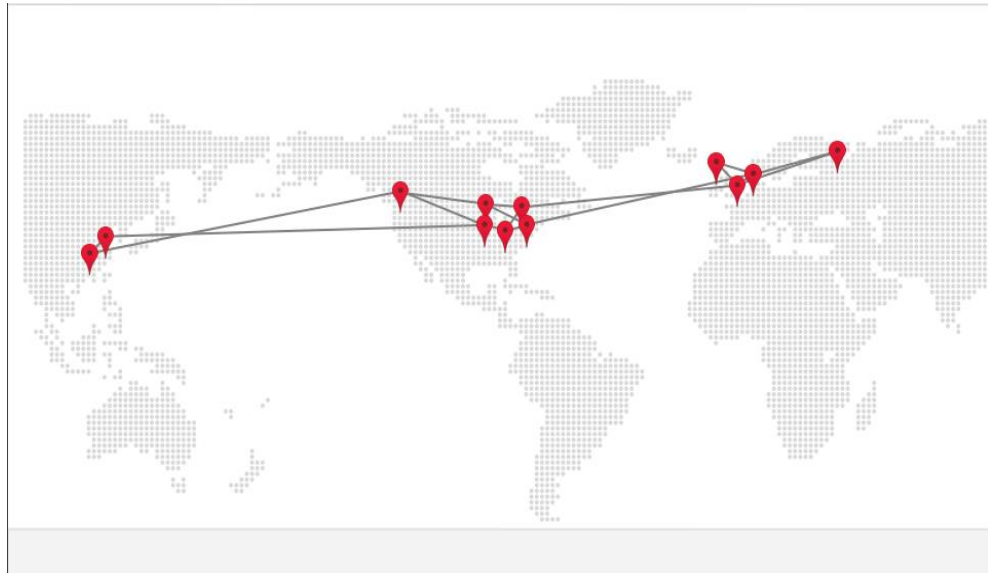
- Στην εξαφάνιση των αποτυχιών των οποίων οι επιπτώσεις πρέπει να μην επηρεάζουν τις τελικές εφαρμογές.
- Οι συσκευές να εξάγουν δεδομένα μέσω μιας διεπαφής(Interface) στο πρόγραμμα καταχωρήσεων του επιπέδου προώθησης δεδομένων που ελέγχεται από το επίπεδο ελέγχου.

Έτσι δημιουργήθηκε το WAN B4 της Google, χρησιμοποιώντας SDN αρχιτεκτονική και πρωτόκολλο ελέγχου Openflow για τη διαχείριση των συσκευών(switches, routers κ.α) . Συγκεκριμένα το SDN ελέγχει

- Την προτεραιότητα εφαρμογών στο δίκτυο από άκρη σε άκρη και τον περιορισμό των διαθέσιμων πόρων για επερχόμενη κίνηση
- Διαχωρισμό μιας πληροφορίας σε πολλά διαφορετικά μονοπάτια για την επίτευξη μέγιστου εύρους και όμοιας κατανομής δικτύου.
- Κατανομή εύρους σε περίπτωση λάθους.

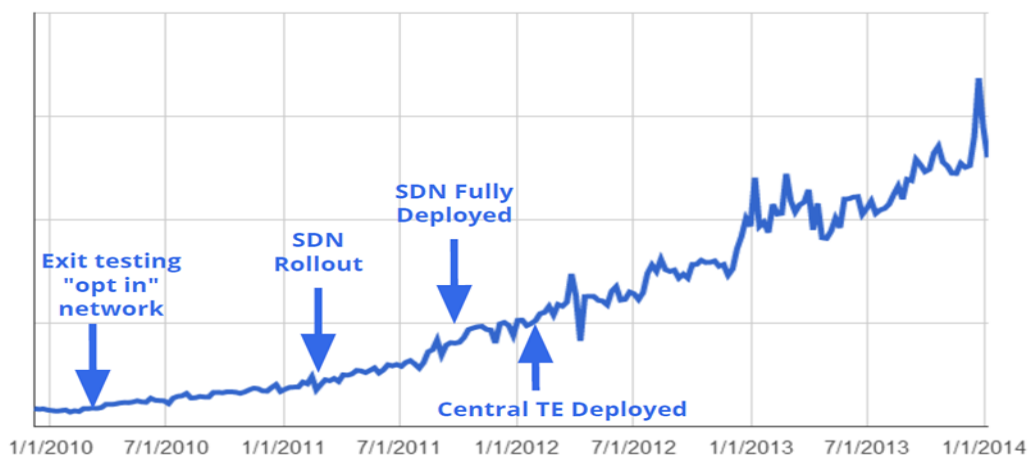
Οι δυνατότητες επιτρέπουν σε πολλούς συνδέσμους Google B4 να λειτουργούν σχεδόν 100% στο μεγαλύτερο πλήθος τους και η μέση χρήση να ξεπερνάει το 70% για

μεγάλες χρονικές περιόδους πράγμα που σημαίνει ότι υπάρχει ένας διπλασιασμός τουλάχιστον σε αύξηση απόδοσης. Το WAN B4 της Google εξελίσσεται από το 2013 και έχει τους ταχύτερους ρυθμούς ανάπτυξης μέχρι τώρα σε σχέση με άλλους οργανισμούς.



Εικόνα 3.12:Χάρτης δικτύου WAN B4 της Google [20].

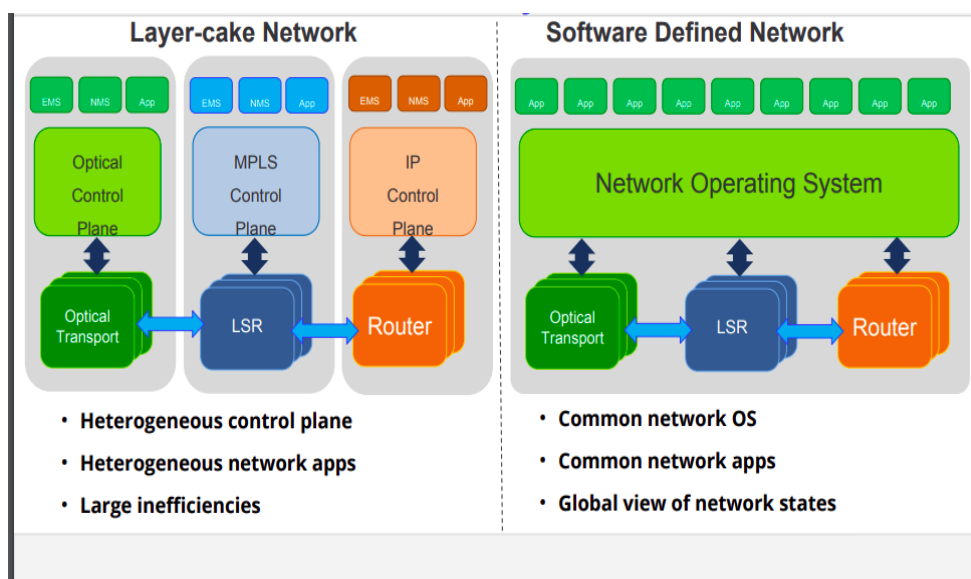
Στην παρακάτω εικόνα βλέπουμε την κίνηση στο δίκτυο της Google πριν την υιοθέτηση SDN τεχνολογιών και κατά τη διάρκεια πως αυξάνεται μαζί με την αποδοτικότητά του.



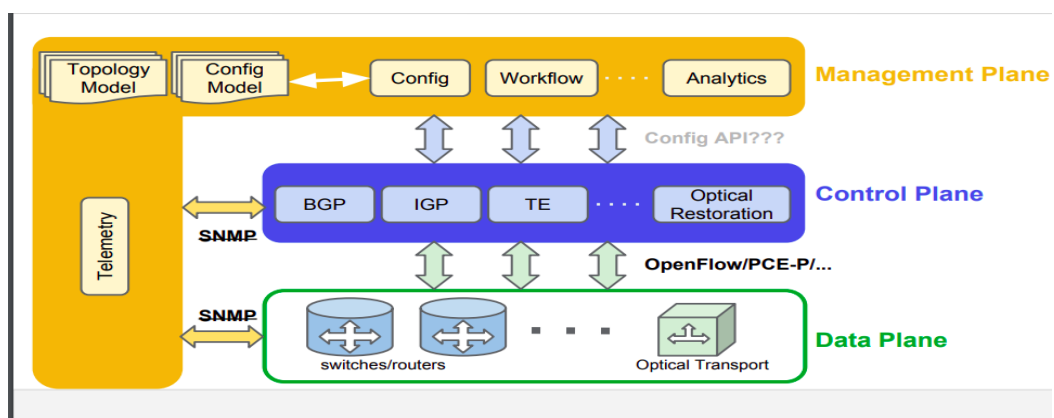
Εικόνα 3.13:Αύξηση κίνησης και αποδοτικότητας στο δίκτυο της Google με την υλοποίηση SDN τεχνολογιών. (Πηγή: [20]).

Στην παρακάτω εικόνα παρατηρούμε μερικές διαφορές δομικά στην εξέλιξη των παραδοσιακών δικτύων σε SDN. Στην SDN εφαρμογή:

- Υπάρχει ένα κοινό λογισμικό που ελέγχει όλες τις συσκευές και παραμέτρους του δικτύου
- Κοινές δικτυακές εφαρμογές
- Καλύτερη προβολή και κατανόηση των καταστάσεων του δικτύου



Εικόνα 3.14: Πίνακας διαφορών παραδοσιακών δικτύων με ένα σύγχρονο SDN δίκτυο. (Πηγή: [20]).



Εικόνα 3.15: Ανατομία SDN Google (Πηγή: [20]).

- Επίπεδο εφαρμογών
  - Περιλαμβάνει τις ρυθμίσεις των εφαρμογών το μοντέλο της τοπολογίας κ.α.

- Επίπεδο ελέγχου

Περιλαμβάνει τα απαραίτητα πρωτόκολλα λειτουργίας και επικοινωνίας όπως:

- BGP(Border Gateway Protocol):Αυτό είναι ένα πρωτόκολλο εξωτερικής δρομολόγησης που μέσω αυτού γίνεται δυνατή η δρομολόγηση πακέτων και δεδομένων μεταξύ αυτόνομων συστημάτων.

- IGP(Interior Gateway Protocol):Το πρωτόκολλο δρομολόγησης δεδομένων που υπάρχει στο εσωτερικό ενός αυτόνομου συστήματος.

- Optical Restoration: Ενίσχυση ταχύτητας και εύρους σε μεταφορά σύνθετων πληροφοριών μεταξύ μεγάλων δικτύων και αποστάσεων.

- Επίπεδο προώθησης δεδομένων

- Περιλαμβάνει τις συσκευές που απαιτούνται για τη μεταφορά τις πληροφορίας(switches, routers κ.α.) η οποίες είναι είτε φυσικές είτε εικονικές [\[20\]](#).



## 4. Περιβάλλον εξομίωσης SDN

Σε αυτό το κεφάλαιο θα γίνει μια σύντομη παρουσίαση του κάθε λογισμικού που θα χρησιμοποιηθεί στην παρούσα εργασία καθώς και των τεχνικών προδιαγραφών του χρησιμοποιηθέντος συστήματος.

### 4.1 Mininet OS

Το Mininet είναι ένα λειτουργικό σύστημα βασισμένο σε Linux-Kernel το οποίο κατά κύριο λόγο χρησιμοποιείται σε VM και είναι το πιο διαδεδομένο λογισμικό για την υλοποίηση δικτύων SDN. Το Mininet 1.0 δημιουργήθηκε από τους φοιτητές Bob Lantz και Brandon Heller του University of Stanford το 2008 και βασιζόταν στο πειραματικό μοντέλου που είχε εξελίξει ο πρώτος. Το Mininet 2.0 το οποίο μοιάζει πολύ με τις εκδόσεις Mininet που χρησιμοποιούνται σήμερα κυκλοφόρησε τον Νοέμβριο του 2012. Μέσω του Mininet ο χρήστης έχει τη δυνατότητα να δημιουργήσει ένα εικονικό δίκτυο με πραγματικά χαρακτηριστικά μέσω κώδικα. Η δημιουργία δικτύων μέσω Mininet εμπίπτει στην κατηγορία των SDN και περιλαμβάνει πληθώρα δυνατοτήτων τα οποία είναι ήδη προ εγκατεστημένα ή μπορούν να προστεθούν με την μορφή addons. Με απλά λόγια το Mininet έχει τη δυνατότητα υλοποίησης κάθε μορφής δικτύου SDN ανεξαρτήτως, χαρακτηριστικών, τοπολογιών κ.ο.κ. Όλες οι λειτουργίες του Mininet εκτελούνται μέσω Linux-Based Terminal και απαιτούνται βασικές γνώσεις προγραμματισμού, δικτύων και αρχιτεκτονικής λειτουργικών συστημάτων. Το Mininet είναι κατασκευασμένο εξολοκλήρου χρησιμοποιώντας τη γλώσσα Python εκτός από ένα μικρό Utility του που κατασκευάστηκε με τη χρήση γλώσσας C.

Η λίστα των βασικών λειτουργιών του αποτελείται από:

- Απλή και φιλική λειτουργία ως προς το χρήστη
- Open-Source (Ανοιχτού Κώδικα)
- Υποστήριξη ενημερώσεων λογισμικού
- Υποστήριξη πολλών Addons για κάθε πιθανό συνδυασμό υλοποίησης δικτύων
- Υποστήριξη δοκιμών-πειραμάτων για ανάπτυξη OpenFlow εφαρμογών
- Υποστήριξη πολλών Developer πάνω στο ίδιο Project
- Υποστήριξη Regression Testing για την εξασφάλιση αξιοπιστίας

- Υποστήριξη Λειτουργιών Συμπλεγματικών Τοπολογιών χωρίς τη χρήση ενός φυσικού δικτύου

- Υποστήριξη CLI για κάθε είδος τοπολογίας και κάθε είδους ελέγχου DEBUGGING

- Υποστήριξη βασικών τοπολογιών με έτοιμη παραμετροποίηση
- Υποστήριξη Custom τοπολογιών καθώς και Custom παραμετροποίησή τους
- Έλεγχος επιδόσεων δικτύων
- Άμεσα προσβάσιμο χωρίς τη χρήση γνώσεων προγραμματισμού
- Αυξημένη ευχέρεια σε χρήστες γνώσεων προγραμματισμού καθώς παρέχει τη δυνατότητα μέσω Python Addon:

- Ανάγνωση , Συγγραφή και Εισαγωγή κώδικα μέσω Python
- Δημιουργίας Custom τοπολογιών
- Παραμετροποίηση Custom τοπολογιών

Το Mininet OS είναι διαθέσιμο στον παρακάτω ιστότοπο μαζί με τις οδηγίες εγκατάστασής του και η έκδοση που θα χρησιμοποιηθεί είναι η 2.2.2. [\[27\]](#)

#### 4.1.1 Στόχος Mininet OS

Το Mininet δημιουργήθηκε με σκοπό τη μείωση του κόστους υλοποίησης πειραμάτων και εκτέλεσης εκτεταμένων δοκιμών δικτύου. Ως εκ τούτου, το Mininet ανοίγει τους ορίζοντες στην αναπαραγωγή δοκιμών δικτύων και σε πειραματικά περιβάλλοντα τα οποία προσφέρουν δυνατότητα:

- Ελαχιστοποίησης του ερευνητικού κόστους
- Εξάλειψης του κινδύνου τεχνικής βλάβης σε δικτυακό εξοπλισμό
- Βελτίωσης διαχείρισης ανθρωπίνου δυναμικού και πόρων
- Έρευνας και ανάπτυξης πάνω σε υπάρχοντα συστήματα δικτύων
- Βελτίωσης των ήδη υπάρχοντων συστημάτων δικτύων
- Δημιουργίας νέων αποδοτικότερων συστημάτων δικτύων

- Διδασκαλίας και εκμάθησης δικτύων σε φοιτητές-μαθητές ιδρυματικών οργανισμών
- Εκπαίδευσης και βελτίωσης δυνατοτήτων εργαζομένων σε σχετικούς φορείς

### **4.1.2 Όρια λειτουργίας Mininet OS**

Από την επίσημη υποστήριξη του Mininet OS γίνονται γνωστά κάποια όρια στη λειτουργία του βάσει και προδιαγραφών συστήματος αλλά και προδιαγραφών λογισμικού. Επίσης το Mininet OS ως προσομοίωση μπορεί να τρέξει μέχρι 4096 Hosts. Ο αριθμός αυτός ενδέχεται να διαφέρει μεταξύ συστημάτων και αναλόγως των τοπολογιών δικτύων. Στις παρακάτω δοκιμές δοκιμάστηκαν τα όρια του και παρατηρήθηκε επίσης ότι το λογισμικό μπορεί να αξιοποιήσει μέχρι και περίπου 9 GB μνήμης RAM από τον υπολογιστή του χρήστη. Μόλις φτάσει αυτό το όριο δεν μπορεί να διαχειριστεί παραπάνω μνήμη RAM ακόμα και αν αυτή είναι διαθέσιμη και ελεύθερη από το σύστημα.

## **4.2 VirtualBox**

Το VirtualBox είναι το πιο δημοφιλές λογισμικό εκτέλεσης VM και δημιουργήθηκε από την εταιρεία ORACLE το 2007. Το βασικότερο πλεονέκτημά του είναι ότι υποστηρίζεται από όλα τα διαθέσιμα λειτουργικά συστήματα για όλους τους υπολογιστές. Παρέχει στον χρήστη τη δυνατότητα μέσα από τον υπολογιστή του να τρέξει εικονικά πολλαπλά λειτουργικά συστήματα ταυτόχρονα από μια λίστα που αποτελείται σχεδόν από το 100% όλων των λειτουργικών συστημάτων που υπάρχουν μέχρι και σήμερα και για αυτό βρίσκει μεγάλη απήχηση σε εκπαιδευτικούς και ερευνητικούς σκοπούς. Το VirtualBox είναι διαθέσιμο στον παρακάτω ιστότοπο μαζί με τις επεκτάσεις του και τις οδηγίες εγκατάστασής του καθώς και επίσημη υποστήριξη μέσω Community και η έκδοση που θα χρησιμοποιηθεί είναι η 6.1.16 [\[54\]](#).

## **4.3 X-Ming**

Το X-MING είναι ένα λογισμικό για Windows υπολογιστές το οποίο επιτρέπει το άνοιγμα πολλαπλών Linux ή MacOS Terminal καθώς και την επικοινωνία μεταξύ Windows και Linux για την αλληλεπίδραση των Terminal μέσω SSH και δημιουργήθηκε το 2004. Αυτό σημαίνει ότι πρακτικά ο χρήστης μπορεί να πληκτρολογήσει μια εντολή στον υπολογιστή του

που χρησιμοποιεί Windows και να επικοινωνήσει με έναν υπολογιστή εικονικό ή μη που χρησιμοποιεί Linux και βρίσκεται συνδεδεμένος στο δίκτυό του ή και αντίστροφα. Αυτό έχει ως αποτέλεσμα να παρακάμπτονται τυχόν δυσκολίες στην χρήση και επικοινωνία υπολογιστών που χρησιμοποιούν Linux και Windows και η χρήση του είναι καθαρά εκπαιδευτική και ερευνητική. Το X-MING είναι διαθέσιμο στον παρακάτω ιστότοπο μαζί με τις οδηγίες εγκατάστασής του και η έκδοση που θα χρησιμοποιηθεί είναι η 6.9.0.31 [\[58\]](#).

#### 4.4 WinSCP

Το WinSCP είναι ένα λογισμικό ανοιχτού κώδικα για Windows υπολογιστές το οποίο παρέχει τις εξής δυνατότητες client:

- SFTP
- FTP
- WebDAV
- S3
- SCP

Ο κύριος σκοπός του WinSCP είναι η μεταφορά αρχείων σε απομακρυσμένες συσκευές είτε εικονικές είτε φυσικές. Επίσης διαθέτει όλες τις βιβλιοθήκες του προγράμματος PuTTY και πλήρη συμβατότητα με αυτό. Το WinSCP είναι διαθέσιμο στον παρακάτω ιστότοπο μαζί με τις οδηγίες εγκατάστασής του και η έκδοση που θα χρησιμοποιηθεί είναι η 5.17.10 [\[59\]](#).

#### 4.5 PuTTY

Το PuTTY δρομολογείται στο 1999 δείχνοντας ότι ξεκίνησε από τον πρώτο καιρό που άρχισαν οι υπολογιστές να χρησιμοποιούνται και σε οικιακό επίπεδο πέρα από το επαγγελματικό και έρχεται να συμπληρώσει το κενό που αφήνει το X-MING καθώς είναι ένα λογισμικό που επιτρέπει μέσω ports την μεταφορά αρχείων μεταξύ υπολογιστών που χρησιμοποιούν επίσημα Windows, Linux και macOS ανήκουν στο ίδιο δίκτυο και αυξάνει την πρακτικότητα τους. Επίσης προσφέρει μέσω της παράλληλης λειτουργίας με το X-MING τη δυνατότητα απεικόνισης και αλληλεπίδρασης των επιπέδων Terminal μεταξύ Windows, Linux

και macOS. Ιδιαίτερη μνεία χρήζουν οι ανεπίσημες αναφορές για την υποστήριξη και λογισμικών κινητών τηλεφώνων(Symbian,Windows Mobile). Η μεταφορά αρχείων γίνεται μέσω κρυπτογράφησης και υποστηρίζει πρωτόκολλα όπως Raw Socket,Telnet,rlogin,SCP και SSH. Το PuTTY είναι διαθέσιμο στον παρακάτω ιστότοπο μαζί με τις οδηγίες εγκατάστασής του και η έκδοση που θα χρησιμοποιηθεί είναι η 0.74 [\[56\]](#).

## 4.6 GEPHI

Το GEPHI δημιουργήθηκε το 2008 και είναι το πιο δημοφιλές λογισμικό γραφικής απεικόνισης το οποίο δέχεται δεδομένα από άλλες εφαρμογές ή αρχεία και τα απεικονίζει γραφικά με όλους τους δυνατούς επιστημονικούς τρόπους και περιλαμβάνει δυνατότητες όπως.

- Υποστήριξη πολλών τύπων δεδομένων
- Ανάλυση δεδομένων δικτύων σε πραγματικό χρόνο
- Ανάλυση συνδέσμων και υποδομών
- Ανάλυση δεδομένων κοινωνικών δικτύων
- Ανάλυση δεδομένων βιολογικών δικτύων
- Δημιουργία επιστημονικών αποτυπωμάτων
- Πλήρη υποστήριξη μετρικών
- Εξαγωγή αποτελεσμάτων συμβατά με πολλαπλούς τύπους εφαρμογών και αρχείων
- Απλό περιβάλλον εργασίας το οποίο δεν απαιτεί εξειδικευμένες γνώσεις ως προς τη χρήση του

Το GEPHI είναι διαθέσιμο στον παρακάτω ιστότοπο μαζί με τις οδηγίες εγκατάστασής του και η έκδοση που θα χρησιμοποιηθεί είναι η 0.9.2 [\[57\]](#).

## 4.7 Τεχνικά χαρακτηριστικά

Στην υλοποίηση της παρούσας έρευνας χρησιμοποιήθηκε ένας φορητός υπολογιστής υψηλών επιδόσεων με τα εξής χαρακτηριστικά :

CPU	RAM	GPU	STORAGE	OS
AMD RYZEN 5 3550H 4core 8threads 3.70 Ghz	2X16GB(32GB) DDR4 2666mhz	NVIDIA GTX 1650 4GB GDDR5	PCI EXPRESS Gen 3.0 SSD 512GB	Windows 10 Home 64-bit

*Πίνακας 4.1: Πίνακας τεχνικών προδιαγραφών Η/Υ που εκτελέστηκαν οι εξομοιώσεις της παρούσας πτυχιακής.*

## 4.8 Εγκατάσταση λογισμικού

Σε αυτή την ενότητα θα παρουσιαστεί ο τρόπος εγκατάστασης των προηγούμενων εφαρμογών υπολογιστή που χρησιμοποιήθηκε στην υλοποίηση της έρευνας αυτής. Επειδή ο υπολογιστής έχει λογισμικό Windows θα εγκατασταθεί πρώτα η εφαρμογή εκτέλεσης VM VirtualBox και μέσω αυτού θα εγκατασταθεί το Mininet OS ως VM .Αυτό έχει ως αποτέλεσμα του να έχουμε τον έλεγχο του εικονικού συστήματος μέσω του υπολογιστή και παρέχει τη δυνατότητα να μελετήσουμε ,να ελέγξουμε και να πάρουμε αποτελέσματα χωρίς να αλλάξει λειτουργικά κάτι στον υπολογιστή των χρηστών. Οι υπόλοιπες εφαρμογές εγκαθίστανται ανεξάρτητα στον υπολογιστή.

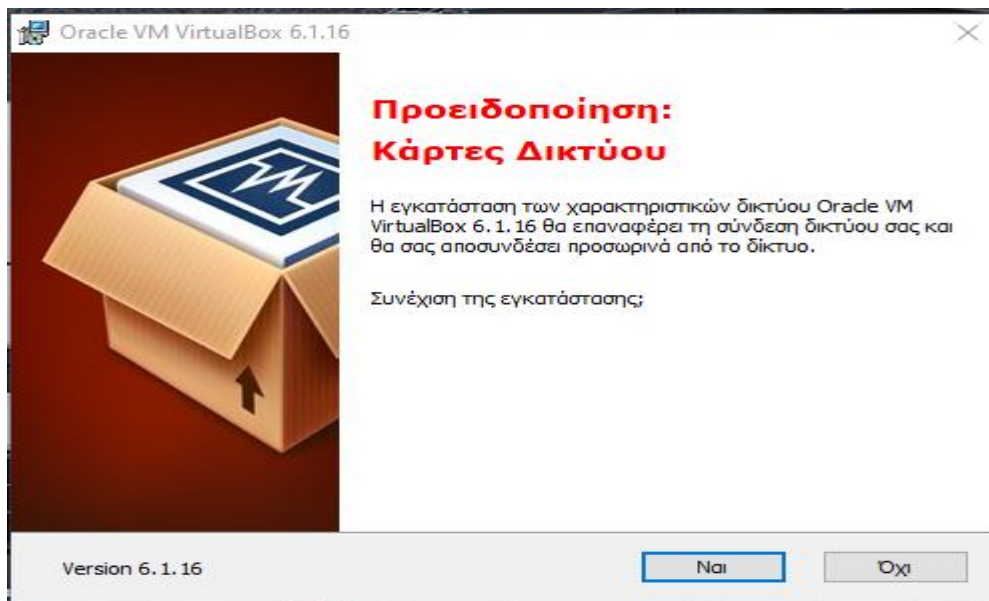
### 4.8.1 Εγκατάσταση VirtualBox

Κατεβάζουμε το VirtualBox 6.1.16 από τον επίσημο ιστότοπο στην έκδοση για Windows [\[54\]](#).



Εικόνα 4.1: Κατέβασμα εφαρμογής VirtualBox για Η/Υ με λειτουργικό σύστημα Windows.

Ακολουθούμε τις προεπιλεγμένες ρυθμίσεις για την εγκατάσταση και δεχόμαστε την προσωρινή αποσύνδεση από το δίκτυό μας διότι έτσι το VirtualBox δημιουργεί την εικονική κάρτα δικτύου που θα χρησιμοποιεί παράλληλα με τη φυσική κάρτα δικτύου του υπολογιστή.

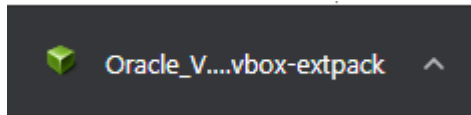


Εικόνα 4.2: Εγκατάσταση VirtualBox.

Επίσης προχωρούμε στην εγκατάσταση του πρόσθετου πακέτου VirtualBox EXTENSION PACK προκειμένου να υπάρχει πλήρης υποστήριξη τεχνικών προδιαγραφών από την εφαρμογή. Ακολουθώντας τις οδηγίες εγκατάστασης.

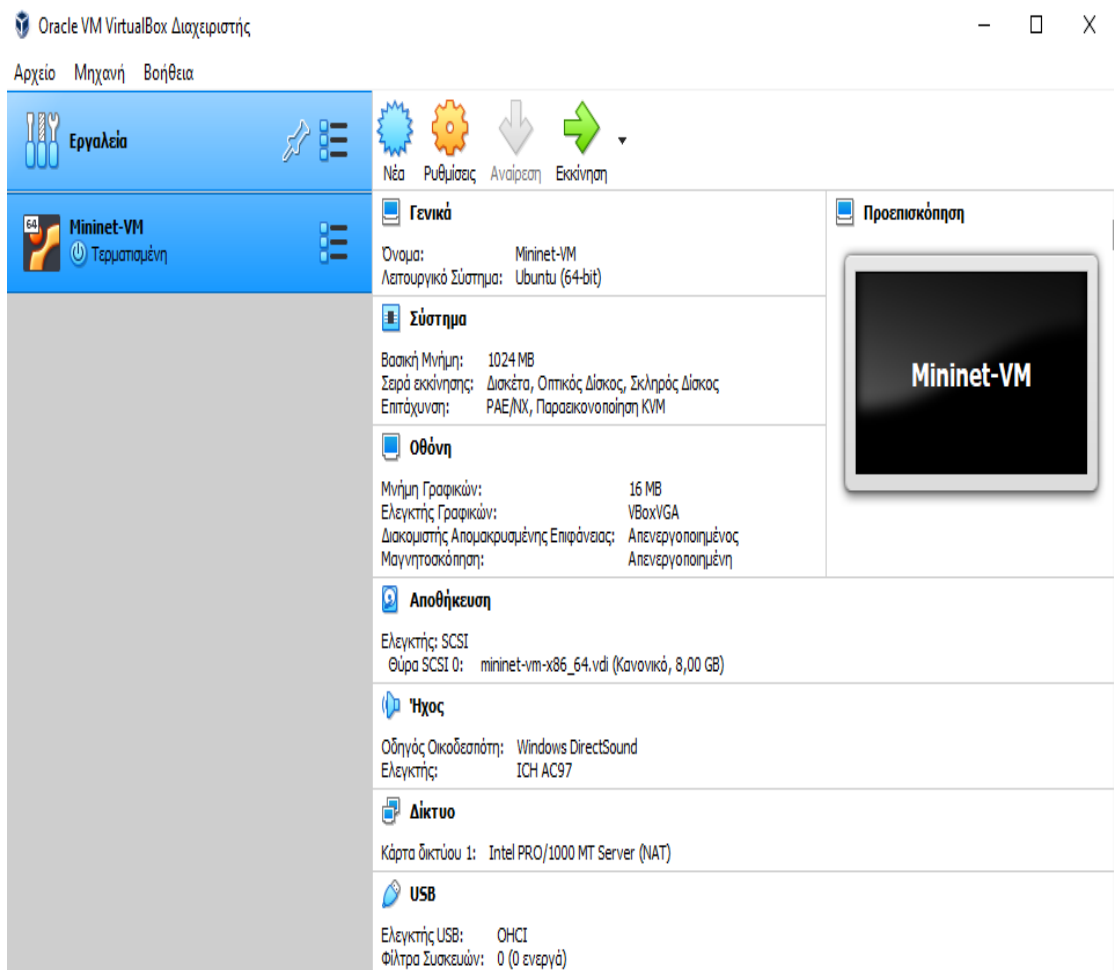
## VirtualBox 6.1.16 Oracle VM VirtualBox Extension Pack

- [⇒ All supported platforms](#)



Εικόνα 4.3: Αρχείο VirtualBox Extension Pack

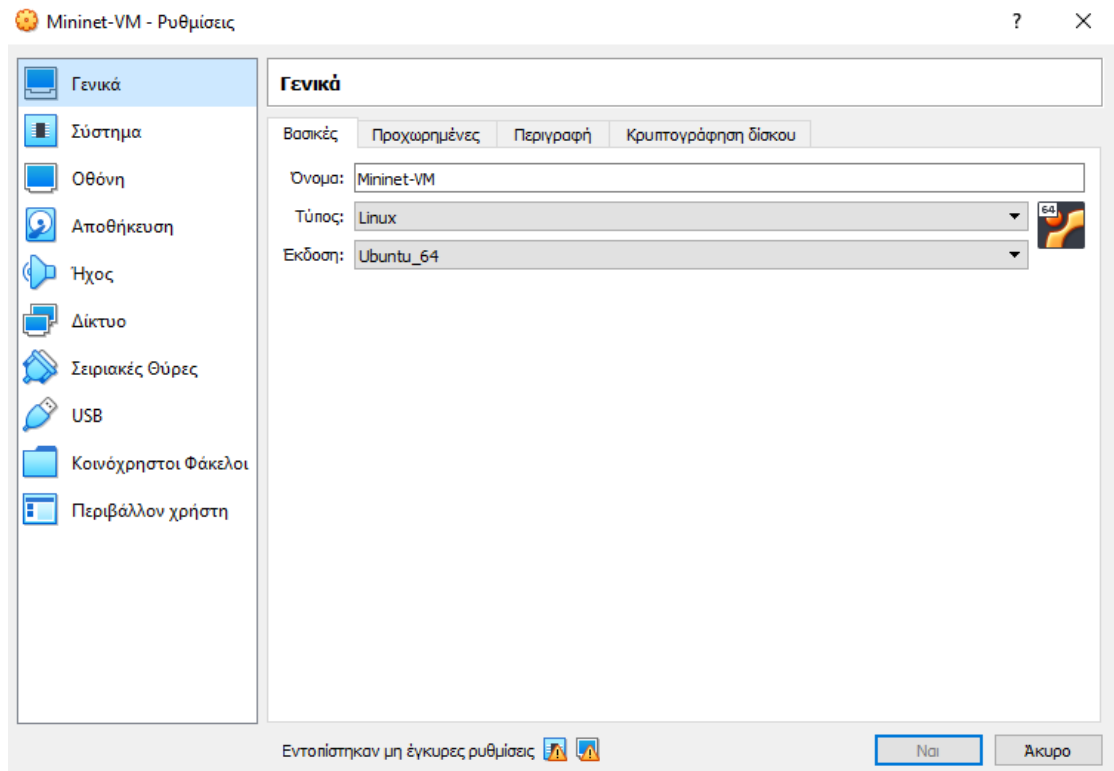
### 4.8.1.1 Περιβάλλον εργασίας VirtualBox



Εικόνα 4.4: Βασικό περιβάλλον εργασίας VirtualBox.



Στα αριστερά έχει την μπάρα με τα εγκατεστημένα εικονικά συστήματα που στην περίπτωση μας είναι το Mininet. Στο κέντρο βλέπουμε τους πόρους που απορροφά το εικονικό σύστημα από το φυσικό σύστημα. Μπορούμε μέσω των ρυθμίσεων να παραμετροποιήσουμε όλες τις προδιαγραφές του συστήματος προκειμένου να το κάνουμε πιο αποδοτικό και πιο συμβατό με τις απαιτήσεις μας. Ενδεικτικά μερικές ρυθμίσεις.



Εικόνα 4.5:Γενικές ρυθμίσεις VirtualBox.

#### 4.8.1.2 CPU Virtualization

Κάθε υπολογιστής προτού τρέξει ένα εικονικό σύστημα πρέπει μέσα από το BIOS της MOTHERBOARD του να ενεργοποιηθεί η ειδική επιλογή VIRTUALIZATION(VT) έτσι ώστε ο επεξεργαστής να μπορέσει να υλοποιήσει προσομοιώσεις με επιτυχία και να προσαρμοστεί η λειτουργία του σε αυτές. Η λειτουργία CPU VIRTUALIZATION επιτρέπει σε έναν CPU με X αριθμό πυρήνων να λειτουργεί σαν να ήταν X ο αριθμός των CPU. Αυτό έχει ως αποτέλεσμα την καλύτερη διαχείριση των πόρων ενός CPU από είτε ένα είτε πολλαπλά εικονικά συστήματα. Η διαδικασία αυτή διαφέρει ελάχιστα(κυρίως στον τρόπο ονομασίας της λειτουργίας) αναλόγως με τον κατασκευαστή του CPU του κάθε υπολογιστή(AMD,INTEL). Αρχικά με την εκκίνηση του συστήματος πατάμε το αντίστοιχο

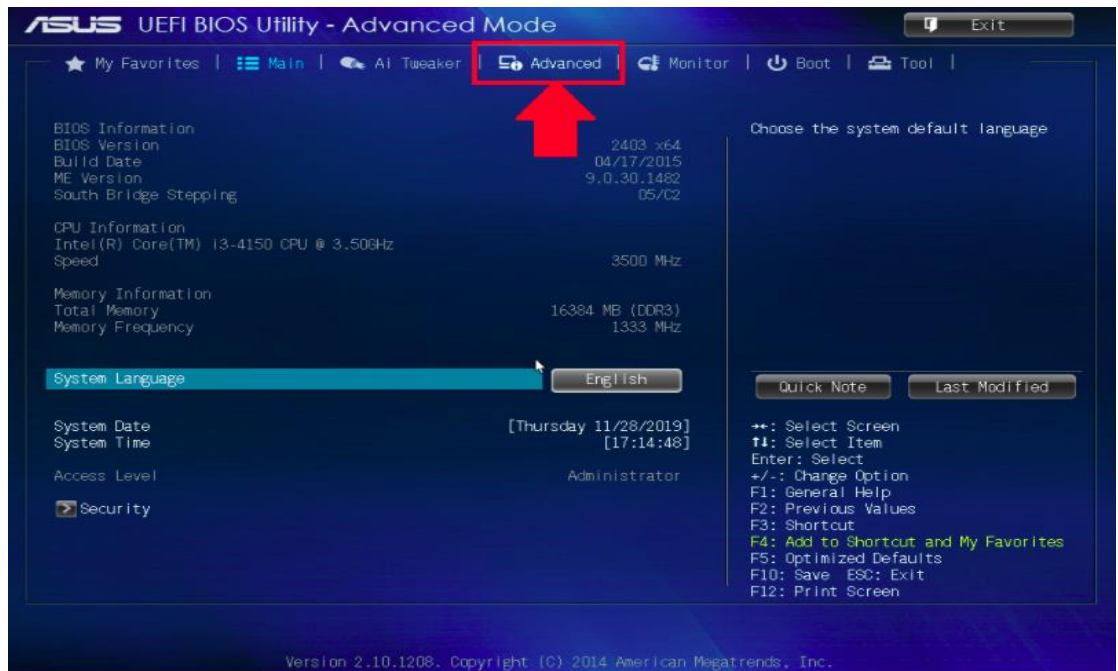
κουμπί(αναλόγως τον κατασκευαστή της MOTHERBOARD) και αποκτούμε πρόσβαση στο BIOS.

Ενδεικτικά σε υπολογιστές με επεξεργαστή INTEL επιλέγουμε την επιλογή «Advanced Mode (F7)».



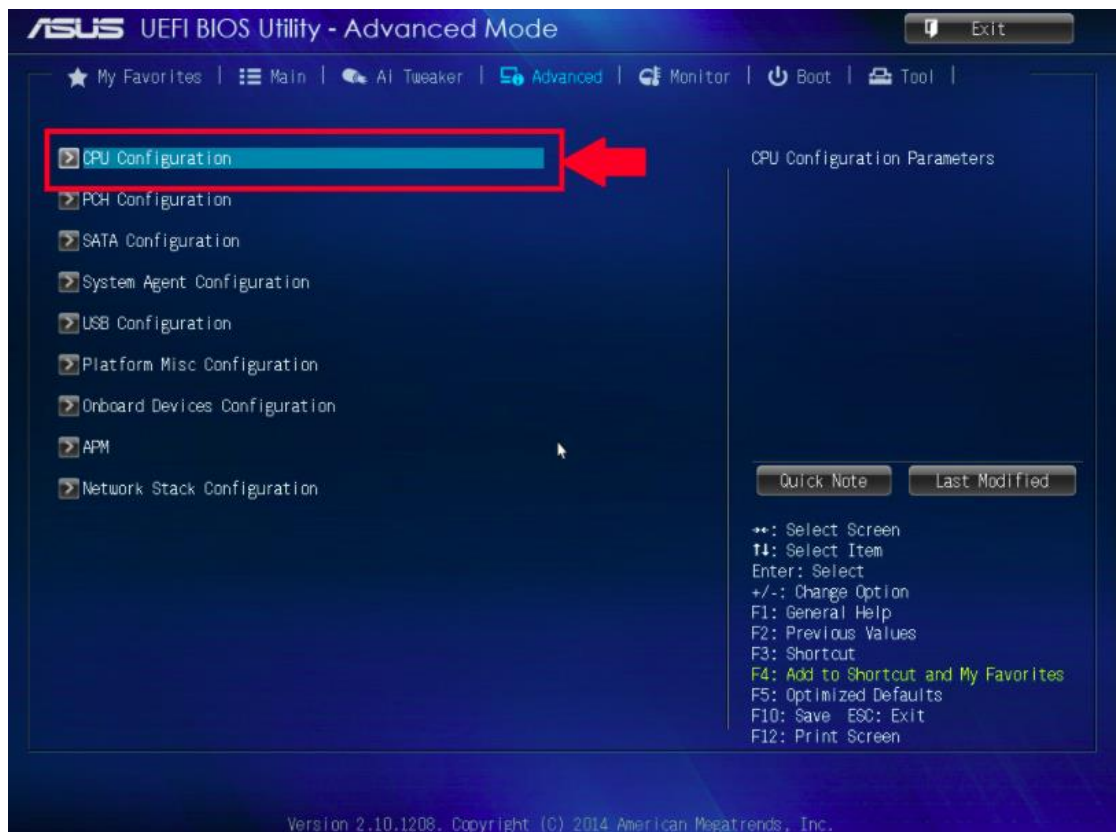
Εικόνα 4.6:Βασικό μενού BIOS μητρικής με επεξεργαστή Intel [18].

Στη συνέχεια προχωράμε επιλέγοντας «Advanced».



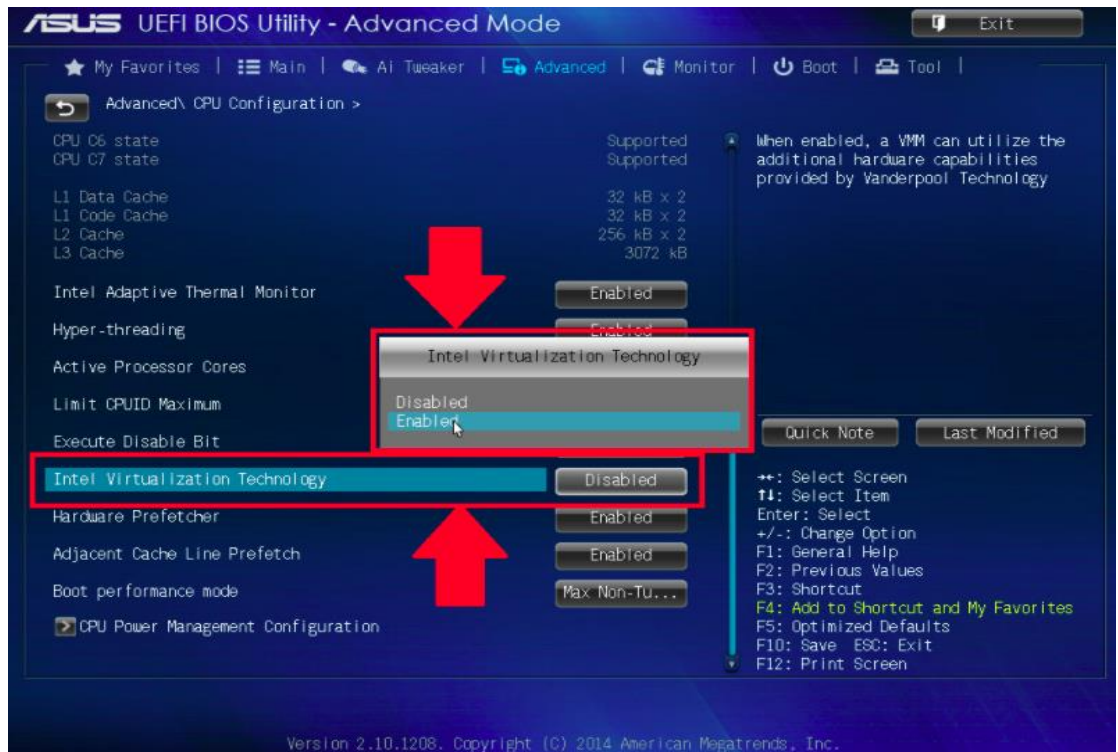
Εικόνα 4.7: Advanced mode στο BIOS μητρικής με επεξεργαστή INTEL [18].

Επιλέγουμε «CPU CONFIGURATION» για να ανοίξουν οι ρυθμίσεις CPU.



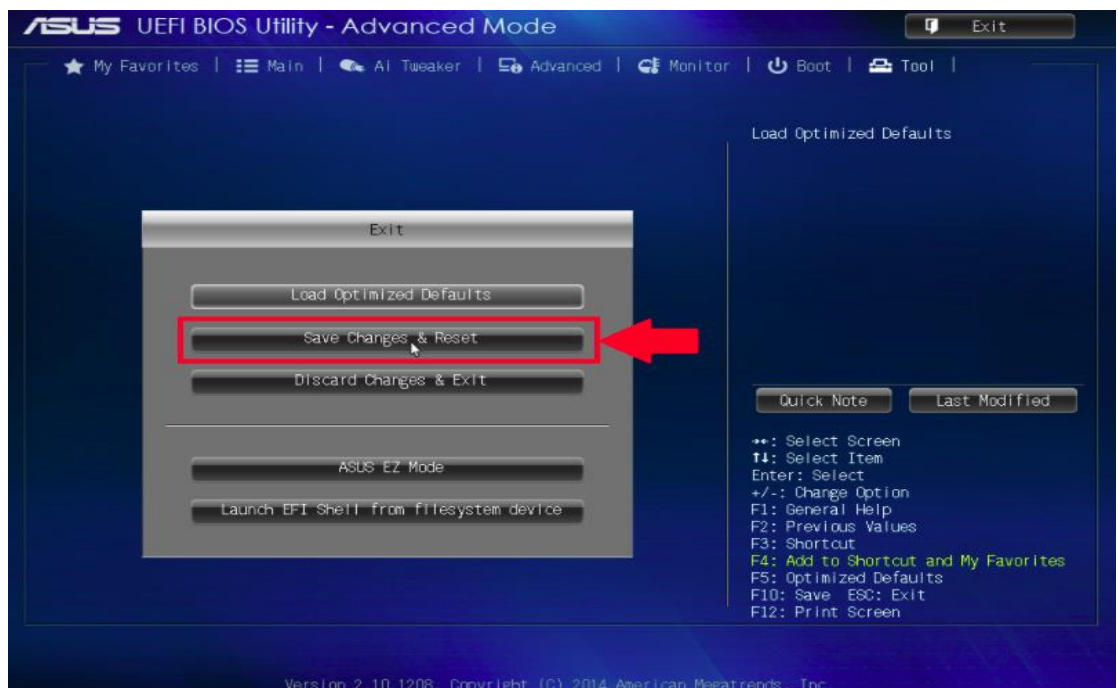
Εικόνα 4.8: Ρυθμίσεις CPU στο BIOS μητρικής με επεξεργαστή INTEL [18].

Ενεργοποιούμε τη λειτουργία «Intel Virtualization Technology».



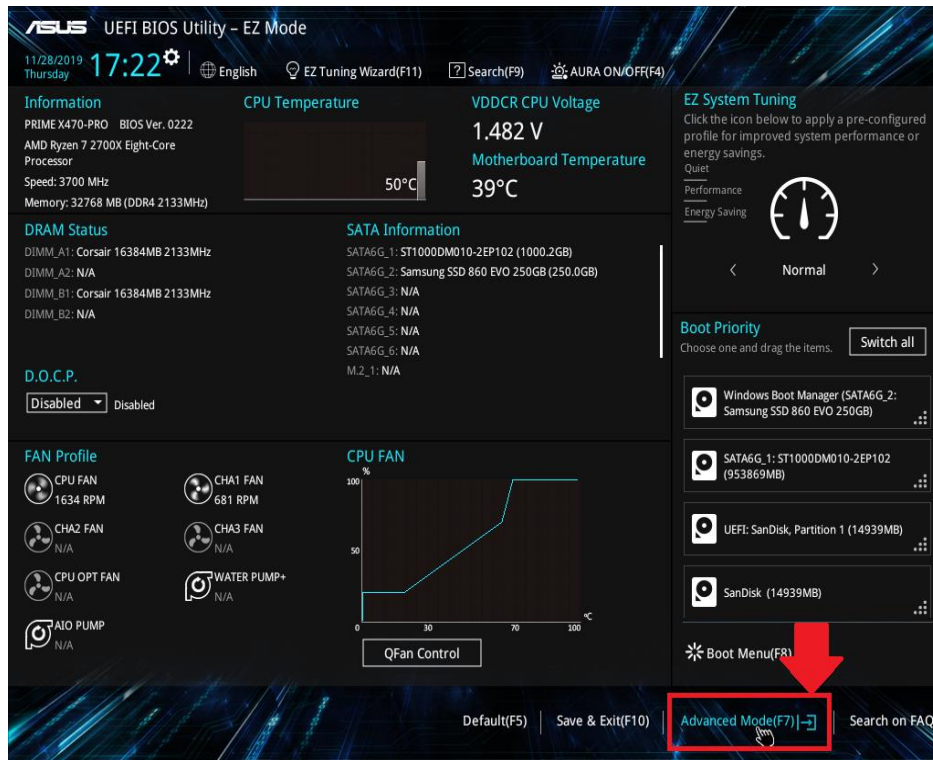
Εικόνα 4.9:Ενεργοποίηση λειτουργίας VIRTUALIZATION στο BIOS μητρικής με επεξεργαστή INTEL [18].

Τέλος αποθηκεύουμε τις αλλαγές που κάναμε στο BIOS και βγαίνοντας από αυτό ο υπολογιστής μας επανεκκινείται αυτόματα.



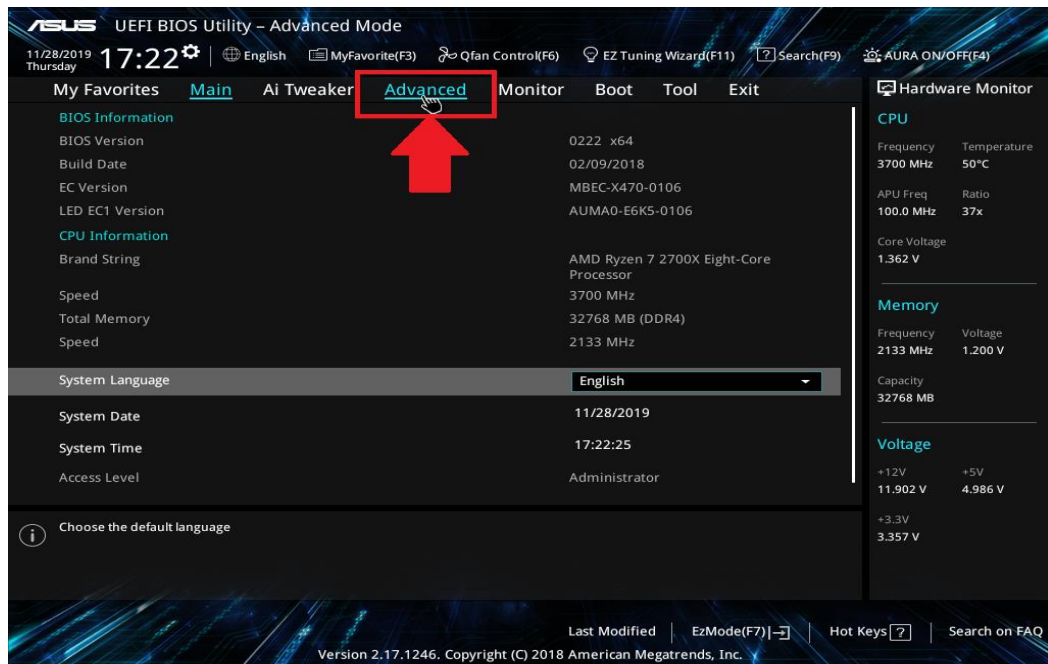
Εικόνα 4.10:Αποθήκευση αλλαγών στο BIOS μητρικής με επεξεργαστή INTEL [18].

Σε υπολογιστές με επεξεργαστή AMD επιλέγουμε επίσης την επιλογή «Advanced Mode (F7)».



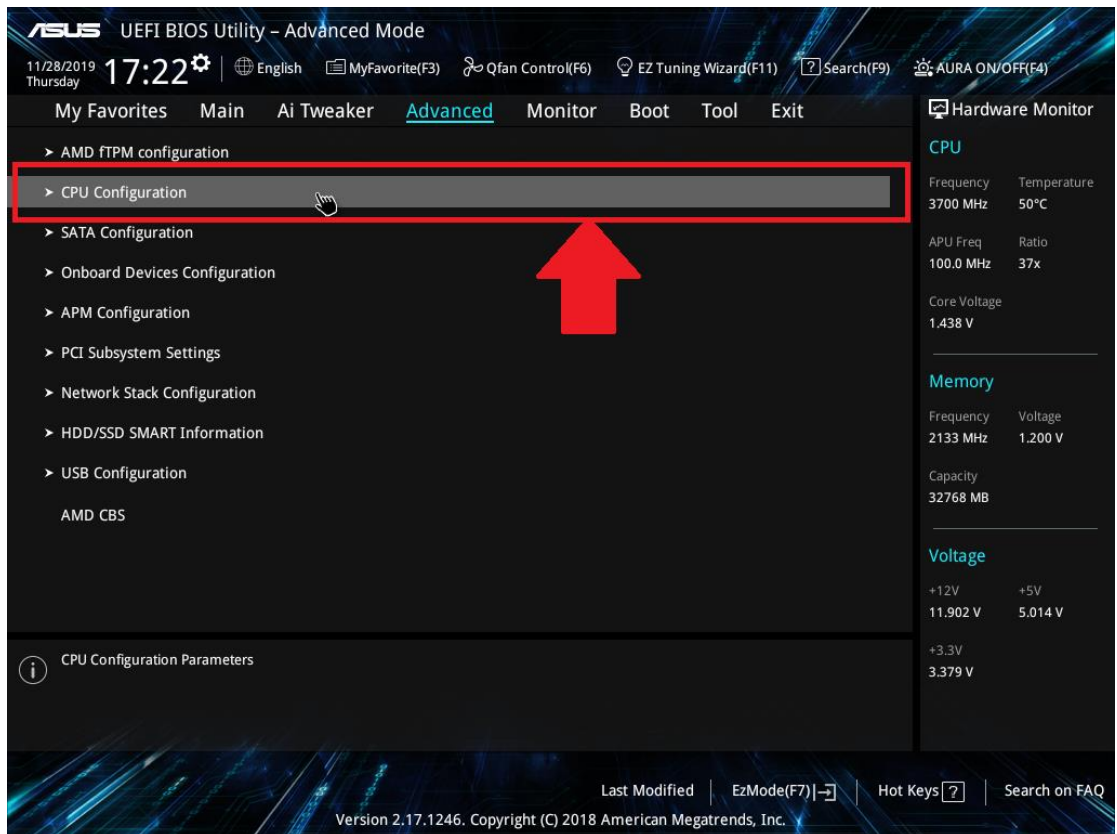
Εικόνα 4.11:Επιλογή Advanced Mode στο BIOS μητρικής με επεξεργαστή AMD [18].

Στη συνέχεια προχωράμε επιλέγοντας «Advanced».



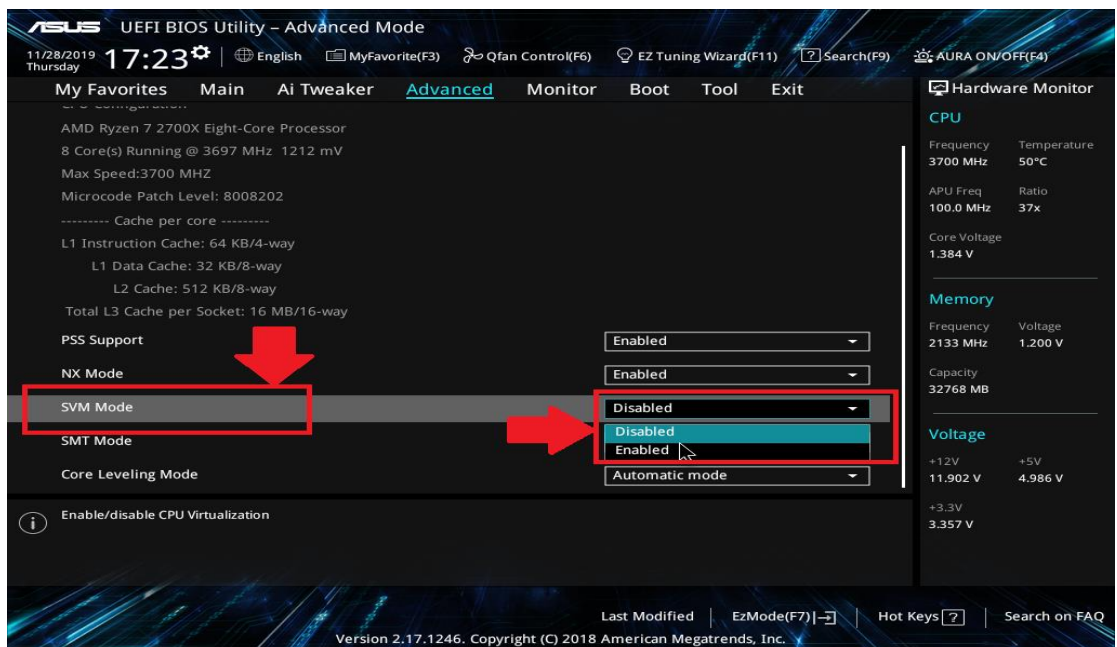
Εικόνα 4.12:Επιλογή Advanced στο BIOS μητρικής με επεξεργαστή AMD [18].

Επιλέγουμε «CPU CONFIGURATION» για να ανοίξουν οι ρυθμίσεις CPU.



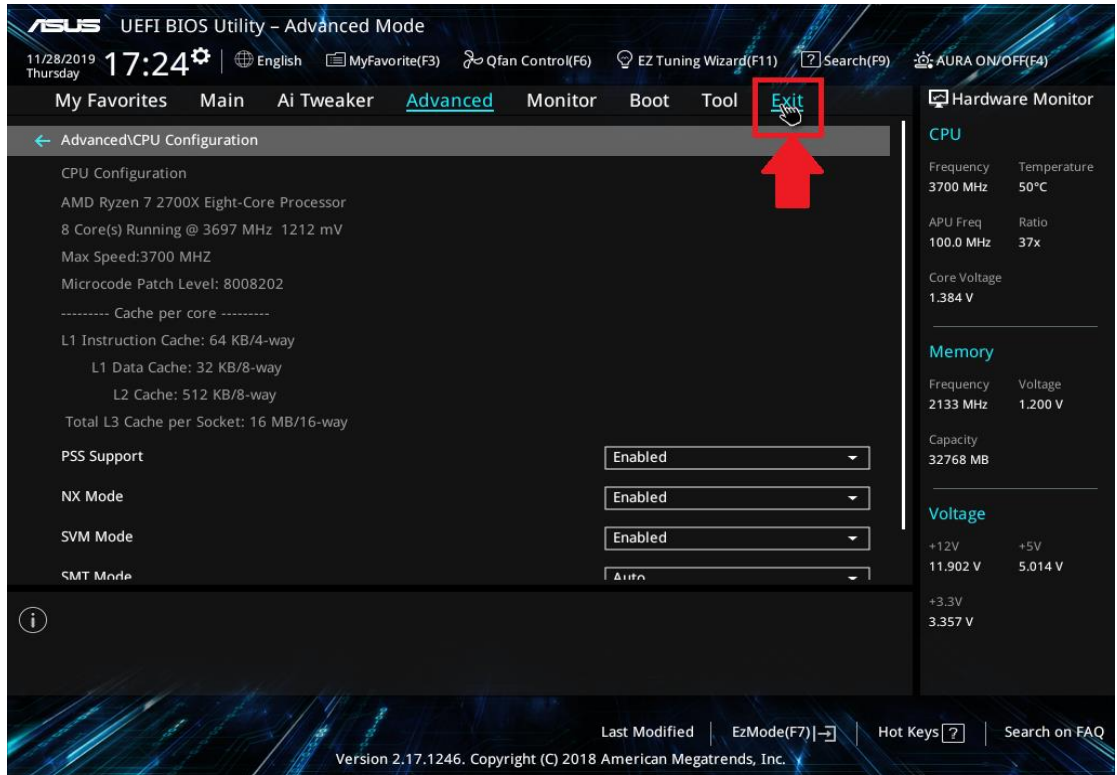
Εικόνα 4.13: Επιλογή CPU Configuration στο BIOS μητρικής με επεξεργαστή AMD [18].

Επιλέγουμε τη λειτουργία «SVM» και την ενεργοποιούμε.



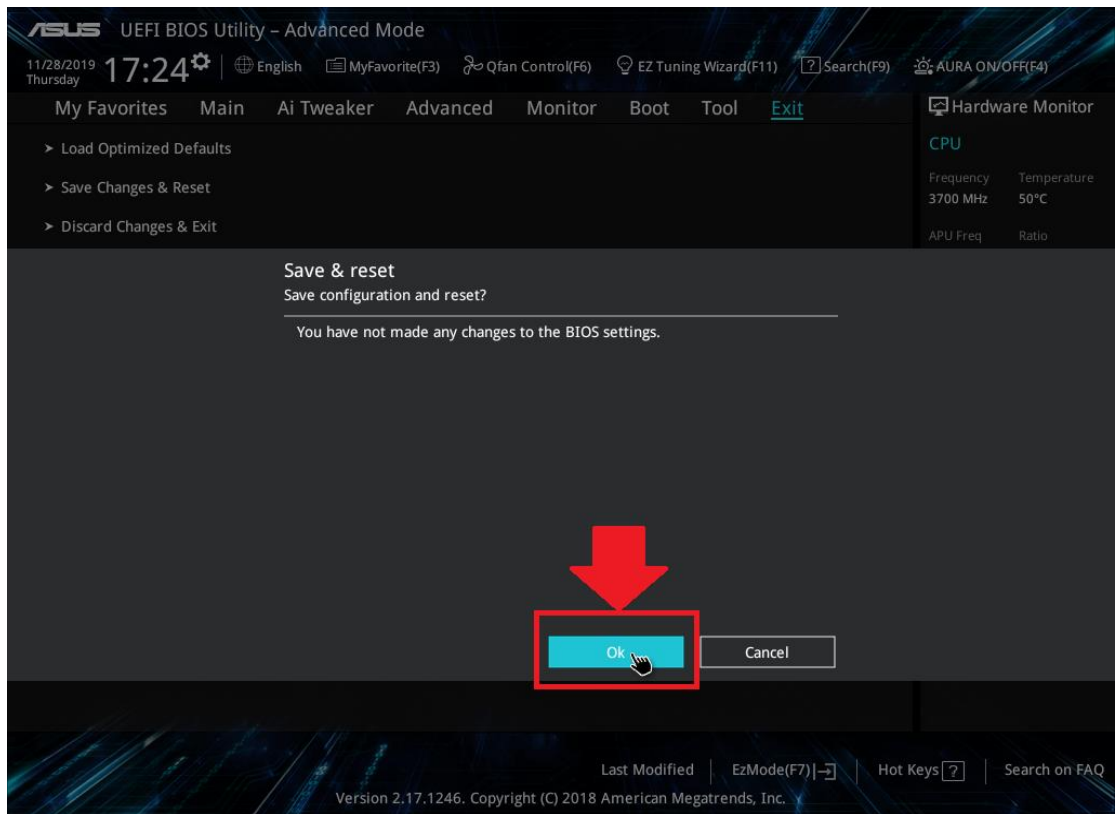
Εικόνα 4.14: Ενεργοποίηση λειτουργίας SVM στο BIOS μητρικής με επεξεργαστή AMD [18].

Επιλέγουμε την επιλογή «EXIT» για να βγούμε από το BIOS.



Εικόνα 4.15: Έξοδος από το BIOS μητρικής με επεξεργαστή AMD [18].


Τέλος αποθηκεύουμε τις αλλαγές που κάναμε στο BIOS και βγαίνοντας από αυτό ο υπολογιστής μας επανεκκινείται αυτόματα.



Εικόνα 4.16: Αποθήκευση αλλαγών στο BIOS μητρικής με επεξεργαστή AMD [18].

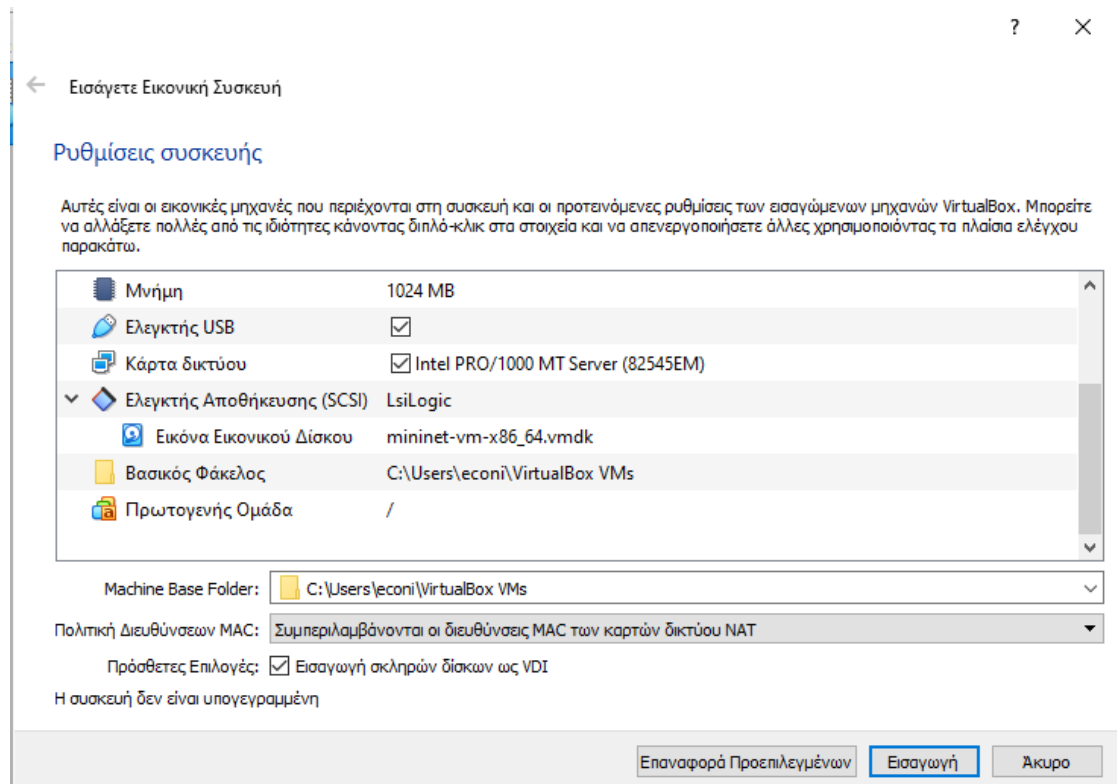
## 4.8.2 Εγκατάσταση Mininet

Κατεβάζουμε το Mininet 2.2.2 από τον ιστότοπό του [27]. Προσέχουμε το αρχείο να είναι στη μορφή OVF για να υποστηρίζεται ως VM από το VirtualBox και να είναι ανάλογο με τις προδιαγραφές του υπολογιστή μας, στο παρόν σύστημα λόγω του ότι χρησιμοποιεί επεξεργαστή AMD χρησιμοποιούμε την αντίστοιχη έκδοση.

 mininet-2.2.2-170321-ubuntu-14.04.4-ser...	21/3/2017 11:29 μμ	Open Virtualizatio...	4 KB
 mininet-vm-x86_64	21/3/2017 11:29 μμ	Virtual Machine Di...	1,999.872 KB

Εικόνα 4.17:Εύρεση αρχείου Mininet μέσω VirtualBox.

Επιλέγουμε το αρχείο και εκτελείται αυτόματα μέσω VirtualBox μόλις το εκτελέσουμε και ακολουθούμε τις προεπιλεγμένες ρυθμίσεις εγκατάστασης του λογισμικού όπως φαίνονται και από το πρόγραμμα.



Εικόνα 4.18:Ρυθμίσεις εγκατάστασης Mininet μέσω εικονικής μηχανής της VirtualBox.

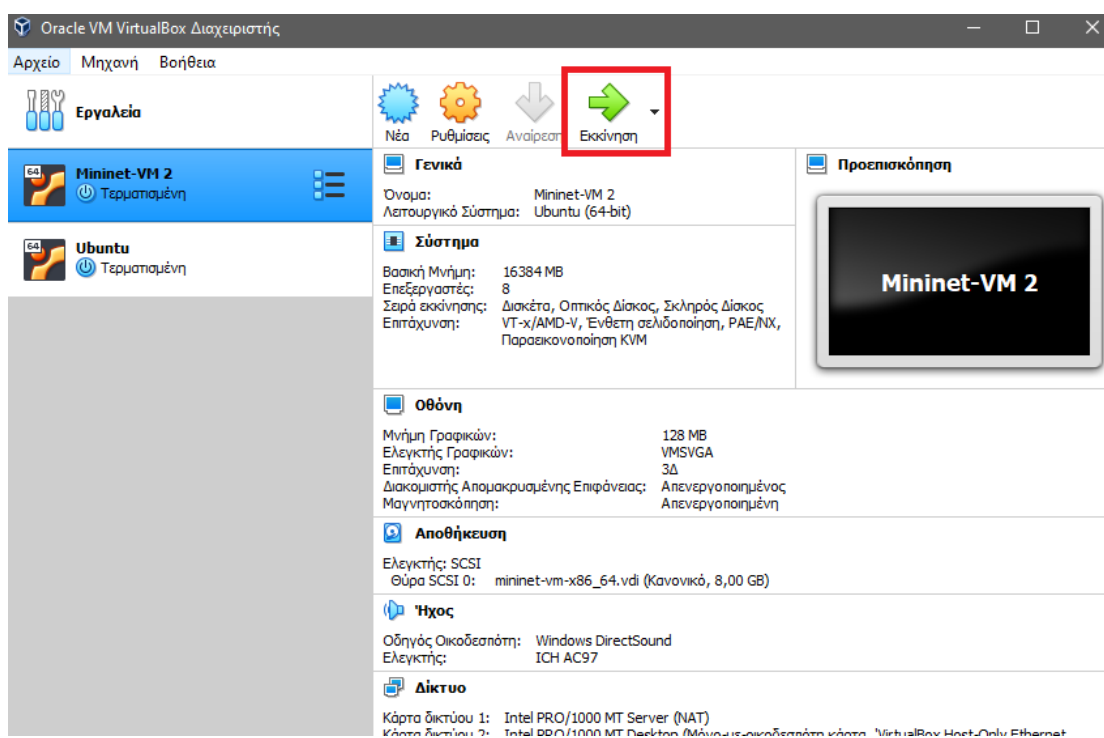
Χρειάζεται προσοχή έτσι ώστε να δημιουργηθεί ένας εικονικός σκληρός δίσκος επαρκούς χωρητικότητας μέσω στον φυσικό σκληρό δίσκο έτσι ώστε να πάρει αυτό το κομμάτι το ρόλο του σκληρού δίσκου στο VM.Η προεπιλογή των 8GB είναι αρκετή για ότι αφορά τις υλοποιήσεις SDN καθώς ακόμα και οι πιο περίπλοκες υλοποιήσεις φτάνουν να καταλαμβάνουν μέχρι και ελάχιστα MB.Επιπλέον στις επιλογές φαίνονται και οι βασικές



παράμετροι του εικονικού συστήματος τις οποίες βέβαια μπορούμε αργότερα να επεξεργαστούμε.

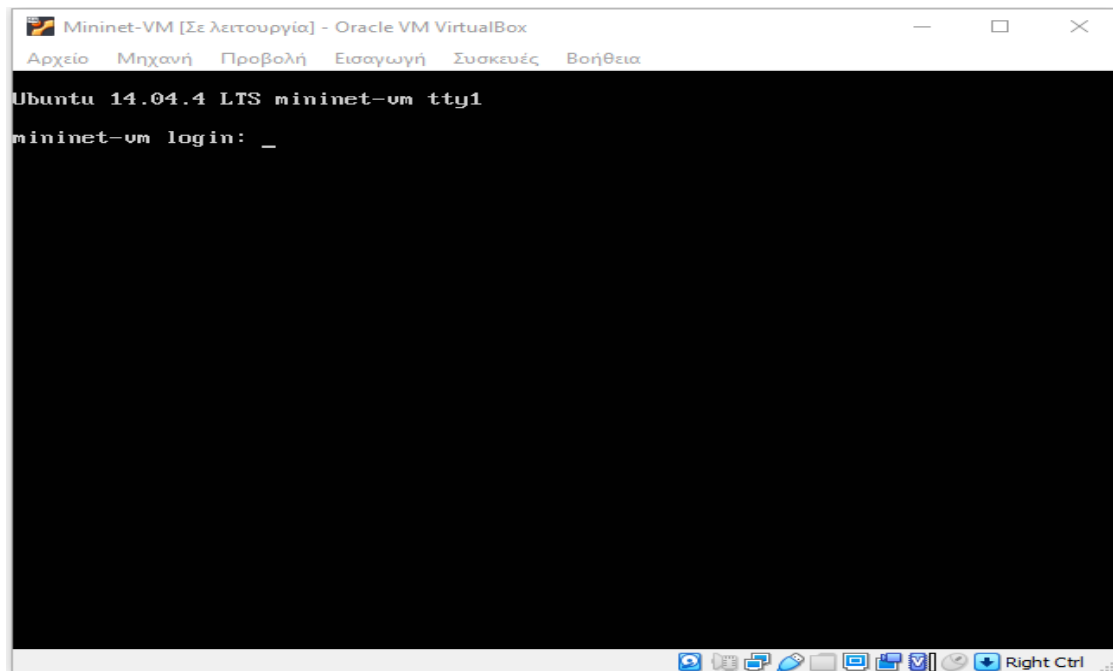
#### 4.8.2.1 Περιβάλλον εργασίας Mininet OS

Πατώντας το κουμπί εκκίνηση πραγματοποιείται αυτό ακριβώς που θα πραγματοποιούνταν και σε έναν φυσικό υπολογιστή πατώντας το power button, εκκινεί το σύστημα ανοίγοντας μια εικονική οθόνη.



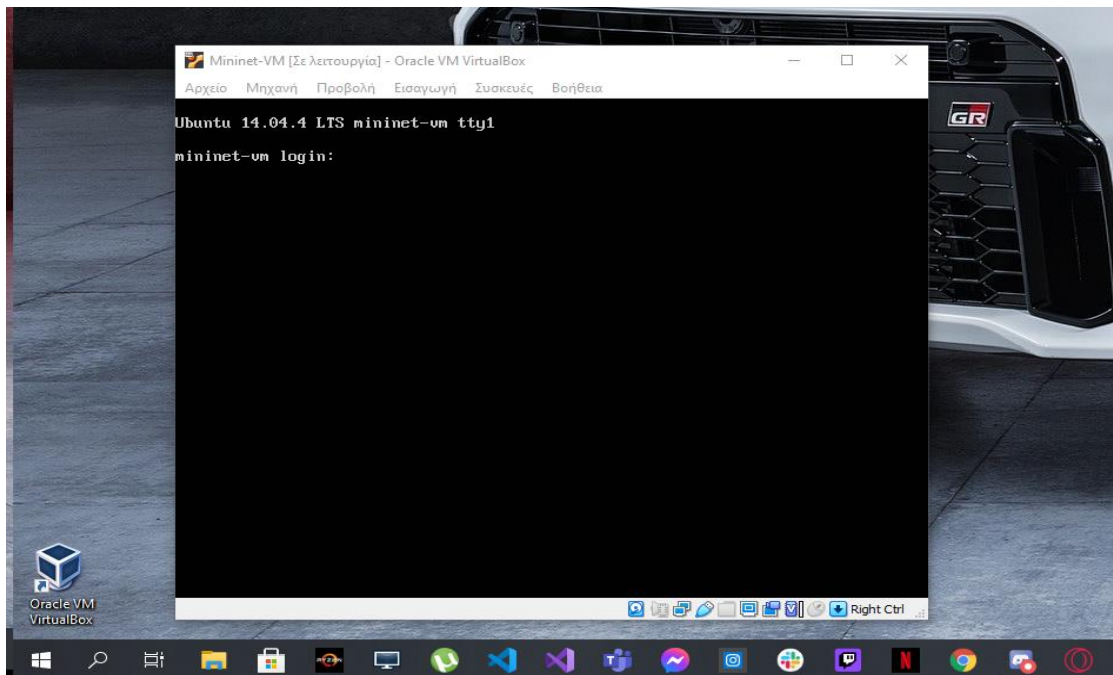
Εικόνα 4.19: Εκκίνηση εικονικής μηχανής Mininet μέσω VirtualBox.

Όπως φαίνεται και στην παρακάτω εικόνα το σύστημα λειτουργεί αυτούσιο σαν ξεχωριστός υπολογιστής.



Εικόνα 4.20:Λειτουργία εικονικού συστήματος.

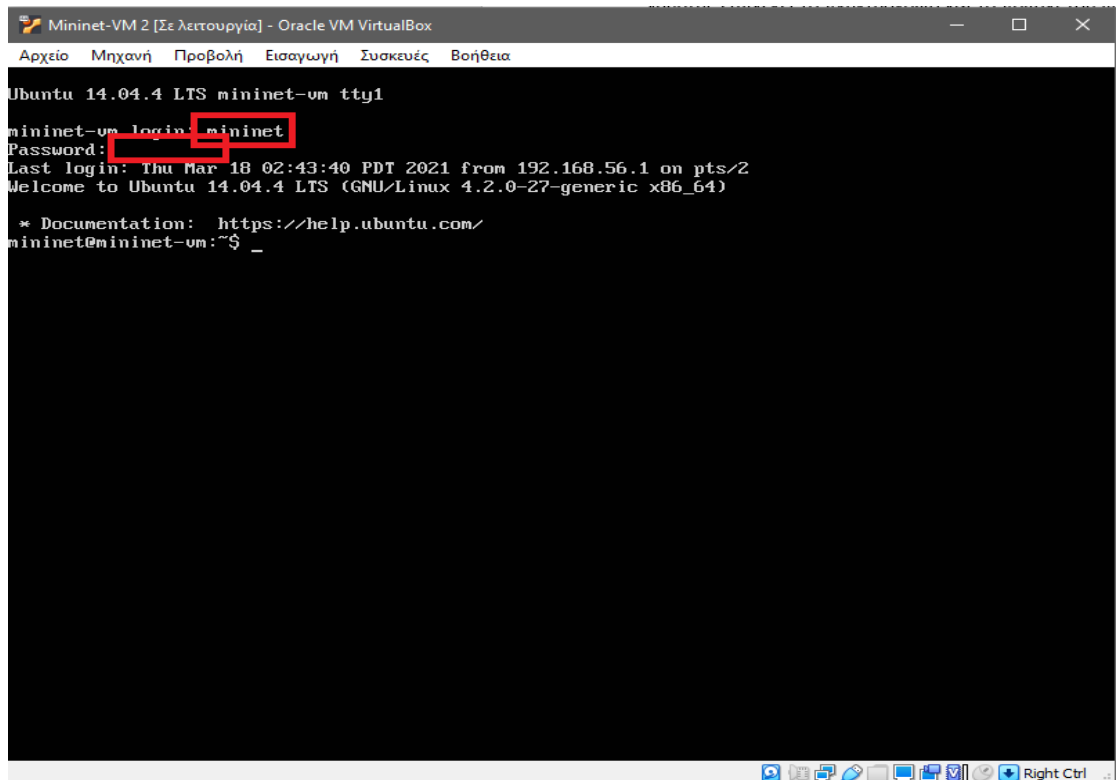
Στην παρακάτω εικόνα φαίνεται η λειτουργία του εικονικού συστήματος μέσω του φυσικού συστήματος με πλήρη αντιδραστικότητα μέσω του συστήματος του κάθε χρήστη.



Εικόνα 4.21:Εμφάνιση εικονικού συστήματος με αλληλεπίδραση φυσικού συστήματος.

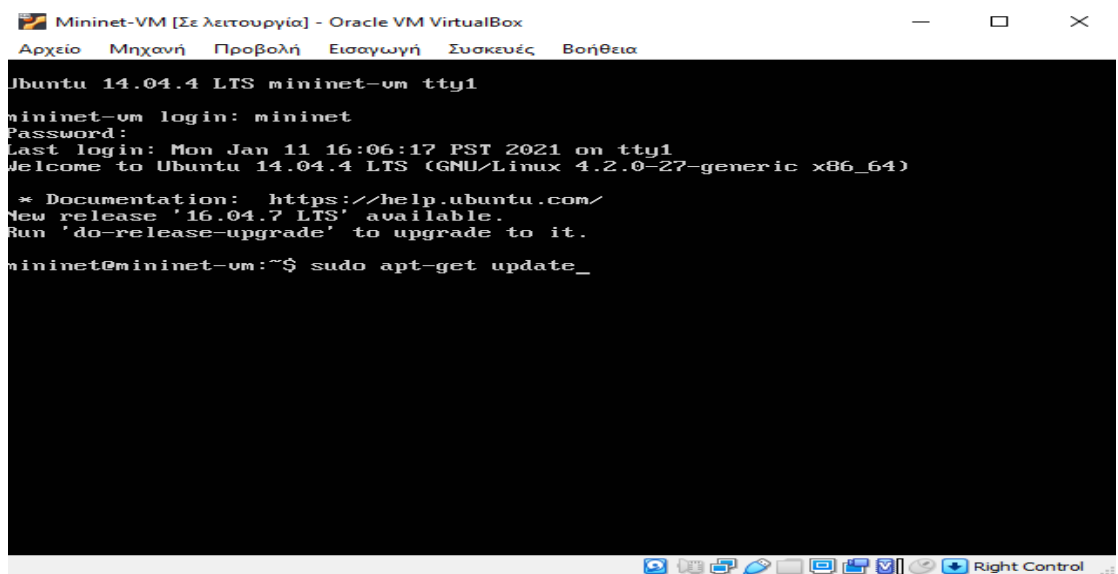
Παρατηρούμε ότι όταν το VirtualBox τρέχει ένα VM έχει ως πλήκτρο ενσωμάτωσης και η το Right Control όπως φαίνεται κάτω δεξιά. Με τη χρήση του άνωθεν πλήκτρου ο χρήστης εμπλέκει το πληκτρολόγιο και το ποντίκι του φυσικού υπολογιστή του στο VM και αντίστροφα. Η σύνδεση στο Mininet OS πραγματοποιείται από τον χρήστη εισάγοντας ως

USERNAME(όνομα χρήστη) και PASSWORD(κωδικό πρόσβασης) τη λέξη:”mininet” και έτσι ο χρήστης αποκτά πρόσβαση.

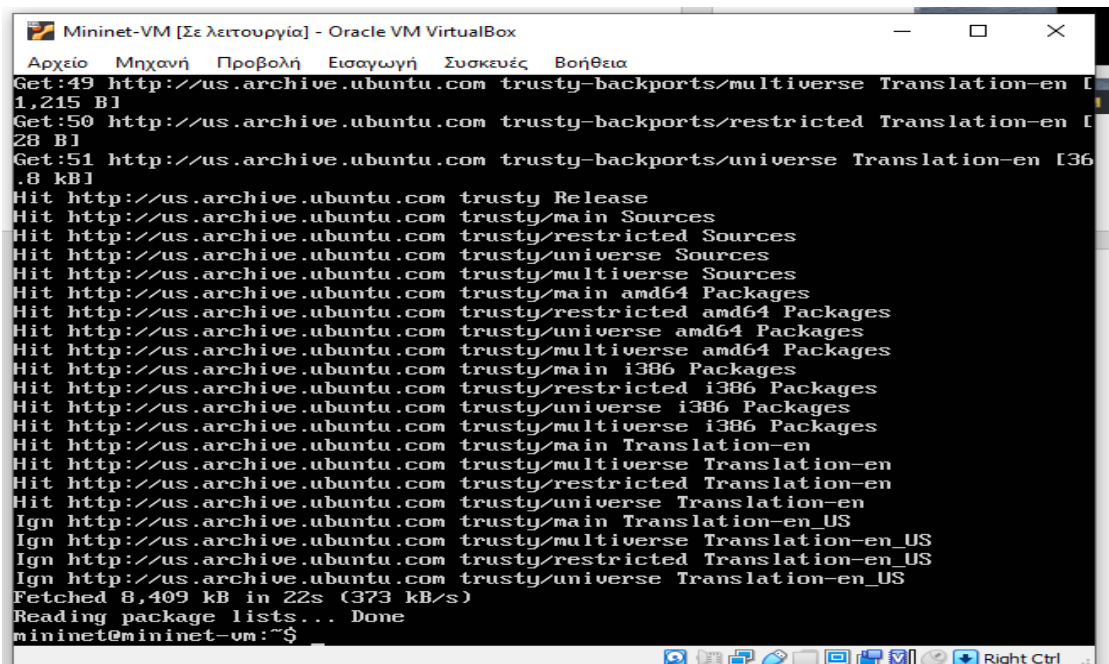


Εικόνα 4.22:Εισαγωγή όνομα χρήστη και κωδικού πρόσβασης για σύνδεση στο Mininet(ο κωδικός πρόσβασης για λόγους ασφαλείας δε φαίνεται στην οθόνη).

Το λογισμικό λειτουργεί με κώδικα Linux Based και ενδεικτικά αν τρέξουμε την εντολή:sudo apt-get update το σύστημα θα κάνει αναζήτηση για τυχών ενημερώσεις ,θα πραγματοποιήσει τη λήψη και την εγκατάστασή τους.

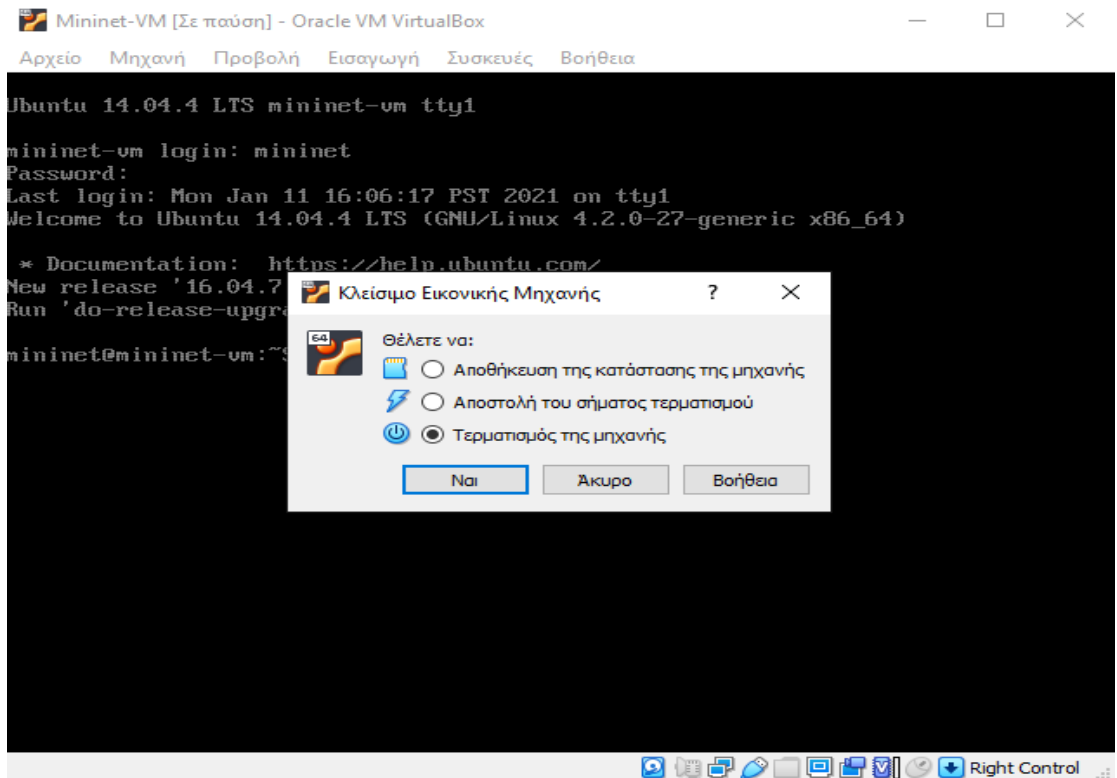


Εικόνα 53:Ενδεικτική εκτέλεση της εντολής ενημέρωσης συστήματος.



Εικόνα 4.24:Απεικόνιση ενημερώσεων.

Για να τερματίσουμε τη χρήση του Mininet και της VM πατάμε το X όπου μας εμφανίζεται η επιλογή τερματισμός.



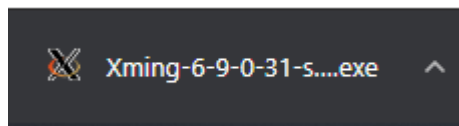
Εικόνα 4.25:Τερματισμός λειτουργίας εικονικού συστήματος.

### 4.8.3 Εγκατάσταση X-Ming

Κατεβάζουμε το X-MING από τον ιστότοπο υποστήριξής του [\[58\]](#).



Εικόνα 4.26: Κατέβασμα εφαρμογής X-Ming για Windows υπολογιστές.



Εικόνα 4.27: Όνομα αρχείου εγκατάστασης X-Ming.

Και προχωρούμε στην εγκατάστασή του ακολουθώντας τις προεπιλεγμένες οδηγίες.

#### 4.8.3.1 Περιβάλλον εργασίας X-Ming

Το X-MING λειτουργεί στο παρασκήνιο μαζί τα υπόλοιπα προγράμματα και τις διεργασίες του κάθε λογισμικού. Δεν έχει περιβάλλον εργασίας ωστόσο απαιτείται η λειτουργία του έτσι ώστε όλα τα παραπάνω προγράμματα να επικοινωνούν με αποτελεσματικότητα μεταξύ τους.

### 4.8.4 Εγκατάσταση WinSCP

Κατεβάζουμε το WinSCP από τον ιστότοπο υποστήριξής του [\[59\]](#).

## WinSCP 5.17 Download

WinSCP 5.17 is a major application update. New features and enhancements include:

- Improvements to sessions and workspace management, so that WinSCP can now easily restore tabs that were open when it was last closed.
- Hardware-accelerated AES.
- Extension *Archive and Download* to archive remote files and download the archive.
- Improvements to Synchronization checklist window.
- Allowed sorting of find results.
- SSH core upgraded to PuTTY 0.73.
- The binaries are signed with new EV certificate valid until February 2023.
- [List of all changes](#).

DOWNLOAD WINSCP 5.17.10 (10.6 MB)

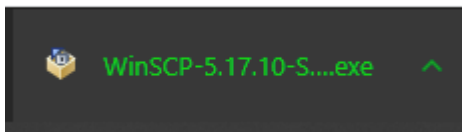


OTHER DOWNLOADS

1,733,651 downloads since 2021-01-26

What is this?

Εικόνα 4.28: Κατέβαση εφαρμογής WinSCP για Windows υπολογιστές.

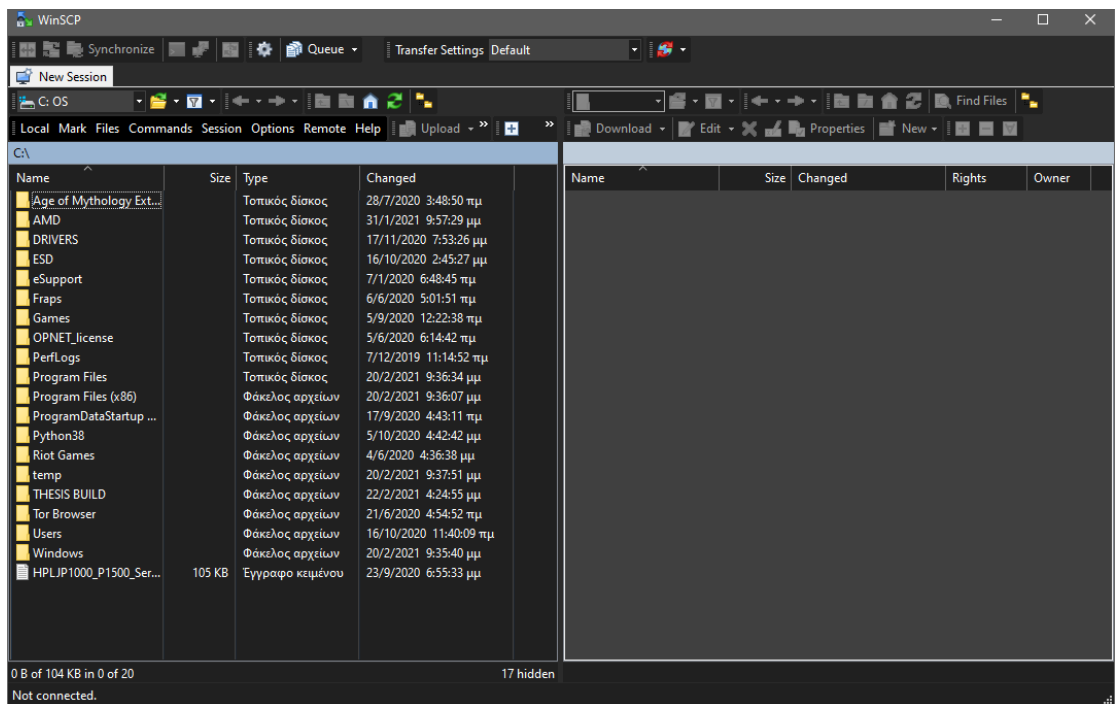


Εικόνα 4.29: Όνομα αρχείου εγκατάστασης WinSCP.

Και προχωρούμε στην εγκατάστασή του ακολουθώντας τις προεπιλεγμένες οδηγίες.

### 4.8.4.1 Περιβάλλον εργασίας WinSCP

Ανοίγοντας το WinSCP βλέπουμε το βασικό παράθυρο εργασίας του το οποίο είναι της μορφής drag & drop και είναι πολύ εύκολο στη χρήση.



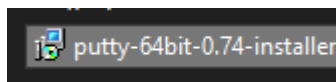
Εικόνα 4.30: Βασικό περιβάλλον εργασίας WinSCP.

## 4.8.5 Εγκατάσταση PuTTY

Κατεβάζουμε το PuTTY από τον ιστότοπο υποστήριξής του. [56]

64-bit: [putty-64bit-0.74-installer.msi](#)

Εικόνα 4.31: Κατέβασμα εφαρμογής PuTTY για Windows υπολογιστές.

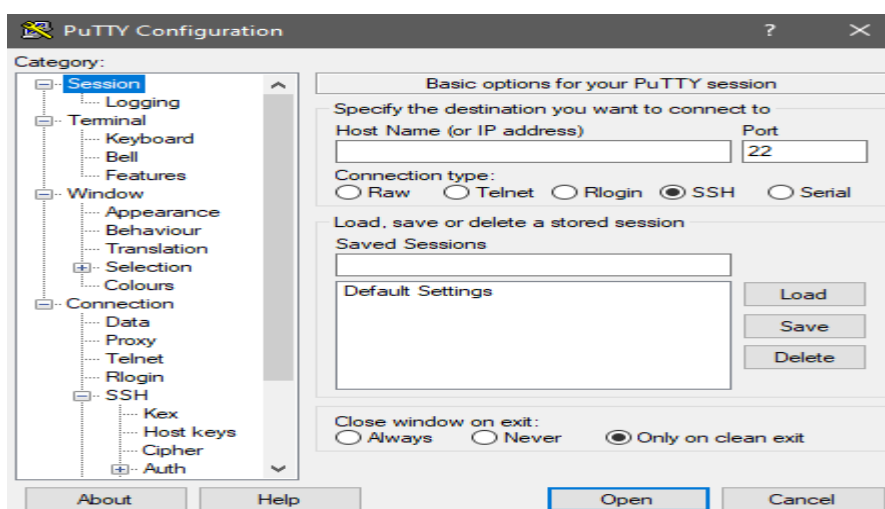


Εικόνα 4.32: Όνομα αρχείου εγκατάστασης PuTTY.

Και προχωρούμε στην εγκατάστασή του ακολουθώντας τις προεπιλεγμένες οδηγίες.

### 4.8.5.1 Περιβάλλον εργασίας PuTTY

Ανοίγοντας το PuTTY έρχεται το βασικό περιβάλλον εργασίας του PuTTY στην οθόνη το οποίο αποτελείται από ένα μενού με τη λίστα των ρυθμίσεών του.



Εικόνα 4.33: Βασικό περιβάλλον εργασίας PuTTY.

Όλες οι ρυθμίσεις γίνονται μέσω επιλογών ή συμπλήρωσης.

## 4.8.6 Εγκατάσταση Gephi

Κατεβάζουμε το Gephi από τον ιστότοπο υποστήριξής του. [57]

# Download

Gephi is an open-source and multiplatform software distributed under the dual license CDDL 1.0 and GNU General Public License v3.

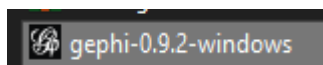
## Official Releases

[Release Notes](#) | [System Requirements](#) | [Installation instructions](#)

Gephi 0.9.2 is the latest stable release.



Εικόνα 64: Κατέβασμα εφαρμογής Gephi για υπολογιστές Windows.

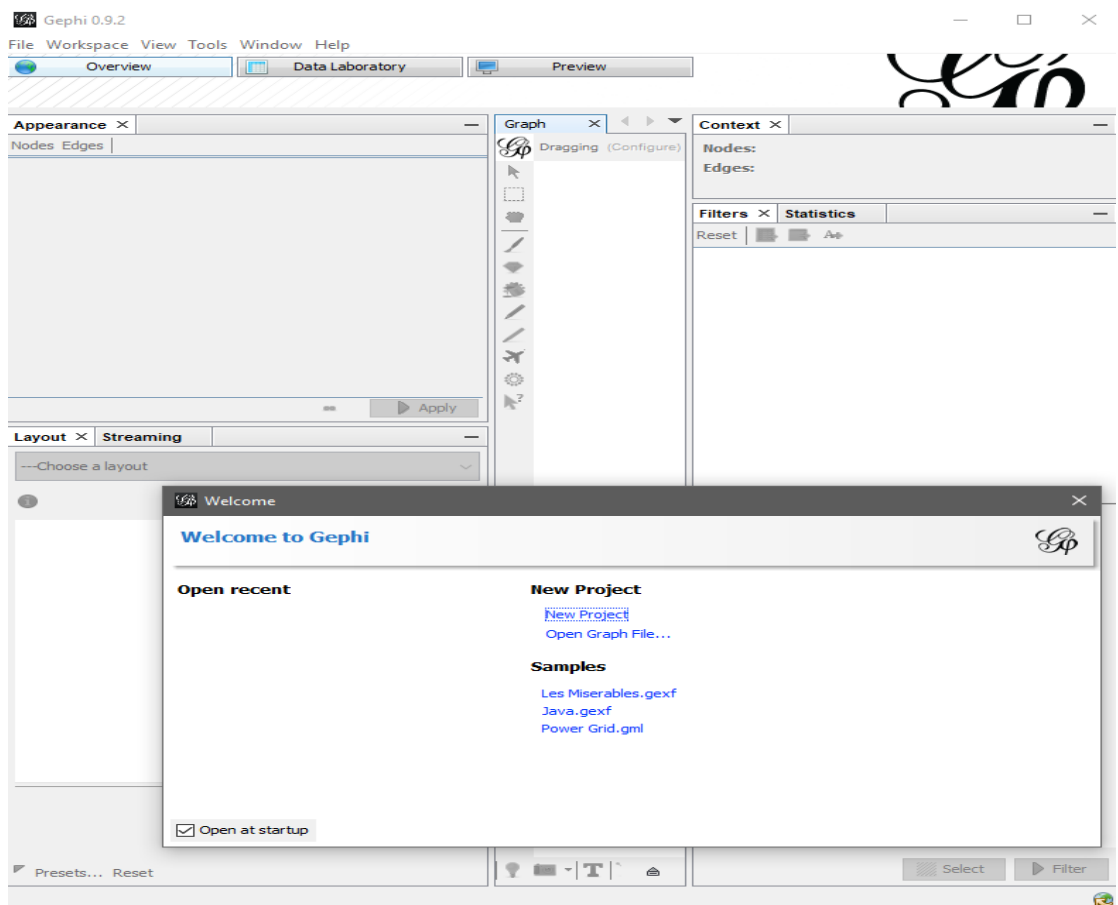


Εικόνα 4.35: Όνομα αρχείου εγκατάστασης PuTTY.

Και προχωρούμε στην εγκατάστασή του ακολουθώντας τις προεπιλεγμένες οδηγίες.

### 4.8.6.1 Περιβάλλον εργασίας Gephi

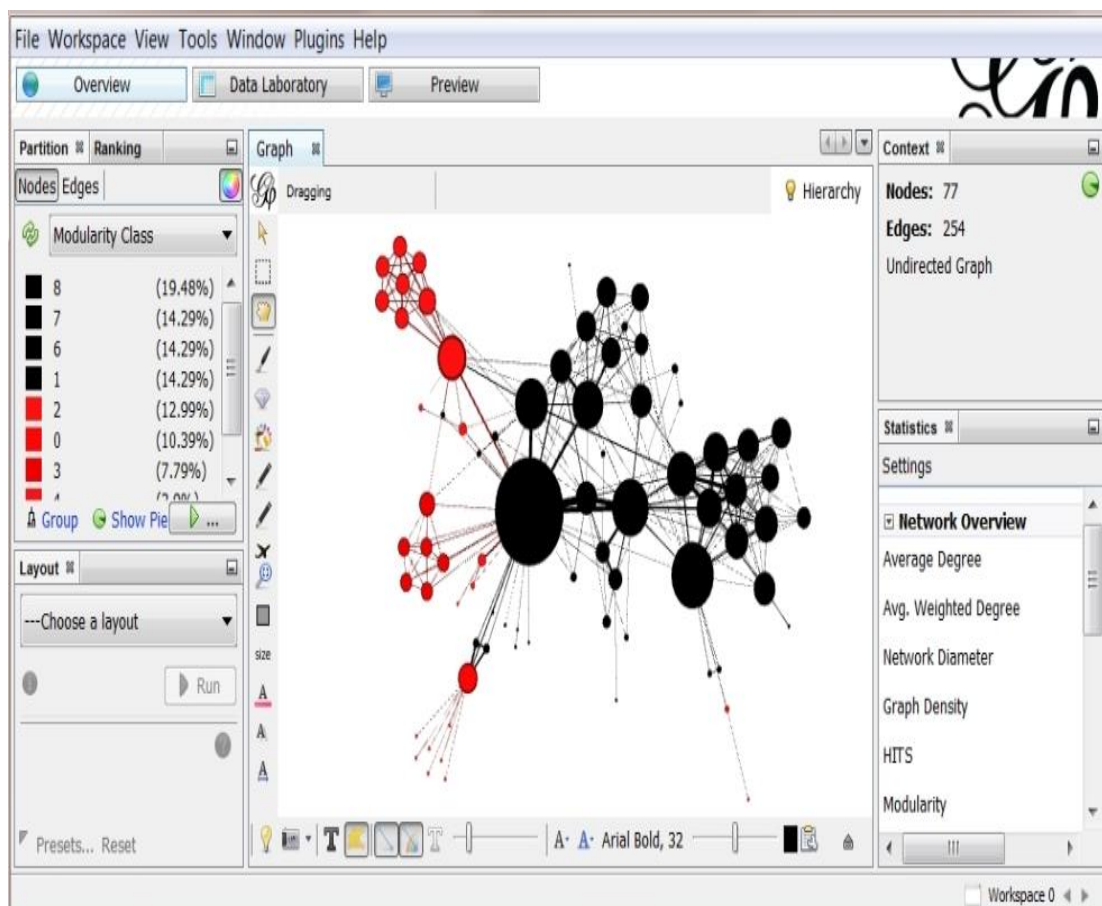
Ανοίγοντας το GEPHI βλέπουμε το βασικό περιβάλλον εργασίας του.



Εικόνα 4.36: Βασικό περιβάλλον εργασίας Gephi.



Μοιάζει πολύ με εφαρμογές τύπου επεξεργασίας εικόνας και βίντεο. Μπορούμε να εισάγουμε ένα ήδη υπάρχον Project ή Γράφημα ή και να δημιουργήσουμε κάποιο δικό μας. Επίσης παρέχει στο χρήστη τη δυνατότητα διασύνδεσης με άλλες εφαρμογές καθώς και ανάλυσης περιεχομένου. Ενδεικτικά μέσα σε ένα παράδειγμα απεικόνισης δικτύου το περιβάλλον εργασίας του Gephi ανταποκρίνεται με τον εξής τρόπο.



Εικόνα 4.37: Ενδεικτική απεικόνιση δικτύου μέσω γραφήματος στο Gephi.

## 5. Υλοποίηση εξομοίωσης SDN

Στο παρόν κεφάλαιο θα ακολουθήσει η ανάλυση των εξομοιώσεων που υλοποιήθηκαν στην παρούσα πτυχιακή εργασία με σκοπό την κατανόηση και τη λειτουργία των δικτύων SDN. Ο βασικός στόχος ήταν η δημιουργία συγκεκριμένων τοπολογιών δικτύων έτσι ώστε να παρατηρηθεί η λειτουργία τους και τα χαρακτηριστικά απόδοσης τους στο πόσο επιβαρύνουν τη λειτουργία ενός συστήματος και πόσο μπορούν να βοηθήσουν τελικά ώστε να εδραιωθεί η τεχνολογία SDN.

### 5.1 Τοπολογίες εξομοίωσης

Στην προσομοίωση που θα ακολουθήσει στην παρούσα πτυχιακή εργασία θα υλοποιηθούν 3 διαφορετικές τοπολογίες προκειμένου να καλυφθούν όλες οι κατηγορίες. Στις βασικές τοπολογίες θα υλοποιηθεί η Linear η οποία χαρακτηρίζεται από την απλότητα στην κατασκευή της. Στις υβριδικές τοπολογίες θα κατασκευαστεί η Tree λόγω του συνδυαστικού της χαρακτήρα ενώ στις τοπολογίες προσαρμοσμένων χαρακτηριστικών θα υλοποιηθεί η Erdos Renyi λόγω του πιο πολύπλοκού της χαρακτήρα. Προκειμένου να πραγματοποιηθεί οποιοδήποτε είδος εξομοίωσης πρέπει να είναι γνωστά τα δεδομένα της. Θα δημιουργηθούν 3 διαφορετικά δίκτυα με βάση της τοπολογίες τους έτσι ώστε να καλυφτεί μια σημαντική μερίδα των βασικότερων τοπολογιών. Σε κάθε τοπολογία θα εκτελεστούν 6 διαφορετικές εξομοιώσεις με κατάλληλα επεξεργασμένα χαρακτηριστικά έτσι ώστε να εξαχθούν ακριβή αποτελέσματα.

- TREE

- Στην τοπολογία TREE θα κρατηθεί σταθερό depth (βάθος) στο δέντρο και θα αλλάζεται το fanout (άπλωμα) με διπλασιασμό του κάθε φορά ώστε να μελετηθεί το πως επηρεάζεται η λειτουργία της τοπολογίας από τη σταθερή αύξηση των switches και των hosts.

- Συγκεκριμένα θα εκτελεστεί εξομοίωση με σταθερό “depth=2” και “fanout”[2,4,8,16,32,64].

- LINEAR

- Στην τοπολογία LINEAR θα κρατηθεί σταθερός ο αριθμός των switch σε 2 και θα μελετηθεί το πως επηρεάζεται η λειτουργία της τοπολογίας από την υπερφόρτωση των switches με τον συνεχή διπλασιασμό των hosts [2,4,8,16,32,64].

- ERDOS RENEYI

- Στην τοπολογία ERDOS RENEYI που διαφέρει από τις 2 παραπάνω καθώς είναι custom και δημιουργεί τυχαία γραφήματα, θα αλλαχθεί ο αριθμός των κόμβων  $n$  καθώς με σταθερή πιθανότητα ενός άκρου μεταξύ κόμβων  $p$  με σκοπό την μελέτη του φόρτου της τοπολογίας. Η πιθανότητα  $p$  θα είναι 0.6 και ο αριθμός κόμβων για να ταιριάζει συγκριτικά και με τις 2 προηγούμενες τοπολογίες θα είναι [2,4,8,16,32,64].

## 5.2 Μετρικές επιδόσεις

Τα αποτελέσματα που θα εξαχθούν θα είναι για την κάθε υλοποίηση της κάθε τοπολογίας και αυτά αποτελούνται από:

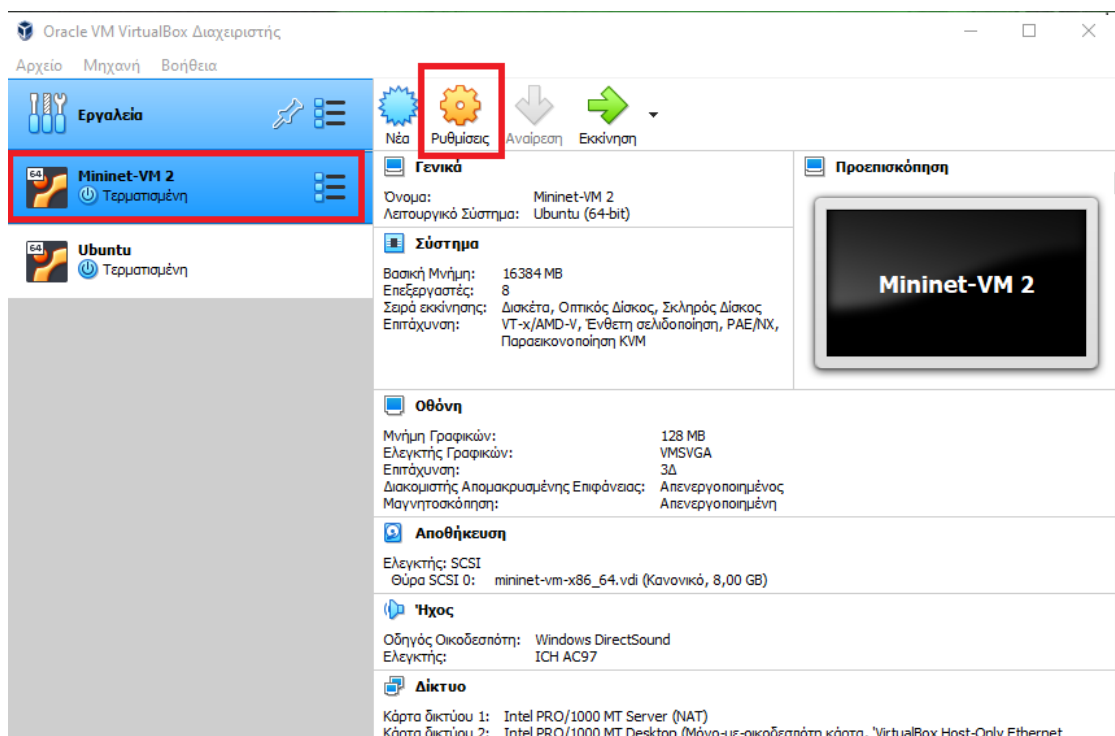
- CPU Usage (%)
  - Ποσοστό χρήσης της CPU από το εκτελέσιμο πρόγραμμα κατά μέσο όρο.
- Memory Usage (MB)
  - Αριθμός χρήσης της μνήμης RAM από το εκτελέσιμο πρόγραμμα κατά μέσο όρο σε MB.
- Switch Number
  - Αριθμός switch στην προσομοίωση.
- Hosts Number
  - Αριθμός hosts στην προσομοίωση.
- Bandwidth (Gbps)
  - Αριθμός εύρους σύνδεσης στην προσομοίωση κατά μέσο όρο σε Gbps.
- Latency (ms)
  - Αριθμός καθυστέρησης της επικοινωνίας μεταξύ κόμβων στην προσομοίωση σε ms.
- Setup Time (s)
  - Αριθμός δημιουργίας της εκάστοτε τοπολογίας σε sec.
- Tear Time (s)
  - Αριθμός καταστροφής της εκάστοτε τοπολογίας σε sec.

## 5.3 Ρύθμιση Προγραμμάτων

Για την επιτυχή εξομίωση πρέπει να ρυθμιστεί το κάθε λογισμικό έτσι ώστε να μπορέσει να υποστηρίξει τις παραπάνω εξομιώσεις και να εξάγει τα επιθυμητά αποτελέσματα. Σημειώνεται ότι για την επιτυχή χρήση όλων των προγραμμάτων πρέπει πάντα να λειτουργεί το Mininet μέσω του VirtualBox καθώς και το X-Ming.

### 5.3.1 Ρύθμιση VirtualBox

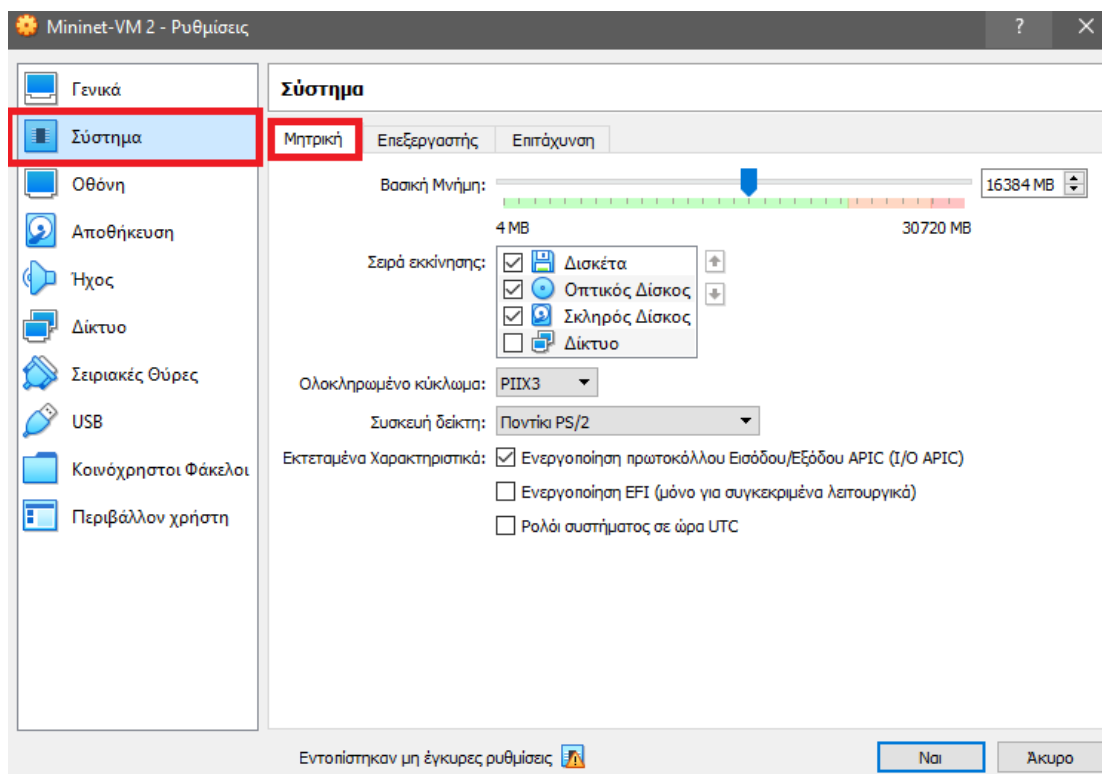
Ανοίγοντας το VirtualBox στο αρχικό του menu επιλέγουμε το εγκατεστημένο εικονικό λογισμικό που θέλουμε να χρησιμοποιήσουμε(στην περίπτωσή μας το Mininet).



Εικόνα 5.1:Ρυθμίσεις VirtualBox.

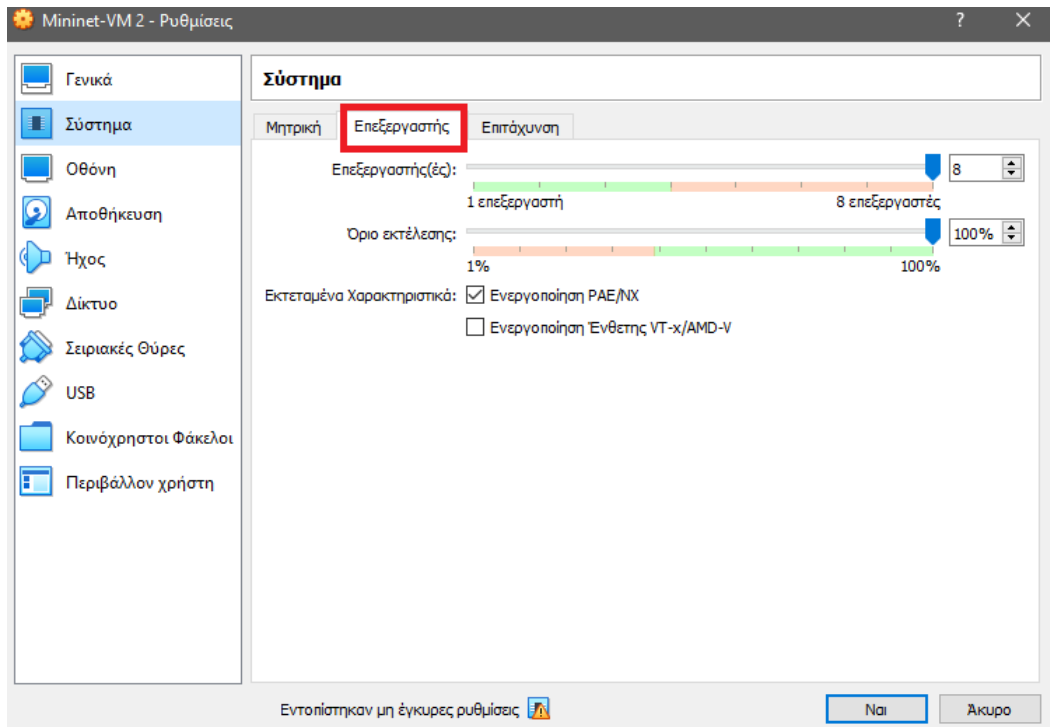
Στη συνέχεια επιλέγουμε τις «Ρυθμίσεις» έτσι ώστε να ρυθμίσουμε τις τεχνικές προδιαγραφές του συστήματος. Οι γενικές ρυθμίσεις μένουν στις προκαθορισμένες για το εκάστοτε λογισμικό και επιλέγουμε την επιλογή «Σύστημα». Στο σύστημα μπορούμε να ρυθμίσουμε τους διαθέσιμους πόρους που θα αντλήσει το εικονικό σύστημα από το πραγματικό σύστημα έχοντας άμεσο αντίκτυπο στις επιδόσεις του. Στο υπομενού(submenu) «Μητρική» επιλέγουμε πόση μνήμη RAM θα μπορεί να χρησιμοποιήσει το εικονικό σύστημα. Στη συγκεκριμένη περίπτωση όπου η εξομίωση θα εκτελεστεί σε ένα High-End σύστημα

εκχωρούμε 16384MB (16GB) μνήμης RAM τα οποία όπως αναφέρθηκε και στο κεφάλαιο 4.1.2 είναι ένα νούμερο παραπάνω από αρκετό για οποιαδήποτε μορφής υλοποίηση.



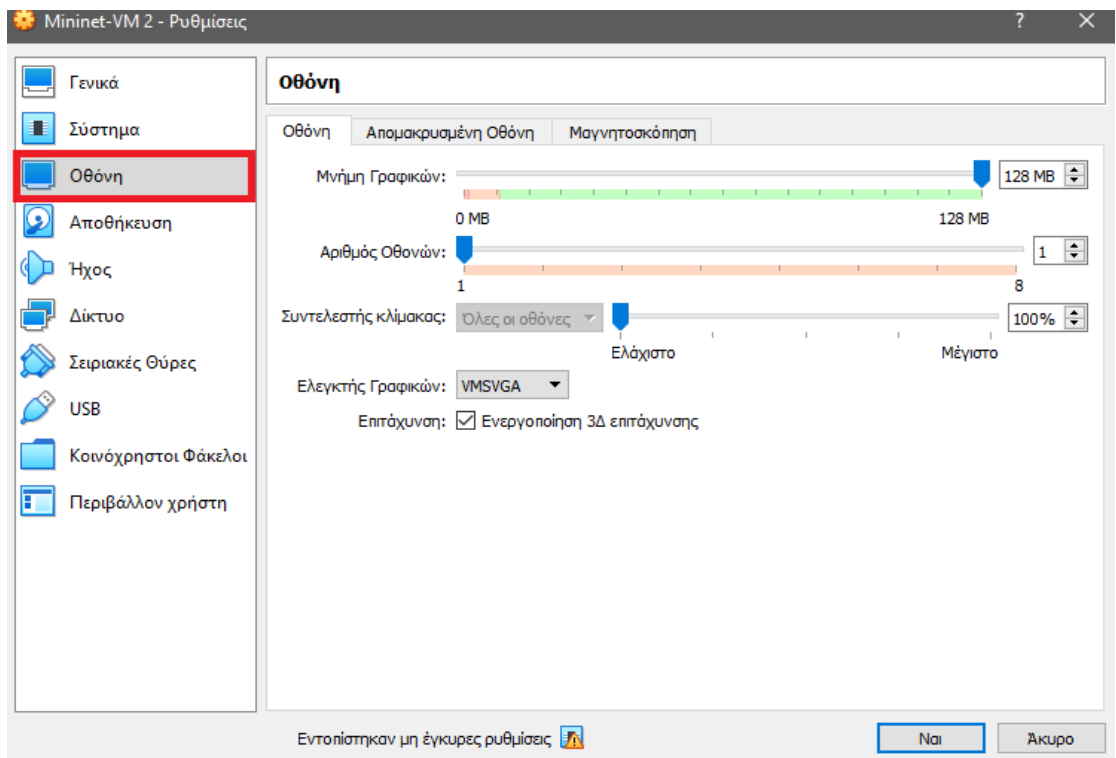
Εικόνα 5.2:Εκχώρηση μνήμης RAM από το φυσικό σύστημα στο εικονικό σύστημα Mininet.

Προχωρώντας στο υπομενού «Επεξεργαστής» ορίζουμε τον αριθμό των CPU THREADS που θα μπορεί να διαχειριστεί το εικονικό σύστημα καθώς και το ποσοστό της ισχύος του επεξεργαστή. Στη συγκεκριμένη έρευνα προκειμένου να ληφθούν αποτελέσματα με όσο το δυνατόν μεγαλύτερη ακρίβεια θα υπάρχει η δυνατότητα για πλήρη εκμετάλλευση της επεξεργαστικής ισχύος, η οποία στην περίπτωση αυτή είναι 8 Threads.



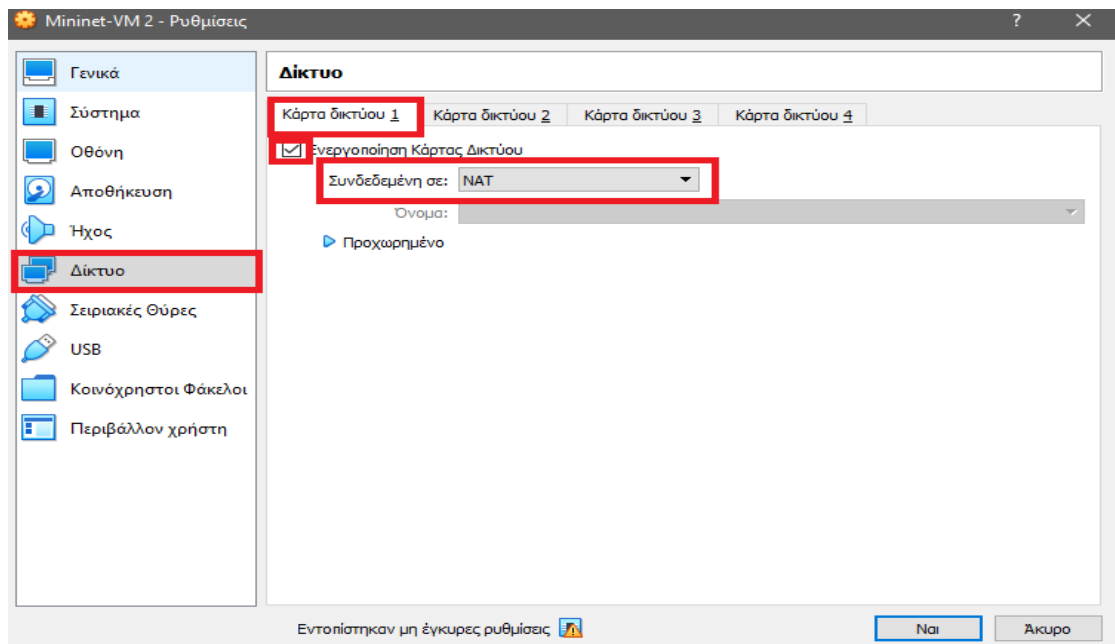
Εικόνα 5.3:Εκχώρηση πυρήνων και νημάτων επεξεργαστή από το φυσικό σύστημα στο εικονικό σύστημα Mininet.

Στην επιλογή «Οθόνη» απλώς εκχωρούμε τη μέγιστη δυνατή μνήμη γραφικών 128MB για απεικόνιση VMSVGA καθώς το Mininet δεν έχει μεγαλύτερες απαιτήσεις απεικόνισης πέρα από τις απολύτως βασικές.



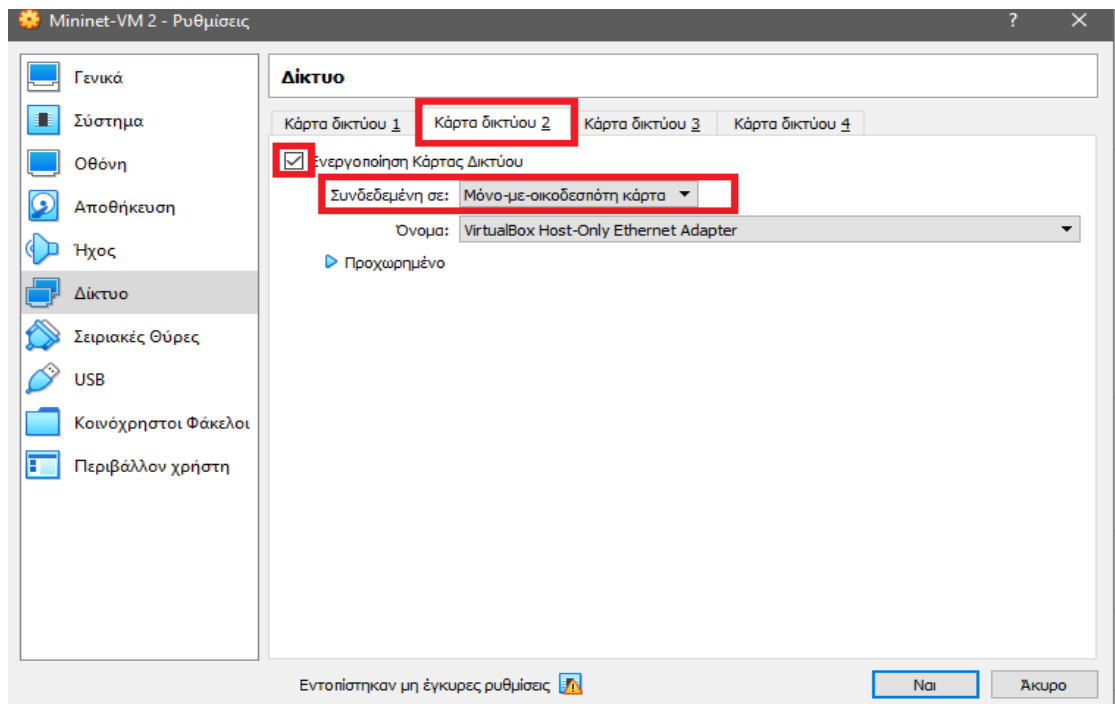
Εικόνα 5.4:Εκχώρηση μνήμης γραφικών (VRAM) από το φυσικό σύστημα στο εικονικό σύστημα Mininet.<

Στην επιλογή «Δίκτυο» στο υπομενού «Κάρτα Δικτύου 1» την ενεργοποιούμε και την ρυθμίζουμε να είναι τύπου «NAT» έτσι ώστε η εικονική μηχανή να έχει πρόσβαση στο διαδίκτυο.



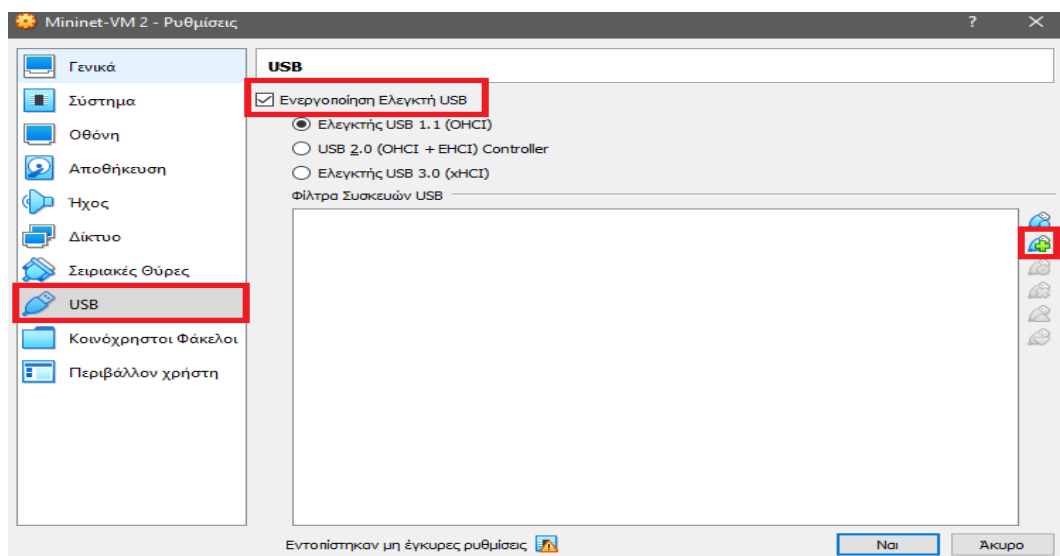
Εικόνα 5.5: Ρύθμιση βασικής κάρτας δικτύου για πρόσβαση της εικονικής μηχανής στο internet.

Στο υπομενού «Κάρτα Δικτύου 2» την ενεργοποιούμε και την ρυθμίζουμε να είναι τύπου «Host-Only» έτσι ώστε η εικονική μηχανή να μπορεί να δεχθεί πρόσβαση τύπου SSH ώστε να επικοινωνεί με το φυσικό σύστημα στο οποίο λειτουργεί. Η χρήση SSH παρέχει τη δυνατότητα μεταφοράς αρχείων και εκτέλεσης πολλαπλών παραθύρων καθώς και επεκτείνει τις εκτελεστικές δυνατότητες της εικονικής μηχανής.



Εικόνα 5.6:Ρύθμιση δευτερεύουσας κάρτας δικτύου για πρόσβαση τύπου SSH.

Στην επιλογή «USB» παρέχεται οι δυνατότητα επικοινωνίας συσκευών USB του φυσικού συστήματος με το εικονικό. Συσκευές όπως ( ποντίκι, πληκτρολόγιο ) ανιχνεύονται αυτόματα αλλά στην περίπτωση που θέλουμε να επικοινωνήσουμε με ένα αποθηκευτικό μέσο όπως «USB FLASH DRIVE» τότε το προσθέτουμε από το αντίστοιχο εικονίδιο.

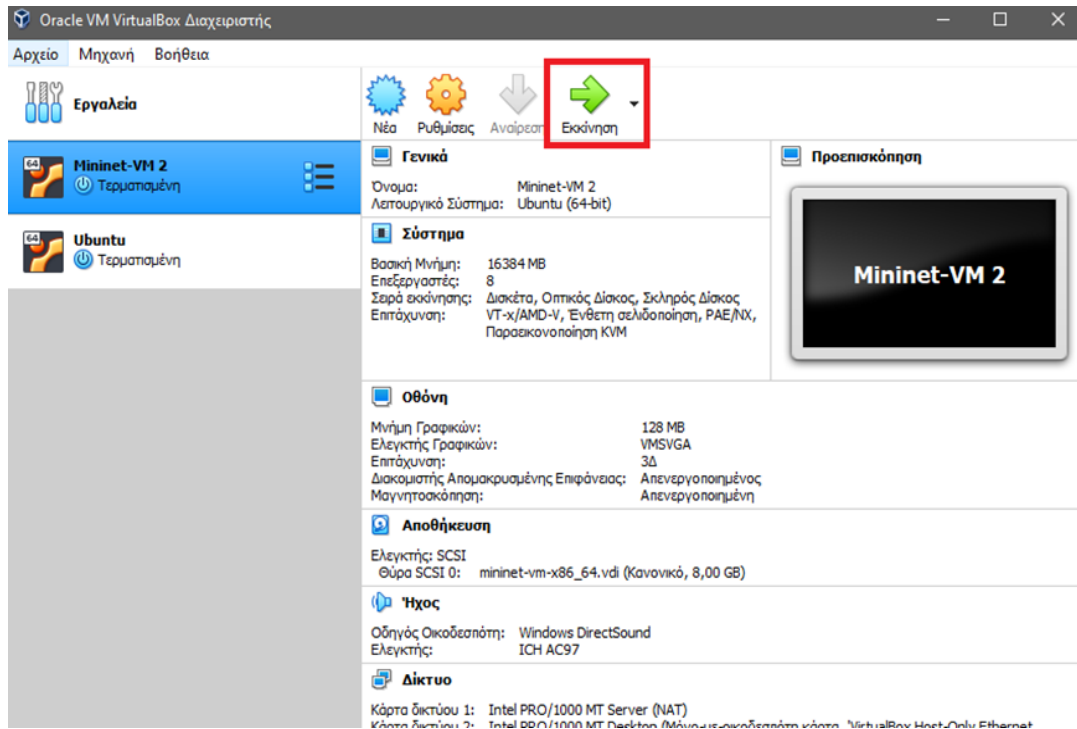


Εικόνα 5.7:Ενεργοποίηση ελεγκτή USB.



### 5.3.2 Ρύθμιση Mininet

Πραγματοποιούμε εκκίνηση του λογισμικού Mininet.



Εικόνα 5.8: Εκκίνηση λογισμικού Mininet.

Με τον ίδιο τρόπο που αναφέρθηκε στο κεφάλαιο 4.7.2.1 συνδεόμαστε στο Mininet και πληκτρολογώντας την εντολή `ifconfig -a` μπορούμε να πάρουμε την διεύθυνση IP (IP Address) της κάρτας δικτύου της εικονικής μηχανής προκειμένου να επικοινωνήσει με το φυσικό σύστημα. Στη συγκεκριμένη περίπτωση είναι η διεύθυνση: 192.168.56.106 και αυτή η πληροφορία θα χρησιμεύσει στη συνέχεια και στη συνεργασία του Mininet με τις υπόλοιπες εφαρμογές καθώς και για δημιουργία σύνδεσης SSH.

```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
Last login: Tue Feb 23 14:33:47 PST 2021 from 192.168.56.1 on pts/1
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)

* Documentation: https://help.ubuntu.com/
mininet@mininet-vm:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:ac:1a:25
          inet addr:192.168.56.106  Bcast:192.168.56.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:161 errors:0 dropped:0 overruns:0 frame:0
          TX packets:181 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18140 (18.1 KB)  TX bytes:29514 (29.5 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:2c:aa:cd
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:250 errors:0 dropped:0 overruns:0 frame:0
          TX packets:165 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:43903 (43.9 KB)  TX bytes:31123 (31.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:11956 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11956 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:604160 (604.1 KB)  TX bytes:604160 (604.1 KB)

vovs-system Link encap:Ethernet  HWaddr 86:11:d0:5c:79:03
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:7866 (7.8 KB)

mininet@mininet-vm:~$
```

Εικόνα 5.9:Εύρεση διεύθυνσης IP κάρτας δικτύου Mininet μέσω της εντολής «ifconfig -a».

Πληκτρολογώντας την εντολή «sudo dhclient» παρέχεται η δυνατότητα στην εικονική μηχανή να έχει πρόσβαση στο ίντερνετ και στη συνέχεια παρομοίως με το κεφάλαιο 4.7.2.1 μέσω της εντολής «sudo apt-get update» θα πραγματοποιηθεί ενημέρωση του παρόντος λογισμικού.

```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσσκευές Βοήθεια

* Documentation: https://help.ubuntu.com/
mininet@mininet-vm:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:ac:1a:25
          inet addr:192.168.56.106 Bcast:192.168.56.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:161 errors:0 dropped:0 overruns:0 frame:0
          TX packets:181 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18140 (18.1 KB)  TX bytes:29514 (29.5 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:2c:aa:cd
          inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:250 errors:0 dropped:0 overruns:0 frame:0
          TX packets:165 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:43903 (43.9 KB)  TX bytes:31123 (31.1 KB)

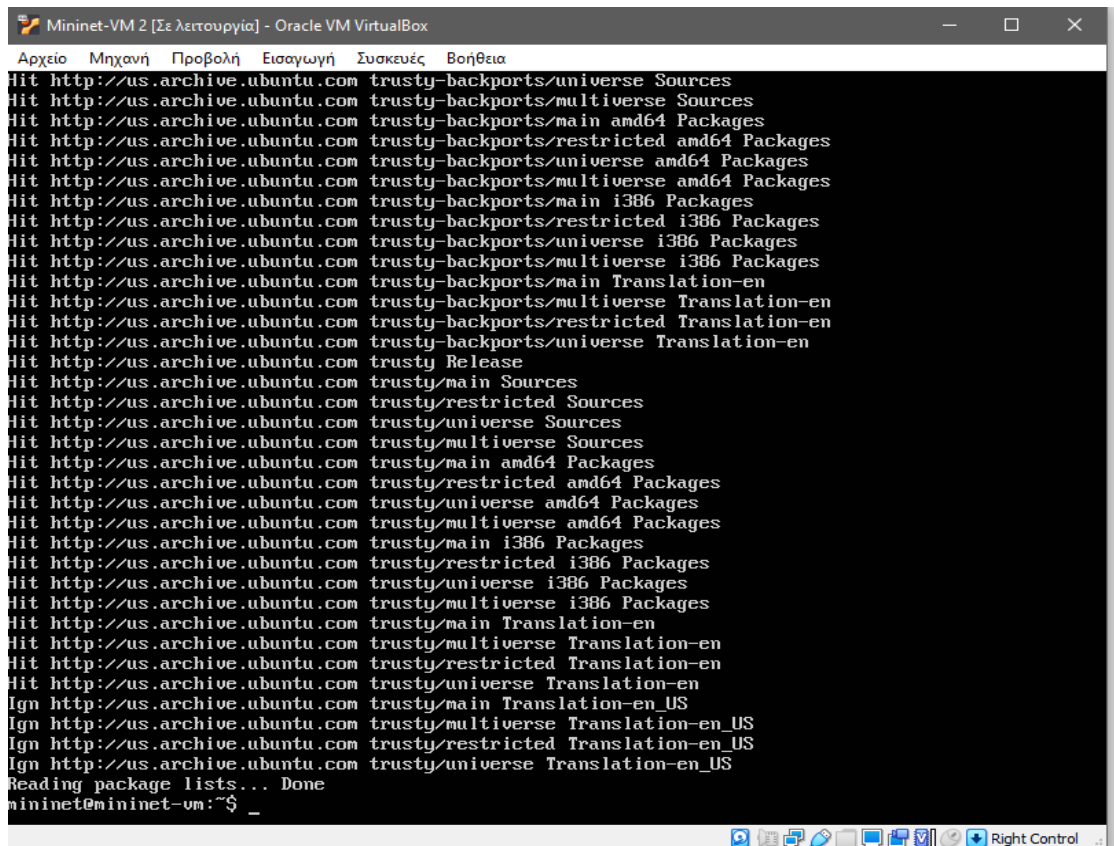
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:11956 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11956 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:604160 (604.1 KB)  TX bytes:604160 (604.1 KB)

ovs-system Link encap:Ethernet  HWaddr 86:11:d0:5c:79:03
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:7866 (7.8 KB)

mininet@mininet-vm:~$ sudo dhclient
RTNETLINK answers: File exists
mininet@mininet-vm:~$ sudo apt-get update_
```

Εικόνα 5.10: Εκτέλεση εντολής «`sudo dhclient`» για πρόσβαση στο internet και «`sudo apt-get update`» για επιτυχή ενημέρωση συστήματος.

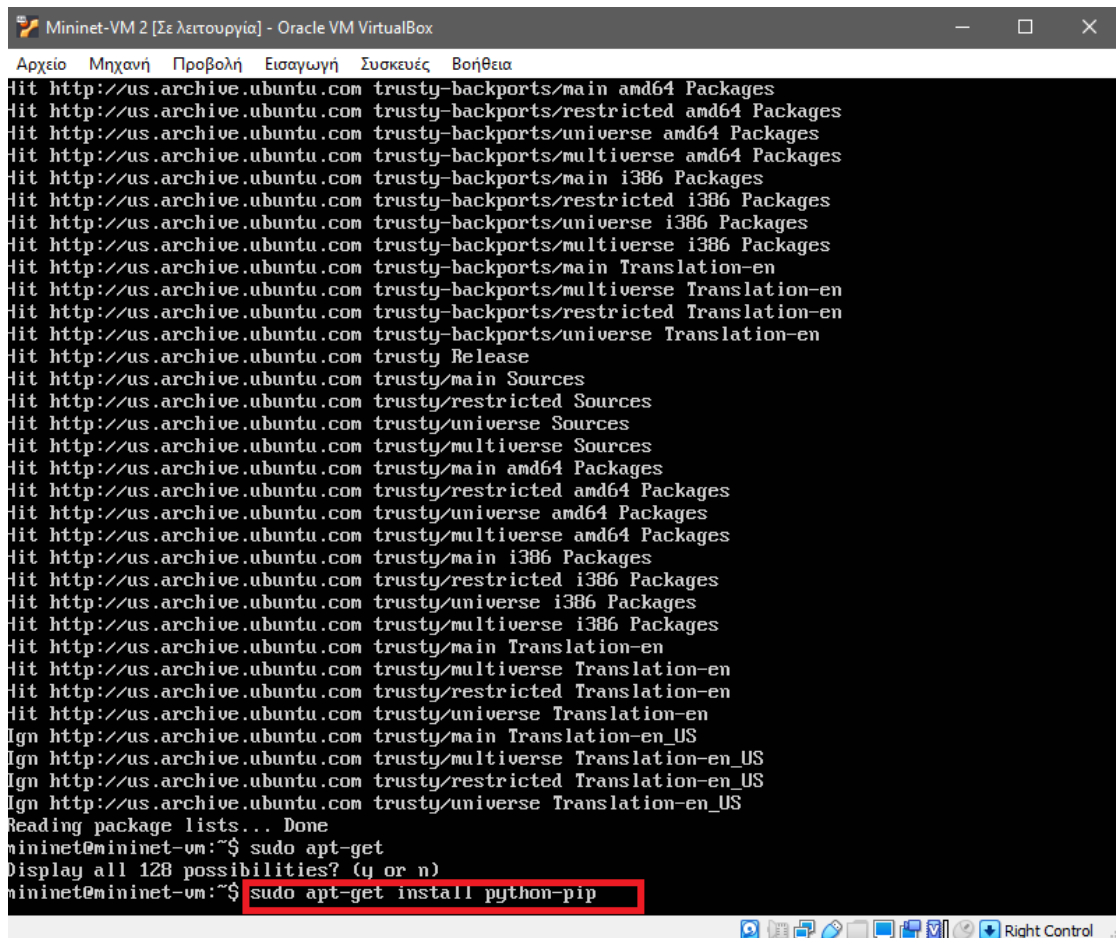
Παρέχοντας τη δυνατότητα αναλυτικής παρακολούθησης σε οτιδήποτε ενημερώνεται.

The image shows a terminal window titled "Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox". The terminal output displays the results of an 'apt-get update' command. It lists various package sources and their corresponding package counts for different architectures (amd64 and i386) and components (Sources, Packages, Translation-en, Release). The output ends with "Reading package lists... Done" and a prompt "mininet@mininet-vm:~\$".

```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο  Μηχανή  Προβολή  Εισαγωγή  Συσκευές  Βοήθεια
Hit http://us.archive.ubuntu.com trusty-backports/universe Sources
Hit http://us.archive.ubuntu.com trusty-backports/multiverse Sources
Hit http://us.archive.ubuntu.com trusty-backports/main amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/restricted amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/universe amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/multiverse amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/main i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/restricted i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/universe i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/multiverse i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/main Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/restricted Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/universe Translation-en
Hit http://us.archive.ubuntu.com trusty Release
Hit http://us.archive.ubuntu.com trusty/main Sources
Hit http://us.archive.ubuntu.com trusty/restricted Sources
Hit http://us.archive.ubuntu.com trusty/universe Sources
Hit http://us.archive.ubuntu.com trusty/multiverse Sources
Hit http://us.archive.ubuntu.com trusty/main amd64 Packages
Hit http://us.archive.ubuntu.com trusty/restricted amd64 Packages
Hit http://us.archive.ubuntu.com trusty/universe amd64 Packages
Hit http://us.archive.ubuntu.com trusty/multiverse amd64 Packages
Hit http://us.archive.ubuntu.com trusty/main i386 Packages
Hit http://us.archive.ubuntu.com trusty/restricted i386 Packages
Hit http://us.archive.ubuntu.com trusty/universe i386 Packages
Hit http://us.archive.ubuntu.com trusty/multiverse i386 Packages
Hit http://us.archive.ubuntu.com trusty/main Translation-en
Hit http://us.archive.ubuntu.com trusty/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty/restricted Translation-en
Hit http://us.archive.ubuntu.com trusty/universe Translation-en
Ign http://us.archive.ubuntu.com trusty/main Translation-en_US
Ign http://us.archive.ubuntu.com trusty/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com trusty/restricted Translation-en_US
Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US
Reading package lists... Done
mininet@mininet-vm:~$
```

Εικόνα 5.11:Επιτυχής ενημέρωση συστήματος.

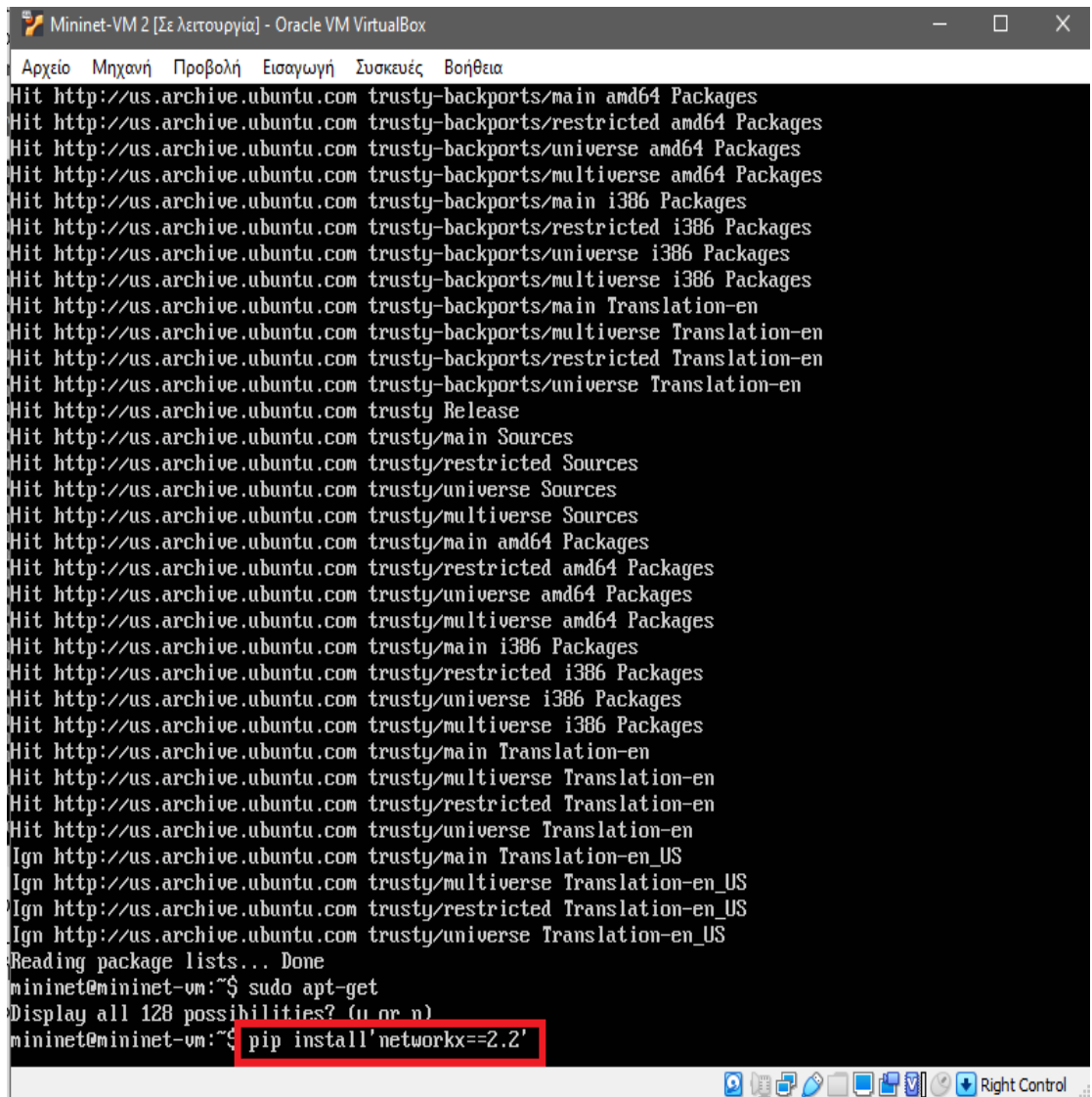
Το Mininet όπως αναφέρθηκε και στο κεφάλαιο 4.1 έχει προεγκατεστημένες όλες τις λειτουργίες για την επιτυχή δημιουργία των κυριότερων τοπολογιών δικτύων συμπεριλαμβανομένων και των TREE και LINEAR που θα κατασκευαστούν στην πορεία. Για την κατασκευή όμως την custom τοπολογίας ERDOS-RENYI θα πρέπει να προστεθούν με τη μορφή addons κάποιες δυνατότητες. Προκειμένου να δημιουργηθεί η τοπολογία αυτή θα πρέπει να μεταφέρουμε ένα αρχείο που περιέχει τον κατάλληλο κώδικα σε γλώσσα PYTHON μέσω της οποίας το Mininet επιτρέπει παραμετροποιήσεις και αλλαγές. Για να επιτρέψουμε στο Mininet να έχει τη δυνατότητα επεξεργασίας, εκτέλεσης και διαβάσματος αρχείων τύπου Python «.py» εγκαθιστούμε την επιπλέον υποστήριξη «python-rip» με την εντολή «sudo apt-get install python-rip».



```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
hit http://us.archive.ubuntu.com trusty-backports/main amd64 Packages
hit http://us.archive.ubuntu.com trusty-backports/restricted amd64 Packages
hit http://us.archive.ubuntu.com trusty-backports/universe amd64 Packages
hit http://us.archive.ubuntu.com trusty-backports/multiverse amd64 Packages
hit http://us.archive.ubuntu.com trusty-backports/main i386 Packages
hit http://us.archive.ubuntu.com trusty-backports/restricted i386 Packages
hit http://us.archive.ubuntu.com trusty-backports/universe i386 Packages
hit http://us.archive.ubuntu.com trusty-backports/multiverse i386 Packages
hit http://us.archive.ubuntu.com trusty-backports/main Translation-en
hit http://us.archive.ubuntu.com trusty-backports/multiverse Translation-en
hit http://us.archive.ubuntu.com trusty-backports/restricted Translation-en
hit http://us.archive.ubuntu.com trusty-backports/universe Translation-en
hit http://us.archive.ubuntu.com trusty Release
hit http://us.archive.ubuntu.com trusty/main Sources
hit http://us.archive.ubuntu.com trusty/restricted Sources
hit http://us.archive.ubuntu.com trusty/universe Sources
hit http://us.archive.ubuntu.com trusty/multiverse Sources
hit http://us.archive.ubuntu.com trusty/main amd64 Packages
hit http://us.archive.ubuntu.com trusty/restricted amd64 Packages
hit http://us.archive.ubuntu.com trusty/universe amd64 Packages
hit http://us.archive.ubuntu.com trusty/multiverse amd64 Packages
hit http://us.archive.ubuntu.com trusty/main i386 Packages
hit http://us.archive.ubuntu.com trusty/restricted i386 Packages
hit http://us.archive.ubuntu.com trusty/universe i386 Packages
hit http://us.archive.ubuntu.com trusty/multiverse i386 Packages
hit http://us.archive.ubuntu.com trusty/main Translation-en
hit http://us.archive.ubuntu.com trusty/multiverse Translation-en
hit http://us.archive.ubuntu.com trusty/restricted Translation-en
hit http://us.archive.ubuntu.com trusty/universe Translation-en
Ign http://us.archive.ubuntu.com trusty/main Translation-en_US
Ign http://us.archive.ubuntu.com trusty/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com trusty/restricted Translation-en_US
Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US
Reading package lists... Done
mininet@mininet-vm:~$ sudo apt-get
Display all 128 possibilities? (y or n)
mininet@mininet-vm:~$ sudo apt-get install python-pip
```

Εικόνα 5.12:Εγκατάσταση προγράμματος «pip» για υποστήριξη επεξεργασίας αρχείων γλώσσας python.

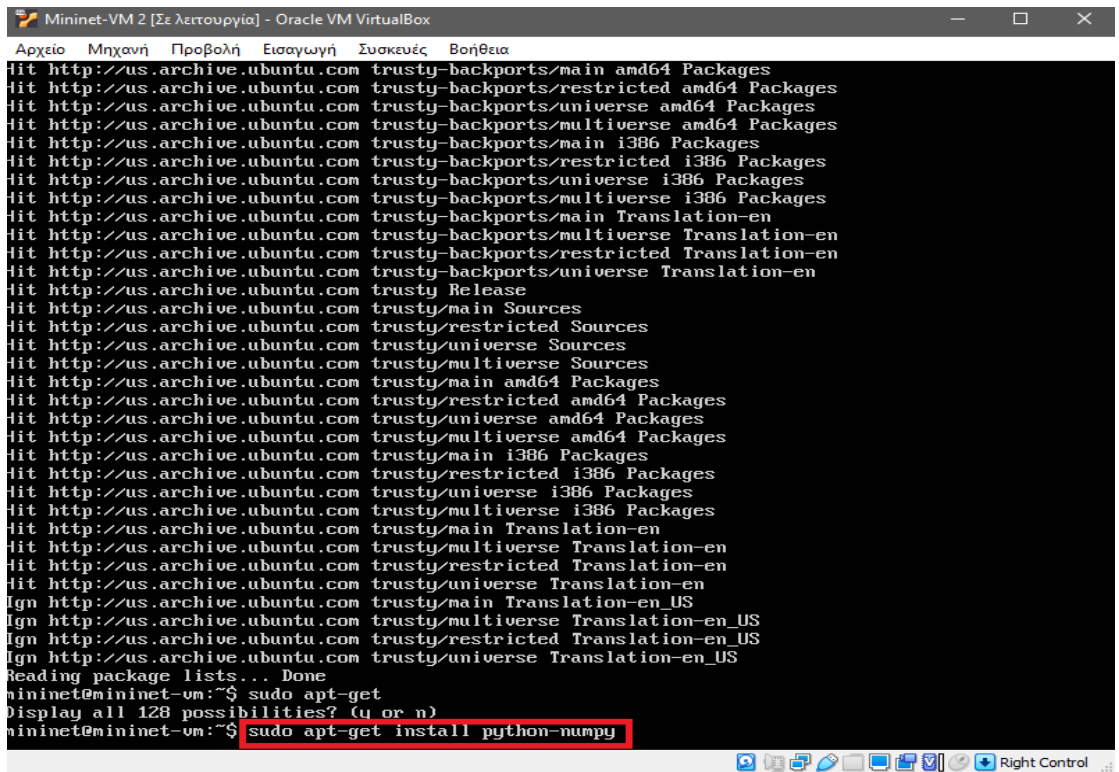
Μέσω του «pip» εγκαθίσταται η δυνατότητα «network» που είναι μία βιβλιοθήκη custom τοπολογιών. Η εγκατάσταση πραγματοποιείται μέσω της εντολής «pip install`networkx==2.2». Η έκδοση 2.2 του «networkx» χρησιμοποιήθηκε λόγω του όπως αποδεικνύεται αυτή η έκδοση λειτουργεί καλύτερα σε εικονικές μηχανές Mininet.



```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
Hit http://us.archive.ubuntu.com trusty-backports/main amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/restricted amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/universe amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/multiverse amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/main i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/restricted i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/universe i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/multiverse i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/main Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/restricted Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/universe Translation-en
Hit http://us.archive.ubuntu.com trusty Release
Hit http://us.archive.ubuntu.com trusty/main Sources
Hit http://us.archive.ubuntu.com trusty/restricted Sources
Hit http://us.archive.ubuntu.com trusty/universe Sources
Hit http://us.archive.ubuntu.com trusty/multiverse Sources
Hit http://us.archive.ubuntu.com trusty/main amd64 Packages
Hit http://us.archive.ubuntu.com trusty/restricted amd64 Packages
Hit http://us.archive.ubuntu.com trusty/universe amd64 Packages
Hit http://us.archive.ubuntu.com trusty/multiverse amd64 Packages
Hit http://us.archive.ubuntu.com trusty/main i386 Packages
Hit http://us.archive.ubuntu.com trusty/restricted i386 Packages
Hit http://us.archive.ubuntu.com trusty/universe i386 Packages
Hit http://us.archive.ubuntu.com trusty/multiverse i386 Packages
Hit http://us.archive.ubuntu.com trusty/main Translation-en
Hit http://us.archive.ubuntu.com trusty/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty/restricted Translation-en
Hit http://us.archive.ubuntu.com trusty/universe Translation-en
Ign http://us.archive.ubuntu.com trusty/main Translation-en_US
Ign http://us.archive.ubuntu.com trusty/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com trusty/restricted Translation-en_US
Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US
Reading package lists... Done
mininet@mininet-vm:~$ sudo apt-get
Display all 128 possibilities? (y or n)
mininet@mininet-vm:~$ pip install networkx==2.2'
```

Εικόνα 5.13: Εγκατάσταση βιβλιοθήκης τοπολογιών networkx.

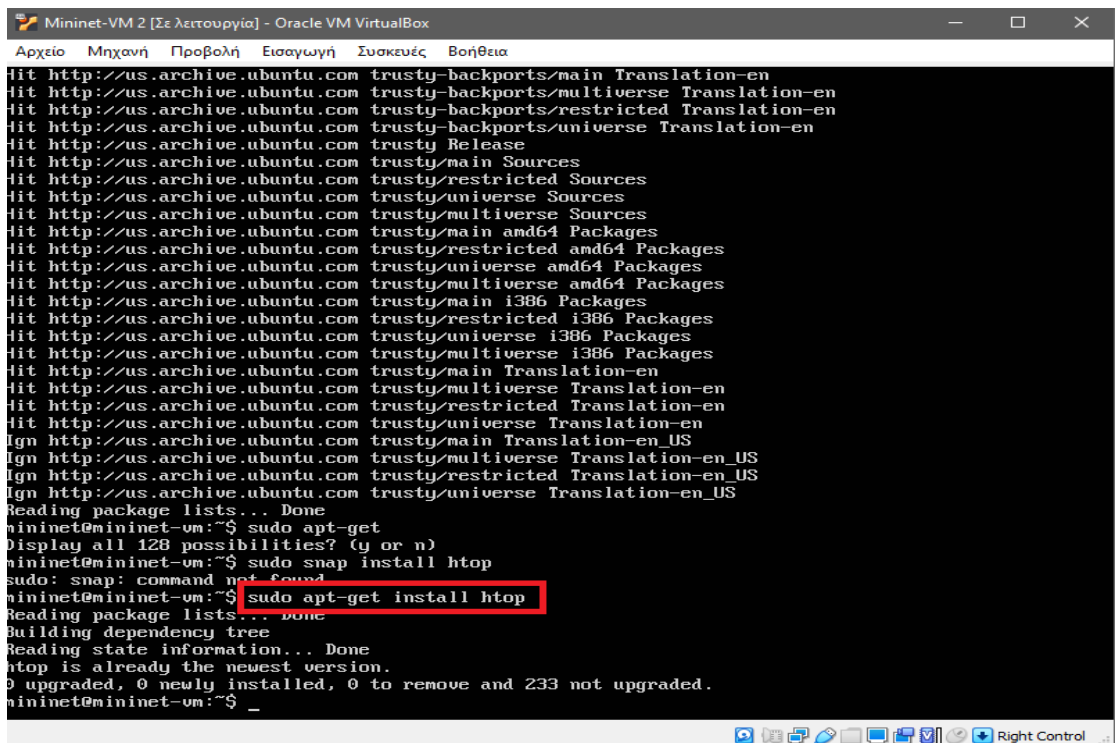
Επιπρόσθετα εγκαθίσταται η βιβλιοθήκη «numrg» που αποτελεί τον πυρήνα λειτουργιών της PYTHON με τη χρήση της εντολής «sudo apt-get install python-numrg».



```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
hit http://us.archive.ubuntu.com trusty-backports/main amd64 Packages
hit http://us.archive.ubuntu.com trusty-backports/restricted amd64 Packages
hit http://us.archive.ubuntu.com trusty-backports/universe amd64 Packages
hit http://us.archive.ubuntu.com trusty-backports/multiverse amd64 Packages
hit http://us.archive.ubuntu.com trusty-backports/main i386 Packages
hit http://us.archive.ubuntu.com trusty-backports/restricted i386 Packages
hit http://us.archive.ubuntu.com trusty-backports/universe i386 Packages
hit http://us.archive.ubuntu.com trusty-backports/multiverse i386 Packages
hit http://us.archive.ubuntu.com trusty-backports/main Translation-en
hit http://us.archive.ubuntu.com trusty-backports/multiverse Translation-en
hit http://us.archive.ubuntu.com trusty-backports/restricted Translation-en
hit http://us.archive.ubuntu.com trusty-backports/universe Translation-en
hit http://us.archive.ubuntu.com trusty Release
hit http://us.archive.ubuntu.com trusty/main Sources
hit http://us.archive.ubuntu.com trusty/restricted Sources
hit http://us.archive.ubuntu.com trusty/universe Sources
hit http://us.archive.ubuntu.com trusty/multiverse Sources
hit http://us.archive.ubuntu.com trusty/main amd64 Packages
hit http://us.archive.ubuntu.com trusty/restricted amd64 Packages
hit http://us.archive.ubuntu.com trusty/universe amd64 Packages
hit http://us.archive.ubuntu.com trusty/multiverse amd64 Packages
hit http://us.archive.ubuntu.com trusty/main i386 Packages
hit http://us.archive.ubuntu.com trusty/restricted i386 Packages
hit http://us.archive.ubuntu.com trusty/universe i386 Packages
hit http://us.archive.ubuntu.com trusty/multiverse i386 Packages
hit http://us.archive.ubuntu.com trusty/main Translation-en
hit http://us.archive.ubuntu.com trusty/multiverse Translation-en
hit http://us.archive.ubuntu.com trusty/restricted Translation-en
hit http://us.archive.ubuntu.com trusty/universe Translation-en
Ign http://us.archive.ubuntu.com trusty/main Translation-en_US
Ign http://us.archive.ubuntu.com trusty/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com trusty/restricted Translation-en_US
Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US
Reading package lists... Done
mininet@mininet-vm:~$ sudo apt-get
Display all 128 possibilities? (y or n)
mininet@mininet-vm:~$ sudo apt-get install python-numpy
```

Εικόνα 5.14:Εγκατάσταση βιβλιοθήκης numpy.

Τέλος εγκαθιστούμε τη λειτουργία htop που είναι ο βασικός μετρητής επιδόσεων σε Linux-distributions όπως είναι και το Mininet, με τη χρήση της εντολής «sudo apt-get install htop».



```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
hit http://us.archive.ubuntu.com trusty-backports/main Translation-en
hit http://us.archive.ubuntu.com trusty-backports/multiverse Translation-en
hit http://us.archive.ubuntu.com trusty-backports/restricted Translation-en
hit http://us.archive.ubuntu.com trusty-backports/universe Translation-en
hit http://us.archive.ubuntu.com trusty Release
hit http://us.archive.ubuntu.com trusty/main Sources
hit http://us.archive.ubuntu.com trusty/restricted Sources
hit http://us.archive.ubuntu.com trusty/universe Sources
hit http://us.archive.ubuntu.com trusty/multiverse Sources
hit http://us.archive.ubuntu.com trusty/main amd64 Packages
hit http://us.archive.ubuntu.com trusty/restricted amd64 Packages
hit http://us.archive.ubuntu.com trusty/universe amd64 Packages
hit http://us.archive.ubuntu.com trusty/multiverse amd64 Packages
hit http://us.archive.ubuntu.com trusty/main i386 Packages
hit http://us.archive.ubuntu.com trusty/restricted i386 Packages
hit http://us.archive.ubuntu.com trusty/universe i386 Packages
hit http://us.archive.ubuntu.com trusty/multiverse i386 Packages
hit http://us.archive.ubuntu.com trusty/main Translation-en
hit http://us.archive.ubuntu.com trusty/multiverse Translation-en
hit http://us.archive.ubuntu.com trusty/restricted Translation-en
hit http://us.archive.ubuntu.com trusty/universe Translation-en
Ign http://us.archive.ubuntu.com trusty/main Translation-en_US
Ign http://us.archive.ubuntu.com trusty/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com trusty/restricted Translation-en_US
Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US
Reading package lists... Done
mininet@mininet-vm:~$ sudo apt-get
Display all 128 possibilities? (y or n)
mininet@mininet-vm:~$ sudo snap install htop
sudo: snap: command not found
mininet@mininet-vm:~$ sudo apt-get install htop
Reading package lists... Done
Building dependency tree
Reading state information... Done
htop is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 233 not upgraded.
mininet@mininet-vm:~$ _
```

Εικόνα 5.15:Εγκατάσταση λειτουργίας htop.

Το βασικό περιβάλλον της εφαρμογής htop μέσα από το οποίο θα αντληθούν σημαντικές πληροφορίες επίδοσης και απόδοσης της εικονικής μηχανής και τον εξομοιώσεων που θα ακολουθήσουν.

```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσσκευές Βοήθεια

 1 [ | 0.7%]
 2 [ | 0.0%]
 3 [ | 0.6%]
 4 [ | 0.6%]
 5 [ | 0.0%]
 6 [ | 0.0%]
 7 [ | 0.0%]
 8 [ | 0.0%]
Mem| | | 125/16047MB]
Swp:1021M used:0K

Load average: 0.03 0.01 0.01
Uptime: 00:01:08

PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%  TIME+  Command
1318 mininet    20   0 26172  3692  2972 R  0.6  0.0  0:00.03 htop
  1 root       20   0  33476  3988  2680 S  0.6  0.0  0:02.22 /sbin/init
 611 messagebu 20   0  39128  2092  1744 S  0.6  0.0  0:00.11 dbus-daemon --system --fork
 734 root       20   0  15424  1836  1436 S  0.6  0.0  0:00.06 upstart-file-bridge --daemon
1231 root       10  -10 20956   424   48 S  0.0  0.0  0:00.07 ovsdb-server: monitoring pid 1232 (he
1264 root       10  -10 21464   436   48 S  0.0  0.0  0:00.07 ovs-vsitchd: monitoring pid 1265 (he
 385 root       20   0  19624  1940  1668 S  0.0  0.0  0:00.21 upstart-udev-bridge --daemon
 391 root       20   0  51780  3544  2840 S  0.0  0.0  0:00.10 /lib/systemd/systemd-udev --daemon
 644 root       20   0  43464  3352  2996 S  0.0  0.0  0:00.15 /lib/systemd/systemd-logind
 649 syslog    20   0  249M  3060  2376 S  0.0  0.0  0:00.00 rsyslogd
 650 syslog    20   0  249M  3060  2376 S  0.0  0.0  0:00.00 rsyslogd
 651 syslog    20   0  249M  3060  2376 S  0.0  0.0  0:00.00 rsyslogd
 648 syslog    20   0  249M  3060  2376 S  0.0  0.0  0:00.01 rsyslogd
 733 root       20   0  15276   216   0 S  0.0  0.0  0:00.07 upstart-socket-bridge --daemon
 749 root       20   0  10244   3180  888 S  0.0  0.0  0:00.00 dhclient -1 -v -pf /run/dhclient.eth0
 970 root       20   0  16072   2144  1992 S  0.0  0.0  0:00.00 /sbin/getty -8 38400 tty4
 972 root       20   0  16072   2152  2000 S  0.0  0.0  0:00.00 /sbin/getty -8 38400 tty5
 975 root       20   0  16072   2108  1952 S  0.0  0.0  0:00.00 /sbin/getty -8 38400 tty2
 976 root       20   0  16072   2084  1924 S  0.0  0.0  0:00.00 /sbin/getty -8 38400 tty3
 978 root       20   0  16072   2052  1900 S  0.0  0.0  0:00.00 /sbin/getty -8 38400 tty6
 993 root       20   0  61396  5420  4736 S  0.0  0.0  0:00.00 /usr/sbin/sshd -D
1000 root       20   0  23668  2224  2016 S  0.0  0.0  0:00.00 cron
1010 root       20   0  19208  2192  1940 S  0.0  0.0  0:00.00 /usr/sbin/irqbalance

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit
```

Εικόνα 5.16: Ενδεικτική λειτουργία της οπτικής διεργασίας htop.

Το htop παρέχει τη δυνατότητα παρακολούθησης:

- Χρήσης CPU στιγμιαία & μέση
- Χρήσης RAM
- Αναλυτική πληροφόρηση για εκτελούμενες διεργασίες που πέρα από τις παραπάνω λειτουργίες περιλαμβάνει:

παραπάνω λειτουργίες περιλαμβάνει:

- Χρόνος εκτελέσεως
- ID
- Χρήστης
- Προτεραιότητα



### 5.3.3 Ρύθμιση X-Ming

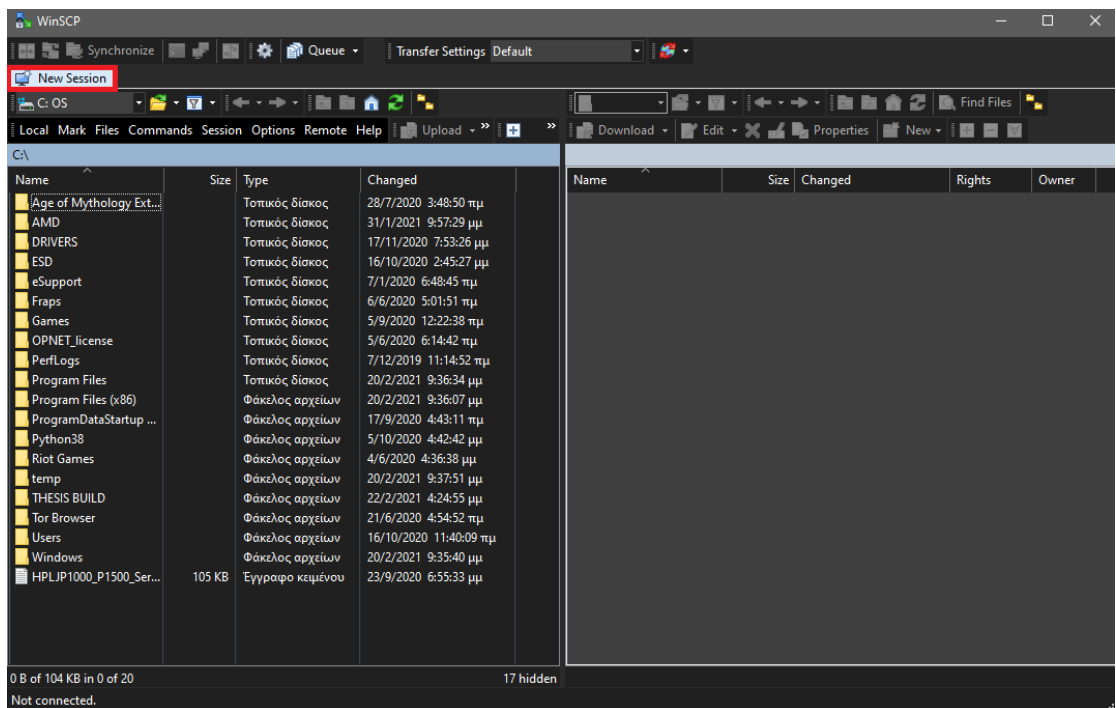
Το πρόγραμμα X-Ming δε χρειάζεται κάποιο είδος παραμετροποίησης καθώς το μόνο που χρειάζεται είναι να μένει ανοιχτό στο παρασκήνιο.



Εικόνα 5.17:Λειτουργία X-Ming στο παρασκήνιο.

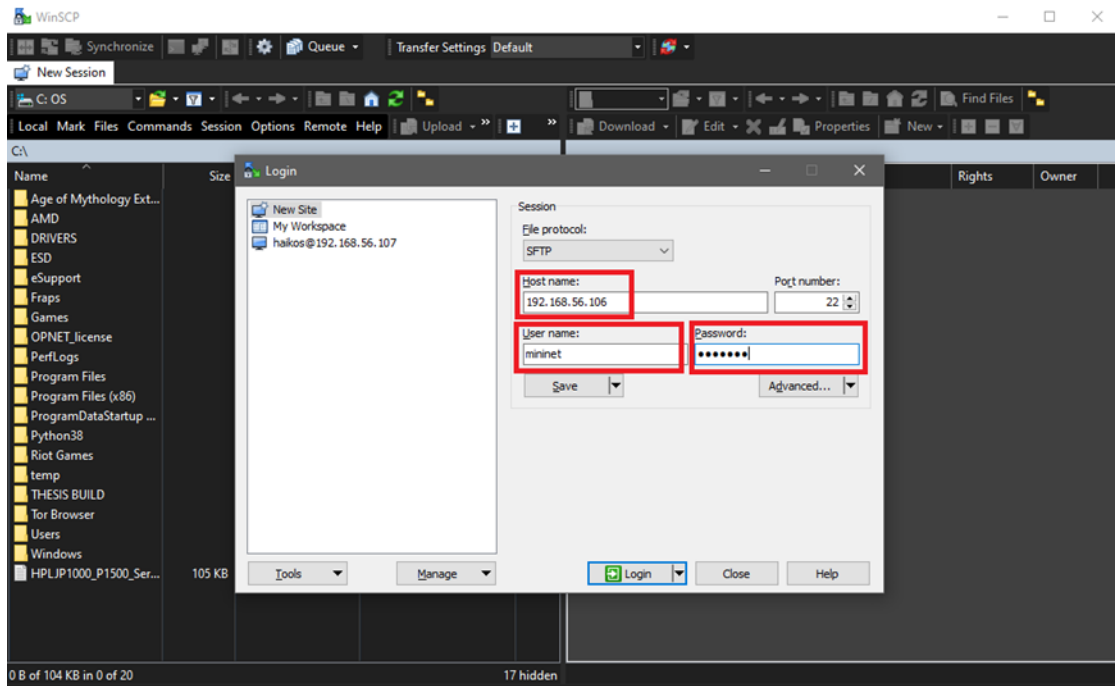
### 5.3.4 Ρύθμιση WinSCP

Ανοίγουμε το πρόγραμμα WinSCP και επιλέγουμε «New Session» για να δημιουργήσουμε μια νέα σύνδεση.

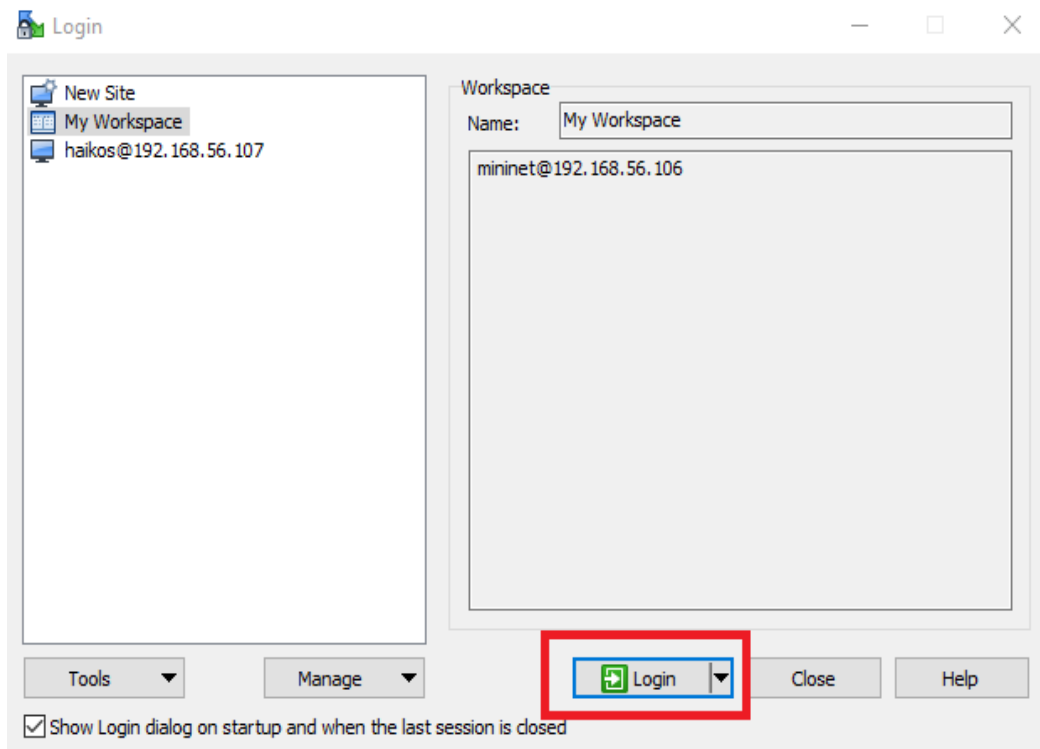


Εικόνα 5.18:Δημιουργία νέας σύνδεσης μέσω WinSCP.

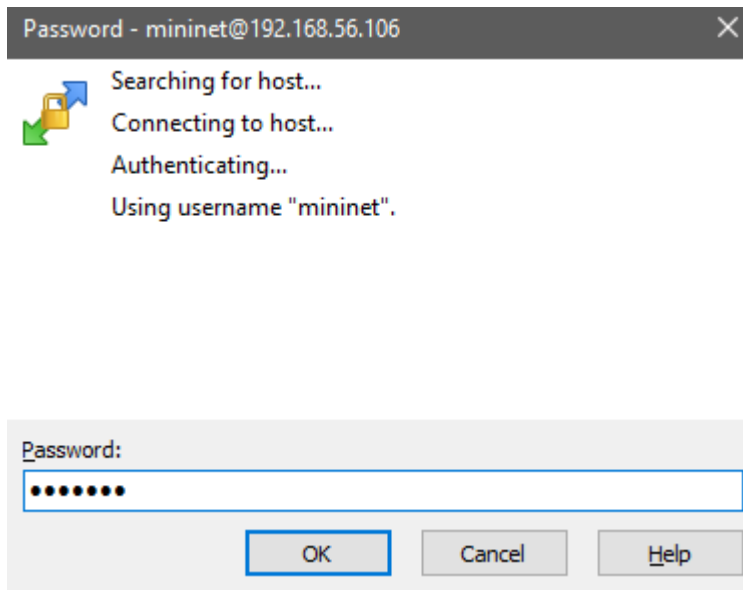
Γράφουμε IP διεύθυνση του Mininet που βρήκαμε στο κεφάλαιο 5.3.2 η οποία είναι η «192.168.56.106» και χρησιμοποιούμε ως username και password τη λέξη «mininet» όπως αναφέρθηκε και στο Κεφάλαιο 4.8.2.1 και έτσι αποθηκεύεται στα Workspaces.Συνδεόμαστε στην διεύθυνση IP του Mininet με τα παρακάτω στοιχεία.



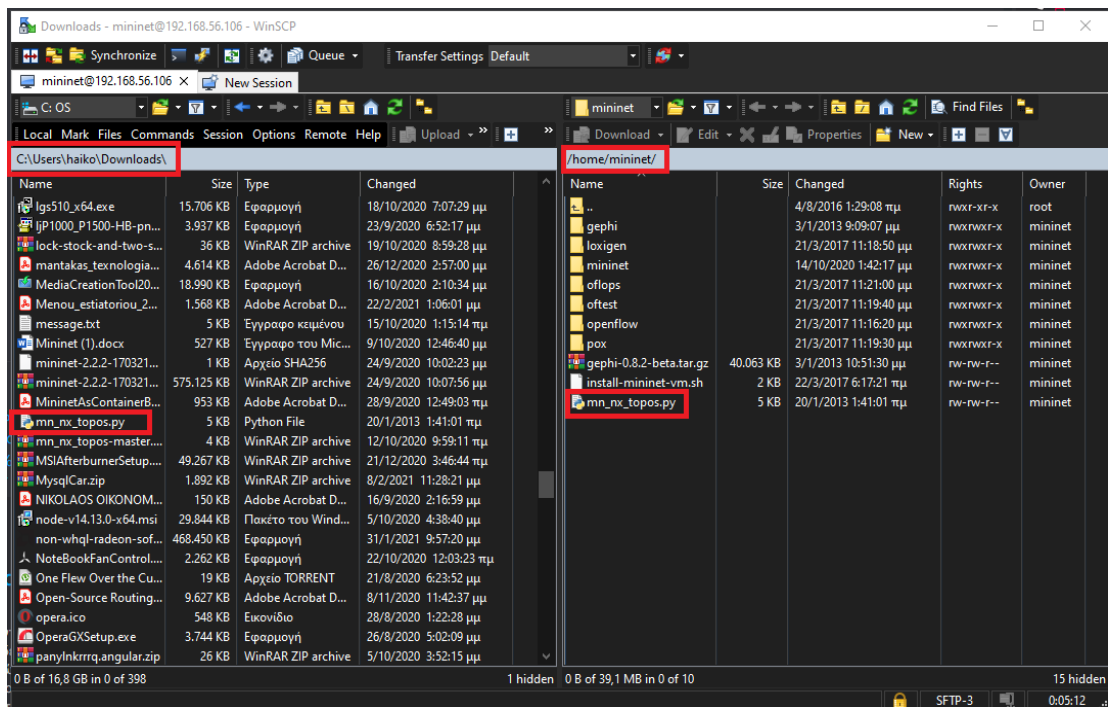
Εικόνα 5.19: Δημιουργία σύνδεσης μεταξύ WinSCP και Mininet.



Εικόνα 5.20: Σύνδεση WinSCP και Mininet αφού αποθηκεύτηκε στη μνήμη.



Εικόνα 5.21:Αυθεντικοποίηση σύνδεσης.

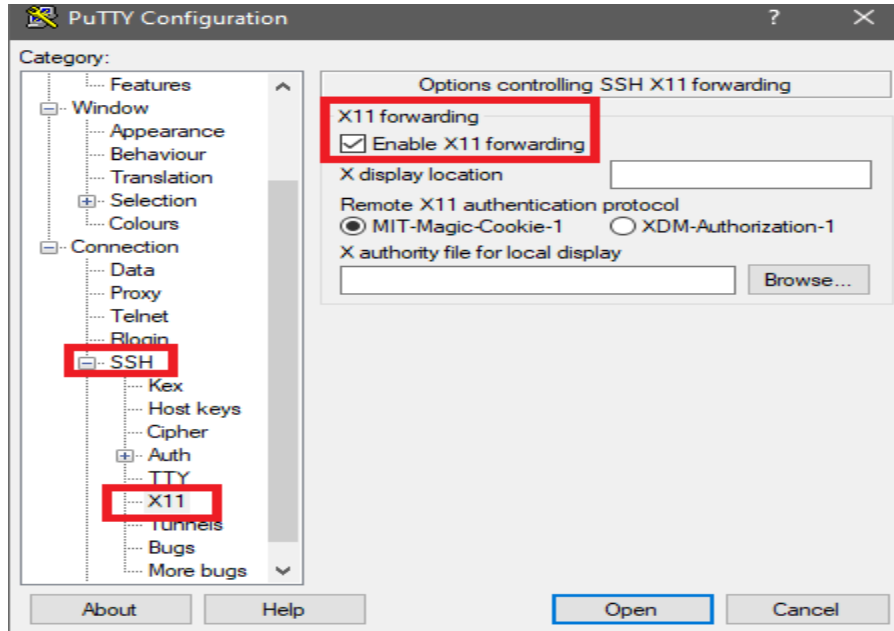


Εικόνα 5.22:Μεταφορά αρχείου φυσικού συστήματος στο εικονικό σύστημα μέσω WinSCP.

Το αρχείο «mn\_nx\_topos.py» περιέχει τον εξής κώδικα που επιτρέπει τη δημιουργία custom τοπολογιών [63].

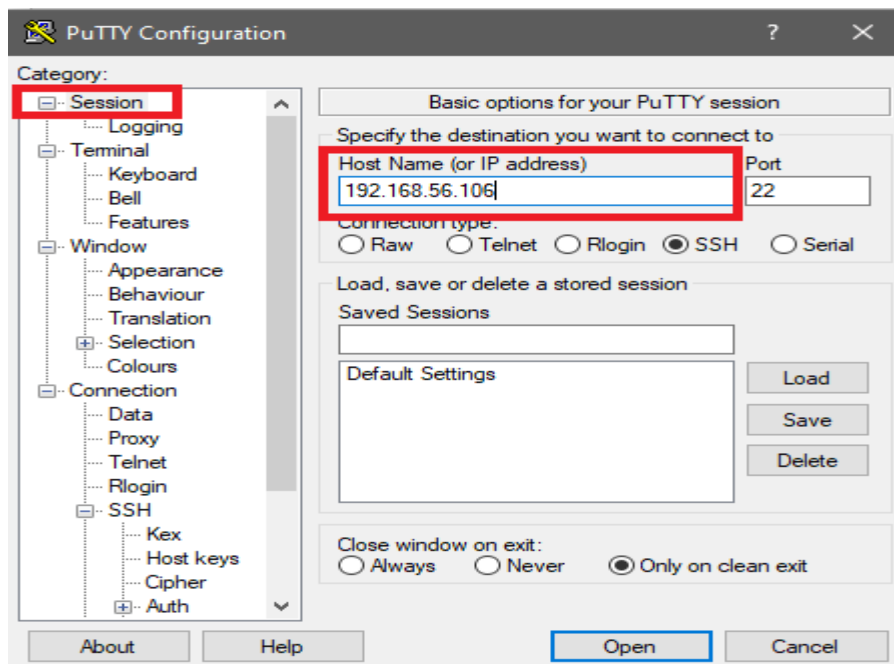
### 5.3.5 Ρύθμιση PuTTY

Ανοίγοντας το πρόγραμμα PuTTY επιλέγουμε το υπομενού «SSH» και στο υπομενού «X11» ενεργοποιούμε τη λειτουργία «X11 forwarding».



Εικόνα 5.23:Ενεργοποίηση λειτουργίας X11 forwarding.

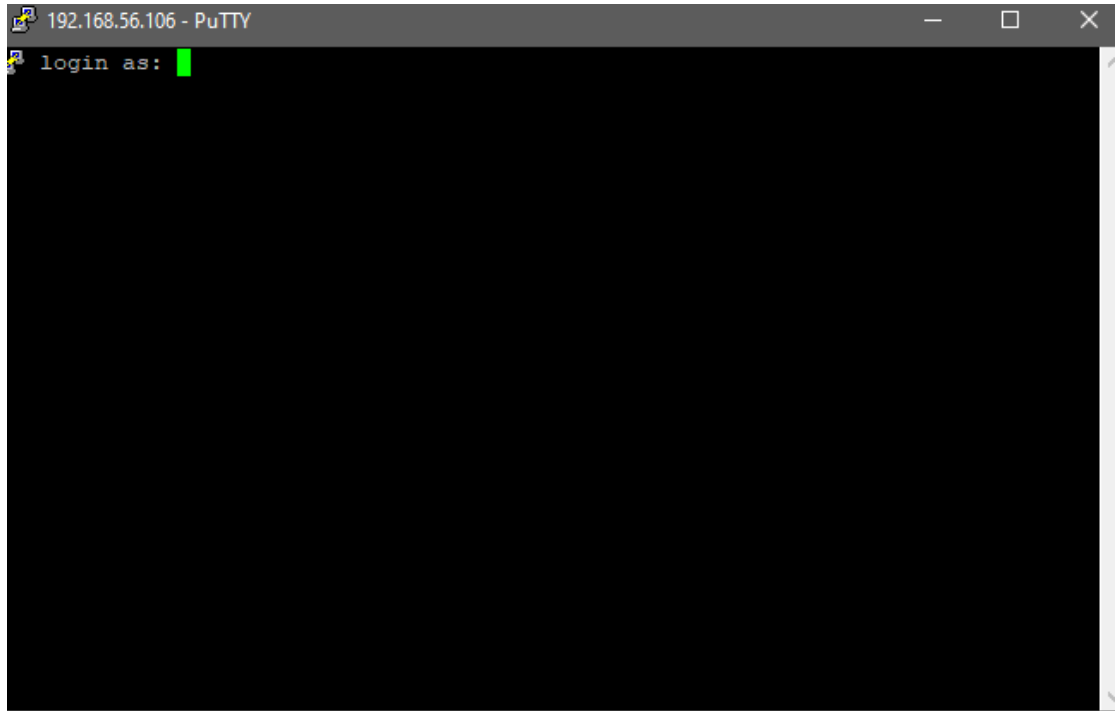
Επιστρέφουμε στο αρχικό μενού του PuTTY στο υπομενού «session» και πληκτρολογούμε την IP ADDRESS «192.168.56.106» του Mininet που έχουμε βρει από το Κεφάλαιο 5.3.2.



Εικόνα 5.24:Εισαγωγή διεύθυνσης IP κάρτας δικτύου Mininet στο PuTTY.

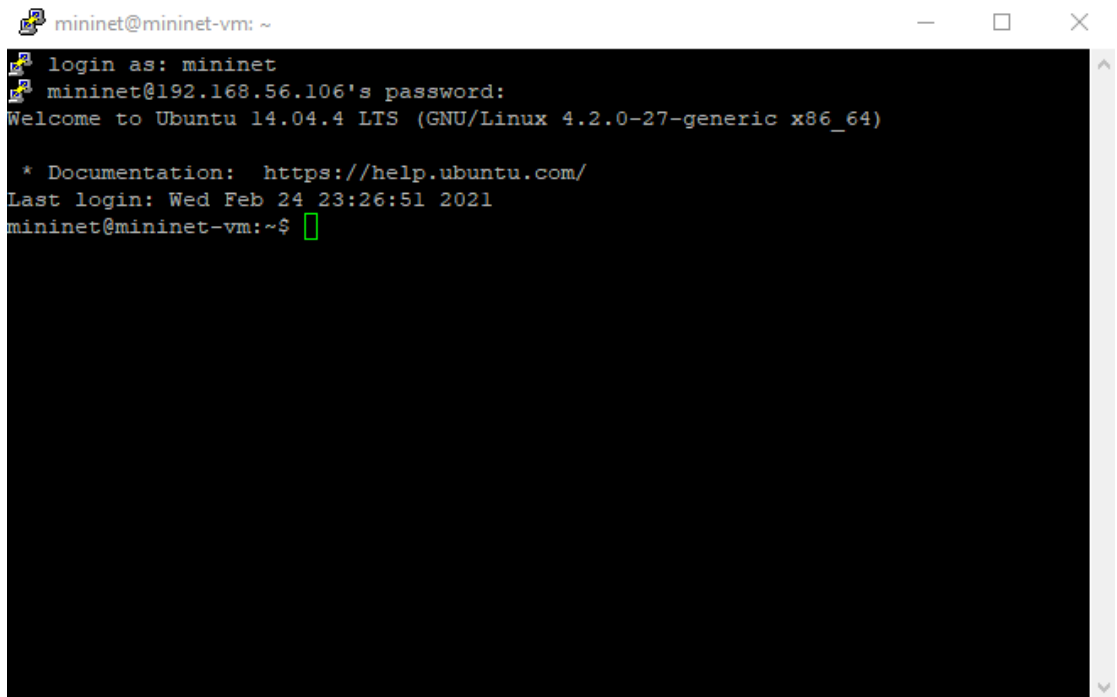
Πραγματοποιούμε τη σύνδεση με τα γνωστά username & password από το κεφάλαιο

4.8.2.1.



Εικόνα 5.25:Εισαγωγή διαπιστευτηρίων Mininet στο επιπλέον τερματικό του PuTTY.

Και έχει ανοίξει με επιτυχία ένα επιπλέον τερματικό Mininet.



Εικόνα 5.26:Επιπλέον τερματικό Mininet μέσω Putty.

Έχοντας πλέον το πλεονέκτημα των πολλαπλών τερματικών μπορούμε να εκτελέσουμε ταυτόχρονα πολλαπλές διεργασίες όπως θα φανεί και στην εξομίωση προκειμένου να αντληθούν όσο το δυνατόν περισσότερα αποτελέσματα. Ενδεικτικά στο παρακάτω παράδειγμα υπάρχει παράλληλη προβολή και του Mininet και μέτρησης χρήσης συστήματος με την εντολή «htop».

```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
Hit http://us.archive.ubuntu.com trusty-backports/universe Sources
Hit http://us.archive.ubuntu.com trusty-backports/multiverse Sources
Hit http://us.archive.ubuntu.com trusty-backports/main amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/restricted amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/universe amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/multiverse amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/main i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/restricted i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/universe i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/multiverse i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/main Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/restricted Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/universe Translation-en
Hit http://us.archive.ubuntu.com trusty Release
Hit http://us.archive.ubuntu.com trusty/main Sources
Hit http://us.archive.ubuntu.com trusty/restricted Sources
Hit http://us.archive.ubuntu.com trusty/universe Sources
Hit http://us.archive.ubuntu.com trusty/multiverse Sources
Hit http://us.archive.ubuntu.com trusty/main amd64 Packages
Hit http://us.archive.ubuntu.com trusty/restricted amd64 Packages
Hit http://us.archive.ubuntu.com trusty/universe amd64 Packages
Hit http://us.archive.ubuntu.com trusty/multiverse amd64 Packages
Hit http://us.archive.ubuntu.com trusty/main i386 Packages
Hit http://us.archive.ubuntu.com trusty/restricted i386 Packages
Hit http://us.archive.ubuntu.com trusty/universe i386 Packages
Hit http://us.archive.ubuntu.com trusty/multiverse i386 Packages
Hit http://us.archive.ubuntu.com trusty/main Translation-en
Hit http://us.archive.ubuntu.com trusty/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty/restricted Translation-en
Hit http://us.archive.ubuntu.com trusty/universe Translation-en
Ign http://us.archive.ubuntu.com trusty/main Translation-en_US
Ign http://us.archive.ubuntu.com trusty/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com trusty/restricted Translation-en_US
Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US
Reading package lists... Done
mininet@mininet-vm:~$
```

```
mininet@mininet-vm: ~
 1  [          0.0%]
 2  [          0.0%]
 3  [          0.0%]
 4  [          0.0%]  Load average: 0.01 0.03 0.05
 5  [          0.0%]  Uptime: 02:01:03
 6  [          0.0%]
 7  [          0.0%]
 8  [          0.0%]
Mem[|||||] 134/16047MB
Swp:1021M used:0K

```

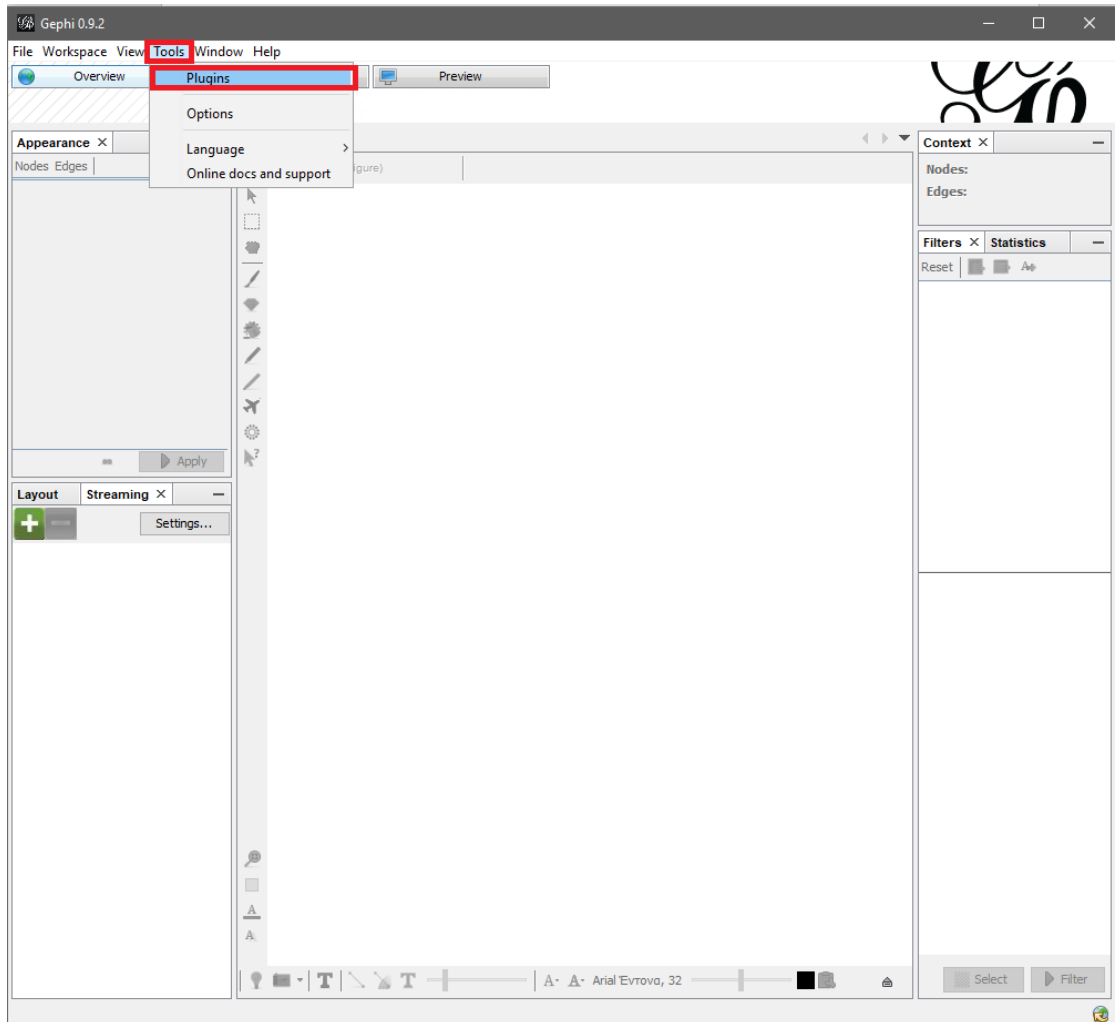
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
734	root	20	0	15424	1836	1436	S	0.7	0.0	0:00.91	upstart-file-bridge --daemon
3209	mininet	20	0	26024	3504	2916	R	0.0	0.0	0:00.12	htop
1231	root	10	-10	20956	424	48	S	0.0	0.0	0:09.25	ovsdb-server: monitoring pid 1232
1	root	20	0	33476	3988	2680	S	0.0	0.0	0:04.75	/sbin/init
385	root	20	0	19624	1940	1668	S	0.0	0.0	0:01.10	upstart-udev-bridge --daemon
391	root	20	0	51780	3544	2840	S	0.0	0.0	0:00.10	/lib/systemd/systemd-udev --daemon

```
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice + F9Kill F10Quit
```

Εικόνα 5.27: Εκτέλεση Mininet στο βασικό τερματικό (πάνω), εκτέλεση htop στο δευτερεύον τερματικό μέσω PuTTY (κάτω).

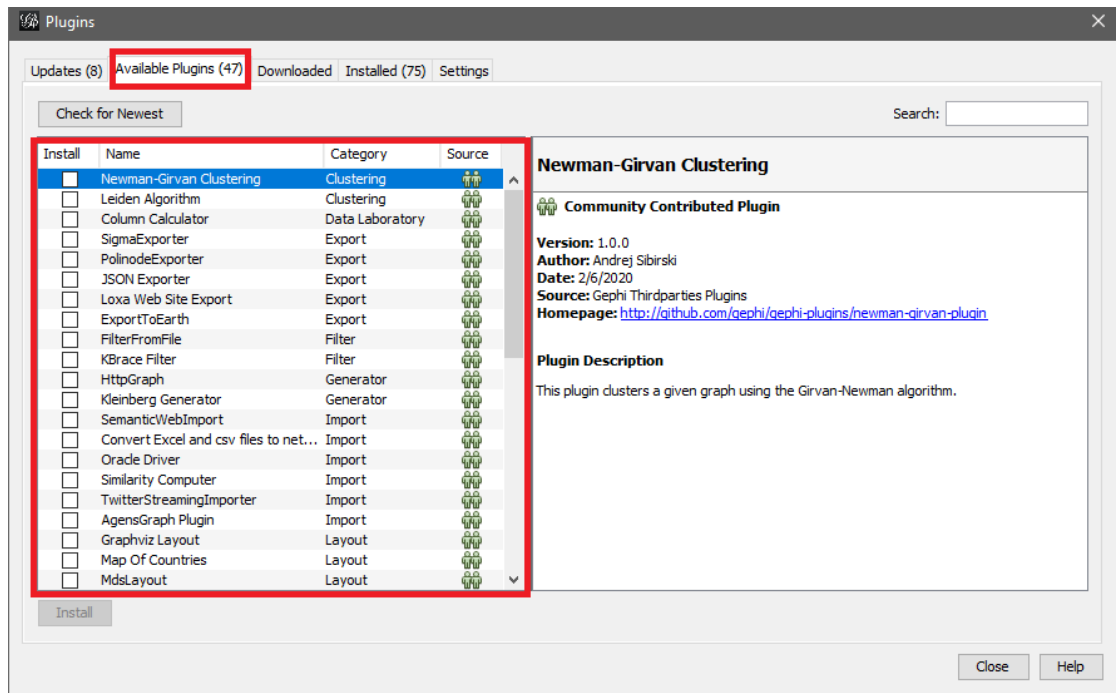
### 5.3.6 Ρύθμιση Gephi

Εκκινώντας το Gephi στο βασικό μενού του επιλέγουμε «Tools» και στη συνέχεια «Plugins»

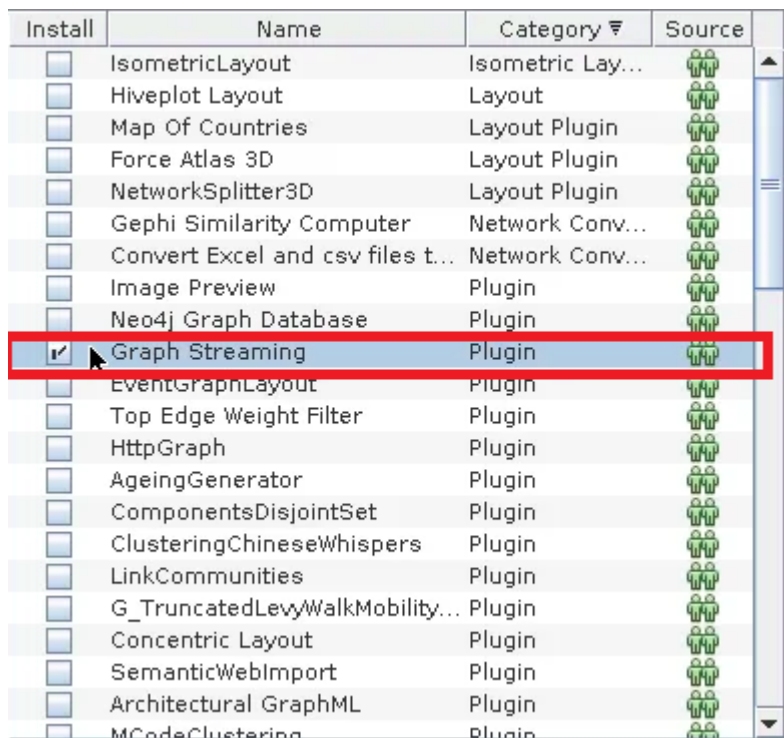


Εικόνα 5.28:Επιλογή Tools και στη συνέχεια Plugins στο Gephi.

Επιλέγουμε την ενότητα «Available Plugins» και ψάχνουμε στη λίστα για το πρόσθετο(Plugin) «Graph Streaming» και το εγκαθιστούμε καθότι θα μας επιτρέψει να έχουμε γραφική απεικόνιση των τοπολογιών-δικτύων που θα τρέξουν στην εξομοίωση.



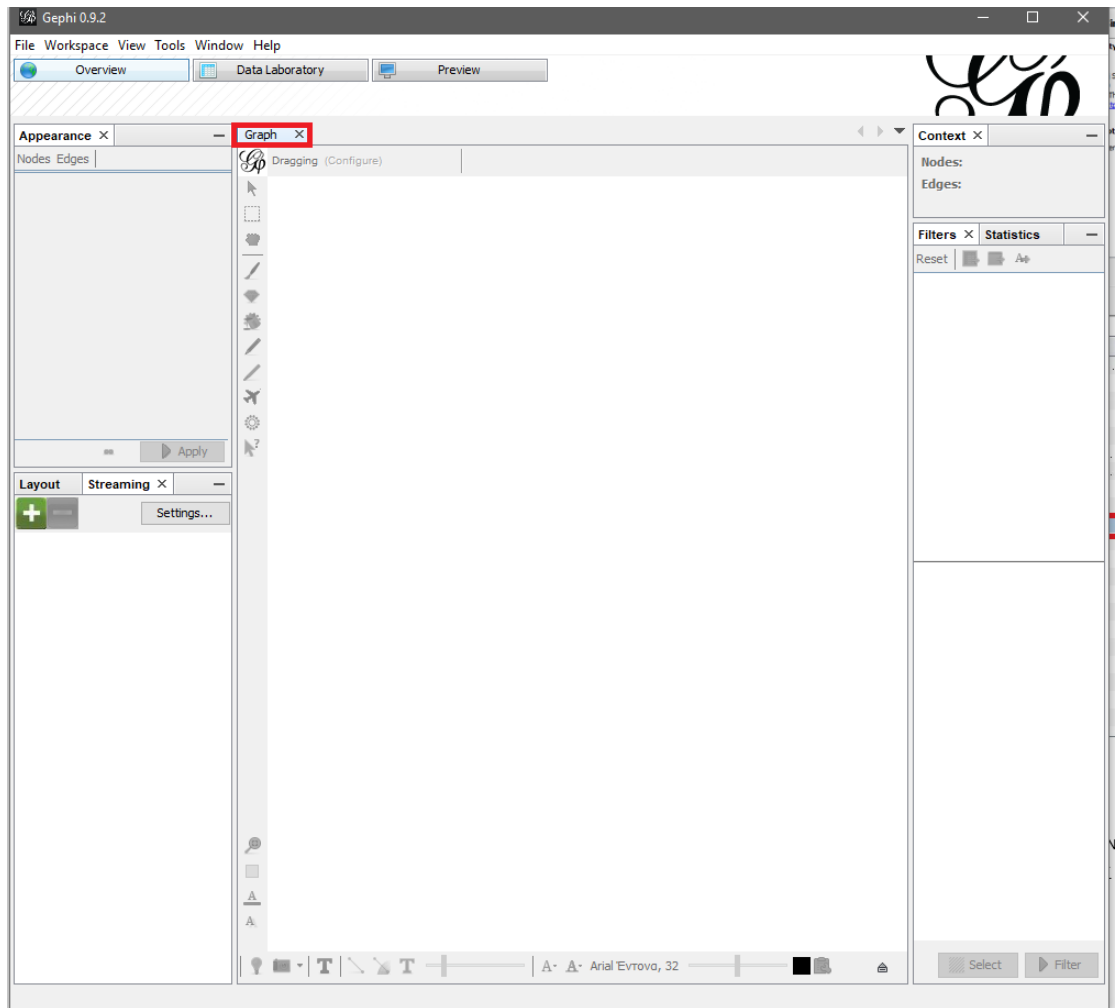
Εικόνα 5.29: Λίστα διαθέσιμων πρόσθετων.



Εικόνα 5.30: Πρόσθετο Graph Streaming.

Στη συνέχεια εμφανίζεται στο αρχικό μενού το πρόσθετο «Graph Streaming» ως «Graph».





Εικόνα 5.31: Παράθυρο λειτουργίας Graph streaming.

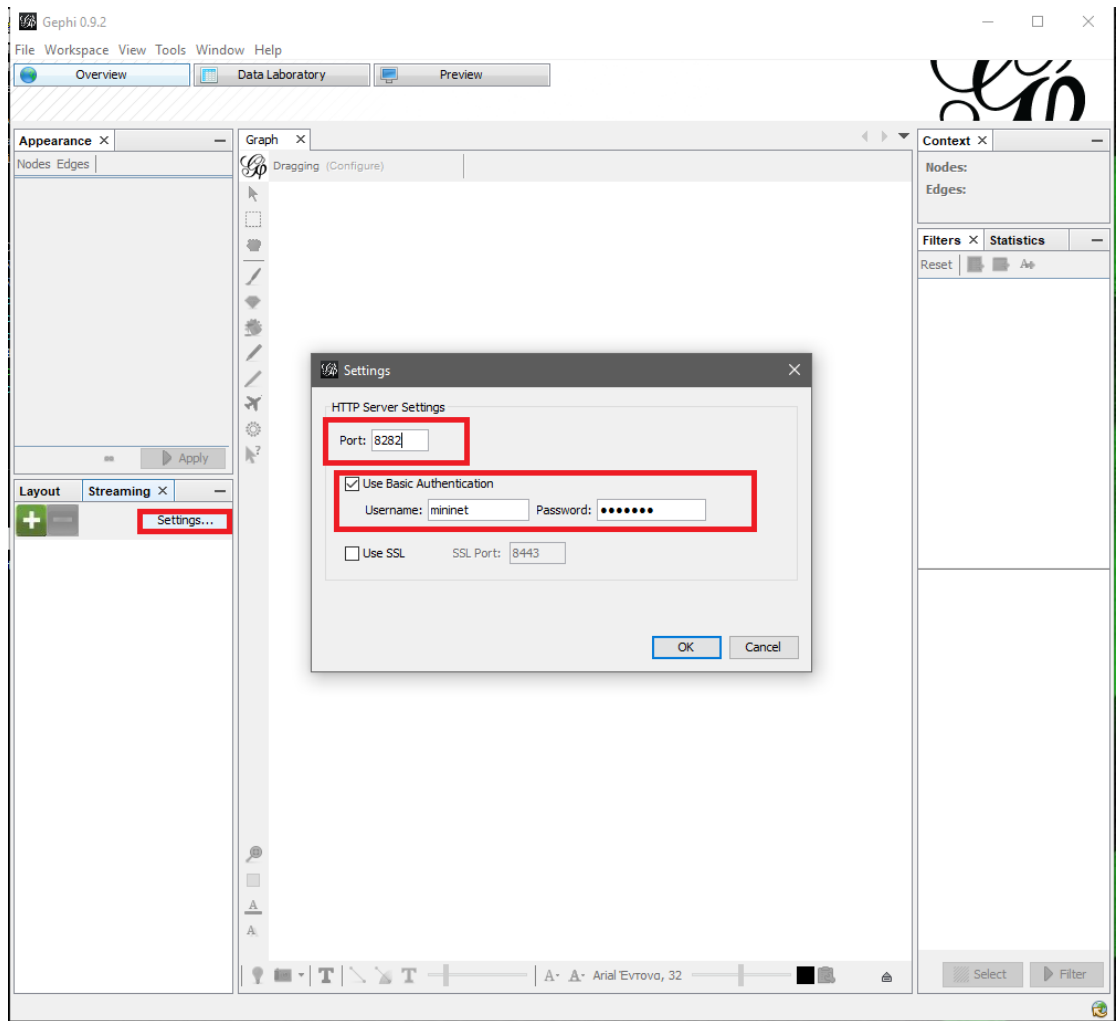
Έχοντας ανοιχτό το Mininet όπως είδαμε στο κεφάλαιο 5.3.2 ανοίγουμε ένα επιπλέον τερματικό μέσω του PuTTY όπως είδαμε στο κεφάλαιο 5.3.5 και πάντα σε συνδυασμό με την εφαρμογή X-Ming να τρέχει στο παρασκήνιο όπως είδαμε στο κεφάλαιο 5.3.3. Επειδή θα χρησιμοποιηθεί ο ελεγκτής SDN POX με το πρωτόκολλο Openflow ο οποίος υπάρχει προεγκατεστημένος στο Mininet ,για να ενεργοποιηθεί και να γίνει απεικόνιση των τοπολογιών-δικτύων πληκτρολογούμε την ακόλουθη εντολή:

```
« sudo ~/pox/pox.py forwarding.l2_learning \  
openflow.discovery misc.gephi_topo \  
openflow.spanning_tree --no-flood --hold-down \  
host_tracker info.packet_dump \  
samples.pretty_log log.level -DEBUG »
```

```
mininet@mininet-vm: ~  
login as: mininet  
mininet@192.168.56.106's password:  
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
Last login: Wed Mar 17 02:55:05 2021  
mininet@mininet-vm:~$ sudo ~/pox/pox.py forwarding.l2_learning \  
> openflow.discovery misc.gephi_topo \  
> openflow.spanning_tree --no-flood --hold-down \  
> host_tracker info.packet_dump \  
> samples.pretty_log log.level --DEBUG  
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et al.  
INFO:host_tracker:host_tracker ready  
INFO:info.packet_dump:Packet dumper running  
[core                ] POX 0.3.0 (dart) going up...  
[core                ] Running on CPython (2.7.6/Oct 26 2016 20:30:19)  
[core                ] Platform is Linux-4.2.0-27-generic-x86_64-with-Ubuntu-  
14.04-trusty  
[core                ] POX 0.3.0 (dart) is up.  
[openflow.of_01     ] Listening on 0.0.0.0:6633
```

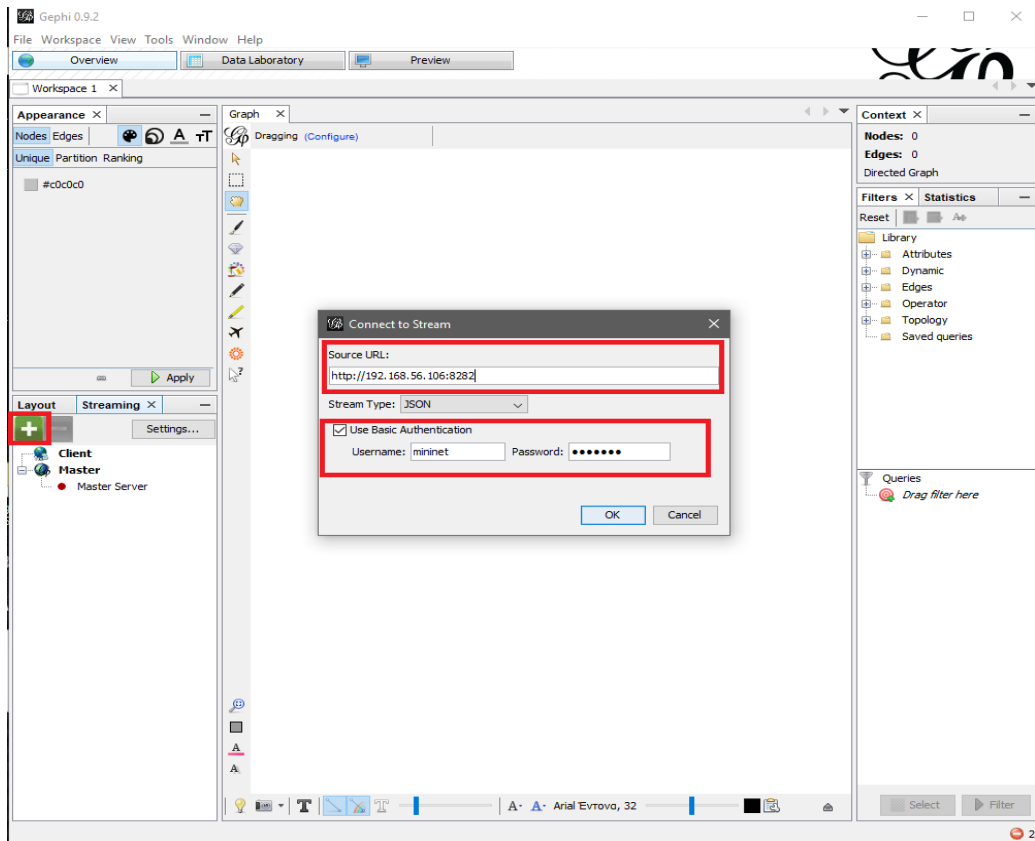
Εικόνα 5.32:Εκτέλεση της εντολής ενεργοποίησης του ελεγκτή POX και της υποστήριξης πρωτοκόλλου Openflow και εμφάνιση επιβεβαίωσης.

Πίσω στο Gεphi επιλέγονται οι ρυθμίσεις «settings» και αφήνουμε την προεπιλεγμένη θύρα επικοινωνίας 8282 και ενεργοποιούμε την επιλογή «Basic Authentication» έτσι ώστε να εισάγουμε τα διαπιστευτήρια σύνδεσης στο Mininet όπως αναφέρθηκαν στο κεφάλαιο 4.8.2.1

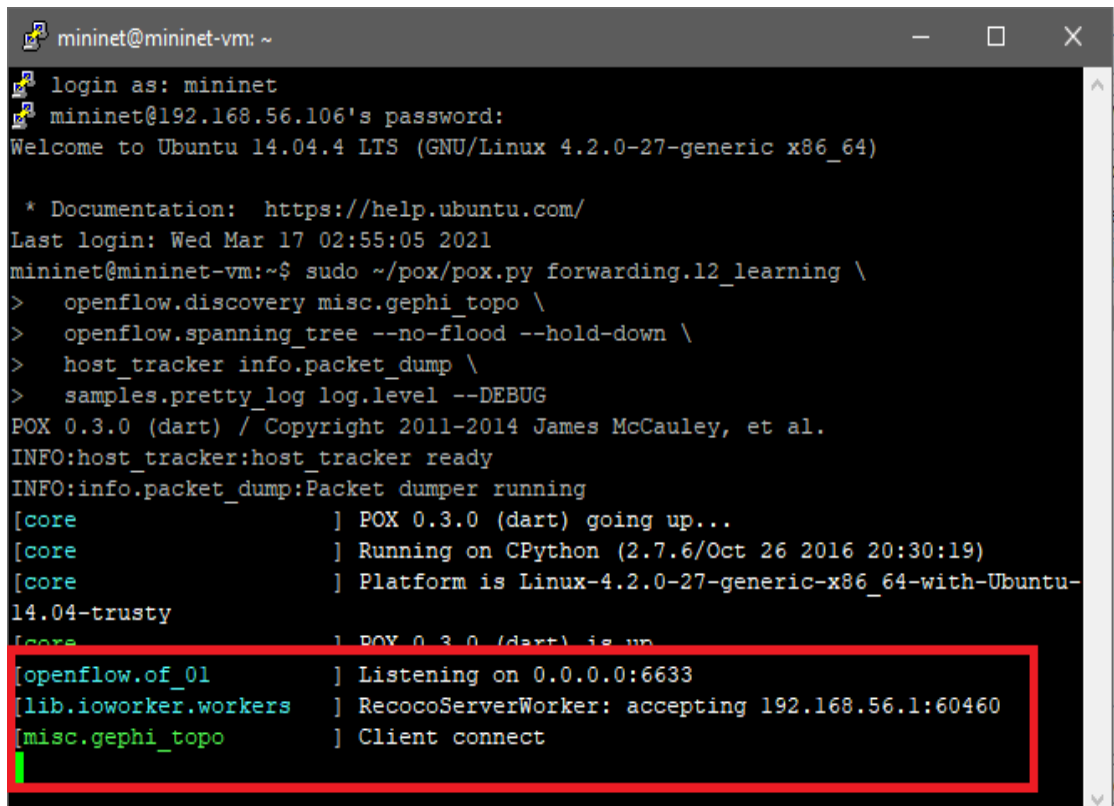


Εικόνα 5.33: Δημιουργία ρυθμίσεων σύνδεσης μεταξύ Mininet και Gephi.

Επιλέγουμε το σύμβολο «+» για να προσθέσουμε μια σύνδεση μεταξύ Gephi και Mininet. Φροντίζουμε ώστε το URL σύνδεσης να είναι της μορφής «http://localhost:HttpServerSettings Port» και δίνουμε τα ίδια διαπιστευτήρια για την πραγματοποίηση σύνδεσης στο Mininet. Με την επιτυχή σύνδεση θα εμφανιστεί αντίστοιχο μήνυμα επιβεβαίωσης στο επιπλέον τερματικό που ανοίχτηκε προηγουμένως μέσω PuTTY.

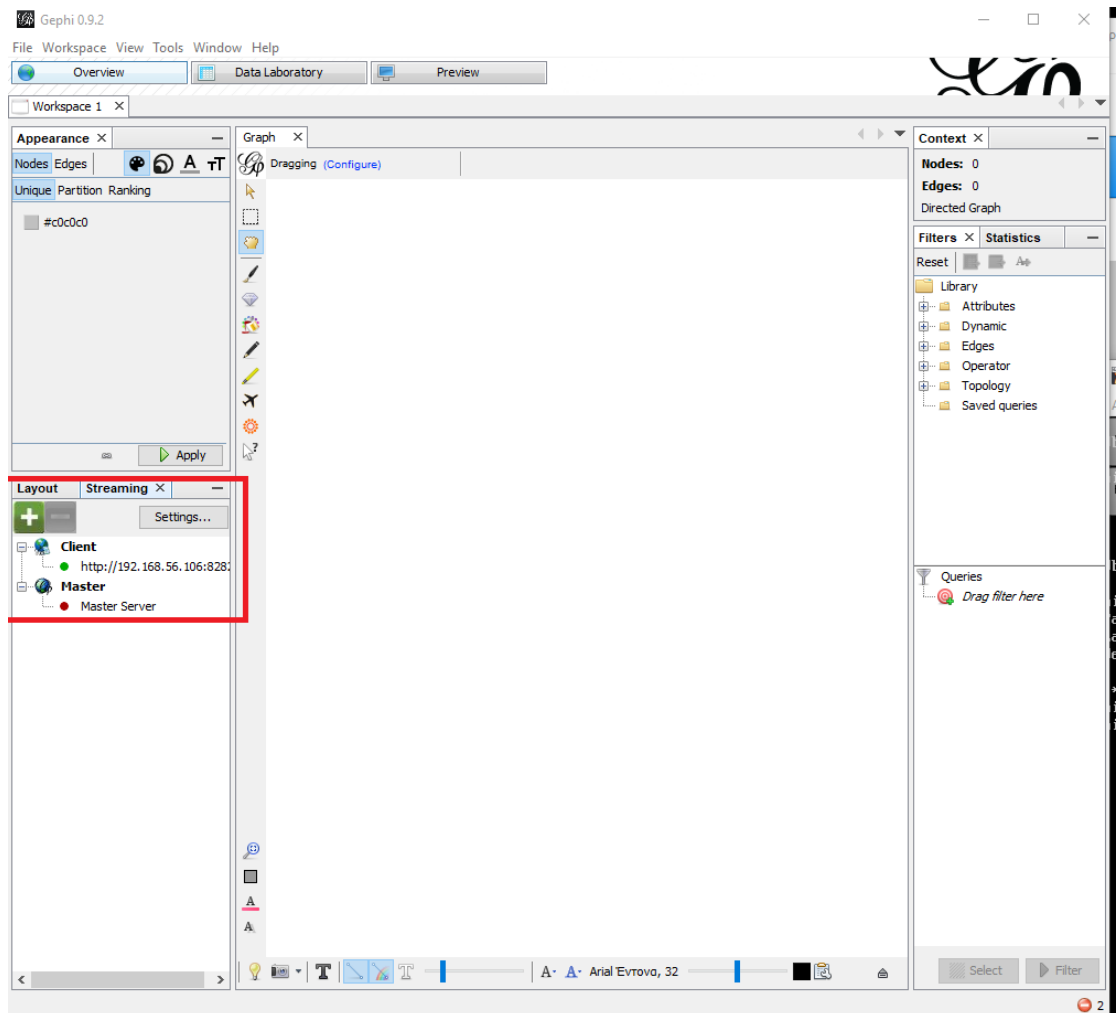


Εικόνα 5.34: Προσθήκη σύνδεσης μεταξύ Mininet και Gephi.



Εικόνα 5.35: Επιβεβαίωση επιτυχούς σύνδεσης μεταξύ Mininet και Gephi.

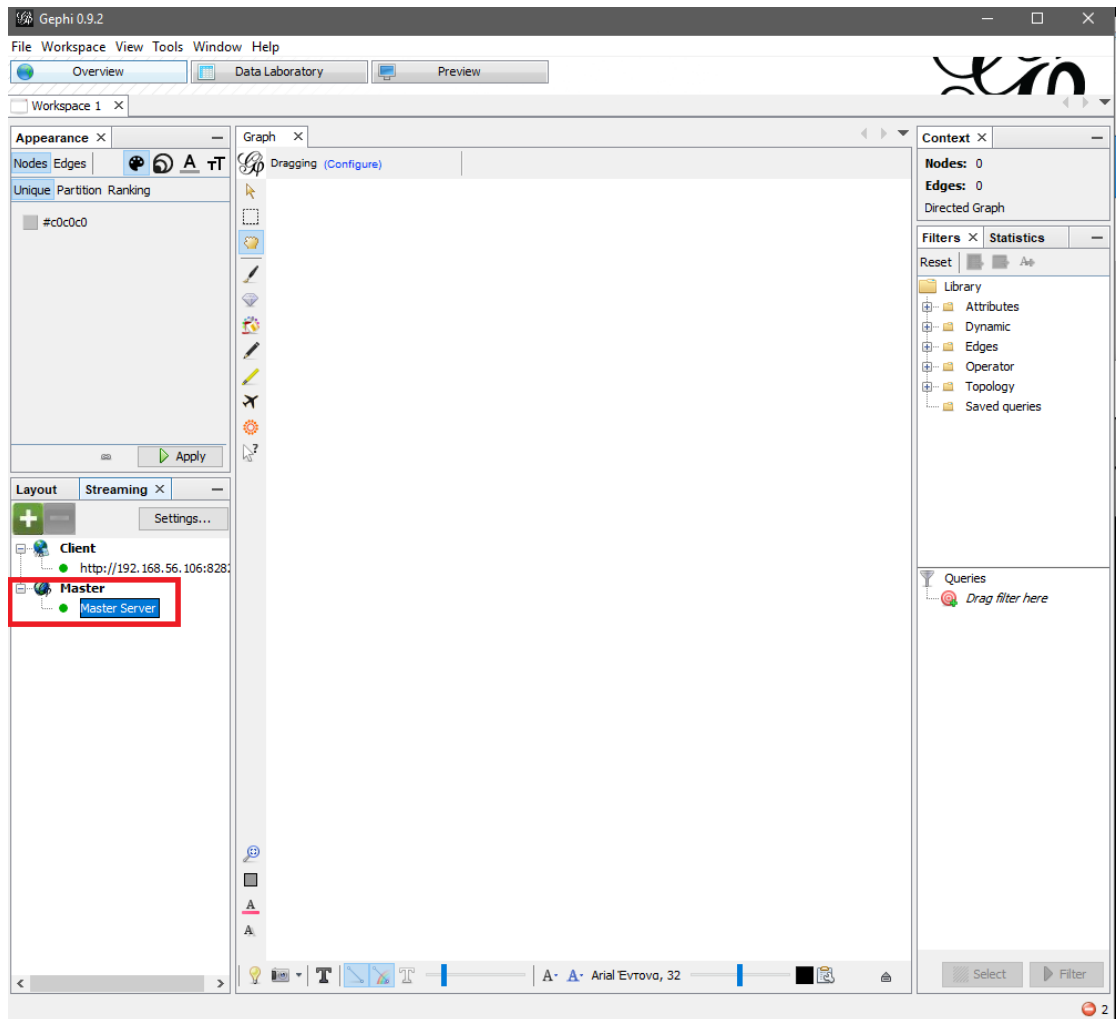
Εφόσον η σύνδεση είναι επιτυχής θα επιβεβαιωθεί και από το πράσινο λαμπάκι δίπλα από την αναφορά της. Για την εκκίνηση του master server του Gephi τον επιλέγουμε και με δεξί κλικ τον εκκινούμε με την επιλογή «Start» που επιβεβαιώνεται και από το πράσινο λαμπάκι δίπλα του. Έτσι έχουμε έτοιμο το Gephi το οποίο κάνοντας τα παραπάνω θα απεικονίσει μια τοπολογία Mininet όταν αυτή τρέξει.



Εικόνα 5.36:Επιβεβαίωση σύνδεσης μέσω Gephi.



Εικόνα 5.37:Ενεργοποίηση Master Server Gephi.



Εικόνα 5.38:Επιβεβαίωση σύνδεσης Master Server Gephi.

## 5.4 Εισαγωγή δεδομένων

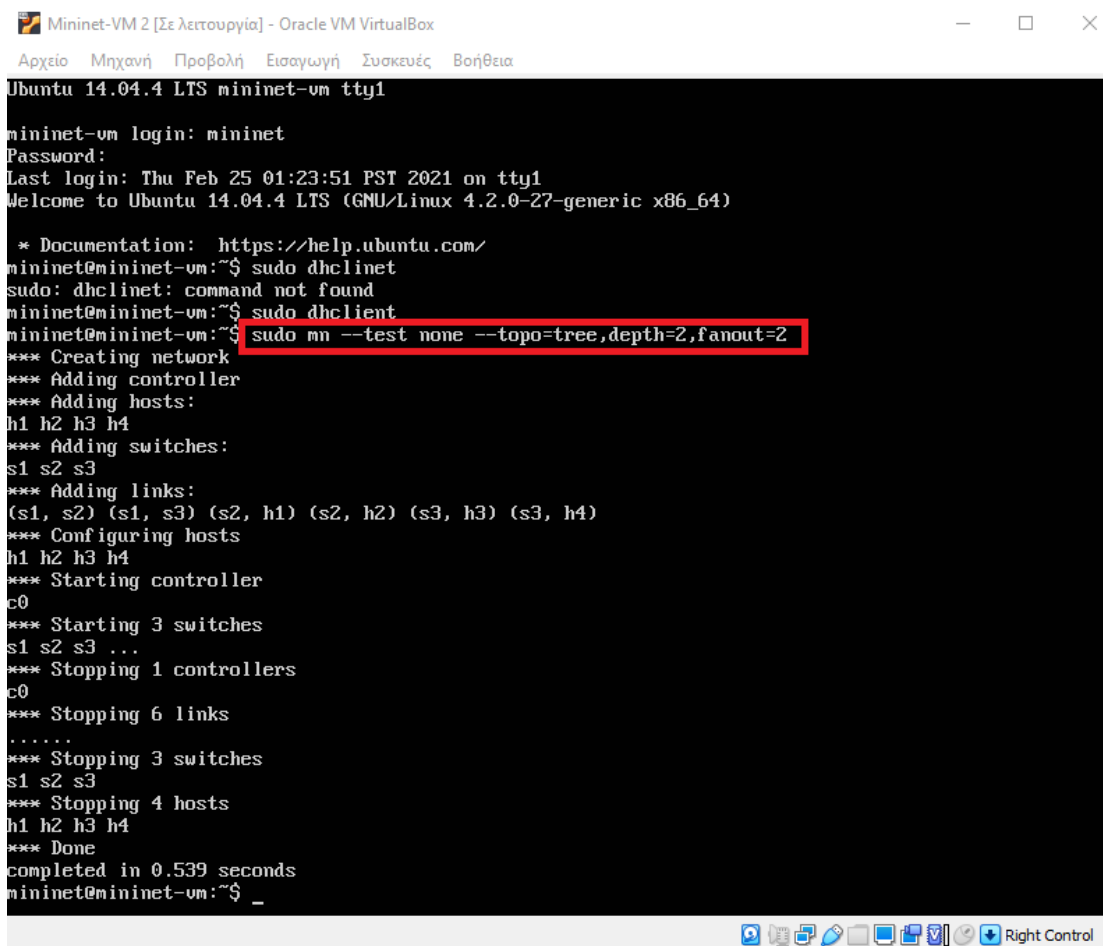
### 5.4.1 Εξομίωση τοπολογίας Tree

Έχοντας ανοιχτό το Mininet παράλληλα με το X-Ming και το PuTTY εκτελώντας την εντολή «htop» μέσω του PuTTY όπως γράφτηκε ακριβώς από πάνω εκτελούμε αρχικά την τοπολογία TREE αρχικά με «depth=2» και «fanout=2» μέσω της εντολής «sudo mn --test none --toro=tree,depth=2,fanout=2» Από την εκτέλεση της τοπολογίας στο Mininet λαμβάνουμε τα στοιχεία:

- Hosts (number)
- Switches (Number)

- Setup & Tear Time (s)

Με κατάλληλη χρονομέτρηση διαχωρίζεται ο χρόνος από τον συνολικό σε Setup και Tear Time ξεχωριστά



```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
Ubuntu 14.04.4 LTS mininet-vm tty1
mininet-vm login: mininet
Password:
Last login: Thu Feb 25 01:23:51 PST 2021 on tty1
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
mininet-vm:~$ sudo dhcclinnet
sudo: dhcclinnet: command not found
mininet-vm:~$ sudo dhclient
mininet-vm:~$ sudo mn --test none --topo=tree,depth=2,fanout=2
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, s2) (s1, s3) (s2, h1) (s2, h2) (s3, h3) (s3, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Stopping 1 controllers
c0
*** Stopping 6 links
.....
*** Stopping 3 switches
s1 s2 s3
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
completed in 0.539 seconds
mininet-vm:~$ _
```

Εικόνα 5.39:Εκτέλεση βασικής εξομοίωσης τοπολογίας Tree.

Ενώ από την εκτέλεση του «htop» σε ξεχωριστό terminal στο PuTTY λαμβάνουμε τα στοιχεία:

- CPU Usage (%)
- Memory Usage (Mb)

```

mininet@mininet-vm: ~
1 [          0.0%]
2 [          0.0%]
3 [          0.0%]
4 [          0.0%]
5 [          0.0%]
6 [          0.0%]
7 [          0.0%]
8 [          0.0%]
Mem[| | |          181/16047MB]
Swp:1021M used:0K

Load average: 0.02 0.07 0.10
Uptime: 00:14:59

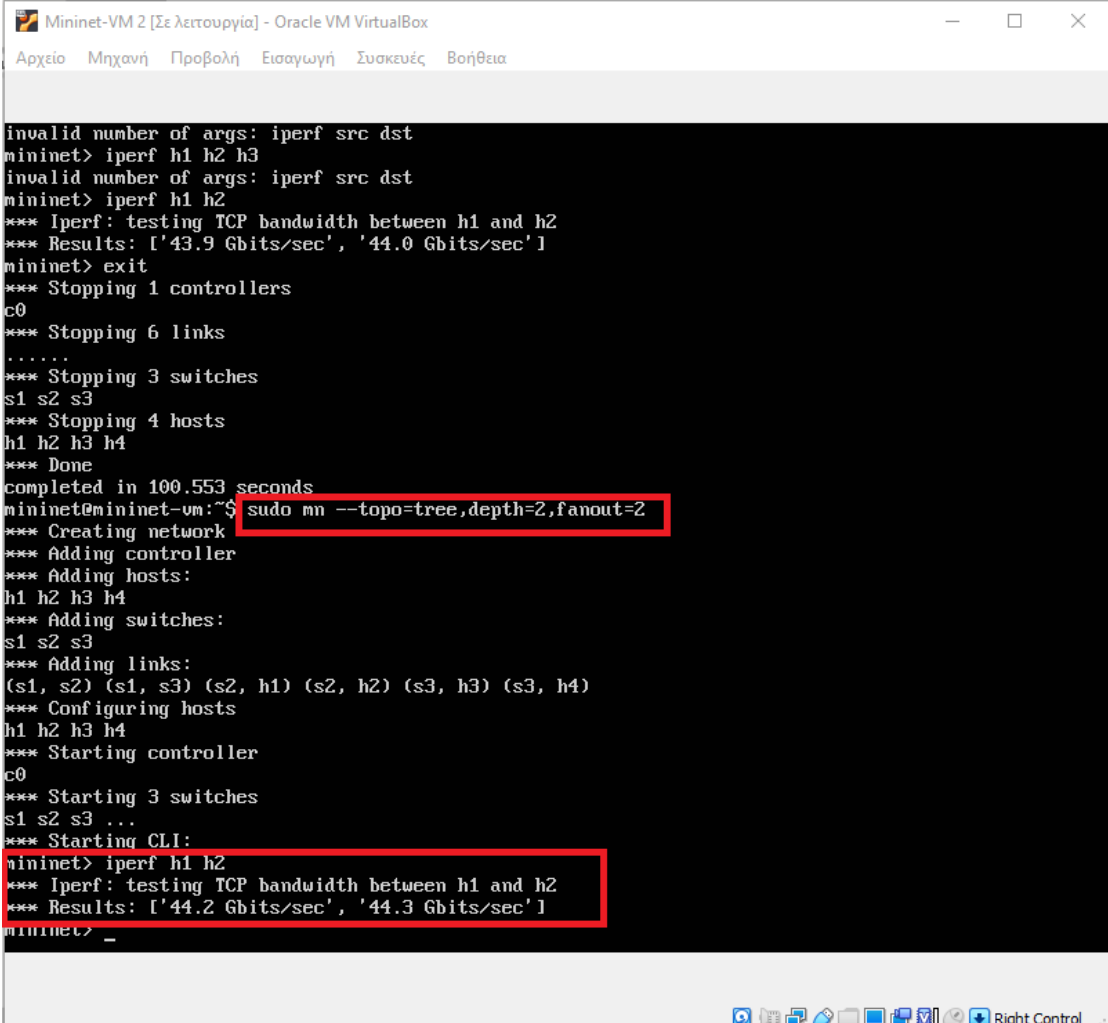
  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
  1 root        20   0 34380  4912  2656  S   0.0  0.0   0:07.36 /sbin/init
384 root        20   0 19756  1952  1684  S   0.0  0.0   0:02.18 upstart-udev-brid
653 messagebu  20   0 39244  2408  2032  S   0.0  0.0   0:01.82 dbus-daemon --sys
901 root        20   0 15688  1904  1340  S   0.0  0.0   0:01.77 upstart-file-brid
914 root        20   0 15276  1612  1332  S   0.0  0.0   0:01.72 upstart-socket-br
1250 root       10  -10 502M  42504  6424  S   0.0  0.3   0:01.47 ovs-vswitchd unix
1236 root       10  -10 20956  420    48  S   0.0  0.0   0:01.14 ovssdb-server: mon
1249 root       10  -10 21464  436    48  S   0.0  0.0   0:01.06 ovs-vswitchd: mon
1237 root       10  -10 37596 19412  2272  S   0.0  0.1   0:01.02 ovssdb-server /etc
6991 mininet    20   0 26324  3928  2924  R   0.0  0.0   0:00.60 htop
 389 root        20   0 51780  3568  2864  S   0.0  0.0   0:00.35 /lib/systemd/syst
1201 ntp         20   0 31476  3780  3292  S   0.0  0.0   0:00.14 /usr/sbin/ntpd -p
742 syslog    20   0 249M  2980  2296  S   0.0  0.0   0:00.13 rsyslogd
694 root        20   0 43464  3316  2956  S   0.0  0.0   0:00.09 /lib/systemd/syst
752 syslog    20   0 249M  2980  2296  S   0.0  0.0   0:00.07 rsyslogd
1337 mininet   20   0 103M  4260  3188  S   0.0  0.0   0:00.05 sshd: mininet@pts
1303 mininet   20   0 22752  5264  3328  S   0.0  0.0   0:00.04 -bash
1338 mininet   20   0 22736  5264  3356  S   0.0  0.0   0:00.03 -bash
750 syslog    20   0 249M  2980  2296  S   0.0  0.0   0:00.03 rsyslogd
751 syslog    20   0 249M  2980  2296  S   0.0  0.0   0:00.03 rsyslogd
1013 root       20   0 19208  2144  1896  S   0.0  0.0   0:00.03 /usr/sbin/irqbala
1192 root       20   0 78200  3616  3060  S   0.0  0.0   0:00.02 /bin/login --
1319 root       20   0 103M  6564  5492  S   0.0  0.0   0:00.01 sshd: mininet [pr
626 root       20   0 10244  4340  2044  S   0.0  0.0   0:00.00 dhclient -l -v -p
972 root       20   0 16072  2144  1988  S   0.0  0.0   0:00.00 /sbin/getty -8 38
974 root       20   0 16072  2140  1988  S   0.0  0.0   0:00.00 /sbin/getty -8 38
977 root       20   0 16072  2028  1880  S   0.0  0.0   0:00.00 /sbin/getty -8 38
978 root       20   0 16072  2080  1924  S   0.0  0.0   0:00.00 /sbin/getty -8 38
980 root       20   0 16072  2160  2000  S   0.0  0.0   0:00.00 /sbin/getty -8 38
1006 root      20   0 23668  2068  1856  S   0.0  0.0   0:00.00 cron
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit

```

Εικόνα 5.40: Παράθυρο διεργασιών μέσω htop σε επιπλέον τερματικό PuTTY όπου φαίνεται ο χρόνος εκτέλεσης λειτουργιών η χρήση CPU, RAM κ.α.

Για τη λήψη του Bandwidth (Gbps) το οποίο μετρείται μεταξύ των Nodes ανά ζευγάρι των 2 θα πρέπει γραφτεί η εντολή αφαιρώντας το κομμάτι «--test none» έτσι ώστε να μη τερματιστεί η εκτέλεση της τοπολογίας και να γραφτεί η εντολή μέσα στην τοπολογία «-iperf h1 h2» χρησιμοποιώντας ως παράδειγμα τους 2 πρώτους Hosts δίνοντας μας το Bandwidth.





```
Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια

invalid number of args: iperf src dst
mininet> iperf h1 h2 h3
invalid number of args: iperf src dst
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['43.9 Gbits/sec', '44.0 Gbits/sec']
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 6 links
.....
*** Stopping 3 switches
s1 s2 s3
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
completed in 100.553 seconds
mininet@mininet-vm:~$ sudo mn --topo=tree,depth=2,fanout=2
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, s2) (s1, s3) (s2, h1) (s2, h2) (s3, h3) (s3, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['44.2 Gbits/sec', '44.3 Gbits/sec']
mininet>
```

Εικόνα 5.41: Εντολή μέτρησης Bandwidth τοπολογιών.

Για τη λήψη του Latency (ms) το οποίο μετριέται μεταξύ των Nodes ανά ζευγάρι των 2 θα πρέπει γραφτεί η εντολή αφαιρώντας το κομμάτι «--test none» έτσι ώστε να μη τερματιστεί η εκτέλεση της τοπολογίας και να γραφτεί η εντολή μέσα στην τοπολογία θα είναι του τύπου «h1 ping -c count -s packetsize h2» όπου «count» μπαίνει ο αριθμός των πακέτων και όπου «packetsize» το μέγεθος του κάθε πακέτου χρησιμοποιώντας ως παράδειγμα τους 2 πρώτους Nodes δίνοντας μας Latency.

```

Mininet-VM 2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
completed in -31951.720 seconds
mininet@mininet-vm:~$ sudo mn --topo=tree,depth=2,fanout=2
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, s2) (s1, s3) (s2, h1) (s2, h2) (s3, h3) (s3, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> h1 ping -c 10 -s 1016 h2
PING 10.0.0.2 (10.0.0.2) 1016(1044) bytes of data.
1024 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.63 ms
1024 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.294 ms
1024 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.042 ms
1024 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.038 ms
1024 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.045 ms
1024 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.070 ms
1024 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.274 ms
1024 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.043 ms
1024 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.041 ms
1024 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.038 ms
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9029ms
rtt min/avg/max/mdev = 0.038/0.251/1.630/0.469 ms
mininet> _

```

Εικόνα 5.42: Εντολή μέτρησης Latency τοπολογιών όπου φαίνεται το μέγεθος πακέτου, ο αριθμός των πακέτων καθώς και αναλυτικοί χρόνοι για την καθυστέρηση μεταφοράς της πληροφορίας.

Αντιστοίχως εκτελούνται και οι υπόλοιπες εξομοιώσεις τοπολογίας «TREE» έχοντας ως εντολές εκτέλεσής τις κάτωθι με τη σειρά.

Πίνακας 5.1: Λίστα εντολών εκτέλεσης όλων των εξομοιώσεων τοπολογίας Tree.

CODE
<code>sudo mn --test none --topo=tree,depth=2,fanout=2</code>
<code>sudo mn --test none --topo=tree,depth=2,fanout=4</code>
<code>sudo mn --test none --topo=tree,depth=2,fanout=8</code>
<code>sudo mn --test none --topo=tree,depth=2,fanout=16</code>
<code>sudo mn --test none --topo=tree,depth=2,fanout=32</code>
<code>sudo mn --test none --topo=tree,depth=2,fanout=64</code>

## 5.4.2 Εξομίωση τοπολογίας Linear

Με την κατάλληλη αλλαγή εντολών για προσαρμογή στην τοπολογία «LINEAR» εκτελούνται οι εξομιώσεις παρομοίως με την τοπολογία «TREE» με εντολές εκτελέσεως.

Πίνακας 5.2: Λίστα εντολών εκτέλεσης όλων των εξομοιώσεων τοπολογίας Linear.

CODE
<code>sudo mn --test none --topo=linear,2,2</code>
<code>sudo mn --test none --topo=linear,2,4</code>
<code>sudo mn --test none --topo=linear,2,8</code>
<code>sudo mn --test none --topo=linear,2,16</code>
<code>sudo mn --test none --topo=linear,2,32</code>
<code>sudo mn --test none --topo=linear,2,64</code>

### 5.4.3 Εξομοίωση τοπολογίας Erdos Renyi

Με την κατάλληλη αλλαγή εντολών για προσαρμογή στην τοπολογία «ERDOS RENYI» εκτελούνται οι εξομοιώσεις παρομοίως με τις προηγούμενες τοπολογίες με εντολές εκτελέσεως.

Πίνακας 5.3: Λίστα εντολών εκτέλεση όλων των εξομοιώσεων τοπολογίας Erdos Renyi.

CODE
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=2,p=0.6 --test none</code>
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=4,p=0.6 --test none</code>
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=8,p=0.6 --test none</code>
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=16,p=0.6 --test none</code>
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=32,p=0.6 --test none</code>
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=64,p=0.6 --test none</code>

### 5.4.4 Γραφική απεικόνιση τοπολογιών μέσω Gerhi

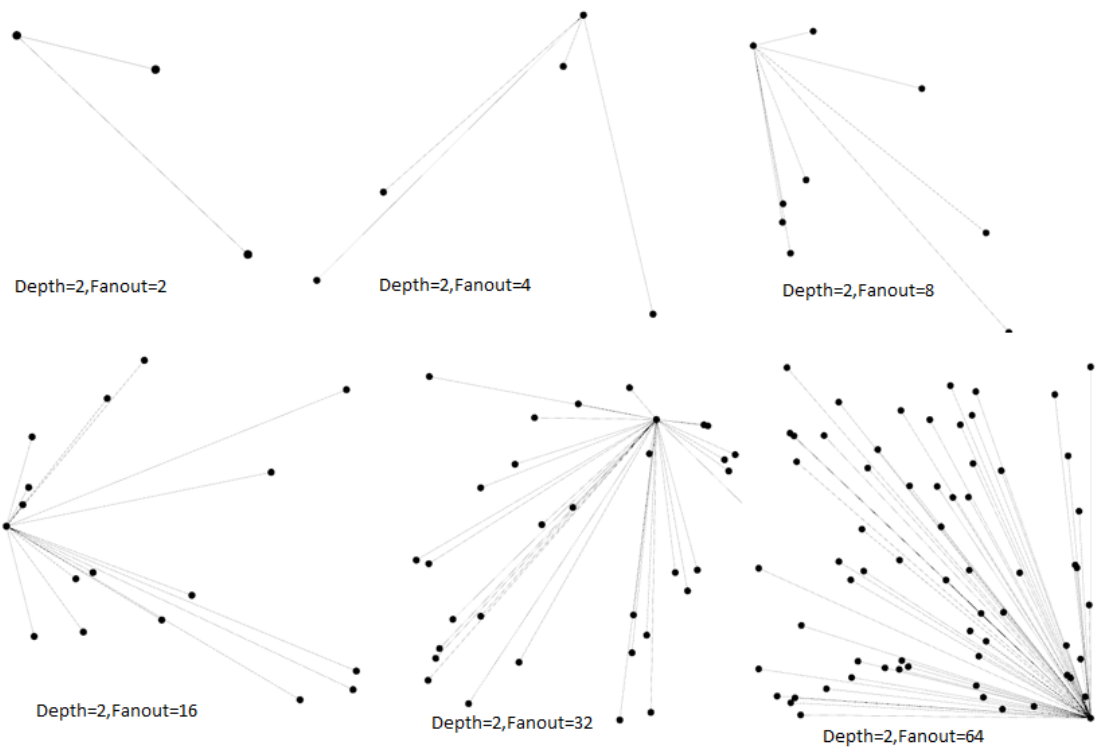
Για να απεικονίσουμε τις τοπολογίες έχουμε όπως είδαμε και στο κεφάλαιο 5.3.6 ανοιχτά τα Mininet, Gerhi και X-Ming καθώς και δημιουργούμε ένα επιπλέον τερματικό μέσω του προγράμματος PuTTY. Εκτελούμε τις τοπολογίες αλλά στον κώδικα τους βγάζουμε την εντολή «--test none» που χρησιμοποιείται να μετράει χρόνο εκτέλεσης και παύσης της τοπολογίας έτσι ώστε να μένει ανοιχτό το δίκτυο για να πάρουμε γραφική απεικόνιση και τη θέση της παίρνει η εντολή «--controller remote» η οποία επιτρέπει την απομακρυσμένη πρόσβαση στον ελεγκτή από το Gerhi

### 5.4.4.1 Γραφική απεικόνιση Tree

Εκτελώντας τους αντίστοιχους κώδικες για την κατασκευή και απεικόνιση της τοπολογίας Tree όπως φαίνονται παρακάτω παίρνουμε τα εξής αποτελέσματα. Επίσης είναι ευδιάκριτα τα χαρακτηριστικά λειτουργίας της τοπολογίας που εμφανίζεται σχηματικά ως κλαδιά δέντρων. Ο αριθμός των hosts σε κάθε δίκτυο με τοπολογία Tree είναι συνδεδεμένος σε όλα τα switch εκτός από το αρχικό που δημιουργεί το δέντρο.

Πίνακας 5.4:Κώδικας για γραφική απεικόνιση των εξομοιώσεων για την τοπολογία Tree στο Gerni.

CODE
<code>sudo mn --controller remote --topo=tree,depth=2,fanout=2</code>
<code>sudo mn --controller remote --topo=tree,depth=2,fanout=4</code>
<code>sudo mn --controller remote --topo=tree,depth=2,fanout=8</code>
<code>sudo mn --controller remote --topo=tree,depth=2,fanout=16</code>
<code>sudo mn --controller remote --topo=tree,depth=2,fanout=32</code>
<code>sudo mn --controller remote --topo=tree,depth=2,fanout=64</code>



Εικόνα 5.43:Γραφική απεικόνιση όλων των τοπολογιών Tree που χρησιμοποιήθηκαν στην εξομοίωση με τη βοήθεια του εργαλείου Gerhi.

#### 5.4.4.2 Γραφική απεικόνιση Linear

Εκτελώντας τους αντίστοιχους κώδικες για την κατασκευή και απεικόνιση της τοπολογίας Linear όπως φαίνονται παρακάτω παίρνουμε τα εξής αποτελέσματα. Στη συγκεκριμένη περίπτωση βλέπουμε πως φαίνονται μόνο τα switch που σχηματίζονται σε μία γραμμή λόγω των χαρακτηριστικών της τοπολογίας.

Πίνακας 5.5:Κώδικας για γραφική απεικόνιση των εξομοιώσεων για την τοπολογία Linear στο Gerni.

CODE
<code>sudo mn --controller remote --topo=linear,2,2</code>
<code>sudo mn --controller remote --topo=linear,2,4</code>
<code>sudo mn --controller remote --topo=linear,2,8</code>
<code>sudo mn --controller remote --topo=linear,2,16</code>
<code>sudo mn --controller remote --topo=linear,2,32</code>
<code>sudo mn --controller remote --topo=linear,2,64</code>

Στις εξομοιώσεις που υλοποιήθηκαν στην τοπολογία Linear λόγω του ότι έμεινε σταθερός ο αριθμός των switch αυτά είναι που απεικονίζονται σε μια ευθεία γραμμή βάση χαρακτηριστικών τοπολογίας. Σε κάθε περίπτωση ο αριθμός των Hosts είναι αντίστοιχα 2,4,8,16,32,64.



Εικόνα 5.44:Γενική γραφική απεικόνιση τοπολογίας Linear με το εργαλείο Gerni.

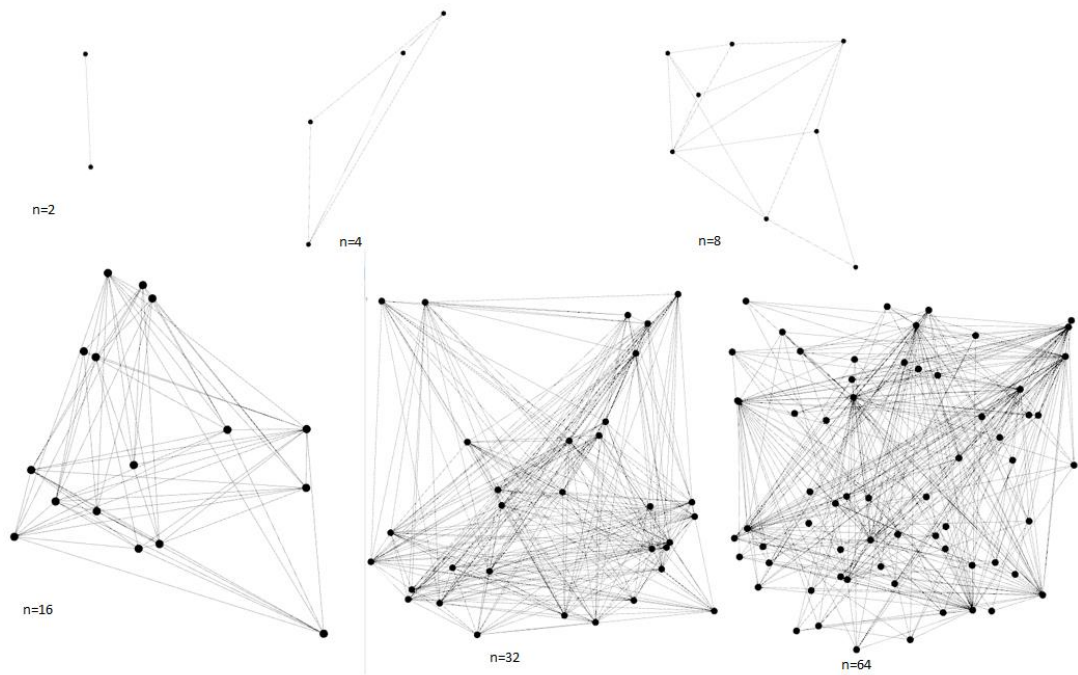
### 5.4.4.3 Γραφική απεικόνιση Erdos Renyi

Εκτελώντας τους αντίστοιχους κώδικες για την κατασκευή και απεικόνιση της τοπολογίας Erdos-Renyi όπως φαίνονται παρακάτω παίρνουμε τα εξής αποτελέσματα. Όπως φαίνεται και στη γραφική απεικόνιση παρακάτω το κύριο χαρακτηριστικό της τοπολογίας Erdos Renyi είναι να παράγει τυχαία γραφήματα.

*Πίνακας 5.6:Κώδικας για γραφική απεικόνιση των εξομοιώσεων για την τοπολογία Erdos Renyi στο Gephi. .*

CODE
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=2,p=0.6 --controller remote</code>
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=4,p=0.6 --controller remote</code>
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=8,p=0.6 --controller remote</code>
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=16,p=0.6 --controller remote</code>
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=32,p=0.6 --controller remote</code>
<code>sudo mn --custom mn_nx_topos.py --topo erdos_renyi,n=64,p=0.6 --controller remote</code>





Εικόνα 5.44:Γραφική απεικόνιση όλων των τοπολογιών Erdos Renyi που χρησιμοποιήθηκαν στην εξομίωση με τη βοήθεια του εργαλείο Gephi.

## 5.5 Συλλογή αποτελεσμάτων

Η κάθε εξομίωση εκτελέστηκε 100 φορές προκειμένου να πραγματοποιηθεί δειγματοληπτικός έλεγχος με ακρίβεια και παρουσίασε κατά μέσο όρο τα εξής αποτελέσματα μην έχοντας μεγάλες αποκλίσεις μεταξύ των επαναλήψεων. Παρακάτω παρουσιάζονται σε πίνακες ανά τοπολογία τα δεδομένα που συλλέχθηκαν κατά τη διάρκεια των εξομοιώσεων σύμφωνα με τον μέσο όρο αυτών των εκτελέσεων. Σε ότι αφορά το Latency αυτό θα εκτελεστεί όπως θα φανεί και παρακάτω σε κάθε τοπολογία σε σταθερό αριθμό

Πίνακας 5.7:Πίνακας συλλογής στατιστικών εξομοιώσεων με τη χρήση της τοπολογίας Tree.

CODE	CPU(%)	MEMORY(MB)	SWITCH	HOST	BANDWIDTH(GBPS)	SETUP/TEAR TIME(SEC)
Depth=2 Fanout=2	5,3	200	3	4	44	0,326/0,213
Depth=2 Fanout=4	17,1	250	5	16	42	0,652/,0510
Depth=2 Fanout=8	25,6	300	9	64	41	2,133/1,802
Depth=2 Fanout=16	43,4	500	17	256	41	5,832/8,050
Depth=2 Fanout=32	47,9	1300	33	1024	40	26,977/41,901
Depth=2 Fanout=64	53,8	4600	65	4096	31	250,031/265,851

Πίνακας 5.8: Πίνακας συλλογής στατιστικών εξομοιώσεων με τη χρήση της τοπολογίας *Linear*.

CODE	CPU(%)	MEMORY(MB)	SWITCH	HOST	BANDWIDTH(GBPS)	SETUP/TEAR TIME(SEC)
2,2	3,7	420	2	4	42	0,156/0,112
2,4	5,2	440	2	8	41	0,251/0,224
2,8	10,1	460	2	16	36	0,382/0,412
2,16	20,3	480	2	32	41	0,759/0,802
2,32	23,5	500	2	64	42	1,530/1,781
2,64	38,3	600	2	128	38	3,258/3,495

Πίνακας 5.9: Πίνακας συλλογής στατιστικών εξομοιώσεων με τη χρήση της τοπολογίας *Erdos Renyi*.

CODE	CPU(%)	MEMORY(MB)	SWITCH	HOST	BANDWIDTH(GBPS)	SETUP/TEAR TIME(SEC)
n=2,p=0.6	4,7	180	2	2	39	0,162/0,165
n=4,p=0.6	5,3	220	4	4	38	0,283/0,299
n=8,p=0.6	16,1	250	8	8	46	0,613/0,747
n=16,p=0.6	23,2	280	16	16	45	2,031/2,792
n=32,p=0.6	38,8	340	32	32	44	4,126/12,853
n=64,p=0.6	52,3	550	64	64	35	21,398/47,956

Σε ότι αφορά το latency των δικτύων εδώ η μέτρηση θα γίνει με λίγο διαφορετικό και τρόπο καθώς θα μετρηθεί σε κάθε δίκτυο σε παρόμοιες συνθήκες με σταθερό μέγεθος πακέτου αυξάνοντας τον αριθμό των πακέτων και βλέποντας πως αυτό επηρεάζει το δίκτυο συλλέγοντας τον μέσο και τον συνολικό χρόνο μεταφοράς. Το μέγεθος του κάθε πακέτου θα είναι 1024Byte(1KB) και θα εκτελεστούν εξομοιώσεις με αριθμό πακέτων αντίστοιχα [1,10,50,100,500] και αυτό έγινε διότι σαν όριο χρήσης στο Mininet ανά περίπτωση παρατηρήθηκαν περίπου τα 600 πακέτα με αυτό το μέγεθος πακέτου. Οι προδιαγραφές των τοπολογιών θα είναι:

- Tree: Depth=2, Fanout=32

Πίνακας 5.10: Πίνακας Latency τοπολογίας Tree.

PACKET NUMBER	MIN LATENCY(MS)	MAX LATENCY(MS)	AVERAGE LATENCY(MS)	TOTAL LATENCY(MS)
1	0.065	0.065	0.065	0.254
10	0.033	0.399	0.072	9008
50	0.035	0.295	0.051	49219
100	0.032	0.365	0.045	99836
500	0.030	0.679	0.059	502848

- Linear: 2,32

Πίνακας 5.11: Πίνακας Latency τοπολογίας Linear.

PACKET NUMBER	MIN LATENCY(MS)	MAX LATENCY(MS)	AVERAGE LATENCY(MS)	TOTAL LATENCY(MS)
1	0.028	0.028	0.028	0.028
10	0.036	0.201	0.056	9013
50	0.034	0.333	0.058	49165
100	0.031	0.411	0.066	99177
500	0.024	0.931	0.044	500457

- Erdos Renyi: n=32, p=0.6

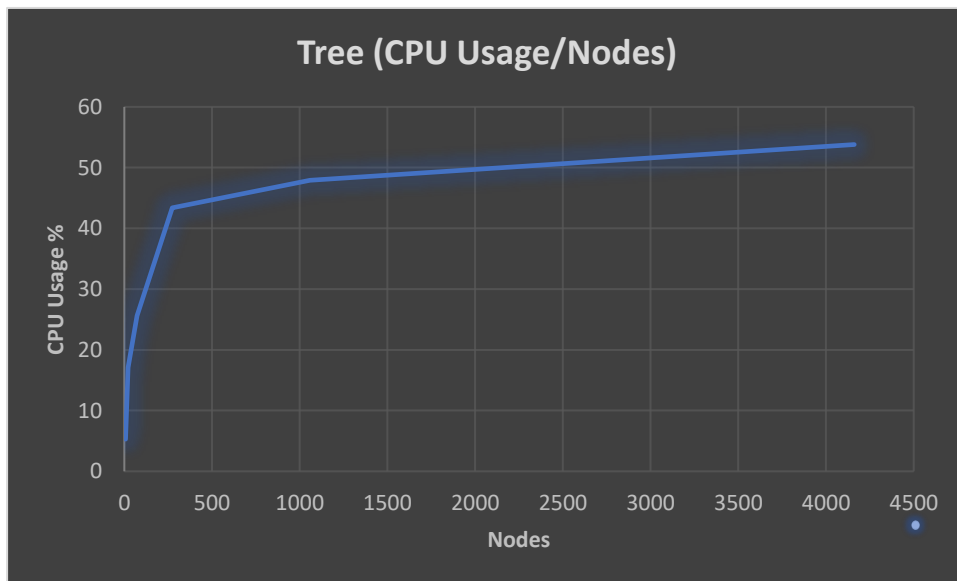
Πίνακας 5.12: Πίνακας Latency τοπολογίας Erdos Renyi.

PACKET NUMBER	MIN LATENCY(MS)	MAX LATENCY(MS)	AVERAGE LATENCY(MS)	TOTAL LATENCY(MS)
1	0.015	0.015	0.016	0.016
10	0.014	0.055	0.027	9034
50	0.016	0.044	0.026	49126
100	0.015	0.046	0.023	99036
500	0.013	0.105	0.022	499131

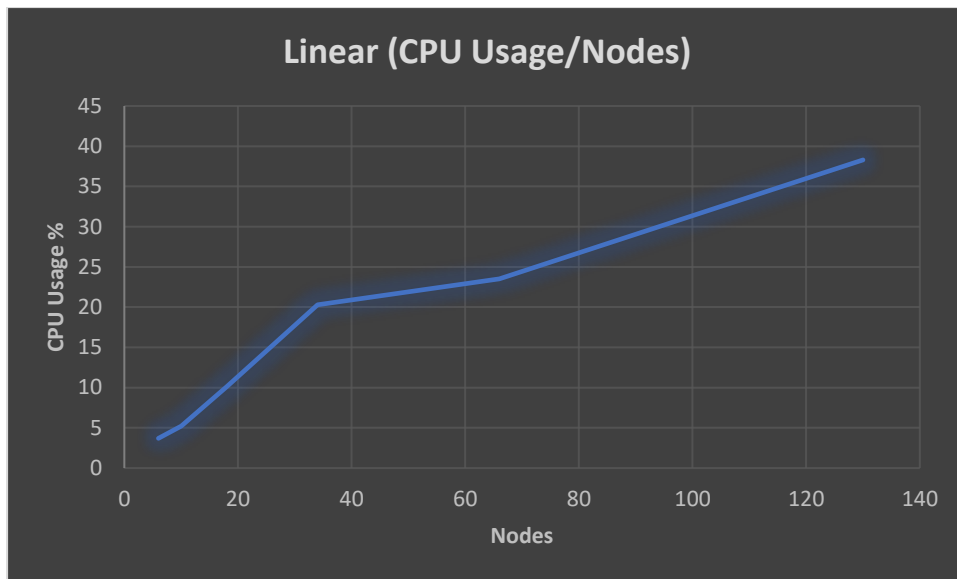
## 5.6 Ανάλυση αποτελεσμάτων

Για την καλύτερη ερμηνεία των παραπάνω αποτελεσμάτων τα μετατρέπουμε σε διαγράμματα. Σε κάθε διάγραμμα όπου είναι απαραίτητο θα υπάρχει ο συνολικός αριθμός κόμβων(Nodes) ο οποίος είναι  $Nodes=Switches+Hosts$ . Επίσης σε κάθε διάγραμμα για καλύτερη απεικόνιση θα χρησιμοποιηθεί ο συνολικός χρόνος δημιουργίας και καταστροφής τοπολογίας (Setup+Tear Time). Τέλος στα διαγράμματα μέτρησης Latency θα συμπεριληφθούν μόνο οι τιμές Average Latency(Μέση Καθυστέρηση) και Total Latency (Συνολική Καθυστέρηση) καθώς αυτές οι τιμές είναι που δίνουν αντιπροσωπευτικά στοιχεία για κάθε τοπολογία και όπως αναφέρθηκε και στην προηγούμενη ενότητα θα μετρηθεί το Latency σε σταθερούς παρόμοιους τύπους στις τοπολογίες αλλά με μεταβαλλόμενο αριθμό πακέτων.

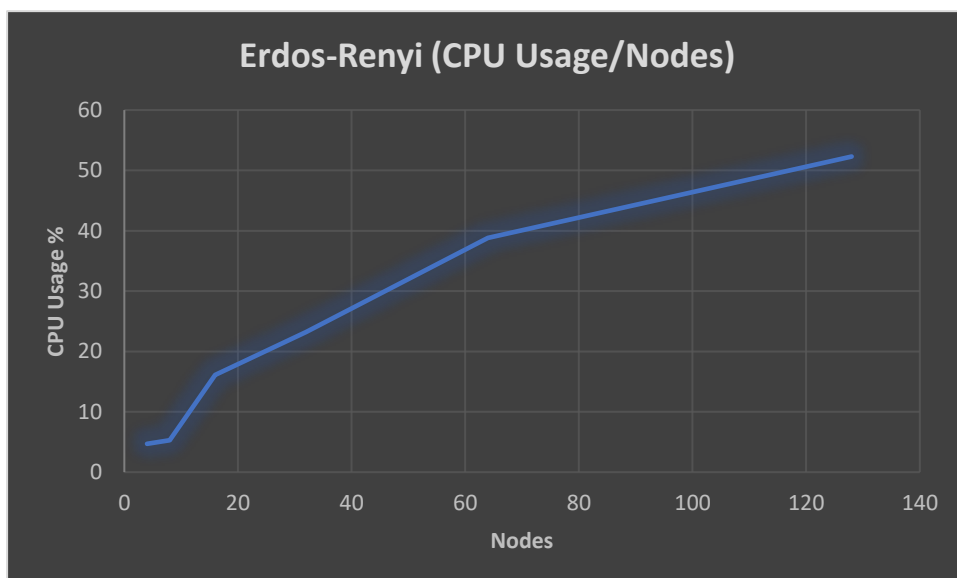
### 5.6.1 CPU Usage



Γράφημα 5.1:Χρήση επεξεργαστή (CPU Usage %) ως προς το συνολικό αριθμό κόμβων στις εξομοιώσεις με χρήση τοπολογίας Tree.



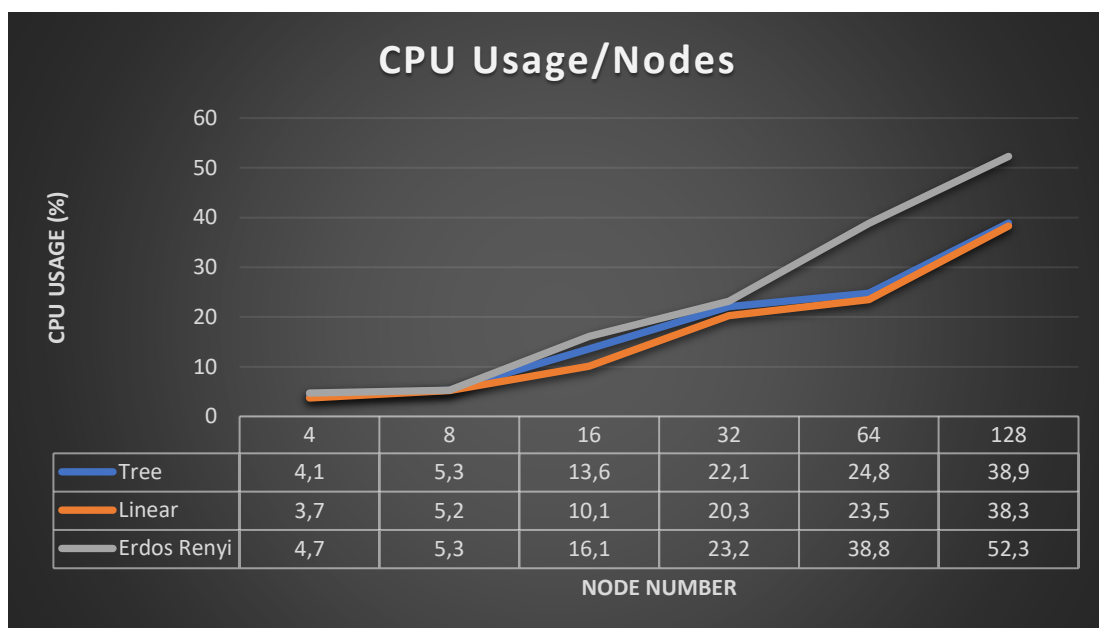
Γράφημα 5.2:Χρήση επεξεργαστή (CPU Usage %) ως προς το συνολικό αριθμό κόμβων στις εξομοιώσεις με χρήση τοπολογίας Linear.



Γράφημα 5.3:Χρήση επεξεργαστή (CPU Usage %) ως προς το συνολικό αριθμό κόμβων στις εξομοιώσεις με χρήση τοπολογίας Erdos Renyi.

Από τα αποτελέσματα των μετρήσεων που παρουσιάζονται στα Γραφήματα 5.1-5.4 παρατηρούμε το ποσοστό χρήσης της CPU για τις τρεις διαφορετικές τοπολογίες. Όπως διαπιστώνουμε από τις απεικονίσεις, και για τις τρεις τοπολογίες παρατηρείται παρόμοια συμπεριφορά. Όσο αυξάνεται ο αριθμός των κόμβων, τόσο αυξάνεται η χρήση της CPU. Στην τοπολογία Tree (Γράφημα 5.1) παρατηρείται σταθερή αύξηση της χρήσης της CPU. Ειδικότερα διαπιστώνεται ότι η αύξηση της χρήσης της CPU είναι ανάλογη του αριθμού των κόμβων. Για παράδειγμα ο διπλασιασμός των κόμβων από 64 σε 128 επιφέρει μια αύξηση στη χρήση της CPU κατά 15% περίπου. Αξίζει βέβαια να σημειωθεί πως αυτό συμβαίνει για

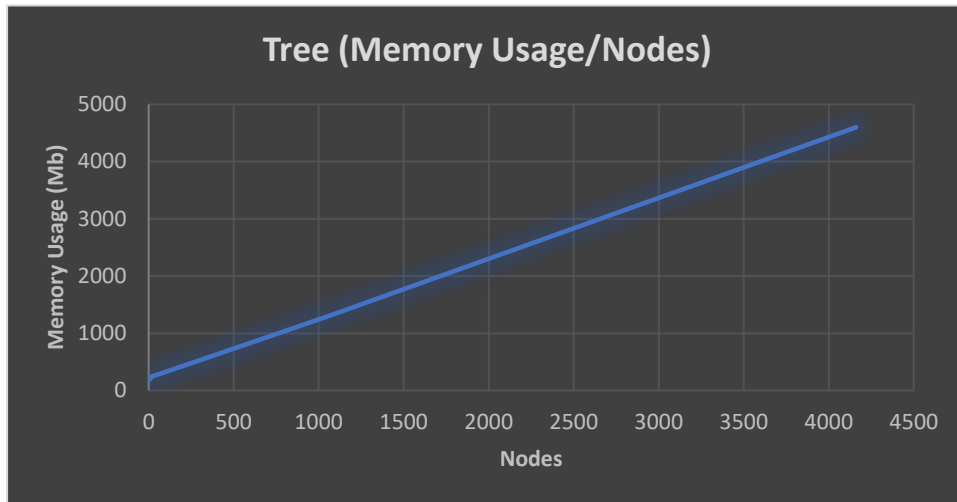
τον αρχικά σχετικά μικρό αριθμό κόμβων. Από ένα σημείο και μετά (περίπου 1000 κόμβοι) η χρήση της CPU κυμαίνεται γύρω από το 50%. Στην τοπολογία Linear (Γράφημα 5.2) με σταθερό αριθμό Switch και σταθερά μεταβαλλόμενο αριθμό Nodes παρατηρήθηκε παρόμοια συμπεριφορά, τουλάχιστον όταν το πλήθος των κόμβων κυμαίνεται από 5 έως 128. Στην custom τοπολογία Erdos Renyi (Γράφημα 5.3) και για σχετικά μικρό αριθμό κόμβων (από 5 έως 32) η χρήση της CPU κυμαίνεται από περίπου 5% έως 23%, δηλαδή συμπεριφορά παρόμοια με αυτή των άλλων δύο τοπολογιών. Από εκεί και μετά παρατηρείται μια απότομη αύξηση της χρήσης της CPU συγκριτικά με τις άλλες δύο τοπολογίες, η οποία μάλιστα ξεπερνά το 50% για όταν το πλήθος των κόμβων είναι 128.



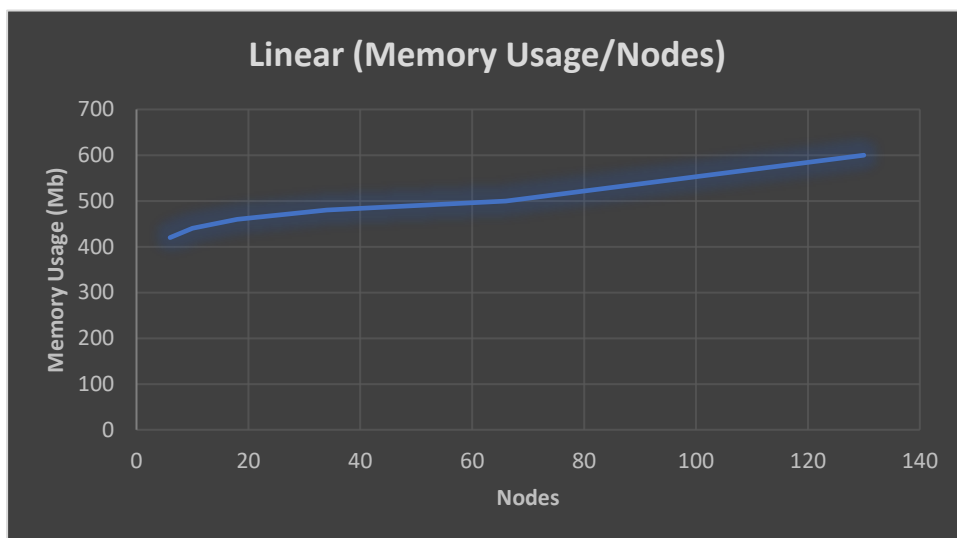
Γράφημα 5.4: Χρήση επεξεργαστή (CPU Usage %) ως προς τον ίδιο αριθμό κόμβων στις εξομοιώσεις συγκριτικά και για τις τρεις τοπολογίες.

Συγκρίνοντας και τις τρεις τοπολογίες σε σχέση και για τον ίδιο αριθμό κόμβων (Γράφημα 5.4) παρατηρούμε ότι μια διαφοροποίηση στη συμπεριφορά της τοπολογία Erdos Renyi ως προς τη χρήση της CPU. Η τοπολογία Erdos Renyi φαίνεται να απασχολεί την CPU σε μεγαλύτερο ποσοστό συγκριτικά με τις άλλες τοπολογίες. Αυτό μπορεί να οφείλεται σε δύο λόγους: α) τον διαφορετικό αριθμό των switches που περιλαμβάνονται στην τοπολογία β) στην πολυπλοκότητα που εισάγει η ίδια η τοπολογία. Πρέπει εδώ να σημειωθεί, ότι απαιτούνται περαιτέρω πειραματικές μετρήσεις για να ξεκαθαριστεί το σημείο αυτό, που όμως είναι πέρα από τα πλαίσια αυτής της εργασίας.

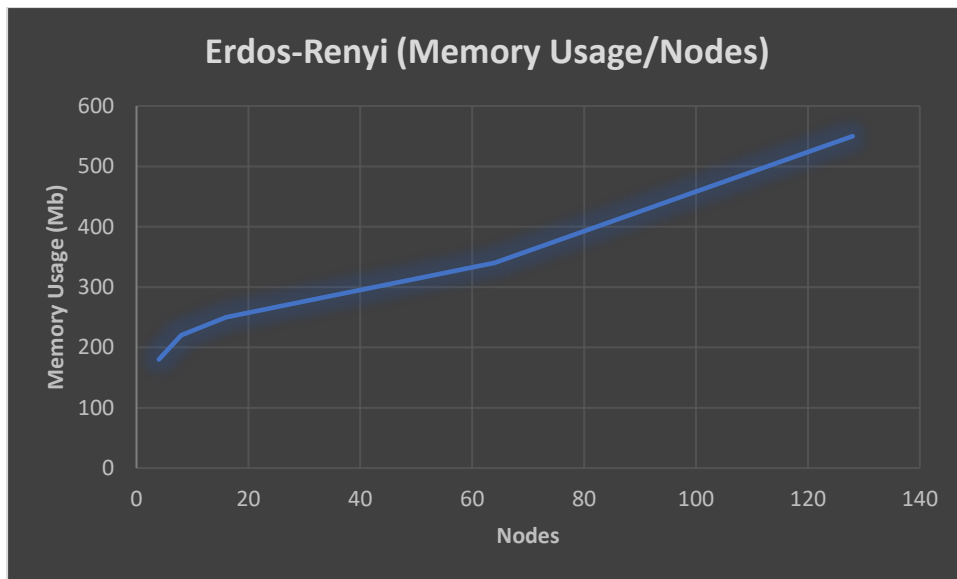
## 5.6.2 Memory Usage



Γράφημα 5.5:Χρήση μνήμης RAM (RAM Usage mb) ως προς το συνολικό αριθμό κόμβων στις εξομοιώσεις με χρήση τοπολογίας Tree.



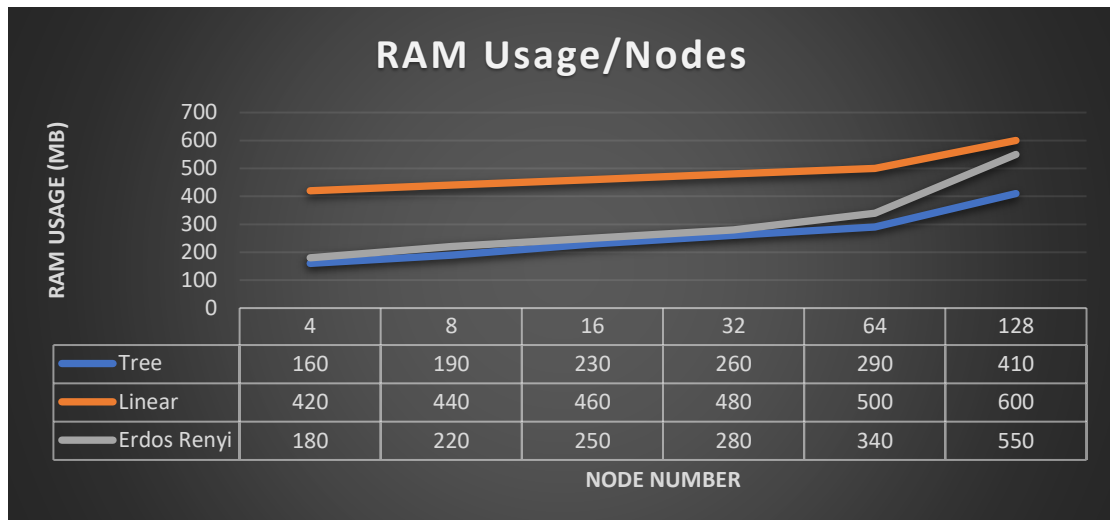
Γράφημα 5.6:Χρήση μνήμης RAM (RAM Usage mb) ως προς το συνολικό αριθμό κόμβων στις εξομοιώσεις με χρήση τοπολογίας Linear.



Γράφημα 5.7:Χρήση μνήμης RAM (RAM Usage mb) ως προς το συνολικό αριθμό κόμβων στις εξομοιώσεις με χρήση τοπολογίας Erdos Renyi.

Τα αποτελέσματα των μετρήσεων για τη χρήση της μνήμης παρουσιάζονται στα Γραφήματα 5.5-5.8 Στην τοπολογία Tree (Γράφημα 5.5) παρατηρείται γραμμική αύξηση της της χρήσης της μνήμης καθώς αυξάνεται το πλήθος των κόμβων. Μάλιστα, για 4500 κόμβους η απαιτήσεις σε μνήμη ξεπερνούν τα 4500MB. Στην τοπολογία Liner (Γράφημα 5.6) παρατηρείται μια σχετικά σταθερή χρήση μνήμης, ενώ στην τοπολογία Erdos Renyi (Γράφημα 5.7) η χρήση της μνήμης αυξάνει σταδιακά με την αύξηση των κόμβων και μεταβάλλεται από 180MB έως 550MB. Η τοπολογία Erdos Renyi έχει μεγαλύτερο αριθμό Switch και παρατηρείται μείωση χρήσης μνήμης RAM διότι όπως διαπιστώθηκε παραπάνω όσο αυξάνεται ο αριθμός των Switch τόσο επιβαρύνεται το CPU Usage. Αυτό που αποδεικνύεται είναι ότι όσο αυξάνεται ο αριθμός των Hosts τόσο αυξάνεται και η χρήση μνήμης RAM. Συγκεκριμένα σε επιπλέον πειράματα που έγιναν πέρα από τα παρουσιαζόμενα και ειδικά στην τοπολογία Tree παρατηρήθηκε ότι σε αριθμό Hosts που πλησίαζε τον αριθμό 10000 η χρήση μνήμης RAM ξεπέρασε τα 9000MB. Συμπερασματικά, η αύξηση των hosts μειονεκτεί ως προς την κλιμάκωση, καθώς από τα συγκεκριμένα πειράματα με πολύ μεγάλο αριθμό κόμβων (πάνω από 10000) έφτανε στα όρια του εξομοιωτή Mininet στο παρόν σύστημα. Από τα αποτελέσματα διαφαίνεται .να υπάρχει σχέση από το είδος των κόμβων, για παράδειγμα hosts ή switches και χρειάζεται περαιτέρω διερεύνηση

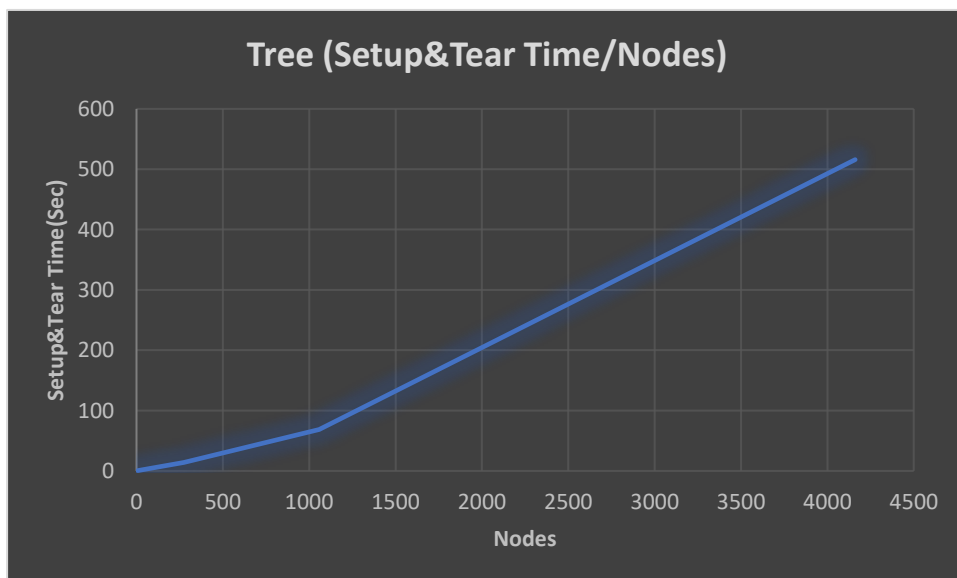




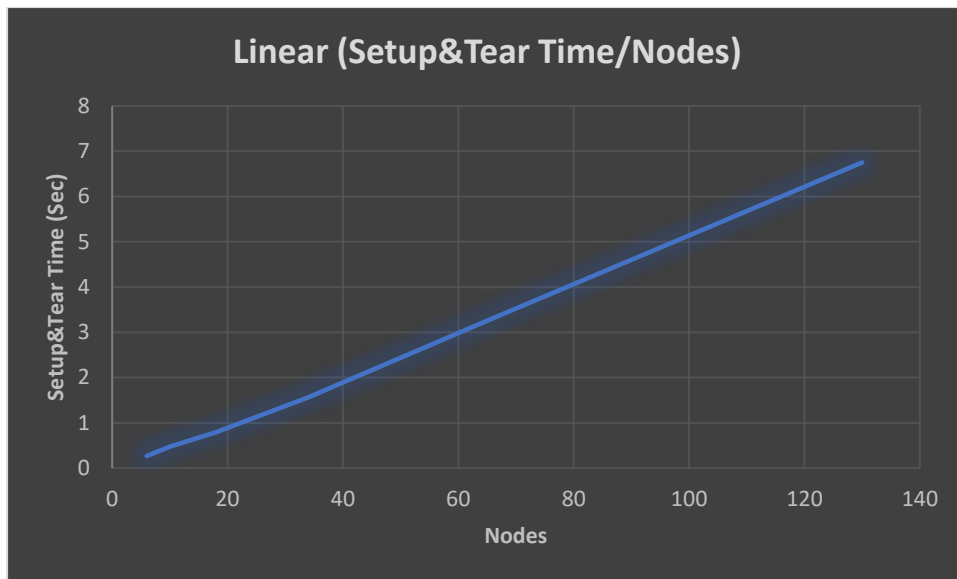
Γράφημα 5.8:Χρήση μνήμης RAM (RAM Usage mb) ως προς το ίδιο αριθμό κόμβων στις εξομοιώσεις συγκριτικά και για τις τρεις τοπολογίες.

Συγκριτικά για και τις τρεις τοπολογίες με τον ίδιο αριθμό κόμβων απεικονίζονται στο Γράφημα 5.8. Είναι προφανές ότι η τοπολογία Linear χρησιμοποιεί περισσότερη μνήμη RAM, χωρίς όμως να μεταβάλλεται ουσιαστικά από τον αριθμό των κόμβων, ενώ για τις άλλες δύο τοπολογίες παρατηρείται μια αύξηση που σχετίζεται με την αύξηση στον αριθμό των switches σε καθεμιά από αυτές.

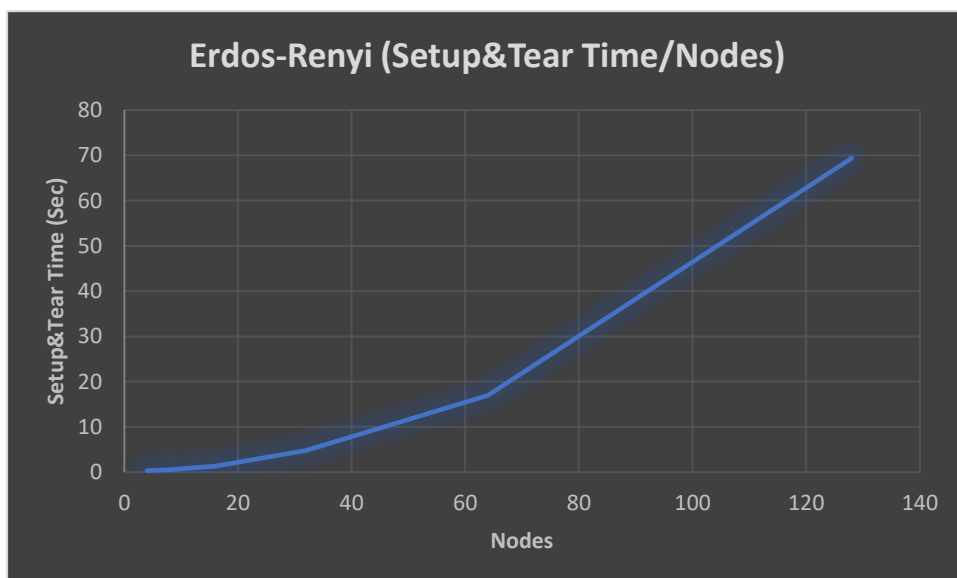
### 5.6.3 Setup & Tear Time



Γράφημα 5.9: Setup & Tear Time ως προς τον αριθμό των κόμβων στις εξομοιώσεις με χρήση τοπολογίας Tree.



Γράφημα 5.10: Setup & Tear Time ως προς τον αριθμό των κόμβων στις εξομοιώσεις με χρήση τοπολογίας Linear.

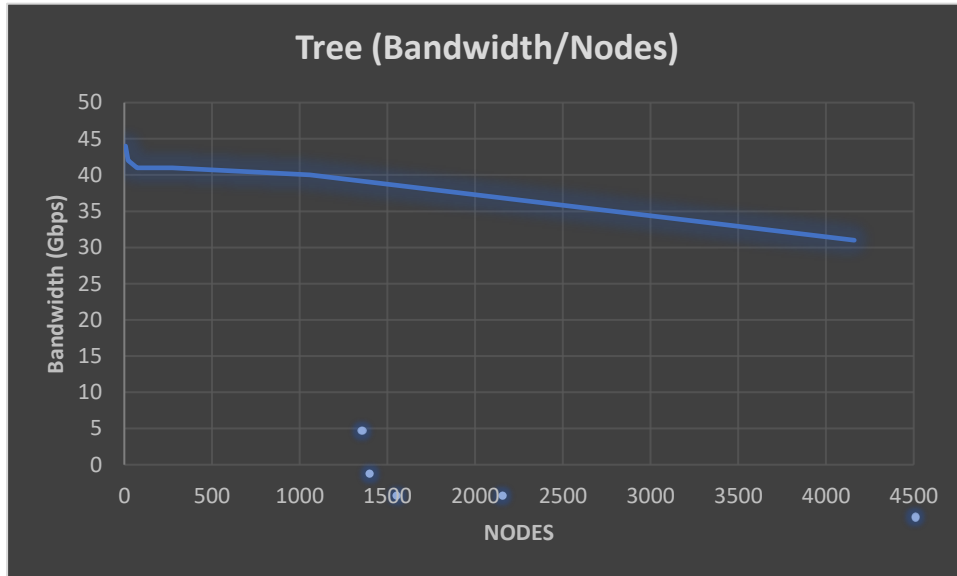


Γράφημα 5.11: Setup & Tear Time ως προς τον αριθμό των κόμβων στις εξομοιώσεις με χρήση τοπολογίας Erdos Renyi.

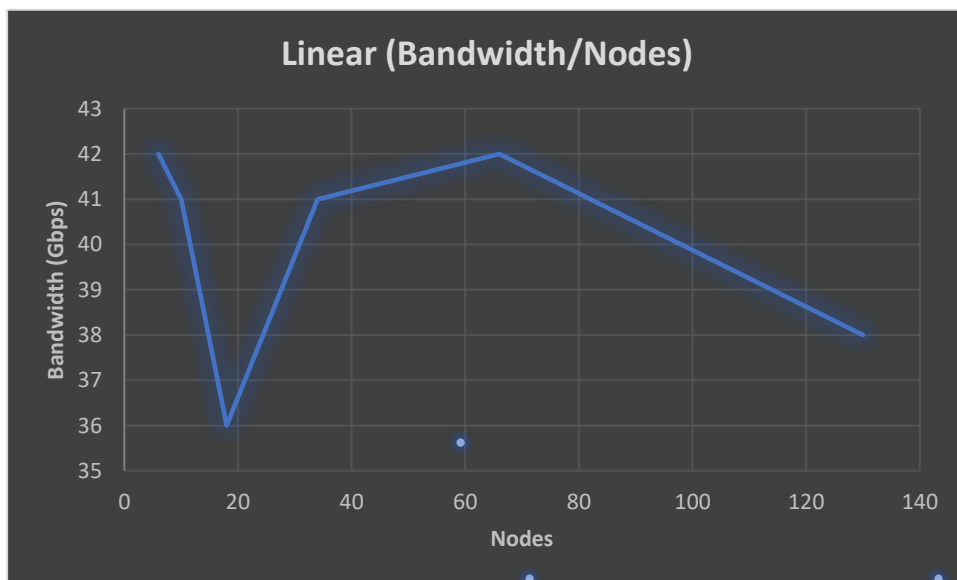
Οι μετρήσεις για το Setup & Tear Time αφορούν και τις δύο τιμές αθροιστικά και απεικονίζονται στα Γραφήματα 5.9 - 5.11. Όπως διαπιστώνεται στην τοπολογία Erdos Renyi απαιτείται περισσότερος χρόνος για την εγκατάσταση και καταστροφή της τοπολογίας σε σχέση με τις άλλες δύο τοπολογίες. Για παράδειγμα για 128 κόμβους το Setup & Tear Time είναι περίπου 13, 7, 68 sec για τις τοπολογίες Tree, Linear, Erdos Renyi αντίστοιχα.. Μεταξύ των τοπολογιών Tree, Linear και Erdos Renyi υπάρχει μεγάλη διαφορά χρόνου καθώς στην τοπολογία Erdos Renyi αυξάνεται εκθετικά διότι λόγω της δομής αλλά και του τρόπου

σηματισμού της. Για παράδειγμα η τοπολογία Erdos Renyi ξεθεώνει δεκαπλάσιο Setup & Tear Time από ότι η τοπολογία Linear.

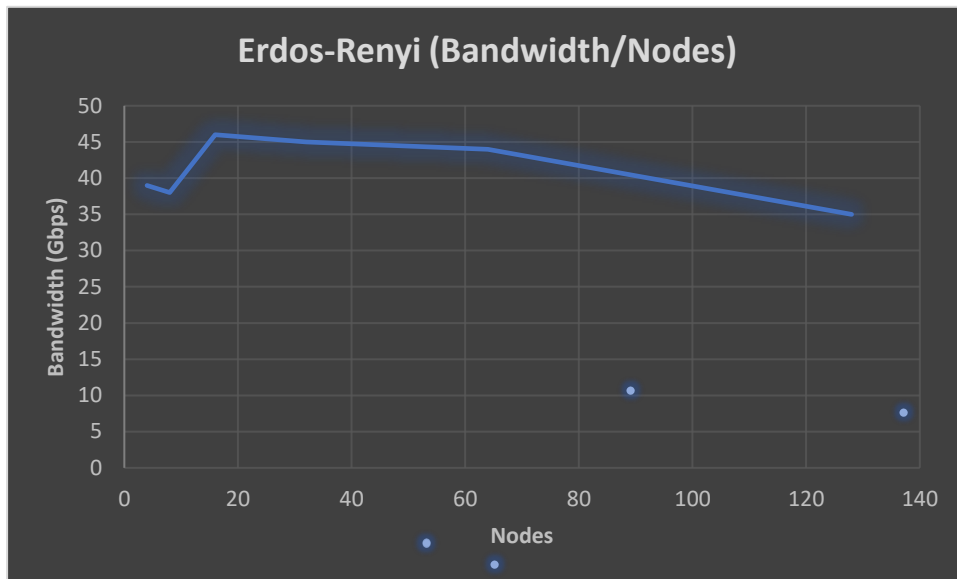
### 5.6.4 Bandwidth



Γράφημα 5.12: Bandwidth (Gbps) ως προς τον αριθμό των κόμβων στις εξομοιώσεις με χρήση τοπολογίας Tree.

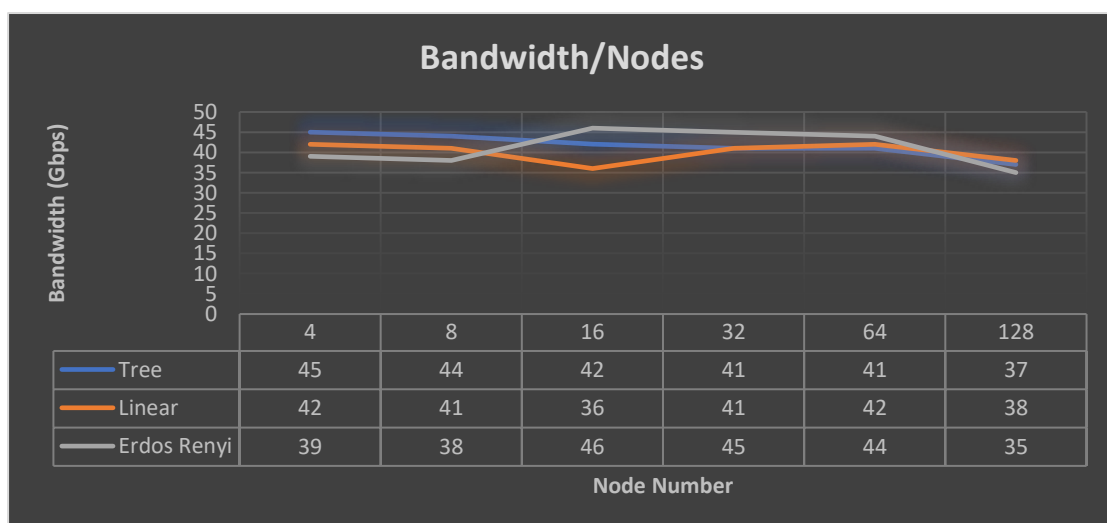


Γράφημα 5.13: Bandwidth (Gbps) ως προς τον αριθμό των κόμβων στις εξομοιώσεις με χρήση τοπολογίας Linear.



Γράφημα 5.14: Bandwidth (Gbps) ως προς τον αριθμό των κόμβων στις εξομοιώσεις με χρήση τοπολογίας Erdos Renyi.

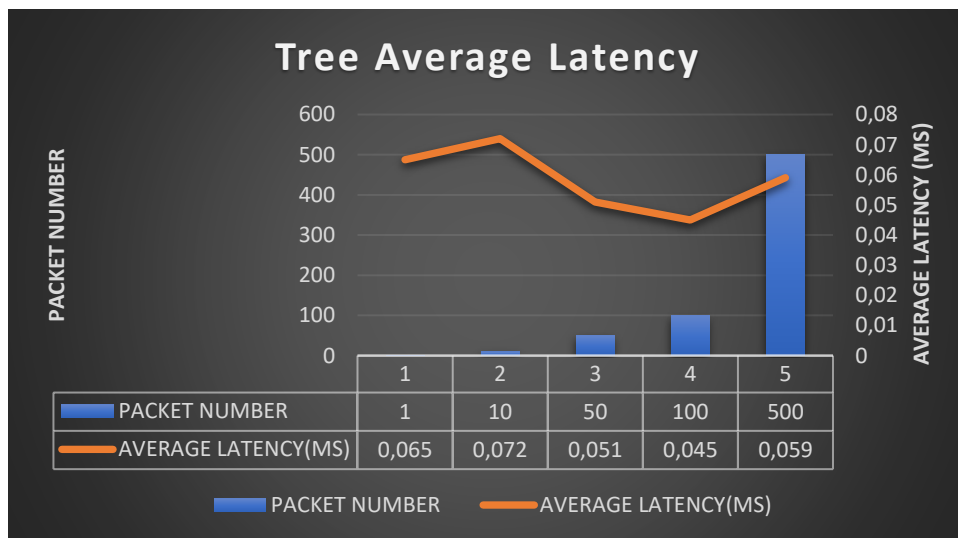
Στα Γραφήματα 5.12-5.15 παρουσιάζεται το διαθέσιμο Bandwidth για κάθε τοπολογία και καθώς ο αριθμός των κόμβων μεταβάλλεται. Η γενική διαπίστωση για όλες τις τοπολογίες είναι ότι το διαθέσιμο εύρος μειώνεται καθώς αυξάνεται το πλήθος των κόμβων. Στην τοπολογία Linear σχεδόν για το σύνολο των μετρήσεων, το διαθέσιμο εύρος παραμένει σταθερό. Αυτό οφείλεται στον σταθερό αριθμό των switches της συγκεκριμένης τοπολογίας. Στις άλλες δυο τοπολογίες, η αύξηση στον αριθμό των switches, συνεπάγεται και αντίστοιχη μείωση στο διαθέσιμο Bandwidth.



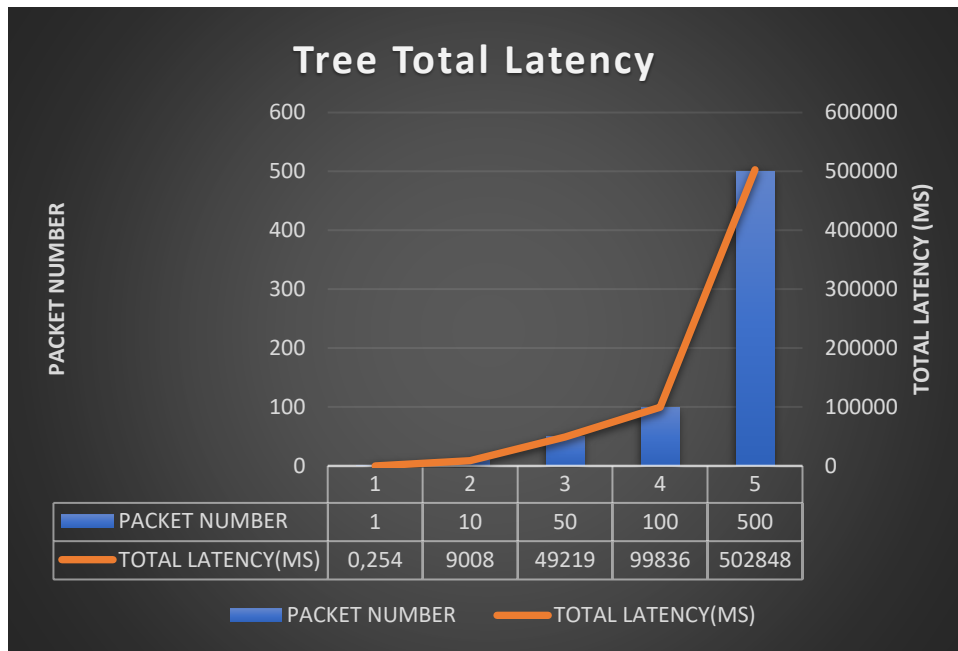
Γράφημα 5.15: Bandwidth (Gbps) ως προς το ίδιο αριθμό κόμβων στις εξομοιώσεις συγκριτικά και για τις τρεις τοπολογίες.

Συγκρίνοντας και τις τρεις τοπολογίες στον ίδιο συνολικό αριθμό κόμβων παρατηρούμε ότι το Bandwidth έχει παρόμοιες τιμές και για τις τρεις τοπολογίες με μερικές τάσεις για ανομοιόμορφη αυξομείωση στην τοπολογία Linear λόγω του μεγαλύτερου φόρτου της σε Hosts και μικρό αριθμό Switch. Στην τοπολογία Tree υπάρχει γραμμική πτώση ανάλογα με τον φόρτο της και στην τοπολογία Erdos Renyi υπάρχουν ανομοιόμορφες αυξομειώσεις λόγω του ότι λειτουργεί πιο κοντά σε ένα πραγματικό δίκτυο και υπάρχουν αυξημένες πιθανότητες για ανεπιτυχή ή το κυριότερο δυσκολότερη επικοινωνία μεταξύ των κόμβων.

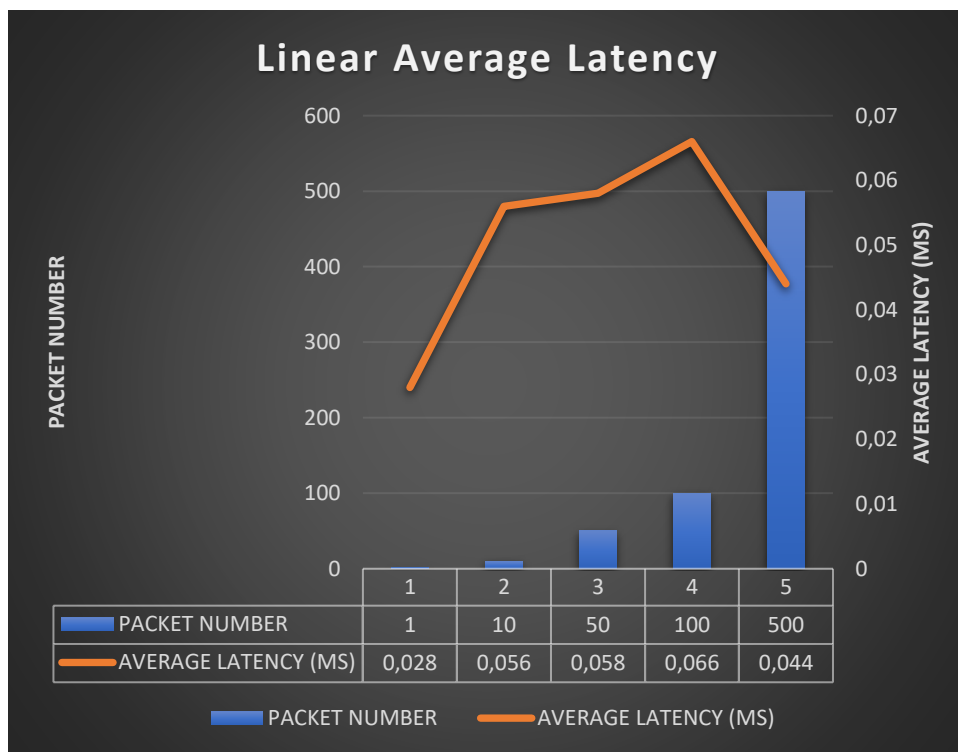
### 5.6.5 Latency



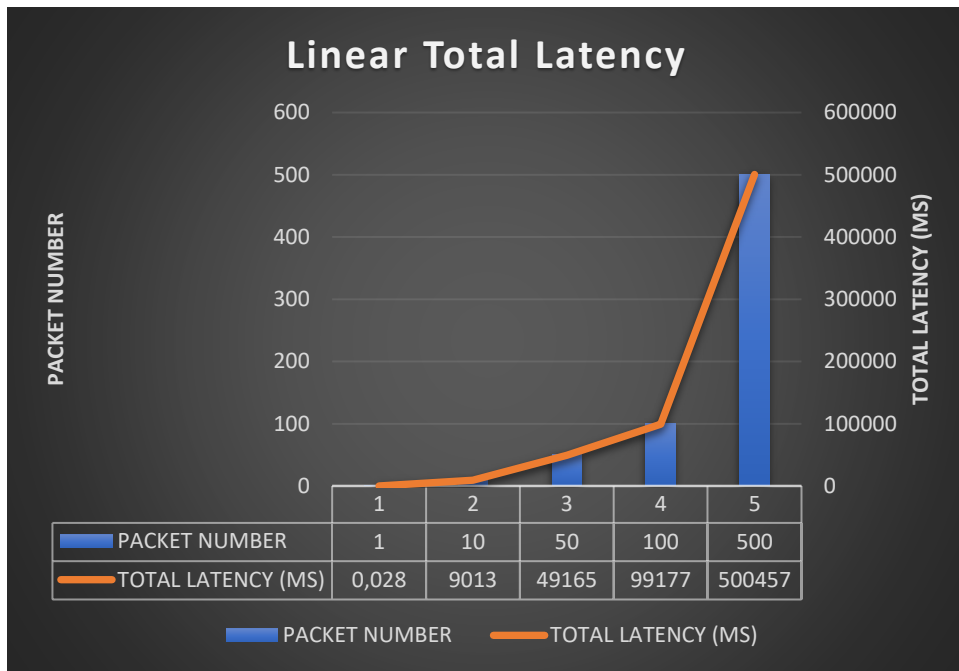
Γράφημα 5.16: Μέση καθυστέρηση τοπολογίας Tree.



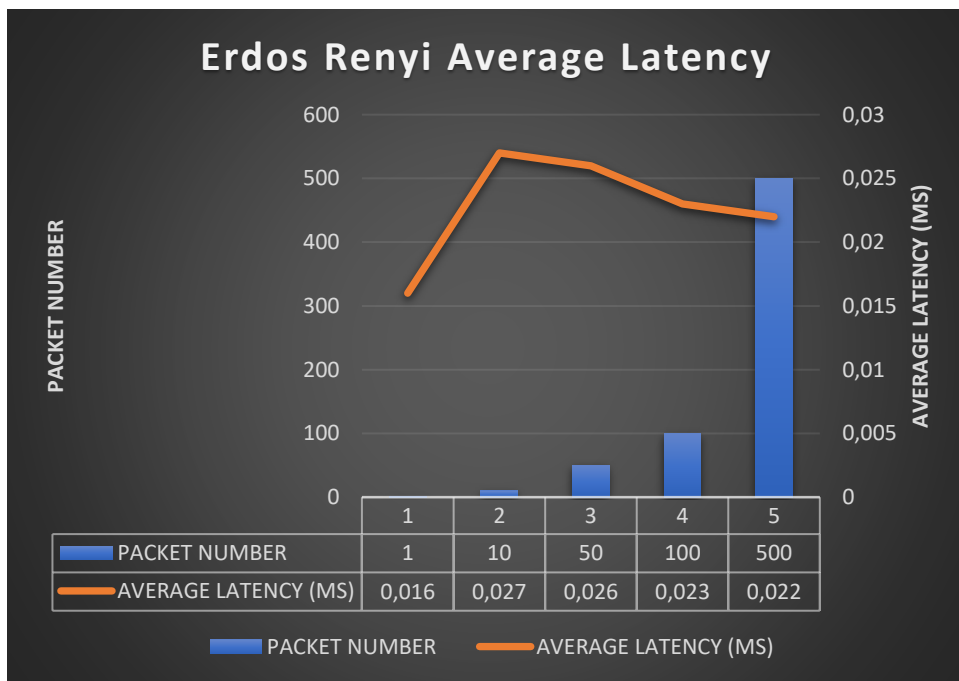
Γράφημα 5.17:Συνολική καθυστέρηση τοπολογίας Tree.



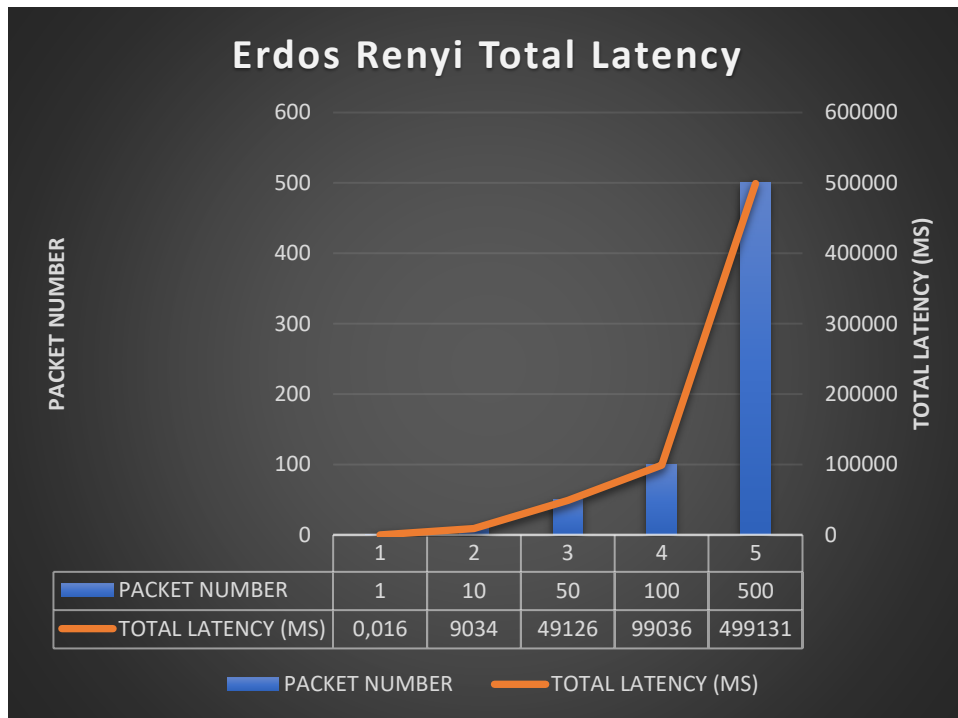
Γράφημα 5.18:Μέση καθυστέρηση τοπολογίας Linear.



Γράφημα 5.19:Συνολική καθυστέρηση τοπολογίας Linear.



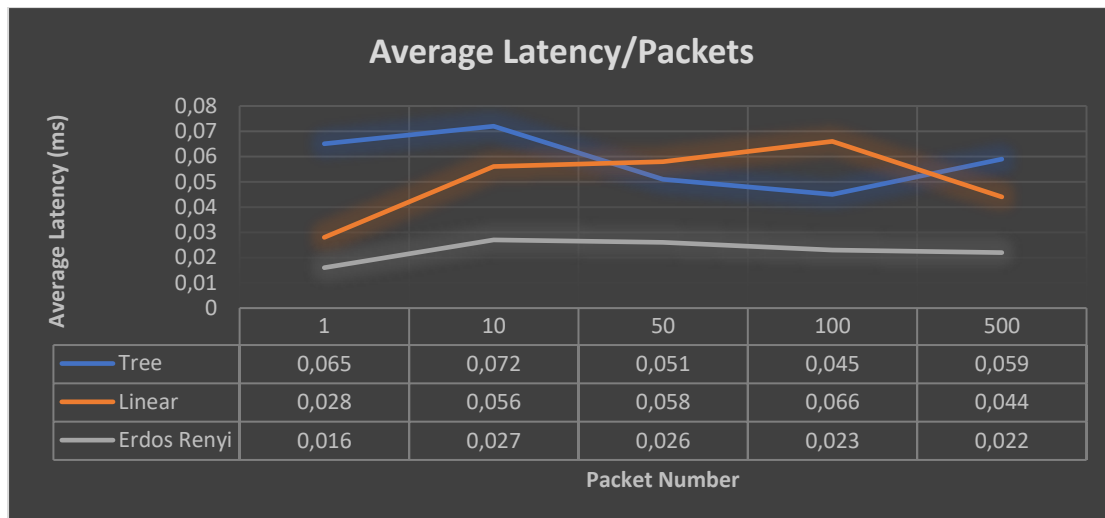
Γράφημα 5.20:Μέση καθυστέρηση τοπολογίας Erdos Renyi.



Γράφημα 5.21:Συνολική καθυστέρηση τοπολογίας Erdos Renyi.

Στα Γραφήματα 5.16, 5.18, 5.20 και 5.17, 5.19, 5.21 απεικονίζονται το Latency κατά μέσο όρο και αθροιστικά για τις τρεις τοπολογίες αντίστοιχα. Η μέση καθυστέρηση στην τοπολογία Tree κυμαίνεται γύρω από μια μέση τιμή (0.06μs), στην τοπολογία Linear κυμαίνεται επίσης γύρω από μια μέση τιμή (0.05μs) αλλά παρατηρούνται μεγαλύτερες αποκλίσεις και τέλος, στην τοπολογία Erdos Renyi κυμαίνεται γύρω από μια μέση τιμή (0.023μs. Στην τοπολογία Erdos Renyi παρατηρείται η μικρότερη μέση καθυστέρηση η οποία απλά και μόνο στη μεταφορά του ενός πακέτου είναι αρκετά μεγαλύτερη για αυτό και αυτή η μεταβολή στην αρχή της καμπύλης αλλά μετά το δίκτυο προσαρμόζεται όσες και να είναι οι απαιτήσεις από και προς αυτό. Τέλος σε ότι αφορά τη συνολική καθυστέρηση αυτή είναι πανομοιότυποι στις τοπολογίες Tree και Linear και σταθερά μικρότερη στην Erdos Renyi όσο αυξάνεται ο φόρτος. Πρέπει όμως να ληφθεί υπόψιν ότι η Erdos Renyi εμπεριέχει περισσότερα στοιχεία της πραγματικής τοπολογίας και για αυτό παρότι είναι εφικτές μεγαλύτερες ταχύτητες μετάδοσης δεδομένων είναι αυξημένη η πιθανότητα λάθους ή σφάλματος σε αντίθεση με τις άλλες δύο τοπολογίες και ειδικά την Tree η οποία αγγίζει κατά προσέγγιση το θεωρητικά τέλειο και θεωρείται ως πρότυπο στα δίκτυα.





Γράφημα 5.22: Μέση καθυστέρηση συγκριτικά και για τις τρεις τοπολογίες.

Συγκρίνοντας τη μέση καθυστέρηση και για τις τρεις τοπολογίες παρατηρείται μείωση της καθυστέρησης μετάδοσης πακέτων στην τοπολογία Erdos Renyi ακριβώς λόγω των custom χαρακτηριστικών της ενώ στις άλλες δύο τοπολογίες πρέπει ο ελεγκτής πρώτα να λάβει κάποια χαρακτηριστικά λειτουργίας που υπάρχουν μέσα στο Mininet στην περίπτωση της custom Erdos Renyi αυτά αποκτώνται αμεσότερα.

## 6.Συμπεράσματα

Τα δίκτυα SDN αποτελούν μια τεχνολογία που το κύριο χαρακτηριστικό της είναι ο διαχωρισμός του επιπέδου ελέγχου από το επίπεδο των δεδομένων. Βασίζονται στο πρωτόκολλο OpenFlow και η δρομολόγηση των δεδομένων αποφασίζεται κεντρικά από τον ελεγκτή που έχει εικόνα για την τοπολογία του δικτύου σε αντίθεση με τον τρόπο λειτουργίας των παραδοσιακών δικτύων.

Τα δίκτυα SDN αποτελούν το μέλλον και σίγουρα είναι η ταχύτερα εξελισσόμενη τεχνολογία δικτύων παρόλα αυτά μένει ακόμα αρκετή δουλειά μέχρι να εδραιωθεί αυτή η νέα τεχνολογία. Αναμένεται στην δεκαετία που διανύουμε να αρχίσουν τα δίκτυα τύπου SDN να εξαφανίζονται από τον χάρτη τα παραδοσιακά δίκτυα. Για να υλοποιηθεί όμως αυτό θα πρέπει και οι αντίστοιχοι φορείς και οι εργαζόμενοι τους να εκπαιδευτούν αντίστοιχα και να μάθουν να χειρίζονται και να αντιμετωπίζουν αυτή τη νέα τεχνολογία.

Σήμερα, είναι εφικτό με τη χρήση διαφόρων εργαλείων, όπως για παράδειγμα το Mininet που χρησιμοποιήθηκε σε αυτή την εργασία, να σχεδιαστούν, να υλοποιηθούν και να δοκιμαστούν τέτοιου είδους δίκτυα. Στα πλαίσια αυτής της εργασίας χρησιμοποιήθηκαν βασικές μετρήσεις σχετικές με τη χρήση της CPU, το Bandwidth, το Latency κ.ά. για να παρακολουθεί και να παρατηρηθεί η συμπεριφορά αυτών των δικτύων για διάφορες δικτυακές τοπολογίες.

Συνοψίζοντας στην παρούσα πτυχιακή εργασία πέρα από την ανάλυση των προδιαγραφών των δικτύων και τη μελέτη των προδιαγραφών τους, μέσω των εξομοιώσεων αποδείχθηκε η ευκολία και η αποτελεσματικότητα στη χρήση τους καθώς και το πόσα πλεονεκτήματα αποφέρουν σε σχέση με τα παραδοσιακά δίκτυα δίνοντας τη δυνατότητα για εκτεταμένες επιχειρησιακές και ερευνητικές ενέργειες κάνοντας τεράστια εξοικονόμηση ανθρωπίνων και υλικών πόρων.

## 7.Βιβλιογραφικές αναφορές

- [1] Advantages of SDN & disadvantages of SDN.Ανακτήθηκε(Φεβρουάριος 2021) από <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-SDN.html>
- [2] Aria ,A(OpenFlow Switch: What Is It and How Does it Work? (2018,30 Ιουλίου).
- [3] Arora, H., 2017. Software Defined Networking (SDN)-Architecture and role of OpenFlow.
- [4] Baukes M., 2020, 6 Big Influencers in Software Defined Networking.
- [5] Benefits of Optical Layer Restoration .Eisenach , R(2017,5 Σεπτεμβρίου).
- [6] Best practices for content delivery. Ανακτήθηκε(Φεβρουάριος 2021) από <https://cloud.google.com/cdn/docs/best-practices>
- [7] Brown, P .SDN Research Report (2019).
- [8] Chaitanya, S Computer Network Topology – Mesh, Star, Bus, Ring and Hybrid(2020,5 Οκτωβρίου).
- [9] Cloud CDN overview.Ανακτήθηκε (Φεβρουάριος 2021) από <https://cloud.google.com/cdn/docs/overview>
- [10] Cloud CDN:Ανακτήθηκε (Φεβρουάριος 2021) από <https://cloud.google.com/cdn>
- [11] Cooney, M. What is SDN and where software-defined networking is going(2019,16 Απριλίου).
- [12] Curtis,B.WHAT IS SOFTWARE-DEFINED NETWORKING? AND ITS FEATURES.
- [13] Difference between Control Plane and Data Plane(2020,18 Αυγούστου).Ανακτήθηκε από <https://www.geeksforgeeks.org/difference-between-control-plane-and-data-plane/>
- [14] External HTTP(S) Load Balancing overview.Ανακτήθηκε (Φεβρουάριος 2021) από <https://cloud.google.com/load-balancing/docs/https>
- [15] Flowgrammable .Ανακτήθηκε (Φεβρουάριος 2021) από <http://flowgrammable.org>
- [16] Hopcroft, J (2006,23 Ιανουαρίου ).Large Graphs.
- [17] How SDN works.Ανακτήθηκε(Φεβρουάριος 2021 από) <https://www.networkurge.com/2017/06/how-sdn-works.html>
- [18] How to enable Virtualization (VT) on Windows 10 for BlueStacks (2020,Οκτώβρης).
- [19] How to Install Python Pip on Ubuntu 20.04(2020,27 Απριλίου.Ανακτήθηκε από <https://linuxize.com/post/how-to-install-pip-on-ubuntu-20.04/>
- [20] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M. and Zolla, J., 2013. B4: Experience with a globally-deployed software defined WAN. ACM SIGCOMM Computer Communication Review, 43(4), pp.3-14.
- [21] Johnson, J .Python Numpy Tutorial (with Jupyter and Colab).
- [22] Kleymann ,B. Top Five Apps and Services That Can Benefit from SDN(2016,31 Μαρτίου).
- [23] Koley, B. Software Defined Networking at Scale(2014).

- [24] Lessing, M. Why Is Software-Defined Networking (SDN) Important? (2019, 15 Οκτωβρίου.)
- [25] List of SDN controller software. Ανακτήθηκε από [https://en.wikipedia.org/wiki/List\\_of\\_SDN\\_controller\\_software](https://en.wikipedia.org/wiki/List_of_SDN_controller_software)
- [26] Menth, M. Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices (2014, Μάιος).
- [27] Mininet. Ανακτήθηκε (Φεβρουάριος 2021) από <https://github.com/mininet/mininet>
- [28] Nadeau, T, Gray K. SDN: Software Defined Networks
- [29] NetworkX Developers (2020, 22 Αυγούστου). Ανακτήθηκε από <https://networkx.org/documentation/stable/install.html>
- [30] Openflow Protocol. Ανακτήθηκε (Φεβρουάριος 2021) από <https://www.sciencedirect.com/topics/computer-science/openflow-protocol>
- [31] Open-Source Routing and Network Simulation (2014, 3 Δεκεμβρίου).
- [32] Open-Source Routing and Network Simulation (2015, 1 Ιουνίου).
- [33] Orenda. How Will SDN Change the Future Network? (2017, 9 Φεβρουαρίου).
- [34] Pros And Cons Of Software-Defined Networking (2018, 21 Ιανουαρίου).
- [35] Scott, L. Custom Mininet Topologies and Introducing Atom (2016, 29 Ιουνίου).
- [36] SDN Fun. Ανακτήθηκε (Φεβρουάριος 2021) από <https://www.grotto-networking.com/SDNfun.html>
- [37] Software Defined Networking. (2013, 30 Αυγούστου).
- [38] Software-Defined Networking (SDN) Definition. Ανακτήθηκε (Φεβρουάριος 2021) από <https://opennetworking.org/sdn-definition/>
- [39] Software-Defined Networking. Ανακτήθηκε (Φεβρουάριος 2021) από <https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html>
- [40] Spengler, D. Understanding and using htop to monitor system resources (2012, 20 Δεκεμβρίου).
- [41] Top SDN Companies. Ανακτήθηκε (Φεβρουάριος 2021) από <https://www.ventureradar.com/keyword/SDN>
- [42] Understanding the SDN Architecture - SDN Control Plane & SDN Data Plane. (2015, 13 Μαρτίου).
- [43] Useful Mininet Setups. Ανακτήθηκε (Φεβρουάριος 2021) από <http://sdnhub.org/resources/useful-mininet-setups/>
- [44] Using Cloud CDN. Ανακτήθηκε (Φεβρουάριος 2021) από <https://cloud.google.com/cdn/docs/using-cdn>
- [45] Using the POX SDN controller. (2015, 20 Απριλίου).
- [46] Verlrajan, S. List of OpenFlow Controller for SDN (2019, 23 Απριλίου).
- [47] VirtualBox Specific Instructions. Ανακτήθηκε (Φεβρουάριος 2021) από <https://github.com/mininet/openflow-tutorial/wiki/VirtualBox-specific-Instructions>
- [48] What is cpu virtualization. Ανακτήθηκε (Φεβρουάριος 2021) από <https://geek-university.com/vmware-esxi/what-is-cpu-virtualization/>

- [49] What is floodlight controller. Ανακτήθηκε (Φεβρουάριος 2021) από <https://opennetworking.org/sdn-resources/sdn-products/big-switch-networks-big-network-controller/>
- [50] What is SDN?(2019,Δεκέμβριος).
- [51] What is Software-Defined Networking (SDN)?Ανακτήθηκε(Φεβρουάριος 2021)από <https://www.ibm.com/services/network/sdn-versus-traditional-networking>
- [52] Wu , J. A Study on the Dependability of Software Defined Networks(Ιανουάριος 2015).
- [53] Xming - Manage graphical Linux apps from a Windows system(2019,9 Δεκεμβρίου).
- [54] <https://www.virtualbox.org>
- [55] <http://www.straightrunning.com/XmingNotes/>
- [56] <https://www.chiark.greenend.org.uk/~sgtatham/putty/>
- [57] <https://gephi.org>
- [58] <https://sourceforge.net/projects/xming/>
- [59] <https://winscp.net/eng/index.php>
- [60] [https://linuxhint.com/update\\_all\\_packages\\_ubuntu/](https://linuxhint.com/update_all_packages_ubuntu/)
- [61] <https://www.ciena.com/insights/what-is/What-Is-SDN.html>
- [62]<https://stackoverflow.com/questions/43159525/how-do-i-set-the-packet-size-in-bytes-when-i-ping-host-from-another>
- [63][https://github.com/jliendo/mn\\_nx\\_topos/blob/master/mn\\_nx\\_topos.py](https://github.com/jliendo/mn_nx_topos/blob/master/mn_nx_topos.py)
- [64]<https://www.cambridgewireless.co.uk/news/2020/jan/28/top-providers-sdn-services/>
- [65] <https://en.wikipedia.org/wiki/OpenFlow>
- [66]<https://tex.stackexchange.com/questions/229536/draw-very-basic-network-topology-using-tikz>
- [67][https://networkx.org/documentation/stable/auto\\_examples/graph/plot\\_erdos\\_renyi.html](https://networkx.org/documentation/stable/auto_examples/graph/plot_erdos_renyi.html)

## 7.1 Παράρτημα

Κώδικας που περιέχεται στο αρχείο mn\_nx\_topos.py . [\[63\]](#)

```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

from mininet.topo import Topo

class NxTopo ( Topo ):
    """Super class for basic NxTopo actions: switch/host add and
    graphing.
    All networkx based topologies must derive from this super
    class"""
    def __init__(self):
        # super
        super(NxTopo, self).__init__()

    def build_nx_topo(self, g):
        """
        1) Create switches, hosts...attach host to switch according to
        nx topo (self.ref_g)
        2) Add one host to each switch (h0->s0, h1->s1, etc.)
        3) Add edges between switches
        """
        # create switches and attach a host to the switch
        # one host per switch
        for n in g.nodes():
            n = n + 1
            self.addSwitch('s%d' % n)
            self.addHost('h%d' % n)
            self.addLink("s%d" % n, "h%d" % n)
        # link switches according to topology
        for (n1, n2) in g.edges():
            n1 = n1 + 1
            n2 = n2 + 1
            self.addLink("s%d" % n1, "s%d" % n2)

    def graph(self):
        """
        Graph function. Has to be overloaded. For graphing purposes
        it can use self.ref_g which is mn_nx's view of the topology
        (this view has to match mininet's view, less the hosts) or
        it can also use self.g which is mininet's topology view
        """
        pass

class BalancedTree( NxTopo ):
    """networkx based BalancedTree topology.
    r = range (fanout), h = height (depth) """
    def __init__(self, **kwargs):
        # super
        super(BalancedTree, self).__init__()
        r = kwargs.get('r', 2)
```

```

    h = kwargs.get('h', 2)
    # mn_nx's topology view of the network
    self.ref_g = nx.balanced_tree(r,h)
    # nx topology definition
    self.build_nx_topo(self.ref_g)

def graph(self):
    """
    Overloaded graph function to draw a hierachical view of
    tree-like topologies. Switches are red and there are no
    hosts on the graph.
    In reality (mininet virtual network) there exists a host
    for each switch in the topology with a name that starts
    with 'hn' where 'n' is the switch number.
    """
    pos = nx.graphviz_layout(self.ref_g, prog='dot')
    nx.draw(self.ref_g, pos)
    plt.show()

class ErdosRenyi( NxTopo ):
    """networkx based Erdos-Renyi topology.
       n = Number of nodes, p = probability of an edge
       between nodes """
    def __init__(self, **kwargs):
        # super
        super(ErdosRenyi, self).__init__()
        n = kwargs.get('n', 5)
        p = kwargs.get('p', 0.8)
        # topology view of the network
        self.ref_g = nx.erdos_renyi_graph(n,p)
        # nx topology definition
        self.build_nx_topo(self.ref_g)

    def graph(self):
        pos = nx.circular_layout(self.ref_g)
        nx.draw(self.ref_g, pos)
        plt.show()

topos = { 'balanced_tree': (lambda **args: BalancedTree(**args)),
          'erdos_renyi'   : (lambda **args: ErdosRenyi(**args)) }

```