

Μελέτη Σχεδίαση και Προσομοίωση Ολοκληρωμένων Δικτυακών Συστημάτων
με χρήση IPv4 και IPv6 με την χρήση ανοιχτών λογισμικών (Open Source) με
στόχο την ενοποίηση πρωτοκόλλων επικοινωνιών (Integrated Communication
Protocols) με πεδίο εφαρμογής το Έξυπνο Ενεργειακό Δίκτυο

Η Διπλωματική Εργασία
παρουσιάστηκε ενώπιον
του Διδακτικού Προσωπικού του
Πανεπιστημίου Αιγαίου

Των

Γιαννιώτη Χρήστου

Πορή Δημήτριου

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2018



Η ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΔΙΔΑΣΚΟΝΤΩΝ ΕΠΙΚΥΡΩΝΕΙ

ΤΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Των

Γιαννιώτη Χρήστου

Πορή Δημήτριου

Σκιάνης Χαράλαμπος, Επιβλέπων

Τμήμα Μηχανικών Πληροφοριακών και

Επικοινωνιακών Συστημάτων

Βουγιούκας Δημοσθένης, Μέλος

Τμήμα Μηχανικών Πληροφοριακών και

Επικοινωνιακών Συστημάτων

Σκούτας Δημήτριος, Μέλος

Τμήμα Μηχανικών Πληροφοριακών και

Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2018





ΠΕΡΙΛΗΨΗ

Στη σύγχρονη κοινωνία που χαρακτηρίζεται από ραγδαίες εξελίξεις σε διάφορους τομείς, καθοριστικός θεωρείται και ο ρόλος των δικτύων υπολογιστών στην καθημερινότητα και στον τομέα των επιχειρήσεων. Αξίζει να σημειώσουμε την μεγάλη ευελιξία και ισχύ του διαδικτύου στην σημερινή εποχή. Η επικοινωνία έχει απλουστευθεί σε σημαντικό βαθμό (viber, messenger, line), οι χρηματικές δοσοληψίες παρουσιάζουν ταχύτατο ρυθμό (χρεωστικές κάρτες, web banking), η συλλογή και αναδιανομή τεράστιου όγκου πληροφορίας εκτελείται με μεγάλη ευκολία και σε πολύ μικρό χρονικό διάστημα (αποθηκευτικά νέφη). Κατά συνέπεια η έρευνα της συμπεριφοράς ενός υπολογιστικού δικτύου αλλά και η γνώση των πρωτοκόλλων που καθορίζουν την επικοινωνία των κόμβων που το απαρτίζουν είναι μεγάλης σημασίας διότι προσφέρει εύφορο έδαφος για αναβαθμίσεις στις ήδη υπάρχουσες τεχνολογίες αλλά και μια εκτενέστερη ανασκόπηση του αντικειμένου. Η τεχνολογία των δικτύων και των επικοινωνιών παρουσιάζεται σαν ένα από τα πιο σημαντικά και γεμάτα ανάπτυξη μέρη του τεχνολογικού κλάδου. Στο εγγύς μέλλον η εφαρμογή των 5G δικτύων εκτός από την ταχύτητα στη μεταφορά των δεδομένων έχει και ως κύριο μέλημα της τη διασύνδεση συστημάτων (Integration). Με τον όρο αυτό αναφερόμαστε στην ενοποίηση πολλών διαφορετικών συστημάτων τα οποία θα είναι σε θέση να συνεργαστούν μεταξύ τους έτσι ώστε να επιτύχουμε το επιθυμητό αποτέλεσμα. Επιπλέον η συνύπαρξη αυτή εμφανίζει ετερογένεια σε ένα δίκτυο εφόσον πρέπει ταυτοχρόνως να συνδυάζουμε υλικό με διάφορα πρωτόκολλα επικοινωνίας. Τις περισσότερες φορές βέβαια η παρατήρηση ενός αληθινού δικτύου καθίσταται αδύνατη είτε για λόγους ασφαλείας, που είναι αρκετά σημαντικό, είτε γιατί η αναζήτηση επιρόσθετων πληροφοριών για λόγους διαχείρισης δημιουργεί περαιτέρω φορτίο κίνησης. Εν κατακλείδι δημιουργείται επιτακτική ανάγκη για την εύρεση μεθόδων για την υλοποίηση υποδομών που θα έχουν παρόμοια συμπεριφορά με ένα real time δίκτυο και θα είναι εύκολα παραμετροποιήσιμες με χαμηλό κόστος. Στο 5^ο κεφάλαιο θα αναφερθούμε σε μία τεχνική που μας δίνει τη δυνατότητα να επιτύχουμε όλα τα παραπάνω και λέγεται εικονικοποίηση δικτύου.

© 2018

Των

Γιαννιώτη Χρήστου

Πορή Δημήτριου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ



ABSTRACT

In today's society characterized by rapid developments in various areas, the role of computer networks in everyday life and business is also crucial. It is worth noting the great flexibility and power of the internet in today's era. Communication has been greatly simplified (viber, messenger, line), money transactions are fast-paced (debit cards, web banking), the collection and redistribution of huge amounts of information is performed with great ease and in a very short time (storm clouds) .Consequently, research into the behavior of a computing network, as well as knowledge of the protocols defining the communication of its nodes, is of great importance because it offers fertile ground for upgrading to existing technologies and a more extensive review of the subject. Network and communications technology is emerging as one of the most important and full-fledged parts of the technology industry. In the near future, the implementation of 5G networks in addition to the speed of data transfer also has as its main concern Integration. By this we mean the integration of many different systems that will be able to cooperate with each other so that we achieve the desired result. In addition, this coexistence is heterogeneous in a network, since we must simultaneously combine material with various communication protocols. Most of the time, however, the observation of a true network becomes impossible either for safety reasons, which is quite important, or because the search for additional information for management reasons creates further traffic load. In conclusion, it is imperative to find methods for implementing infrastructures that will have a similar behavior to a real time network and will be easily configurable at low cost. In Chapter 5 we will discuss a technique that enables us to achieve all of the above and is called network virtualization.

© 2018

Gianniotis Christos

Poris Dimitrios

Department of Information and Communication Systems Engineering

UNIVERCITY OF THE AEGEAN



ΕΥΧΑΡΙΣΤΙΕΣ – ΑΦΙΕΡΩΣΕΙΣ

Αρχικά θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα της διπλωματικής μας, Πρόεδρο του Τμήματος κ. Χαράλαμπο Σκιάνη για την μεγάλη βοήθεια που μας παρείχε αλλά και για την ευκαιρία που μας έδωσε να κάνουμε την πρακτική μας άσκηση στα εργαστήριά του, μέσα από την οποία εμπνευστήκαμε το θέμα της διπλωματικής που εκπονήσαμε. Ακόμη θα θέλαμε να ευχαριστήσουμε τον υποψήφιο διδάκτορα κ. Αγγελή Νικόλαο για την πολύτιμη βοήθειά του και την συνεχή καθοδήγησή του καθ' όλη την διάρκεια συγγραφής της διπλωματικής μας. Τέλος ευχαριστούμε αλλά και αφιερώνουμε την εργασία μας αυτήν στους γονείς μας Ιωάννα Καλατζή, Αθανάσιο Γιαννιώτη και Δωροθέα Γουλιώτη, Σάββα Πορή καθώς και τα αδέρφια μας Βασίλειο Γιαννιώτη, Αικατερίνη Πορή για την αγάπη και την υποστήριξή τους όλα αυτά τα χρόνια.



Δομή Διπλωματικής Εργασίας

Η Διπλωματική μας εργασία χωρίζεται σε δύο μέρη όπου στο πρώτο μέρος παρουσιάζεται το θεωρητικό υπόβαθρο που μελετήθηκε για την ευρύτερη κατανόηση των εννοιών της δικτύωσης. Στο δεύτερο μέρος παρατίθεται η μελέτη που εκπονήθηκε στα Open source λογισμικά. Δίνεται βάση στον τρόπο λειτουργίας και εγκατάστασης των λογισμικών και στη συνέχεια στην υλοποίηση τοπολογιών με χρήση διάφορων δικτυακών πρωτοκόλλων. Επίσης ιδιαίτερη σημασία δίνεται και στη διασύνδεση πλατφορμών μεταξύ τους καθώς και η χρήση εξωτερικών ελεγκτών που βοηθούν στο remote monitoring των μελλοντικών δικτυακών συσκευών. Πιο συγκεκριμένα το Μέρος Α απαρτίζεται από πέντε κεφάλαια βιβλιογραφικής αναφοράς των σημαντικότερων εννοιών της διπλωματικής μας εργασίας και το Μέρος Β από άλλα πέντε κεφάλαια που υλοποιήθηκαν διάφορες προσομοιώσεις και συνπροσομοιώσεις (co-simulations) με την χρήση των Open source λογισμικών.

© 2018

Των

Γιαννιώτη Χρήστου

Πορή Δημήτριου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ



Κατάλογος Περιεχομένων

| | |
|--|----|
| 1 Βιβλιογραφική αναφορά στις Έννοιες της Δικτύωσης-Αρχιτεκτονική Δικτύων και Μοντέλων Σχεδίασης..... | 26 |
| 1.1 ΒΑΣΙΚΕΣ ΈΝΝΟΙΕΣ ΔΙΚΤΥΩΣΗΣ..... | 26 |
| 1.2 ΒΑΣΙΚΕΣ ΤΟΠΟΛΟΓΙΕΣ ΔΙΚΤΥΩΣΗΣ | 28 |
| 1.2.1 Τοπολογία Διαύλου(Bus Topology)..... | 29 |
| 1.2.2 Τοπολογία Αστέρα (Star Topology)..... | 30 |
| 1.2.3 Τοπολογία Δακτυλίου (Ring Topology)..... | 31 |
| 1.2.4 Τοπολογία Δέντρου(Tree Topology)..... | 32 |
| 1.2.5 Υβριδική Τοπολογία(Hybrid Topology) | 33 |
| 1.3 ΒΑΣΙΚΟΙ ΤΥΠΟΙ ΔΙΚΤΥΩΣΗΣ | 34 |
| 1.3.1 Δίκτυα Προσωπικής Περιοχής (PAN)..... | 34 |
| 1.3.2 Τοπικά Δίκτυα (Local Area Networks, LAN) | 35 |
| 1.3.3 Ασύρματα Τοπικά Δίκτυα (Wireless Local Area Networks, WLAN)..... | 36 |
| 1.3.4 Εικονικά Τοπικά Δίκτυα (Virtual Local Area Networks, VLAN)..... | 38 |
| 1.3.5 Δίκτυα Ευρείας Περιοχής (WAN)..... | 39 |
| 1.3.5.1 Βασικές Αρχιτεκτονικές των Δικτύων Ευρείας Περιοχής (WAN)..... | 40 |
| 1.3.6 Μητροπολιτικά Δίκτυα (Metropolitan Area Networks, MAN)..... | 44 |
| 1.3.7 Οικιακό Δίκτυο (Home Area Network ,HAN) | 44 |
| 1.3.8 Near me Area Networks (NAN) | 45 |
| 1.3.9 Field Area Network (FAN)..... | 46 |
| 1.4 ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΔΙΚΤΥΟΥ | 46 |
| 1.4.1 Το μοντέλο αναφοράς OSI..... | 53 |
| 2 Διευθυνσιοδότηση Internet Protocol έκδοση 6 (IPv6)..... | 58 |
| 2.1 ΠΡΩΤΟΚΟΛΛΟ IP | 58 |
| 2.1.1 TCP/IP-UDP..... | 58 |
| 2.1.1.1 Πρωτόκολλο TCP - Δομή πακέτου..... | 59 |
| 2.1.1.2 Πρωτόκολλο UDP - Δομή πακέτου | 63 |
| 2.2 ΔΙΕΥΘΥΝΣΙΟΔΟΤΗΣΗ IPV6 | 64 |
| 2.2.1 Μορφή IPV6 διευθυνσιοδότησης..... | 64 |
| 2.2.2 Δομή ενός πακέτου IPV6..... | 65 |
| 2.2.2.1 Επικεφαλίδες Επέκτασης..... | 67 |
| 2.3 ΛΟΓΟΙ ΜΕΤΑΒΑΣΗΣ ΣΤΗΝ IP VERSION 6..... | 67 |
| 2.3.1 Quality of Service (QoS) | 70 |



| | | |
|---------|--|-----|
| 2.4 | ΝΕΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ IPV6 | 70 |
| 2.4.1 | Τύποι Διευθύνσεων | 70 |
| 2.4.1.1 | Ειδικοί Τύποι Διευθύνσεων | 72 |
| 2.4.2 | Ανίχνευση Ίδιων Διευθύνσεων (Duplicate Address Detection, DAD) | 72 |
| 2.4.3 | Χρόνος ζωής μια διεύθυνσης IPV6 | 73 |
| 2.4.4 | Πρωτόκολλο ICMPv6..... | 73 |
| 2.4.5 | Router Discovery | 74 |
| 2.4.6 | Εντοπισμός γειτόνων | 75 |
| 3 | Μελέτη Δικτυακών Πρωτοκόλλων | 76 |
| 3.1 | ETHERNET..... | 76 |
| 3.2 | ΕΙΣΑΓΩΓΗ ΣΤΗ ΔΡΟΜΟΛΟΓΗΣΗ..... | 77 |
| 3.2.1 | Στατική Δρομολόγηση..... | 78 |
| 3.2.2 | Δυναμική Δρομολόγηση..... | 78 |
| 3.2.3 | DHCP (Dynamic Host Configuration Protocol)..... | 78 |
| 3.2.4 | NAT (Network Address Translation) | 79 |
| 3.2.5 | Πρωτόκολλο Δρομολόγησης CDP (Cisco Discovery Protocol)..... | 80 |
| 3.3 | ΠΡΩΤΟΚΟΛΛΟ VTP | 81 |
| 3.4 | ΠΡΩΤΟΚΟΛΛΟ SPANNING TREE(STP)..... | 82 |
| 4 | Interior gateway protocols (IGP) | 84 |
| 4.1 | RIP (ROUTING INFORMATION PROTOCOL)..... | 85 |
| 4.2 | OSPF (OPEN SHORTEST PATH FIRST) | 86 |
| 4.2.1 | Autonomous Systems | 87 |
| 4.2.2 | Λειτουργία OSPF..... | 89 |
| 4.3 | INTERIOR GATEWAY ROUTING PROTOCOL (IGRP)..... | 90 |
| 4.3.1 | Χαρακτηριστικά του IGRP..... | 91 |
| 4.4 | ENHANCED INTERIOR GATEWAY ROUTING PROTOCOL (EIGRP)..... | 92 |
| 4.4.1 | Δομικά στοιχεία EIGRP πρωτοκόλλου | 94 |
| 4.4.2 | Λειτουργία του EIGRP | 97 |
| 4.4.2.1 | Μετρικές Πρωτοκόλλων | 97 |
| 4.5 | EXTERIOR GATEWAY PROTOCOLS (EGP) | 98 |
| 4.5.1 | Border Gateway Protocols (BGP)..... | 99 |
| 4.5.1.1 | Δομικά Στοιχεία του BGP..... | 99 |
| 4.5.1.2 | Λειτουργία του BGP | 99 |
| 5 | Εισαγωγή στην Εικονικοποίηση Δικτύων..... | 101 |
| 5.1 | ΧΡΗΣΗ ΥΛΙΚΟΥ (HARDWARE)..... | 102 |



| | | |
|-------|--|-----|
| 5.2 | ΧΡΗΣΗ ΠΡΟΣΟΜΟΙΩΣΗΣ ΣΕ ΥΠΟΛΟΓΙΣΤΙΚΟ ΣΥΣΤΗΜΑ..... | 102 |
| 5.3 | ΧΡΗΣΗ ΕΞΟΜΟΙΩΣΗΣ ΣΕ ΥΠΟΛΟΓΙΣΤΙΚΟ ΣΥΣΤΗΜΑ | 102 |
| 6 | Ενασχόληση με το Περιβάλλον του δικτυακού προσομοιωτή GNS3..... | 104 |
| 6.1 | ΔΙΑΔΙΚΑΣΙΑ ΔΗΜΙΟΥΡΓΙΑΣ-ΦΟΡΤΩΣΗΣ ΕΝΟΣ GNS3 PROJECT | 104 |
| 6.2 | ΒΑΣΙΚΗ ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ GNS3..... | 105 |
| 6.2.1 | Εκχώρηση IP Διευθύνσεων στην τοπολογία αστέρα..... | 107 |
| 6.2.2 | Πειράματα επιβεβαίωσης ορθής λειτουργίας του δικτύου | 109 |
| 6.3 | ΥΛΟΠΟΙΗΣΗ ΤΟΠΟΛΟΓΙΑΣ ΕΜΠΟΡΙΚΟΥ ΚΕΝΤΡΟΥ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑ ΜΕΤΑΞΥ ΤΩΝ ΥΠΟΔΙΚΤΥΩΝ ΤΗΣ | 110 |
| 6.3.1 | Τμήμα Φροντηστηρίου | 110 |
| 6.3.2 | Τμήμα Κινηματογράφου..... | 128 |
| 6.3.3 | Απομακρυσμένη πρόσβαση και υλοποίηση NAT | 131 |
| 6.3.4 | Κατάστημα ηλεκτρονικών ειδών..... | 133 |
| 6.3.5 | Καταστήματα Ρουχισμού (InterVlan τοπολογία) | 135 |
| 6.4 | ΔΗΜΙΟΥΡΓΙΑ ΔΙΚΤΥΟΥ ΠΡΟΣΟΜΟΙΩΣΗΣ ΕΝΟΣ ΜΙΚΡΟΜΕΣΑΙΟΥ ΟΡΓΑΝΙΣΜΟΥ | 140 |
| 6.4.1 | Υλοποίηση τεχνικής υποδικτύωσης VLSM(Subnetting) | 140 |
| 6.5 | ΥΛΟΠΟΙΗΣΗ ΤΟΠΟΛΟΓΙΑΣ ΟΡΓΑΝΙΣΜΟΥ | 145 |
| 6.5.1 | Επιβεβαίωση λειτουργίας | 169 |
| 6.6 | ΜΕΛΕΤΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΟΠΟΛΟΓΙΑΣ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΡΩΤΟΚΟΛΛΟΥ BGP ΣΕ ΣΥΝΕΡΓΑΣΙΑ ΜΕ ΤΟ EIGRP/OSPF ΜΕ ΧΡΗΣΗ GRE TUNNELING | 179 |
| 6.6.1 | Έλεγχος της κίνησης του tunnel διαμέσων του Wireshark..... | 194 |
| 6.7 | REDISTRIBUTION(ΑΝΑΚΑΤΑΝΟΜΗ) ΜΕΤΑΞΥ EGP ΚΑΙ IGP ΠΡΩΤΟΚΟΛΛΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ | 201 |
| 6.7.1 | Επαλήθευση λειτουργίας του δικτύου | 208 |
| 6.7.2 | Έλεγχος της κυκλοφορίας με το Wireshark..... | 211 |
| 7 | Μελέτη του Εξομοιωτή Mininet | 213 |
| 7.1 | ΠΕΡΙΓΡΑΦΗ ΤΟΥ MININET | 213 |
| 7.2 | ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ MININET | 213 |
| 7.3 | ΡΥΘΜΟΝ SCRIPTS ΠΟΥ ΥΛΟΠΟΙΗΘΗΚΑΝ..... | 221 |
| 7.4 | ΔΗΜΙΟΥΡΓΙΑ ΤΟΠΟΛΟΓΙΩΝ ΔΙΑΜΕΣΩΝ ΤΟΥ SCRIPT MINIEDIT..... | 225 |
| 7.4.1 | ARP αιτήσεις | 230 |
| 7.4.2 | Capture πακέτων διαμέσων του Wireshark | 231 |
| 8 | Integration των δικτυακών προσομοιωτών GNS3 & Mininet | 235 |
| 8.1 | ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ MININET | 235 |
| 8.2 | ΒΗΜΑΤΑ ΕΓΚΑΤΑΣΤΑΣΗΣ ΤΩΝ ΕΙΚΟΝΙΚΩΝ ΜΗΧΑΝΩΝ | 235 |



| | |
|--|-----|
| 8.3 ΔΗΜΙΟΥΡΓΙΑ ΠΡΩΤΗΣ ΤΟΠΟΛΟΓΙΑΣ ΔΙΑΣΥΝΔΕΣΗΣ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΟΥ NAT ΠΡΩΤΟΚΟΛΛΟΥ | 241 |
| 8.4 ΥΛΟΠΟΙΗΣΗ ΔΕΥΤΕΡΗ ΤΟΠΟΛΟΓΙΑΣ ΔΙΑΣΥΝΔΕΣΗΣ ΜΕΤΑΞΥ GNS3-MININET-OPENFLOWCONTROLLER..... | 252 |
| 8.4.1 Βήματα Integrate | 252 |
| 8.4.2 Προσθήκη εσωτερικού Browser..... | 258 |
| 8.5 ΠΕΙΡΑΜΑΤΑ ΠΟΥ ΥΛΟΠΟΙΗΘΗΚΑΝ ΣΤΟΝ ΔΙΑΣΥΝΔΕΔΕΜΕΝΟ ΜΗΧΑΝΙΣΜΟ | 263 |
| 8.5.1 1 ^ο πείραμα..... | 263 |
| 8.5.2 Προσθήκη Δρομολογητή που επικοινωνεί με τον διασυνδεδεμένο μηχανισμό ... | 272 |
| 8.5.2.1 Έλεγχος επικοινωνίας δρομολογητών με τα άλλα εικονικά μηχανήματα που έχουμε εισάγει..... | 274 |
| 8.5.2.2 Δημιουργία τοπολογίας στο Mininet και επικοινωνία των hosts με τους Δρομολογητές..... | 278 |
| 8.6 ΥΛΟΠΟΙΗΣΗ 3 ^{ΗΣ} ΤΟΠΟΛΟΓΙΑΣ ΔΙΑΣΥΝΔΕΔΕΜΕΝΟΥ ΜΗΧΑΝΙΣΜΟΥ | 283 |
| 8.6.1 Συνέχεια Τοπολογίας Προσθήκη Δρομολογητή που συνδέεται διαμέσων ενός άλλου με το Mininet και τους hosts..... | 290 |
| 8.6.2 Προσθήκη επιπλέον ζεύξης-δικτύου για πλήρη επικοινωνία R2-R3 | 296 |
| 8.6.2.1 Έλεγχος επικοινωνίας του δικτύου | 299 |
| 9 Εγκατάσταση και Βασική Παραμετροποίηση του πειραματικού δικτυακού εξομοιωτή Core 303 | |
| 9.1 ΕΙΣΑΓΩΓΗ ΣΤΟ CORE | 303 |
| 9.1.1 Αρχιτεκτονική του Core | 303 |
| 9.1.2 Τρόπος Λειτουργίας του Core | 305 |
| 9.2 ΟΔΗΓΙΕΣ ΚΑΙ ΒΗΜΑΤΑ ΕΓΚΑΤΑΣΤΑΣΗΣ ΤΟΥ CORE ΕΞΟΜΟΙΩΤΗ | 305 |
| 9.3 ΠΕΙΡΑΜΑΤΑ ΠΟΥ ΥΛΟΠΟΙΗΘΗΚΑΝ | 309 |
| 9.3.1 Υλοποίηση Σεναρίου Έξυπνων Σπιτιών Στη Σάμο | 310 |
| 9.3.1.1 Smart Enviroment | 310 |
| 9.3.1.2 Smart Homes..... | 311 |
| 9.3.1.3 Υλοποίηση σεναρίου στο Core | 311 |
| 9.3.2 Έξυπνα Σπίτια Καρλοβάσσου | 312 |
| 9.3.3 Έλεγχος επικοινωνίας των Smart Devices..... | 316 |
| 9.3.4 Capture Πακέτων διαμέσων του Wireshark | 324 |
| 9.4 ΈΞΥΠΝΑ ΣΠΙΤΙΑ ΒΑΘΥ | 325 |
| 9.4.1 Έλεγχος επικοινωνίας των Smart Devices..... | 328 |
| 9.5 ΈΞΥΠΝΑ ΣΠΙΤΙΑ ΠΥΘΑΓΟΡΕΙΟΥ | 331 |
| 9.5.1 Έλεγχος επικοινωνίας των Smart homes | 334 |



| | |
|--|-----|
| 10 Εγκατάσταση και βασική Παραμετροποίηση με τον πειραματικό δικτυακό εξομοιωτή IMUNES | 337 |
| 10.1 ΕΙΣΑΓΩΓΗ ΣΤΟ IMUNES | 337 |
| 10.2 ΤΟΠΟΛΟΓΙΑ ΤΟΥ IMUNES | 338 |
| 10.3 ΒΗΜΑΤΑ ΕΓΚΑΤΑΣΤΑΣΗΣ ΤΟΥ ΕΞΟΜΟΙΩΤΗ IMUNES..... | 341 |
| 10.4 ΤΟΠΟΛΟΓΙΑ ΔΙΑΣΥΝΔΕΣΗΣ IMUNES & GNS3 ΕΞΟΜΟΙΩΤΗ..... | 343 |
| 10.4.1 Βήματα Διασύνδεσης (Integration) | 343 |
| 10.4.2 Redistribution των δρομολογητών με BGP&OSPF πρωτόκολλο | 345 |
| 10.4.3 Πειράματα επικοινωνίας των δρομολογητών | 350 |
| 10.4.4 Τοπολογία έξυπνων σπιτιών στο περιβάλλον του IMUNES..... | 352 |
| 10.5 ΈΛΕΓΧΟΣ ΕΠΙΚΟΙΝΩΝΙΑΣ ΤΗΣ ΔΙΑΣΥΝΔΕΔΕΜΕΝΗΣ ΤΟΠΟΛΟΓΙΑΣ..... | 357 |
| 10.5.1 Wireshark Captures | 362 |
| 10.5.2 Μελλοντική Έρευνα | 363 |
| 11 Βιβλιογραφία..... | 364 |

Κατάλογος Εικόνων

| | |
|--|-----------|
| <i>Εικόνα 1-1 Point to Point Connection</i> | <i>26</i> |
| <i>Εικόνα 1-2 Full Mesh Network Topology</i> | <i>27</i> |
| <i>Εικόνα 1-3 One Hop Communication</i> | <i>28</i> |
| <i>Εικόνα 1-4 Τοπολογία Διαύλου (Bus Topology).....</i> | <i>30</i> |
| <i>Εικόνα 1-5 Τοπολογία Αστέρα (Star Topology).....</i> | <i>31</i> |
| <i>Εικόνα 1-6 Τοπολογία Δακτυλίου (Ring Topology).....</i> | <i>32</i> |
| <i>Εικόνα 1-7 Τοπολογία Δέντρου (Tree Topology).....</i> | <i>32</i> |
| <i>Εικόνα 1-8 Υβριδική Τοπολογία (Hybrid Topology)</i> | <i>33</i> |
| <i>Εικόνα 1-9 Δίκτυα Προσωπικής Περιοχής (PAN)</i> | <i>34</i> |
| <i>Εικόνα 1-10 Δίκτυα Τοπικής Περιοχής (LAN).....</i> | <i>36</i> |
| <i>Εικόνα 1-11 Ασύρματα Τοπικά Δίκτυα (WLAN).....</i> | <i>37</i> |
| <i>Εικόνα 1-12 Ασύρματο δίκτυο τοπολογίας ad-hoc με παρουσία βλάβης.....</i> | <i>37</i> |



| | |
|---|-----|
| <i>Εικόνα 1-13 Εικονικά Τοπικά Δίκτυα (VLAN)</i> | 38 |
| <i>Εικόνα 1-14 LAN δίκτυο συνδεδεμένο σε WAN</i> | 39 |
| <i>Εικόνα 1-15 Δίκτυο Peer to Peer (P2P)</i> | 40 |
| <i>Εικόνα 1-16 Δίκτυο Peer to Peer (P2P) κοινός εκτυπωτής</i> | 41 |
| <i>Εικόνα 1-17 Δίκτυο Πελάτη - Διακομιστή (Client-Server)</i> | 42 |
| <i>Εικόνα 1-18 Παροχή Υπηρεσιών WiMAX</i> | 44 |
| <i>Εικόνα 1-19 Γειτονικά Δίκτυα (NAN)</i> | 45 |
| <i>Εικόνα 1-20 Field Area Network (FAN)</i> | 46 |
| <i>Εικόνα 1-21 Δίκτυο Μεταγωγής</i> | 48 |
| <i>Εικόνα 1-22 Προεπιλεγμένη πύλη σύνδεσης τοπικού μεταγωγέα με ISP δρομολογητή</i> | 49 |
| <i>Εικόνα 1-23 Τοποθέτηση Γέφυρας (Bridge)</i> | 49 |
| <i>Εικόνα 1-24 Τοποθέτηση γέφυρας μέσα σε ένα δίκτυο (Bridge)</i> | 50 |
| <i>Εικόνα 1-25 Σύνδεση συσκευών σε ένα συγκεντρωτή</i> | 51 |
| <i>Εικόνα 1-26 Σύστημα τριών σημείων πρόσβασης (WLAN) και η διασύνδεση του με LAN δίκτυο</i> | 52 |
| <i>Εικόνα 1-27 Συνδεσμολογία δρομολογητών με fastEthernet interface και serial interfaces</i> | 53 |
| <i>Εικόνα 1-28 Τα επίπεδα του μοντέλου αναφοράς OSI</i> | 54 |
| <i>Εικόνα 2-1 OSI TCP διαστρωμάτωση</i> | 59 |
| <i>Εικόνα 2-2 Επικοινωνία με TCP και UDP πρωτόκολλα</i> | 60 |
| <i>Εικόνα 2-3 Δομή ενός πακέτου IPv6</i> | 66 |
| <i>Εικόνα 2-4 Επικεφαλίδες Επέκτασης</i> | 67 |
| <i>Εικόνα 2-5 Κύκλος ζωής μιας IPv6 διεύθυνσης</i> | 73 |
| <i>Εικόνα 2-6 Δομή ενός ICMPv6 μηνύματος</i> | 74 |
| <i>Εικόνα 3-1 Εξέλιξη του Ethernet διαμέσων τεσσάρων γενιών</i> | 76 |
| <i>Εικόνα 3-2 Τρόπος Λειτουργίας NAT μεθόδου</i> | 80 |
| <i>Εικόνα 3-3 Χρήση CDP πρωτοκόλλου στις γαλάζιες ζεύξεις</i> | 81 |
| <i>Εικόνα 4-1 Δίκτυο δρομολόγησης με βάση το RIP πρωτόκολλο</i> | 86 |
| <i>Εικόνα 4-2 Δίκτυο που λειτουργεί με OSPF πρωτόκολλο</i> | 88 |
| <i>Εικόνα 4-3 BGP πρωτόκολλο δρομολόγησης</i> | 100 |
| <i>Εικόνα 5-1 Κλασσική Προσέγγιση Εικονικοποίησης</i> | 101 |
| <i>Εικόνα 5-2 Προσέγγιση Ελαφριάς Εικονικοποίησης</i> | 101 |
| <i>Εικόνα 6-1 Προσθήκη router image</i> | 104 |
| <i>Εικόνα 6-2 Εγκαθίδρυση σύνδεσης</i> | 105 |
| <i>Εικόνα 6-3 Δρομολογητής c2691</i> | 105 |
| <i>Εικόνα 6-4 Επιλογή Ethernet switch</i> | 106 |
| <i>Εικόνα 6-5 Σύνδεση των δυο υλικολογισμικών</i> | 106 |



| | |
|--|------------|
| <i>Εικόνα 6-6 Παραμετροποίηση των ports του μεταγωγέα.....</i> | <i>106</i> |
| <i>Εικόνα 6-7 Σύνδεση του τερματικού με το μεταγωγέα.....</i> | <i>107</i> |
| <i>Εικόνα 6-8 Δημιουργία τοπολογίας αστέρα.....</i> | <i>107</i> |
| <i>Εικόνα 6-9 Απόδοση IPv4 διεύθυνσης στο δρομολογητή CCSL.....</i> | <i>109</i> |
| <i>Εικόνα 6-10 Εκχώρηση IP διεύθυνσης σε VPC.....</i> | <i>109</i> |
| <i>Εικόνα 6-11 Ping από το PC1 προς το τερματικό PC3.....</i> | <i>110</i> |
| <i>Εικόνα 6-12 Τοπολογία εμπορικού κέντρου.....</i> | <i>110</i> |
| <i>Εικόνα 6-13 Επεξεργασία των network interfaces.....</i> | <i>111</i> |
| <i>Εικόνα 6-14 PCMCIA για την αποθήκευση των διαφορετικών vlans.....</i> | <i>111</i> |
| <i>Εικόνα 6-15 Αρχική συνδεσμολογία.....</i> | <i>112</i> |
| <i>Εικόνα 6-16 Προβολή vlan.....</i> | <i>114</i> |
| <i>Εικόνα 6-17 Προβολή πληροφοριών trunk του central switch.....</i> | <i>118</i> |
| <i>Εικόνα 6-18 ping από pc που ανήκουν στο ίδιο (VLAN10).....</i> | <i>121</i> |
| <i>Εικόνα 6-19 Επιπλέον ping σε ίδιο vlan.....</i> | <i>121</i> |
| <i>Εικόνα 6-20 Δεν υπάρχει επικοινωνία διαφορετικών VLAN.....</i> | <i>122</i> |
| <i>Εικόνα 6-21 Μη ύπαρξη επικοινωνίας μεταξύ διαφορετικών VLAN.....</i> | <i>122</i> |
| <i>Εικόνα 6-22 Ping VLAN20.....</i> | <i>123</i> |
| <i>Εικόνα 6-23 Ping VLAN30.....</i> | <i>123</i> |
| <i>Εικόνα 6-24 Τοπολογία φροντιστηρίου με τα δίκτυα που υπάρχουν.....</i> | <i>124</i> |
| <i>Εικόνα 6-25 Δίκτυο διαφήμισης 1st floor.....</i> | <i>124</i> |
| <i>Εικόνα 6-26 Γειτονικά και μη δίκτυα του 1st floor μετά το RIP.....</i> | <i>125</i> |
| <i>Εικόνα 6-27 Δίκτυο διαφήμισης 2nd floor.....</i> | <i>125</i> |
| <i>Εικόνα 6-28 Γειτονικά και μη δίκτυα του 2nd floor μετά το RIP.....</i> | <i>126</i> |
| <i>Εικόνα 6-29 Δίκτυο διαφήμισης 3rd floor.....</i> | <i>126</i> |
| <i>Εικόνα 6-30 Γειτονικά και μη δίκτυα του 2nd floor μετά το RIP.....</i> | <i>127</i> |
| <i>Εικόνα 6-31 Δίκτυο διαφήμισης 4th floor.....</i> | <i>127</i> |
| <i>Εικόνα 6-32 Γειτονικά και μη δίκτυα του 2nd floor μετά το RIP.....</i> | <i>127</i> |
| <i>Εικόνα 6-33 Δίκτυα διαφήμισης Central switch.....</i> | <i>128</i> |
| <i>Εικόνα 6-34 Γειτονικά και μη δίκτυα του Central switch μετά το RIP.....</i> | <i>128</i> |
| <i>Εικόνα 6-35 Μέρος τοπολογίας του τμήματος κινηματογράφου.....</i> | <i>129</i> |
| <i>Εικόνα 6-36 Δυναμική απόδοση IP.....</i> | <i>129</i> |
| <i>Εικόνα 6-37 Επικοινωνία με τον DHCP server.....</i> | <i>130</i> |
| <i>Εικόνα 6-38 ping στο 4th floor Switch.....</i> | <i>130</i> |
| <i>Εικόνα 6-39 Διαδρομή που θα ακολουθήσουν τα πακέτα.....</i> | <i>131</i> |
| <i>Εικόνα 6-40 Επιτυχής επικοινωνία με τον Server.....</i> | <i>132</i> |



| | |
|---|-----|
| Εικόνα 6-41 Αποτέλεσμα απόκρυψης IP από το wireshark | 132 |
| Εικόνα 6-42 Διαφίμηση των δικτύων του Router2..... | 132 |
| Εικόνα 6-43 Γειτονικά και μη δίκτυα του Router2 (κινηματογράφος) μετά το RIP | 132 |
| Εικόνα 6-44 Κατάστημα ηλεκτρονικών ειδών | 133 |
| Εικόνα 6-45 Χρήστης pc4 δικτύου 192.168.1.0 | 133 |
| Εικόνα 6-46 Χρήστης pc8 δικτύου 192.168.1.0 | 134 |
| Εικόνα 6-47 Επικοινωνία ring μεταξύ των τερματικών των δύο δικτύων..... | 134 |
| Εικόνα 6-48 Μονοπάτι μεταξύ των τερματικών..... | 134 |
| Εικόνα 6-49 Ping από το Router1 στο multilayerSwitch..... | 134 |
| Εικόνα 6-50 InterVlan τοπολογία..... | 135 |
| Εικόνα 6-51 Vlan10 & Vlan20..... | 135 |
| Εικόνα 6-52 Δημιουργία των δυο DHCP pool για τα vlan | 138 |
| Εικόνα 6-53 Απόδοση IP δυναμικά με DHCP του administration vlan | 138 |
| Εικόνα 6-54 Εντολή dhcp binding για την απόδοση των IP διευθύνσεων | 138 |
| Εικόνα 6-55 Ping από host ίδιου δικτύου 192.168.24.0 | 139 |
| Εικόνα 6-56 Ping από host διαφορετικού δικτύου 192.168.24.0 & 192.168.25.0..... | 139 |
| Εικόνα 6-57 Γνωστοποίηση των δικτύων με την χρήση του RIP | 139 |
| Εικόνα 6-58 Μοντέλο δικτύου εταιρίας | 140 |
| Εικόνα 6-59 Παραμετροποίηση slot 1 με κάρτα δικτύου NM-16 ESW | 145 |
| Εικόνα 6-60 Βασικές παραμετροποιήσεις ασφαλείας του μεταγωγέα..... | 146 |
| Εικόνα 6-61 Βλέπουμε χωρίς κάποια μορφή κρυπτογράφησης ελεύθερα τον κωδικό | 146 |
| Εικόνα 6-62 Εκτέλεση εντολών κρυπτογράφησης..... | 146 |
| Εικόνα 6-63 Κρυπτογραφημένος κωδικός | 147 |
| Εικόνα 6-64 Αποτροπή πρόσβασης σε HTTP Servers..... | 147 |
| Εικόνα 6-65 Δημιουργία και προσθήκη VLAN δικτύων..... | 148 |
| Εικόνα 6-66 Επιτυχή προσθήκη VLANs..... | 148 |
| Εικόνα 6-67 Επιτυχής απόδοση διεύθυνσης IP στα VLANs | 149 |
| Εικόνα 6-68 Είσοδος στη βάση του VLAN..... | 150 |
| Εικόνα 6-69 Δημιουργία vnr_domain | 150 |
| Εικόνα 6-70 Πληροφορίες σχετικά με τα status των διεπαφών..... | 150 |
| Εικόνα 6-71 Επιτυχή σύνδεση με Fa1/0 που θα είναι ο διάλογος επικοινωνίας με τα άλλα switch | 151 |
| Εικόνα 6-72 Εφαρμογή κωδικών ασφαλείας όπως και στο Switch_L3A | 151 |
| Εικόνα 6-73 Μετατροπή του Switch_L3B σε vnr client..... | 151 |
| Εικόνα 6-74 Ορισμός switchport trunk στον Switch_L3B στο f1/0 | 152 |



| | |
|--|-----|
| <i>Εικόνα 6-75 Επιτυχής σύνδεση με τον Switch_L3A</i> | 152 |
| <i>Εικόνα 6-76 Επιτυχές πέρασμα των VLANs στον Switch_L3B</i> | 153 |
| <i>Εικόνα 6-77 Πληροφορίες κατάστασης Vtp(Client)</i> | 153 |
| <i>Εικόνα 6-78 Απόδοση IP διευθύνσεων</i> | 154 |
| <i>Εικόνα 6-79 Διευθύνσεις που θα γίνουν ring από τον Switch_L3B</i> | 154 |
| <i>Εικόνα 6-80 Επιτυχείς προσπάθειες μετάδοσης πακέτων</i> | 155 |
| <i>Εικόνα 6-81 Standby λειτουργία προτεραιότητα 255</i> | 156 |
| <i>Εικόνα 6-82 Standby λειτουργία προτεραιότητα 1</i> | 156 |
| <i>Εικόνα 6-83 Standby λειτουργία στον Switch_L3B προτεραιότητα 1</i> | 157 |
| <i>Εικόνα 6-84 Standby λειτουργία στον Switch_L3B προτεραιότητα 255</i> | 157 |
| <i>Εικόνα 6-85 Λειτουργία Standby (HSRP)</i> | 158 |
| <i>Εικόνα 6-86 DHCP pool στα VLANs</i> | 158 |
| <i>Εικόνα 6-87 Δημιουργημένα pools 1/2</i> | 158 |
| <i>Εικόνα 6-88 Δημιουργημένα pools 2/2</i> | 159 |
| <i>Εικόνα 6-89 Επίπεδο πρόσβασης</i> | 159 |
| <i>Εικόνα 6-90 Βασική παραμετροποίηση SWA</i> | 160 |
| <i>Εικόνα 6-91 Προσθήκη και απόδοση IP στο VLAN_admin</i> | 160 |
| <i>Εικόνα 6-92 Πληροφορίες των interfaces του SWA</i> | 160 |
| <i>Εικόνα 6-93 Επιτυχές πέρασμα των VLANs</i> | 161 |
| <i>Εικόνα 6-94 Σύνδεση με τον backup μεταγωγέα</i> | 161 |
| <i>Εικόνα 6-95 Δημιουργία σύνδεσης με τον Switch_L3A</i> | 161 |
| <i>Εικόνα 6-96 Δημιουργία σύνδεσης με τον Switch_L3B</i> | 162 |
| <i>Εικόνα 6-97 Σύνδεση με τα Switch level 3</i> | 162 |
| <i>Εικόνα 6-98 Spanning Tree priority SWL3A</i> | 162 |
| <i>Εικόνα 6-99 Spanning Tree priority SWL3B</i> | 163 |
| <i>Εικόνα 6-100 Διαφορετικές προτεραιότητες στα VLANs</i> | 163 |
| <i>Εικόνα 6-101 Εισαγωγή μεταγωγέων πρόσβασης για τους ορόφους</i> | 164 |
| <i>Εικόνα 6-102 Σύνδεση με Switch_L3A και Switch_L3B</i> | 164 |
| <i>Εικόνα 6-103 Καθορισμός ranges ανάλογα με τα τμήματα του κάθε ορόφου</i> | 164 |
| <i>Εικόνα 6-104 Περιορισμός πρόσβασης ανάλογα με το VLAN του κάθε ορόφου</i> | 165 |
| <i>Εικόνα 6-105 Πόρτες που έχουν πρόσβαση</i> | 165 |
| <i>Εικόνα 6-106 Ranges 3^ο ορόφου ανάλογα με τα τμήματα</i> | 166 |
| <i>Εικόνα 6-107 Access ports</i> | 166 |
| <i>Εικόνα 6-108 Ranges 2^ο ορόφου ανάλογα με τα τμήματα</i> | 167 |
| <i>Εικόνα 6-109 Access ports</i> | 167 |



| | |
|---|-----|
| <i>Εικόνα 6-110 Ranges 1^{ος} ορόφου ανάλογα με τα τμήματα.....</i> | 168 |
| <i>Εικόνα 6-111 Access ports</i> | 168 |
| <i>Εικόνα 6-112 Τοπολογία του οργανισμού.....</i> | 169 |
| <i>Εικόνα 6-113 Θεωρητική προσέγγιση Firewall</i> | 169 |
| <i>Εικόνα 6-114 Πρωτόκολλο διαφήμισης OSPF</i> | 170 |
| <i>Εικόνα 6-115 Τοπολογία Οργανισμού</i> | 170 |
| <i>Εικόνα 6-116 Ping από το 1^ο όροφο προς το 2^ο.....</i> | 171 |
| <i>Εικόνα 6-117 Ping από το 1^ο όροφο προς το 3^ο.....</i> | 171 |
| <i>Εικόνα 6-118 Ping από το 1^ο όροφο προς το 4^ο.....</i> | 171 |
| <i>Εικόνα 6-119 Trace προς τις IP των ορόφων.....</i> | 172 |
| <i>Εικόνα 6-120 Ping από 2^ο όροφο προς 1^ο</i> | 172 |
| <i>Εικόνα 6-121 Ping από 2^ο όροφο προς 3^ο</i> | 172 |
| <i>Εικόνα 6-122 Ping από 2^ο όροφο προς 4^ο.....</i> | 173 |
| <i>Εικόνα 6-123 Trace προς τις IP των ορόφων.....</i> | 173 |
| <i>Εικόνα 6-124 Trace με κλειστό τον Switch_L3B.....</i> | 173 |
| <i>Εικόνα 6-125 Ping από 3^ο όροφο προς 1^ο.....</i> | 174 |
| <i>Εικόνα 6-126 Ping από το 3^ο προς 2^ο</i> | 174 |
| <i>Εικόνα 6-127 Ping από το 3^ο προς 4^ο</i> | 174 |
| <i>Εικόνα 6-128 Trace προς τις IP των ορόφων.....</i> | 175 |
| <i>Εικόνα 6-129 Pings προς όλους τους ορόφους.....</i> | 175 |
| <i>Εικόνα 6-130 Trace προς τις IP των ορόφων.....</i> | 175 |
| <i>Εικόνα 6-131 Ping από το Server προς τον Switch_L3A.....</i> | 176 |
| <i>Εικόνα 6-132 Ping από το Server προς τον Switch_L3B.....</i> | 176 |
| <i>Εικόνα 6-133 Ενσωμάτωση του Wireshark στο GNS3</i> | 177 |
| <i>Εικόνα 6-134 Capturing μέσω Wireshark.....</i> | 177 |
| <i>Εικόνα 6-135 ICMP πακέτα-Request από την πλευρά του wguest και reply από την πλευρά του Switch_L3A</i> | 178 |
| <i>Εικόνα 6-136 Πορεία του πακέτου διαμέσων του 4thFloorSW.....</i> | 178 |
| <i>Εικόνα 6-137 Διαδρομή πακέτου</i> | 178 |
| <i>Εικόνα 6-138 Ping από το Secretary προς το wguest.....</i> | 179 |
| <i>Εικόνα 6-139 Τοπολογία του GRE Tunneling</i> | 181 |
| <i>Εικόνα 6-140 Προγραμματισμός interfaces lo0 και g0/0.....</i> | 181 |
| <i>Εικόνα 6-141 Απόδοση IP διεύθυνσης στις g1/0 θύρες των ISP.....</i> | 182 |
| <i>Εικόνα 6-142 Έλεγχος έγκυρης απόδοσης IP διεύθυνσης</i> | 182 |
| <i>Εικόνα 6-143 Στιγμιότυπο που βλέπουμε τους εσωτερικούς γείτονες.....</i> | 183 |



| | |
|--|-----|
| <i>Εικόνα 6-144 Επιτυχής Λειτουργία του BGP διαμέσων του OSPF</i> | 184 |
| <i>Εικόνα 6-145 Ενημερωμένοι πίνακες του ISP1</i> | 184 |
| <i>Εικόνα 6-146 Τα δίκτυα που έμαθε από το OSPF</i> | 185 |
| <i>Εικόνα 6-147 Επιτυχής ενεργοποίηση του BGP στον ISP3 και ISP1</i> | 186 |
| <i>Εικόνα 6-148 BGP πρωτόκολλο</i> | 186 |
| <i>Εικόνα 6-149 BGP log στον ISP1 και rings στα loopbacks των άλλων ISP</i> | 186 |
| <i>Εικόνα 6-150 BGP log στον ISP2 και rings στα loopbacks των άλλων ISP</i> | 187 |
| <i>Εικόνα 6-151 Μελέτη της πλευράς του Customer</i> | 187 |
| <i>Εικόνα 6-152 DHCP POOL στον ISP1</i> | 187 |
| <i>Εικόνα 6-153 Απόδοση διευθύνσεων</i> | 187 |
| <i>Εικόνα 6-154 Δεσμευμένη IP από τον CS1</i> | 188 |
| <i>Εικόνα 6-155 Επιτυχής επικοινωνία με τους ISP routers</i> | 188 |
| <i>Εικόνα 6-156 IP που πρέπει να διαφημίσουμε</i> | 188 |
| <i>Εικόνα 6-157 Απόδοση διευθύνσεων στον CS2</i> | 189 |
| <i>Εικόνα 6-158 Επιτυχής επικοινωνία των δρομολογητών CS1 και CS2</i> | 190 |
| <i>Εικόνα 6-159 Άνοιγμα των Hosts(VPCs)</i> | 190 |
| <i>Εικόνα 6-160 Απόδοση IP διεύθυνσης με DHCP στον Host1</i> | 191 |
| <i>Εικόνα 6-161 Απόδοση IP διεύθυνσης με DHCP στον Host2</i> | 191 |
| <i>Εικόνα 6-162 Pings των hosts στην default gateway</i> | 191 |
| <i>Εικόνα 6-163 Tunnel interfaces</i> | 192 |
| <i>Εικόνα 6-164 Tunnel0 ενεργοποιημένο για τον CS1</i> | 192 |
| <i>Εικόνα 6-165 Tunnel0 ενεργοποιημένο για τον CS1</i> | 193 |
| <i>Εικόνα 6-166 Ping στα tunnel interfaces vice versa</i> | 193 |
| <i>Εικόνα 6-167 Routing table του ISP2 χωρίς την γνώση του δικτύου 10.1.5.0 (tunnel interfaces)</i> | 194 |
| <i>Εικόνα 6-168 Capture του link μέσα από το Wireshark</i> | 194 |
| <i>Εικόνα 6-169 Κίνηση μέσα από το περιβάλλον του Wireshark</i> | 195 |
| <i>Εικόνα 6-170 Ενθυλακωμένη κίνηση στις public ip των δρομολογητών</i> | 195 |
| <i>Εικόνα 6-171 Ενεργοποίηση πρωτοκόλλου EIGRP για την εκμάθηση των εσωτερικών δικτύων</i> | 195 |
| <i>Εικόνα 6-172 Ανεπιτυχείς προσπάθειες επικοινωνίας με εσωτερικό δίκτυο του CS2</i> | 196 |
| <i>Εικόνα 6-173 Ενεργοποίηση του EIGRP διαμέσων του tunnel 0</i> | 197 |
| <i>Εικόνα 6-174 Ενεργοποίηση του EIGRP διαμέσων του tunnel 0</i> | 197 |
| <i>Εικόνα 6-175 Ανανεωμένος πίνακας δρομολογήσης</i> | 197 |
| <i>Εικόνα 6-176 Επιτυχής επικοινωνία στα εσωτερικά δίκτυα των δρομολογητών</i> | 198 |



| | |
|--|-----|
| <i>Εικόνα 6-177 Επιτυχής επικοινωνία των hosts</i> | 198 |
| <i>Εικόνα 6-178 Παρακολούθηση κίνησης διαμέσων Wireshark</i> | 198 |
| <i>Εικόνα 6-179 Παρακολούθηση IPv4 διευθύνσεων του Internet</i> | 199 |
| <i>Εικόνα 6-180 Παρακολούθηση κίνησης του GRE Tunneling</i> | 199 |
| <i>Εικόνα 6-181 Παρακολούθηση ICMP πακέτων</i> | 199 |
| <i>Εικόνα 6-182 Μόνο EIGRP στους Customer δρομολογητές(CS1 και CS2)</i> | 200 |
| <i>Εικόνα 6-183 Κεντρικός ISP πάροχος δεν γνωρίζει το δίκτυο των Customers 10.0.X.X</i> | 200 |
| <i>Εικόνα 6-184 Τοπολογία Redistribution Διαφορετικών πρωτοκόλλων</i> | 201 |
| <i>Εικόνα 6-185 Επιτυχημένη γειτνίαση μεταξύ των R9-R6</i> | 202 |
| <i>Εικόνα 6-186 Routing Table του R6</i> | 203 |
| <i>Εικόνα 6-187 Επιτυχής γειτνίαση R5-R6</i> | 204 |
| <i>Εικόνα 6-188 Επιτυχής εκμάθηση δικτύου με το BGP</i> | 204 |
| <i>Εικόνα 6-189 Routes που έχουν γίνει redistributed στο OSPF</i> | 205 |
| <i>Εικόνα 6-190 Routing table του R8</i> | 208 |
| <i>Εικόνα 6-191 Routing table του R8</i> | 208 |
| <i>Εικόνα 6-192 Routing table του R8</i> | 208 |
| <i>Εικόνα 6-193 Επιτυχής επικοινωνία του R8 προς τον R9</i> | 209 |
| <i>Εικόνα 6-194 Διαδρομή που ακολούθησε το πακέτο διαμέσων του R2</i> | 209 |
| <i>Εικόνα 6-195 Διαδρομή που ακολούθησε το πακέτο διαμέσων του R1 κλείνοντας την πόρτα του R2</i> | 210 |
| <i>Εικόνα 6-196 Επιτυχής μετάδοση πακέτου από τον R9-R8</i> | 210 |
| <i>Εικόνα 6-197 Routing tables του ΑΠΟΜΑΚΡΥΣΜΕΝΟΥ SITE</i> | 210 |
| <i>Εικόνα 6-198 Capture της ser0/4 & αποστολή πακέτου R8-R9</i> | 211 |
| <i>Εικόνα 6-199 Πληροφορίες της ser0/4</i> | 211 |
| <i>Εικόνα 6-200 Capture της F0/0 & αποστολή πακέτου R4-R9</i> | 212 |
| <i>Εικόνα 6-201 Πληροφορίες της F0/0</i> | 212 |
| <i>Εικόνα 7-1 Επιτυχής επικοινωνία στο περιβάλλον του Mininet</i> | 215 |
| <i>Εικόνα 7-2 Μορφή τοπολογίας που δημιουργεί το mininet</i> | 215 |
| <i>Εικόνα 7-3 Αρχιτεκτονική του Mininet</i> | 216 |
| <i>Εικόνα 7-4 Δημιουργία ξεχωριστών CLI για κάθε host</i> | 216 |
| <i>Εικόνα 7-5 Διαθέσιμες επιλογές στο mininet</i> | 216 |
| <i>Εικόνα 7-6 Links που φαίνονται στην τοπολογία mn</i> | 217 |
| <i>Εικόνα 7-7 Αναλυτικές πληροφορίες τοπολογίας mn</i> | 217 |
| <i>Εικόνα 7-8 ifconfiig στο περιβάλλον του h1</i> | 217 |
| <i>Εικόνα 7-9 ping από τον h1 προς τον h2</i> | 218 |



| | |
|---|-----|
| <i>Εικόνα 7-10 Python Script δυο switch με ένα host το καθένα</i> | 218 |
| <i>Εικόνα 7-11 Δημιουργία τοπολογίας με 1024 host</i> | 219 |
| <i>Εικόνα 7-12 Δημιουργία τοπολογίας bus</i> | 220 |
| <i>Εικόνα 7-13 Πληροφορίες σύνδεσης των hosts και των switches</i> | 220 |
| <i>Εικόνα 7-14 Pingall στους hosts</i> | 221 |
| <i>Εικόνα 7-15 Σχηματική αναπαράσταση του κώδικα</i> | 222 |
| <i>Εικόνα 7-16 4swINROW.py</i> | 222 |
| <i>Εικόνα 7-17 Σχηματική αναπαράσταση του κώδικα</i> | 223 |
| <i>Εικόνα 7-18 2HONESW.py</i> | 223 |
| <i>Εικόνα 7-19 4host1Switch.py</i> | 225 |
| <i>Εικόνα 7-20 Γραφικό περιβάλλον του Miniedit script</i> | 226 |
| <i>Εικόνα 7-21 Τοπολογία στο Miniedit</i> | 226 |
| <i>Εικόνα 7-22 Bridges της τοπολογίας του Miniedit</i> | 227 |
| <i>Εικόνα 7-23 Background δεδομένα της τοπολογίας</i> | 227 |
| <i>Εικόνα 7-24 Επιτυχές φόρτωση δημιουργημένου script από το Miniedit στο Mininet</i> | 228 |
| <i>Εικόνα 7-25 Πληροφορίες της τοπολογίας MiniEdit</i> | 228 |
| <i>Εικόνα 7-26 Ενεργοποίηση των Open vSwitches</i> | 229 |
| <i>Εικόνα 7-27 Επιτυχή μετάδοση και παρακολούθηση από h1 σε h2</i> | 229 |
| <i>Εικόνα 7-28 Επιτυχή μετάδοση και παρακολούθηση πακέτων από τον h1 στον h4</i> | 230 |
| <i>Εικόνα 7-29 Επιτυχή μετάδοση και παρακολούθηση από h1 σε h6</i> | 231 |
| <i>Εικόνα 7-30 Φόρτωση του rython script</i> | 232 |
| <i>Εικόνα 7-31 Παρακολούθηση τοπολογίας μέσω του Wireshark</i> | 232 |
| <i>Εικόνα 7-32 Παρατήρηση πακέτων στην ροή του LOOPBACK</i> | 233 |
| <i>Εικόνα 7-33 Φιλτράρισμα ICMP packets</i> | 233 |
| <i>Εικόνα 7-34 Αναλυτικά τα ICMP και ARP πακέτα στον h1</i> | 234 |
| <i>Εικόνα 7-35 Αναλυτικά τα ICMP και ARP πακέτα του h1 που στέλνει στον h6</i> | 234 |
| <i>Εικόνα 8-1 Χαρακτηριστικά εικονικής μηχανής</i> | 235 |
| <i>Εικόνα 8-2 Είσοδος στο Mininet</i> | 236 |
| <i>Εικόνα 8-3 Προβολή των διαθέσιμων interfaces</i> | 236 |
| <i>Εικόνα 8-4 Ανάθεση IP με DHCP πρωτόκολλο</i> | 237 |
| <i>Εικόνα 8-5 IP του SDN Controller μέσω DHCP</i> | 237 |
| <i>Εικόνα 8-6 Μετάβαση στον ιστότοπο του Aruba SDN Controller</i> | 238 |
| <i>Εικόνα 8-7 Μετάβαση στον ιστότοπο του Aruba SDN Controller</i> | 238 |
| <i>Εικόνα 8-8 Είσοδος στο περιβάλλον του Aruba SDN Controller</i> | 239 |
| <i>Εικόνα 8-9 Γραφικό περιβάλλον του Aruba SDN Controller</i> | 239 |



| | |
|---|-----|
| <i>Εικόνα 8-10 Εισαγωγή Mininet στο GNS3</i> | 240 |
| <i>Εικόνα 8-11 Εισαγωγή Mininet στο GNS3</i> | 240 |
| <i>Εικόνα 8-12 Εισαγωγή Mininet στο GNS3</i> | 240 |
| <i>Εικόνα 8-13 Πρώτη τοπολογία GNS3-Mininet</i> | 241 |
| <i>Εικόνα 8-14 Πρόβλημα με την απόδοση IP</i> | 241 |
| <i>Εικόνα 8-15 Interfaces path</i> | 242 |
| <i>Εικόνα 8-16 Προσθήκη adapters στο Mininet</i> | 243 |
| <i>Εικόνα 8-17 IPv4 Mininet-eth0</i> | 244 |
| <i>Εικόνα 8-18 Επικοινωνία Router 1 -> Mininet</i> | 244 |
| <i>Εικόνα 8-19 Προσθήκη Cloud</i> | 245 |
| <i>Εικόνα 8-20 Παραμετροποίηση Cloud - Προσθήκη Ethernet</i> | 245 |
| <i>Εικόνα 8-21 Default gateway Ethernet δικτύου</i> | 246 |
| <i>Εικόνα 8-22 Ping στη google και στην cisco</i> | 246 |
| <i>Εικόνα 8-23 Προσθήκη DNS-server στο Mininet</i> | 247 |
| <i>Εικόνα 8-24 Επικοινωνία LAN - WAN</i> | 247 |
| <i>Εικόνα 8-25 Επικοινωνία Mininet - google.com</i> | 248 |
| <i>Εικόνα 8-26 Επιτυχής διασύνδεση και των τριών πλατφορμών</i> | 248 |
| <i>Εικόνα 8-27 Δημιουργία τοπολογίας</i> | 249 |
| <i>Εικόνα 8-28 Pings των hosts</i> | 249 |
| <i>Εικόνα 8-29 Τοπολογία στον Aruba Controller</i> | 249 |
| <i>Εικόνα 8-30 Προσθήκη NAT στο interface eth0 του Mininet</i> | 250 |
| <i>Εικόνα 8-31 Προσθήκη NAT πρωτοκόλλου</i> | 250 |
| <i>Εικόνα 8-32 Επιτυχής επικοινωνία hosts με το διαδίκτυο</i> | 251 |
| <i>Εικόνα 8-33 Αποστολή πακέτων προς την Cisco</i> | 251 |
| <i>Εικόνα 8-34 Κίνηση που παρατηρείται στο Wireshark</i> | 251 |
| <i>Εικόνα 8-35 Νέα ονόματα pre-built machines</i> | 252 |
| <i>Εικόνα 8-36 Διασύνδεση εξομοιωτών με το GNS3</i> | 253 |
| <i>Εικόνα 8-37 Προσθήκη Mininet και SDN controller</i> | 253 |
| <i>Εικόνα 8-38 Απόδοση IP διεύθυνσης με DHCP</i> | 254 |
| <i>Εικόνα 8-39 Απόδοση IP διεύθυνσης με DHCP</i> | 254 |
| <i>Εικόνα 8-40 Προσθήκη Ethernet</i> | 255 |
| <i>Εικόνα 8-41 IP του Ethernet και του GNS3</i> | 255 |
| <i>Εικόνα 8-42 Τοπολογία που γίνεται Integate</i> | 256 |
| <i>Εικόνα 8-43 Απόδοση IP DHCP από το τοπικό μας δίκτυο</i> | 256 |
| <i>Εικόνα 8-44 Επιτυχής Επικοινωνία των δύο εξομοιωτών</i> | 257 |



| | |
|--|------------|
| <i>Εικόνα 8-45 Capture από το περιβάλλον του Wireshark.....</i> | <i>257</i> |
| <i>Εικόνα 8-46 Αποτυχία πρόσβασης στον controller.....</i> | <i>258</i> |
| <i>Εικόνα 8-47 Boot του Virtual Box.....</i> | <i>258</i> |
| <i>Εικόνα 8-48 Προσθήκη στα devices</i> | <i>259</i> |
| <i>Εικόνα 8-49 Νέα μορφή της τοπολογίας μας</i> | <i>259</i> |
| <i>Εικόνα 8-50 Επιτυχή απόδοση IP διεύθυνσης στο Windows 7 machine</i> | <i>260</i> |
| <i>Εικόνα 8-51 Περιβάλλον του SDN Controller.....</i> | <i>261</i> |
| <i>Εικόνα 8-52 Επιτυχής επικοινωνία Mininet-Windows machine</i> | <i>261</i> |
| <i>Εικόνα 8-53 Σύνδεση με τον remote controller.....</i> | <i>262</i> |
| <i>Εικόνα 8-54 Τοπολογία που μας εμφανίζεται στον ελεγκτή Aruba SDN Controller</i> | <i>262</i> |
| <i>Εικόνα 8-55 4hosts-linear-4switch.....</i> | <i>263</i> |
| <i>Εικόνα 8-56 Monitoring της τοπολογίας</i> | <i>263</i> |
| <i>Εικόνα 8-57 Γραμμική τοπολογία 10hosts-10OpenVswitch</i> | <i>264</i> |
| <i>Εικόνα 8-58 Δημιουργία 100 switches-100hosts</i> | <i>264</i> |
| <i>Εικόνα 8-59 100 switches στον ελεγκτή.....</i> | <i>265</i> |
| <i>Εικόνα 8-60 Εμφάνιση hosts</i> | <i>265</i> |
| <i>Εικόνα 8-61 Επιλογή μικρότερου μονοπατιού</i> | <i>266</i> |
| <i>Εικόνα 8-62 Απομόνωση του καλύτερου μονοπατιού</i> | <i>266</i> |
| <i>Εικόνα 8-63 Star Topology</i> | <i>267</i> |
| <i>Εικόνα 8-64 Απομόνωση Host με IP 10.0.0.43</i> | <i>267</i> |
| <i>Εικόνα 8-65 Capture του link e0/0.....</i> | <i>268</i> |
| <i>Εικόνα 8-66 Hello messages</i> | <i>268</i> |
| <i>Εικόνα 8-67 Απάντηση switch με κάποια features.....</i> | <i>269</i> |
| <i>Εικόνα 8-68 Port description</i> | <i>269</i> |
| <i>Εικόνα 8-69 Tree topology fanout=3 depth=3</i> | <i>270</i> |
| <i>Εικόνα 8-70 Πακέτα που περνούν από το κεντρικό switch</i> | <i>271</i> |
| <i>Εικόνα 8-71 Echo Request και echo reply</i> | <i>271</i> |
| <i>Εικόνα 8-72 Πεδίο Openflow classes.....</i> | <i>272</i> |
| <i>Εικόνα 8-73 Τοπολογία με προσθήκη δύο δρομολογητών</i> | <i>273</i> |
| <i>Εικόνα 8-74 Δυναμική απόδοση IP διευθύνσεων</i> | <i>274</i> |
| <i>Εικόνα 8-75 Pings R2 δρομολογητή προς τις εικονικές μηχανές</i> | <i>275</i> |
| <i>Εικόνα 8-76 Ping από τις εικονικές μηχανές προς τον R2.....</i> | <i>276</i> |
| <i>Εικόνα 8-77 Pings R3 δρομολογητή προς τις εικονικές μηχανές</i> | <i>277</i> |
| <i>Εικόνα 8-78 Ping από τις εικονικές μηχανές προς τον R3.....</i> | <i>278</i> |
| <i>Εικόνα 8-79 Δημιουργία τοπολογίας στον Mininet</i> | <i>279</i> |



| | |
|--|-----|
| Εικόνα 8-80 Οπτικοποίηση τοπολογίας στον Aruba SDN..... | 279 |
| Εικόνα 8-81 Γεφύρωση του h1 με το s1..... | 280 |
| Εικόνα 8-82 Επικοινωνία πριν την παραμετροποίηση..... | 280 |
| Εικόνα 8-83 Επικοινωνία μετά την παραμετροποίηση..... | 281 |
| Εικόνα 8-84 Pings h1->R2..... | 281 |
| Εικόνα 8-85 Pings R2->h1..... | 282 |
| Εικόνα 8-86 Επιτυχής επικοινωνία του εσωτερικού host με το υπόλοιπο δίκτυο..... | 283 |
| Εικόνα 8-87 3 ^η τοπολογία διασύνδεση Mininet με δύο δρομολογητές..... | 284 |
| Εικόνα 8-88 Script διασύνδεσης με router..... | 285 |
| Εικόνα 8-89 Επιτυχής γεφύρωση εικονικών switches με physical ports..... | 286 |
| Εικόνα 8-90 Εικόνα από Aruba SDN πριν μάθει τους εξωτερικούς δρομολογητές..... | 287 |
| Εικόνα 8-91 Επιτυχής επικοινωνία των εσωτερικών hosts με τους εξωτερικούς routers..... | 288 |
| Εικόνα 8-92 Επιτυχής επικοινωνία του R1 ->h1,h2 και R1->R2..... | 288 |
| Εικόνα 8-93 Πέρασμα πακέτων διαμέσων του Mininet..... | 289 |
| Εικόνα 8-94 Επιτυχής επικοινωνία του R2 ->h1,h2 και R2->R1..... | 289 |
| Εικόνα 8-95 Ενημερωμένη τοπολογία με 4 hosts στον Controller..... | 290 |
| Εικόνα 8-96 Μορφή τοπολογίας με ένα ακόμη δρομολογητή..... | 291 |
| Εικόνα 8-97 Επικοινωνία R3-R1..... | 293 |
| Εικόνα 8-98 Επικοινωνία h1 ->R1 & h1->R3..... | 294 |
| Εικόνα 8-99 Επικοινωνία h2 ->R1 & h2->R3..... | 295 |
| Εικόνα 8-100 Επικοινωνία R3->h1 & R3->h2..... | 295 |
| Εικόνα 8-101 Τοπολογία στον SDN με δρομολογητή διαφορετικού δικτύου..... | 296 |
| Εικόνα 8-102 Νέα ζεύξη R2-R3..... | 297 |
| Εικόνα 8-103 R3 προς όλες τις IP του δικτύου..... | 300 |
| Εικόνα 8-104 R1 προς όλες τις IP του δικτύου..... | 300 |
| Εικόνα 8-105 R2 προς όλες τις IP του δικτύου..... | 301 |
| Εικόνα 8-106 h1 προς όλες τις IP του δικτύου..... | 301 |
| Εικόνα 8-107 h2 προς όλες τις IP του δικτύου..... | 302 |
| Εικόνα 8-108 Εικόνα τοπολογίας μέσα από τον Aruba SDN Controller..... | 302 |
| Εικόνα 9-1 Container based αρχιτεκτονική..... | 304 |
| Εικόνα 9-2 Αρχιτεκτονική του Core emulator..... | 305 |
| Εικόνα 9-3 παραμετροποίηση των πεδίων zebra και ospfd..... | 308 |
| Εικόνα 9-4 Αλλαγή των πεδίων με την τιμή yes..... | 308 |
| Εικόνα 9-5 Γραφικό περιβάλλον του Core..... | 309 |
| Εικόνα 9-6 Τοπολογία Έξυπνων σπιτιών στη Σάμο..... | 312 |



| | |
|---|-----|
| Εικόνα 9-7 Λειτουργία των Έξυπνων συσκευών..... | 314 |
| Εικόνα 9-8 Δίκτυα που θα τρέχουν στο δίκτυο του Smart Home 1 | 314 |
| Εικόνα 9-9 Τοπολογία Σπιτιού Νο1 στο Καρλόβασι | 315 |
| Εικόνα 9-10 Τοπολογία Σπιτιού Νο1 στο Καρλόβασι | 316 |
| Εικόνα 9-11 Ping Lamp->Sensor Movement | 317 |
| Εικόνα 9-12 TraceRoute από Lamp->Sensor Management..... | 317 |
| Εικόνα 9-13 Ping από το Smart Lock->Router 1 | 318 |
| Εικόνα 9-14 Ping6 από τον Thermostat->Humidifier..... | 319 |
| Εικόνα 9-15 Διάγραμμα κίνησης σε kbps..... | 319 |
| Εικόνα 9-16 Δίκτυα στην πόλη του Καρλοβάσου | 320 |
| Εικόνα 9-17 Διαθέσιμα Δίκτυα και Διαθέσιμα Services του μεταγωγέα Καρλοβάσου | 321 |
| Εικόνα 9-18 Τοπολογία σπιτιών Καρλοβάσου | 322 |
| Εικόνα 9-19 Ping σπιτιού νούμερο 2 | 322 |
| Εικόνα 9-20 Ping6 σπιτιού νούμερο 2 | 323 |
| Εικόνα 9-21 Επικοινωνία Devices Διαφορετικών σπιτιών | 323 |
| Εικόνα 9-22 Εύρεση μονοπατιού Movement Sensor του σπιτιού νούμερο 2->Garage του σπιτιού νούμερο 1 | 324 |
| Εικόνα 9-23 Capturing πακέτων στο RI-eth1 | 324 |
| Εικόνα 9-24 ICMP πακέτα | 325 |
| Εικόνα 9-25 ICMPv6 πακέτα..... | 325 |
| Εικόνα 9-26 Link Καρλόβασι-Βαθύ..... | 326 |
| Εικόνα 9-27 Link Βαθύ-Καρλόβασι..... | 326 |
| Εικόνα 9-28 Επικοινωνία των smart home στο Βαθύ..... | 329 |
| Εικόνα 9-29 Επικοινωνία των smart home Βαθύ-Καρλόβασι | 329 |
| Εικόνα 9-30 Εύρεση Διαδρομής από ανεμιστήρα (home No4)- κλειδαριά(home No1)..... | 330 |
| Εικόνα 9-31 IPv4 Routes switchL3VATHU | 331 |
| Εικόνα 9-32 Διασύνδεση και των τριών πόλεων..... | 332 |
| Εικόνα 9-33 ping6 Humidifier(No6)->Garage(No1)..... | 335 |
| Εικόνα 9-34 ping6 Sprinkler(No3)->Windows(No5) | 335 |
| Εικόνα 9-35 traceroute Sprinkler(No3)->Windows(No5) | 336 |
| Εικόνα 10-1 Unix shell που ανοίγουμε για τους κόμβους του δικτύου..... | 338 |
| Εικόνα 10-2 Τοπολογία όπως καθορίζεται στο IMUNES | 339 |
| Εικόνα 10-3 Τοπολογία όπως αναπτύσσεται στον πυρήνα (kernel) | 340 |
| Εικόνα 10-4 Docker υποδοχέας (Container)..... | 341 |
| Εικόνα 10-5 Γραφικό Περιβάλλον του IMUNES | 342 |



| | |
|---|-----|
| <i>Εικόνα 10-6 Τοπολογία διασύνδεσης GNS3-IMUNES</i> | 344 |
| <i>Εικόνα 10-7 Παραμετροποίηση interfaces του IMUNES machine</i> | 344 |
| <i>Εικόνα 10-8 Επιτυχής επικοινωνία GNS3-IMUNES machine</i> | 345 |
| <i>Εικόνα 10-9 Μηνύματα ενημέρωσης της επιτυχής λειτουργίας των πρωτοκόλλων</i> | 350 |
| <i>Εικόνα 10-10 Επιτυχής επικοινωνία R1->R3</i> | 351 |
| <i>Εικόνα 10-11 Επιτυχής επικοινωνία R3->R1</i> | 351 |
| <i>Εικόνα 10-12 Εύρεση διαδρομής διαμέσων του R2 από τον R1</i> | 351 |
| <i>Εικόνα 10-13 Εύρεση διαδρομής διαμέσων του R2 από τον R3</i> | 351 |
| <i>Εικόνα 10-14 Τοπολογία Smart Home 1</i> | 353 |
| <i>Εικόνα 10-15 Static route ethernet port</i> | 353 |
| <i>Εικόνα 10-16 Smart Home 2</i> | 354 |
| <i>Εικόνα 10-17 Smart Home 3</i> | 355 |

Κατάλογος Πινάκων

| | |
|--|-----|
| <i>Πίνακας 2-1 Multicast Διεύθυνση στο IPv6</i> | 71 |
| <i>Πίνακας 2-2 Παραδείγματα απεικόνισης διευθύνσεων και Συμπίεση μηδενικών στις IPv6 διευθύνσεις</i> | 72 |
| <i>Πίνακας 2-3 Βασικά μηνύματα του IPv6</i> | 74 |
| <i>Πίνακας 4-1 Μετρικές πρωτοκόλλων</i> | 98 |
| <i>Πίνακας 6-1 Πίνακας των bits και subnet mask</i> | 141 |
| <i>Πίνακας 9-1 IP διευθύνσεις smart home No1</i> | 313 |
| <i>Πίνακας 9-2 IP διευθύνσεις smart home No2</i> | 321 |
| <i>Πίνακας 9-3 IP διευθύνσεις smart home No3</i> | 327 |
| <i>Πίνακας 9-4 IP διευθύνσεις smart home No4</i> | 328 |
| <i>Πίνακας 9-5 IP διευθύνσεις smart home No5</i> | 333 |
| <i>Πίνακας 9-6 IP διευθύνσεις smart home No6</i> | 334 |
| <i>Πίνακας 10-1 Διευθύνσεις IP του Smart Home 1</i> | 354 |
| <i>Πίνακας 10-2 Διευθύνσεις IP του Smart Home 2</i> | 355 |
| <i>Πίνακας 10-3 Διευθύνσεις IP του Smart Home 3</i> | 356 |
| <i>Πίνακας 10-4 Διευθύνσεις IP GNS3 project</i> | 356 |



1 Βιβλιογραφική αναφορά στις Έννοιες της Δικτύωσης-Αρχιτεκτονική Δικτύων και Μοντέλων Σχεδίασης

1.1 Βασικές Έννοιες Δικτύωσης

Δίκτυο Υπολογιστών

Με τον όρο αυτό αναφερόμαστε σε ένα σύνολο από δύο ή παραπάνω τερματικά τα οποία είναι συνδεδεμένα μεταξύ τους με ένα ή και περισσότερα φυσικά μέσα.

Τα τερματικά καλούνται **κόμβοι (nodes)**

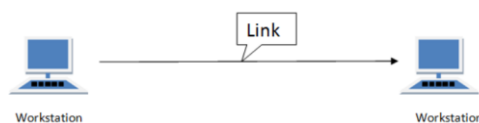
- κόμβος ενός δικτύου μπορεί να είναι κάθε είδους υπολογιστής ή τερματικό
- ο κάθε κόμβος χαρακτηρίζεται από μία τουλάχιστον αλφαριθμητική τιμή που ορίζεται ως διεύθυνση

Το οποιοδήποτε φυσικό μέσο ονομάζεται σύνδεση(link) ή κανάλι

- π.χ. οπτικές ίνες, ηλεκτρικά καλώδια(ομοαξονικά)
- Οι ζεύξεις και οι κόμβοι ονομάζονται πόροι σε ένα δίκτυο

Δισημειακή Τοπολογία(Point to Point Connection)

Στην αρχιτεκτονική point – to –point, μόνο ένα ζεύγος αποστολέα – δέκτη(transmitter-receiver) έχει επικοινωνία σε μια καθορισμένη συχνότητα φέροντος (carrier frequency).Ο κύριος κόμβος στο δίκτυο εκτελεί το συντονισμό σε όλο το εύρος του δικτύου, αντίθετα ο απομακρυσμένος(remote) client έχει ρόλο υφισταμένου(subordinate client).Η απευθείας σύνδεση δύο κόμβων μπορεί να είναι μονόδρομη (simplex),διαδοχική(half duplex) ή και αμφίδρομη(full duplex).[1]



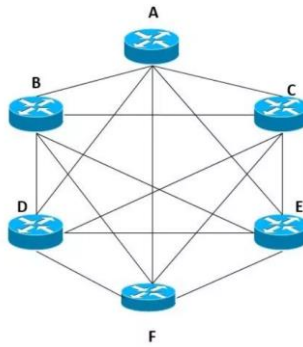
Εικόνα 1-1 Point to Point Connection



Καταναμημένη Τοπολογία(Full Mesh Network)

Για να κατανοήσουμε καλύτερα τον όρο full mesh δίκτυο δίνεται ένα παράδειγμα όπου αν υπάρχουν 5 κόμβοι οι συνδέσεις που πρέπει να γίνουν είναι 10 και αυτό προκύπτει από τη παρακάτω μαθηματική σχέση.

- $X(X - 1)/2$, όπου X ο αριθμός των κόμβων του δικτύου



Εικόνα 1-2 Full Mesh Network Topology

Η full mesh δικτυακή τοπολογία δεν είναι ενδείκνυται για μεγάλη κλίμακας τοπολογίες η με ότι έχει να κάνει με παγκόσμια δίκτυα εξαιτίας διαφόρων οικονομικών, τεχνολογικών και άλλων παραγόντων. Με τον όρο κλιμάκωση αναφερόμαστε στην ικανότητα ενός δικτύου να μεγαλώνει ή να μικραίνει την έκταση του διατηρώντας σε ικανοποιητικό βαθμό την απόδοση σε χρόνο και κόστος.

- Ορισμένοι οικονομικοί παράγοντες θεωρούνται το κόστος που προκύπτει από το μεγάλο όγκο των απαιτούμενων ζεύξεων και το γεγονός ότι το κόστος μιας ζεύξης δεν παρουσιάζει ομαλή αύξηση συγκριτικά με το μήκος της.
- Πιο συγκεκριμένα τεχνολογικός παράγοντας μπορεί να θεωρηθεί ότι κάθε υπολογιστής πρέπει να είναι ικανός να διευθετήσει ένα μεγάλο αριθμό συνδέσμων.[2][5]

Κοινή χρήση πόρων

Για να έχουμε ένα αποδοτικό και εύκολα διαβαθμίσιμο δίκτυο σημαντικό ρόλο εμφανίζει η κοινή χρήση πόρων σε κόμβους-συνδέσμους.

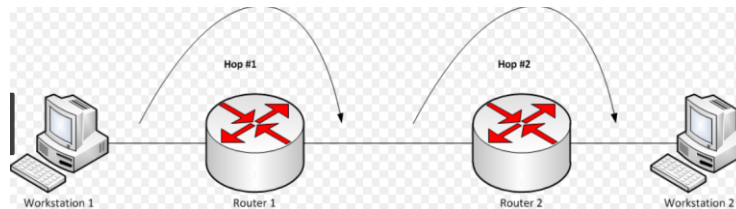


Απόδοση Δικτύου

Για να αυξήσουμε την απόδοση σε μια δικτυακή τοπολογία πρέπει να μειώσουμε το πλήθος και το μήκος των ζεύξεων γι' αυτό και ένα μέρος των κόμβων του δικτύου καλό είναι να χρησιμοποιούνται για τη διασύνδεση.

Κοινή χρήση συνδέσμου

Με τον όρο αυτό αναφερόμαστε στο γεγονός ότι γίνεται χρήση ενός μοναδικού και κοινού από όλους σύνδεσμο. Έτσι ο κάθε κόμβος κάνει προσπάθεια για να εξασφαλίσει την πρόσβαση του στον κοινό σε όλους σύνδεσμο και να κατοχυρώσει την αποκλειστικότητα του. Έχοντας πάρει την πρόσβαση ο κόμβος είναι πλέον ικανός να επικοινωνήσει ταχύτατα με την απλή μετάδοση πληροφορίας. Η συγκεκριμένη διαδικασία ονομάζεται επικοινωνία ενός hop(άλμα).



Εικόνα 1-3 One Hop Communication

Τα συγκεκριμένα δίκτυα λέγονται *direct link*(άμεσου συνδέσμου) ή *multiple access*(πολλαπλών προσβάσεων).

1.2 Βασικές Τοπολογίες Δικτύωσης

Με την έννοια τοπολογία δικτύου (network topology) αναφερόμαστε στη διάταξη των διαφόρων στοιχείων (συνδέσεις, κόμβοι, κλπ.) των τηλεπικοινωνιακών δικτύων. Είναι η τοπολογική αναπαράσταση ενός δικτύου και μπορεί να απεικονιστεί *φυσικά* ή *λογικά*.

Η *φυσική τοπολογία* παρουσιάζει τις θέσεις των στοιχείων του δικτύου (συσκευές, καλώδια, κλπ.) όπως θα ήταν στο χώρο. Η *λογική τοπολογία* παρουσιάζει την ροή των δεδομένων μέσα στο δίκτυο. Δύο δίκτυα που έχουν την ίδια φυσική τοπολογία ενδέχεται να έχουν διαφορετική λογική (τοπολογία) αν διαφέρουν στην τεχνολογία των συσκευών και των μέσων μετάδοσης. [3][5]



Πιο αναλυτικά, η φυσική τοπολογία (physical topology) αφορά τη διάταξη των ζεύξεων που χρησιμοποιούνται για τη σύνδεση συσκευών μέσα στο χώρο. Αναφέρεται στη διάταξη των καλωδίων, στις θέσεις των κόμβων, στις αποστάσεις που καλύπτουν και στους συνδέσμους μεταξύ των κόμβων και της καλωδίωσης. Κάνει λόγο για τις δυνατότητες των συσκευών, των μέσων μεταφοράς της πληροφορίας και τον τρόπο με τον οποίο συνδέονται μέσα στο δίκτυο (ενσύρματα ή ασύρματα). Η φυσική τοπολογία δίνει έμφαση στις αποστάσεις διότι το σήμα εξασθενεί σε απομακρυσμένες συνδέσεις, έχοντας ως αποτέλεσμα την αύξηση του αριθμού των σφαλμάτων και τη μείωση της ταχύτητα μετάδοσης. Επικεντρώνεται στη δημιουργία της καλύτερης δυνατής διάταξης καθιστώντας ευκολότερο τον εντοπισμό και τη διόρθωση σφαλμάτων. Αφορά το κόστος της σύνδεσης που μπορεί να είναι είτε καλωδιακή, είτε ασύρματη ή οποιοδήποτε άλλο παρεχόμενο τηλεπικοινωνιακό κύκλωμα.[4][5]

Όσον αφορά τη λογική τοπολογία ή τοπολογία σήματος (Logical topology ή signal topology) είναι η ηλεκτρονική και προγραμματιστική πραγματοποίηση της επικοινωνίας. Δεν δίνει βάση στην ισχύ του σήματος όταν αυτό διαδίδεται μέσα σε ένα σύστημα. Δίνει βάση στον τρόπο και την λογική της διάδοσης του σήματος δίχως να στηρίζεται εξολοκλήρου στη φυσική διασύνδεση των συσκευών, και τον τρόπο που περνούν τα δεδομένα από το ένα σημείο του δικτύου στο άλλο, ακόμα και αν αυτό αφορά το εσωτερικό μιας δικτυακής συσκευής. Χρησιμοποιώντας διαφορετικά τις δύο τοπολογίες έχει ως επακόλουθο στο ίδιο δίκτυο η φυσική από την λογική τοπολογία να παρουσιάζουν διαφορές.

Στην παραπάνω ενότητα αναφερθήκαμε στην δισημειακή τοπολογία(point-to-point) και στη κατανεμημένη τοπολογία (mesh topology) εδώ θα αναλύσουμε τις εξής τοπολογίες:

- Τοπολογία διαύλου (bus topology)
- Τοπολογία αστέρα (star topology)
- Τοπολογία δακτυλίου (ring topology)
- Τοπολογία δέντρου(tree topology)
- Υβριδική τοπολογία(hybrid topology)

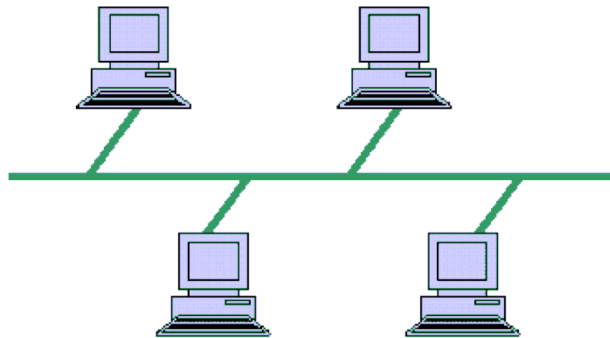
1.2.1 Τοπολογία Διαύλου(Bus Topology)

Η τοπολογία διαύλου (bus topology) είναι αρκετά απλή καθώς κάθε κόμβος συνδέεται σε ένα κεντρικό καλώδιο. Το συγκεκριμένο κεντρικό καλώδιο αποτελεί τον κορμό (backbone ή bus) του δικτύου και καλείται δίαυλος ή αρτηρία. Ένα πακέτο δεδομένων το οποίο αποτελείται από τις πληροφορίες ελέγχου και τα δεδομένα του χρήστη, έχει αφετηρία έναν από τους κόμβους ταξιδεύει αμφίδρομα και σειριακά περνάει από όλους τους άλλους κόμβους του διαύλου.

Ο κάθε κόμβος επιθεωρεί τη διεύθυνση παραλήπτη του πακέτου και αν υπάρχει ταυτοποίηση με την δική του το αποδέχεται, αλλιώς το απορρίπτει. Οι bus topologies σε θέμα κόστους είναι αρκετά χαμηλές και εύκολες στην εγκατάσταση κατά κύριο λόγο στα μικρά σε έκταση δίκτυα εξαιτίας της κεντρικής ζεύξης. Γνωρίζοντας ότι τα πακέτα μεταφέρονται σε ολόκληρο το δίκτυο ανεξάρτητα της τοποθεσίας του κόμβου(receiver) υπάρχει μεγάλη πιθανότητα να επιβαρύνει την συνολική απόδοση του. Συμπληρωματικά το επίπεδο ασφάλειας είναι αρκετά χαμηλό διότι όλα τα τερματικά λαμβάνουν το σήμα που έστειλε ο



κόμβος(transmitter)χωρίς να περνάει κάποια διαδικασία ελέγχου(encryption-decryption).Ένα από τα σημαντικότερα μειονεκτήματα είναι ότι αν προσθέσουμε ή αφαιρέσουμε έναν κόμβο, τότε πρέπει να τεθεί σε αδράνεια ολόκληρο το δίκτυο. Αντίστοιχο αποτέλεσμα έχουμε και στην περίπτωση κάποιας βλάβης στη κεντρική ζεύξη. Εν κατακλείδι οι λόγοι που προαναφέρθηκαν καθιστούν αυτή την τοπολογία ανίκανη να αποδώσει σε μεγάλης κλίμακας δίκτυα. .[6][8]



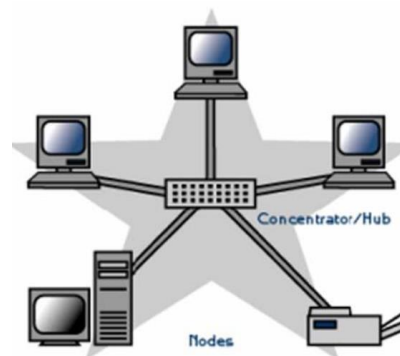
Εικόνα 1-4 Τοπολογία Διαύλου (Bus Topology)

1.2.2 Τοπολογία Αστέρα (Star Topology)

Στην *τοπολογία αστέρα (star topology)* κάθε κόμβος (node) είναι συνδεδεμένος σε ένα "κεντρικό" κόμβο. Στην συγκεκριμένη τοπολογία ένα πακέτο δεδομένων έχοντας ως έναρξη έναν απομακρυσμένο κόμβο κατευθύνεται πάντα στον κεντρικό κόμβο ο οποίος μέσω ξεχωριστών συνδέσεων επικοινωνίας αναμεταδίδει σε όλους τους κόμβους. Οι περιφερειακοί κόμβοι εγκαθιδρύουν την μεταξύ τους επικοινωνία με αποστολές και λήψεις στον κεντρικό κόμβο.

Για να έχουμε μέγιστα ποσοστά απόδοσης στη λειτουργία του δικτύου σημαντικό ρόλο αποτελεί ο κεντρικός κόμβος ο οποίος λέγεται και διαχειριστής κίνησης. Αν το πακέτο το οποίο θα λάβει από ένα κόμβο είναι απλό hub(διανομέας) θα το αποστείλει σε όλους τους υπόλοιπους κόμβους και εν τέλει θα το παραλάβει εκείνος ο κόμβος που έχει την διεύθυνση παραλήπτη για το πακέτο αυτό ενώ οι υπόλοιποι θα το παραλείψουν. Αντίστοιχα και με τη *bus topology* η απόδοση του δικτύου παρουσιάζει πτώση εξαιτίας της μεταφοράς πακέτων προς όλους τους κόμβους.

Το δίκτυο ανεβάζει σε πολύ μεγάλο βαθμό την απόδοση του στην περίπτωση που ο κεντρικός κόμβος είναι *μεταγωγέας (switch)*. Ο μεταγωγέας ακολουθεί μια διαδικασία η οποία πρώτα ελέγχει την διεύθυνση παραλήπτη του πακέτου και στη συνέχεια το προωθεί μοναδικά στον receiver node. Στην τοπολογία αστέρα δίνεται η δυνατότητα πρόσθεσης ή αφαίρεσης κόμβου με μεγάλη ευκολία δίχως να επηρεάζει την εύρυθμη λειτουργία του υπόλοιπου δικτύου. Σημαντικό μειονέκτημα είναι το γεγονός ότι αν απενεργοποιηθεί ο central node τότε ολόκληρο δίκτυο παύει να λειτουργεί. Αξίζει να σημειωθεί ότι κεντρικός κόμβος υποστηρίζει περιορισμένο αριθμό ζεύξεων και πρέπει να είναι μεγάλος σε χωρητικότητα για να εξυπηρετεί με συνέπεια το δίκτυο.[5][9]



Εικόνα 1-5 Τοπολογία Αστέρα (Star Topology)

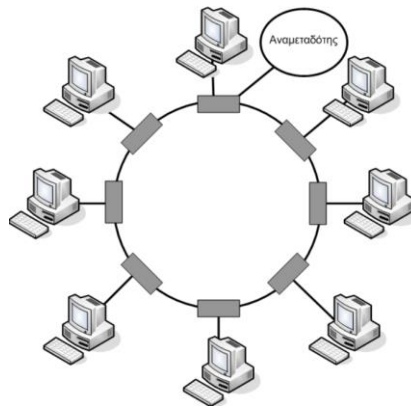
1.2.3 Τοπολογία Δακτυλίου (Ring Topology)

Η τοπολογία δακτυλίου (ring topology) είναι παρόμοια με την τοπολογία διαύλου (bus) μόνο που εδώ όλα τα τερματικά συνδέονται μεταξύ τους σε ένα κλειστό βρόχο, με αποτέλεσμα το κάθε τερματικό να είναι συνδεδεμένο με δύο άλλα. Η ροή των δεδομένων είναι προς μία κατεύθυνση, αν και υπάρχουν δακτύλιοι διπλής κατεύθυνσης. Τη στιγμή όπου ένας κόμβος στέλνει πακέτα πληροφορίας σε έναν άλλον, τα δεδομένα περνούν από οποιοδήποτε κόμβο μεσολαβεί καθώς διαπερνούν τον δακτύλιο μέχρι να φτάσουν στον προορισμό τους.

Γνωρίζοντας ότι το σήμα στις συνδέσεις (links) των δικτύων εξασθενεί καθώς μεγαλώνει η απόσταση, αξίζει να σημειωθεί ότι ο κάθε παρεμβαλλόμενος κόμβος πέραν του ότι ελέγχει τη διεύθυνση του πακέτου για να δει αν είναι προορίζεται για αυτόν, το ενισχύει και το προωθεί στον επόμενο κόμβο εκτελώντας την λειτουργία ενός διανομέα. Ως επακόλουθο η συγκεκριμένη τοπολογία είναι κατάλληλη για την κάλυψη μεγάλων αποστάσεων.[7]

Αντίστοιχα και με τη bus topology αν υπάρξει σε κάποιο σημείο του δικτύου βλάβη, τότε και το υπόλοιπο δίκτυο παύει να λειτουργεί. Έχει σημαντικό πλεονέκτημα σε σύγκριση με την τοπολογία αστέρα στο γεγονός ότι λειτουργεί χωρίς "κεντρικό" κόμβο. Επιπλέον κάθε υπολογιστής έχει ισότιμες ευκαιρίες πρόσβασης σε πόρους και όταν παρουσιαστεί αύξηση στον όγκο του φορτίου, το δίκτυο λειτουργεί σε υψηλά επίπεδα.

Κύρια μειονεκτήματα της εν λόγω τοπολογίας είναι ότι αν μία ζεύξη μεταξύ των κόμβων έχει χαμηλό ρυθμό μεταφοράς δεδομένων τότε καθυστερεί ολόκληρο το δίκτυο και αν παρουσιαστεί κάποιο πρόβλημα μέσα στο δίκτυο είναι αρκετά δύσκολο να εντοπίσουμε την αιτία που το προκάλεσε.[5]



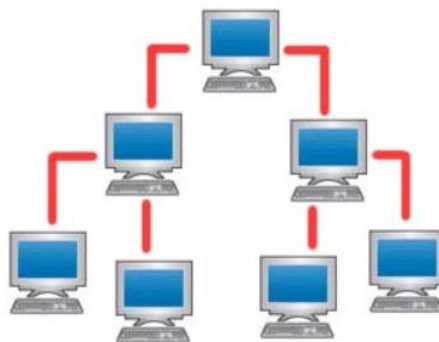
Εικόνα 1-6 Τοπολογία Δακτυλίου (Ring Topology)

1.2.4 Τοπολογία Δέντρου(Tree Topology)

Η τοπολογία δέντρου αποτελεί μια παραλλαγή της τοπολογίας διαύλου. Στην tree topology το μέσο που μεταδίδει τα πακέτα είναι ένα διακλαδιζόμενο καλώδιο χωρίς κλειστούς βρόχους, το οποίο ξεκινά από έναν κόμβο που ονομάζεται κεφαλή ή ρίζα. Η κεφαλή (head node) διαβιβάζει σε όλο το εύρος του δικτύου το σήμα το οποίο λαμβάνει από τον κάθε εκπεμπόμενο κόμβο, έχοντας ως αποτέλεσμα το κανάλι που διέρχεται από τη ρίζα να έχει πολύ μεγάλο φόρτο κίνησης. Κάθε κανάλι(διάυλος) που διαπερνά τη ρίζα μπορεί να έχει διακλαδώσεις, οι οποίες κατ' επέκταση να έχουν και άλλες διακλαδώσεις, δημιουργώντας έτσι πολύπλοκα στρώματα.

Η τοπολογία δέντρου και η τοπολογία διαύλου εμφανίζουν παρόμοια πλεονεκτήματα και μειονεκτήματα. Κύριο πλεονέκτημα της είναι ότι όλο το δίκτυο χωρίζεται σε Star υποδίκτυα πράγμα το οποίο καθιστά εύκολη τη συντήρηση και τη διαχείριση του. Επιπλέον αν ένα υποδίκτυο υποστεί βλάβη δεν θα επηρεάσει τη λειτουργία του υπόλοιπου δικτύου.

Πρόσθετο όμως μειονέκτημα σε σύγκριση με την τοπολογία bus αποτελεί ο σημαντικός ρόλος της κεφαλής(head node) στην μετάδοση, όπου σε τυχόν βλάβη της έχουμε κατάρρευση ολόκληρου του υποδικτύου που διαχειρίζεται. Επίσης η επέκταση του δικτύου είναι εξαρτώμενη από τον τύπο του καλωδίου που χρησιμοποιείται.[5][10]



Εικόνα 1-7 Τοπολογία Δέντρου (Tree Topology)

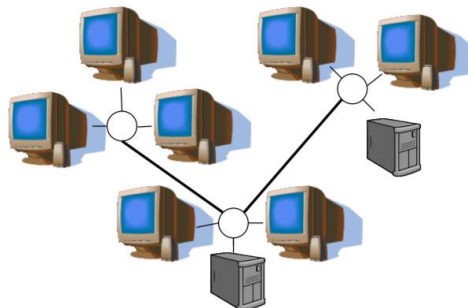


1.2.5 Υβριδική Τοπολογία(Hybrid Topology)

Με τον όρο υβριδική ονομάζουμε κάθε συνδυασμό των παραπάνω μεθόδων που αναφέραμε. Χαρακτηριστικό παράδειγμα είναι μια τοπολογία δέντρου η οποία συνδυάζει τα χαρακτηριστικά των γραμμικών τοπολογιών bus και αστέρα. Βασίζεται σε ομάδες διαμορφωμένων τερματικών σταθμών που συνδέονται με ένα γραμμικό βασικό καλώδιο bus. Οι τοπολογίες αυτές έχουν τη δυνατότητα να αναμιχθούν. Για παράδειγμα ένα δίκτυο διαύλου-αστέρα απαρτίζεται από ένα κανάλι υψηλού-εύρους ζώνης, αποκαλούμενο σπονδυλική στήλη και συνδέει επιμέρους δίκτυα υπολογιστών με τοπολογία αστέρα παρουσιάζοντας χαμηλή ταχύτητα.

Χρησιμοποιώντας δίκτυα υβριδικής τοπολογίας έχουμε τη δυνατότητα να προβλέψουμε κάποια λάθη και να τα περιορίσουμε. Επιπρόσθετα ενδεχόμενη επέκταση στο δίκτυο μας γίνεται αρκετά εύκολα και δεν επηρεάζει την τρέχουσα αρχιτεκτονική η το δίκτυο σε θέμα λειτουργίας. Αφού μιλάμε για υβριδική τοπολογία η οποία όπως προαναφέρθηκε είναι αρκετά ευέλικτη και προσαρμόζεται σύμφωνα με τις απαιτήσεις και χρησιμοποιεί σε μέγιστο βαθμό τους διαθέσιμους πόρους.

Ιδιαίτερη προσοχή πρέπει να δοθεί στο σχεδιασμό ενός υβριδικού δικτύου καθώς πρέπει να προσαρμόζεται στις απαιτήσεις του εκάστοτε οργανισμού, πράγμα το οποίο πολλές φορές δεν είναι εφικτό. Η πολυπλοκότητα των τοπολογιών υβριδικής αρχιτεκτονικής είναι αρκετά σύνθετη, οι τοπολογίες είναι μεγάλες σε έκταση και αυτό έχει ως αποτέλεσμα τη χρήση πολλαπλών ζευξεων και διάφορων δικτυακών και άλλων υπολογιστικών συσκευών. [5][11]



Εικόνα 1-8 Υβριδική Τοπολογία (Hybrid Topology)



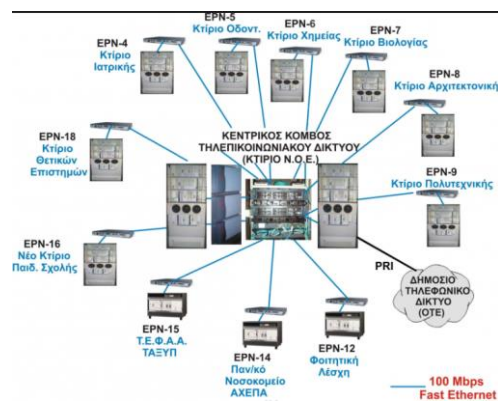
1.3 Βασικοί Τύποι Δικτύωσης

1.3.1 Δίκτυα Προσωπικής Περιοχής (PAN)

Τα δίκτυα προσωπικής περιοχής (Personal area Networks PAN) επιτρέπουν σε συσκευές να επικοινωνούν μέσα στην εμβέλεια ενός ατόμου. Ένα κλασικό παράδειγμα είναι το ασύρματο δίκτυο που συνδέει τον υπολογιστή με τις περιφερειακές συσκευές του. Σχεδόν σε κάθε υπολογιστή συνδέεται μία οθόνη, ένα πληκτρολόγιο, ένα ποντίκι και ένας εκτυπωτής. Αν δεν χρησιμοποιηθεί ασύρματη τεχνολογία, οι συνδέσεις αυτές θα πρέπει να γίνουν με καλώδια.

Για την διευκόλυνση των χρηστών, ορισμένες εταιρείες αποφάσισαν να συνεργαστούν στη σχεδίαση ενός ασύρματου δικτύου μικρής εμβέλειας που ονομάζεται Bluetooth, το οποίο συνδέει αυτά τα περιφερειακά χωρίς την ανάγκη των καλωδίων. Για πολλούς ανθρώπους αυτή η ευκολία χρήσης αποτελεί μεγάλο πλεονέκτημα. Στην απλούστερη μορφή τους, τα δίκτυα Bluetooth χρησιμοποιούν το υπόδειγμα κυρίου-υπηρέτη (master-slave).

Ο κύριος είναι υπεύθυνος για το ποιες διευθύνσεις και σε ποιες συχνότητες θα χρησιμοποιούνται, τότε και για πόσο θα πραγματοποιείται εκπομπή. Τα δίκτυα PAN μπορούν επίσης να χρησιμοποιούν και άλλες τεχνολογίες που επικοινωνούν σε μικρές αποστάσεις όπως τα RFID που βρίσκονται σε έξυπνες κάρτες. [12]



Εικόνα 1-9 Δίκτυα Προσωπικής Περιοχής (PAN)



1.3.2 Τοπικά Δίκτυα (Local Area Networks, LAN)

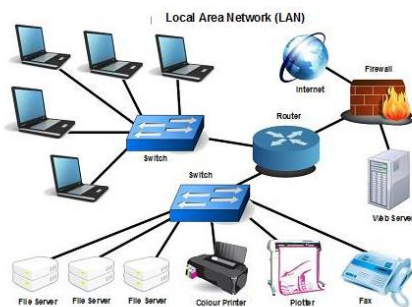
Το επόμενο βήμα είναι τα τοπικά δίκτυα (Local Area Networks, LAN) που συνήθως αποκαλούνται LAN. Τα δίκτυα LAN είναι ιδιωτικά δίκτυα τα οποία βρίσκονται μέσα και γύρω από ένα μόνο κτίριο, για παράδειγμα μια κατοικία, γραφείο ή εργοστάσιο. Χρησιμοποιούνται ευρέως για τη διασύνδεση προσωπικών υπολογιστών και ηλεκτρικών συσκευών με στόχο την κοινοχρησία πόρων (πχ εκτυπωτές) και την ανταλλαγή πληροφοριών. Όταν τα δίκτυα LAN χρησιμοποιούνται από εταιρίες αποκαλούνται εταιρικά δίκτυα (enterprise networks).

Στις μέρες μας τα ασύρματα δίκτυα LAN είναι ιδιαίτερα δημοφιλή, ειδικά σε σπίτια, παλαιότερα κτίρια γραφείων, καφετέριες και άλλα μέρη όπου είναι πολύ δύσκολη η τοποθέτηση καλωδίων. Στα συστήματα αυτά το κάθε τερματικό διαθέτει ένα ραδιομόντεμ και μια κεραία που χρησιμοποιεί για να επικοινωνήσει με τα υπόλοιπα. Στις περισσότερες περιπτώσεις, ο κάθε υπολογιστής επικοινωνεί με μια συσκευή, οποία ονομάζεται Σημείο Πρόσβασης (Access Point) ή ασύρματος δρομολογητής (wireless router) ή σταθμός βάσης (base station). Ειδικότερα, ο ρόλος του είναι η αναμετάδοση πακέτων μεταξύ των ασύρματων υπολογιστών, καθώς και μεταξύ των υπολογιστών αυτών και του διαδικτύου. Σε περίπτωση που οι άλλοι υπολογιστές βρίσκονται αρκετά κοντά μεταξύ τους, μπορούν να επικοινωνήσουν μεταξύ τους άμεσα σε μια ομότιμη διεύθυνση.

Τα ενσύρματα δίκτυα LAN χρησιμοποιούν αρκετές διαφορετικές τεχνολογίες μετάδοσης. Πιο συγκεκριμένα, οι περισσότερες από αυτές χρησιμοποιούν καλώδια χαλκού, αλλά και οπτικές ίνες. Ακόμη, έχουν περιορισμένο μέγεθος, γεγονός που σημαίνει ότι ο χρόνος μετάδοσης στη χειρότερη περίπτωση βρίσκεται εντός συγκεκριμένων ορίων και είναι γνωστός εκ των προτέρων. Η γνώση αυτού του ορίου μας επιτρέπει να σχεδιάσουμε πρωτόκολλα δικτύου. Τυπικά τα δίκτυα LAN έχουν ταχύτητες από 100 Mbps έως 1 Gbps, έχουν χαμηλή καθυστέρηση (μικροδευτερόλεπτα ή νανοδευτερόλεπτα) και παρουσιάζουν πολύ λίγα σφάλματα. Τα πιο σύγχρονα δίκτυα LAN μπορούν να λειτουργούν σε ταχύτητα μέχρι 10 Gbps. Σε σύγκριση με τα ασύρματα δίκτυα, τα ενσύρματα δίκτυα LAN τα ξεπερνούν σε όλους τους τομείς των επιδόσεων. Είναι απλώς ευκολότερο να στείλει κανείς σήματα μέσω ενός καλωδίου ή μιας οπτικής ίνας, παρά μέσω του αέρα.

Η τοπολογία σε πολλά ενσύρματα LAN χτίζεται πάνω σε συνδέσμους σημείου προς σημείο (point-to-point). Το πρότυπο IEEE 802.3, είναι με διαφορά ο πιο δημοφιλής τύπος ενσύρματου δικτύου LAN.

Στις μέρες μας, οι περισσότερες ηλεκτρικές συσκευές του σπιτιού έχουν τη δυνατότητα επικοινωνίας με τις υπόλοιπες και ταυτόχρονα όλες έχουν πρόσβαση στο διαδίκτυο. Πολλές συσκευές διαθέτουν τη δυνατότητα δικτύωσης, όπως είναι φορητοί υπολογιστές (laptops), tablets, έξυπνες τηλεοράσεις (smart-TVs) και εγκαταστάσεις υποδομής όπως μετρητές και θερμοστάτες. [12]



Εικόνα 1-10 Δίκτυα Τοπικής Περιοχής (LAN)

1.3.3 Ασύρματα Τοπικά Δίκτυα (Wireless Local Area Networks, WLAN)

Υπάρχει για τα ασύρματα LAN ένα πρότυπο με όνομα IEEE 802.11 ευρέως γνωστό ως Wi-Fi, το οποίο έχει γίνει πολύ διαδεδομένο. Λειτουργεί σε ταχύτητες από 11 έως και εκατοντάδες Mbps. Γενικότερα, μετράται η ταχύτητα των γραμμών σε megabit/δευτερόλεπτο, όπου 1 Mbps είναι 1.000.000 bit/δευτερόλεπτο και σε gigabit/δευτερόλεπτο, όπου 1 Gbps είναι 1.000.000.000 bit/δευτερόλεπτο. Τα δίκτυα 802.11 μπορούν να χρησιμοποιηθούν σε δυο καταστάσεις λειτουργίας.

Αρχιτεκτονική

Η πιο διαδεδομένη κατάσταση λειτουργίας είναι η σύνδεση πελατών, για παράδειγμα φορητών υπολογιστών και έξυπνων τηλεφώνων, με ένα άλλο δίκτυο όπως το εταιρικό ενδοδίκτυο ή το Internet. Στην κατάσταση υποδομής, ο κάθε πελάτης συσχετίζεται με ένα Σημείο Πρόσβασης (Access Point), το οποίο με την σειρά του συνδέεται με το άλλο δίκτυο. Ο πελάτης στέλνει και λαμβάνει τα πακέτα του μέσω αυτού. Μπορούν να υπάρχουν πολλά σημεία πρόσβασης που συνδέονται μεταξύ τους, τυπικά με ένα ενσύρματο δίκτυο που ονομάζεται σύστημα διανομής (distribution system), για τον σχηματισμό ενός εκτεταμένου δικτύου 802.11. Στην περίπτωση αυτή, οι πελάτες μπορούν να στέλνουν τα πλαίσια σε άλλους πελάτες μέσω των σημείων πρόσβασής τους. Κάθε ασύρματος σταθμός 802.11 έχει μια διεύθυνση MAC 6 bytes, που αποθηκεύεται στο υλικολογισμικό (firmware) του προσαρμογέα του σταθμού (δηλ. στην κάρτα διεπαφής δικτύου 802.11). Κάθε σταθμός βάσης έχει μια διεύθυνση MAC για την ασύρματη διεπαφή της.

Με τον όρο υποδομή, τα ασύρματα LAN υποδομής (infrastructure wireless LANs) αναφέρονται στα AP καθώς και στην ενσύρματη υποδομή Ethernet, που συνδέει τα σημεία πρόσβασης με έναν δρομολογητή. Οι σταθμοί IEEE 802.11 μπορούν επίσης να ομαδοποιηθούν για να δημιουργήσουν ένα ad hoc δίκτυο – ένα δίκτυο χωρίς κεντρικό έλεγχο και χωρίς συνδέσεις με τον "έξω κόσμο". Στην συγκεκριμένη περίπτωση το δίκτυο δημιουργείται από κινητές συσκευές, που έχουν βρεθεί, οι οποίες έχουν την ανάγκη να επικοινωνήσουν και δεν βρίσκουν προϋπάρχουσα υποδομή δικτύου στις τοποθεσίες τους. Ένα ad hoc δίκτυο μπορεί να δημιουργηθεί όταν συναθροίζονται άνθρωποι με φορητούς υπολογιστές (πχ σε μια αίθουσα διασκέψεων). [12]

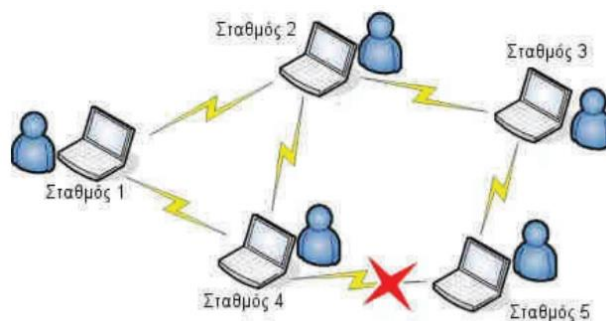


Εικόνα 1-11 Ασύρματα Τοπικά Δίκτυα (WLAN)

Πλεονεκτήματα Ad-Hoc δικτύου

Ο απομακρυσμένος χαρακτήρας των wireless ad hoc δικτύων τα χαρακτηρίζει συμβατά και επιθυμητά σε διάφορες εφαρμογές οι οποίες δεν είναι βασισμένες σε κεντρικούς κόμβους και default WLAN δίκτυα. Η ταχύτατη εγκατάσταση και η μηδαμινή σχεδόν διαδικασία παραμετροποίησης τους τα καθιστά χρήσιμα σε ακραίες περιπτώσεις όπου χρειαζόμαστε άμεσα δημιουργία και εγκατάσταση δικτύου όπως παραδείγματος χάριν μετά από μεγάλες φυσικές καταστροφές. Ακόμη δίνουν τη δυνατότητα διόρθωσης της δικτύωσης σε περιπτώσεις δυσλειτουργίας κάποιου κόμβου αφού πολύ εύκολα μπορεί να χρησιμοποιηθεί νέα διαδρομή δρομολόγησης.

Βλέποντας την παρακάτω εικόνα βγάζουμε το συμπέρασμα ότι αν παρουσιαστεί βλάβη μεταξύ του σταθμού 4 και 5 αυτό δεν σημαίνει ότι θα διακοπεί η επικοινωνία και με τους υπόλοιπους σταθμούς αφού μπορεί και ακολουθεί εναλλακτική διαδρομή διαμέσων των σταθμών 2 και 3.



Εικόνα 1-12 Ασύρματο δίκτυο τοπολογίας ad-hoc με παρουσία βλάβης

Συμπερασματικά τα δίκτυα ad hoc είναι αρκετά ευέλικτα και κατάλληλα για την point to point σύνδεση δύο συσκευών, δίχως να χρησιμοποιήσουμε ένα κεντρικό σημείο πρόσβασης.

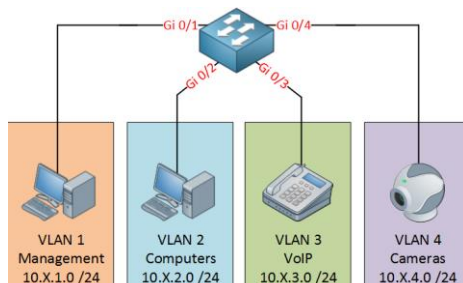


Μειονεκτήματα Ad-Hoc δικτύου

Βασικό μειονέκτημα θεωρείται η υψηλή απαίτηση σε πόρους από τα μηχανήματα για να διατηρηθεί η σύνδεση αν αυτές αλλάζουν τη θέση τους. Ακόμη η εμβέλεια στα συστήματα σύνδεσης είναι αρκετά μικρότερη συγκριτικά με ένα αμετάβλητο σημείο πρόσβασης. Το κυριότερο μειονέκτημα των δικτύων ad hoc είναι ότι αδυνατούν να προβλέψουν όλο το εύρος των καταστάσεων και των προβλημάτων που μπορούν να συμβούν εξαιτίας του μεταβλητού χαρακτήρα που έχει η εγκαθίδρυση της επικοινωνίας. [13]

1.3.4 Εικονικά Τοπικά Δίκτυα (Virtual Local Area Networks, VLAN)

Μπορούμε επίσης να χωρίσουμε ένα μεγάλο φυσικό δίκτυο LAN σε δυο μικρότερα λογικά LAN. Αυτό συμβαίνει, διότι η διάταξη του δικτυακού εξοπλισμού δεν ταιριάζει με την οργανωτική δομή. Για παράδειγμα, τα τμήματα τεχνολογίας και οικονομικών μιας επιχείρησης μπορεί να έχουν υπολογιστές που βρίσκονται στο ίδιο φυσικό δίκτυο, επειδή βρίσκονται στην ίδια πτέρυγα του κτηρίου, όμως η διαχείρισή τους θα ήταν ευκολότερη αν το καθένα από αυτά είχε το δικό του Εικονικό δίκτυο LAN (Virtual LAN). Σε αυτό τον σχεδιασμό η κάθε θύρα θα έχει το δικό της "χρώμα", έστω το πράσινο για το τμήμα τεχνολογίας και το κόκκινο για το τμήμα οικονομικών. Στη συνέχεια ο μεταγωγέας προωθεί τα πακέτα έτσι ώστε οι υπολογιστές που είναι συνδεδεμένοι σε πράσινες θύρες να είναι χωρισμένοι από εκείνους που είναι συνδεδεμένοι με κόκκινες. Έτσι τα πακέτα ευρείας εκπομπής που στέλνονται σε μια κόκκινη θύρα δεν θα παραληφθούν από μια πράσινη θύρα σαν να επρόκειτο για δυο διαφορετικά LAN.



Εικόνα 1-13 Εικονικά Τοπικά Δίκτυα (VLAN)

Πλεονεκτήματα Εικονικών Τοπικών Δικτύων (VLAN)

➤ *Έλεγχος της κίνησης του δικτύου*

Σε ένα δίκτυο όσο περισσότερες συσκευές έχουμε τόσο η κίνηση του μεγαλώνει με αποτέλεσμα την χαμηλή απόδοση της συνολικής τοπολογίας. Χρησιμοποιώντας την τεχνική τμηματοποίησης ενός δικτύου σε υποδίκτυα VLAN, περιορίζουμε την κίνηση μέσα στα όρια του εκάστοτε VLAN.

➤ *Ασφάλεια*

Διαχωρίζοντας ένα Local δίκτυο σε άλλα εικονικά υποδίκτυα, μπορώ να έχω κόμβους όπου έχουν αυξημένες ανάγκες σε θέματα ασφάλειας οι οποίοι αποτελούν ένα



αυτόνομο VLAN από μόνοι τους και έτσι να μην γίνεται εύκολη η πρόσβαση από κόμβους που χρησιμοποιούν οι άλλοι χρήστες.

➤ *Ενοποιημένοι πόροι*

Για να έχουμε υψηλότερη απόδοση σε θέματα που αφορούν την εξυπηρέτηση μπορούμε να ομαδοποιήσουμε σε ένα αυτόνομο VLAN συσκευές-χρήστες με παρόμοιες απαιτήσεις ή και ίδιες ανάγκες από το δίκτυο. Παραδείγματος χάριν ενοποιημένοι τηλεφωνικοί κόμβοι σε ένα εικονικό τοπικό δίκτυο έτσι ώστε να έχουμε μικρότερη καθυστέρηση στην ανταλλαγή και στη διεκπεραίωση πακέτων.

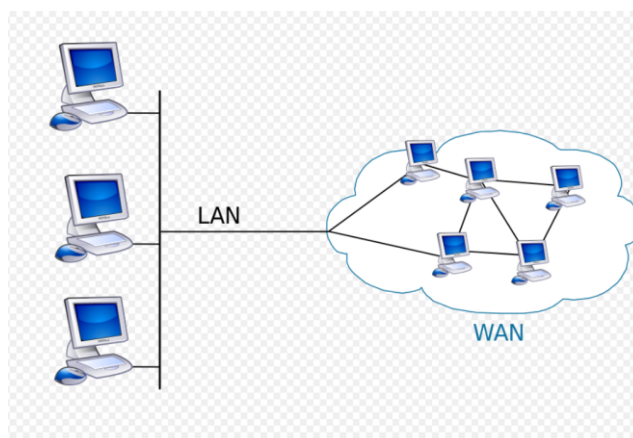
➤ *Εύκολα διαχειρίσιμο και εξυπηρετήσιμο δίκτυο*

Το μοναδικό μειονέκτημα σε μία καλά μελετημένη υλοποίηση με εικονικό τοπικό δίκτυο είναι η αποφυγή επέκτασης του *ίδιου εικονικού δικτύου* πάνω από ένα κτίριο έτσι ώστε ο διαχειριστής να έχει μια ολοκληρωμένη και ευδιάκριτη εικόνα του συνολικού δικτύου. [12]

1.3.5 Δίκτυα Ευρείας Περιοχής (WAN)

Ένα δίκτυο ευρείας περιοχής ή ζώνης (Wide Area Network, WAN) αποτελεί μία ομάδα υπολογιστικών μηχανών που έχουν έκταση σε μια ευρεία γεωγραφική περιοχή [ή πιο απλά πολλά LAN μαζί] δημιουργώντας μεταξύ τους ένα δίκτυο επικοινωνίας (π.χ. η δικτύωση των υποκαταστημάτων της Adidas σε Ευρώπη, Ασία, Αφρική).

Τα δίκτυα ευρείας περιοχής τις περισσότερες φορές διασυνδέουν μεταξύ τους τοπικά δίκτυα υπολογιστών (*LAN networks*). Για τη συγκεκριμένη υλοποίηση γίνεται χρήση μισθωμένων δημόσιων τηλεπικοινωνιακών γραμμών ή και δορυφορικών τηλεπικοινωνιών. Το πιο δημοφιλές δίκτυο ευρείας περιοχής είναι το Διαδίκτυο (Internet). Η δικτύωση των τερματικών ακολουθεί συνήθως τα πρότυπα ενός LAN network (τύπου star, ring, διαύλου, bus) ή τα πρότυπα του Διαδικτύου. [12]



Εικόνα 1-14 LAN δίκτυο συνδεδεμένο σε WAN



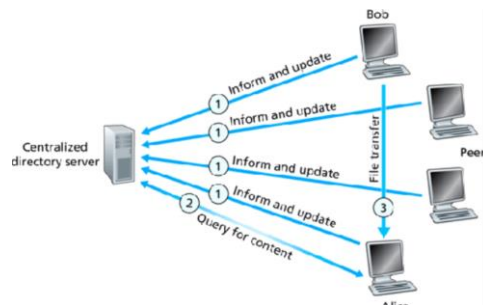
1.3.5.1 Βασικές Αρχιτεκτονικές των Δικτύων Ευρείας Περιοχής (WAN)

Οι δύο κύριες αρχιτεκτονικές που συναντάμε στα Δίκτυα Ευρείας Περιοχής (WAN) είναι αυτές του Πελάτη-Διακομιστή (*Client-Server*) και των ισότιμων-ομότιμων δικτύων (*peer-to-peer* [P2P]). [14]

Ομότιμα-Ισότιμα Δίκτυα(P2P)

Σε ένα ισότιμο δίκτυο, όπως γίνεται αντιληπτό και από την ονομασία του, όλα τα τερματικά τα λαμβάνουμε ως ίσα (ομότιμα) μεταξύ τους, χωρίς δηλαδή να υπάρχει κάποια συγκεκριμένη ιεραρχία μεταξύ τους. Δεν έχουμε ούτε *αποκλειστικούς διακομιστές* (dedicated servers), ούτε κάποιον *head node* (κεντρικό κόμβο) που έχουν ευθύνη εποπτείας ή για την παροχή υπηρεσιών στα άλλα τερματικά του δικτύου.

Το κάθε τερματικό έχει ταυτόχρονη λειτουργία πελάτη και διακομιστή. Ως απόρροια αυτού είναι η απουσία κάποιου κεντρικού διαχειριστή υπεύθυνου για τη λειτουργία ολόκληρου το δικτύου, με τον κάθε χρήστη να μπορεί να καθορίζει το σύνολο των δεδομένων που θα είναι κοινώς προσβάσιμα και θα μπορούν να χρησιμοποιηθούν από τους υπόλοιπους χρήστες του δικτύου. Ένα παράδειγμα ομότιμου δικτύου φαίνεται στο παρακάτω σχήμα:



Εικόνα 1-15 Δίκτυο Peer to Peer (P2P)

Στα ισότιμα δίκτυα, όλοι οι χρήστες του συστήματος είναι ίσοι μεταξύ τους. Ο κάθε χρήστης έχει τη δυνατότητα πρόσβασης σε αρχεία, καταλόγους και γενικά στους υπόλοιπους πόρους (π.χ. εκτυπωτή) ενός άλλου χρήστη. Απαραίτητη προϋπόθεση είναι ο τελευταίος να τους έχει θέσει ως *κοινόχρηστους* (*shared*). Σε μικρού εύρους δίκτυα η συγκεκριμένη τακτική έχει πολύ καλή απόδοση. Το μέγεθος των συγκεκριμένων δικτύων είναι αρκετά μικρό, περίπου 10 έως 30 χρήστες το μέγιστο. Λόγω του περιορισμένου τους μεγέθους, τα ομότιμα δίκτυα ονομάζονται και *ομάδες εργασίας* (*workgroups*).

Η διαδικασία υλοποίησης ενός ομότιμου δικτύου είναι σχετικά απλή. Το μόνο που κάνουμε είναι μία σύνδεση όλων των τερματικών μεταξύ τους, καθιστώντας το κάθε τερματικό υπεύθυνο μονάχα για τον έλεγχο και το διαμοιρασμό των δικών του δεδομένων. Αυτό έχει ως αποτέλεσμα ότι δεν είναι απαραίτητη η εγκατάσταση ενός ή περισσότερων ισχυρών διακομιστών, οι οποίοι θα πρέπει να είναι ικανοί να παρέχουν υπηρεσίες σε όλους τους υπολογιστές του δικτύου. Έτσι έχουμε χαμηλό κόστος υλοποίησης μιας και έχουμε σχετικά περιορισμένες απαιτήσεις σε υλικό.



Επιπλέον στα ομότιμα δίκτυα το λογισμικό που χρειαζόμαστε για την δικτύωση δεν είναι απαραίτητο να έχει επίπεδο απόδοσης και ασφάλειας συγκριτικά με το αντίστοιχο λογισμικό που χρησιμοποιούμε για διακομιστές δικτύου, όπου εκεί η ασφάλεια και η απόδοση παίζουν το σημαντικότερο ρόλο. Επίσης, έχοντας ομότιμα δίκτυα ενσωματωμένα στα περισσότερα λειτουργικά συστήματα, συνήθως δεν είναι αναγκαία η εγκατάσταση επιπλέον λογισμικού.

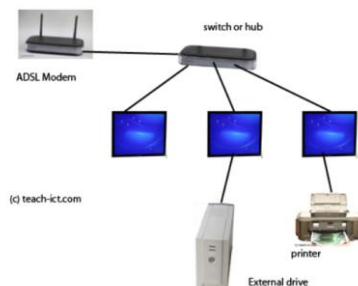
Πλεονεκτήματα Ομότιμων Δικτύων

- Η σύνδεση των υπολογιστών είναι απλή και εύκολη στην κατανόηση
- Δεν χρειάζεται κεντρικός διαχειριστής, έτσι οι χρήστες μπορούν και αποφασίζουν τον τρόπο που θα διασφαλίζουν και διαμοιράζουν τα δεδομένα τους

Μειονεκτήματα Ομότιμων Δικτύων

Παραδείγματος χάριν έχουμε ένα δίκτυο στο οποίο υπάρχει μόνο ένας εκτυπωτής, ο οποίος είναι συνδεδεμένος σε έναν υπολογιστή(τον λέμε *main*) και τον διαμοιράζονται όλα τα υπόλοιπα τερματικά. Για αυτό το λόγο ο *main* υπολογιστής θα πρέπει να έχει ορίσει τον εκτυπωτή του ως *κοινόχρηστο*, έτσι ώστε να μπορούν να τον χρησιμοποιήσουν και οι υπόλοιποι. Άρα ο *main* έχει το ρόλο του *διακομιστή* όσον αφορά τις εκτυπώσεις, διεκπεραιώνοντας τις αιτήσεις που έρχονται για εκτύπωση από τα υπόλοιπα τερματικά.

Το μειονέκτημα που έχουμε σε αυτήν την περίπτωση, είναι ότι αν ο *main* αδρανήσει είτε δυσλειτουργήσει τότε κανένα τερματικό δεν θα μπορεί να χρησιμοποιήσει τον εκτυπωτή. Επίσης, αν ένα άλλο τερματικό (X) κάνει επανεκκίνηση τη στιγμή που ένα άλλο τερματικό (Y) χρησιμοποιεί κάποιον από τους πόρους του, τότε το (Y) θα αποσυνδεθεί και δε θα μπορεί να χρησιμοποιήσει άλλο τον πόρο. Ένα σχηματικό παράδειγμα δίνεται πιο κάτω:



Εικόνα 1-16 Δίκτυο Peer to Peer (P2P) κοινός εκτυπωτής

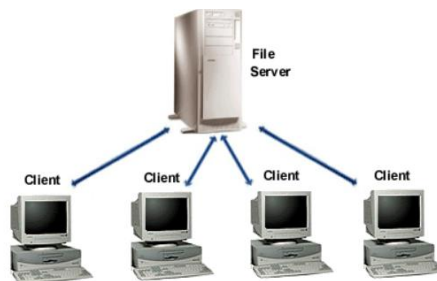


Επίσης χαρακτηριστικό μειονέκτημα της ομότιμης δικτύωσης είναι ότι χαρακτηρίζεται από χαμηλό επίπεδο επιδόσεων. Αυτό οφείλεται στο γεγονός του διαμοιρασμού των πόρων μεταξύ των τερματικών. Αν παραδείγματος χάριν ο main υπολογιστής προσπελάζει τους πόρους του υπολογιστή (X), τότε ο main κατ' επέκταση θα κάνει χρήση ενός ποσοστού από τους πόρους του (X), όπως χρόνο, επεξεργαστική ισχύς και μεγάλο μέρος της μνήμης του. Ως αποτέλεσμα ανεξαρτήτως δυνατοτήτων του κάθε τερματικού, τα επίπεδα απόδοσης του θα μειώνονται κάθε φορά που κάποιο άλλο τερματικό(user)θα θέλει να χρησιμοποιήσει τους πόρους του.

Εν τέλει λαμβάνοντας υπόψη και τα πιο πάνω μειονεκτήματα, τα ομότιμα δίκτυα παρουσιάζονται σαν ιδανική λύση για μικρές δικτυακές τοπολογίες όπως μια μικρή επιχείρηση ή ένα home network. [12][15][16]

Δίκτυα Πελάτη- Διακομιστή(Client-Server)

Όταν σε ένα δίκτυο ο αριθμός των χρηστών είναι αρκετά μεγάλος (μεγαλύτερο από 30 χρήστες), τότε η επιλογή ενός ομότιμου δικτύου δεν αποτελεί την καλύτερη επιλογή. Στη συγκεκριμένη περίπτωση, ένα *Client-Server network* αποτελεί μια καλύτερη λύση. Τέτοιου τύπου δίκτυα αποτελούνται από αρκετούς ιδιόκτητους διακομιστές (dedicated servers). Οι dedicated servers είναι υπολογιστικές μηχανές, οι οποίες έχουν αποκλειστική λειτουργία διακομιστή και όχι πελάτη ή *work station*. Οι διακομιστές έχουν την ικανότητα εξαιτίας του πολύ γρήγορου λογισμικού τους να εξυπηρετούν σε ταχύρυθμα τα αιτήματα των πελατών του δικτύου εξασφαλίζοντας πάντα την ασφάλεια των αρχείων. Ένα παράδειγμα Client-Server δικτύου βλέπουμε πιο κάτω.



Εικόνα 1-17 Δίκτυο Πελάτη - Διακομιστή (Client-Server)

Στην παραπάνω εικόνα βλέπουμε ότι υπάρχει μόνο ένας διακομιστής(server), ο οποίος λειτουργεί σαν διακομιστής αρχείων (file & print server). Καθώς μεγαλώνει το μέγεθος της τοπολογίας του δικτύου, τόσο μεγαλώνει το μέγεθος της κυκλοφορίας και οι αποστάσεις ενδιάμεσα των συνδεδεμένων τερματικών, καθιστώντας απαραίτητη τη χρήση επιπλέον διακομιστών. Έχοντας περισσότερους servers σε ένα δίκτυο ο διαμοιρασμός της εργασίας είναι καλύτερος και ταυτόχρονα γίνεται ταχύτερη και αποτελεσματικότερη η εξυπηρέτηση των αιτημάτων των πελατών. Τα είδη των διακομιστών που υπάρχουν είναι πάρα πολλά μιας και ο κάθε διακομιστής μπορεί να διαχειρίζεται διαφορετικά αιτήματα. Αυτού του τύπου διακομιστές λέγονται (*specialized servers*), επειδή έχουν εξειδικευμένο λογισμικό για να ικανοποιούν



συγκεκριμένα αιτήματα. Οι μεγάλες επιχειρήσεις χρησιμοποιούν μια μεγάλη γκάμα εξειδικευμένων διακομιστών. Ορισμένοι *specialized servers*:

- *διακομιστές αρχείων & εκτύπωσης*
- *διακομιστές εφαρμογών (application servers)* εκτελούνται στο server και αποτελούν το 50% των εφαρμογών (*client-server*). Το άλλο 50% είναι οι εφαρμογές πελάτη και εκτελούνται στο τοπικό δίκτυο των πελατών και από εκεί μπορούν και αποστέλουν αίτημα στο διακομιστή εφαρμογών. Χαρακτηριστικό παράδειγμα αποτελεί ο διακομιστή βάσης δεδομένων (*database server*) όπου εδώ είναι αποθηκευμένοι τεράστιοι όγκοι δεδομένων κατάλληλα οργανωμένοι για την εύκολη αναζήτηση και ανάκτηση.
- *διακομιστές Υπηρεσιών Καταλόγου (Directory Service Servers)*, όπου με τις υπηρεσίες καταλόγου, οι χρήστες μπορούν να εντοπίζουν, να αποθηκεύουν και να διασφαλίζουν τα δεδομένα τους στο δίκτυο.

Ένα δίκτυο πελάτη-διακομιστή παρουσιάζει περισσότερα πλεονεκτήματα συγκριτικά με ένα ομότιμο δίκτυο. Ορισμένα απ' αυτά είναι:

- *Μεγαλύτερη ασφάλεια.* Στα *client-server networks* υπάρχει ένας κεντρικός *server* ο οποίος ελέγχει και διαχειρίζεται τα θέματα ασφαλείας όλων των χρηστών του δικτύου με τη χρήση μιας προκαθορισμένης πολιτικής. Ακολουθώντας την πολιτική αυτή έχουμε αύξηση στα επίπεδα της ασφαλείας του δικτύου, αντίθετα με τα ομότιμα δίκτυα, όπου τα επίπεδα είναι χαμηλά, καθώς η ασφάλεια των δεδομένων είναι προσωπική υπόθεση του κάθε χρήστη.
- *Μεγαλύτερος αριθμός χρηστών.* Τα *client-server networks* δύνανται για την υποστήριξη μεγάλου αριθμού χρηστών σε αντίθεση με τα ομότιμα δίκτυα που καθίσταται αδύνατη τέτοια υποστήριξη. Τα προγράμματα εποπτείας και διαχείρισης δικτύων (*Network Monitoring and Management Software*), έχουν τη δυνατότητα χειρισμού χιλιάδων χρηστών από ένα κεντρικό σημείο, χρίζοντας τα *client-server networks* κατάλληλα για τη διαχείριση μεγάλων δικτυακών τοπολογιών.
- *Χρήση Εφεδρικού Αντίγραφου.* Έχοντας τα ευαίσθητα δεδομένα (ονόματα χρηστών, κωδικοί πρόσβασης) συγκεντρωμένα σε διάφορους διακομιστές ανά το διαδίκτυο, η λήψη αντίγραφων ασφαλείας είναι μία καλή λύση και οφείλει να πραγματοποιείται ανά τακτά χρονικά διαστήματα. Έτσι το δίκτυο είναι ικανό να ανταπεξέλθει σε ακραίες καταστάσεις (π.χ. μερική ή ολική καταστροφή δεδομένων ενός server).
- *Ευελιξία στο hardware.* Στα *client-server networks*, το υλικό του υπολογιστή - πελάτη προσαρμόζεται στις ανάγκες εκάστοτε χρήστη. Έτσι οι πελάτες μπορεί να έχουν πολύ μικρό ποσοστό σε μνήμες (RAM, αποθηκευτικό χώρο στο σκληρό δίσκο), αφού δεν είναι απαραίτητο να παρέχουν υπηρεσίες στο δίκτυο, ή να αποθηκεύουν «εξωτερικά» δεδομένα, όπως κάνουν οι διακομιστές. [12][15][16]



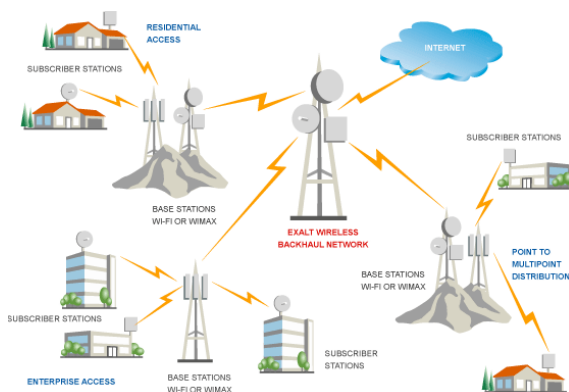
1.3.6 Μητροπολιτικά Δίκτυα (Metropolitan Area Networks, MAN)

Το μητροπολιτικό δίκτυο (metropolitan area network), ή δίκτυο MAN, καλύπτει μια πόλη. Το πιο γνωστό παράδειγμα δικτύου MAN είναι το δίκτυο καλωδιακής τηλεόρασης που υπάρχει σε πολλές πόλεις. Αυτό το σύστημα είναι εξέλιξη των παλαιότερων συστημάτων κοινοτικών κεραιών που χρησιμοποιούνταν σε περιοχές με κακή τηλεοπτική λήψη από αέρος. Σε αυτά τα πρώιμα συστήματα, μια μεγάλη κεραιά ήταν τοποθετημένη στην κορυφή ενός κοντινού λόφου και στην συνέχεια το σήμα στέλλονταν στα σπίτια των συνδρομητών.

Με την ολοένα και αυξανόμενη χρήση του Internet, οι επιχειρήσεις δικτύου καλωδιακής τηλεόρασης άρχισαν να αντιλαμβάνονται ότι με ορισμένες αλλαγές στο σύστημα θα μπορούσαν να παρέχουν αμφίδρομες υπηρεσίες Internet σε μη χρησιμοποιούμενα τμήματα φάσματος. Γι' αυτό τον λόγο το σύστημα καλωδιακής τηλεόρασης άρχισε να μεταλλάσσεται από έναν τρόπο διανομής τηλεοπτικού σήματος σε ένα μητροπολιτικό δίκτυο. Όμως η καλωδιακή τηλεόραση δεν είναι το μοναδικό δίκτυο MAN. Οι πρόσφατες εξελίξεις ως προς την ασύρματη πρόσβαση υψηλής ταχύτητας στο Internet, είχαν ως αποτέλεσμα ένα άλλο δίκτυο MAN, το οποίο τυποποιήθηκε ως 802.16 και είναι γνωστό στο ευρύ κοινό ως WiMAX.

Αρχιτεκτονική

Οι σταθμοί βάσης συνδέονται απευθείας με το δίκτυο κορμού του παρόχου, το οποίο με τη σειρά του συνδέεται με το διαδίκτυο. Οι σταθμοί βάσης επικοινωνούν με τους άλλους σταθμούς μέσω της ασύρματης διασύνδεσης αέρα. Υπάρχουν δυο είδη σταθμών. Οι σταθμοί συνδρομητών παραμένουν σε μια σταθερή θέση, παρέχοντας έτσι πρόσβαση στο διαδίκτυο για τα σπίτια. Οι κινητοί σταθμοί μπορούν να λαμβάνουν υπηρεσίες "εν κινήσει", όπως για παράδειγμα συμβαίνει σε ένα αυτοκίνητο εξοπλισμένο με WiMAX.[12]



Εικόνα 1-18 Παροχή Υπηρεσιών WiMAX

1.3.7 Οικιακό Δίκτυο (Home Area Network ,HAN)

Το εν λόγω δίκτυο επιτρέπει σε συσκευές που βρίσκονται εντός ενός κτιρίου να επικοινωνούν μεταξύ τους. Γενικά όσο αφορά το Smart Grid, οι συσκευές αυτές θα μπορούσαν να περιλαμβάνουν έξυπνους μετρητές, έξυπνες συσκευές και συσκευές διαχείρισης ενέργειας σε ένα σπίτι. Στην ίδια κατηγορία παρατηρούμε επίσης τα Building Area Networks (BANs) και Industrial Area Networks (IANs).

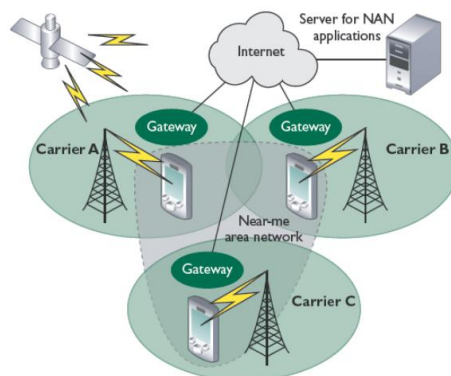


Οι υπηρεσίες ενός HAN δικτύου περιλαμβάνουν αυτοματοποίηση σπιτιών και κτιρίων, γεγονός το οποίο αναφέρεται στην αποστολή δεδομένων των μετρήσεων από τις συσκευές στα controllers εντός της μικρής αυτής ακτίνας. Οι εφαρμογές αυτές επίσης, δεν απαιτούν μετάδοση με ιδιαίτερα υψηλές ταχύτητες, οπότε οι επικοινωνίες του δικτύου αυτού πρέπει να είναι ενεργειακής κλάσης, δηλαδή χαμηλή σε κατανάλωση και κόστος, σχετικά απλές και ασφαλείς. Με τα παραπάνω συμπεραίνουμε ότι, αρκούν τεχνολογίες που προσφέρουν ταχύτητες έως 100kbps σε μικρή απόσταση της τάξης των 100 μέτρων. Τέλος, σύννηθες φαινόμενο στη λειτουργία εφαρμογών δικτύων HAN, ή BAN ή IAN είναι η χρήση ZigBee, Wi-Fi, Z-Wave, PLC (Power Line Carrier) ή HomePlug, Bluetooth και Ethernet.[17]

1.3.8 Near me Area Networks (NAN)

Με τον όρο NAN αναφερόμαστε σε δίκτυα μιας μικρής οριοθετημένης περιοχής, όπως για παράδειγμα μια γειτονιά. Ειδικότερα, τα τεμαχικά προσπαθούν να επιτύχουν την επικοινωνία με άλλα της περιοχής, έτσι δημιουργείται ένα διασυνδεδεμένο πλέγμα έξυπνων συσκευών.

Βασίζονται σε δικτύωση διευθύνσεων IPv6, προκειμένου να μπορέσουν να ανταπεξέλθουν στις ανάγκες του smart grid. Οι υπηρεσίες, αφορούν τις έξυπνες μετρήσεις, την κάλυψη της ζήτησης και την αυτοματοποίηση της διανομής, τα πακέτα πρέπει να μεταδίδονται από ένα μεγάλο αριθμό πελατών και συσκευών σε ένα συλλέκτη δεδομένων ή υποσταθμό και να υπάρχει αμφίδρομη ροή. Συμπερασματικά, απαιτούνται συνδέσεις υψηλότερων ταχυτήτων (100kbps-10Mbps) και μεγαλύτερη ακτίνα κάλυψης έως και 10 χιλιόμετρα. Η λειτουργία του δικτύου αυτού επιτυγχάνεται με πλέγμα δικτύων βασισμένων στο ZigBee, με πλέγμα δικτύων Wi-Fi, με PLC, καθώς και με ασύρματες και ενσύρματες ζεύξεις μεγάλων αποστάσεων, όπως WiMAX, Cellular, DSL και ομοαξονικό καλώδιο.[17]

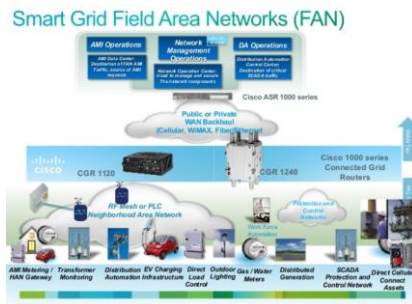


Εικόνα 1-19 Γειτονικά Δίκτυα (NAN)



1.3.9 Field Area Network (FAN)

Μια υποκατηγορία των NAN δικτύων, είναι τα Field Area Networks (FANs). Τα FAN αποτελούν ουσιαστικά το συνδυασμό των NAN και τοπικών συσκευών, που συνδέονται σε ένα Field Area δρομολογητή (FAR), προσφέροντας τις διεπαφές του WAN. Μπορούν να λειτουργήσουν ως υποδεέστερα δίκτυα για το σύνολο των συσκευών του έξυπνου ενεργειακού δικτύου.[17]



Εικόνα 1-20 Field Area Network (FAN)

1.4 Δομικά Στοιχεία Δικτύου

Δρομολογητής(router)

Ο δρομολογητής είναι η συσκευή στην οποία είναι συνδεδεμένα περισσότερα από ένα τερματικά ενός τοπικού δικτύου. Ο router είναι υπεύθυνος για τη μεταφορά των δεδομένων από και προς το κατάλληλο τερματικό του δικτύου, ακολουθώντας πάντα ορισμένα κριτήρια που θέτει ο διαχειριστής του, π.χ. IP address, πρωτόκολλα επικοινωνίας(NAT,DHCP) κ.ά. που θα αναλύσουμε εκτενέστερα παρακάτω. Ένας δρομολογητής (router) παίζει σημαντικό ρόλο καθώς διασυνδέει δύο δίκτυα, συνήθως ένα τοπικό ασύρματο (WLAN) και ένα τοπικό ενσύρματο (LAN), έτσι μιλάμε για έναν Wireless Router, είτε για τοπικό ενσύρματο (LAN) και ενός ευρέους δικτύου (WAN), όπως είναι το Internet για συνδέσεις μέσω xDSL2, έτσι μιλάμε για xDSL Router.

Ένας router πραγματοποιεί κατάλληλες δρομολογήσεις δεδομένων από ένα δίκτυο σε συγκεκριμένο υπολογιστικό μηχάνημα άλλου δικτύου. Στις περισσότερες περιπτώσεις έχει δύο IP διευθύνσεις, και έχει τη δυνατότητα σύνδεσης με παραπάνω από έναν υπολογιστές.

Επιπρόσθετα με τη χρήση ορισμένων διαδικασιών, αναλαμβάνει να δρομολογήσει τα εισερχόμενα πακέτα, στον κατάλληλο υπολογιστή. Η πλειοψηφία των routers της αγοράς, υποστηρίζουν το πρωτόκολλο NAT και έχουν τη δυνατότητα να κάνουν Port Forwarding, δηλαδή αντιστοίχιση συγκεκριμένης πόρτας σε συγκεκριμένη διεύθυνση IP του τοπικού δικτύου. Ένα παράδειγμα δίνεται παρακάτω.

Έστω η διεύθυνση ενός δρομολογητή στο Ίντερνετ xx.xx.xx.xx, προσπάθειες για επικοινωνία γίνονται σε συγκεκριμένη port, (εδώ βάζουμε 1111) αυτό γίνεται με τη χρήση της διεύθυνσης xx.xx.xx.xx:1111, έτσι θα κάνει τον router να αντιληφθεί ότι κάποιος υπολογιστής προσπαθεί να επικοινωνήσει μέσω της θύρας 1111 μαζί του. Ο δρομολογητής, μιας και είναι ρυθμισμένος να προωθεί σε μια static IP του τοπικού δικτύου (πχ 192.168.1.2) όλα τα εισερχόμενα πακέτα που θα φθάσουν μέσω της θύρας 1111 σε αυτό, θα προωθήσει τα πακέτα



στη διεύθυνση 192.168.1.2. Ως αποτέλεσμα οποιοσδήποτε προσπαθεί να επικοινωνήσει με τη διεύθυνση xx.xx.xx.xx:1111, στην πραγματικότητα επικοινωνεί με τον υπολογιστή του local network που βρίσκεται συνδεδεμένο "πίσω" από το router, με διεύθυνση IP 192.168.1.2.[18][19]

Μεταγωγέας (Switch)

Με τον όρο μεταγωγέα αναφερόμαστε σε μια ηλεκτρονική συσκευή που χρησιμοποιείται σε δίκτυα υπολογιστών. Στην ουσία έχει ορισμένα στοιχεία του επαναλήπτη (Hub) και της γέφυρας (bridge) στο σύνολό του. Αρχικά, οι μεταγωγείς χρησιμοποιήθηκαν σε Ethernet δίκτυα, ενώ αντίθετα στις μέρες μας, υπάρχουν μεταγωγείς και για διαφορετικά πρωτόκολλα όπως για παράδειγμα FDDI, ATM. Πιο ειδικά δίνεται η δυνατότητα ανάπτυξης ταχυτήτων του μεγέθους των Gigabits. Ακόμη, μπορεί να γίνει αντικατάσταση με τους επαναλήπτες Hubs χωρίς να επέλθει καμία επανασχεδίαση της τοπολογίας του δικτύου, προσθέτοντας όμως επιπρόσθετο εύρος ζώνης στους συνδεδεμένους σταθμούς βάσεις.

Στις μέρες μας ο σχεδιασμός τοπικών δικτύων γίνεται με δίκτυα τύπου Ethernet και αποτελούνται κατά κόρον από μεταγωγείς για Ethernet. Αυτό που κάνει το μεταγωγέα να ξεχωρίζει είναι το γεγονός ότι κάθε θύρα του προσδίδει συγκεκριμένο εύρος ζώνης, σε αντίθεση με την πλήμνη, όπου όλες οι συσκευές οι οποίες βρίσκονται συνδεδεμένες σε αυτό διαμοιράζονται το εύρος ζώνης του μέσου. Επιπλέον κάθε θύρα του μεταγωγέα ορίζει μοναδικό πεδίο συγκρούσεων (collision domain).

Ο μεταγωγέας δημιουργεί πίνακες δρομολόγησης με τον ίδιο τρόπο όπως και οι γέφυρες, εφαρμόζοντας τον Spanning tree αλγόριθμο.

Όταν πρόκειται δύο σταθμοί να επιτύχουν επικοινωνία, βρισκόμενοι σε διαφορετικές θύρες του μεταγωγέα (unicast πλαίσιο), ο ίδιος εξετάζει τον πίνακα δρομολόγησης για να εντοπίσει τη διεύθυνση MAC προορισμού και σε ποια πόρτα να μεταβιβάσει το πακέτο. Έτσι αφού εξαχθεί η καταχώρηση θα αποσταλεί το πακέτο στην κατάλληλη θύρα. Έτσι το switch καταφέρνει και μειώνει αισθητά την κίνηση ή τις όποιες συγκρούσεις επρόκειτο να υπάρξουν και αυξάνει την απόδοση του δικτύου, καθώς και το διαθέσιμο εύρος ζώνης των σταθμών βάσεις.

Δυο κύριες λειτουργίες των μεταγωγών είναι οι εξής:

- **Store and forward:** Εξετάζεται όλο το πλαίσιο και σε περίπτωση που εντοπιστεί λάθος στο πεδίο ακολουθίας ελέγχου πλαισίου (Frame Check Sequence, FCS) το πλαίσιο ακυρώνεται.
- **Cut-through:** Εξετάζεται από το πλαίσιο μονάχα η διεύθυνση προορισμού (destination MAC) και προχωρά στην προώθηση του πλαισίου.

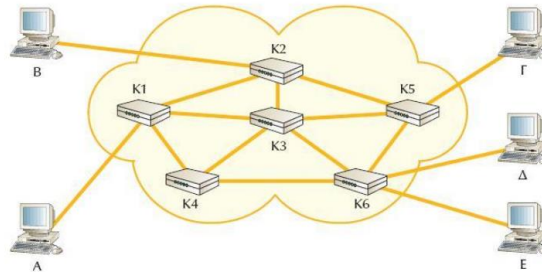
Από τα παραπάνω συμπεραίνουμε, ότι η λειτουργία Cut-through είναι σαφώς πιο γρήγορη από την Store and forward

Οι βασικοί τύποι switches είναι δυο και αναφέρονται σε *διαχειριζόμενα* και *μη διαχειριζόμενα*.

Ένα μη διαχειριζόμενο switch χρησιμοποιείται απευθείας, όπως έχει δημιουργηθεί από τον κατασκευαστή και δεν επιτρέπει καθόλου τροποποίηση. Οι εξοπλισμοί οικιακής δικτύωσης απαρτίζονται τις περισσότερες φορές από μη διαχειριζόμενα switches.



Ένα διαχειριζόμενο switch δίνει την δυνατότητα πρόσβασης και προγραμματισμού. Πράγμα που παρέχει μεγαλύτερη ευελιξία, καθώς ο μεταγωγέας μπορεί να παρακολουθείται και να προσαρμόζεται τοπικά ή απομακρυσμένα επιτρέποντας τον έλεγχο της κυκλοφορίας και της πρόσβασης χρηστών στο δίκτυο.[20]



Εικόνα 1-21 Δίκτυο Μεταγωγής

Πύλη (Gateway)

Σε ένα δίκτυο υπολογιστών, μια πύλη (gateway) είναι ένας κόμβος (δρομολογητής), ο οποίος λειτουργεί ως ένα σημείο πρόσβασης σε άλλο δίκτυο. Πιο συγκεκριμένα έχουν την δυνατότητα να λειτουργήσουν σε αρκετά από τα ανώτερα στρώματα του μοντέλου OSI, κατά κύριο λόγο σε αυτά της συνόδου, παρουσίασης και εφαρμογών[21]. Ακόμη, τις περισσότερες φορές τέτοιου είδους συσκευές μεσολαβούν μεταξύ του τοπικού δικτύου μίας εταιρείας για παράδειγμα και του διαδικτύου.

Χαρακτηριστικά

Όπως αναφέραμε παραπάνω μια πύλη δικτύου (gateway) λειτουργεί ως συνδετικός κρίκος διαφορετικών δικτύων. Η λειτουργία της έχει να κάνει με τον τύπο της σύνδεσης που πρόκειται να επιτευχθεί μεταξύ τους και μπορεί να περιλαμβάνει διάφορες υπηρεσίες όπως

- Μετατροπή format ή και μεγέθους των πακέτων, για παράδειγμα από ASCII σε EBCDIC
- Μετατροπή πρωτοκόλλου (protocol conversion)
- Μετατροπή ή μετάφραση δεδομένων (data conversion ή translation)
- Multiplexing

Επιπλέον ορισμένες πύλες προσφέρουν ειδικές υπηρεσίες, όπως email ή fax. Σε περιβάλλον εταιρίας, μία συσκευή πύλης χρησιμοποιείται επίσης για να φίλτρο για τα δεδομένα τα οποία μεταφέρονται μεταξύ των υπολογιστών της εταιρείας και του διαδικτύου. Με τον τρόπο αυτό τους προστατεύει από τις διάφορες απειλές που ελλοχεύουν κατά την δρομολόγηση των πακέτων, όπως ιούς ή δούρειους ίππους. Συμπερασματικά, η πύλη επιτελεί ρόλο firewall και proxy server.

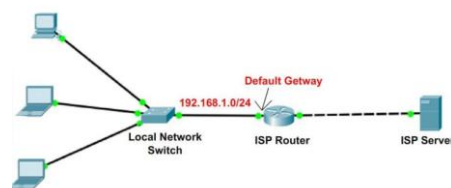
Για να μπορεί να παρέχει αυτές τις υπηρεσίες η πύλη λειτουργεί στα ανώτερα στρώματα του μοντέλου OSI, αναγκασμένη να μεταδώσει σωστά τα πακέτα. Γι' αυτό και δίνει έμφαση στο



περιεχόμενο και τη μορφή τους. Η συσκευή που χρησιμοποιείται ως gateway έχει το δικό της επεξεργαστή και συνήθως μεγάλο αποθηκευτικό χώρο (storage) και αρκετή μνήμη RAM ,έτσι ώστε να κάνει τις απαραίτητες μετατροπές. Για να επιτευχθεί η σύνδεση δικτύων με διαφορετικές αρχιτεκτονικές, μια πύλη χρησιμοποιεί διαφορετική κάρτα δικτύου (NIC) για το καθένα.

Προεπιλεγμένη πύλη (Default gateway)

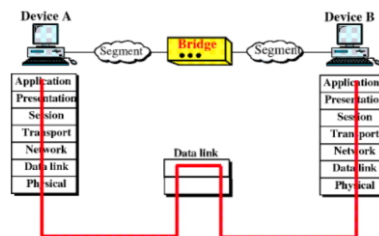
Η προεπιλεγμένη πύλη(default gateway) είναι η διεύθυνση IP της πύλης ή ο δρομολογητής που μετακινεί τα πακέτα μεταξύ διαφορετικών δικτύων. Στην περίπτωση που υπάρχουν περισσότερες από μια πύλες, η προεπιλεγμένη πύλη είναι συνήθως η διεύθυνση της πρώτης ή αυτής που βρίσκονται πιο κοντά. Αντίθετα όταν δεν υπάρχουν πύλες ή δρομολογητές, τότε η προεπιλεγμένη πύλη συνήθίζεται να παίρνει τη διεύθυνση IP του κόμβου του δικτύου.



Εικόνα 1-22 Προεπιλεγμένη πύλη σύνδεσης τοπικού μεταγωγέα με ISP δρομολογητή

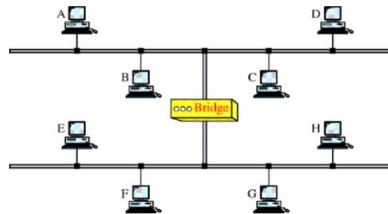
Γέφυρα(Bridge)

Η λειτουργία της γέφυρας είναι στο επίπεδο διασύνδεσης δεδομένων του OSI model. Πιο αναλυτικά είναι η υπεύθυνη για τη σύνδεση υποδικτύων τα οποία χρησιμοποιούν το ίδιο πρωτόκολλο επικοινωνίας. Με άλλα λόγια κατακερματίζουν ένα υπερφορτωμένο δίκτυο σε μικρότερα δίκτυα. Στο OSI μοντέλο η γέφυρα τοποθετείται ενδιάμεσα από δύο συσκευές όπως φαίνεται και πιο κάτω.



Εικόνα 1-23 Τοποθέτηση Γέφυρας (Bridge)

Αντίθετα με τους επαναλήπτες, οι γέφυρες μπορούν να διαχωρίζουν και να μεταβιβάζουν το σήμα μόνο στην πλευρά του παραλήπτη, βοηθώντας με αυτόν τον τρόπο τη ρύθμιση της κυκλοφορίας και την απομόνωση ενδεχόμενων προβλημάτων. Πιο κάτω βλέπουμε τη θέση της γέφυρας μέσα σε ένα δίκτυο.



Εικόνα 1-24 Τοποθέτηση γέφυρας μέσα σε ένα δίκτυο (Bridge)

Όταν τώρα ένα πακέτο φτάσει στη γέφυρα, αυτή την ίδια στιγμή αναλύει τη διεύθυνση προορισμού ενισχύοντας σε πρώτο χρόνο το σήμα. Το καινούργιο αντίγραφο του σήματος μεταδίδεται στην πλευρά όπου βρίσκεται η διεύθυνση προορισμού. Εισάγοντας το πακέτο, αναλύεται η διεύθυνση που υπάρχει μέσα σε αυτό και έπειτα γίνεται σύγκριση με έναν πίνακα που έχει αποθηκευμένες όλες τις διευθύνσεις των τερματικών σε όλη την έκταση του δικτύου. Εφόσον βρεθεί η διεύθυνση προορισμού, τότε η γέφυρα επαληθεύει σε ποιο κομμάτι της δικτυακής τοπολογίας βρίσκεται και στέλνει το πακέτο μόνο εκεί.

Αναλογιζόμενοι με τον τρόπο που έχει υλοποιηθεί ο πίνακας διευθύνσεων, χωρίζουμε τις γέφυρες στις εξής κατηγορίες:

- Απλές γέφυρες (simple bridges)
- Γέφυρες εκμάθησης (learning bridges)
- Γέφυρες πολλαπλών θυρών (multiport bridges)

Απλές γέφυρες (simple bridges)

Η απλή γέφυρα όπως είπαμε και πιο πάνω δύναται για τη διασύνδεση δύο τμημάτων περιλαμβάνοντας έναν πίνακα με τις διευθύνσεις όλων των τερματικών που βρίσκονται σε κάθε σημείο του δικτύου. Σημαντικό μειονέκτημα θεωρείται το γεγονός ότι οι διευθύνσεις πρέπει να εισάγονται χειροκίνητα (*manually*) πριν χρησιμοποιηθεί η γέφυρα. Έτσι καθώς εισάγουμε ένα νέο τμήμα στο δίκτυο, ο πίνακας πρέπει να παραμετροποιείται, ενώ αν αφαιρεθεί κάποιο τμήμα από το δίκτυο, η διεύθυνσή του πρέπει να διαγραφεί από τον πίνακα.

Γέφυρες εκμάθησης (learning bridges)

Οι γέφυρες εκμάθησης διαβάζουν τον πίνακα διευθύνσεων των τμημάτων του δικτύου από μόνες τους. Στην αρχή έχουμε έναν άδειο πίνακα από διευθύνσεις. Καθώς εμφανίζεται ένα πακέτο, η γέφυρα διαβάζει τις διευθύνσεις αποστολέα και παραλήπτη. Αν δεν έχουμε επιτυχή διαδικασία αναγνώρισης, το πακέτο αποστέλλεται σε όλους τους σταθμούς και στις δύο πλευρές. Η διεύθυνση αποστολέα χρησιμοποιείται για να δημιουργήσουμε τον πίνακα. Η γέφυρα αναζητά σε ποιο τμήμα ανήκει η διεύθυνση και τη συνδυάζει με αυτό.

Αφότου γίνει μετάδοση του πρώτου πακέτου από κάθε σταθμό, η γέφυρα γνωρίζει ποιο τμήμα αντιστοιχίζεται με το σταθμό αυτό. Προοδευτικά, η γέφυρα έχει στη διάθεση της αποθηκευμένο έναν πλήρη πίνακα από διευθύνσεις. Ένα τέτοιου είδους γέφυρα ενημερώνεται αυτόματα. Εξαιτίας της αρχιτεκτονικής που πρέπει να ακολουθεί μια τέτοια γέφυρα, το κόστος



της είναι αρκετά υψηλό συγκριτικά με μία απλή γέφυρα. Αναλογιζόμενοι βέβαια το αποτέλεσμα που επιφέρει μια τέτοια γέφυρα το κόστος είναι αμελητέο. Έτσι οι γέφυρες εκμάθησης χρησιμοποιούνται με υψηλά αποτελέσματα σε πολλές εφαρμογές.

Γέφυρες πολλαπλών θυρών (multiport bridges)

Τέτοιες γέφυρες μπορεί να έχουν ρόλο εκμάθησης ή απλής γέφυρας και χρησιμοποιούνται ευρέως για να συνδέσουν δύο και περισσότερα τμήματα του δικτύου.[22]

Επαναλήπτης (Hub)

Ένας επαναλήπτης hub ή αλλιώς συγκεντρωτής είναι μια συσκευή στην οποία συνδέονται δικτυακοί κόμβοι με καλώδια σύστροφου ζεύγους ή οπτικής ίνας ώστε να λειτουργούν ενιαία. Η χρήση τους γίνεται σε LAN ethernet δίκτυα. Τα hubs δρουν στο physical layer του OSI μοντέλου το οποίο θα μελετήσουμε παρακάτω. Η μορφή που έχει είναι σαν ένα επαναλήπτη πολλαπλών θυρών. Τα ethernet hubs αναλαμβάνουν τον εντοπισμό και την προώθηση κάποιου σήματος συμφόρησης σε όλες τις θύρες, αφότου βρεθεί κάποια σύγκρουση.

Στο εσωτερικό ενός hub υλοποιείται η τοπολογία bus. Διαθέτει αρκετές εισόδους/εξόδους και οι συσκευές που συνδέονται πάνω στον επαναλήπτη ουσιαστικά συνδέονται πάνω στο διάλυο του δικτύου ο οποίος υλοποιείται στο εσωτερικό του hub.

Η σύνδεση τους γίνεται από BNC υποδοχή ή ακόμη και AUI (Attachment Unit Interface) έτσι ώστε να μπορούμε να έχουμε και σύνδεση-επικοινωνία και με πιο παλιά μέρη του δικτύου π.χ. 10BASE2. Έχοντας πλέον στη διάθεση μας φτηνούς διακόπτες ethernet έχουμε βάλει στο παρασκήνιο τους επαναλήπτες τους οποίους τους συναντάμε σε πιο παλιές δικτυακές εγκαταστάσεις και σε εξειδικευμένες εφαρμογές.



Εικόνα 1-25 Σύνδεση συσκευών σε ένα συγκεντρωτή

Η λειτουργία του hub επικεντρώνεται στην προώθηση οποιασδήποτε πληροφορίας εμφανίζεται σε κάποια από τις εισόδους του διαμέσων του κοινού διαύλου. Έτσι αυτό έχει ως αποτέλεσμα να βλέπουμε τη συγκεκριμένη πληροφορία και σε όλες τις εξόδους του. Εξαιτίας αυτού παρουσιάζονται πολύ συχνά συγκρούσεις πακέτων συγκριτικά με τις πιο εξελιγμένες συσκευές, όπως είναι οι μεταγωγείς (switches). Σε ένα hub πολλαπλών θυρών, μπορούμε να έχουμε στη μια θύρα ένα τερματικό, σε κάποια άλλη θύρα ένα άλλο τερματικό ή ακόμη και κάποιο άλλο hub. Το σήμα έχοντας ενισχυθεί μέσα στον επαναλήπτη, συνεχίζει τη διαδρομή του

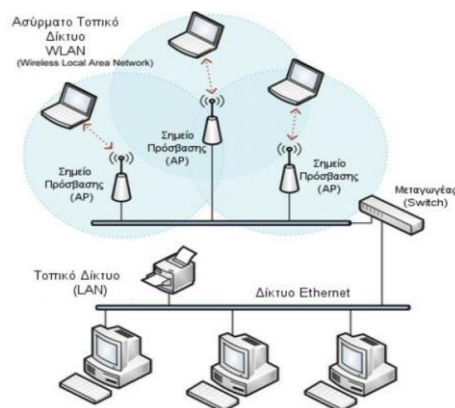


μέσα στο δίκτυο έχοντας τη δυνατότητα να διανύσει μεγαλύτερες αποστάσεις και να ξεπεράσει τους περιορισμούς που βάζει το μέγιστο μήκος του καλωδίου.

Η ανάγκη όμως ύπαρξης τερματικών μηχανημάτων ικανών στην ανίχνευση συγκρούσεων, θέτει ένα κατώφλι στον αριθμό των συγκεντρωτών και στο συνολικό μέγεθος της δικτυακής τοπολογίας που υλοποιείται με συγκεντρωτές. Για την κατασκευή δικτύων ταχύτητας 10Mbps χρησιμοποιώντας συγκεντρωτές, πρέπει να τηρείται ο κανόνας 5-4-3: Με την έννοια 5-4-3 εννοούμε ότι μεταξύ δύο σταθμών μπορούμε να έχουμε έως 5 τμήματα (4 δηλαδή συγκεντρωτές) και μόνο τα 3 από τα 5 να έχουν συνδεδεμένους σταθμούς. Σε δίκτυα ταχύτητας 100Mbps, το κατώφλι πέφτει σε 3 τμήματα (2 δηλαδή συγκεντρωτές) μεταξύ 2 οποιονδήποτε τερματικών σταθμών και αυτό είναι εφικτό με την προϋπόθεση ότι θα χρησιμοποιούμε μεταγωγείς κλάσης II.[23]

Σημείο Πρόσβασης (Access Point)

Με τον όρο *σημείο πρόσβασης (access point)* αναφερόμαστε στην συσκευή που έχει ευθύνη για την λειτουργία της επικοινωνίας με τους ασύρματους σταθμούς. Μία τέτοιου είδους συσκευή μπορεί να είναι ενσύρματα συνδεδεμένη με έναν router (δρομολογητή) ή εσωτερικά μέσα σε ένα δρομολογητή. Επιπλέον μπορεί να έχουμε υλοποίηση χρησιμοποιώντας κάποιο λογισμικό και μια PCI κάρτα σε ένα τερματικό. Το access point αναλαμβάνει τη λειτουργία ενός base station (σταθμού βάσης), συγκεντρώνει δηλαδή την κίνηση που υπάρχει από τους wireless σταθμούς και το διαμοιράζει στο υπόλοιπο δίκτυο. Επίσης αυθεντικοποιεί και αναλύει τα στοιχεία τους νέων σταθμών που κάνουν αίτηση πρόσβασης στο δίκτυο.[24]



Εικόνα 1-26 Σύστημα τριών σημείων πρόσβασης (WLAN) και η διασύνδεση του με LAN δίκτυο

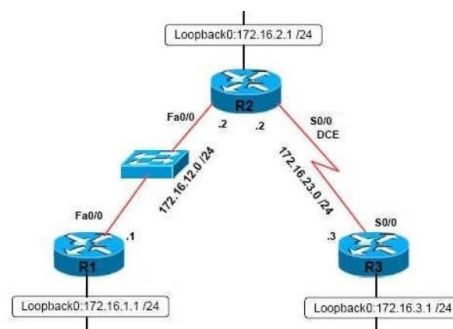
Διεπαφή (Interface)

Με τον όρο interface ή διεπαφή αναφερόμαστε στο όριο μεταξύ του host και της φυσικής ζεύξης (network interface card). Κάθε δρομολογητής έχει αρκετά interfaces αναλογιζόμενοι πάντα



το πλήθος των συνδέσεων που έχει. Κάθε IP διεύθυνση έχει συσχέτιση με μία διεπαφή (NIC) και όχι με ένα *host* ή με κάποιο router που περιέχει την διεπαφή.

Η κάρτα δικτύου (ή αλλιώς *ελεγκτής διασύνδεσης δικτύου, NIC*) αποτελεί ένα κομμάτι του υλικού που αναλαμβάνει τη συνδέση ενός τερματικού σε ένα δίκτυο τερματικών. Αποτελεί *physical layer 1* του OSI (φυσικό επίπεδο) και *layer 2* (επίπεδο ζεύξης δεδομένων), αφού παρέχει πρόσβαση στο φυσικό μέσο δικτύωσης καθώς και ένα σύστημα παροχής διευθύνσεων χαμηλού επιπέδου χρησιμοποιώντας MAC διευθύνσεις. Οι χρήστες μεταξύ τους συνδέονται είτε ενσύρματα είτε ασύρματα. Παλαιότερα οι κάρτες δικτύου χρησιμοποιούνταν ως κάρτες επέκτασης που εισάγονταν σε κάποιο κενό slot ενός υπολογιστή, εξαιτίας του χαμηλού κόστους και της διάδοσης του *Ethernet*. Τα περισσότερα υπολογιστικά μηχανήματα στις μέρες μας έχουν ενσωματωμένη κάρτα δικτύου στη μητρική τους κάρτα.



Εικόνα 1-27 Συνδεσμολογία δρομολογητών με *fastEthernet interface* και *serial interfaces*

Πομποδέκτες (Transceivers)

Με την έννοια πομποδέκτης αναφερόμαστε στο τμήμα της κάρτας δικτύου που είναι υπεύθυνο για τη μεταφορά και τη λήψη του σήματος από και προς το μέσο μετάδοσης. Σε ορισμένες περιπτώσεις (Thick Ethernet) ο transceiver δε βρίσκεται στην κάρτα αλλά στο καλώδιο.[25]

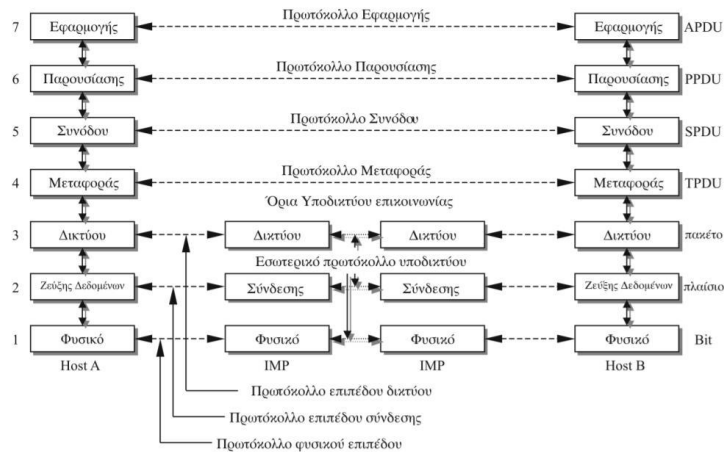
1.4.1 Το μοντέλο αναφοράς OSI

Το μοντέλο αυτό βασίζεται σε μια πρόταση που αναπτύχθηκε από τον Διεθνή Οργανισμό Προτύπων (International Standards Organization ή ISO) ως ένα πρώτο βήμα για τη διεθνή τυποποίηση των πρωτοκόλλων που χρησιμοποιούνται στα διάφορα επίπεδα των δικτύων. Πιο αναλυτικά, ονομάζεται Μοντέλο Αναφοράς ISO OSI (ISO OSI Reference Model), όπου OSI σημαίνει Διασύνδεση Ανοικτών Συστημάτων (Open Systems Interconnection), επειδή ασχολείται με τη διασύνδεση ανοικτών συστημάτων – δηλαδή συστημάτων που είναι ανοικτά στην επικοινωνία με άλλα συστήματα.

Το μοντέλο OSI αποτελείται από επτά επίπεδα. Για να καταλήξουμε στα επίπεδα αυτά χρειάστηκε να εφαρμοστούν ορισμένες αρχές. Αρχικά, όπου χρειάζεται μια διαφορετική λογική αφαίρεση πρέπει να δημιουργείται ένα επίπεδο. Επιπλέον, κάθε επίπεδο πρέπει να εκτελεί μια σαφώς καθορισμένη λειτουργία. Ακόμη, η λειτουργία κάθε επιπέδου πρέπει να επιλέγεται στοχευμένα με βάση τα διεθνή τυποποιημένα πρωτόκολλα. Ακόμη, καλό είναι να αναλογιστεί κανείς ότι τα σύνορα των επιπέδων πρέπει να επιλέγονται έτσι ώστε να ελαχιστοποιείται η ροή



πληροφοριών μέσω της διασύνδεσης των επιπέδων. Τέλος, το πλήθος των επιπέδων πρέπει να είναι αρκετά μεγάλο έτσι ώστε να μην χρειάζεται να ανακατεύονται χωρίς λόγο διαφορετικές λειτουργίες στο ίδιο επίπεδο και ταυτόχρονα αρκετά μικρό έτσι ώστε η αρχιτεκτονική να είναι κατανοητή.



Εικόνα 1-28 Τα επίπεδα του μοντέλου αναφοράς OSI

Αξίζει να σημειωθεί ότι το μοντέλο OSI δεν αποτελεί από μόνο του μια αρχιτεκτονική δικτύου, επειδή δεν προσδιορίζει τις ακριβείς υπηρεσίες και πρωτόκολλα που πρέπει να χρησιμοποιούνται σε κάθε επίπεδο. Το μοντέλο απλώς ορίζει τι οφείλει να κάνει το κάθε επίπεδο. Έχουν δημιουργηθεί όμως και πρότυπα για όλα τα επίπεδα αν και αυτά δεν αποτελούν μέρος του μοντέλου αναφοράς. Το καθένα από τα πρωτόκολλα αυτά έχει δημοσιευθεί ως ξεχωριστό διεθνές πρότυπο. Το μοντέλο χρησιμοποιείται ευρέως ως ένα βαθμό αν και τα συσχετιζόμενα πρωτόκολλα έχουν παραγκωνιστεί με την πάροδο του χρόνου.

Το φυσικό επίπεδο (Επίπεδο 1)

Το επίπεδο αυτό ορίζει τις οποιεσδήποτε ηλεκτρικές και φυσικές προδιαγραφές των συσκευών. Σ' αυτές περιλαμβάνονται οι σχηματισμοί των ακίδων, οι προκαθορισμένες τάσεις, οι απαιτήσεις των καλωδίων. Πιο συγκεκριμένα, οι συσκευές φυσικού επιπέδου είναι οι διανομείς, οι αναμεταδότες, οι κάρτες δικτύου, οι προσαρμοστές αρτηρίας.

Οι κυριότερες λειτουργίες και υπηρεσίες του φυσικού επιπέδου είναι:

- Έναρξη και τερματισμός της ηλεκτρικής σύνδεσης μιας επικοινωνιακής συσκευής.
- Συμμετοχή σε διαδικασίες όπου οι επικοινωνιακές συσκευές εξυπηρετούν με επάρκεια μεγάλο μέρος χρηστών, επιλύοντας με αυτό τον τρόπο προβλήματα προτεραιότητας πρόσβασης και ελέγχου ροής δεδομένων.
- Διαμόρφωση και αποδιαμόρφωση των ψηφιακών δεδομένων όσον αφορά τη μετάδοση από συσκευή σε συσκευή. Για παράδειγμα, τα ψηφιοποιημένα σήματα μπορούν να διαδοθούν ως αναλογικά σε χάλκινο καλώδιο, μετά σε οπτική ίνα, μετά να μεταφερθούν από ραδιοζεύξη ή δορυφορικά, να φθάσουν πάλι αναλογικά σε χάλκινο καλώδιο, και να μετατραπούν σε ψηφιακά στον παραλήπτη.



Οι παράλληλες αρτηρίες SCSI λειτουργούν στο επίπεδο αυτό. Στα επίπεδα 1 και 2 λειτουργούν το Πρωτόκολλο Ethernet, το Token ring, το FDDI (Fiber Distributed Data Interface) πράγμα που σημαίνει διεπαφή κατανεμημένων δεδομένων σε οπτικές ίνες και το πρωτόκολλο IEEE 802.11.

Το επίπεδο συνδέσμου μετάδοσης δεδομένων(Επίπεδο 2)

Το *επίπεδο Ζεύξης Δεδομένων*, είναι υπεύθυνο για την δημιουργία της βασικής συνόδου επικοινωνίας μεταξύ των τερματικών χρησιμοποιώντας το φυσικό μέσο . Με αυτήν την μορφή επικοινωνίας όλα τα επίπεδα πάνω από αυτό μπορούν πλέον να ανταλλάξουν αξιόπιστα δεδομένα με άλλους υπολογιστές. Η δημιουργία της συνόδου επικοινωνίας γίνεται χρησιμοποιώντας ειδικές επικεφαλίδες (headers) και ουρές (tails) προσθέτοντάς τες στα πακέτα (data frames) δημιουργώντας από τα δεδομένα που λαμβάνει από το επίπεδο 3. Η επικεφαλίδα και η ουρά που έχει το πακέτο περιλαμβάνει ορισμένες πληροφορίες για τον παραλήπτη του πακέτου και τον αποστολέα, το μέγεθος των δεδομένων και έναν έλεγχο ορθότητας δεδομένων.

Με βάση αυτές τις πληροφορίες στο πακέτο, το τερματικό το οποίο λαμβάνει μπορεί να εξακριβώσει κάθε φορά αν τα δεδομένα έφτασαν σωστά ή αν υπήρξε κάποια αλλοίωση. Αν η υπηρεσία είναι αξιόπιστη, ο παραλήπτης επιβεβαιώνει την ορθή λήψη κάθε πλαισίου επιστρέφοντας ένα πλαίσιο επιβεβαίωσης (acknowledgment frame). Το πλαίσιο δεδομένων (data frame) που δημιουργείται μεταφέρεται στο επίπεδο 1 για να αποσταλεί στο δίκτυο. Στο επίπεδο αυτό λειτουργούν τα network switches και τα network bridges. Η MAC διεύθυνση των υπολογιστών χρησιμοποιείται για να αναγνωριστούν οι σταθμοί εργασίας. Με βάση αυτήν τη διεύθυνση, το network switch, όπως και το network bridge, μπορεί να επιλέξει την θύρα στην οποία πρέπει να στείλει το πακέτο δεδομένων για να φτάσει στον προορισμό του.

Το επίπεδο δικτύου (Επίπεδο 3)

Το *network layer* είναι υπεύθυνο, να ελέγχει την λειτουργία του υποδικτύου. Ένα από τα βασικά ζητήματα σχεδιασμού είναι ο καθορισμός του τρόπου δρομολόγησης των πακέτων απ' την προέλευση προς τον προορισμό τους. Τα δρομολόγια μπορεί να βασίζονται σε στατιστικούς πίνακες οι οποίοι είναι "προσαρτημένοι" στο δίκτυο και μεταβάλλονται σπάνια, ή συχνότερα μπορεί να προσδιορίζονται στην αρχή της επικοινωνίας, δηλαδή , όταν πραγματοποιείται μια σύνδεση σε κάποιο απομακρυσμένο μηχάνημα. Ακόμη μπορεί να είναι εντελώς δυναμικά, δηλαδή να καθορίζονται εκ νέου για κάθε πακέτο. Ο έλεγχος συμφόρησης αποτελεί επίσης ευθύνη του επιπέδου δικτύου , σε συνδυασμό με τα υψηλότερα επίπεδα που προσαρμόζουν το φορτίο το οποίο τοποθετούν στο δίκτυο.

Γενικότερα η παρεχόμενη ποιότητα υπηρεσιών (καθυστέρηση, χρόνος διέλευσης, παραμόρφωση χρονισμού) είναι κάποια ακόμη ζητήματα του επιπέδου δικτύου. Το πιο συνηθισμένο πρωτόκολλο στο επίπεδο 3 είναι το πρωτόκολλο IP (Internet Protocol). Οι επικεφαλίδες που προστίθενται στο πακέτο περιλαμβάνουν την διεύθυνση IP του παραλήπτη, την διεύθυνση IP του αποστολέα και πληροφορίες που αφορούν τα δεδομένα που περιλαμβάνει το πακέτο.



Το επίπεδο μεταφοράς(Επίπεδο 4)

Το *επίπεδο μεταφοράς (transport layer)* διεκπεραιώνει την μεταφορά των δεδομένων από τερματικό σε τερματικό, απαλλάσσοντας έτσι τα ανώτερα επίπεδα από κάθε προσφορά αξιόπιστης και οικονομικής μεταφοράς δεδομένων. Το επίπεδο μεταφοράς ελέγχει την αξιοπιστία ενός χρησιμοποιούμενου καναλιού με έλεγχο ροής (flow control), τμηματοποίηση (segmentation) και αποτμηματοποίηση (desegmentation), και έλεγχο σφαλμάτων (error control). Ορισμένα πρωτόκολλα καταγράφουν καταστάσεις και συνδέσεις, οπότε καταγράφουν την αποστολή και λήψη των πακέτων και προχωρούν σε επανεκπομπή όσων δεν παραλήφθηκαν σωστά. Τα διάφορα πρωτόκολλα σχηματίζουν διαφορετικά τα πακέτα πληροφοριών.

Το καλύτερο παράδειγμα πρωτοκόλλου μεταφοράς είναι το TCP (Transmission Control Protocol, πρωτόκολλο ελέγχου μετάδοσης). Άλλα πρωτόκολλα μεταφοράς είναι τα UDP (User Datagram Protocol, πρωτόκολλο για γράμμα δεδομένων από έναν χρήστη), SCTP (Stream Control Transmission Protocol, πρωτόκολλο ελέγχου της ροής μετάδοσης).

Το επίπεδο συνδιάλεξης (Επίπεδο 5)

Το *επίπεδο περιόδου σύνδεσης (session layer)* αλλιώς επιτρέπει σε χρήστες διαφορετικών μηχανημάτων να εγκαθιδρύουν συνδιαλέξεις (sessions) μεταξύ τους. Οι συνδιαλέξεις προσφέρουν διάφορες υπηρεσίες, στις οποίες περιλαμβάνονται ο έλεγχος διαλόγου(dialog control), η παρακολούθηση του ποιος έχει σειρά να μεταδώσει), η διαχείριση της σκυτάλης (token management, η αποτροπή των δύο πλευρών από το να επιχειρήσουν ταυτόχρονα την εκτέλεση της ίδιας κρίσιμης λειτουργίας) και ο συγχρονισμός (synchronization, η τήρηση σημείων ελέγχου σε μακρόχρονες μεταδόσεις, έτσι ώστε αυτές να μπορούν να συνεχιστούν από το σημείο όπου διακόπηκαν μετά από μια κατάρρευση (crash) του συστήματος).

Το επίπεδο παρουσίασης(Επίπεδο 6)

Το *presentation layer* τροποποιεί τα δεδομένα σε τυπική μορφή που την αναμένει το επίπεδο εφαρμογών. Στο επίπεδο αυτό πραγματοποιείται στα δεδομένα κρυπτογράφηση, συμπίεση, κωδικοποίηση, ή οποιαδήποτε άλλη διαμόρφωση απαιτεί η μορφή δεδομένων ή ο σχεδιαστής του πρωτοκόλλου.

Το επίπεδο εφαρμογών(Επίπεδο 7)

Το *application layer* αλλιώς, περιλαμβάνει μια ποικιλία πρωτοκόλλων που απαιτούνται συχνά από τους χρήστες. Πρόκειται για αυτό που καθορίζει τα μέρη της επικοινωνίας, τη ποιότητα της υπηρεσίας, η αυθεντικότητα του χρήστη και τον όποιο περιορισμό χρειαστεί η σύνταξη των δεδομένων. Ένα ευρέως διαδεδομένο πρωτόκολλο εφαρμογής είναι το Πρωτόκολλο Μεταφοράς Υπερ-κειμένου ή HTTP (Hyper-Text Transfer Protocol), το οποίο είναι η βάση του Παγκόσμιου Ιστού. Άλλα πρωτόκολλα εφαρμογών χρησιμοποιούνται για τη μεταφορά αρχείων, το ηλεκτρονικό ταχυδρομείο και τις ομάδες ειδήσεων δικτύου.[12]





2 Διευθυνσιοδότηση Internet Protocol έκδοση 6 (IPv6)

2.1 Πρωτόκολλο IP

Το Πρωτόκολλο Διαδικτύου (Internet Protocol, IP), είναι το βασικότερο πρωτόκολλο επικοινωνίας για τη μετάδοση δεδομενογραμμμάτων (datagrams), δηλαδή πακέτων δεδομένων, σε ένα διαδίκτυο. Το Πρωτόκολλο IP ασχολείται με τη δρομολόγηση των πακέτων δεδομένων μεταξύ διαφόρων δικτύων χωρίς να δοθεί ιδιαίτερη σημασία στην υποδομή τους, αποτελώντας το πρωτόκολλο σταθμό που είναι στηριγμένο όλο το Διαδίκτυο.

Το Internet Protocol βρίσκεται στο επίπεδο Δικτύου στο TCP/IP μοντέλο. Προσδιορίζει τη μορφή που θα έχουν τα πακέτα που αποστέλλονται στο διαδίκτυο, καθώς και τους μηχανισμούς που χρησιμοποιούνται για την προώθηση των πακέτων από ένα τερματικό προς έναν προορισμό διαμέσων ενός ή περισσότερων δρομολογητών. Γ' αυτές τις επιδιώξεις, το IP, κάνει χρήση κάποιων μεθόδων διευθυνσιοδότησης και δομές για την ενθυλάκωση των πακέτων δεδομένων. Η πρώτη ευρέως διαδεδομένη έκδοση του IP, ήταν η έκδοση 4 (IPv4) η οποία χρησιμοποιείται κατά κόρον στις μέρες μας σε όλο το Διαδίκτυο. Βέβαια οι διευθύνσεις πλέον εξαντλούνται, και έτσι πιο πρόσφατα, έχει σχεδιαστεί και υλοποιηθεί η έκδοση που θα πάρει τη θέση του IPv4, η έκδοση 6 (IPv6), η οποία χρησιμοποιείται παγκοσμίως έχοντας υπό κατασκευή διάφορα νέα χαρακτηριστικά.

Η λειτουργία του IP όπως είπαμε και πιο πάνω, είναι υπεύθυνη για τη διευθυνσιοδότηση των κόμβων και την δρομολόγηση των πακέτων από έναν υπολογιστή προς έναν τελικό προορισμό, στην έκταση ενός ή περισσότερων δικτύων. Για το λόγο αυτό, το IP έχει ένα σύστημα διευθυνσιοδότησης δύο λειτουργιών. Συνεπώς το κάθε πακέτο IP, έχει μια κεφαλίδα (header) και μετά ακολουθούν τα δεδομένα. Η κεφαλίδα αυτή περιέχει πληροφορίες που αφορούν τα δεδομένα του πακέτου και τις διευθύνσεις αποστολέα και παραλήπτη. Η διαδικασία πρόσθεσης της κεφαλίδας σε ένα πακέτο λέγεται ενθυλάκωση. Το IP είναι μια υπηρεσία που λειτουργεί χωρίς σύνδεση, δεν εξαρτάται από το υλικό που χρησιμοποιεί το εκάστοτε δίκτυο και είναι απαραίτητο να το γνωρίζει πριν γίνει η μετάδοση. [12]

2.1.1 TCP/IP-UDP

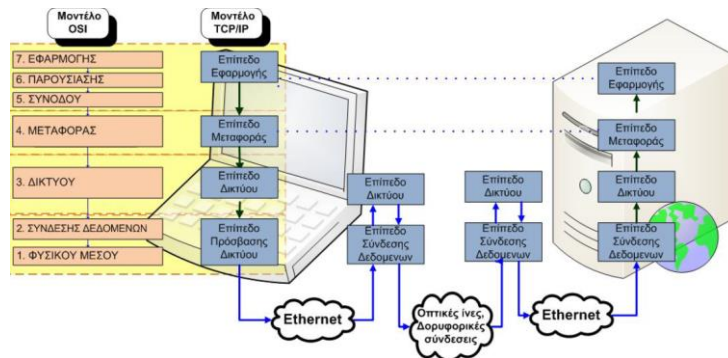
Πρωτόκολλα με ή χωρίς εγκατάσταση σύνδεσης

Οι δικτυακές εφαρμογές, οι οποίες λειτουργούν στους κόμβους ενός δικτύου, σε τερματικά, σε έξυπνες φορητές συσκευές κ.α., επικοινωνούν ανταλλάσσοντας πακέτα δεδομένων. Όπως προαναφέρθηκε, το επίπεδο μεταφοράς είναι υπεύθυνο για την αποστολή και λήψη των δεδομένων τα οποία προέρχονται από το επίπεδο εφαρμογής μεταξύ του τερματικού (κόμβου) αφετηρίας και του τερματικού (κόμβου) προορισμού. Με άλλα λόγια την επικοινωνία από-άκρο-σε-άκρο (*end-to-end*), προσανατολισμένα στην ύπαρξη σύνδεσης ή όχι. Διαφορετικά στην πρώτη περίπτωση, αρχικά επιτυγχάνεται εγκατάσταση σύνδεσης και ένα πρόγραμμα του τερματικού αφετηρίας συνομιλεί με ένα παραπλήσιο πρόγραμμα του τερματικού προορισμού, σε αντίθεση με την δεύτερη περίπτωση όπου χωρίς να εγκατασταθεί σύνδεση μεταξύ των κόμβων, το πρόγραμμα στην αφετηρία μεταδίδει αμέσως τα δεδομένα στο πρόγραμμα προορισμού. Όσον



αφορά το γεγονός της επιτυχής εγκατάστασης της σύνδεσης, οι πληροφορίες της εγκατεστημένης σύνδεσης αποθηκεύονται στις επικεφαλίδες του πακέτου και στα μηνύματα ελέγχου.

Η οικογένεια πρωτοκόλλων TCP/IP διαθέτει στο επίπεδο μεταφοράς τα πρωτόκολλα TCP και UDP που πραγματοποιούν τις διαδικασίες μεταφοράς των μηνυμάτων δεδομένων.



Εικόνα 2-1 OSI TCP διαστρωμάτωση

Τα πρωτόκολλα αυτά διαχωρίζονται μεταξύ τους: στο TCP που είναι πρωτόκολλο προσανατολισμένο σε σύνδεση (Connection oriented) και UDP που είναι πρωτόκολλο χωρίς σύνδεση (Connectionless).

Πρωτόκολλο προσανατολισμένο στη σύνδεση είναι αυτό που αρχικά, πριν ξεκινήσει η μετάδοση των δεδομένων εγκαθιστά μια σύνδεση από άκρο σε άκρο για να εξασφαλιστεί μια διαδρομή (υποθετικό κύκλωμα) για τη μετάδοση των πακέτων. Όλα τα πακέτα μεταφέρονται στο ίδιο υποθετικό κύκλωμα. Την στιγμή που ξεκινήσει η μετάδοση εξασφαλίζεται ότι τα δεδομένα θα φτάσουν στον παραλήπτη χωρίς σφάλματα.

Πρωτόκολλο χωρίς σύνδεση είναι αυτό στο οποίο ξεκινά η μετάδοση των δεδομένων δίχως να έχει προηγηθεί επικοινωνία με τον παραλήπτη. Τα δεδομένα μεταδίδονται σε αυτοδύναμα πακέτα (datagrams) χωρίς την εγκατάσταση σύνδεσης μέσω υποθετικών κυκλωμάτων. Τα πρωτόκολλα αυτά θεωρούνται αναξιόπιστα, διότι δεν εξασφαλίζουν ότι τα δεδομένα θα φτάσουν στο προορισμό τους. Η πληροφορία που μεταδίδεται από άκρο σε άκρο στο επίπεδο μεταφοράς οργανώνεται σε ακολουθία από ομαδοποιημένα δεδομένα που ονομάζονται datagrams. Κάθε ένα datagram μετράτε σε οκτάδες ψηφίων (byte) και αντιμετωπίζεται απολύτως ανεξάρτητα από το δίκτυο.

2.1.1.1 Πρωτόκολλο TCP - Δομή πακέτου

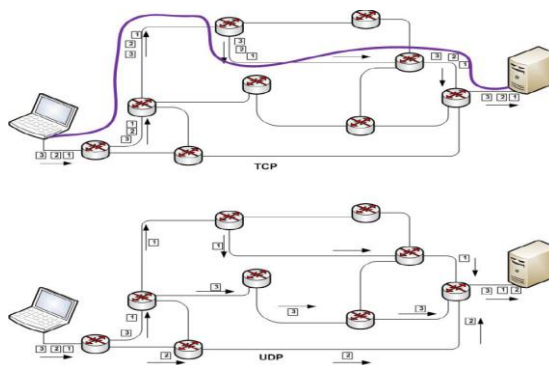
Για να κατανοηθεί η λειτουργία του πρωτοκόλλου TCP ας δούμε ένα παράδειγμα:

Έστω ότι θέλουμε να αποστείλουμε ένα μήνυμα μέσω ηλεκτρονικού ταχυδρομείου. Αρχικά η εφαρμογή χρησιμοποιώντας τα πρωτόκολλα του επιπέδου εφαρμογής παράγει μια σειρά πληροφοριών υπό μορφή δεδομένων με τις εντολές και το περιεχόμενο που ανταλλάσσουν δυο κόμβοι μέσω του δικτύου. Προϋπόθεση είναι η αξιόπιστη μετάδοση των πληροφοριών μέσω του δικτύου. Η πληροφορία συμπεριλαμβάνεται στο επίπεδο μεταφοράς από το πρωτόκολλο TCP που με την σειρά του αναλαμβάνει να μεταφέρει τα δεδομένα από το ένα άκρο στο άλλο.



Έστω ότι στο παραπάνω παράδειγμα το TCP παραλαμβάνει από την εφαρμογή ηλεκτρονικού ταχυδρομείου δεδομένα μεγέθους 6000 octets. Ελέγχει το δίκτυο και διαπιστώνει ότι δεν μπορεί να διαχειριστεί *datagram* μεγαλύτερα από ένα συγκεκριμένο μέγεθος. Στην πραγματικότητα τα δύο άκρα δηλώνουν το μεγαλύτερο μέγεθος *datagram* που μπορούν να διαχειριστούν. Για να αντιμετωπιστεί η κατάσταση το αρχικό *datagram* διασπάται σε μικρότερα ίσα κομμάτια, τα οποία αποστέλλονται ανεξάρτητα από το ένα άκρο στο άλλο. Τα μικρότερα αυτά *datagrams* συμφωνημένου μεγέθους ονομάζονται Τμήματα (*segments*). Επομένως στο πρωτόκολλο TCP η μονάδα δεδομένων που διαχειρίζεται (*PDU*) αναφέρεται ως Τμήμα (*segment*). Βέβαια στο Τμήμα μεταξύ των δύο άκρων μπορεί να χωρά ολόκληρο το *datagram*, οπότε δεν θα χρειαστεί να διασπαστεί.

Στο TCP/IP πρωτόκολλο, θεωρείται ότι υπάρχει ένας αρκετά μεγάλος αριθμός ανεξάρτητων δικτύων που διασυνδέονται με εξωτερικές πύλες δρομολόγησης (*gateways*). Τα τμήματα μεταφέρονται από αρκετά διαφορετικά δίκτυα πριν φτάσουν στο προορισμό τους. Σε πολλές περιπτώσεις το μονοπάτι είναι διαφορετικό για κάθε τμήμα και η διαδρομή δεν είναι γίνεται γνωστή στο χρήστη. Όταν φτάσουν στο άλλο άκρο θα επανασυνδεθούν για να διαμορφώσουν το αρχικό μήνυμα των 6000 octets. Όμως τα ανεξάρτητα τμήματα είναι πολύ πιθανόν να φτάσουν με διαφορετική σειρά, για παράδειγμα το έβδομο τμήμα να φτάσει πριν το τρίτο. Ακόμη λόγω σφάλματος δικτύου σε κάποιο σημείο της διαδρομής υπάρχει το ενδεχόμενο κάποιο τμήμα να καταστραφεί. Τότε θα χρειαστεί να σταλεί ξανά το συγκεκριμένο τμήμα.



Εικόνα 2-2 Επικοινωνία με TCP και UDP πρωτόκολλα

Το TCP στην φάση της επανασύνδεσης του αρχικού μηνύματος πρέπει να γνωρίζει ποια είναι η προέλευση (*source*) του μηνύματος και ποιος ο προορισμός (*destination*).

Έτσι το TCP εξασφαλίζει την Αξιοπιστία της σύνδεσης με:

- Την Εγκαθίδρυση Σύνδεσης από την προέλευση στον προορισμό.
- Τεμαχίζει τα δεδομένα αν χρειαστεί από το δίκτυο.
- Επιβεβαιώνει την παραλαβή δεδομένων.
- Τοποθετεί στη σειρά τα τμήματα κατά την παραλαβή



Όλες αυτές οι πληροφορίες οι οποίες χρειάζονται για να ελεγχθεί και να ανασυντεθεί το αρχικό μήνυμα περιέχονται στην επικεφαλίδα (*header*) που παράγεται κατά τον αρχικό σχηματισμό του τμήματος.

Η επικεφαλίδα είναι ένα σύνολο δεδομένων πριν από τα γνήσια δεδομένα και προστίθεται στην αρχή του τμήματος. Η επικεφαλίδα έχει ελάχιστο μήκος 20 octets και μέγιστο 60 octets μαζί με το μη αναγκαίο πεδίο *options*. Οι πληροφορίες που εισάγει το TCP στην επικεφαλίδα ώστε να πετύχει την αξιοπιστία της μεταφοράς του μηνύματος είναι:

- Ο Αριθμός Θύρας Προέλευσης (*source port number*) και Αριθμός Θύρας Προορισμού (*destination port number*). Οι αριθμοί θύρας χρησιμεύουν στην αναγνώριση των ποικίλων συνομιλιών μεταξύ των δύο πλευρών. Έστω ότι δυο διαφορετικοί άνθρωποι στέλνουν από ένα μήνυμα ηλεκτρονικού ταχυδρομείου προς ένα τρίτο. Το TCP αποδίδει τις θύρες με αριθμούς 100 και 200 στις διεργασίες των υπηρεσιών ηλεκτρονικού ταχυδρομείου των αποστολέων αντίστοιχα και τη θύρα 25 με την υπηρεσία που θα παραδοθεί το μήνυμα στον ηλεκτρονικό υπολογιστή του παραλήπτη στο άλλο άκρο. Όταν αποστέλνεται ένα τμήμα στο *header* των δύο *segments*, τα νούμερα 1024 και 2024 αναφέρονται στις πόρτες από τις οποίες προήλθαν. Αναμφίβολα, το πρωτόκολλο οφείλει να γνωρίζει ποια είναι η πόρτα προορισμού στην άλλη πλευρά και για αυτό το λόγο προσθέτει τον αριθμό 25 στην επικεφαλίδα στο ίδιο πεδίο (του προορισμού). Εν τέλει, στην περίπτωση που το άλλο άκρο χρειαστεί να στείλει πίσω ένα τμήμα τότε τα πεδία της θύρας προέλευσης και προορισμού πρέπει να τροποποιηθούν σύμφωνα με την *header* του αντίστοιχου *segment*.
- Ο Αριθμός Σειράς (*Sequence Number*). Ο αριθμός αυτός έχει λόγω ύπαρξης για την στιγμή που ο παραλήπτης στο άλλο άκρο προσπαθεί να τοποθετήσει τα τμήματα στη σωστή σειρά καθώς ανασυνθέτει το αρχικό τμήμα. Αυτό συμβαίνει διότι η σειρά με την οποία έχει ληφθεί το πακέτο ενδέχεται να είναι μην είναι η ίδια με τη σειρά που έχει σταλεί. Το πρωτόκολλο αριθμεί τα τμήματα με βάση τα octets, γι' αυτό και αν το οποιοδήποτε *segment* αποτελείται από 700 octets, τότε ο αριθμός σειράς στην επικεφαλίδα του πρώτου τμήματος θα έχει τον αριθμό 0, στον δεύτερο 700, στον τρίτο 1400 κ.ο.κ.
- Ο Αριθμός Επιβεβαίωσης (*Acknowledgment*), χρησιμοποιείται για να εξασφαλιστεί ότι κάθε τμήμα έχει φτάσει στον προορισμό του. Όταν ο παραλήπτης στο άλλο άκρο λάβει το τμήμα στέλνει ένα νέο τμήμα (ACK- επιβεβαίωσης) του οποίου το πεδίο Αριθμός επιβεβαίωσης, είναι συμπληρωμένο. Για παράδειγμα, στέλνοντας ένα τμήμα με επιβεβαίωση τον αριθμό 1201, σημαίνει ότι έχουν φτάσει όλα τα δεδομένα μέχρι και το octet με αριθμό 1200. Αν δεν γίνει λήψη της επιβεβαίωσης μέσα σε ένα καθορισμένο χρονικό διάστημα, στέλνονται πάλι τα δεδομένα.
- Το Μέγεθος Παραθύρου (*Window*). Κυρίως για λόγους επιτάχυνσης της επικοινωνίας το πρωτόκολλο δεν αναμένει την λήψη της επιβεβαίωσης για να στείλει το επόμενο τμήμα. Δεν γίνεται όμως να στέλνονται ακατάπανστα δεδομένα διότι ένας ιδιαίτερα γρήγορος αποστολέας στο ένα άκρο θα μπορούσε να ξεπεράσει τις δυνατότητες απορρόφησης δεδομένων από ένα αργό παραλήπτη. Έτσι με το πεδίο *Window* κάθε πλευρά αναφέρει



πόσα νέα δεδομένα μπορεί να δεχθεί βάζοντας σ' αυτό το πεδίο τον αριθμό από octets που έχει ελεύθερα ο buffer. Όμως το μέγεθος του προσωρινού χώρου που μένει ελεύθερο μειώνεται όσο το τερματικό λαμβάνει δεδομένα ανάλογα με τις απαιτήσεις επεξεργασίας του παραλήπτη. Αν ο χώρος αυτός γεμίσει πρέπει ο αποστολέας να παύσει την αποστολή νέων δεδομένων, καθώς σ' αυτή την περίπτωση τα δεδομένα θα απορριφθούν. Όταν ο παραλήπτης απελευθερώσει χώρο δηλώνει με το πεδίο Window ότι είναι έτοιμος να δεχτεί νέα δεδομένα.

- Το Άθροισμα Ελέγχου (*Checksum*). Ο αριθμός αυτός στο πεδίο της επικεφαλίδας τοποθετείται από τον αποστολέα μιας και υπολογίζει το άθροισμα απ' όλα τα octets σε ένα datagram. Το πρωτόκολλο στο άλλο άκρο υπολογίζει πάλι το άθροισμα και το συγκρίνει με αυτό έλαβε. Σε περίπτωση που τα δύο αποτελέσματα είναι διαφορετικά, τότε κάτι έγινε κατά τη μεταφορά και το datagram δεν γίνεται δεκτό.
- Τα πεδία Σημαίες Ελέγχου (*Flags*) είναι ωφέλιμα για τον αντιμετώπιση των συνδέσεων και αντιστοιχούν σε 9 bit όπου τα κυριότερα από αυτά είναι:
 1. URG (*Urgent Pointer*). Το πεδίο URG δίνει την έγκριση στη μια πλευρά να γνωστοποιήσει στην άλλη κάτι σημαντικό, όπως για παράδειγμα να προχωρήσει στην επεξεργασία ενός συγκεκριμένου octet, τη διακοπή της εξόδου με την πληκτρολόγηση κάποιου χαρακτήρα ελέγχου (*control character*) κ.α.
 2. ACK (*Acknowledgment*). Το πεδίο αυτό ανακοινώνει ότι ο κόμβος που στέλνει το bit με τιμή 1 (On) επικυρώνει τη λήψη δεδομένων.
 3. PSH (*Push*). Το πεδίο αυτό ενημερώνει τον παραλήπτη ότι χρειάζεται όσο ταχύτερα γίνεται προώθηση της πληροφορίας στο επίπεδο εφαρμογής.
 4. RST (*Reset*). Το πεδίο αυτό εφιστεί την προσοχή για να επιτευχθεί καθαρισμός της σύνδεσης.
 5. SYN (*Synchronize*). Το πεδίο αυτό χρησιμοποιείται για το συγχρονισμό της εγκατάστασης μιας νέας σύνδεσης κάνοντας χρήση του πεδίου Αριθμός Σειράς έτσι ώστε να γίνει εκκίνηση μίας σύνδεσης.
 6. FIN (*Finalize*). Το πεδίο πληροφορεί ότι ο αποστολέας έχει ολοκληρώσει την μεταφορά δεδομένων.

Συμπερασματικά, η δομή του πακέτου του πρωτοκόλλου TCP περιέχει όλες πληροφορίες που απαιτούνται σε μια επικοινωνία που παρέχει υπηρεσίες με σύνδεση και αφορούν τα εξής:

- Την Εγκαθίδρυση σύνδεσης με προκαθορισμένες απαιτήσεις όσον αφορά την επικοινωνία μεταξύ των δυο άκρων
- Την Αξιοπιστία στην μετάδοση των δεδομένων. Απώλεια δεδομένων μετά τον έλεγχο σφαλμάτων απαιτεί αναμετάδοση.



- Τον Έλεγχο ροής δεδομένων ,έτσι ώστε να μην κατακλυστεί ο παραλήπτης με δεδομένα από το αποστολέα.
- Τον Έλεγχο Συμφόρησης δεδομένων ,ώστε να μην κατακλυστεί ένα αργός διάυλος επικοινωνίας με δεδομένα με κίνδυνο υποχώρησης. [12][26]

2.1.1.2 Πρωτόκολλο UDP - Δομή πακέτου

Το πρωτόκολλο User Datagram Protocol είναι ένα εν μέρει πιο απλό πρωτόκολλο σε σχέση με το TCP που χρησιμοποιείται στο επίπεδο μεταφοράς.

Για την μεταφορά των datagrams δεν πραγματοποιείται εγκατάσταση σύνδεσης μεταξύ των δύο άκρων και δεν χωρίζεται το μήνυμα σε μικρότερα τμήματα(*segments*) όταν δεν υποστηρίζεται το μέγεθος του datagram. Κάθε αυτοδύναμο πακέτο διαδίδεται μέσω δικτύων από κόμβο σε κόμβο μέχρι να καταλήξει στη τελική διεύθυνση δίχως να εγγυάται κανείς ότι δεν θα χαθεί ή θα καταστραφεί. Αντίθετα όμως αυτή η απλότητα της δομής του και η ανεπάρκεια ελέγχων προσδίδει στο UDP το πλεονέκτημα της αυξημένης ταχύτητας μετάδοσης των πληροφοριών και την απώλεια σε overhead ,δηλαδή της αισθητής ελαχιστοποίησης στη χρήση των πόρων του δικτύου για ανωφελείς εργασίες.

- Ο αριθμός Θύρας Προέλευσης και ο αριθμός Θύρας Προορισμού. (Source Port & Destination Port)
- Το μήκος του datagram (Length). Το ελάχιστο μήκος είναι 8 octets δηλαδή μόνο η επικεφαλίδα, και το μέγιστο μέγεθος φτάνει τα 64534 octets (64Kb) μαζί με την επικεφαλίδα.
- Το Άθροισμα Ελέγχου (Checksum). Είναι προαιρετικό πεδίο των 16-bit το οποίο χρησιμοποιείται για επαλήθευση της εγκυρότητας του datagram κατά την λήψη του στην πλευρά του παραλήπτη. Ακόμη κάνει τον υπολογισμό του αθροίσματος της κεφαλίδας και των δεδομένων και η ενέργειά του είναι παρόμοια με του TCP.

Επομένως όπως έχει ήδη περιγραφεί το TCP είναι κατάλληλο για εφαρμογές που απαιτούν την αξιόπιστη μεταφορά των δεδομένων. Αντίθετα το UDP χρησιμοποιείται σε εφαρμογές όπου δεν έχει τόση σημασία η πληρότητα της μεταφοράς των δεδομένων σε σύγκριση με την ταχύτητα που θα παραληφθούν.

Τέτοιες εφαρμογές είναι:

- Ορισμένες οι οποίες μεταδίδουν σε πραγματικό χρόνο stream video και ήχου (*real-time audio/video*), όπως IPTV, VoIP. Στην συγκεκριμένη περίπτωση μας ενδιαφέρει τα δεδομένα να φτάνουν τη προκαθορισμένη χρονική στιγμή. Οποιαδήποτε απώλεια τους έχει επίπτωση μόνο σε ποιοτικό βαθμό όσο αφορά το αναπαραγόμενο σήμα.
- Servers, οι οποίοι λειτουργούν για μικρά αιτήματα ενός τεράστιου αριθμού από πελάτες/clients, όπως στα δικτυακά online παιχνίδια. Οι Servers, χρησιμοποιώντας UDP, δεν ασχολούνται με το να ελέγχουν την κατάσταση της κάθε σύνδεσης και έτσι καταφέρνουν να εξυπηρετήσουν έναν ιδιαίτερα σημαντικό αριθμό χρηστών σε αντίθεση με το αν έκαναν χρήση TCP πρωτοκόλλου.



- Ωστόσο, αν χρειαστεί να επιλυθούν ζητήματα, όπως θέματα αξιοπιστίας, ελέγχου ροής, τμηματοποίησης των πακέτων, τότε αναλαμβάνει το επίπεδο εφαρμογής. Επίσης πρέπει να σημειωθεί το πρόβλημα δικτυακής συμφόρησης που πρέπει να αναλάβει το επίπεδο εφαρμογής στην περίπτωση κατά την οποία ένας αποστολέας UDP κατακλύσει το δίκτυο με πακέτα. Επίσης είναι αναγκαίο οι συσκευές του ενδιάμεσου δικτύου (δρομολογητές) να χρησιμοποιούν τεχνικές ελέγχου, που αποθηκεύουν προσωρινά.[12][26]

2.2 Διευθυνσιοδότηση IPV6

Το IPv4 πρωτόκολλο από τις αρχές του 1980 ανταποκρίθηκε επιφέροντας πολύ καλά αποτελέσματα στον παγκόσμιο ιστό. Βέβαια με τον αυξανόμενο ρυθμό των συνδεδεμένων χρηστών το πρωτόκολλο είχε φτάσει σε ένα σημείο κορεσμού με αποτέλεσμα να εξαντλούνται και οι εναπομείνουσες διευθύνσεις. Συνεπώς υπήρχε επιτακτική ανάγκη για την ανεύρεση νέων διευθύνσεων, έχοντας ως αποτέλεσμα τη δημιουργία της IPv6 έκδοσης η οποία έδωσε σωτήριες λύσεις σε βασικά προβλήματα που είχε το IPv4.

Για να γίνει κατανοητή η υπεροχή που έχει η έκδοση IPv6 συγκριτικά με την έκδοση IPv4, στο κομμάτι που αφορά τις διευθύνσεις, θα υπάρχουν 2^{128} ξεχωριστές διευθύνσεις ή πιο συγκεκριμένα *340.282.366.920.938.463.463.374.607.431.768.211.456* διευθύνσεις.

2.2.1 Μορφή IPV6 διευθυνσιοδότησης

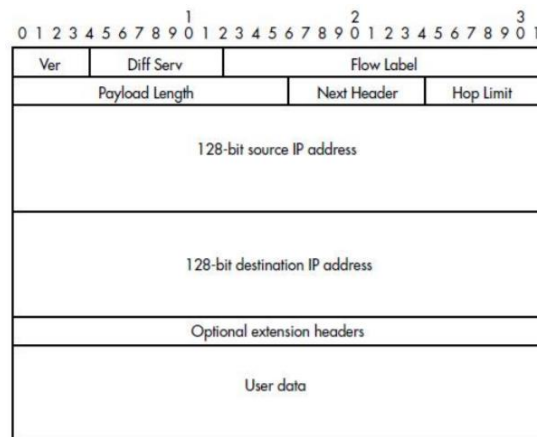
Οι διευθύνσεις που ακολουθούν την έκδοση IPv6 γράφονται δεκαεξαδική μορφή και συμβολίζονται από οκτάδες που χωρίζονται από άνω και κάτω τελεία. Το ακριβές μέγεθος μιας διεύθυνσης v6 είναι 128 Bits και ένα παράδειγμα δίνεται παρακάτω.

FE80:0000:0000:0000:0202:B3FF:FE1E:8329

Για να έχουμε μια πιο ξεκάθαρη όψη της διεύθυνσης που θα μας βοηθά και στην μελέτη της μπορούμε να κάνουμε δύο ενέργειες

- Μπορεί να γίνει παράληψη των τελευταίων μηδενικών, έτσι η διεύθυνση γίνεται της μορφής: **FE80:0:0:0:202: B3FF: FE1E: 8329**
- Ακόμη για να βελτιώσουμε μορφολογικά τη διεύθυνση αντικαθιστούμε τα μηδέν με δύο φορές άνω και κάτω τελεία, και έχουμε το παρακάτω αποτέλεσμα **FE80::202:B3FF: FE1E: 8329**

Σε αυτό το σημείο πρέπει να τονίσουμε πως η *διπλή στήλη(:)* αντικαθιστά οποιονδήποτε αριθμό έχει ακολουθία με συνεχόμενα μηδέν, παρόλα αυτά μια διεύθυνση συμπεριλαμβάνει μονάχα μια διπλή στήλη. Ακολουθεί το ίδιο μοτίβο με την έκδοση v4 όπου ορισμένα bits προς τα αριστερά προσδιορίζουν τη μάσκα. Αυτό το κομμάτι της διεύθυνσης IP version 6 ονομάζεται



Εικόνα 2-3 Δομή ενός πακέτου IPv6

Παρατηρούμε ότι ορισμένα fields όπως αυτά της κεφαλίδας, του *header checksum* και του *fragment offset* λείπουν. Επιπλέον έχουμε νέα fields τα οποία αποκαλούνται *Hop Limit* και *Differentiated Services* στη θέση των *Time to Live* και *Type of Service* αντίστοιχα. Ακόμη βλέπουμε η επικεφαλίδα ενός πακέτου IPv6 έχει 9 fields, με τα 8 από αυτά να είναι υποχρεωτικά και το 9^ο να μην είναι. Στην version 6 κάθε field έχει προδιαγεγραμμένο μέγεθος.

Στη συνέχεια θα αναλύσουμε κάθε field του IPV6 ξεχωριστά για να κατανοήσουμε καλύτερα τις έννοιες του πακέτου.

- *Version*: Το field αυτό αποτελείται από 4 bits και καθορίζει με ποια έκδοση υλοποιήθηκε το πακέτο δεδομένων στην προκειμένη περίπτωση έχει την τιμή 6.
- *Differentiated Services*: Το field απαρτίζεται από 8 bits και η χρήση του επικεντρώνεται στη διεύθυνση της κυκλοφορίας σαν κομμάτι του συστήματος της ποιότητας των υπηρεσιών(QoS).
- *Flow Label*: Αυτό το πεδίο αποτελείται από 8bits και ορίζει σε ποιο *stream*(ροή) υπάγεται το πακέτο. Ανάλογα με την κατηγορία του *stream*(ροή) αντίστοιχη θα είναι και η αντιμετώπιση του πακέτου.
- *Payload Length*: Σε αυτό το field έχουμε 16 bits μέγεθος και μας ενημερώνει για το πεδίο των δεδομένων χωρίς το header.
- *Next Header*: Αυτό το field προσδιορίζει τι τύπου επικεφαλίδα είναι η βασική κεφαλίδα IP. Ο τύπος που μπορεί να είναι το header είναι α) επέκταση της κεφαλίδας IPv6,β) κάποια IPv4 header,γ) κάποιο πρωτόκολλο π.χ. TCP.



2.2.2.1 Επικεφαλίδες Επέκτασης

IPv6 extension headers

- Hop-by-Hop Options
 - Routers will look at it
- Routing
 - Source routes
- Fragment
 - Fragmentation, ONLY at originating node
- Destination Options
 - Pass info to final destination
- IPsec (AH, ESP)
- Header chain
 - Header parsing code becomes somewhat different from IPv4

```
IPv6(next=TCP) TCP payload
IPv6(next=routing) Routing(next=TCP) TCP payload
IPv6(next=fragment) Fragment(next=TCP) TCP payload
```

Εικόνα 2-4 Επικεφαλίδες Επέκτασης

Οι επικεφαλίδες επέκτασης μας δίνουν τη δυνατότητα για περισσότερες επιλογές. Η επιλογή *Hop-by-Hop* και *Destination Options* είναι *headers* που έχουν χώρο για πολλές άλλες επιλογές. Τα *Hop-by-Hop Options* μπορούν να δεχθούν επεξεργασία από οποιονδήποτε δρομολογητή περνά το πακέτο που έχει αυτή την συγκεκριμένη επιλογή. Όλες οι άλλες επιλογές(*options*) δέχονται επεξεργασία μόνο από τον προορισμό τους. Οι επικεφαλίδες επέκτασης αναφέρονται με λεπτομέρεια στο *RFC 2460*. [27][28]

2.3 Λόγοι μετάβασης στην IP version 6

Η ήδη υπάρχουσα έκδοση IP version 4 στάθηκε δυνατή και αντάξια των προσδοκιών της. Κάλυψε σε ικανοποιητικό βαθμό τις απαιτήσεις σχεδιασμού της και εδραιώθηκε τελικά στον παγκόσμιο ιστό. Όμως, εδώ και κάποια χρόνια είχαν αρχίσει να σημειώνονται σοβαρά προβλήματα στα οποία η έκδοση αυτή δεν ήταν σε θέση να τα επιλύσει. Οι τεράστιες εξελίξεις στον τεχνολογικό και στο δικτυακό τομέα καθώς και το πλήθος των χιλιάδων νέων εφαρμογών που εμφανίστηκαν, κατέστησαν το IPv4 αδύναμο και μη αξιόπιστο σε ικανοποιητικά επίπεδα μιας και όταν είχε δημιουργηθεί η έκδοση 4 υπήρχαν πολύ λίγα δίκτυα υπολογιστών. Έπειτα θα αναφέρουμε τους κυριότερους λόγους και απαιτήσεις που οδήγησαν στο σχεδιασμό και την υλοποίηση του IPv6.

➤ Έλλειψη σε IP διευθύνσεις

Οι σχεδιαστές όταν δημιουργήθηκε το IPv4 χρησιμοποίησαν 32 bit για μια διεύθυνση IP που για τα τότε δεδομένα θεωρούνταν ένα πολύ καλό νούμερο και μπορούσε να καλύψει τις απαιτήσεις. Βέβαια με την αυξημένη ανάπτυξη του διαδικτύου αυτό το νούμερο δεν απέδιδε στα μέγιστα επίπεδα. Έτσι χρειάζονταν μεγαλύτερες διευθύνσεις έτσι ώστε να επιλυθεί το πρόβλημα και να συνεχιστεί να αυξάνεται το μέγεθος του διαδικτυακού ιστού. Το μέγεθος τώρα της επόμενης έκδοσης IPv6 έγινε 128 bits, τέσσερις φορές μεγαλύτερο από τη IPv4 έκδοση δηλαδή, περίπου 6x10²⁰ διευθύνσεις σε κάθε τετραγωνικό μέτρο της επιφάνειας της γης. Έτσι ο κάθε χρήστης μπορεί να κατέχει πολλαπλές διευθύνσεις IP για τις διάφορες συσκευές που έχει στην διάθεση του (φορητούς υπολογιστές, tablets).

➤ Δυσκολία Διαχείρισης



Βασικό πρόβλημα στο IPv4 υπήρχε στο κομμάτι της διαχείρισης και από τη πλευρά του χρήστη και από την πλευρά του διαχειριστή. Αυτό συνέβη διότι το πρωτόκολλο είχε σχεδιαστεί για να ικανοποιήσει τις απαιτήσεις εκείνης της εποχής χωρίς να έχει δοθεί ιδιαίτερη προσοχή στο κομμάτι της διαχείρισης καθιστώντας τη αρκετά δύσκολη και πολύπλοκη. Αυτό βέβαια είχε ως επακόλουθο και την αύξηση του κόστους καθώς όσο πιο περίπλοκο γινόταν το πρωτόκολλο τόσο ανέβαιναν και οι οικονομικές απαιτήσεις. Για την επίλυση προβλημάτων που αφορούν την ευκολία διαχείρισης δημιουργήθηκαν πρωτόκολλα διαμοιρασμού IP διευθύνσεων αυτόματα(DHCP). Ακόμη σε ορισμένους κόμβους υπήρχε δυνατότητα απόδοσης διευθύνσεων χειρωνακτικά από τον διαχειριστή του συστήματος. Με την διαδικασία αυτή βέβαια ανεβαίνει η πολυπλοκότητα διότι είναι απαραίτητη η χρήση DHCP server και ο λεπτομερής καθορισμός όλων των στοιχείων και των λεπτομερειών των επιμέρους κόμβων είναι περίπλοκο και απαιτεί χρόνο για να γίνει από έναν άνθρωπο.

Με την χρήση του IPv6 πρωτοκόλλου έγινε αναβάθμιση του DHCP σε DHCP6 στο οποίο συμπεριλήφθηκαν επιπλέον παραμετροποιήσεις για αυτόματη ρύθμιση διεύθυνσης, όπως η stateless(δίχως διατήρηση κατάστασης) και η statefull(κατάσταση που διατηρείται), όπου όλα τα τερματικά που είναι συνδεδεμένα στο διαδίκτυο έχουν ίδιο prefix το 64bit. Τα εναπομείναντα 64 bits έως τα 128 που έχουμε στο σύνολο απαρτίζονται από 48 bit που παίρνει η mac address των συσκευών και τα άλλα 16 bit αναπαρίστανται με άσσους. Έτσι δεν είναι ανάγκη κάθε φορά να αποκτά καινούργια IP διεύθυνση το ίδιο τερματικό όταν συνδέεται στο ίδιο δίκτυο, μπορεί και διατηρεί την πρώτη IP address που του αποδόθηκε και συνδέεται οποιαδήποτε στιγμή επιθυμεί με αυτήν. Συμπληρωματικά όσον αφορά το κομμάτι της αριθμοδότησης σε ένα δίκτυο το IPv6 έχει επιφέρει σημαντικές αλλαγές καθώς ο διαχειριστής είναι ικανός να αλλάξει τις διευθύνσεις των κόμβων που είναι συνδεδεμένοι στο δίκτυο, πραγματοποιώντας αλλαγή του prefix στον κεντρικό router.

➤ Υποστήριξη Φορητότητας

Με την έννοια αυτή αναφερόμαστε στην ικανότητα σύνδεσης των συσκευών(devices) σε ένα δίκτυο από διαφορετικά μέρη ανά πάσα στιγμή. Το IPv4 είχε τις προδιαγραφές να υποστηρίζει την έννοια της φορητότητας αλλά εξαιτίας του πλήθους των συσκευών που είναι πολύ μεγαλύτερο από το πλήθος των IP διευθύνσεων που έχουν απομείνει το πρωτόκολλο δεν δούλεψε διότι δεν γίνονταν αντιστοίχιση κάθε συσκευής με μία ξεχωριστή IPv4 διεύθυνση. Το IPv6 εισήγαγε την εφαρμογή MobileIPv6 με την οποία μια συσκευή είναι ικανή να αλλάξει το σημείο από το οποίο συνδέεται στο Internet. Όταν τώρα ένας κινούμενος κόμβος (mobile node) έγκειται σε διαφορετικό δίκτυο παίρνει μια Local IP η οποία λέγεται Care of Address (CoA). Στη συνέχεια ο κινούμενος κόμβος στέλνει την CoA στο τοπικό δίκτυο (home agent) του νέου δικτύου έτσι αυτός να την “κλειδώσει” για να μην αποδοθεί η συγκεκριμένη διεύθυνση σε κάποιον άλλον σύντομα. Όταν επιτευχθεί η παραπάνω διαδικασία(binding)ο home agent κάνει προώθηση πακέτων διαμέσων ενός τούνελ στον κινούμενο κόμβο στην διεύθυνση CoA που είναι δηλωμένη.Καθόλη τη διάρκεια που ο mobile node περνάει μέσα από διαφορετικά δίκτυα αποστέλνει binding updates με την CoA. Για να μην υπάρξουν δυσκολίες στη δρομολόγηση το IPv6 διαθέτει συστήματα ασφαλείας τα οποία επαληθεύουν την ταυτότητα του κινούμενου κόμβου και ποιά είναι η διεύθυνση επικοινωνίας του. Επίσης ο mobile node ενημερώνει οποιονδήποτε κόμβο επικοινωνεί μαζί του να του τα αποστέλνει πακέτα στην προσωρινή και όχι στην home address.



➤ *Αύξηση Αποδοτικότητας*

Φυσικά στο νέο πρωτόκολλο δεν θα έλειπαν και βελτιώσεις που αφορούν την αποδοτικότητα. Διατηρήθηκαν ορισμένα καλά στοιχεία που υπήρχαν στο IPv4 και αφαιρέθηκαν αρκετά άλλα τα οποία δεν είχαν πλέον απήχηση. Ορισμένες βελτιώσεις στην έκδοση 6 είναι το σταθερό μέγεθος του header που οδηγεί πρακτικά σε άνετη διαχείριση και χαμηλή πολυπλοκότητα. Στη διαδικασία δρομολόγησης το πακέτο δεν είναι αναγκαίο να χωρίζεται υπό πακέτα και υπάρχει και η δυνατότητα επικοινωνίας ώστε να υποδέχονται πακέτα μικρότερα σε μέγεθος. Στην έκδοση 4 όταν κάποια συσκευή ήθελε να εξετάσει ένα μήνυμα αναστέλλονταν όλες οι συσκευές που ήταν συνδεδεμένες εκείνη την ώρα. Από την άλλη μεριά στην έκδοση 6 εξαιτίας της λειτουργίας *Multicast* που διαθέτει αναστέλλονται μονάχα αυτές που επιθυμούν να εκτελέσουν την επεξεργασία και οι υπόλοιπες παραμένουν σε σύνδεση.

➤ *Ασφάλεια*

Στους βασικούς λόγους που χρειάστηκε η μετάβαση από το IPv4 στο IPv6 είναι φυσικά η ασφάλεια. Η χρήση της *IPsec* δημιούργησε κάποια προβλήματα στην έκδοση 4. Η *IPsec* που χρησιμοποιούν όλα τα πρωτόκολλα IP για την ασφαλή μεταφορά δεδομένων μεταξύ υπολογιστών έχει εφαρμογή στο επίπεδο δικτύου 3. Το κυριότερο πλεονέκτημα όσον υπηρεσιών το χρησιμοποιούν είναι ότι προστατεύονται κατά τη διαδικασία μεταφοράς δεδομένων και έτσι τα δεδομένα τους παραμένουν αμετάβλητα και μη τροποποιήσιμα. Με άλλα λόγια αυτό που κάνει είναι μία διαδικασία κρυπτογράφησης των δεδομένων. Στο IPv4 η χρήση του *IPsec* είναι optional έτσι η υποστήριξη του προσθέτει επιπλέον παραμετροποίηση άρα και μεγαλύτερη πολυπλοκότητα. Ακόμη βασικό πρόβλημα της έκδοσης 4 όσον αφορά την *IPsec* είναι ότι χρησιμοποιεί το *NAT* (*Network Address Translation*) το οποίο διακόπτει την από άκρο σε άκρο επικοινωνία που απαιτεί το *IPsec* για να αποδίδει στα μέγιστα. Αυτό συμβαίνει διότι με την μετάφραση που κάνει το *NAT* η πρωταρχική διεύθυνση μεταβάλλεται και δεν ταυτίζεται με την τελική συνεπώς το *IPSec* δεν ανταποκρίνεται.

Στο IPv6 τώρα, τα προβλήματα αυτά έχουν εξαλειφθεί μιας και η λειτουργία της ασφάλειας είναι ενσωματωμένη. Επιπλέον οι μηχανισμοί ασφαλείας που χρησιμοποιεί το IPv6 έχουν τη δυνατότητα να χρησιμοποιηθούν και από άλλους μηχανισμούς, αντίθετα στο IPv4 για οποιαδήποτε μεταβολή π.χ. μια επέκταση ή μια προσθήκη θα πρέπει ο μηχανισμός να βρίσκει μηχανισμούς ασφαλείας.

Σε ένα άλλο κομμάτι της ασφάλειας που είναι πολύ καλύτερη η έκδοση 6 αφορά το θέμα των ιομορφικών λογισμικών. Οι ιοί και πιο λεπτομερώς τα worms παρεκκλύουν την ταχύτητα των υπολογιστών και χρησιμοποιούν πόρους απ' αυτούς για δικές τους επιδιώξεις όπως παραδείγματος χάριν μετάδοση του ιού σε άλλο τερματικό. Στο IPv4 όπως γνωρίζουμε, οι συσκευές ενός υποδικτύου έχουν το μέγιστο 16bit και έτσι ήταν αρκετά εύκολο για κάποιον ιό να σαρώσει όλες τις συσκευές ταχύτατα. Αντίθετα στο IPv6 οι συσκευές του υποδικτύου έχουν 64 bit ως επακόλουθο να είναι ανέφικτη τέτοιου τύπου επίθεση, γιατί θεωρητικά θα είναι σαν να πρέπει να σαρώσει δύο φορές όλο το IPv4 διαδίκτυο.[27][28]



2.3.1 Quality of Service (QoS)

Μια θεμελιώδης περιοχή που χρειαζόταν βελτίωση στο IPv4 ήταν αυτή της ποιότητας των υπηρεσιών (*Quality of Service*) μιας και διατελούσαν ορισμένα προβλήματα που έπρεπε να επιλυθούν. Το IPv4 είχε τη δυνατότητα να υποβαστάξει μηχανισμούς QoS στο επίπεδο 3(*network*) με τη χρήση του *field type of service* που εντοπίζεται στην επικεφαλίδα του. Το πεδίο αυτό μας υποδεικνύει τι είδος υπηρεσίας χρειάζεται η κάθε εφαρμογή, αυτό έκανε αρκετά δύσκολη τη δουλειά των διαχειριστών αλλά και των δημιουργών της κάθε εφαρμογής. Αυτό συνέβαινε διότι κάθε δρομολογητής(*router*) πρέπει να γνωρίζει λεπτομέρειες για τον πλήθος των μονοπατιών που έπρεπε να πάρει το πακέτο για να καταλήξει στον προορισμό του, ρίχνοντας αισθητά τα επίπεδα της απόδοσης.

Επιπλέον στο IPv4 αν οποιοσδήποτε δρομολογητής επιθυμούσε να γνωρίσει κάτι για μια κίνηση(ροή) τότε χρειαζόταν να μελετήσει και να αναλύσει τους κόμβους που υπάρχουν στην επικοινωνία. Η ίδια διαδικασία έπρεπε να προηγηθεί και για τη θύρα στην επικεφαλίδα του πρωτοκόλλου μεταφοράς. Έτσι με όλη αυτή τη διαδικασία που έπρεπε να ‘‘τρέξουν’’ οι δρομολογητές πρόσθεταν επιπλέον φόρτο και κατ’ επέκταση και κόστος, έχοντας ως απόρροια την μεταβολή στην ποιότητα της υπηρεσίας.

Το IPv6 υλοποιήθηκε για να επιλύσει τέτοιου είδους προβλήματα. Στο *header (επικεφαλίδα)* της έκδοσης 6 προστέθηκαν δύο καινούργια *fields* το *Traffic class* και το *Flow label*. Αυτά τα πεδία είναι αναγκαία για να μπορούν να σταθούν ικανοποιητικά μηχανισμοί και *services* με προκαθορισμένες απαιτήσεις ποιότητας. Στο *Flow label* υπάρχουν όλα τα δεδομένα που πρέπει να γνωρίζει ένας δρομολογητής έτσι ώστε να πραγματοποιήσει άρτια τη δρομολόγηση ενός πακέτου προς το τερματικό του αποφεύγοντας περιττή αναζήτηση σε περαιτέρω *fields*. Ως αποτέλεσμα έχουμε πολύ καλά επίπεδα απόδοσης σε μικρό χρονικό διάστημα.

Σε ένα άλλο τομέα που η QoS είναι αναγκαία είναι οι νέες πολυμεσικές εφαρμογές δικτύων. Οι εφαρμογές αυτές εμπεριέχουν συνδυασμό εικόνας και ήχου καθώς και συνεργασία με αυτούς που τις επεξεργάζονται δηλαδή επικοινωνία σε πραγματικό χρόνο (*real time*). Αυτό υποδηλώνει ένα μεγάλο όγκο δεδομένων και μια συνεχόμενη ροή πληροφορίας διαμέσων του διαδικτύου έτσι ώστε να διατηρείται η ροή αυτών των πληροφοριών μέσω του Internet χωρίς αναστολές στη σύνδεση. Το πρωτόκολλο IP χρειάζεται να μην αλλάζει σε τακτό χρονικό διάστημα τα δρομολόγια και να κρατάει ένα σταθερό ρυθμό *bit* στη μετάδοση. Καθοριστικό ρόλο στην ποιότητα του αποτελέσματος παίζει και η καθυστέρηση στη μετάδοση της πληροφορίας. Οι συγκεκριμένοι λόγοι καθιστούν δύσκολη τη χρήση τέτοιου τύπου πολυμεσικών εφαρμογών από την *version 4* που δεν είναι κατάλληλη για μετάδοση πληροφοριών σε *real time*. Συνεπώς το IPv6 αναλύθηκε και σχεδιάστηκε έτσι ώστε να μπορεί να καλύψει αυτές τις απαιτήσεις δημιουργώντας μηχανισμούς για να επιτύχει αυτό το σκοπό.[31]

2.4 Νέα Χαρακτηριστικά του IPV6

2.4.1 Τύποι Διευθύνσεων

Πρώτα θα δούμε τις κατηγορίες των διευθύνσεων και πως τις υποστηρίζει το πρωτόκολλο.

- *Multicast*: Το χρησιμοποιούμε όταν θέλουμε να αποστείλουμε πακέτα σε πολλαπλούς προορισμούς. Η λειτουργία του επικεντρώνεται στην αποστολή πακέτων σε *interfaces* που είναι σε μια *multicast* ομάδα. Η πιο συνηθισμένη χρήση του είναι για



streaming, καθώς η πηγή στέλνει το πακέτο μια φορά και μπορεί να υπάρχουν πολλοί παραλήπτες. Μία Multicast address είναι υλοποιημένη με την παρακάτω μορφή.

| | | | | |
|-------------|----------|---------|-------------|--------------------|
| 8bits | 4bits | 4bits | 112bits | 128 bits |
| (a)11111111 | (b)flags | (c)scop | (d)group id | Συνολική διεύθυνση |

Πίνακας 2-1 Multicast Διεύθυνση στο IPv6

- (a) Προσδιορισμός διεύθυνσης ως multicast
- (b) Καθορίζει πότε μια διεύθυνση είναι multicast
- (c) Προσδιορισμός του εύρους της ομάδας και του σκοπού της διεύθυνσης
- (d) Καθορισμός multicast ομάδας

- *Unicast*: αποτελεί τον πιο συνηθισμένο τύπο διεύθυνσης IP. Γίνεται χρήση για μία διεπαφή στο σημείο όπου γίνεται η σύνδεση του κόμβου με το δίκτυο. Ένας κόμβος όπως θα δούμε και στη διαδικασία προσομοίωσης σε παρακάτω κεφάλαια μπορεί να έχει πολλαπλά interface. Η Unicast address εισάγεται στην κεφαλίδα του πακέτου προορισμού. Έχει τη δυνατότητα υποστήριξης διεθύνσεων όπως site local, IPV4 compatible, global aggregatable. Οι Unicast διευθύνσεις έχουν την παρακάτω διάκριση:

- Link local address
- Global unicast addresses
- Unique local addresses
- Special addresses
- Transition addresses

- *Loopback*: Η loopback ή αλλιώς διεύθυνση ανατροφοδότησης είναι ένας ειδικευμένος τύπος διεύθυνσης που γίνεται χρήση για το routing ηλεκτρονικών σημάτων. Ένα τερματικό ακόμη και αν δεν έχει κανένα δικτυακό interface αν στείλει πακέτα στη loopback address ουσιαστικά το πακέτο στέλνεται πίσω στον εαυτό του.
- *Unspecified*: Η μορφή μιας τέτοιας διεύθυνσης είναι 0:0:0:0:0:0:0: και δεν ανατίθεται σε καμία περίπτωση σε κόμβο. Κάτι τέτοιο υποδηλώνει την απουσία IPV6 διεύθυνσης. Παραδείγματος χάριν, κόμβοι που έχουν μόλις δημιουργηθεί,



χρησιμοποιούν μια τέτοια διεύθυνση σαν διεύθυνση “ξεκίνημα” μέχρι να πάρουν μία IPV6 διεύθυνση.

| <i>Τύπος</i> | <i>Κανονική μορφή</i> | <i>Απόδοση</i> |
|----------------------------|-----------------------------------|-----------------------------|
| <i>Unicast Address</i> | <i>1080:0:0:0:8:800:200C:417A</i> | <i>1080:8:800:200C:417A</i> |
| <i>Multicast address</i> | <i>FF01:0:0:0:0:0:101</i> | <i>FF01::101</i> |
| <i>Loopback address</i> | <i>0:0:0:0:0:0:0:1</i> | <i>::1</i> |
| <i>Unspecified address</i> | <i>0:0:0:0:0:0:0:0</i> | <i>::</i> |

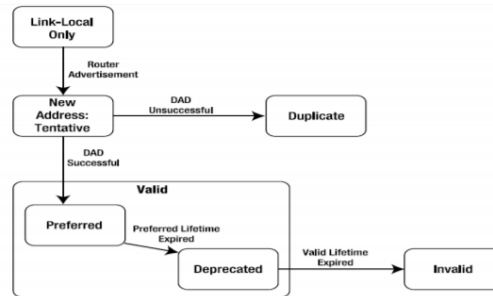
Πίνακας 2-2 Παραδείγματα απεικόνισης διευθύνσεων και Συμπίεση μηδενικών στις IPv6 διευθύνσεις

2.4.1.1 Ειδικοί Τύποι Διευθύνσεων

- *IPv4MappedIPv6Address*: Οι διευθύνσεις αυτές δίνουν άδεια σε εφαρμογές του νέου πρωτοκόλλου να δουλεύουν σε κόμβους του IPv4 και του IPv6. Πιο αναλυτικά οι συγκεκριμένες διευθύνσεις βοηθούν αρκετά γιατί χρειάζεται αρκετό χρονικό διάστημα για την πλήρη χρήση του IPv6 σε όλους τους σταθμούς κι έτσι οι εφαρμογές της προηγούμενης έκδοσης θα συνεχίσουν να υπάρχουν.
- *IPv4CompatibleIPv6Address*: Τέτοιου τύπου διευθύνσεις οι διευθύνσεις αποδίδονται σε κόμβους που υποστηρίζουν και τις δύο εκδόσεις αλλά δεν έχουν γειτονικό router που να υποστηρίζει την έκδοση 6.
- *Link Local Address*: Τέτοιες διευθύνσεις είναι κατάλληλες για την επικοινωνία μέσα σε ένα τοπικό δίκτυο. Η καταχώρηση των διευθύνσεων αυτών γίνεται χειροκίνητα από τον διαχειριστή ή αυτόματα από το σύστημα. Αυτού του τύπου οι διευθύνσεις χρησιμοποιούνται για όσους κάνουν σύνδεση σε δίκτυα οργανισμών διαμέσων τηλεφωνικής σύνδεσης.[32]

2.4.2 Ανίχνευση Ίδιων Διευθύνσεων (Duplicate Address Detection, DAD)

Η έκδοση 6 για να ελαχιστοποιήσει την πιθανότητα δύο τερματικά να έχουν την ίδια IPV6 διεύθυνση, ανιχνεύουν ίδιες διευθύνσεις για τις νέες διευθύνσεις IPV6 πριν διατεθούν προς χρήση. Οι unicast και οι link-local διευθύνσεις χρησιμοποιούν το DAD. Οι διευθύνσεις τύπου anycast δεν χρησιμοποιούν DAD μιας και η χρήση της anycast διεύθυνσης είναι ότι πολλαπλά μηχανήματα έχουν την ίδια IP address. Επίσης χρήση του DAD γίνεται σε διευθύνσεις όπου η «ταυτότητα διεπαφής» έχει ελεγχθεί ωρίτερα.



Εικόνα 2-5 Κύκλος ζωής μιας IPv6 διεύθυνσης

Από την παραπάνω εικόνα βλέπουμε ότι το σύστημα ξεκινά με μία link-local διεύθυνση. Όταν λάβει μία διεύθυνση δρομολογητή που έχει ένα ή και περισσότερα prefix με το flag autonomous address configuration ενεργοποιημένο, γίνεται δημιουργία διευθύνσεων που έχουν ταυτότητες διεπαφής σύμφωνα με το RFC 3041. Η διεύθυνση χαρακτηρίζεται ως δοκιμαστική και εκτελείται το DAD.

Έχουμε τα εξής πιθανά αποτελέσματα:

- Ο υπολογιστής ειδοποιείται από κάποιο γειτονικό κόμβο ότι χρησιμοποιείται ήδη η διεύθυνση.
- Το τερματικό δέχεται μήνυμα “neighbor solicitation”, από κάποιο άλλο τερματικό που εκτελεί DAD.
- Δεν επιστρέφεται κάποια απάντηση πίσω [33][34]

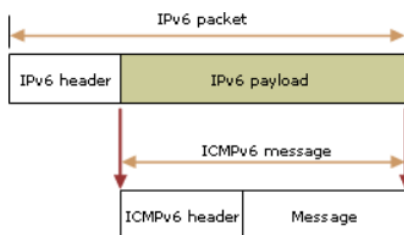
2.4.3 Χρόνος ζωής μια διεύθυνσης IPV6

Οι διευθύνσεις έχουν τη δυνατότητα αυτορύθμισης. Το συγκεκριμένο χαρακτηριστικό της αυτορύθμισης μένει μέχρι το μήνυμα για τον προτεινόμενο χρόνο ζωής από το δρομολογητή περατωθεί. Αυτό βέβαια δε συμβαίνει σχεδόν ποτέ, γιατί νέα πακέτα από το δρομολογητή θα ενημερώσουν τους μετρητές. Αν από την άλλη δεν υπάρχουν τέτοιου είδους νέα πακέτα, ο συνιστάμενος χρόνος ζωής θα τελειώσει και η διεύθυνση θα καταχωρηθεί σαν “παρωχημένη”. Τα νέα sessions που ξεκινάνε δεν ενδεικνύονται να χρησιμοποιούν τέτοιες διευθύνσεις αλλά να προτιμούν νέες διευθύνσεις αν βέβαια υπάρχουν σε διαθεσιμότητα. Εντούτοις ορισμένες σύνοδοι (sessions) που υπάρχουν συνεχίζουν να κάνουν χρήση παρωχημένων IP διευθύνσεων. Αυτό έχει ως αποτέλεσμα ο χρόνος που είναι έγκυρες οι διευθύνσεις να τελειώσει και αυτό έχει ως επακόλουθο την απότομη αποσύνδεση τους από το interface που χρησιμοποιούνται. Αυτό βέβαια θα δίνει τέλος σε όποιο session χρησιμοποιεί ακόμη το interface.

2.4.4 Πρωτόκολλο ICMPv6

Το ICMPv6 αποτελεί βασικό μέρος του IPv6 καθώς εκτελεί λειτουργία ενημέρωσης όσον αφορά τα σφάλματα που προκύπτουν, ή ορισμένες άλλες λειτουργίες διάγνωσης παραδείγματος χάριν ping. Επίσης εμπεριέχει ένα πλήρες πλαίσιο για την προσθήκη επεκτάσεων και για την εκτέλεση μελλοντικών παραμετροποιήσεων. Επιπλέον διαχειρίζεται multicast ομάδες.

Τα μηνύματα τα οποία εμφανίζει το ICMPv6 είναι: a) μηνύματα εμφάνισης λάθους b) μηνύματα πληροφορίας



Εικόνα 2-6 Δομή ενός ICMPv6 μηνύματος

Παρακάτω παραθέτουμε τα βασικά μηνύματα που μπορεί να μας εμφανιστεί στο IPv6 και την κυριότερη περιγραφή τους

| Μήνυμα ICMPv6 | Περιγραφή |
|-------------------------|--|
| Destination Unreachable | Μήνυμα που μας ενημερώνει ότι προέκυψε κάποιο λάθος και ότι το πακέτο δεν μπορεί να παραδοθεί |
| Packet Too Big | Μήνυμα σφάλματος που μας ενημερώνει ότι το πακέτο είναι πολύ μεγάλο για να αποσταλεί |
| Time Exceeded | Μήνυμα σφάλματος που μας ενημερώνει ότι το hop limit ενός IPv6 πακέτου έχει λήξει |
| Parameter Problem | Μήνυμα που μας ενημερώνει ότι έχει εντοπιστεί ένα σφάλμα κατά την επεξεργασία της κεφαλίδας IPv6 ή της κεφαλίδας επέκτασης |
| Echo Request | Ένα ενημερωτικό μήνυμα που χρησιμοποιείται για να προσδιοριστεί εάν ένας κόμβος IPv6 μπορεί να διατεθεί στο δίκτυο. |
| Echo Reply | Ένα ενημερωτικό μήνυμα που χρησιμοποιείται για να απαντήσει στο Echo Request. |

Πίνακας 2-3 Βασικά μηνύματα του IPv6

2.4.5 Router Discovery

Το *router discovery* γίνεται είτε με κάποιο είδος διαφήμισης του δρομολογητή είτε σαν απάντηση του router σε κάποια αιτήματα από IPv6 hosts. Τα router advertisements (διαφημίσεις) περιλαμβάνουν μια λίστα από προθέματα. Με τη χρήση αυτών των προθεμάτων γίνεται αυτόματη σύνθεση IP διευθύνσεων και εντοπισμός διπλών διευθύνσεων. Εάν σε ένα πακέτο υπάρχει πρόθεμα on-link τότε προχωράει στον επόμενο δρομολογητή. Τα μηνύματα διαφήμισης μπορεί να είναι:

- προθέματα on-link IPv6 που χρησιμοποιούνται από τους τοπικούς κόμβους για να ρυθμιστούν αυτόματα οι διευθύνσεις IPv6
- Στοιχεία μέσα στη διαφήμιση όπως το πόσο καιρό θα είναι ενεργό το πρόθεμα
- Ένα τμήμα των flags που μας δείχνουν το είδος της αυτόματης διαμόρφωσης (stateless, statefull)



- Στοιχεία του default δρομολογητή που αποστέλλει τη διαφήμιση, και το χρόνο που χρειάζεται να σταλεί το μήνυμα
- Στοιχεία των hosts, (hop limit, maximum transmission unit-MTU).

2.4.6 Εντοπισμός γειτόνων

Η λειτουργία της ανίχνευσης γειτόνων ακολουθεί την παρακάτω διαδικασία. Υπάρχει ένας host ο οποίος αποστέλλει ένα πακέτο IPv6, κάνει έλεγχο την cache της γειτονικής διεύθυνσης και στη συνέχεια προσδιορίζει ποια θα είναι η διεύθυνση του συνδέσμου στον επόμενο κόμβο. Ένα επίπεδο πρόσβασης εξηγεί αν μπορούμε να έχουμε πρόσβαση στη γειτονική διεύθυνση. Στην έκδοση 6 μία διεύθυνση λογίζεται σαν προσβάσιμη αν έχει δεχθεί μήνυμα επιβεβαίωσης ότι έχει λάβει τα πακέτα από κάποια γειτονική διεύθυνση. Αυτό μπορεί να επιτευχθεί με τις παρακάτω ενέργειες.

Με το να ληφθεί ένα *advertisement* από κάποια διεύθυνση γειτονικού κόμβου, σαν απάντηση σε πρόσκληση γείτονα που έχει σταλεί από κάποιο host. Επίσης μπορεί να ενημερωθεί από πρωτόκολλα ανώτερου επιπέδου για το αν είναι προσβάσιμη η διεύθυνση. Αυτά τα “handshakes” καταχωρούνται σε κάποια λίστα έτσι ώστε να γνωρίζουμε ποιες διευθύνσεις είναι προσβάσιμες. Ο host στέλνει πρόσκληση σε γειτονικό κόμβο αν δεν υπάρχει προσβάσιμη διεύθυνση στη λίστα έτσι ώστε να οριστεί κάποια διεύθυνση που θα είναι προσβάσιμη για να μπορεί να στείλει το πακέτο. [35]

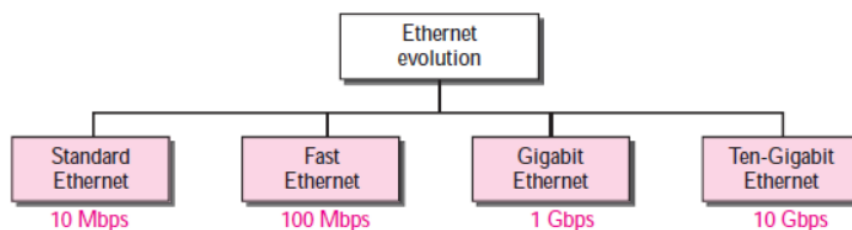


3 Μελέτη Δικτυακών Πρωτοκόλλων

3.1 Ethernet

Το κυριότερο πρωτόκολλο που χρησιμοποιούμε ευρέως για μικρής τοπολογίας δίκτυα είναι το Ethernet και αποτελεί την πιο γνωστή μέθοδο για υλοποίηση τοπικών δικτύων σε τοπολογίες Star ή BUS, βασισμένοι στην αρχιτεκτονική που ακολουθούν τα δίκτυα διαχωρίζονται σε *Peer-to-Peer* και *Client-Server* όπως αναφέραμε και στις παραπάνω ενότητες. Το συγκεκριμένο πρωτόκολλο αποτελείται από δύο βασικές υποκατηγορίες και διαφοροποιούνται στο ρυθμό μεταφοράς δεδομένων.

Η πρώτη υποκατηγορία είναι η απλή Ethernet με ταχύτητα 10Mbps και η άλλη είναι η Fast Ethernet με ταχύτητα 100Mbps. Ακόμη υπάρχει και η Gigabit Ethernet η οποία κυμαίνεται σε ταχύτητα 1000Mbps(1Gbps) αλλά εξαιτίας του πολύ μεγάλου κόστους της δεν την βλέπουμε συχνά. Εμείς θα μελετήσουμε σε σενάρια προσομοίωσης στα επόμενα κεφάλαια αυτές τις κατηγορίες σε διαφορετικές δικτυακές τοπολογίες. Το Ethernet μεταδίδει πακέτων δεδομένων μεταβλητά σε μέγεθος της τάξης των 72 έως και των 1518Byte χρησιμοποιώντας την τεχνολογία CSMA. Το κάθε πακέτο έχει στο περιεχόμενο του ένα header (κεφαλή-κεφαλίδα) όπου εκεί βρίσκουμε πληροφορίες σχετικά με την διεύθυνση αποστολής και λήψης.



Εικόνα 3-Εξέλιξη του Ethernet διαμέσων τεσσάρων γενιών

Fast Ethernet Interface

Το Fast Ethernet όπως είπαμε και στην παραπάνω παράγραφο μας παρέχει ένα εύρος ζώνης της τάξης των 100Mbps. Εκτός από τη μεγαλύτερη ταχύτητα που παρέχει το Fast Ethernet, μεγάλη επιμέλεια δόθηκε στο να μην “συγχυστεί” η τρέχουσα καλωδιακή υποδομή. Για αυτό το λόγο δημιουργήθηκαν επιμέρους υποκατηγορίες όπως το 100Base-TX, 100Base-FX και 100Base-T4. Οι τρεις αυτές διαφορετικές εκδόσεις έχουν προκύψει φυσικά αναλογιζόμενοι την ποιότητα της καλωδίωσης. Η υποκατηγορία 100 Base-TX για καλώδια DTP κατηγορίας 5,η 100 Base-FX χρησιμοποιείται για οπτικές ίνες και η 100 Base-T4 για απλό μη θωρακισμένο συστρέφου ζεύγους καλωδίων (UTP) κατηγορίας 3.

Όσον αφορά την έκδοση 100 Base-T4 η πρόσβαση στο μέσο γίνεται με μεταδόσεις CFMA/CD και half- duplex. Οι υποκατηγορίες 100 Base-TX και 100 Base-FX υποστηρίζουν επιπλέον full-duplex μετάδοση με ταχύτητα 100 Mbps για λήψη και 100 Mbps για την αποστολή δεδομένων. Το fast ethernet χρησιμοποιεί κωδικοποίηση σε bit για να διατηρεί τη συχνότητας του ρολογιού σε χαμηλά επίπεδα.



Gigabit Ethernet Interface

Το *Gigabit Ethernet* αποτελεί το νεότερο πρότυπο της οικογένειας IEEE 802.3 και είναι γνωστό σαν standard IEEE 802.3z. Στόχος του απ'τη μία είναι η ομαλή ενσωμάτωση ακολουθώντας το πρότυπο 802.3 και απ' την άλλη η αύξηση στην ταχύτητα επικοινωνίας της τάξης των 1000 Mbps. Κάνει χρήση οπτικής ίνας από 500 m αν μιλάμε για πολύτροπη οπτική ίνα μέχρι 3 km για μονότροπη οπτική ίνα ή UTP κατηγορίας 5 για μικρές αποστάσεις μέχρι 100m. Η λειτουργία του γίνεται σε full duplex και σε half duplex transmission, όπου η δεύτερη κάνει χρήση του CSMA/CD πρωτοκόλλου. Το *Gigabit Ethernet interface* μπορεί να χρησιμοποιηθεί σαν δίκτυο κορμού για μία κτιριακή υποδομή ή για τη διασύνδεση ισχυρών servers που χρησιμοποιούν πολλά μηχανήματα πολυεπεξεργασίας και απαιτητικές εφαρμογές.

Αυτό που παρατηρείται είναι ότι το gigabit Ethernet δημιουργεί τεράστιες δυνατότητες και προδιαγραφές στον τομέα των τοπικών δικτύων με την πολύ μεγάλη ταχύτητα που μπορεί να προσφέρει. Πιο συγκεκριμένα με την παραλλαγή του 1000BaseT γίνεται αρκετά κερδοφόρο και αποτελεσματικό, γιατί μπορεί να χρησιμοποιήσει την υπάρχουσα καλωδιακή υποδομή τύπου cat 5. Από τώρα και στο εξής οι περισσότερες κατασκευαστικές εταιρίες που ασχολούνται με δίκτυα δουλεύουν και βγάζουν προς πώληση μια μεγάλη ποικιλία από gigabit switches σε πολύ καλές τιμές συγκριτικά με άλλες τεχνολογίες που προσφέρουν πιο μικρό εύρος ζώνης. Οι πρόσφατες εκδόσεις των Gigabit Ethernet έχουν 10 gb/s, 40gb/s, 100gb/s ρυθμό μεταφοράς δεδομένων. Τα δίκτυα αυτού του τύπου λέγονται δίκτυα των 10Gb, των 40Gb και των 100Gb Ethernet. Υπό κατασκευή βρίσκονται τα δίκτυα των 400Gb.[12]

3.2 Εισαγωγή Στη Δρομολόγηση

Οι αλγόριθμοι δρομολόγησης μπορούν να ομαδοποιηθούν σε δυο μεγάλες κατηγορίες: τους προσαρμοστικούς και τους μη προσαρμοστικούς. Οι μη προσαρμοστικοί αλγόριθμοι (*nonadaptive algorithms*) δεν βασίζονται στις αποφάσεις δρομολόγησης σε μετρήσεις ή εκτιμήσεις της τρέχουσας τοπολογίας ή κίνησης. Αντιθέτως, η επιλογή του δρομολογίου που θα χρησιμοποιηθεί για να φτάσουμε από τον I στον J (για κάθε I και J) υπολογίζεται προκαταβολικά, όχι δυναμικά, και μεταφέρεται στους δρομολογητές κατά την εκκίνηση του δικτύου. Η διαδικασία αυτή ονομάζεται *στατική δρομολόγηση (static routing)*. Για τον λόγο ότι δεν μπορεί να αντιδράσει σε αστοχίες, η συγκεκριμένη δρομολόγηση είναι κυρίως χρήσιμη σε περιπτώσεις όπου η επιλογή δρομολόγησης είναι ξεκάθαρη.

Αντίθετα, οι προσαρμοστικοί αλγόριθμοι (*adaptive algorithms*) μεταβάλλουν τις αποφάσεις δρομολόγησης έτσι ώστε να αντανακλούν τις αλλαγές στην τοπολογία, αλλά και τις αλλαγές στην κίνηση μερικές φορές. Αυτοί οι αλγόριθμοι *δυναμικής δρομολόγησης (dynamic routing)* διακρίνονται ανάλογα με το από που λαμβάνουν τα δεδομένα τους παραδείγματος χάριν τοπικά ή από γειτονικούς δρομολογητές, ως προς το πότε αλλάζουν τα δρομολόγια παραδείγματος χάριν κάθε ΔΤ δευτερόλεπτα, ή όποτε αλλάζει το φορτίο ή όποτε αλλάζει η τοπολογία και ως προς το μέτρο σύγκρισης (*metric*) που χρησιμοποιείται για τη βελτιστοποίηση παραδείγματος χάριν την απόσταση, το πλήθος αλμάτων ή τον εκτιμώμενο χρόνο διέλευσης.



3.2.1 Στατική Δρομολόγηση

Στη στατική δρομολόγηση ο διαχειριστής του δικτύου ρυθμίζει χειροκίνητα στο δρομολογητή το δρόμο για τα δίκτυα προορισμού. Όταν υπάρξει μια αλλαγή στην τοπολογία του δικτύου καλό είναι ο διαχειριστής του να ενημερώνει τους πίνακες δρομολόγησης (*routing tables*). Έτσι διαπιστώνει κανείς ότι σε μεγάλα δίκτυα με πολλούς δρομολογητές η διαχείριση των πινάκων δρομολόγησης τους γίνεται ιδιαίτερα δύσκολη. Η συγκεκριμένη μέθοδος προσφέρει ένα σπυρωτό επίπεδο ελέγχου στη δρομολόγηση αλλά γρήγορα καταλαβαίνουμε πως δεν είναι πρακτική σε μεγάλα δίκτυα. Οι στατικοί δρομολογητές έχουν Διαχειριστική Απόσταση (*Administrative Distance, AD*) ίση με 1 με αποτέλεσμα να προτιμώνται έναντι σε δυναμικούς εφόσον η AD παραμένει σταθερή.

Μία στατική διαδρομή με προσαρμοσμένη AD λέγεται *floating static route*.

Πλεονεκτήματα:

- Ελάχιστη επιβάρυνση CPU/Μνήμης
- Σπυρωτός έλεγχος της δρομολόγησης της δικτυακής κίνησης.
- Μεγαλύτερη ασφάλεια, διότι μόνο ο διαχειριστής επιτρέπει την πρόσβαση σε συγκεκριμένα δίκτυα.

Μειονεκτήματα:

- Οι αλλαγές στην υποδομή πρέπει να γίνουν χειροκίνητα.
- Ανυπαρξία δυναμικής ανοχής σφαλμάτων αν αποτύχει κάποιο link.
- Μη πρακτική σε μεγάλα δίκτυα.

3.2.2 Δυναμική Δρομολόγηση

Ένας δρομολογητής μπορεί να τρέξει περισσότερα από ένα πρωτόκολλα δρομολόγησης συγχρόνως, ειδικά όταν λειτουργεί ως αυτόνομο σύστημα δρομολόγησης που βρίσκεται στα όρια μεταξύ των τμημάτων ενός δικτύου, το οποίο (δίκτυο) καλείται να τρέχει διαφορετικά πρωτόκολλα δρομολόγησης. Σε αυτήν την περίπτωση, η ανακατανομή (*redistribution*) μπορεί να χρησιμοποιηθεί για το διαμοιρασμό των δεδομένων μεταξύ των διαφορετικών πρωτοκόλλων που λειτουργούν στον ίδιο δρομολογητή. Αν μια καθορισμένη διαδρομή καθίσταται μη διαθέσιμη, οι υπάρχοντες κόμβοι πρέπει να καταλήξουν σε μια διαφορετική διαδρομή που θα χρησιμοποιήσουν για να στείλουν τα δεδομένα στον προορισμό τους.

Τις περισσότερες φορές το καταφέρνουν αυτό λόγω της χρήσης πρωτοκόλλων δρομολόγησης που χρησιμοποιούν μία από τις δυο διευρυμένες κλάσεις αλγορίθμων δρομολόγησης: αλγορίθμους διανύσματος απόστασης (*Distance-Vector*) και αλγορίθμους κατάστασης συνδέσμων (*Link-State*), οι οποίες συμπεριλαμβάνουν σχεδόν καθένα αλγόριθμο δρομολόγησης που χρησιμοποιείται σήμερα στο διαδίκτυο. Για παράδειγμα στην πρώτη κατηγορία ανήκουν τα *RIP* και *IGRP*, ενώ στην δεύτερη ανήκουν τα *OSPF* και *IS-IS*. Το *EIGRP* παρουσιάζει χαρακτηριστικά και των δύο ,γι' αυτό και αποτελεί υβριδικό πρωτόκολλο.[12]

3.2.3 DHCP (Dynamic Host Configuration Protocol)

Με το πρωτόκολλο Δυναμικής Διευθέτησης Υπολογιστών Υπηρεσίας, πρέπει να έχει ένα διακομιστή DHCP που είναι υπεύθυνος για την διευθέτηση υπολογιστών. Όταν ξεκινάει ο υπολογιστής, διαθέτει μια διεύθυνση Ethernet ή άλλου συνδέσμου μετάδοσης δεδομένων που



είναι ενσωματωμένη στην κάρτα δικτύου, όμως δεν έχει διεύθυνση IP. Ο υπολογιστής εκπέμπει στο δίκτυο μια αίτηση για διεύθυνση IP. Αυτό το κάνει στέλνοντας ένα πακέτο ΑΝΑΚΑΛΗΨΗ DHCP (discover). Το πακέτο πρέπει να φτάσει στον διακομιστή DHCP. Στην περίπτωση που ο διακομιστής δεν είναι άμεσα συνδεδεμένος στο ίδιο δίκτυο, ο δρομολογητής θα έχει διευθετηθεί ώστε να λαμβάνει τις εκπομπές DHCP και να τις αναμεταδίδει στον διακομιστή DHCP, όπου και αν βρίσκεται αυτός.

Όταν ο διακομιστής λάβει την αίτηση, εκχωρεί μια ελεύθερη διεύθυνση IP και τη στέλνει στον υπολογιστή υπηρεσίας μέσα σε ένα πακέτο ΠΡΟΣΦΟΡΑ DHCP (offer), η οποία μπορεί να αναμεταδοθεί μέσω του δρομολογητή. Για να καταφέρει να το κάνει αυτό, ακόμα και όταν οι υπολογιστές υπηρεσίας δεν διαθέτουν διευθύνσεις IP, ο διακομιστής προσδιορίζει τον υπολογιστή υπηρεσίας χρησιμοποιώντας τη διεύθυνση του Ethernet (η οποία μεταφέρεται μέσα στο πακέτο ΑΝΑΚΑΛΗΨΗ DHCP).

Ένα από τα κύρια ζητήματα που έχει προκύψει με την αυτόματη απόδοση διευθύνσεων IP από μια δεξαμενή διευθύνσεων είναι το χρονικό διάστημα για το οποίο θα γίνει η εκχώρηση της διεύθυνσης IP. Αν ο υπολογιστής υπηρεσίας φύγει από το δίκτυο και δεν επιστρέψει την διεύθυνσή του στον διακομιστή, έχει ως αποτέλεσμα την απώλεια αυτής. Μετά από ορισμένο χρονικό διάστημα, ενδέχεται να χαθούν πολλές διευθύνσεις. Για να μη συμβεί αυτό, οι αποδόσεις διευθύνσεων IP μπορεί να ισχύουν για μια σταθερή χρονική περίοδο, μια τεχνική που ονομάζεται εκμίσθωση (leasing). Λίγο πριν λήξει η εκμίσθωση, ο υπολογιστής υπηρεσίας πρέπει να ζητήσει από τον διακομιστή DHCP μια ανανέωση. Αν αποτύχει να κάνει μια τέτοια αίτηση, ή η αίτηση απορριφθεί, ο υπολογιστής υπηρεσίας δεν θα μπορεί πια να χρησιμοποιήσει τη διεύθυνση που του είχε εκχωρηθεί προηγουμένως.

Αναλυτικότερα χρησιμοποιείται ευρέως στο διαδίκτυο για τη διευθέτηση διαφόρων μορφών παραμέτρων, εκτός από την παροχή διευθύνσεων IP στους υπολογιστές υπηρεσίας. Όπως συμβαίνει και στα εταιρικά και οικιακά δίκτυα, το DHCP χρησιμοποιείται από τους ISP για τον καθορισμό των παραμέτρων των συσκευών μέσω της γραμμής σύνδεσης με το διαδίκτυο, ώστε να μην χρειάζεται να πάρουν τηλέφωνο οι πελάτες στον ISP για να λάβουν αυτές τις πληροφορίες. Συνηθισμένα παραδείγματα δεδομένων που διευθετούνται είναι η μάσκα δικτύου, η διεύθυνση IP της default gateway και οι διευθύνσεις IP των διακομιστών DNS και χρόνου.[19][34]

3.2.4 NAT (Network Address Translation)

Η βασική ιδέα πίσω από την μέθοδο Μετάφρασης Διεύθυνσης Δικτύου είναι η εκχώρηση σε κάθε σπίτι ή εταιρία μία μόνο διεύθυνση IP, ή το πολύ, ένας μικρός αριθμός διευθύνσεων για την κίνηση στο διαδίκτυο. Μέσα στο δίκτυο του πελάτη, ο κάθε υπολογιστής παίρνει μια μοναδική διεύθυνση IP η οποία χρησιμοποιείται για δρομολόγηση εντός του δικτύου αυτού. Όταν όμως ένα πακέτο φεύγει από το δίκτυο και πηγαίνει στον ISP, εκτελείται μια μετάφραση διευθύνσεων από την μοναδική εσωτερική διεύθυνση IP στην κοινόχρηστη δημόσια διεύθυνση IP. Αυτή η μετάφραση χρησιμοποιεί τρεις περιοχές διευθύνσεων που έχουν οριστεί ως ιδιωτικές. Τα δίκτυα μπορούν να τις χρησιμοποιούν εσωτερικά όπως αυτά επιθυμούν. Ο μόνος κανόνας είναι ότι δεν πρέπει να εμφανιστούν ποτέ στο ίδιο το διαδίκτυο πακέτα τα οποία να περιέχουν αυτές τις διευθύνσεις.

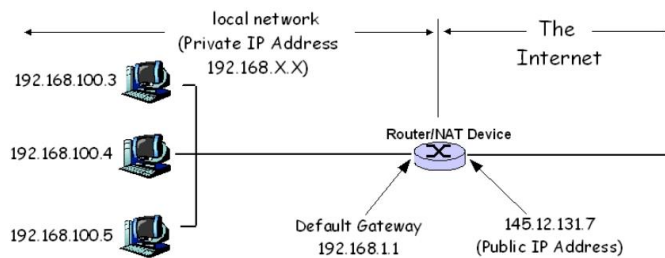
Μέσα στις εγκαταστάσεις του πελάτη, το κάθε μηχάνημα έχει μια μοναδική διεύθυνση της μορφής 10.x.y.z. Όταν όμως ένα πακέτο αφήνει τις εγκαταστάσεις του πελάτη, περνά μέσα από ένα κουτί NAT (NAT box) το οποίο μετατρέπει την εσωτερική διεύθυνση προέλευσης (π.χ. 10.0.0.1) στην πραγματική διεύθυνση IP του πελάτη, που είναι η 192.60.42.12 για παράδειγμα. Το κουτί NAT συνδυάζεται συχνά με μια συσκευή με μια αντιπυρική ζώνη (ή τείχος προστασίας, firewall), η οποία παρέχει ασφάλεια ελέγχοντας προσεκτικά τι εισέρχεται και τι εξέρχεται από το



δίκτυο του πελάτη. Είναι πλέον γεγονός η ενσωμάτωση του κουτιού NAT στον δρομολογητή ή στο μόντεμ ADSL.

Ειδικότερα οι σχεδιαστές της μεθόδου NAT παρατήρησαν ότι τα περισσότερα πακέτα μεταφέρουν ωφέλιμα φορτία είτε με TCP είτε με UDP πρωτόκολλα. Όπως έχει προαναφερθεί και τα δύο έχουν κεφαλίδες που περιέχουν μια θύρα προέλευσης και μια θύρα προορισμού. Οι θύρες αυτές παρέχουν το πεδίο που χρειάζεται για να λειτουργήσει η μέθοδος NAT.

Χρησιμοποιώντας το πεδίο Θύρα προέλευσης, είναι εφικτό να επιλυθεί το ζήτημα της αντιστοίχισης. Όταν μπαίνει στο NAT box ένα εξερχόμενο πακέτο, η διεύθυνση προέλευσης αντικαθίσταται από την πραγματική διεύθυνση IP του πελάτη. Ακόμη το πεδίο Θύρα προέλευσης του TCP αντικαθίσταται από ένα δείκτη προς έναν πίνακα μετάφρασης NAT. Η αντίστοιχη καταχώρηση του πίνακα περιέχει την αρχική διεύθυνση IP και την αρχική θύρα προέλευσης. Τέλος, υπολογίζονται ξανά τα αθροίσματα ελέγχου των κεφαλίδων IP και TCP και εισάγονται στο πακέτο.[36][37]



Εικόνα 3-2 Τρόπος Λειτουργίας NAT μεθόδου

3.2.5 Πρωτόκολλο Δρομολόγησης CDP (Cisco Discovery Protocol)

Το Πρωτόκολλο ανακαλύψεων Cisco (Cisco Discovery Protocol, CDP) είναι ένα ιδιόκτητο πρωτόκολλο δικτύων στο Επίπεδο ζεύξης δεδομένων, το οποίο αναπτύχθηκε από τη Cisco Systems και υλοποιείται στις περισσότερες συσκευές δικτύων της Cisco. Χρησιμοποιείται για την κοινοποίηση πληροφοριών ανάμεσα σε άμεσα συνδεδεμένες συσκευές Cisco, όπως την έκδοση του λειτουργικού συστήματος και τη διεύθυνση IP. Μπορεί επίσης να χρησιμοποιηθεί για τη Δρομολόγηση κατ' απαίτηση, μια μέθοδο για περίληψη πληροφοριών δρομολόγησης μέσα στις ανακοινώσεις του CDP, ώστε να μην είναι αναγκαίο κάποιο πρωτόκολλο δρομολόγησης σε μικρά δίκτυα.

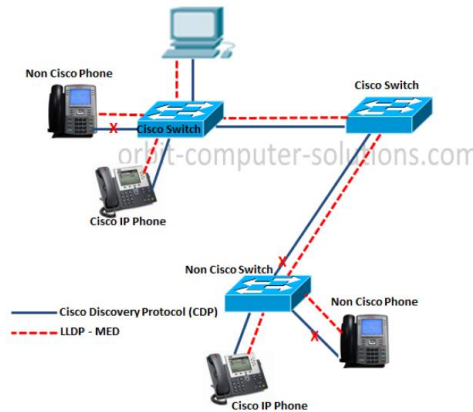
Οι συσκευές Cisco στέλνουν ανακοινώσεις CDP στον προορισμό multicast 01-00-0c-cc-cc-cc, ο οποίος επίσης χρησιμοποιείται και από άλλα πρωτόκολλα της Cisco, όπως το VTP. Από προεπιλογή, οι ανακοινώσεις CDP αποστέλλονται κάθε 60 δευτερόλεπτα από τις διεπαφές της συσκευής που υποστηρίζουν κεφαλίδες Subnetwork Access Protocol (SNAP), όπως οι διεπαφές Ethernet, Frame Relay και Asynchronous Transfer Mode (ATM). Κάθε συσκευή Cisco που υποστηρίζει το CDP αποθηκεύει τις πληροφορίες που λαμβάνει από άλλες συσκευές σε ένα πίνακα, ο οποίος εμφανίζεται με την εντολή show cdp neighbors. Οι πληροφορίες στον πίνακα CDP ενημερώνονται κάθε φορά που λαμβάνεται μια ανακοίνωση, και μηδενίζεται ο χρόνος κατακράτησης για τη συγκεκριμένη καταχώρηση. Ο χρόνος κατακράτησης (από προεπιλογή: 60 δευτερόλεπτα) καθορίζει τη διάρκεια ζωής μιας καταχώρησης μέσα στον πίνακα. Αν δεν ληφθούν



ανακοινώσεις από μια συσκευή πριν τη λήξη του χρόνου κατακράτησης, τότε οι πληροφορίες της συσκευής διαγράφονται από τον πίνακα.

Οι πληροφορίες που περιέχονται στις ανακοινώσεις CDP διαφέρουν ανάλογα με τον τύπο της συσκευής και την έκδοση του λειτουργικού συστήματος που αυτή τρέχει. Αυτές οι πληροφορίες μπορεί να περιέχουν την έκδοση του λειτουργικού συστήματος, το όνομα της συσκευής (hostname), όλες τις διευθύνσεις (π.χ. διευθύνσεις IP) που έχουν ρυθμιστεί από όλα τα πρωτόκολλα που λειτουργούν πάνω στη συγκεκριμένη διεπαφή από την οποία στέλνονται οι ενημερώσεις, το αναγνωριστικό της θύρας από την οποία έρχονται οι ανακοινώσεις, τον τύπο και το μοντέλο της συσκευής, τη ρύθμιση duplex, τον τομέα VTP, το εγγενές VLAN, την κατανάλωση ρεύματος (για συσκευές Power over Ethernet, και άλλες πληροφορίες ανάλογα με τη συσκευή. Οι λεπτομέρειες που περιέχονται σε αυτές τις ανακοινώσεις επεκτείνονται εύκολα χάρη στη δυνατότητα χρήσης της μορφής τύπος-μήκος-τιμή (type-length-value, TLV).

Η CDP έκδοση-2 (CDPv2), ως η πιο πρόσφατη απελευθέρωση του πρωτοκόλλου, παρέχει τα ευφύστερα χαρακτηριστικά γνωρίσματα συσκευών. Αυτά περιλαμβάνουν έναν μηχανισμό υποβολής αναφορών που επιτρέπει τη γρηγορότερη υπόδειξη λάθους. Έτσι επιτυγχάνεται μείωση του χρόνου διακοπής.[38]



Εικόνα 3-3 Χρήση CDP πρωτοκόλλου στις γαλάζιες ζεύξεις

3.3 Πρωτόκολλο VTP

Το VTP είναι ένα πρωτόκολλο δικτύου της Cisco, το οποίο μεταδίδει τον ορισμό των εικονικών τοπικών δικτύων (VLAN) σε ολόκληρο το τοπικό δίκτυο. Όπως γνωρίζουμε για να ρυθμίσουμε ένα VLAN πρέπει να το κάνουμε χειροκίνητα μέσα από κάθε σύνδεση με κάθε switch που θα μετέχει σ' αυτό. Όταν βέβαια έχουμε 10 ή 20 switches δεν μπορούμε να το κάνουμε με αυτόν τον τρόπο. Με το πρωτόκολλο VTP οι ρυθμίσεις VLAN (δημιουργία, διαγραφή, μετονομασία) μπορούν να γίνονται από έναν switch και να μεταδίδονται σε όλους τους υπόλοιπους, οι οποίοι ρυθμίζονται χωρίς άλλη επέμβαση. Το VTP χρειάζεται συνδέσεις trunk για να μεταφέρει τα layer-2 (επίπεδο ζεύξης) μηνύματα του. Τα μηνύματα του VTP, και επομένως η τηλερύθμιση των switches γίνονται στο **domain** που ορίζουμε εμείς. Ένα switch μπορεί να ανήκει μόνο σε ένα domain. Ακόμη ένας switch μπορεί να ανήκει σε μία από τις τρεις παρακάτω κατηγορίες, οι οποίες ρυθμίζουν τον τρόπο που θα συμπεριφέρεται μέσα στο domain :

- **Server** : Ο Server μπορεί να προσθέτει, να αλλάζει, διαγράφει VLANs από το configuration του. Ο Server δέχεται αλλαγές και μέσω VTP μηνυμάτων. Σε κάθε περίπτωση όταν υποστεί κάποια αλλαγή την διαφημίζει σε όλες τις (trunk) θύρες του.



- **Client** : Ο Client δεν μπορεί να κάνει αλλαγές στην διαμόρφωση του. Οι αλλαγές που δέχεται είναι μόνον αυτές που λαμβάνει από κάποιον Server με VTP. Όποτε λάβει κάποιο VTP μήνυμα αλλαγής, αφού την ενσωματώσει στην διαμόρφωση του, προωθεί το μήνυμα στις άλλες trunk θύρες του.
- **Transparent** : Ο Transparent μπορεί επίσης να προσθέτει, αλλάζει, διαγράφει VLANs από το configuration του. Οι αλλαγές όμως δεν διαφημίζονται, δεν δημιουργεί δηλαδή VTP μηνύματα. Ακόμη αγνοεί τα VTP μηνύματα που λαμβάνει, απλά τα προωθεί σε όλες τις θύρες χωρίς να τα λάβει υπόψιν στο σχηματισμό της διαμόρφωσης του.

Αν ο διαχειριστής δεν ορίσει κάτι ένα switch εντάσσεται αυτόματα στην κατηγορία Server. Συνήθως, ορίζεται ένας switch ως server και οι υπόλοιποι ως client. Αυτή η διαμόρφωση είτε διατηρείται διαρκώς και ο διαχειριστής χρησιμοποιεί τον server για να ρυθμίζει όλους τους switch του domain, ή αφού γίνει η ρύθμιση, μετατρέπονται όλοι σε Transparent για να αποφεύγονται λάθη.

Υπάρχουν και περιπτώσεις, όπως στο **VTP pruning**, όπου απαιτείται όλοι οι switch να είναι server. Μια trunk σύνδεση εξ ορισμού ανήκει σε όλα τα VLANs επομένως προωθεί όλα τα broadcasts είτε στην άλλη άκρη υπάρχει μέλος (host) του VLAN είτε όχι. Το VTP pruning είναι μία λειτουργία με την οποία αφαιρούνται (ή και προστίθενται) VLANs σε ένα trunk.

Η λειτουργία του γίνεται μέσω μηνυμάτων που δημιουργούν ή και προωθούν οι switches ανάλογα με την κατάσταση (mode) στην οποία βρίσκονται. Τα μηνύματα, μεταξύ άλλων, περιέχουν ένα αριθμό τον **configuration revision number**. Κάθε φορά που ο switch στέλνει ένα μήνυμα μεταβολής αυξάνει κατά 1 τον αριθμό αυτό. Όσο μεγαλύτερος είναι επομένως, τόσο πιο πρόσφατο είναι το configuration που περιέχει το μήνυμα. [58][59]

3.4 Πρωτόκολλο Spanning Tree(STP)

Η δημιουργία βρόγχων με πλεονάζουσες γέφυρες ή μεταγωγείς αυξάνει την αξιοπιστία του δικτύου μας δημιουργεί όμως και προβλήματα. Παραδείγματος χάριν έχουμε δύο τμήματα δικτύου(Τμήμα 1, Τμήμα 2) που τα έχουμε συνδέσει με δύο bridges. Στην περίπτωση που η bridge 1 σταματήσει να λειτουργεί το δίκτυο δεν θα πέσει διότι η κυκλοφορία πηγαίνει διαμέσω της bridge 2. Το πρόβλημα που μπορεί να δημιουργηθεί είναι ότι μια γέφυρα πλημμυρίζει τα broadcasts. Έστω ότι ένας host από το Τμήμα 1 εκπέμπει ένα broadcast που λαμβάνεται τόσο από την bridge 1 όσο και από την bridge 2. Αφού μιλάμε για broadcast και οι δύο γέφυρες το στέλνουν σε όλες τις υπόλοιπες θύρες τους. Έτσι το broadcast εμφανίζεται δύο φορές στο Τμήμα 2. Κάθε γέφυρα βλέπει το broadcast της άλλης εντός του Τμήματος 2 και το προωθεί στις άλλες θύρες της επομένως περνά σε δύο αντίτυπα στο Τμήμα 1. Η διαδικασία που αναφέρθηκε επαναλαμβάνεται ατέρμονα καταναλώνοντας bandwidth και υπολογιστικούς πόρους σε όλους τις συσκευές των τμημάτων 1 και 2 μιας και πρέπει να επεξεργαστούν το broadcast ξανά και ξανά.

Τη λύση στο πρόβλημα αυτό ήρθε να δώσει το STP πρωτόκολλο που εκτελούν οι γέφυρες (μεταγωγείς) για να εξαφανίσουν αυτά τα loops. Το STP εκτελείται και μπλοκάρει τις πλεονάζουσες θύρες. Έτσι όταν ένα broadcast μήνυμα φτάσει στο port1 της bridge 1 και bridge 2 θα περάσει στο Τμήμα 2 μέσα από το port 2 της bridge 1 καθώς οι υπόλοιπες έχουν μπλοκαριστεί.

Το STP προϋποθέτει επικοινωνία μεταξύ των bridges(switches) που συμμετέχουν. Αυτό πραγματοποιείται με την αποστολή ειδικών μηνυμάτων τα BDPUs(Bridge Protocol Data Units).



Τα BPDUs στέλνονται σαν multicast κάθε 2 sec (Hello timer) και περιέχουν πληροφορίες που βοηθούν switches να ανακαλύψουν την τοπολογία του δικτύου και την ύπαρξη loops. Πιο αναλυτικά εκτελούνται τα παρακάτω βήματα :

1. Εκλέγουν ένα από όλους σαν Root Bridge (Root Switch). Η εκλογή γίνεται με βάση το μικρότερο Switch ID. Το switch ID δημιουργείται από δύο κομμάτια, την προτεραιότητα του switch (priority) και την MAC address του switch (6 bytes length)
2. Με βάση το κόστος θύρας όπου *το κόστος εξαρτάται ανάλογα με το είδος σύνδεσης*, κάθε switch υπολογίζει το κόστος κάθε διαδρομής του προς το Root Switch.
3. Μετά την εκλογή του Root Switch κάθε άλλος switch στο δίκτυο ορίζει μια θύρα του ως root port (RP) για να επικοινωνεί με την root switch. Η εκλογή γίνεται με τα παρακάτω κριτήρια κατά σειρά προτεραιότητας :
 - a. Συσσωρευμένο κόστος κάθε διαδρομής προς τον switch, πόρτα με μικρότερο αθροιστικό κόστος γίνεται root port,
 - b. Σε φάση ισοπαλίας εκλέγεται η θύρα που επικοινωνεί με τον γείτονα switch που έχει μικρότερο ID
 - c. Στη συνέχεια η θύρα με τη μικρότερη προτεραιότητα
 - d. Και τέλος αυτή με τη μικρότερη αρίθμηση (π.χ. e0/0)
4. Τελευταίο βήμα η εκλογή για κάθε Τμήμα του δικτύου της **designated port (DP)** ,μιας θύρας σε ένα μόνο switch του τμήματος, μέσω της οποίας θα επικοινωνεί το τμήμα αυτό με τον root switch. Ο switch λέγεται και designated switch για το συγκεκριμένο τμήμα του δικτύου. Για κάθε τμήμα δικτύου η εκλογή γίνεται με την παρακάτω, κατά σειρά προτεραιότητας διαδικασία :
 - a. Ο switch (port) με το μικρότερο αθροιστικό κόστος
 - b. Αν υπάρχει ισοπαλία στα κόστη ανάμεσα σε διαφορετικούς switches επιλέγεται αυτό με το μικρότερο ID.
 - c. Αν έχουμε τα ίδια κόστη στον ίδιο switch (δηλ. συνδέεται με δύο συνδέσεις στο τμήμα του LAN) τότε επιλέγεται η θύρα με την μικρότερη προτεραιότητα
 - d. Τέλος επιλέγεται η θύρα με τη μικρότερη αρίθμηση. [60][61]



4 Interior gateway protocols (IGP)

Ένα *Interior Gateway Protocol (IGP)* ή πρωτόκολλο εσωτερικών πυλών λέγεται το πρωτόκολλο δρομολόγησης που το χρησιμοποιούμε μέσα σε ένα αυτόνομο σύστημα (AS).

Από την άλλη πλευρά, το *Exterior Gateway Protocol (EGP)* ή πρωτόκολλο εξωτερικού τομέα γίνεται χρήση όταν δρομολόγηση μεταξύ συσκευών διαφορετικών AS και χρησιμοποιεί πρωτόκολλα εσωτερικών πυλών για να ανακαλύψει τη διαδρομή μέσα σε ένα αυτόνομο σύστημα.

Τα Interior Gateway Protocols διασπώνται σε δύο τομείς:

- Distance-vector routing protocol (διανύσματος απόστασης)
- Link-state routing protocol (κατάστασης συνδέσμων)

Distance-vector routing protocol (διανύσματος απόστασης)

Τα περισσότερα αν όχι όλα τα πρωτόκολλα αυτού του τύπου έχουν ορισμένα κοινά χαρακτηριστικά. Αρχικά οι περιοδικές ενημερώσεις του πλήρους πίνακα δρομολόγησης αποστέλλονται στους “γείτονες”, Ακόμη, πάσχουν από αργή σύγκλιση (έτσι ορίζεται η διαδικασία ανεύρεσης νέου ιδανικού μονοπατιού μεταξύ δύο πλευρών χωρίς βρόχους) και παρουσιάζουν ιδιαίτερη ευαισθησία σε βρόχους. Επιπλέον κάποια μορφή απόστασης χρησιμοποιείται για να υπολογισθεί η μετρική της διαδρομής και ο αλγόριθμος *Bellman-Ford* χρησιμοποιείται για την επιλογή της συντομότερης διαδρομής.

Ο αλγόριθμος αρχίζει με τον να διαφημίζει τα άμεσα συνδεδεμένα δίκτυα στους γείτονές του (για παράδειγμα, *RIP*-κάθε 30 δευτερόλεπτα, *IGRP*- κάθε 90 δευτερόλεπτα). Οι γείτονες με την σειρά τους θα προσθέσουν τις διαδρομές από αυτές τις ενημερώσεις στους δικούς τους πίνακες δρομολόγησης. Κάθε γείτονας εμπιστεύεται αυτές τις πληροφορίες πλήρως και προωθεί τον πλήρη πίνακά του σε κάθε άλλο γείτονά του. Έτσι οι δρομολογητές βασίζονται στο μέγιστο βαθμό στους γειτονικούς τους για τα δεδομένα δρομολόγησης, ένα πλάνο που είναι γνωστό ως routing by rumor.

Παρουσιάζονται αρκετά μειονεκτήματα στη συμπεριφορά αυτή. Επειδή οι πληροφορίες δρομολόγησης προωθούνται από γείτονα σε γείτονα μέσω περιοδικών ενημερώσεων, ο αλγόριθμος υποφέρει από αργή σύγκλιση. Αυτό σε συνδυασμό με την εμπιστοσύνη που δείχνει στις γειτονικές ενημερώσεις αυξάνουν την ευαισθησία σε βρόχους δρομολόγησης.

Οι αλγόριθμοι distance-vector χρησιμοποιούν μία μορφή απόστασης για να υπολογίσουν τη μετρική μιας διαδρομής. Το *RIP* χρησιμοποιεί hop-count, ενώ το *IGRP* χρησιμοποιεί ένα συνδυασμό εύρους ζώνης και καθυστέρησης. [12][19]

Link-state routing protocol (κατάστασης συνδέσμων)

Όταν γίνεται εφαρμογή αυτού του τύπου των αλγορίθμων, ο κάθε κόμβος χρησιμοποιεί ως αρχικά δεδομένα ένα χάρτη του δικτύου με την μορφή γράφου. Για την παραγωγή αυτού, κάθε κόμβος κατακλύζει ολόκληρο το δίκτυο με πληροφορίες σχετικά με το ποιους άλλους κόμβους μπορεί να συνδεθεί, στην συνέχεια κάθε κόμβος συγκεντρώνει όλες αυτές τις πληροφορίες και σχηματίζει έναν χάρτη. Με την χρήση του χάρτη αυτού, κάθε δρομολογητής αποφασίζει ανεξάρτητα το καλύτερο μονοπάτι από εκεί που βρίσκεται προς κάθε άλλο κόμβο.



Ο αλγόριθμος που χρησιμοποιείται για να επιλεγεί η βέλτιστη διαδρομή, ονομάζεται αλγόριθμος του Dijkstra. Είναι αποτελεσματικός εφόσον λειτουργεί, δημιουργώντας μια δομή δεδομένων, ένα δέντρο, με τον τρέχοντα κόμβο σαν ρίζα του δέντρου, που εμπεριέχει όλους τους εναπομείναντες κόμβους του δικτύου. Αρχίζει με ένα δέντρο που περιέχει μόνο τον εαυτό του. Μετά, έναν ένα κάθε φορά, από το σύνολο των κόμβων που δεν έχουν προστεθεί στο δέντρο, προσθέτει τον κόμβο που έχει το μικρότερο κόστος για να φτάσει έναν γειτονικό κόμβο ο οποίος ήδη υπάρχει στο δέντρο. Αυτό συνεχίζεται μέχρις ότου όλοι οι κόμβοι να βρίσκονται στο δέντρο.

Αυτό το δέντρο εξυπηρετεί στην κατασκευή του πίνακα δρομολόγησης του κάθε κόμβου, δείχνοντας το καλύτερο επόμενο βήμα (hop), για να φτάσει από τον εαυτό του σε οποιονδήποτε άλλο κόμβο στο δίκτυο.[12][19]

4.1 RIP (Routing Information Protocol)

Ένα πρωτόκολλο δρομολόγησης ενδο-αυτοδύναμου συστήματος χρησιμοποιείται ώστε να καθορίσει πως γίνεται η δρομολόγηση μέσα σ' ένα αυτόνομο σύστημα (AS). Τα πρωτόκολλα δρομολόγησης ενδο-AS είναι όπως αναφέρθηκε και πιο πάνω γνωστά ως πρωτόκολλα εσωτερικής πύλης (interior gateway protocols). Ιστορικά, δύο πρωτόκολλα δρομολόγησης έχουν χρησιμοποιηθεί εκτεταμένα για δρομολόγηση μέσα σε ένα αυτόνομο σύστημα στο Διαδίκτυο: Το RIP (πρωτόκολλο δρομολόγησης πληροφοριών, Routing Information Protocol) και το OSPF (ανοιχτό-πρώτα η βραχύτερη διαδρομή, Open Shortest Path First).

Το Πρωτόκολλο Δρομολόγησης Πληροφοριών (RIP) ήταν ένα από τα αρχικά πρωτόκολλα δρομολόγησης Διαδικτύου ενδο-AS και συνεχίζει να χρησιμοποιείται ευρέως σήμερα. Το RIP είναι ένα πρωτόκολλο διανύσματος απόστασης, που μετράει τα άλματα ως μετρική κόστους, δηλαδή κάθε ζεύξη έχει κόστος 1. Στο RIP το κόστος ορίζεται από τον δρομολογητή προέλευσης προς ένα υποδίκτυο προορισμού. Το RIP χρησιμοποιεί τον όρο hop (άλμα), που είναι ο αριθμός των υποδικτύων που διασχίζονται κατά μήκος της διαδρομής βραχύτερης διαδρομής από τον δρομολογητή προέλευσης μέχρι το υποδίκτυο προορισμού, περιλαμβανομένου και του υποδικτύου προορισμού. Το μέγιστο κόστος μιας διαδρομής περιορίζεται σε 15, περιορίζοντας έτσι την χρήση του RIP σε αυτόνομα συστήματα, τα οποία έχουν διάμετρο μικρότερη των 15 αλμάτων. Στο RIP ανταλλάσσονται ενημερώσεις δρομολόγησης ανάμεσα σε γειτονικούς κόμβους περίπου κάθε 30 δευτερόλεπτα, χρησιμοποιώντας ένα μήνυμα απόκρισης RIP (RIP response message).

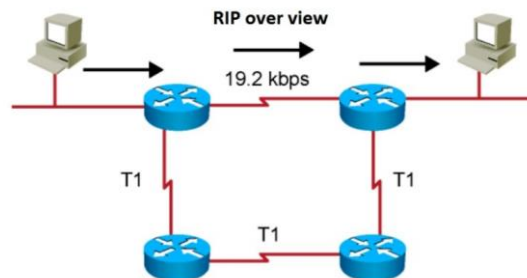
Το μήνυμα απόκρισης που στέλνεται από έναν δρομολογητή ή έναν υπολογιστή περιέχει μια λίστα μέχρι 25 υποδικτύων προορισμού μέσα στο AS, όπως και την απόσταση του αποστολέα προς κάθε ένα απ' αυτά τα υποδίκτυα. Τα μηνύματα απόκρισης είναι επίσης γνωστά ως διαφημίσεις RIP (RIP advertisements). Κάθε δρομολογητής διατηρεί ένα πίνακα RIP, που είναι γνωστός ως πίνακας δρομολόγησης (routing table). Ο πίνακας δρομολόγησης ενός δρομολογητή περιλαμβάνει και το διάνυσμα απόστασης του δρομολογητή και τον πίνακα προώθησης του δρομολογητή. Ο πίνακας δρομολόγησης έχει τρεις στήλες. Η πρώτη στήλη είναι για το υποδίκτυο προορισμού, η δεύτερη στήλη υποδηλώνει την ταυτότητα του επόμενου δρομολογητή κατά μήκος της βραχύτερης διαδρομής προς το υποδίκτυο προορισμού και η τρίτη στήλη υποδηλώνει τον αριθμό των αλμάτων (δηλ. τον αριθμό των υποδικτύων που πρέπει να διασχίσει, περιλαμβανομένου και του υποδικτύου προορισμού), για να φθάσει στο υποδίκτυο προορισμού, κατά μήκος της βραχύτερης διαδρομής.



Δεδομένου ότι οι δρομολογητές RIP ανταλλάσσουν διαφημίσεις ανά περίπου 30 δευτερόλεπτα. Εάν ένας δρομολογητής δεν ακούσει κάτι από τον γείτονα του τουλάχιστον μία φορά κάθε 180 δευτερόλεπτα, ο γείτονας θεωρείται ότι δεν είναι πλέον προσεγγίσιμος· αυτό σημαίνει ότι ο γείτονας έχει καταρρεύσει ή ότι η ζεύξη σύνδεσης έχει πέσει. Όταν συμβεί αυτό, το RIP τροποποιεί τον τοπικό πίνακα δρομολόγησης και μετά διαδίδει αυτήν την πληροφορία στέλνοντας διαφημίσεις στους γειτονικούς του δρομολογητές (αυτούς που μπορεί ακόμη να προσεγγίσει). Ένας δρομολογητής μπορεί επίσης να ζητήσει πληροφορίες για το κόστος του γείτονα του προς έναν δεδομένο προορισμό, χρησιμοποιώντας ένα μήνυμα αίτησης RIP.

Διαφορές RIPv1 και RIPv2:

- Η πρώτη έκδοση είναι distance-vector πρωτόκολλο, ενώ η δεύτερη είναι υβριδικό πρωτόκολλο.
- Στην πρώτη έκδοση οι ενημερώσεις αποστέλλονται σε όλα τα μέλη του υποδικτύου, ενώ στη δεύτερη μόνο σε όσους ενδιαφέρονται χωρίς να χρησιμοποιούνται άσκοπα πόροι του δικτύου.
- Η πρώτη έκδοση δεν υποστηρίζει VLSM (Variable Length Subnet Masking), ενώ η δεύτερη έκδοση υποστηρίζει.
- Η πρώτη έκδοση είναι classful πρωτόκολλο, δηλαδή δέχεται μόνο δίκτυα που δεν έχουν υποδίκτυα, κατ' επέκταση δεν στέλνει πληροφορίες για subnet mask με τις ενημερώσεις δρομολόγησης. Αντίθετα η δεύτερη είναι classless και εγκρίνει τη χρήση υποδικτύων.
- Η πρώτη έκδοση δεν υποστηρίζει την αυθεντικοποίηση των μηνυμάτων ενημέρωσης, σε αντίθεση με την δεύτερη η οποία την υποστηρίζει έτσι ώστε να υπάρξει επιβεβαίωση ότι οι ενημερώσεις προέρχονται από εγκεκριμένη πηγή. [19]



Εικόνα 4-1 Δίκτυο δρομολόγησης με βάση το RIP πρωτόκολλο

4.2 OSPF (Open Shortest Path First)

Το OSPF αποτελεί ένα link-state πρωτόκολλο που δημιουργήθηκε στα μέσα της δεκαετίας του 80' με σκοπό να αντιμετωπίσει τις ελλείψεις και κυρίως την αδυναμία χρήσης του RIP σε μεγάλα δίκτυα. Επειδή βασίζεται σε ανοιχτά πρότυπα γρήγορα έγινε ευρέως αποδεκτό και σήμερα χρησιμοποιείται σε πολλά εταιρικά δίκτυα. Τα πλεονεκτήματά του συνοπτικά είναι:

- Μπορεί να τρέξει σε οποιοδήποτε δρομολογητή καθώς βασίζεται σε ανοιχτά πρότυπα και η υλοποίησή του δεν δεσμεύεται από κάποιον κατασκευαστή.
- Παρέχει δρομολόγηση χωρίς loops, χρησιμοποιώντας τον αλγόριθμο SPF.
- Παρέχει γρήγορη σύγκλιση (convergence), κάνοντας χρήση προκαλούμενων (triggered) ενημερώσεων.



- Είναι ένα πρωτόκολλο classless, δηλαδή καταλαβαίνει μάσκες δικτύου και υποδικτύωση (subnetting), και συνεπώς επιτρέπει ιεραρχικό σχεδιασμό στη δρομολόγηση αξιοποιώντας τις τεχνικές VLSM και CIDR.

Βέβαια έχει και ορισμένα μειονεκτήματα. Αναφέρουμε συνοπτικά:

- Απαιτεί περισσότερη μνήμη καθώς διατηρεί πληροφορίες σε διάφορες βάσεις δεδομένων.
- Απαιτεί περισσότερη υπολογιστική ισχύ για να τρέξει τον αλγόριθμο SPF, ιδιαίτερα κατά την εκκίνηση της διεργασίας του OSPF.
- Είναι πολύπλοκο στην παραμετροποίηση και ακόμα περισσότερο στα μεγάλα δίκτυα όπου απαιτείται προσεκτικός σχεδιασμός για επιτυχημένη ιεραρχική δρομολόγηση.
- Η αποσφαλμάτωση του είναι επίσης δύσκολη διαδικασία.

OSPF Areas

Ένα από βασικά χαρακτηριστικά του OSPF είναι η ικανότητα του να φέρει εις πέρας τη δρομολόγηση μεγάλων δικτύων που εφαρμόζουν VLSM. Την ικανότητα του αυτή την οφείλει στην υποστήριξη της έννοιας Autonomous Systems.

OSPF Areas

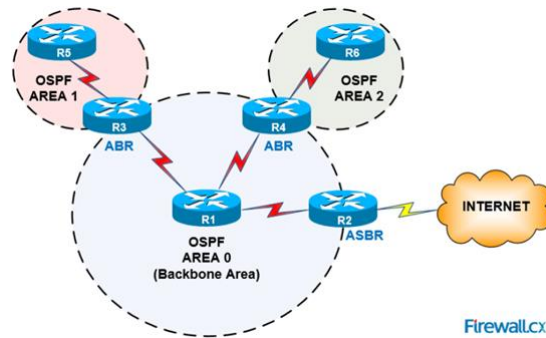
Ένα από βασικά χαρακτηριστικά του OSPF είναι η ικανότητα του να φέρει εις πέρας τη δρομολόγηση μεγάλων δικτύων που εφαρμόζουν VLSM. Την ικανότητα του αυτή την οφείλει στην υποστήριξη της έννοιας Autonomous Systems.

4.2.1 Autonomous Systems

Autonomous Systems είναι ένα σύνολο από δίκτυα που κάτω από κοινό διαχειριστικό έλεγχο. Σε κάθε AS αντιστοιχίζεται ένας μοναδικός αριθμός από 1 έως 65,535. Για τον χειρισμό της δρομολόγησης μέσα σε ένα Autonomous System χρησιμοποιούνται τα πρωτόκολλα δρομολόγησης IGP, ενώ για την ανταλλαγή πληροφορίας δρομολόγησης μεταξύ αυτόνομων συστημάτων χρησιμοποιούνται τα πρωτόκολλα EGP.

Μέσα σε ένα AS, τα areas παρέχουν ιεραρχική δρομολόγηση. Χρησιμοποιούνται για τον έλεγχο της ποσότητας της πληροφορίας δρομολόγησης που διακινείται απ' άκρη σ' άκρη στο δίκτυο. Αλλαγές τοπικής σημασίας που συμβαίνουν μέσα στα όρια ενός area είναι δυνατόν να περιοριστούν και να μην ανακοινώνονται στα υπόλοιπα areas.

Το OSPF εφαρμόζει ιεραρχία δύο επιπέδων: το area κορμού (backbone) και τα area εκτός κορμού. Τα areas ταυτοποιούνται με αριθμούς από 0 έως 65,535. Στο backbone area αντιστοιχίζεται πάντα το 0. Όλα τα areas πρέπει να συνδέονται στο backbone area και επικοινωνία μεταξύ των areas γίνεται διαμέσου του backbone.



Εικόνα 4-2 Δίκτυο που λειτουργεί με OSPF πρωτόκολλο

Πλεονεκτήματα Υιοθέτησης areas

- Λιγότερη πληροφορία στους πίνακες δρομολόγησης
- Τα τοπικά προβλήματα παραμένουν τοπικά και δεν επηρεάζουν την σταθερότητα του υπόλοιπου δικτύου.
- Το OSPF δύναται να αναπτυχθεί σε δίκτυα μεγαλύτερα από τα distance-vector πρωτόκολλα, όπως το RIP.

Link State Advertisements (LSAs)

Το link state είναι μια περιγραφή της διεπαφής και της σχέσης με τους γειτονικούς δρομολογητές (IP, interfaces, subnet mask). Οι βάσεις δεδομένων του OSPF συντηρούνται από την ανταλλαγή διαφημιστικών ανακοινώσεων γνωστές σαν Link State Advertisements (LSAs). Το LSA είναι ένα σετ δεδομένων που περιγράφουν την κατάσταση ενός δρομολογητή ή ενός δικτύου. Ουσιαστικά είναι διαφημίσεις των link-states ενός δρομολογητή.

Metric

Σε αντίθεση με το RIP που χρησιμοποιεί τον αριθμό των hops σαν μέτρο σύγκρισης διαδρομών, το OSPF κάνει χρήση του εύρους ζώνης (bandwidth) μιας σύνδεσης. Για την ακρίβεια χρησιμοποιεί τον όρο κόστος (cost) που είναι το αντίστροφο του εύρους ζώνης. Ο τύπος συσχέτισης των μεγεθών είναι :

$$\text{Κόστος} = 10^8 / \text{bandwidth}$$

Συνεπώς όσο μεγαλύτερη η ταχύτητα μεταφοράς δεδομένων μιας γραμμής, τόσο μικρότερο είναι το κόστος. Το συνολικό κόστος μιας διαδρομής προκύπτει από το άθροισμα των επιμέρους κοστών των φυσικών συνδέσεων που την αποτελούν. Χρησιμοποιώντας το κόστος μιας διαδρομής αντί για τον αριθμό των hops, το OSPF επιλέγει διαδρομές πιο έξυπνα. Μεταξύ δύο διαδρομών προς τον ίδιο προορισμό το OSPF επιλέγει αυτή με το μικρότερο κόστος. Σε περίπτωση διαδρομών με το ίδιο κόστος το OSPF θα μοιράσει τον φόρτο μεταξύ των διαδρομών (load balancing).

Router-ID

Κάθε δρομολογητής που συμμετέχει στο OSPF δίκτυο πρέπει να έχει μοναδικό ID για να ξεχωρίζει από τους υπόλοιπους. Η επιλογή του Router-ID γίνεται σύμφωνα με τα παρακάτω :



- Πρώτα εξετάζονται οι IP διευθύνσεις των loopback διεπαφών και επιλέγεται η πιο μεγάλη
- Εάν δεν βρεθεί loopback interface επιλέγεται η πιο μεγάλη IP διεύθυνση από τις ενεργές φυσικές διεπαφές.
- Σε περίπτωση που δεν έχουμε ενεργή διεπαφή δεν ξεκινά το OSPF

Η καλύτερη πρακτική είναι η χρήση των loopbacks interfaces γιατί είναι πάντα ενεργά και άρα το Router-ID πάντα το ίδιο.

Βάσεις Δεδομένων

Οι δρομολογητές διατηρούν δύο βάσεις δεδομένων :

- Βάση γειτνίασης (adjacency database) είναι η λίστα ουσιαστικά με όλους τους OSPF γείτονες, με τον όρο γείτονες αναφερόμαστε στην αμφίδρομη επικοινωνία μεταξύ τους.
- Βάση τοπολογίας (topology database) περιέχει όλους τους δρομολογητές, τις διαδρομές ,τους προορισμούς και τα κόστη που ανακοινώνουν οι προορισμοί.

4.2.2 Λειτουργία OSPF

Hello Protocol : Είναι ένα επιμέρους πρωτόκολλο του OSPF που χρησιμοποιείται στην εδραίωση και διατήρηση σχέσεων γειτνίασης. Σε δίκτυα broadcast προσφέρει την δυνατότητα ανακάλυψης νέων δρομολογητών δυναμικά. Εξασφαλίζει ότι η επικοινωνία μεταξύ γειτόνων είναι αμφίδρομη. Πακέτα Hello στέλνονται περιοδικά (κάθε 10 sec) από όλες τις διεπαφές που συμμετέχουν στο OSPF.

Σε τοπικά δίκτυα τύπου broadcast (π.χ. Ethernet), κάθε δρομολογητής διαφημίζει τον εαυτό του στέλνοντας περιοδικά πακέτα Hello στην multicast IP διεύθυνση 224.0.0.5 (all-OSPF-routers διεύθυνση). Έτσι ανακαλύπτονται νέοι γείτονες δυναμικά. Τα hello πακέτα περιέχουν πληροφορία για το ποιος είναι ο Designated Router(DR) και λίστα με τους γειτονικούς δρομολογητές από όπου έχει λάβει hello πακέτα πρόσφατα.

Neighbor Routers : Όταν ένας δρομολογητής ενεργοποιηθεί, η OSPF διεργασία θα αρχίσει να παράγει και να στέλνει από τις διεπαφές που έχουν οριστεί στην OSPF παραμετροποίηση Hello πακέτα. Τα Hello πακέτα, μεταξύ άλλων, μεταφέρουν τα ακόλουθα στοιχεία:

- Τον αριθμό του area,
- Τους ρυθμιστές χρόνου (timers) HelloInterval και DeadInterval
- Το OSPF συνθηματικό (εφόσον έχει ρυθμιστεί πιστοποίηση ταυτότητας)

Για να προχωρήσουν δύο δρομολογητές στο σχηματισμό σχέσης γειτνίασης (neighbors) πρέπει να συμφωνούν στα παραπάνω στοιχεία.

Οι OSPF δρομολογητές περνάνε από τα εξής στάδια μέχρι να γίνουν γείτονες, που συνολικά καλείται exchange process(διαδικασία ανταλλαγής).

- **Down state** : Ο δρομολογητής Rx δεν έχει ανταλλάξει OSPF μηνύματα με κανέναν άλλο δρομολογητή .
- **Init state**: Ο δρομολογητής Rx αρχίζει να στέλνει Hello packets στην IP multicast διεύθυνση 224.0.0.5. Ο δρομολογητής Ry τα λαμβάνει και προσθέτει τον Rx στην λίστα των γειτόνων (εφόσον συμφωνούν τα στοιχεία τους) ,με την ένδειξη ότι βρίσκεται σε κατάσταση Init. Ακόμη βέβαια έχουμε επικοινωνία μονόδρομη



- **Two-way state:** Ο δρομολογητής Rx λαμβάνει Hello packet από τον Ry στο οποίο εντοπίζει το δικό του Router-ID. Ο Rx προσθέτει τον Ry στην δική του λίστα γειτόνων. Και στις δύο λίστες οι δρομολογητές έχουν φτάσει σε κατάσταση Two-way. Εδώ έχουμε επικοινωνία αμφίδρομη και οι δρομολογητές θεωρούνται γείτονες.

Κάθε δρομολογητής που έχει το OSPF πρωτόκολλο ενεργοποιημένο τότε περιμένει να λάβει πακέτο Hello από τους γείτονες του κάθε HelloInterval δευτερόλεπτα. Ένα περάσει χρόνος ίσος με DeadInterval (πάνω από 40 δευτερόλεπτα) χωρίς να λάβει πακέτο Hello από κάποιο neighbor, τότε ο γείτονας θεωρείται “νεκρός” και αφαιρείται από την λίστα των γειτόνων. Το γεγονός αυτό το μαθαίνουν και οι υπόλοιποι γείτονες.

Designated Router και Backup Designated Router

Το επόμενο βήμα είναι η ανταλλαγή πληροφοριών δρομολόγησης μεταξύ των γειτόνων. Για κάθε τοπικό δίκτυο πολλαπλής πρόσβασης εκλέγεται ένας δρομολογητής σαν κεντρικό σημείο συναλλαγής. Ένας δεύτερος δρομολογητής εκλέγεται σαν αντικαταστάτης σε περίπτωση αστοχίας του πρώτου. Οι δρομολογητές αυτοί καλούνται Designated Router (DR) και Backup Designated Router (BDR), αντίστοιχα.

Η εκλογή των DR και BDR γίνεται βάση του αριθμού OSPF προτεραιότητας (priority), ο οποίος εξορισμού είναι 1 (τιμές 0-255). Σε περίπτωση ισοπαλίας, εξετάζεται το Router-ID των δρομολογητών και τα δύο μεγαλύτερα κερδίζουν τον διαγωνισμό. Εάν παρουσιαστεί πρόβλημα στην λειτουργία του DR, τότε ο BDR προβιβάζεται σε DR και κάποιος άλλος δρομολογητής εκλέγεται σαν BDR.[12][19]

4.3 Interior Gateway Routing Protocol (IGRP)

Το IGRP αποτελεί ένα distance vector πρωτόκολλο έχοντας βέβαια και χαρακτηριστικά και linkstate πρωτοκόλλων. Κύριο πλεονέκτημα του είναι ότι προϋποθέτει έναν συγκεκριμένο αριθμό αυτόνομου συστήματος (ASN) ο οποίος πρέπει να ισοδυναμεί σε όλους τους γειτονικούς δρομολογητές με τους οποίους θα ανταλλάξει πληροφορίες. Μία ενημέρωση με αριθμό αυτόνομου συστήματος διαφορετικό από το προβλεπόμενο παραλείπεται. Έτσι δεν έχουμε κανένα σφάλμα στη δρομολόγηση και οι πληροφορίες είναι ορθές.

Επιπλέον, ένα στοιχείο που καθιστά το IGRP αρκετά ευέλικτο είναι ο υπολογισμός πολλών διαδρομών-μονοπατιών (multipath) προς την ίδια κατεύθυνση. Βασικό πλεονέκτημα είναι ότι το φορτίο διαχωρίζεται σε διάφορα μονοπάτια και έτσι αποφορτίζεται ολόκληρη η κυκλοφορία του δικτύου αυξάνοντας την αξιοπιστία και τις επιδόσεις του.[39]

Βασικά γνωρίσματα του IGRP πρωτοκόλλου αποτελούν τα παρακάτω:

- Η δυνατότητα να χειρίζεται αξιοσημείωτα σύνθετες και πολύπλοκες τοπολογίες
- Η δυνατότητα να χωρίζεται το δίκτυο σε υποδίκτυα με διαφορετικά χαρακτηριστικά bandwidth και delay.
- Η δυνατότητα να χειρίζεται τεράστιες σε μέγεθος τοπολογίες



Εξορισμού το IGRP χρησιμοποιεί σαν μετρικές το Bandwidth και την καθυστέρηση. Μπορεί όμως να ρυθμιστεί έτσι ώστε να συνδυάζει και άλλες παραμέτρους προκειμένου να σχηματίσει μια σύνθετη μετρική η οποία είναι πολύ πιο ακριβής από το απλό hop count που χρησιμοποιεί το RIP. Αυτές οι παράμετροι είναι:

- Bandwidth (το εύρος ζώνης μιας διαδρομής)
- Καθυστέρηση (η συνολική καθυστέρηση κατά μήκος μιας διαδρομής)
- Φόρτος (ο φόρτος ενός συνδέσμου μετρούμενος σε bits/sec)
- Αξιοπιστία (η αξιοπιστία ενός συνδέσμου όπως καθορίζεται από την ανταλλαγή keepalives)

Το IGRP μπορεί να χρησιμοποιηθεί για τρεις τύπους διαδρομών:

- Εσωτερικές(Interior)
- Εντός Αυτόνομου Συστήματος(System)
- Εξωτερικές(Exterior)

Εσωτερικές Διαδρομές

Οι εσωτερικές διαδρομές είναι διαδρομές μεταξύ των υποδικτύων ενός δικτύου συνδεδεμένου σε ένα router interface. Εάν το δίκτυο που είναι συνδεδεμένο στο router δεν είναι υποδικτυωμένο, το IGRP δεν διαφημίζει εσωτερικές διαδρομές.

Διαδρομές Συστήματος

Τα system routes είναι διαδρομές εντός ενός αυτόνομου συστήματος. Το λειτουργικό των Cisco routers έχει τη δυνατότητα να εξάγει διαδρομές system από άμεσα συνδεδεμένα interfaces καθώς και να λάβει system routes που παρέχονται από άλλους IGRP routes ή access servers. Τα system routes δεν περιέχουν πληροφορία υποδικτύου.

Εξωτερικές διαδρομές

Εξωτερικές ονομάζονται οι διαδρομές προς δίκτυα εκτός του αυτόνομου συστήματος.

4.3.1 Χαρακτηριστικά του IGRP

Αποφυγή routing loops

Το IGRP υποστηρίζει τις ακόλουθες τεχνικές αποφυγής routing loops

- Holddowns
- Split horizons
- Poison reverse updates



Επίσης το IGRP υλοποιεί αρκετούς από τους timers που είδαμε και στο RIP:

- Update timer
- Invalid timer
- Holddown timer
- Flush timer

Ο *update timer* για το IGRP έχει default τιμή 90 sec. Η default τιμή για τον *invalid timer* είναι 3 φορές μεγαλύτερη του update(270 sec). Ο *holddown timer* έχει default τιμή $3 * \text{update timer} + 10 \text{ sec}$ (280 sec). Ο *flush timer* έχει default τιμή επταπλάσια του *update timer*(630 sec).

Επιτάχυνση της σύγκλισης

Για να επιτύχει καλύτερους χρόνους σύγκλισης, το IGRP υποστηρίζει triggered updates όταν ανιχνευθεί αλλαγή στο δίκτυο.

Classful Operation

Το IGRP είναι classful πρωτόκολλο και δεν υποστηρίζει VLSMs. Μάσκες μεταβλητού μεγέθους υποστηρίζονται με το Enhanced IGRP, το διάδοχο του IGRP, που όμως είναι υβριδικό πρωτόκολλο και ενσωματώνει στοιχεία από link-state αλγορίθμους.

Load Balancing

Το IGRP υποστηρίζει τόσο equal όσο και unequal path load balancing, δηλαδή μπορεί να εκτελέσει λειτουργίες εξισορρόπησης φόρτου τόσο μεταξύ συνδέσεων με ίδια μετρική, όσο και μεταξύ συνδέσεων με διαφορετικές μετρικές.

Ρύθμιση του IGRP

Το IGRP ρυθμίζεται ως εξής αν και θα δούμε λεπτομερώς την λειτουργία των πρωτοκόλλων στα παρακάτω κεφάλαια προσομοιώσεων.

```
Router(config)#router igrp autonomous_system_number
```

```
Router(config-router)#network network_address
```

Το IGRP γνωρίζει την έννοια του Autonomous System και απαιτεί τον ορισμό AS για να δρομολογήσει σωστά. Προκειμένου δυο routers να μπορέσουν να ανταλλάξουν routing updates βάσει του IGRP, θα πρέπει να ανήκουν στο ίδιο Autonomous System. Τα IGRP updates περιέχουν τον αριθμό AS του router που τα στέλνει. Όταν ένας router παραλάβει ένα update, εξετάζει τον αριθμό AS του update και τον συγκρίνει με το δικό του. Εάν δεν ταιριάζουν ο router απορρίπτει το update.[40]

4.4 Enhanced Interior Gateway Routing Protocol (EIGRP)

Το Enhanced Interior Gateway Routing Protocol (EIGRP) είναι ένα υβριδικό πρωτόκολλο κατασκευασμένο από την Cisco που συνδυάζει χαρακτηριστικά link-state και distance-vector



πρωτοκόλλου. Είναι βασισμένο στο IGRP με πολλές βελτιώσεις ενσωματωμένες. Οι βελτιώσεις αυτές έδωσαν στο EIGRP τα link-state στοιχεία που χρειαζόταν για να μπορεί να αναπτυχθεί σε μεγάλα εταιρικά δίκτυα.

Αρχικά θα γίνει μια σύντομη αντιπαράθεση των δύο EIGRP και IGRP και έπειτα θα αναλυθούν τα πιο αξιόλογα σημεία.

Ομοιότητες

- Και τα δύο πρωτόκολλα προσφέρουν εξισορρόπηση φορτίου(load balancing) ανάμεσα σε 6 διαφορετικές διαδρομές.
- Έχουν παρόμοια metric που βασίζονται στα ίδια μεγέθη.
- Η λειτουργία και των δύο πρωτοκόλλων βασίζεται στην ανταλλαγή διανυσμάτων απόστασης προς κάθε προορισμό. Δεν ανταλλάσσεται καμία πληροφορία για την τοπολογία του δικτύου. Δηλαδή, ακόμα και το EIGRP στην ουσία είναι πρωτόκολλο distance-vector, ενισχυμένο με κάποια χαρακτηριστικά/τεχνικές από πρωτόκολλα link-state.

Διαφορές

- Το EIGRP έχει γρηγορότερη σύγκλιση λόγω των προκαλούμενων(triggered) ενημερώσεων και της αποθήκευσης τοπικά των πινάκων δρομολόγησης των γειτόνων.
- Το EIGRP έχει μικρότερη επίδραση στους δικτυακούς πόρους, μιας και χρησιμοποιεί προσαυξητικές(incremental)ενημερώσεις.
- Μέρος της επικοινωνίας του EIGRP γίνεται multicast,ενώ στο IGRP όλη η επικοινωνία γίνεται broadcast.
- Οι δρομολογητές στο EIGRP σχηματίζουν σχέσεις γειτονίας μεταξύ τους μέσω μηχανισμού παρόμοιου με τον αντίστοιχο του OSPF.Οι IGRP δρομολογητές δεν συσχετίζονται μεταξύ τους.
- Στο EIGRP χρησιμοποιείται ο αλγόριθμος DUAL για την επιλογή διαδρομών χωρίς loops.
- Στο IGRP,ο μέγιστος αριθμός hops είναι 255. Στο EIGRP ο αριθμός είναι 224. Ο αριθμός είναι ικανός για να εξυπηρετήσει μεγάλα εταιρικά δίκτυα.
- Το EIGRP μπορεί να υποστηρίξει την δρομολόγηση πολλαπλών δρομολογούμενων(routed) πρωτοκόλλων.

Τα δύο πρωτόκολλα είναι συμβατά μεταξύ τους. Σε ένα Autonomous System με IGRP και EIGRP δρομολογητές, πληροφορίες δρομολόγησης ανταλλάσσονται αυτόματα μεταξύ τους. Στο παρασκήνιο, οι EIGRP δρομολογητές πραγματοποιούν μετατροπή των metrics, λόγω του διαφορετικού τους μήκους. Συγκεκριμένα, τα metrics των διαδρομών που λαμβάνουν από IGRP δρομολογητές πολλαπλασιάζονται με 256, ενώ τα metrics των διαδρομών που αποστέλλονται στους IGRP δρομολογητές διαιρούνται με 256.[41]



4.4.1 Δομικά στοιχεία EIGRP πρωτοκόλλου

Metric

Τα πρωτόκολλα EIGRP και IGRP χρησιμοποιούν την ίδια δομή στο metric. Τα μεγέθη που χρησιμοποιούν και τα δύο είναι το εύρος ζώνης (bandwidth), η καθυστέρηση (delay), η αξιοπιστία (reliability) και το MTU (Maximum Transfer Unit). Εξ' ορισμού, μόνο τα bandwidth και delay λαμβάνονται υπόψη στον υπολογισμό του metric. Η μόνη διαφορά είναι στο μήκος του metric: το IGRP έχει 24 bit metric, ενώ το EIGRP 32 bit.[42][43]

Ο μαθηματικός τύπος για τον υπολογισμό του metric στο IGRP είναι:

$$\text{metric} = \text{bandwidth} + \text{delay}$$

ενώ ο αντίστοιχος τύπος για τον υπολογισμό του metric στο EIGRP είναι:

$$\text{metric} = (\text{bandwidth} + \text{delay}) * 256$$

Περιλήψεις Διαδρομών

Αντίθετα με το IGRP, το EIGRP υποστηρίζει τόσο αυτόματες όσο και χειροκίνητες περιλήψεις διαδρομών (route summarization). Επειδή κατά βάθος είναι πρωτόκολλο distance-vector, το EIGRP κάνει αυτόματη σύνοψη των διαδρομών στα όρια των κλάσεων δικτύων A, B και C. Για πιο αποτελεσματική περίληψη πρέπει να γίνει απενεργοποίηση της αυτόματης περίληψης και να εισαχθεί χειροκίνητα η επιθυμητή.

Πολλαπλά Δρομολογούμενα Πρωτόκολλα

Το EIGRP υποστηρίζει πολλαπλά Layer 3 (δικτύου) πρωτόκολλα: εκτός του IP μπορεί να παρέχει υπηρεσίες δυναμικής δρομολόγησης στα IPX και AppleTalk. Είναι δυνατόν μάλιστα να ανταλλάσσει πληροφορίες δρομολόγησης και για τα τρία ταυτόχρονα. Σε δίκτυα υποστηρίζουν πολλαπλά Layer 3 πρωτόκολλα, το EIGRP είναι ιδανική επιλογή. Η δυνατότητα αυτή του EIGRP οφείλεται στο γεγονός ότι σαν πρωτόκολλο transport δεν χρησιμοποιεί το TCP, αλλά πρωτόκολλο της δικής του σουίτας, το EIGRP RTP.

Αλγόριθμος DUAL

Το EIGRP χρησιμοποιεί τον αλγόριθμο DUAL (Diffusing Update Algorithm) για να ενημερώσει τον πίνακα δρομολόγησης. Αυτός ο αλγόριθμος προσδίδει στο EIGRP το χαρακτηριστικό της γρήγορης σύγκλισης. Κύρια ευθύνη του είναι η επιλογή των καλύτερων διαδρομών (Successors και Feasible Successors) προς κάθε γνωστό προορισμό και για το σκοπό αυτό αντλεί στοιχεία από τον πίνακα γειτόνων και τον πίνακα τοπολογίας που θα δούμε πιο κάτω.

Ο αλγόριθμος αποθηκεύει στον πίνακα τοπολογίας τις πληροφορίες δρομολόγησης των γειτόνων. Εάν η πρωτεύουσα διαδρομή για κάποιο προορισμό αποτύχει ο DUAL συμβουλευέται τον πίνακα



τοπολογίας για εφεδρική διαδρομή και εάν υπάρχει την τοποθετεί στον πίνακα δρομολόγησης. Με αυτόν τον τρόπο αποφεύγει να μιλήσει με τους γειτονικούς EIGRP δρομολογητές.

Πίνακας Γειτόνων (Neighbor Table)

Κάθε δρομολογητής κρατάει πληροφορίες για την κατάσταση των γειτονικών δρομολογητών σε αυτόν τον πίνακα. Τα στοιχεία που καταγράφονται είναι η IP διεύθυνση, το interface και ο χρόνος HoldTime του γείτονα. Ο δρομολογητής διατηρεί διαφορετικό πίνακα γειτόνων για κάθε δρομολογούμενο πρωτόκολλο. Επίσης, ο πίνακας των γειτόνων περιλαμβάνει πληροφορίες απαραίτητες για την λειτουργία του EIGRP RTP, όπως οι σειριακοί αριθμοί που χρησιμοποιούνται την επιβεβαίωση λήψης των πακέτων. Στον πίνακα καταγράφεται ο τελευταίος σειριακός αριθμός από τα πακέτα κάθε γείτονα, καθώς και κάθε πακέτο που δεν έχει επιβεβαιωθεί ακόμη η παραλαβή του από τους γείτονες. Τα πακέτα που δεν έχουν επιβεβαιωθεί μπαίνουν σε ουρά για αναδιανομή. Τέλος για κάθε γείτονα, υπολογίζονται τα βέλτιστα διαστήματα επανεκπομπής (retransmission interval).

Πίνακας Τοπολογίας (Topology Table)

Ο πίνακας αυτός περιέχει τους προορισμούς που έχουν διαφημιστεί από όλους τους γείτονες. Κάθε καταχώρηση στον πίνακα αναφέρεται σε διαφορετικό προορισμό και περιλαμβάνει τους γείτονες που διαφήμισαν τον προορισμό. Μαζί με κάθε γείτονα αποθηκεύεται το διαφημιζόμενο metric ή *Reported Distance (RD)* στην ορολογία του *EIGRP*. Στην καταχώριση, επίσης, υποδεικνύεται το τελικό metric που θα χρησιμοποιηθεί για τον συγκεκριμένο προορισμό. Το συγκεκριμένο metric καλείται *Feasible Distance (FD)* και ισοδυναμεί με το μικρότερο αριθμό που προκύπτει από το άθροισμα του κάθε RD και του αντίστοιχου κόστους πρόσβασης στον γείτονα του RD. Ο γείτονας στον οποίο αντιστοιχεί το *FD* καλείται *Successor*. Το *FD* και ο *Successor* εισάγονται στον πίνακα δρομολόγησης, ενώ το *FD* είναι το *metric* που θα διαφημίζει ο δρομολογητής για τον εν λόγω προορισμό.

Σημαντικός κανόνας των distance vector protocols είναι ότι τα metrics που διαφημίζονται αντιστοιχούν σε διαδρομές που χρησιμοποιούν οι ίδιοι οι διαφημιστές-δρομολογητές και άρα βρίσκονται στον πίνακα δρομολόγησης τους.

Successors & Feasible Successors

Ο δρομολογητής *successor* όπως είπαμε και πιο πάνω είναι η διαδρομή που επιλέχθηκε από τον DUAL σαν η καλύτερη προς δεδομένο προορισμό. Για κάποιον προορισμό υπάρχει πιθανότητα να υπάρχουν μέχρι και τέσσερις successors και εισάγονται όλες στον πίνακα δρομολόγησης.

Οι *successors* αντίθετα με τους *feasible successors* που θεωρούνται εφεδρικοί είναι πρωτεύων δρομολογητές. Οι *feasible successors* δεν εισάγονται στον πίνακα δρομολόγησης, αλλά μένουν στον πίνακα τοπολογίας. Για να αναδειχθεί κάποιος γείτονας σαν *feasible successor* πρέπει να ικανοποιείται η ακόλουθη διατύπωση: Ο δρομολογητής που μας λέει ότι είναι πιο κοντά στον προορισμό από ότι εμείς προάγεται σε *feasible successor*.

Κατάσταση Προορισμών

Ένας προορισμός μπορεί να βρίσκεται σε κατάσταση passive ή active.



Passive κατάσταση: Ονομάζεται η κατάσταση όπου ο δρομολογητής δεν εκτελεί υπολογισμό του successor. Σε περίπτωση βέβαια που αποτύχει ο successor υπάρχει και ο feasible successor που μπαίνει στη θέση του κύριου successor και δεν χρειάζεται να γίνει εκ νέου υπολογισμός.

Active κατάσταση: Ένας προορισμός θεωρείται ότι βρίσκεται σε αυτήν την κατάσταση όταν ο δρομολογητής εκτελεί υπολογισμό για νέο successor, επειδή δεν υπάρχει feasible successor (δηλαδή δεν υπάρχει εφεδρική διαδρομή διαθέσιμη στον πίνακα τοπολογίας). Η διαδικασία του υπολογισμού ξεκινάει με την αποστολή πακέτων Query σε όλους τους γείτονες. Οι γείτονες πρέπει είτε να απαντήσουν με πληροφορίες για τον εν λόγω προορισμό, είτε να ενημερώσουν ότι δεν έχουν γνώση. Αφού εκτιμηθούν οι απαντήσεις και εφόσον βρεθεί νέος successor τότε ο προορισμός ξαναγυρίζει σε κατάσταση passive.

Είδη μηνυμάτων που χρησιμοποιεί το EIGRP

- Hello: Τα μηνύματα αυτά αποστέλλονται multicast και σκοπός τους είναι η ανακάλυψη νέων γειτόνων και η διατήρηση επαφής με τους υπάρχοντες. Η λήψη τους δεν χρειάζεται επιβεβαίωση.
- Acknowledgement: Ένα άδειο πακέτο Hello αποτελεί ένα πακέτο Acknowledgement(Ack). Τα πακέτα Ack στέλνονται unicast και σκοπός τους είναι η επιβεβαίωση λήψης των άλλων τύπων πακέτων. Δηλαδή, η λήψη των πακέτων update, query και reply, υποχρεώνει το δρομολογητή να απαντήσει με ένα ack στον αποστολέα.
- Updates: Χρησιμοποιούνται για την μεταφορά των περιεχομένων του πίνακα τοπολογίας από δρομολογητή σε δρομολογητή. Τα πακέτα update στέλνονται προς ένα γείτονα unicast. Στην περίπτωση προκαλούμενων ενημερώσεων, τα update στέλνονται multicast (στην ίδια διεύθυνση όπως και τα Hello).
- Query: Όταν η κατάσταση ενός προορισμού υπεισέρχεται σε κατάσταση active τότε ο δρομολογητής στέλνει πακέτα query στην multicast διεύθυνση του EIGRP. Εάν σε απάντηση ενός query στείλουμε κάποιο άλλο query τότε το δεύτερο στέλνεται unicast.
- Reply: Ο δρομολογητής που θα λάβει πακέτο query πρέπει να ανταποκριθεί ακόμη και αν δεν έχει να προτείνει κάτι, με ένα πακέτο reply απ' ευθείας στον αποστολέα του query (unicast).

Ετικέτες σε διαδρομές

Το EIGRP τοποθετεί "ετικέτες" στις διαδρομές που μαθαίνει από το IGRP ή από οποιαδήποτε άλλη πηγή, μαρκάροντας τις σαν εξωτερικές επειδή δεν προέρχονται από EIGRP δρομολογητή. Το IGRP δεν καταλαβαίνει αυτούς δεν καταλαβαίνει αυτούς τους διαχωρισμούς, για αυτό το λόγο και στις διαδρομές που διαφημίζονται με IGRP δρομολογητές αφαιρούνται οι ετικέτες.



4.4.2 Λειτουργία του EIGRP

Το EIGRP χρησιμοποιεί πακέτα Hello για να ανακαλύψει νέους δρομολογητές και να διατηρήσει σχέσεις γειτνίασης με αυτούς. Σε δίκτυα broadcast, point-to-point και point-to-multipoint τα πακέτα Hello αποστέλλονται κάθε 5 δευτερόλεπτα (hello interval). Σε συνδέσεις με ταχύτητες μικρότερες από T1/E1 τα Hello στέλνονται κάθε 60 δευτερόλεπτα. Τα πακέτα Hello στέλνονται στην multicast IP διεύθυνση 224.0.0.10.

Στα πακέτα Hello περιέχεται η παράμετρος HoldTime. Η τιμή της ορίζει τον χρόνο μέσα στον οποίο ένας δρομολογητής πρέπει να λάβει ένα πακέτο Hello από ένα γείτονα του, ώστε ο γείτονας να παραμείνει ενεργός στον πίνακα γειτόνων του δρομολογητή. Είναι αντίστοιχο με τον χρόνο DeadInterval του OSPF.

Για να συμβεί EIGRP γειτνίαση πρέπει να ταιριάζουν τα παρακάτω στοιχεία:

- Ο αριθμός του αυτόνομου συστήματος (AS)
- Οι μεταβλητές K των metric.

≠ Αντίθετα τώρα με το OSPF χρειάζεται να ταιριάζουν οι ρυθμιστές χρόνου (timers) για να έχουμε επιτυχή γειτνίαση. Η διαδικασία που ακολουθείται για να επιτευχθεί η γειτνίαση είναι η ακόλουθη:

1. Έστω ο δρομολογητής R1 που χρησιμοποιεί το πρωτόκολλο EIGRP. Ο R1 στέλνει πακέτα Hello από όλες τις διεπαφές που συμμετέχουν στον EIGRP.
2. Ο R2 λαμβάνει πακέτο Hello από τον R1. Εάν υπάρχει ταύτιση στον AS ο R2 θα απαντήσει με πακέτο Hello ακολουθούμενο από το πακέτο update. Το τελευταίο περιέχει όλες τις διαδρομές που γνωρίζει ο R2 και μεταδίδεται unicast στον R1.
3. Ο δρομολογητής R1 θα απαντήσει με πακέτο επιβεβαίωσης (acknowledgement), γνωστοποιώντας έτσι στον R2 ότι έλαβε το πακέτο του R2. Μετά θα στείλει στον R2 το δικό του πακέτο update.
4. Ο δρομολογητής R2 θα απαντήσει με πακέτο acknowledgement. Σε αυτό το σημείο οι δύο routers έχουν συγχρονιστεί.

≠ Σε αντίθεση με το OSPF η ανταλλαγή πληροφοριών όσον αφορά την δρομολόγηση δεν πραγματοποιείται με κάποιον default δρομολογητή αλλά με οποιονδήποτε. Έτσι κάθε neighbor router είναι και adjacent.[42][43]

4.4.2.1 Μετρικές Πρωτοκόλλων

| Μετρική | Πρωτόκολλο | Περιγραφή |
|-------------------------|-------------|------------------------------|
| Εύρος ζώνης (bandwidth) | EIGRP, IGRP | Χωρητικότητα γραμμής σε Kbps |



| | | |
|---------------------------------|------------|--|
| | | |
| Κόστος (κόστος) | OSPF | Παράγωγο μέγεθος βασισμένο στο bandwidth της γραμμής |
| Καθυστέρηση(delay) | EIGRP,IGRP | Χρόνος που απαιτείται ώστε να φτάσει ένα πακέτο στον προορισμό του |
| Αριθμός βημάτων (hop count) | RIP(v1,v2) | Αριθμός routers που πρέπει να περάσει το πακέτο μέχρι να φτάσει στον προορισμό |
| MTU (Maximun Transmission Unit) | IGRP,EIGRP | Η διαδρομή που υποστηρίζει τα μεγαλύτερα μεγέθη πλαισίων |
| Φόρτος (load) | IGRP,EIGRP | Η διαδρομή με το μικρότερο βαθμό χρήσης |
| Αξιοπιστία (reliability) | EIGRP,IGRP | Η διαδρομή με το μικρότερο αριθμό λαθών η το μικρότερο χρόνο εκτός λειτουργίας |

Πίνακας 4-1 Μετρικές πρωτοκόλλων

4.5 Exterior Gateway Protocols (EGP)

Το Exterior Gateway Protocol (EGP) αποτελεί ένα πρωτόκολλο ανταλλαγής δεδομένων δρομολόγησης ανάμεσα από δύο γειτονικούς gateway hosts σε ένα autonomous system network. Η χρήση του EGP γίνεται ανάμεσα σε hosts στο διαδίκτυο για να ανταλλάξουν πληροφορίες των πινάκων δρομολόγησης. Βλέποντας τον πίνακα δρομολόγησης βλέπουμε ότι περιλαμβάνει μια λίστα γνωστών δρομολογητών, τις IP διευθύνσεις στις οποίες έχουν πρόσβαση, και έναν μετρητή για το κόστος ο οποίος είναι συνδεδεμένος με κάθε μονοπάτι σε κάθε router έτσι ώστε να γίνεται επιλογή της βέλτιστης διαδρομής που υπάρχει. Ο καθένας router επικοινωνεί με τον γειτονικό του σε διαστήματα από 120 έως 480 δευτερολέπτων και ο γειτονικό δρομολογητής απαντά αποστέλλοντας τον πλήρη πίνακα δρομολόγησης του.

Ο λόγος για τον οποίο δημιουργήθηκε ήταν η ενεργοποίηση αυτόνομων συστημάτων τα οποία θα μεταφέρουν την κίνηση την οποία ένα άλλο αυτόνομο σύστημα είχε φτιάξει και είχε ένα άλλο αυτόνομο σύστημα ως τέρμα, και ο τελικός χρήστης θα έβλεπε τη συνένωση αυτών των αυτόνομων συστημάτων σαν ένα αδιαίρετο Διαδίκτυο. Η διαδρομή που ένα datagram ακολουθεί διαμέσων του Διαδικτύου, όλων των αυτόνομων συστημάτων που διέρχεται, έπρεπε να είναι ευδιάκριτος στον τελικό χρήστη.[12]

Στην δική μας μελέτη έγινε χρήση του πρωτοκόλλου BGP που ανήκει στην κατηγορία των EGP πρωτοκόλλων που θα δούμε την ανάλυση του στην παρακάτω ενότητα και την υλοποίηση του στα επόμενα κεφάλαια.



4.5.1 Border Gateway Protocols (BGP)

4.5.1.1 Δομικά Στοιχεία του BGP

Πριν δώσουμε μια ανασκόπηση του BGP, προσδιορίζουμε την ορολογία που μελετάται μέσα στο BGP:

- **BGP speaker** : Ένας μηχανισμός που είναι ενεργός και στέλνει μηνύματα keep alive κάθε 60 sec.
- **BGP neighbors**: Ένα ζεύγος από speakers που εναλλάσσονται πληροφορίες σχετικά με τη δρομολόγηση. Υπάρχουν οι *εσωτερικοί* γείτονες που βρίσκονται στο ίδιο AS και πρέπει να παρουσιάζουν μια ομοιόμορφη παρουσία του AS προς τους εξωτερικούς γείτονες. Από την άλλη οι *εξωτερικοί* γείτονες είναι ένας ζεύγος από speakers διαφορετικού αυτόνομου συστήματος.
- **BGP session**: Είναι ένα session που κάνει χρήση TCP πρωτοκόλλου ανάμεσα από BGP neighbors που εναλλάσσονται δεδομένα δρομολόγησης με τη χρήση του BGP στην πόρτα 179. Οι γείτονες διαχειρίζονται το state του session με την αποστολή *keep alive* μηνυμάτων ανά 30 sec.
- **AS border router (ASBR)**: Δρομολογητής που είναι συνδεδεμένος σε πολλά AS. Υπάρχουν οι *εσωτερικοί*, όπου ο δρομολογητής βρίσκεται στο επόμενο άλμα στο ίδιο AS με τον BGP speaker. Αντίθετα *εξωτερικοί* βρίσκονται στο επόμενο άλμα διαφορετικού όμως AS απ' ότι ο BGP speaker.[12][19]

4.5.1.2 Λειτουργία του BGP

Το Πρωτόκολλο Συνοριακής Πύλης Δικτύου (BGP) είναι μια μορφή πρωτοκόλλου διανυσμάτων απόστασης, αλλά είναι αρκετά διαφορετικό από τα πρωτόκολλα ενδοπεριοχής δρομολόγησης με διανύσματα απόστασης(όπως το RIP). Μια διαφορά είναι ότι, αντί να διατηρεί μόνο το κόστος προς κάθε προορισμό, κάθε δρομολογητής BGP διατηρεί και την διαδρομή που χρησιμοποιείται. Αυτή η προσέγγιση ονομάζεται πρωτόκολλο διανύσματος διαδρομής(path vector protocol). Η διαδρομή αυτή αποτελείται από τον δρομολογητή επόμενου άλματος (ο οποίος μπορεί να βρίσκεται στο άλλο άκρο του ISP και όχι κάπου κοντά) και την ακολουθία αυτόνομων συστημάτων, ή διαδρομή AS(AS path), το οποίο έχει ακολουθήσει το δρομολόγιο. Τέλος τα ζευγάρια δρομολογητών BGP επικοινωνούν μεταξύ τους εγκαθιδρύοντας συνδέσεις TCP. Έτσι υπάρχει αξιόπιστη επικοινωνία και επίσης αποκρύπτονται όλες οι λεπτομέρειες των δικτύων από τα οποία περνά η επικοινωνία.

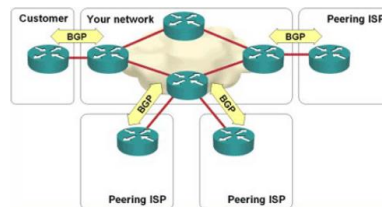
Το BGP είναι από τα κυριότερα πρωτόκολλα δρομολόγησης και ανήκει στα EGPs (Exterior Gateway Protocols). Έχει τη δυνατότητα να διατηρεί και να παρακολουθεί τα δίκτυα IP που προσφέρουν πρόσβαση δικτύου σε αυτόνομα συστήματα (το σύνολο των IP που σκιαγραφούν τη διαδικασία δρομολόγησης στο Internet). Ακόμη είναι εκείνο που αντικατέστησε το EGP (Exterior Gateway Protocol), η χρήση του οποίου έχει παύσει εντελώς πλέον.

Στο BGP οι δρομολογητές ανταλλάσσουν πληροφορίες προσβασιμότητας δικτύου με τους κοντινότερους γείτονές τους. Δηλαδή, οι routers στέλνουν μεταξύ τους τις ομάδες διευθύνσεων



στις οποίες έχουν πρόσβαση καθώς και τη διεύθυνση του επόμενου άλματος στην οποία πρέπει να σταλούν τα δεδομένα για να φτάσουν σε αυτές τις διευθύνσεις. Αυτό αντιτίθεται στη λειτουργία των link-state IGP, στα οποία ανταλλάσσουν πληροφορίες τοπολογίας και υπολογίζουν τοπικά τις δικές τους διαδρομές, καθώς στα EGP οι δρομολογητές ανταλλάσσουν διαδρομές μεταξύ τους.

Όταν δημιουργηθεί μία σύνδεση με μία συσκευή δικτύου, ο BGP router στέλνει όλες τις διαδρομές του τοπικού BGP πίνακα δρομολόγησης στη συσκευή αυτή με μηνύματα ενημερώσεων. Η συσκευή αυτή χρησιμοποιεί τα δεδομένα που έλαβε για να τοποθετήσει νέες διαδρομές στον τοπικό πίνακά της. Σε περίπτωση που μάθει περισσότερες από μία διαδρομές για τον ίδιο προορισμό, εκτελεί μια διαδικασία για να αποφασίσει ποια είναι η προτιμότερη, να την τοποθετήσει στον πίνακά της και να τη διαφημίσει στις άλλες BGP συσκευές. Οι διαδρομές του BGP πίνακα συνδυάζονται με διαδρομές που γνωρίζουμε από άλλα πρωτόκολλα έτσι ώστε να παραχθεί ο πλήρης routing table.



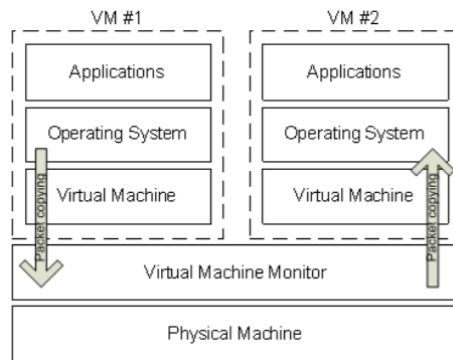
Εικόνα 4-3 BGP πρωτόκολλο δρομολόγησης

Το πρωτόκολλο αφήνει την παραμετροποίηση των διαδρομών πριν διανεμηθούν σε άλλες συσκευές. Επίσης χρησιμοποιεί χρονόμετρα για να αποφύγει τη συνεχή διαφήμιση κάποιας διαρκώς μεταβαλλόμενης διαδρομής. Τέλος οι ανταλλαγές πληροφοριών BGP μπορούν να αυθεντικοποιηθούν.[19]

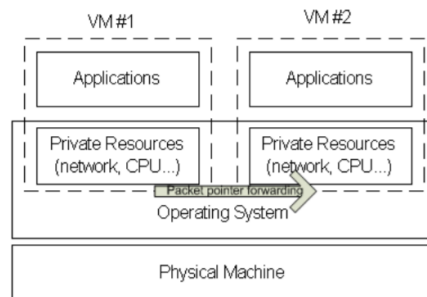


5 Εισαγωγή στην Εικονικοποίηση Δικτύων

Ο όρος εικονικοποίηση δικτύου αναφέρεται στην αφαιρετική προσέγγιση ενός πραγματικού δικτύου δίνοντας τη δυνατότητα υποστήριξης σύνθετων λογικών δικτύων τα οποία μοιράζονται το ίδιο υλικό. Πιο συγκεκριμένα έχουμε τη δυνατότητα να εργαζόμαστε με δρομολογητές(routers), μεταγωγείς(switches) και υπολογιστές(hosts) οι οποίοι τρέχουν πάνω στην ίδια υποδομή. Για να το επιτύχουμε αυτό θα πρέπει αρχικά να εικονικοποιήσουμε τους κόμβους του δικτύου. Μία ευρέως γνωστή τεχνική που χρησιμοποιείται είναι η χρήση εικονικών μηχανημάτων(virtual machines) η οποία δεν είναι ιδιαίτερα αποδοτική. Μία ακόμη τεχνική την οποία θα μελετήσουμε είναι η ελαφριά εικονικοποίηση με χρήση Linux host. Η κύρια ιδέα είναι ότι οι κόμβοι του δικτύου κατέχουν ξεχωριστή οπτική μέσω της απομόνωσης.



Εικόνα 5-1 Κλασική Προσέγγιση Εικονικοποίησης



Εικόνα 5-2 Προσέγγιση Ελαφριάς Εικονικοποίησης

Στην κλασική προσέγγιση εικονικοποίησης τα πακέτα πηγαίνουν από το VM1 στο VM2 αντιγράφονται από το χώρο του χρήστη στο χώρο του πυρήνα και το αντίστροφο. Αντίθετα στην ελαφριά εικονικοποίηση ο δείκτης του πακέτου μεταβιβάζεται από το VM1 στο VM2.

Στην επιστήμη των δικτύων υπολογιστών μελετούμε πακέτα που μεταδίδονται από τερματικό σε τερματικό και χρησιμεύουν σε κάτι απλό όπως η επικύρωση ότι ένας κόμβος λειτουργεί με επιτυχία μέχρι και στο συντονισμό πολλαπλών κόμβων για τη ταχύτερη μετάδοση πληροφορίας στον προορισμό της. Συνεπώς η παρατήρηση αυτών των πακέτων μας δίνει μία



πιο ολοκληρωμένη εικόνα του τρόπου λειτουργίας αυτών των δικτύων. Οι πλατφόρμες που συνήθως τίθενται σε λειτουργία για την επίτευξη αυτού του σκοπού είναι με χρήση υλικού (hardware) για την πλήρη δημιουργία ενός δικτύου, με χρήση κάποιου προσομοιωτή είτε με χρήση κάποιου εξομοιωτή.

5.1 Χρήση υλικού (hardware)

Για την δημιουργία ενός δικτύου υπολογιστών χρησιμοποιούμε αληθινά μηχανήματα και καλώδια για τις συνδέσεις τις οποίες θα υλοποιήσουμε αφού μελετήσουμε πρώτα οποιαδήποτε παράμετρο αφορά το δίκτυο μας, στη συνέχεια να προβούμε σε εγκατάσταση κάποιου ανιχνευτή-αναλυτή πακέτων π.χ. Wireshark, σε όλα τα μηχανήματα έτσι ώστε να παρακολουθούμε και να καταγράφουμε την κίνηση των πακέτων. Η συγκεκριμένη μέθοδος χαρακτηρίζεται από την ακρίβεια και την ταχύτητά της. Σημαντικό μειονέκτημα μπορεί να θεωρηθεί το γεγονός ότι οποιοσδήποτε επιθυμεί να μελετήσει το δίκτυο είναι αναγκαίο να μεταβεί στη τοποθεσία που τα μηχανήματα είναι εγκατεστημένα. Βέβαια όμως η χρήση της συγκεκριμένης μεθόδου δεν είναι καθόλου οικονομική διότι απαιτείται η αγορά συγκεκριμένου εξοπλισμού(υπολογιστές, μεταγωγείς, δρομολογητές) καθώς και η συντήρηση του. Ως αποτέλεσμα της παραπάνω αναφοράς είναι η αδυναμία στην κλιμάκωση μιας και το πλήθος των κόμβων ενός δικτύου μειώνεται σημαντικά αναλογιζόμενοι το κόστος. Εν τέλει οποιαδήποτε αλλαγή χρειάζεται να συμβεί ανάλογα με τη τρέχουσα τοπολογία που μελετάται, χρειάζεται αναδιάταξη στα καλώδια ή ακόμη και αλλαγή ενός σημαντικού μέρους του εξοπλισμού.

5.2 Χρήση προσομοίωσης σε υπολογιστικό σύστημα

Για να εξάγουμε κάποια αποτελέσματα όσον αφορά την απόδοση και την κίνηση ενός δικτύου χρησιμοποιούμε ορισμένα μαθηματικά μοντέλα. Η συγκριμένη μέθοδος δεν έχει μεγάλο κόστος, είναι εύκολη στη χρήση από τρίτους καθώς το πρόγραμμα εκτελείται και αρκετά πιο γρήγορη τις περισσότερες φορές σε σύγκριση με τον πραγματικό χρόνο. Ορισμένα προβλήματα που εμφανίζονται αφορούν την ελαστικότητα της εκάστοτε τοπολογίας καθώς οποιαδήποτε αλλαγή επιθυμεί ο χρήστης να κάνει απαιτεί συνήθως την αλλαγή του κώδικα και κατ' επέκταση του μοντέλου που γίνεται χρήση. Επιπλέον οι συγκεκριμένες εφαρμογές δεν υποστηρίζουν κώδικα λειτουργικού συστήματος πράγμα που τις κάνει να μην είναι ελαστικές και έτσι δεν δίνεται η δυνατότητα στον χρήστη να προβεί σε αλλαγές στον πραγματικό χρόνο.

5.3 Χρήση εξομοίωσης σε υπολογιστικό σύστημα

Με την έννοια αυτή αναφερόμαστε στην υλοποίηση προγραμμάτων δημιουργίας εικονικών κλώνων αληθινών μηχανημάτων με παρόμοια συμπεριφορά, τα οποία έχουν τη δυνατότητα να υποστηρίζουν κώδικα λειτουργικού συστήματος. Επιπρόσθετα εάν ο χρήστης θελήσει να προβεί σε κάποια αλλαγή στο σύστημα το μόνο που έχει να κάνει είναι να επεκτείνει τον υπάρχοντα κώδικα. Επιπλέον το υλικό που δημιουργείται από τις εικονικές μηχανές έχει μηδαμινό κόστος μιας και η μονάδα μέτρησης του κόστους είναι το ποσοστό μνήμης που είναι απαραίτητο για την εύρυθμη λειτουργία του κάθε μηχανήματος. Επίσης είναι αρκετά εύκολο να συμπειστούν σε μία εικόνα σκληρού δίσκου, ο κώδικας της μηχανής, οι ρυθμίσεις και τα δεδομένα που είναι απαραίτητα και να αποθηκευτούν σε κάποιο νέφος (cloud storage) στο διαδίκτυο έτσι ώστε να χρησιμοποιηθούν έπειτα από λήψη από άλλους χρήστες. Ως επακόλουθο είναι ανοιχτή σε οποιαδήποτε επέκταση πράγμα που σημαίνει ότι έχουμε τη δυνατότητα να πραγματοποιήσουμε αλλαγές σε πραγματικό χρόνο με το μόνο μειονέκτημα την ταχύτητα και τον συγχρονισμό συγκριτικά με τον αληθινό εξοπλισμό.



Συμπερασματικά στην περίπτωση που παρουσιάσουμε ενδιαφέρον στη διακίνηση πακέτων που μεταφέρονται μέσα στο δίκτυο και όχι τόσο στην απαρίθμηση των παραμέτρων του και στην υλοποίηση γραφικών παραστάσεων η χρήση εξομοιωτών επιφέρει τα καλύτερα αποτελέσματα.[44][45]

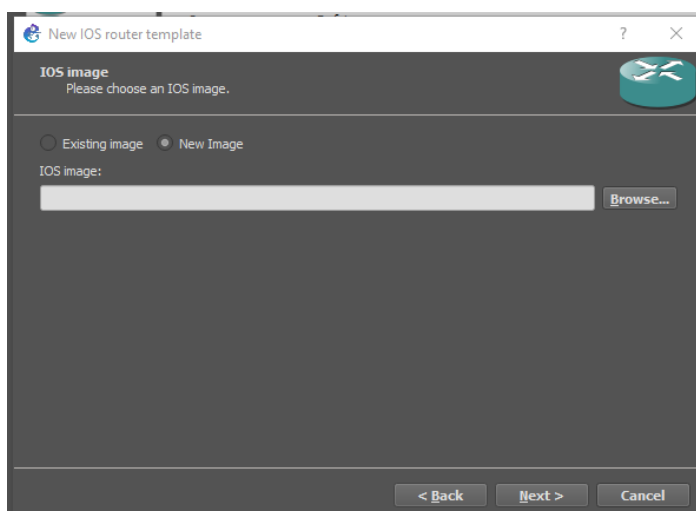


6 Ενασχόληση με το Περιβάλλον του δικτυακού προσομοιωτή GNS3

Έχοντας κατεβάσει την νεότερη έκδοση του προσομοιωτή GNS3 για το λειτουργικό Windows 10 που χρησιμοποιούμε από [εδώ](#) και έχοντας κάνει λογαριασμό με το ακαδημαϊκό μας email προχωρούμε στην εγκατάσταση ή οποία για πρακτικούς λόγους δεν παρουσιάζεται μιας και ακολουθεί τυπική διαδικασία εγκατάστασης στα Windows.[57]

6.1 Διαδικασία δημιουργίας-φόρτωσης ενός GNS3 project

Αφότου γίνει επιτυχής εγκατάσταση του GNS3 για να χρησιμοποιήσουμε το εργαλείο θα πρέπει να φορτώσουμε τα κατάλληλα images για τα IOS Routers που υπάρχουνε στο χώρο που εργαστήκαμε. Έχοντας ανοίξει το πρόγραμμα πηγαίνουμε [Edit->Preferences->IOS Routers](#) πατάμε [New](#) και πατώντας την παρακάτω επιλογή

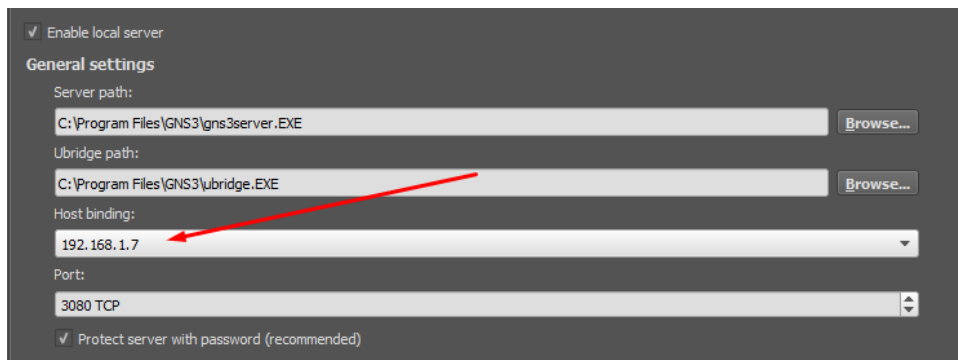


Εικόνα 6-1 Προσθήκη router image

Κάνουμε Browse βρίσκουμε το φάκελο όπου έχουμε τοποθετήσει τα [GNS3 IOS](#) και φορτώνουμε ένα-ένα τα images για να μπορούμε να δούμε και να παραμετροποιήσουμε τους δρομολογητές.

Στη συνέχεια για να συνδεθούμε αν επιθυμούμε με απομακρυσμένη σύνδεση(VPN) ενεργοποιούμε τη σύνδεση VPN στο Aegean domain και συνδεόμαστε στην IP των CCSL labs με τον παρακάτω τρόπο

[Edit->Preferences->Server](#) και στο Host Binding αφότου έχουμε συνδεθεί απομακρυσμένα επιλέγουμε IP της μορφής αυτής



Εικόνα 6-2 Εγκαθίδρυση σύνδεσης

Και πατάμε Apply->OK

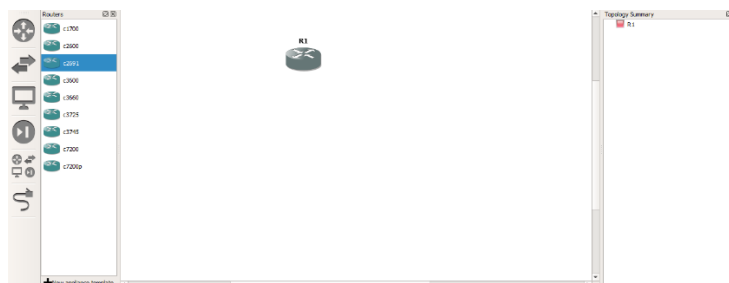
Αν θέλουμε να συνδεθούμε τοπικά βρίσκουμε την IPv4 address και την επιλέγουμε στο Host binding.

Στη συνέχεια για να ανοίξουμε ένα ήδη υπάρχον πρότζεκτ πηγαίνουμε **File->Open Project** και κάνουμε browse το πρότζεκτ που θέλουμε να ανοίξουμε επιλέγοντας αρχείο τύπου GNS3. Δεν θα αντιμετωπίσουμε κάποιο πρόβλημα με την αποθήκευση καθώς από την έκδοση 2.0 και έπειτα διατίθεται η λειτουργία του autosave.

Για την δημιουργία καινούριου πρότζεκτ πηγαίνουμε **File->Open new Blank Project** δίνουμε όνομα και παρατηρούμε το path που αποθηκεύεται by default.

6.2 Βασική συνδεσμολογία στο περιβάλλον του GNS3

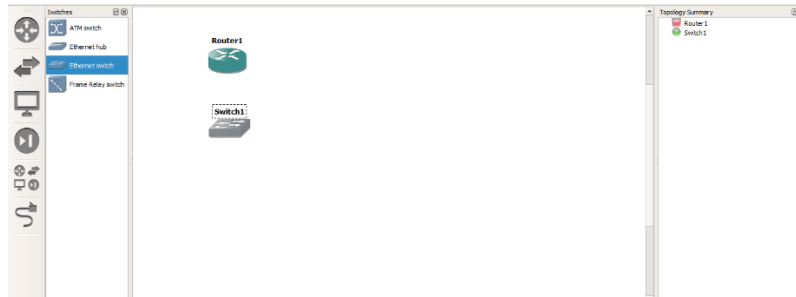
Μετά την επιτυχή εγκατάσταση του GNS3 θα προβούμε στην υλοποίηση ενός βασικού σεναρίου επιλέγοντας τους κατάλληλους δρομολογητές(routers) και μεταγωγείς (switches). Έχοντας φορτώσει τα κατάλληλα images, τα οποία ουσιαστικά είναι dynamips τα οποία προσομοιώνουν την λειτουργία ενός πραγματικού Cisco router, επιλέγουμε τον δρομολογητή *c2691* που περιέχει περισσότερα από ένα adapter slots. Επιλέγοντας διπλό κλικ στο όνομα του router έχω τη δυνατότητα να αλλάξω το όνομά του.



Εικόνα 6-3 Δρομολογητής c2691

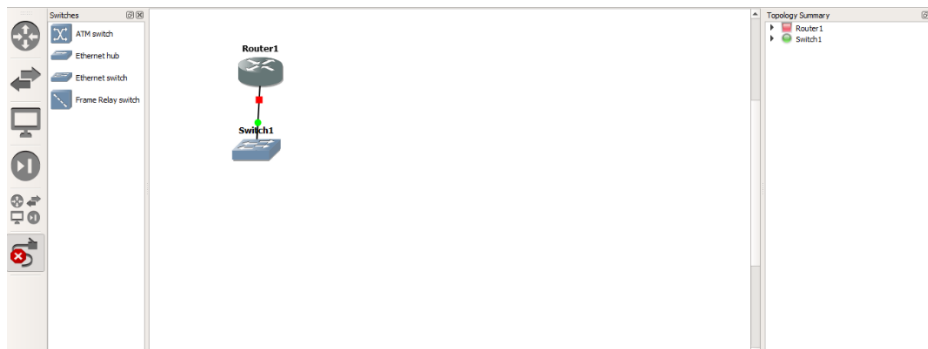


Στη συνέχεια επιλέγουμε το Ethernet switch από την καρτέλα Browse switches



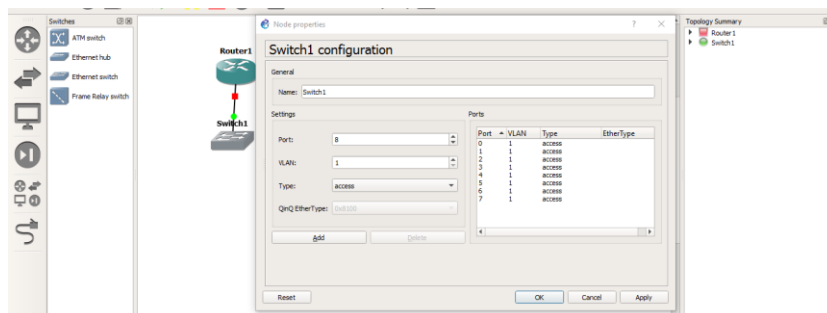
Εικόνα 6-4 Επιλογή Ethernet switch

και πραγματοποιούμε την σύνδεση με τον router επιλέγοντας την καρτέλα add a link συνδέοντας το Router1 με το Switch 1 επιλέγοντας από την πλευρά του router fastethernet0/0 και στο switch Ethernet0.



Εικόνα 6-5 Σύνδεση των δυο υλικολογισμικών

Κάνοντας διπλό κλικ στο switch μπορώ να παραμετροποιήσω τα ports του πατώντας την επιλογή **add->Apply->OK** προσθέτω ένα επιπλέον port στο μεταγωγέα.

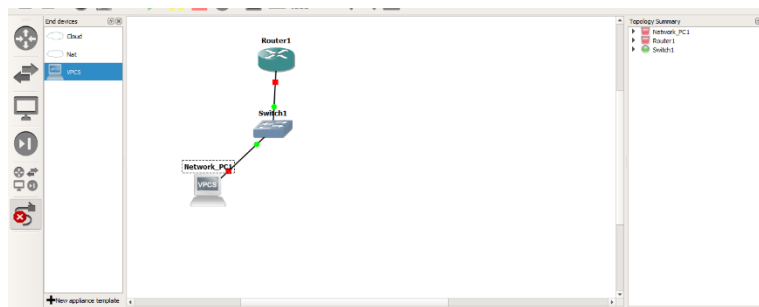


Εικόνα 6-6 Παραμετροποίηση των ports του μεταγωγέα



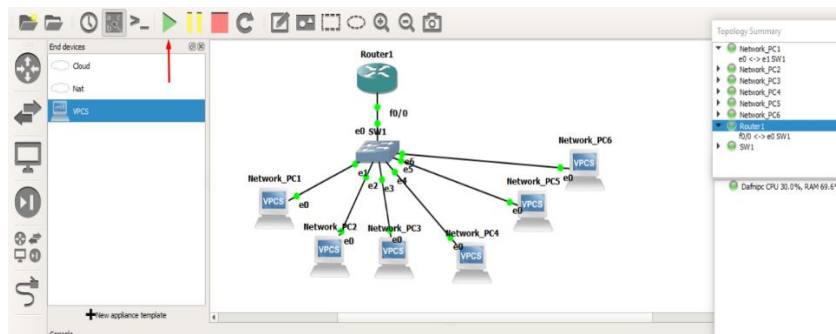
Στην καρτέλα Browse end devices έχω την δυνατότητα να δω και να επιλέξω τερματικό, επιλέγω ένα VPC το οποίο προσομοιώνει τη συμπεριφορά ενός πραγματικού τερματικού και το σέρνω μέσα στο project.

Συνδέω το pc αφού έχω αλλάξει το όνομα του σε Network_PC1 με το switch χρησιμοποιώντας ένα link από την καρτέλα add a link,βάζοντας στο switch την επιλογή Ethernet1 και στο Network_PC1 Ethernet0.



Εικόνα 6-7 Σύνδεση του τερματικού με το μεταγωγέα

Συνεχίζουμε την ίδια διαδικασία για τα υπόλοιπα 4 τερματικά δημιουργώντας μία τοπολογία αστερά και πατώντας την επιλογή Start έχουμε το παρακάτω αποτέλεσμα.



Εικόνα 6-8 Δημιουργία τοπολογίας αστερά

6.2.1 Εκχώρηση IP Διευθύνσεων στην τοπολογία αστερά

Στη συνέχεια παραθέτουμε τις βασικές εντολές για την παραμετροποίηση του δρομολογητή όπου επιλέγουμε να αλλάζουμε το hostname και να εισάγουμε κωδικό για την πρόσβαση σε αυτόν. Μέσω του command line,πατάμε δεξί κλικ στο Router1 επιλέγουμε Console και τρέχουμε τις παρακάτω εντολές.

```
Router1#conf term
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Router1(config)#no ip domain-lookup
```



```
Router1(config)#line con 0
Router1(config-line)#enable password cisco
Router1(config)#line con 0
Router1(config-line)#password cisco login
Router1(config-line)#exit
Router1(config)#line con 0
Router1(config-line)#password cisco
Router1(config-line)#login
Router1(config-line)#exit
Router1(config)#hostname CCSL
CCSL(config)#banner motd #CCSL NETWORK#
CCSL(config)#exit
```

```
CCSL#*Mar 1 00:05:16.019: %SYS-5-CONFIG_I: Configured from console by console
CCSL#copy running-config startup-config
Destination filename [startup-config]? startup-config
Building configuration...
[OK]
```

Απόδοση IPv4 διεύθυνσης στον Router1

```
CCSL#conf term
Enter configuration commands, one per line. End with CNTL/Z.
CCSL(config)#interface f0/0
CCSL(config-if)#ip add 192.168.1.1 255.255.255.0
CCSL(config-if)#no shut
```



```
CCSL#conf term
Enter configuration commands, one per line. End with CNTL/Z.
CCSL(config)#interface f0/0
CCSL(config-if)#ip add 192.168.1.1 255.255.255.0
CCSL(config-if)#no shut
CCSL(config-if)#exit
CCSL(config)#exit
CCSL#sho
*Mar 1 00:04:16.771: %SYS-5-CONFIG_I: Configured from console by console
CCSL#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet0/0
CCSL#
```

Εικόνα 6-9 Απόδοση IPv4 διεύθυνσης στο δρομολογητή CCSL

Παρακάτω βλέπουμε πως θα δώσουμε στατικά IPv4 διεύθυνση στα VPCs. Πατώντας διπλό κλικ πάνω στο PC1 μπαίνουμε στο command line του και με τις παρακάτω εντολές εκχωρούμε IP του δικτύου 192.168.1.0 και δίνουμε την επόμενη IP δηλαδή την 192.168.1.2 και default gateway αυτή του Router1. Το VPC θα ψάξει για διπλότυπη διεύθυνση και αν δεν βρει θα το εκχωρήσει στο τερματικό. Με την εντολή *save* αποθηκεύεται η διεύθυνση στο αρχείο *startup.vpc* του κάθε τερματικού έτσι κάθε φορά που θα ανοίγουμε το εκάστοτε VPC θα ανοίγει και το ανάλογο αρχείο με τις αποθηκευμένες ρυθμίσεις.



```
PC1
VPCS> ip 192.168.1.2/24 192.168.1.1
Checking for duplicate address...
PC1 : 192.168.1.2 255.255.255.0 gateway 192.168.1.1

VPCS> save
Saving startup configuration to startup.vpc
. done

VPCS> █
```

Εικόνα 6-10 Εκχώρηση IP διεύθυνσης σε VPC

Ακολουθώντας την ίδια διαδικασία δίνουμε IP διεύθυνση και στα υπόλοιπα τερματικά της τοπολογίας μας με την σειρά που τα έχουμε συνδεδεμένα

6.2.2 Πειράματα επιβεβαίωσης ορθής λειτουργίας του δικτύου

Θα εκτελέσουμε την εντολή *ping* από το τερματικό PC1 προς το τερματικό PC3 το οποίο έχει IP διεύθυνση 192.168.1.3 και θα διαπιστώσουμε ότι τα πακέτα μεταδίδονται επιτυχώς.

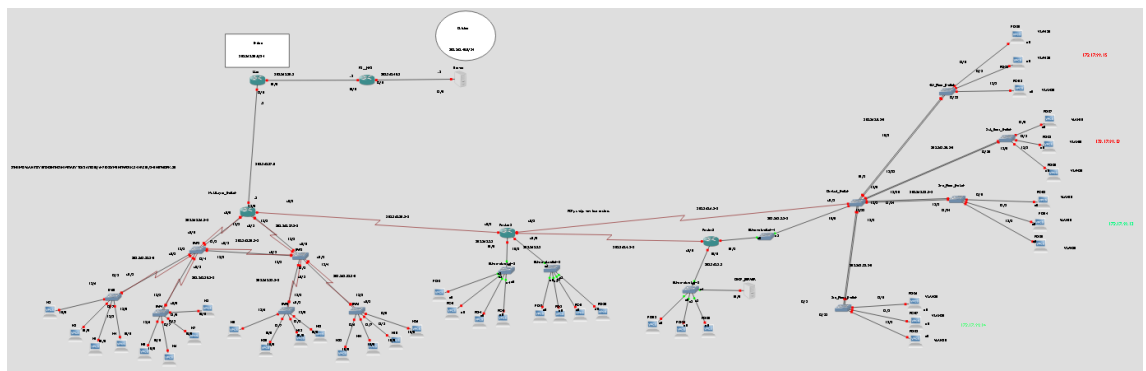


```
Network_PC1
VPCS> ping 192.168.1.3
84 bytes from 192.168.1.3 icmp_seq=1 ttl=64 time=0.000 ms
84 bytes from 192.168.1.3 icmp_seq=2 ttl=64 time=0.000 ms
84 bytes from 192.168.1.3 icmp_seq=3 ttl=64 time=0.000 ms
84 bytes from 192.168.1.3 icmp_seq=4 ttl=64 time=0.000 ms
84 bytes from 192.168.1.3 icmp_seq=5 ttl=64 time=0.000 ms
VPCS>
```

Εικόνα 6-11 Ping από το PC1 προς το τερματικό PC3

6.3 Υλοποίηση τοπολογίας εμπορικού κέντρου και επικοινωνία μεταξύ των υποδικτύων της

Σε αυτή την τοπολογία θα ασχοληθούμε κυρίως με πρωτόκολλα τα οποία αφορούν εικονικά τοπικά δίκτυα(VLAN) , όπως το VTP καθώς και άλλα που αφορούν την γειτνίαση RIP ή την δυναμική απόδοση IP διευθύνσεων όπως το DHCP ή την απόκρυψη της IP διεύθυνσης στο δίκτυο όπως το NAT. Για τις ανάγκες του σεναρίου θα προσομοιάσουμε το δίκτυο ενός εμπορικού κέντρου μιας πόλης , το οποίο θα απαρτίζεται από ένα φροντιστήριο, κάποια καταστήματα ένδυσης και υπόδησης, ένα κατάστημα ηλεκτρονικών ειδών και έναν κινηματογράφο. Εν τέλει η επικοινωνία θα ελεγχθεί πραγματοποιώντας εντολές ping και traceroute.



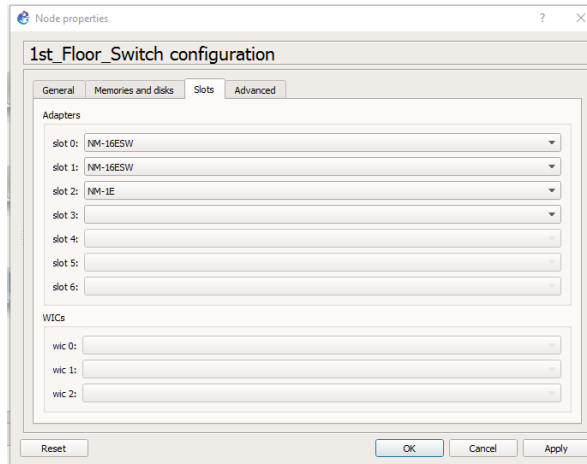
Εικόνα 6-12 Τοπολογία εμπορικού κέντρου

6.3.1 Τμήμα Φροντιστηρίου

Αρχικά θα αναφερθούμε στο φροντιστήριο και στην συνύπαρξη διαφορετικών εικονικών δικτύων για το προσωπικό(faculty/staff), το οποίο θα είναι το vlan 10, για τους μαθητές (students) , το vlan 20 , για τους επισκέπτες (guest) θα είναι το vlan 30 ενώ για την διαχείριση(management) το vlan 99.

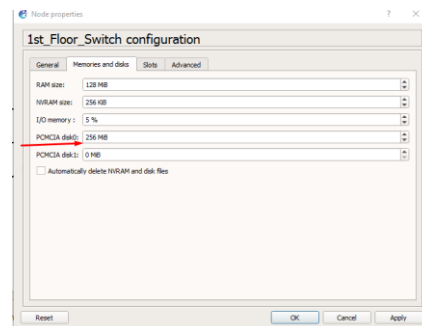


Εισάγουμε κατάλληλα 5 switches, τα οποία μπορούν ταυτόχρονα να κάνουν και δρομολόγηση διότι είναι επιπέδου 3 το 1st_Floor_Switch, το 2nd_Floor_Switch το 3rd_Floor_Switch, το 4th_Floor_Switch και το κεντρικό Switch(Central_Switch) που επιτυγχάνει τη διασύνδεση με τα υπόλοιπα. Στην συνέχεια επεξεργαζόμαστε τις διαδικτυακές διεπαφές στα slot που υπάρχουν σε 15 και 15 fastethernet πόρτες για κάθε slot σε κάθε μεταγωγέα..



Εικόνα 6-13 Επεξεργασία των network interfaces

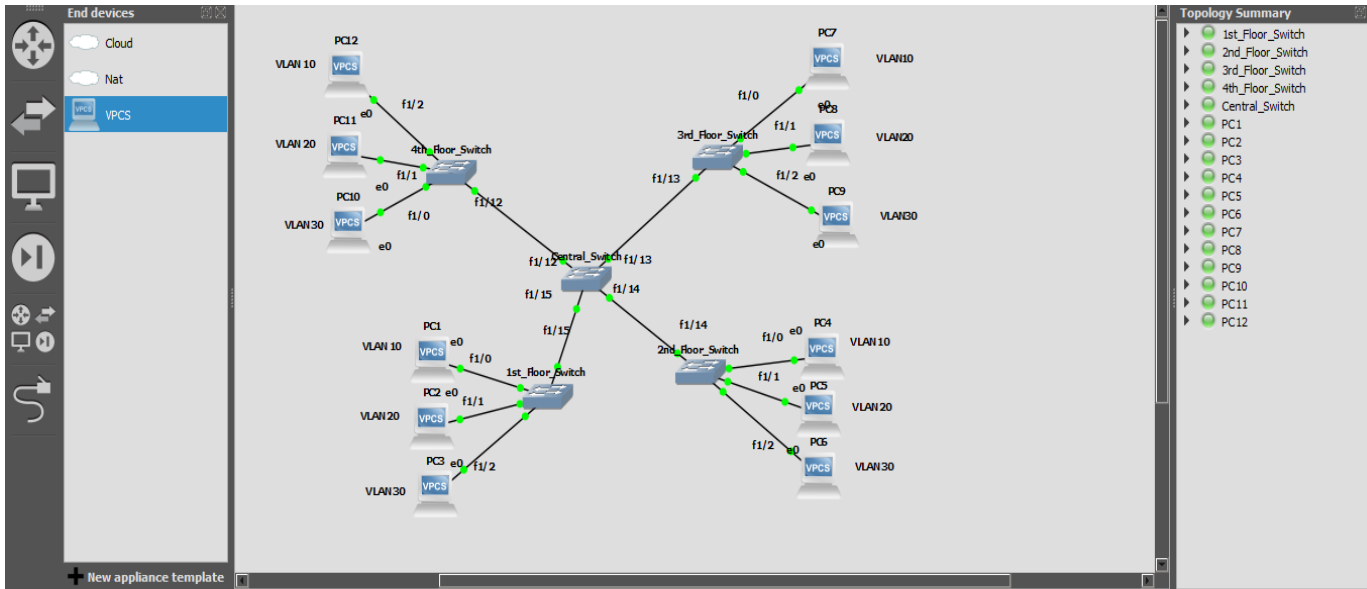
Επιλέγουμε να μην σβήνει αυτόματα την NVRAM και στο PCMCIA disk0 το αφήνουμε 256MiB για να μπορεί να αποθηκεύσει τη vlan database ο δρομολογητής.



Εικόνα 6-14 PCMCIA για την αποθήκευση των διαφορετικών vlans



Παρακάτω φαίνεται η συνδεσμολογία των πέντε μεταγωγών και η σύνδεση τους στις κατάλληλες πόρτες.



Εικόνα 6-15 Αρχική συνδεσμολογία

Πηγαίνουμε στο terminal του 1st_floor_switch και γράφουμε τον παρακάτω κώδικα:

1st floor

```
conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
no ip domain-lookup
```

```
no logging console
```

```
hostname 1stFloor_Switch
```

```
enable secret ccsl
```

```
vtp domain
```

```
vtp domain vtp.lan
```

Changing VTP domain name from NULL to vtp.lan

```
vtp password ccsl
```

Setting device VLAN database password to ccsl

```
vlan 10
```

```
name faculty/staff
```

```
exit
```




```
vlan 20
name students
exit
vlan 30
name guest
exit
vlan 99
name management
exit
exit
conf t
int fa1/0
description Access for Faculty Staff
switchport access vlan 10
exit
int fa1/1
description Access for Students
switchport access vlan 20
exit
int fa1/2
description access for guest
switchport access vlan 30
exit
int range fa1/0 - 15
switchport mode access
no shut
exit
conf t
interface vlan 99
$ VLAN RemoteManagement of 1st_Floor_Switch
ip add 172.17.99.12 255.255.255.0
no shut
```



```
exit
int fa 1/15
description Trunk with 1st_Floor_Switch
switchport mode trunk
switchport trunk native vlan 99
$ trunk allowed vlan 1,1002-1006,10,20,30,99
exit
wr
```

Αφότου τρέξω αυτές τις εντολές στο 1st_Floor_Switch στη συνέχεια τρέχω την παρακάτω εντολή για να δω αν έχουν δημιουργηθεί και ενεργοποιηθεί σωστά τα VLAN.

```
1st_Floor_Switch#show int fa1/15 trunk
Port      Mode      Encapsulation  Status      Native vlan
Fa1/15    on        802.1q         trunking    99

Port      Vlans allowed on trunk
Fa1/15    1-4094

Port      Vlans allowed and active in management domain
Fa1/15    1,10,20,30,99,1025

Port      Vlans in spanning tree forwarding state and not pruned
Fa1/15    1,10,20,30,99,1025
1st_Floor_Switch#
```

Εικόνα 6-16 Προβολή vlan

Εκτελώ τον ίδιο κώδικα και στο 2nd_floor_switch αλλάζοντας το hostname σε 2nd_Floor Switch και το range από f1/0 – 15 σε f1/0 -14.

2nd floor

```
conf t
Enter configuration commands, one per line. End with CNTL/Z.
no ip domain-lookup
no logging console
hostname 2ndFloor_Switch
enable secret ccs1
vtp domain
```

```
vtp domain vtp.lan
Changing VTP domain name from NULL to vtp.lan
```



```
vtp password ccs1
Setting device VLAN database password to ccs1
vlan 10
name faculty/staff
exit
vlan 20
name students
exit
vlan 30
name guest
exit
vlan 99
name management
exit
exit
conf t
int fa1/0
description Access for Faculty Staff
switchport access vlan 10
exit
int fa1/1
description Access for Students
switchport access vlan 20
exit
int fa1/2
description access for guest
switchport access vlan 30
exit
int range fa1/0 - 14
switchport mode access
no shut
exit
```



```
interface vlan 99
$ VLAN RemoteManagement of 2nd _Floor_Switch
ip add 172.17.99.13 255.255.255.0
no shut
exit
int fa 1/14
description
description Trunk with 2nd _Floor_Switch
switchport mode trunk
switchport trunk native vlan 99
$ trunk allowed vlan 1,1002-1006,10,20,30,99
exit
wr
```

3rd_Floor

```
conf t
Enter configuration commands, one per line. End with CNTL/Z.
no ip domain-lookup
no logging console
hostname 3rdFloor_Switch
enable secret ccs1
vtp domain

vtp domain vtp.lan
Changing VTP domain name from NULL to vtp.lan
vtp password ccs1
```



Setting device VLAN database password to ccs1

vlan 10

name faculty/staff

exit

vlan 20

name students

exit

vlan 30

name guest

exit

vlan 99

name management

exit

exit

conf t

int fa1/0

description Access for Faculty Staff

switchport access vlan 10

exit

int fa1/1

description Access for Students

switchport access vlan 20

exit

int fa1/2

description access for guest

switchport access vlan 30

exit

int range fa1/0 - 13



```
switchport mode access
```

```
no shut
```

```
exit
```

```
interface vlan 99
```

```
$ VLAN RemoteManagement of 3rd_Floor_Switch
```

```
ip add 172.17.99.14 255.255.255.0
```

```
no shut
```

```
exit
```

```
int fa 1/13
```

```
description
```

```
description Trunk with 3rd_Floor_Switch
```

```
switchport mode trunk
```

```
switchport trunk native vlan 99
```

```
$ trunk allowed vlan 1,1002-1006,10,20,30,99
```

```
exit
```

```
wr
```

Για το Central switch ακολουθούμε την ίδια διαδικασία με παραπάνω. Βλέπουμε από το Central_Switch τα VLAN

```
Central_Switch#show int trunk

Port      Mode      Encapsulation  Status      Native vlan
Fa1/12    on        802.1q         trunking    99
Fa1/13    on        802.1q         trunking    99
Fa1/14    on        802.1q         trunking    99
Fa1/15    on        802.1q         trunking    99

Port      Vlans allowed on trunk
Fa1/12    1-4094
Fa1/13    1-4094
Fa1/14    1-4094
Fa1/15    1-4094

Port      Vlans allowed and active in management domain
Fa1/12    1,10,20,30,99,1025-1027
Fa1/13    1,10,20,30,99,1025-1027
Fa1/14    1,10,20,30,99,1025-1027
Fa1/15    1,10,20,30,99,1025-1027

Port      Vlans in spanning tree forwarding state and not pruned
Fa1/12    1,10,20,30,99,1025-1027
Fa1/13    1,10,20,30,99,1025-1027
Fa1/14    1,10,20,30,99,1025-1027

Port      Vlans in spanning tree forwarding state and not pruned
Fa1/15    1,10,20,30,99,1025-1027
Central_Switch#
```

Εικόνα 6-17 Προβολή πληροφοριών trunk του central switch



Εκτελείται ο ίδιος κώδικας και στο 4th floor_switch. Για την f1/12 τρέγω τον κώδικα του 4th_Floor

4th_floor

conf t

Enter configuration commands, one per line. End with CNTL/Z.

no ip domain-lookup

no logging console

hostname 4thFloor_Switch

enable secret ccs1

vtp domain

vtp domain vtp.lan

Changing VTP domain name from NULL to vtp.lan

vtp password ccs1

Setting device VLAN database password to ccs1

vlan 10

name faculty/staff

exit

vlan 20

name students

exit

vlan 30

name guest

exit

vlan 99

name management

exit

exit



```
conf t
int fa1/0
description Access for Faculty Staff
```

```
switchport access vlan 10
exit
int fa1/1
description Access for Students
switchport access vlan 20
exit
int fa1/2
description access for guest
switchport access vlan 30
exit
int range fa1/0 - 12
switchport mode access
no shut
exit
```

```
conf t
interface vlan 99
$ VLAN RemoteManagement of 4th _Floor_Switch
ip add 172.17.99.15 255.255.255.0
no shut
exit
int fa 1/12
description
description Trunk with 4th_Floor_Switch
switchport mode trunk
```




```
switchport trunk native vlan 99
$ trunk allowed vlan 1,1002-1006,10,20,30,99
exit
wr
```

Εκτέλεση εντολών ping για να διαπιστώσουμε ότι υπάρχει επικοινωνία

Εκτελούμε ping απο το PC10 στο PC7 και το αντίθετο του VLAN10 από το 4th floor στο 3rd floor

```
PC10:
VPCS> show
NAME IP/MASK GATEWAY MAC LPORT R
VPCS1 172.17.10.30/24 172.17.10.1 00:50:79:66:68:09 10050 1
fe80::250:79ff:fe66:6809/64
VPCS> ping 172.17.10.27
84 bytes from 172.17.10.27 icmp_seq=1 ttl=64 time=0.000 ms
84 bytes from 172.17.10.27 icmp_seq=2 ttl=64 time=0.000 ms
84 bytes from 172.17.10.27 icmp_seq=3 ttl=64 time=0.000 ms
84 bytes from 172.17.10.27 icmp_seq=4 ttl=64 time=0.000 ms
84 bytes from 172.17.10.27 icmp_seq=5 ttl=64 time=0.000 ms
VPCS>

PC7:
VPCS> show
NAME IP/MASK GATEWAY MAC
LPORTRHOST:PORT
VPCS1 172.17.10.27/24 172.17.10.1 00:50:79:66:68:
06 10044 127.0.0.1:10045
fe80::250:79ff:fe66:6806/64
VPCS> ping 172.17.10.30
84 bytes from 172.17.10.30 icmp_seq=1 ttl=64 time=0.000 ms
84 bytes from 172.17.10.30 icmp_seq=2 ttl=64 time=0.000 ms
84 bytes from 172.17.10.30 icmp_seq=3 ttl=64 time=0.000 ms
84 bytes from 172.17.10.30 icmp_seq=4 ttl=64 time=0.000 ms
84 bytes from 172.17.10.30 icmp_seq=5 ttl=64 time=0.000 ms
VPCS>
```

Εικόνα 6-18 ping από pc που ανήκουν στο ίδιο (VLAN10)

Εκτελούμε ping απο το PC26 στο PC20 και το αντίθετο του VLAN10 από το 3rd floor προς το 4th floor

```
PC-20:
VPCS> ping 172.17.10.33
84 bytes from 172.17.10.33 icmp_seq=1 ttl=64 time=0.998 ms
84 bytes from 172.17.10.33 icmp_seq=2 ttl=64 time=0.998 ms
84 bytes from 172.17.10.33 icmp_seq=3 ttl=64 time=0.998 ms
84 bytes from 172.17.10.33 icmp_seq=4 ttl=64 time=0.998 ms
84 bytes from 172.17.10.33 icmp_seq=5 ttl=64 time=1.000 ms
VPCS>

PC-26:
Copyright (c) 2007-2014, Paul Meng (mirkshi@gmail.com)
All rights reserved.
VPCS is free software, distributed under the terms of the "BSD" licenc
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.
Press '?' to get help.
Executing the startup file
Checking for duplicate address...
PC1 : 172.17.10.33 255.255.255.0
PC-26>
PC-26> ping 172.17.10.30
84 bytes from 172.17.10.30 icmp_seq=1 ttl=64 time=0.999 ms
84 bytes from 172.17.10.30 icmp_seq=2 ttl=64 time=1.000 ms
84 bytes from 172.17.10.30 icmp_seq=3 ttl=64 time=1.000 ms
84 bytes from 172.17.10.30 icmp_seq=4 ttl=64 time=1.000 ms
84 bytes from 172.17.10.30 icmp_seq=5 ttl=64 time=0.998 ms
PC-26>
```

Εικόνα 6-19 Επιπλέον ping σε ίδιο vlan



Ping σε διαφορετικό vlan δεν μπορεί να επιτευχθεί δηλαδή από pc 20 του vlan 10 4th floor switch σε pc 22 vlan 20 του 4th floor switch

```
PC-22
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS> show ip
NAME      : VPCS[1]
IP/MASK   : 172.17.30.20/24
GATEWAY   : 255.255.255.0
DNS       :
MAC       : 00:50:79:66:68:10
LPORT    : 10362
RHOST:PORT : 127.0.0.1:10363
MTU      : 1500

VPCS> ping 172.17.10.33
host (255.255.255.0) not reachable

VPCS>

PC-20
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS>
VPCS> show ip
NAME      : VPCS[1]
IP/MASK   : 172.17.10.30/24
GATEWAY   : 255.255.255.0
DNS       :
MAC       : 00:50:79:66:68:0e
LPORT    : 10360
RHOST:PORT : 127.0.0.1:10361
MTU      : 1500

VPCS> ping 172.17.30.20
host (255.255.255.0) not reachable

VPCS>
```

Εικόνα 6-20 Δεν υπάρχει επικοινωνία διαφορετικών VLAN

Αντίστροφα pc 19 του vlan 30 1st floor με pc 24 vlan 20 2nd floor

```
PC-19
Welcome to Virtual PC Simulator, version 0.6.1
Dedicated to Daling.
Build time: Jun  1 2015 11:42:32
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" li
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

Checking for duplicate address...
PC1 : 172.17.30.21 255.255.255.0

PC-19>
PC-19> ping 172.17.20.42
No gateway found

PC-19>

PC-24
Welcome to Virtual PC Simulator, version 0.6.1
Dedicated to Daling.
Build time: Jun  1 2015 11:42:32
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

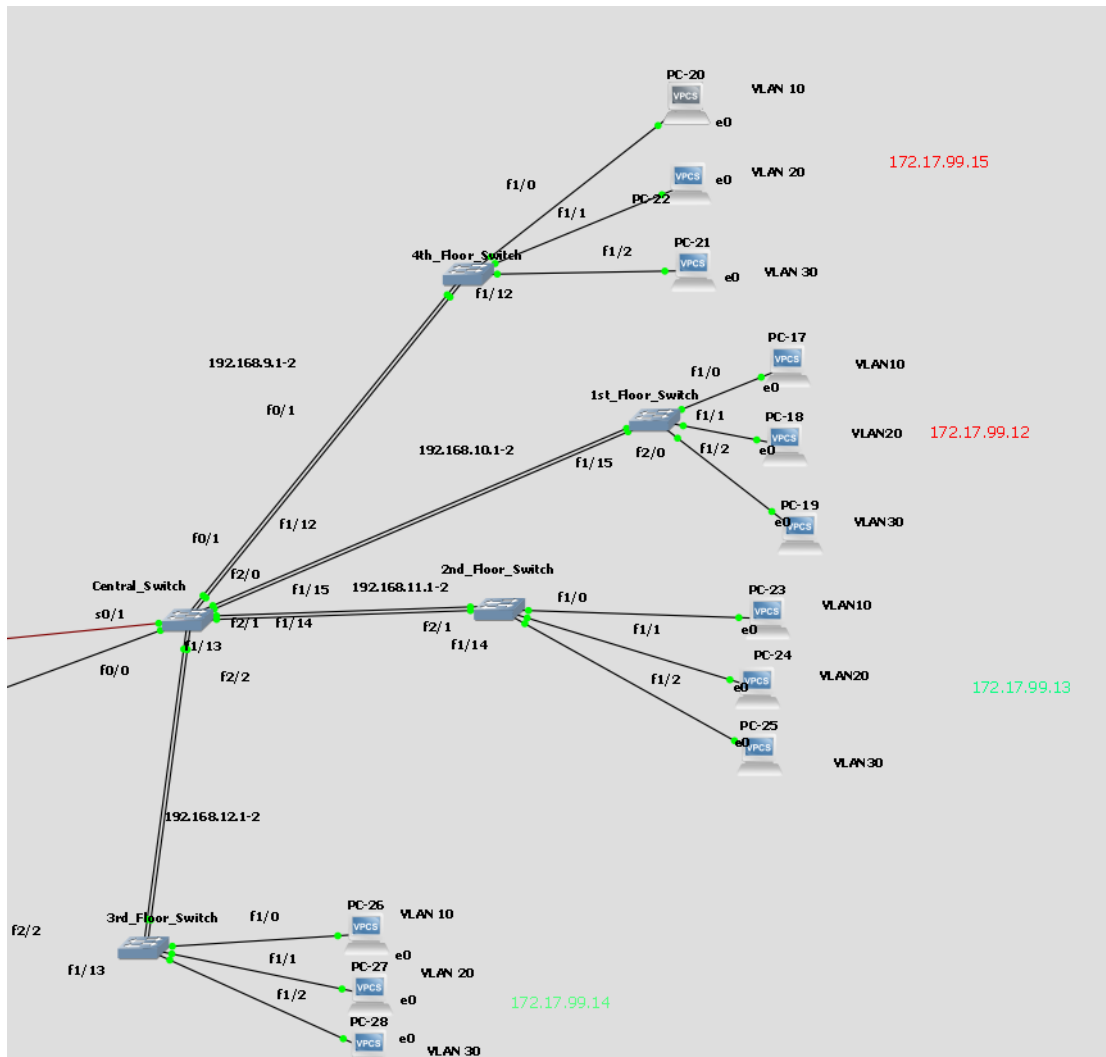
Checking for duplicate address...
PC1 : 172.17.20.42 255.255.255.0

PC-24>
PC-24> ping 172.17.30.21
No gateway found

PC-24>
```

Εικόνα 6-21 Μη ύπαρξη επικοινωνίας μεταξύ διαφορετικών VLAN

Ping από το ίδιο vlan αλλά σε διαφορετικό όροφο pc 18 first floor vlan 20 σε pc 24 2nd floor vlan 20



Εικόνα 6-24 Τοπολογία φρονιτηστηρίου με τα δίκτυα που υπάρχουν

Με διαφορετικό χρώμα παρουσιάζονται οι διευθύνσεις IP για την διαχείριση του κάθε ορόφου , οι οποίες ανήκουν στο vlan 99. Όσο αναφορά την ύπαρξη των διπλών ζεύξεων μεταξύ των ορόφων είναι διότι μεταφέρονται και διαφημίζονται διαφορετικά δίκτυα από ολόκληρη την τοπολογία έτσι ώστε να υπάρχει επικοινωνία μέχρι το επίπεδο του ορόφου κάθε φορά και όχι των host των vlan. Για να επιτευχθεί το γεγονός αυτό , χρειάστηκε να διαφημίσουμε τα δίκτυα με το πρωτόκολλο RIP.

1st floor

Χρειάστηκε να διαφημίσουμε το δίκτυο του τρέχοντα κατάλληλα το RIP

```
!
router rip
 network 192.168.10.0
```

Εικόνα 6-25 Δίκτυο διαφήμισης 1st floor



Εντολή `sh ip route` για να ελέγξουμε τα δίκτυα με τα οποία έχει επιτευχθεί η γειτνίαση

1st_Floor_Switch

```
Press RETURN to get started!

1st_Floor_Switch#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R    192.168.12.0/24 [120/1] via 192.168.10.1, 00:00:08, FastEthernet2/0
R    192.168.8.0/24 [120/1] via 192.168.10.1, 00:00:08, FastEthernet2/0
R    192.168.9.0/24 [120/1] via 192.168.10.1, 00:00:08, FastEthernet2/0
C    192.168.10.0/24 is directly connected, FastEthernet2/0
     172.17.0.0/24 is subnetted, 1 subnets
C     172.17.99.0 is directly connected, Vlan99
R    192.168.11.0/24 [120/1] via 192.168.10.1, 00:00:08, FastEthernet2/0
R    192.168.3.0/24 [120/2] via 192.168.10.1, 00:00:08, FastEthernet2/0
1st_Floor_Switch#
```

Εικόνα 6-26 Γειτονικά και μη δίκτυα του 1st floor μετά το RIP

2nd floor

```
!
router rip
 network 192.168.11.0
```

Εικόνα 6-27 Δίκτυο διαφήμισης 2nd floor



2nd_Floor_Switch

```
!
!
end

2ndFloor_Switch#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R    192.168.12.0/24 [120/1] via 192.168.11.1, 00:00:08, FastEthernet2/1
R    192.168.8.0/24 [120/1] via 192.168.11.1, 00:00:08, FastEthernet2/1
R    192.168.9.0/24 [120/1] via 192.168.11.1, 00:00:08, FastEthernet2/1
R    192.168.10.0/24 [120/1] via 192.168.11.1, 00:00:08, FastEthernet2/1
R    172.17.0.0/24 is subnetted, 1 subnets
C      172.17.99.0 is directly connected, Vlan99
C    192.168.11.0/24 is directly connected, FastEthernet2/1
R    192.168.3.0/24 [120/2] via 192.168.11.1, 00:00:08, FastEthernet2/1
2ndFloor_Switch#
```

Εικόνα 6-28 Γειτονικά και μη δίκτυα του 2nd floor μετά το RIP

3rd floor

```
!
router rip
network 192.168.12.0
```

Εικόνα 6-29 Δίκτυο διαφήμισης 3rd floor



3rd_Floor_Switch

```
!  
!  
end  
  
3rdFloor_Switch#sh ip route  
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2  
ia - IS-IS inter area, * - candidate default, U - per-user static route  
o - ODR, P - periodic downloaded static route  
  
Gateway of last resort is not set  
  
C    192.168.12.0/24 is directly connected, FastEthernet2/2  
R    192.168.8.0/24 [120/1] via 192.168.12.1, 00:00:17, FastEthernet2/2  
R    192.168.9.0/24 [120/1] via 192.168.12.1, 00:00:17, FastEthernet2/2  
R    192.168.10.0/24 [120/1] via 192.168.12.1, 00:00:17, FastEthernet2/2  
    172.17.0.0/24 is subnetted, 1 subnets  
C    172.17.99.0 is directly connected, Vlan99  
R    192.168.11.0/24 [120/1] via 192.168.12.1, 00:00:17, FastEthernet2/2  
R    192.168.3.0/24 [120/2] via 192.168.12.1, 00:00:17, FastEthernet2/2  
3rdFloor_Switch#
```

Εικόνα 6-30 Γειτονικά και μη δίκτυα του 2nd floor μετά το RIP

4th floor

```
!  
router rip  
network 192.168.9.0
```

Εικόνα 6-31 Δίκτυο διαφήμισης 4th floor

```
4thFloor_Switch#sh ip route  
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2  
ia - IS-IS inter area, * - candidate default, U - per-user static route  
o - ODR, P - periodic downloaded static route  
  
Gateway of last resort is not set  
  
R    192.168.12.0/24 [120/1] via 192.168.9.1, 00:00:12, FastEthernet0/1  
R    192.168.8.0/24 [120/1] via 192.168.9.1, 00:00:12, FastEthernet0/1  
C    192.168.9.0/24 is directly connected, FastEthernet0/1  
R    192.168.10.0/24 [120/1] via 192.168.9.1, 00:00:12, FastEthernet0/1  
    172.17.0.0/24 is subnetted, 1 subnets  
C    172.17.99.0 is directly connected, Vlan99  
R    192.168.11.0/24 [120/1] via 192.168.9.1, 00:00:12, FastEthernet0/1  
R    192.168.3.0/24 [120/2] via 192.168.9.1, 00:00:12, FastEthernet0/1  
4thFloor_Switch#
```

Εικόνα 6-32 Γειτονικά και μη δίκτυα του 2nd floor μετά το RIP



Το central switch είναι η δίοδος με το υπόλοιπο δίκτυο, διότι έχει διαφημιστεί σωστά με τους γειτονικούς του κόμβους.

 Central_Switch

```
interface Vlan99
 ip address 172.17.99.15 255.255.255.0
!
router rip
 network 192.168.2.0
 network 192.168.3.0
 network 192.168.4.0
 network 192.168.5.0
 network 192.168.6.0
 network 192.168.8.0
 network 192.168.9.0
 network 192.168.10.0
 network 192.168.11.0
 network 192.168.12.0
!
```

Εικόνα 6-33 Δίκτυα διαφήμισης Central switch

```
Central_Switch#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

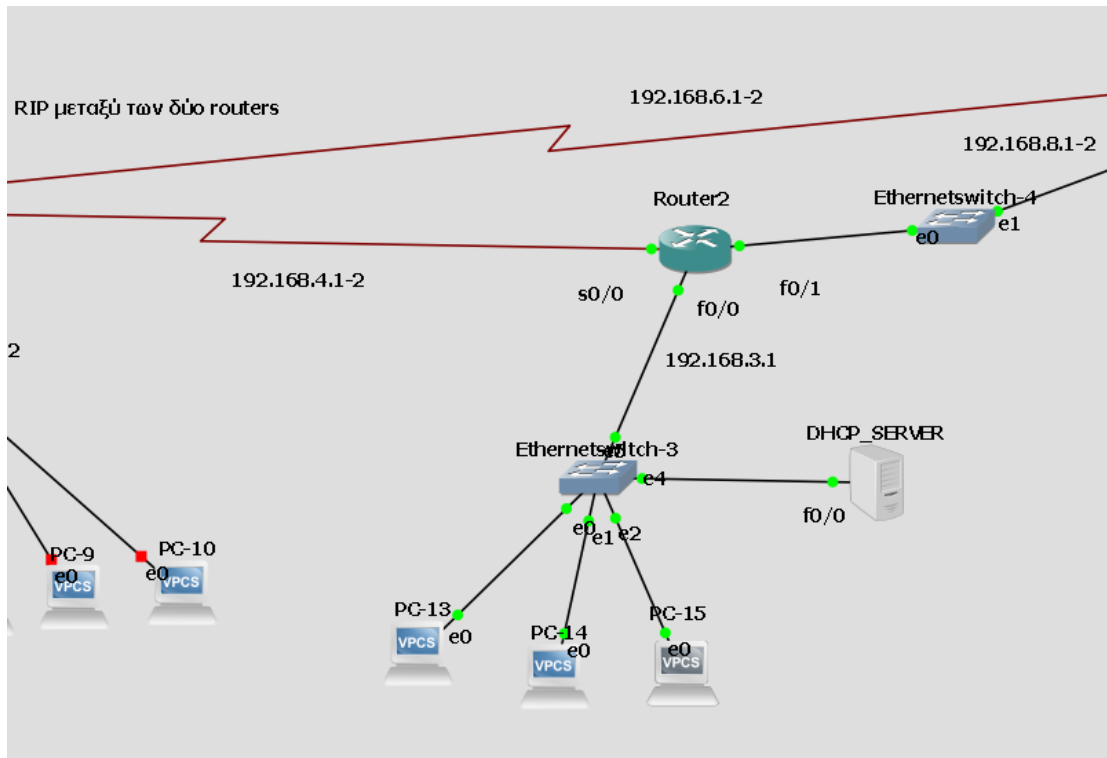
Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet2/2
C    192.168.8.0/24 is directly connected, FastEthernet0/0
C    192.168.9.0/24 is directly connected, FastEthernet0/1
C    192.168.10.0/24 is directly connected, FastEthernet2/0
     172.17.0.0/24 is subnetted, 1 subnets
C       172.17.99.0 is directly connected, Vlan99
C    192.168.11.0/24 is directly connected, FastEthernet2/1
R    192.168.3.0/24 [120/1] via 192.168.8.1, 00:00:27, FastEthernet0/0
Central_Switch#
```

Εικόνα 6-34 Γειτονικά και μη δίκτυα του Central switch μετά το RIP

6.3.2 Τμήμα Κινηματογράφου

Στο συγκεκριμένο μέρος της τοπολογίας το οποίο είναι ένας κινηματογράφος , υπάρχει ένας κεντρικός server , ο οποίος αποδίδει IP διευθύνσεις στα τερματικά που είναι συνδεδεμένα με τον μεταγωγέα EthernetSwitch, όπως απεικονίζεται στην παρακάτω εικόνα.



Εικόνα 6-35 Μέρος τοπολογίας του τμήματος κινηματογράφου

Την στιγμή που ανοίγει το τερματικό pc14, του αποδίδεται IP διεύθυνση 192.168.3.2 δυναμικά με το πρωτόκολλο DHCP.



```
Welcome to Virtual PC Simulator, version 0.6.1
Dedicated to Daling.
Build time: Jun 1 2015 11:42:32
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

DD
ORA IP 192.168.3.2/24 GW 192.168.3.1

PC-14>
PC-14> █
```

Εικόνα 6-36 Δυναμική απόδοση IP



Ελέγχουμε την επικοινωνία με τον διακομιστή (DHCP server)

```
DHCP_SERVER - PC-15
privilege level 15
logging synchronous
line vty 0 4
login
!
!
end

DHCP_SERVER#sh ip int br
Interface                IP-Address      OK? Method Status
FastEthernet0/0          192.168.3.254  YES NVRAM  up
Serial0/0                 unassigned     YES NVRAM  administratively do
FastEthernet0/1          unassigned     YES NVRAM  administratively do
Serial0/1                 unassigned     YES NVRAM  administratively do
Serial0/2                 unassigned     YES NVRAM  administratively do
Serial0/3                 unassigned     YES NVRAM  administratively do
FastEthernet1/0          unassigned     YES NVRAM  administratively do
DHCP_SERVER#ping 192.168.3.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/14/32 ms
DHCP_SERVER#

PC-15
Press '?' to get help.
Executing the startup file
DDORA IP 192.168.3.4/24 GW 192.168.3.1

PC-15>
PC-15> ping 192.168.3.1
84 bytes from 192.168.3.1 icmp_seq=1 ttl=255 time=2.997 ms
84 bytes from 192.168.3.1 icmp_seq=2 ttl=255 time=2.997 ms
84 bytes from 192.168.3.1 icmp_seq=3 ttl=255 time=65.943 ms
84 bytes from 192.168.3.1 icmp_seq=4 ttl=255 time=33.971 ms
84 bytes from 192.168.3.1 icmp_seq=5 ttl=255 time=5.995 ms

PC-15> ping 192.168.3.254
84 bytes from 192.168.3.254 icmp_seq=1 ttl=255 time=20.982 ms
84 bytes from 192.168.3.254 icmp_seq=2 ttl=255 time=5.994 ms
84 bytes from 192.168.3.254 icmp_seq=3 ttl=255 time=4.995 ms
84 bytes from 192.168.3.254 icmp_seq=4 ttl=255 time=1.999 ms
84 bytes from 192.168.3.254 icmp_seq=5 ttl=255 time=7.993 ms

PC-15>
```

Εικόνα 6-37 Επικοινωνία με τον DHCP server

Αποστολή πακέτων με τον τέταρτο όροφο του φροντιστηρίου (4th floor)

```
PC-13
Build time: Jun 1 2015 11:42:32
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.
Executing the startup file
DDORA IP 192.168.3.3/24 GW 192.168.3.1

PC-13>
PC-13> ping 192.168.9.2
84 bytes from 192.168.9.2 icmp_seq=1 ttl=253 time=85.927 ms
84 bytes from 192.168.9.2 icmp_seq=2 ttl=253 time=117.899 ms
84 bytes from 192.168.9.2 icmp_seq=3 ttl=253 time=52.955 ms
84 bytes from 192.168.9.2 icmp_seq=4 ttl=253 time=81.930 ms
84 bytes from 192.168.9.2 icmp_seq=5 ttl=253 time=53.954 ms

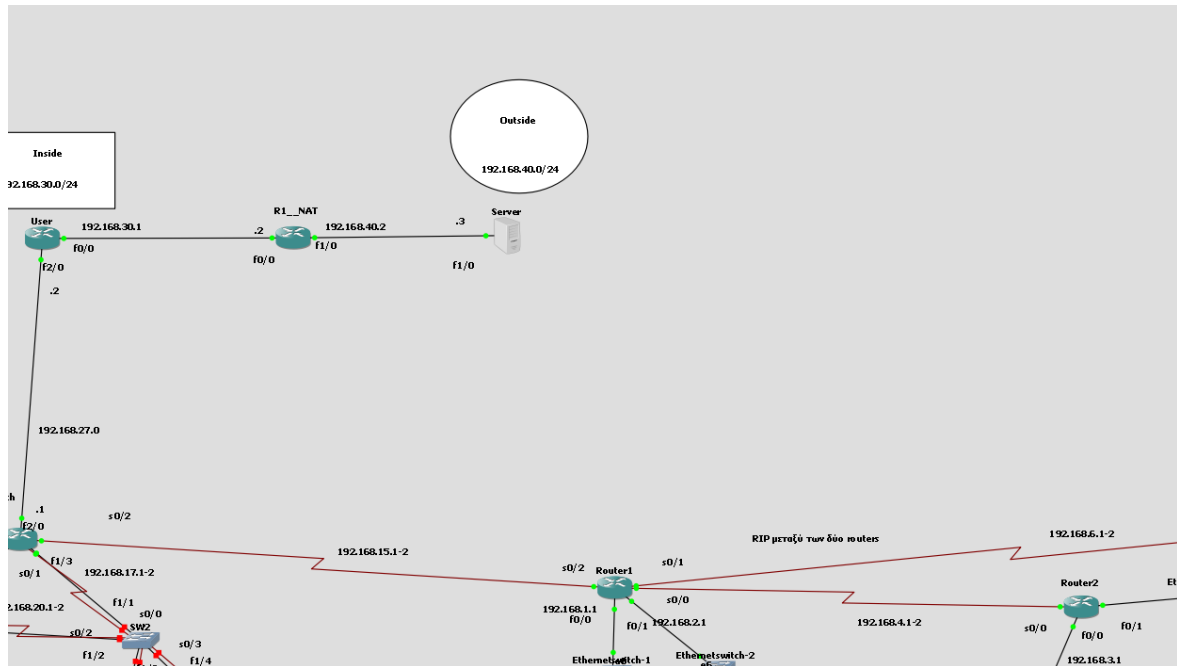
PC-13>
```

Εικόνα 6-38 ping στο 4th floor Switch



6.3.3 Απομακρυσμένη πρόσβαση και υλοποίηση NAT

Επικοινωνία του δρομολογητή Router1 με τον απομακρυσμένο server , ο οποίος παρέχει υπηρεσίες streaming , κατάλληλες για να εξυπηρετήσει τις ανάγκες συγκεκριμένου τμήματος του κινηματογράφου.



Εικόνα 6-39 Διαδρομή που θα ακολουθήσουν τα πακέτα

Υλοποίηση NAT στον R1_NAT

```
R1_NAT#conf t
```

```
R1_NAT(config)#int fa0/0
```

```
R1_NAT(config-if)#ip nat inside
```

```
R1_NAT(config-if)#int fa1/0
```

```
R1_NAT(config-if)#ip nat outside
```

```
R1_NAT(config-if)#exit
```

```
R1_NAT(config)#ip nat inside source static 192.168.30.2 192.168.40.2
```

Στο πάνω αριστερά κομμάτι της τοπολογίας έχουμε υλοποιήσει το πρωτόκολλο NAT έτσι ώστε να αποκρύπτεται η πηγαία διεύθυνση IP στην επικοινωνία με τον Server. Με λίγα λόγια η οποιαδήποτε IP του δικτύου που προσπαθεί να επικοινωνήσει αποκρύπτεται και εμφανίζεται μόνο αυτή της διεπαφής f1/0 του δρομολογητή R1_NAT 192.168.40.2. Γεγονός που παρατηρεί κάποιος και μέσα από το Wireshark κάνοντας capture.



```
Router2#ping 192.168.40.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.40.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/69/84 ms
Router2#
```

Εικόνα 6-40 Επιτυχής επικοινωνία με τον Server

| | | | | | | |
|----|-----------|--------------|--------------|------|-------------------------|---|
| 12 | 31.593003 | 192.168.40.3 | 192.168.40.2 | ICMP | 114 Echo (ping) reply | id=0x0001, seq=0/0, ttl=255 (request in 11) |
| 13 | 31.619979 | 192.168.40.2 | 192.168.40.3 | ICMP | 114 Echo (ping) request | id=0x0001, seq=1/256, ttl=254 (reply in 14) |
| 14 | 31.625974 | 192.168.40.3 | 192.168.40.2 | ICMP | 114 Echo (ping) reply | id=0x0001, seq=1/256, ttl=255 (request in 13) |
| 15 | 31.652954 | 192.168.40.2 | 192.168.40.3 | ICMP | 114 Echo (ping) request | id=0x0001, seq=2/512, ttl=254 (reply in 16) |
| 16 | 31.658945 | 192.168.40.3 | 192.168.40.2 | ICMP | 114 Echo (ping) reply | id=0x0001, seq=2/512, ttl=255 (request in 15) |

Εικόνα 6-41 Αποτέλεσμα απόκρυψης IP από το wireshark

Εκτελώντας το Rip πρωτοκόλλου στο Router 2 είχαμε ως αποτέλεσμα την γνωστοποίησή του με τα εξής δίκτυα .

```
Router2
!
router rip
 network 172.17.0.0
 network 192.168.3.0
 network 192.168.4.0
 network 192.168.5.0
 network 192.168.8.0
```

Εικόνα 6-42 Διαφήμιση των δικτύων του Router2

```
Router2#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

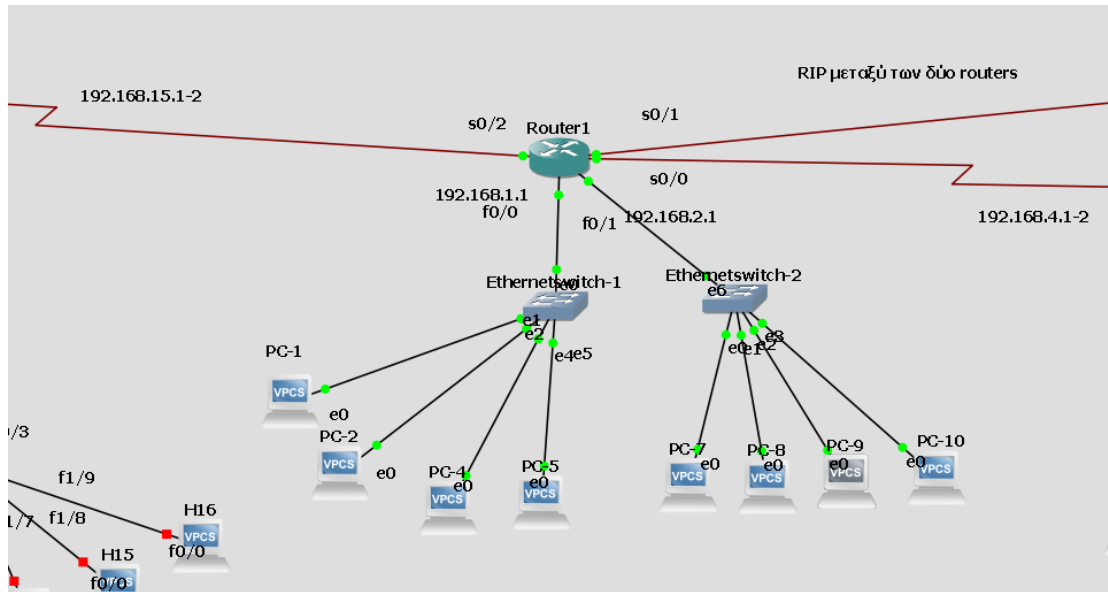
R    192.168.30.0/24 [120/3] via 192.168.4.1, 00:00:12, Serial0/0
R    192.168.15.0/24 [120/1] via 192.168.4.1, 00:00:12, Serial0/0
C    192.168.8.0/24 is directly connected, FastEthernet0/1
R    192.168.27.0/24 [120/2] via 192.168.4.1, 00:00:12, Serial0/0
R    192.168.40.0/24 [120/4] via 192.168.4.1, 00:00:12, Serial0/0
C    192.168.4.0/24 is directly connected, Serial0/0
R    192.168.1.0/24 [120/1] via 192.168.4.1, 00:00:12, Serial0/0
R    192.168.2.0/24 [120/1] via 192.168.4.1, 00:00:12, Serial0/0
C    192.168.3.0/24 is directly connected, FastEthernet0/0
Router2#
```

Εικόνα 6-43 Γειτονικά και μη δίκτυα του Router2 (κινηματογράφος) μετά το RIP



6.3.4 Κατάστημα ηλεκτρονικών ειδών

Στην συνέχεια απεικονίζεται ένα κατάστημα ηλεκτρικών ειδών το οποίο απαρτίζεται από ένα κεντρικό δρομολογητή , δυο απλούς μεταγωγείς και δέκα τερματικά. Τα pc αναγνωρίζουν ως default gateway τις IP διευθύνσεις του Router1 στα interfaces fastEthernet0/0 και fastEthernet0/1 , 192.168.1.1 και 192.168.2.1 αντίστοιχα.



Εικόνα 6-44 Κατάστημα ηλεκτρονικών ειδών

PC-4

```
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

Checking for duplicate address...
PC1 : 192.168.1.5 255.255.255.0 gateway 192.168.1.1

PC-4>
PC-4> show ip

NAME       : PC-4[1]
IP/MASK    : 192.168.1.5/24
GATEWAY    : 192.168.1.1
DNS        :
MAC        : 00:50:79:66:68:02
LPORT     : 10388
RHOST:PORT : 127.0.0.1:10389
MTU       : 1500

PC-4> █
```

Εικόνα 6-45 Χρήσης pc4 δικτύου 192.168.1.0



```
PC-8
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

Checking for duplicate address...
PC1 : 192.168.2.3 255.255.255.0 gateway 192.168.2.1

PC-8>
PC-8> show ip

NAME       : PC-8[1]
IP/MASK    : 192.168.2.3/24
GATEWAY    : 192.168.2.1
DNS        :
MAC        : 00:50:79:66:68:05
LPORT     : 10390
RHOST:PORT : 127.0.0.1:10391
MTU       : 1500

PC-8>
```

Εικόνα 6-46 Χρήστης pc8 δικτύου 192.168.1.0

```
PC-4> ping 192.168.2.3
192.168.2.3 icmp_seq=1 timeout
84 bytes from 192.168.2.3 icmp_seq=2 ttl=63 time=13.989 ms
84 bytes from 192.168.2.3 icmp_seq=3 ttl=63 time=21.981 ms
84 bytes from 192.168.2.3 icmp_seq=4 ttl=63 time=19.984 ms
84 bytes from 192.168.2.3 icmp_seq=5 ttl=63 time=19.983 ms

PC-4>

PC-8> ping 192.168.1.5
84 bytes from 192.168.1.5 icmp_seq=1 ttl=63 time=20.982 ms
84 bytes from 192.168.1.5 icmp_seq=2 ttl=63 time=21.982 ms
84 bytes from 192.168.1.5 icmp_seq=3 ttl=63 time=20.980 ms
84 bytes from 192.168.1.5 icmp_seq=4 ttl=63 time=21.980 ms
84 bytes from 192.168.1.5 icmp_seq=5 ttl=63 time=19.983 ms

PC-8>
```

Εικόνα 6-47 Επικοινωνία ping μεταξύ των τερματικών των δύο δικτύων

```
PC-4> trace 192.168.2.3
trace to 192.168.2.3, 8 hops max, press Ctrl+C to stop
 1 192.168.1.1 4.996 ms 9.991 ms 8.993 ms
 2 *192.168.2.3 20.982 ms (ICMP type:3, code:3, Destination port unreachable)

PC-4>

PC-8> trace 192.168.1.5
trace to 192.168.1.5, 8 hops max, press Ctrl+C to stop
 1 192.168.2.1 11.990 ms 8.993 ms 9.992 ms
 2 *192.168.1.5 19.983 ms (ICMP type:3, code:3, Destination port unreachable)

PC-8>
```

Εικόνα 6-48 Μονοπάτι μεταξύ των τερματικών

```
Router1
#
 privilege level 15
 logging synchronous
 line vty 0 4
 login
!
!
end

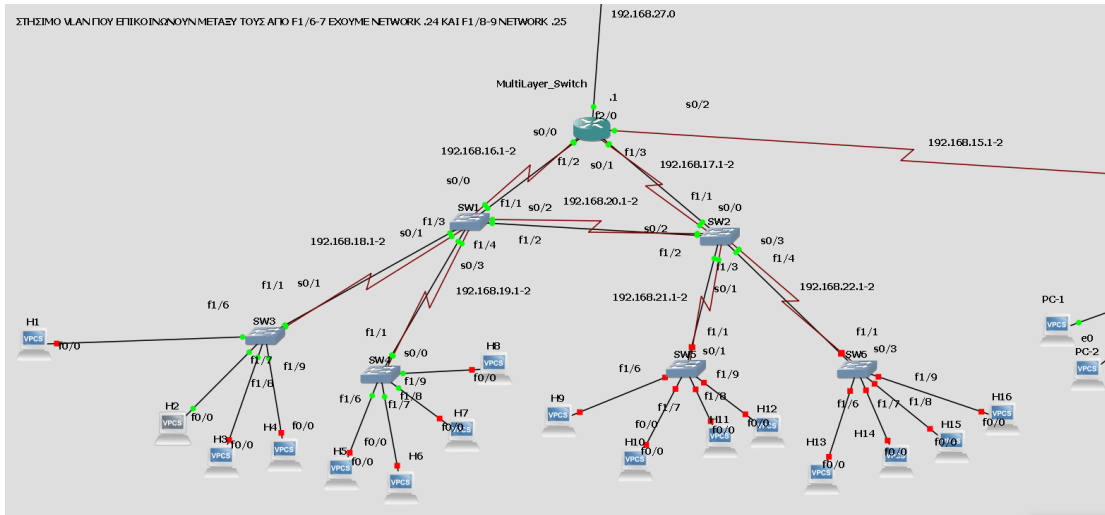
Router1#sh ip int br
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 192.168.1.1 YES NVRAM up up
Serial0/0 192.168.4.1 YES NVRAM up down
FastEthernet0/1 192.168.2.1 YES NVRAM up up
Serial0/1 192.168.6.1 YES NVRAM up down
Serial0/2 192.168.15.2 YES NVRAM up up
Serial0/3 unassigned YES NVRAM administratively down down
FastEthernet1/0 unassigned YES NVRAM administratively down down
Router1#ping 192.168.15.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.15.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Router1#
```

Εικόνα 6-49 Ping από το Router1 στο multilayerSwitch



6.3.5 Καταστήματα Ρουχισμού (InterVlan τοπολογία)



Εικόνα 6-50 InterVlan τοπολογία

Το κομμάτι αυτό αναφέρεται σε υποκαταστήματα ρουχισμού και είναι στην ουσία είναι ένα InterVlan , έτσι υπάρχει επικοινωνία μεταξύ όλων των host , είτε είναι στο ίδιο δίκτυο αλλά διαφορετικό Network layer 3 switch είτε σε διαφορετικό δίκτυο αλλά ίδιο δικτυακού επιπέδου switch. Πιο συγκεκριμένα στα interfaces f1/6 – f1/7 μεταφέρεται δίκτυο 192.168.24.0 του vlan10 administration , ενώ στις f1/8 – f1/9 δίκτυο 192.168.25.0 του vlan20 staff . Η λειτουργία αυτή συμβαίνει διότι τα δικτυακά switch λειτουργούν το πρωτόκολλο του VTP.

```
VLAN ISL Id: 10
Name: administration
Media Type: Ethernet
VLAN 802.10 Id: 100010
State: Operational
MTU: 1500

VLAN ISL Id: 20
Name: staff
Media Type: Ethernet
VLAN 802.10 Id: 100020
State: Operational
MTU: 1500
```

Εικόνα 6-51 Vlan10 & Vlan20



Προγραμματισμός

Αρχικά θα πάμε στο multilayer_Switch το οποίο εκτελεί τις περισσότερες ενέργειες όπως διαμοιρασμός ip διευθύνσεων και δημιουργία των VLAN.

Multilayer_Switch

```
conf t
int ra fa1/2 -5
switchport trunk encapsulation dot1q
switchport mode trunk
exit
vtp version 2
vtp domain Christos
no ip domain-lookup
```

Στη συνέχεια πηγαίνω σε όλα τα switches και γράφω τον παρακάτω κώδικα που λειτουργούν ως client

```
conf t
vtp mode client
int ra fa1/1 -5
switchport mode trunk
```

Μετά θα ορίσω τα VLAN στο multilayer switch

```
conf t
vlan 10
name administration
exit
vlan 20
name staff
exit
ip dhcp pool administration
```




```
network 192.168.24.0 255.255.255.0
default-router 192.168.24.1
ip dhcp pool staff
network 192.168.25.0 255.255.255.0
default-router 192.168.25.1
int vlan 10
ip add 192.168.24.1 255.255.255.0
exit
int vlan 20
ip add 192.168.25.1 255.255.255.0
```

Και τέλος πηγαίνω πάλι σε όλα τα switch και τρέχω τα παρακάτω για να καθορίσω ποιες πόρτες θα είναι στο VLAN 10 και ποιες στο VLAN 20.

```
conf t
int ra fa1/6 -7
switchport access vlan 10
exit
int ra fa1/8 -9
switchport access vlan 20
exit
exit
wr
copy run start
```

Ο χρήστης h1 έχει ενεργοποιημένο το πρωτόκολλο DHCP στην διαδικτυακή του διεπαφή f1/6 και παρατηρούμε ότι δευτερόλεπτα αφότου ανοίξει παίρνει IP διεύθυνση με δυναμικό τρόπο την 192.168.24.3.



```
ip dhcp pool administration
network 192.168.24.0 255.255.255.0
default-router 192.168.24.1
!
ip dhcp pool staff
network 192.168.25.0 255.255.255.0
default-router 192.168.25.1
!
```

Εικόνα 6-52 Δημιουργία των δυο DHCP pool για τα vlan

```
H1
% This file system device reports an error

Press RETURN to get started!

*Aug 23 19:25:54.043: %IFMGR-7-NO_IFINDEX_FILE: Unable to open nvram:/ifIndex-table No such file or directory
*Aug 23 19:25:54.395: %DEC21140-1-INITFAIL: Unsupported PHY brand timed out, csr5=0x0
*Aug 23 19:26:11.415: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Aug 23 19:26:12.935: %SYS-5-CONFIG I: Configured from memory by console
*Aug 23 19:26:13.383: %SYS-5-RESTART: System restarted --
Cisco IOS Software, 7200 Software (C7200-ADVIPSERVICESK9-M), Version 15.2(4)85, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2014 by Cisco Systems, Inc.
Compiled Thu 20-Feb-14 06:51 by prod_rel_team
*Aug 23 19:26:13.875: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
H1#sh ip in
*Aug 23 19:26:24.859: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP address 192.168.24.3, m
sk 255.255.255.0, hostname H1

H1#sh ip int br
Interface          IP-Address      OK? Method Status          Protocol
FastEthernet0/0    192.168.24.3    YES DHCP    up              up
H1#
```

Εικόνα 6-53 Απόδοση IP δυναμικά με DHCP του administration vlan

Το κεντρικό switch , το οποίο είναι επιπέδου 3 πραγματοποιεί και δρομολόγηση είναι ο server , ο οποίος αποδίδει τις IP διευθύνσεις.

```
MultiLayer_Switch#sh ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/
Hardware address/
User name
192.168.24.2        0063.6973.636f.2d63.
6131.642e.3230.3034.
2e30.3030.302d.4661.
302f.30
192.168.24.3        0063.6973.636f.2d63.
6131.662e.3165.3638.
2e30.3030.302d.4661.
302f.30
MultiLayer_Switch#
```

Εικόνα 6-54 Εντολή dhcp binding για την απόδοση των IP διευθύνσεων



Έλεγχος ping

```
HI
Cisco IOS Software, 7200 Software (C7200-ADVIPSERVICESK9-M), Version 15
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2014 by Cisco Systems, Inc.
Compiled Thu 20-Feb-14 06:51 by prod_rel_team
*Aug 23 19:26:13.875: %LINEPROTO-5-UPDOWN: Line protocol on Interface Fa
HI#sh ip in
*Aug 23 19:26:24.859: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0
sk 255.255.255.0, hostname HI
HI#sh ip int br
Interface                IP-Address      OK? Method Status
FastEthernet0/0         192.168.24.3    YES DHCP  up
HI#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
HI#ping 192.168.24.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.24.4, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/261/1060
ms
HI#

H5
Press RETURN to get started!
*Aug 23 19:43:12.443: %DEC21140-1-INITFAIL: Unsupported PHY brand tim
*Aug 23 19:43:32.259: %LINK-3-UPDOWN: Interface FastEthernet0/0, chan
*Aug 23 19:43:33.319: %LINEPROTO-5-UPDOWN: Line protocol on Interface
*Aug 23 19:43:34.059: %SYS-5-CONFIG_I: Configured from memory by cons
*Aug 23 19:43:34.827: %SYS-5-RESTART: System restarted --
Cisco IOS Software, 7200 Software (C7200-ADVIPSERVICESK9-M), Version
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2014 by Cisco Systems, Inc.
Compiled Thu 20-Feb-14 06:51 by prod_rel_team
R13#
*Aug 23 19:43:46.387: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0
sk 255.255.255.0, hostname R13
R13#ping 192.168.24.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.24.3, timeout is 2 seconds:
!!!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 40/114/176
ms
R13#
```

Εικόνα 6-55 Ping από host ίδιου δικτύου 192.168.24.0

```
HI
*Aug 23 19:26:24.859: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0
sk 255.255.255.0, hostname HI
HI#sh ip int br
Interface                IP-Address      OK? Method Status
FastEthernet0/0         192.168.24.3    YES DHCP  up
HI#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
HI#ping 192.168.24.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.24.4, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/261/1060
ms
HI#ping 192.168.25.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.25.2, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 256/332/372
ms
HI#

H11
Press RETURN to get started!
*Aug 23 19:47:31.527: %DEC21140-1-INITFAIL: Unsupported PHY bran
*Aug 23 19:47:57.283: %LINK-3-UPDOWN: Interface FastEthernet0/0,
*Aug 23 19:47:58.523: %LINEPROTO-5-UPDOWN: Line protocol on Inte
*Aug 23 19:47:59.507: %SYS-5-CONFIG_I: Configured from memory by
*Aug 23 19:48:00.443: %SYS-5-RESTART: System restarted --
Cisco IOS Software, 7200 Software (C7200-ADVIPSERVICESK9-M), Ver
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2014 by Cisco Systems, Inc.
Compiled Thu 20-Feb-14 06:51 by prod_rel_team
*Aug 23 19:48:12.199: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthe
sk 255.255.255.0, hostname R8
R8#sh ip int br
Interface                IP-Address      OK? Method Status
FastEthernet0/0         192.168.25.2    YES DHCP  up
R8#ping 192.168.24.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.24.3, timeout is 2 sec
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 292/
ms
R8#
```

Εικόνα 6-56 Ping από host διαφορετικού δικτύου 192.168.24.0 & 192.168.25.0

```
interface Vlan10
 ip address 192.168.24.1 255.255.255.0
!
interface Vlan20
 ip address 192.168.25.1 255.255.255.0
!
router rip
 network 192.168.15.0
 network 192.168.16.0
 network 192.168.17.0
 network 192.168.27.0
 network 192.168.30.0
!
```

Εικόνα 6-57 Γνωστοποίηση των δικτύων με την χρήση του RIP

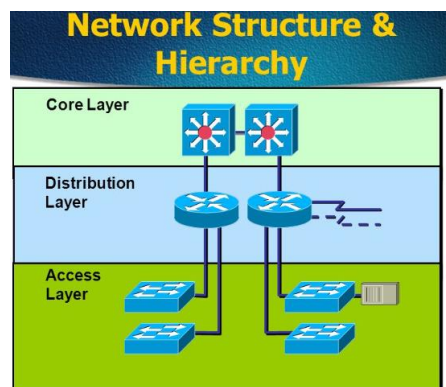


Η ύπαρξη διπλών ζεύξεων (serial) , αποσκοπεί στην διασύνδεση και επικοινωνία με τα υπόλοιπα δίκτυα της τοπολογίας καθώς και για λόγους πρόληψης σε περίπτωση βλάβης και απώλειας της σύνδεσης. Έχουμε διαφημίσει κατάλληλα τα δίκτυα με ολόκληρο το υπόλοιπο δίκτυο για να υπάρχει δίοδος επικοινωνίας με τα υπόλοιπα καταστήματα του εμπορικού κέντρου.

6.4 Δημιουργία δικτύου προσομοίωσης ενός μικρομεσαίου οργανισμού

6.4.1 Υλοποίηση τεχνικής υποδικτύωσης VLSM(Subnetting)

Στη συγκεκριμένη τοπολογία που θα προσομοιώσουμε δίνεται προσοχή στη σωστή υποδικτύωση και ορθή επικοινωνία ενός οργανισμού μεσαίας κλίμακας. Το δίκτυο που θα κατασκευάσουμε σταδιακά στηρίζεται στο παρακάτω ιεραρχικό μοντέλο, όπου υπάρχει επίπεδο διανομής, πυρήνα και πρόσβασης. Η τοπολογία θα υλοποιείται σταδιακά για την καλύτερη κατανόηση της.



Εικόνα 6-58 Μοντέλο δικτύου εταιρίας

Ο συγκεκριμένος οργανισμός θα απαρτίζεται από ορισμένα τμήματα όπου εκεί θα δουλέψουμε συγκεκριμένα πρωτόκολλα δρομολόγησης και θα ρυθμίσουμε κάποιους κανόνες. Ο οργανισμός που θα προσομοιώσουμε αποτελείται από 4 ορόφους που σε κάθε όροφο έχουμε τα εξής τμήματα.

- **Πρώτος όροφος :** Γραμματειακή Υποστήριξη(secretariat) και LAN Διαχείρισης(VLAN_admin)και WLAN(AP)
- **Δεύτερος όροφος:** Διεύθυνση Ανθρώπινων πόρων(human_resources) και Διεύθυνση οικονομικού(economic_dep) και WLAN(AP)
- **Τρίτος όροφος:** Τμήμα Ανάλυσης&Σχεδιασμού(analysis_and_design) και Τμήμα Πληροφορικής(IT),Server LAN(Server_LAN) και WLAN(AP)
- **Τέταρτος όροφος:** Υποστήριξη Πελατών(customer_support) και WLAN(AP)

Μέσα στις παρενθέσεις βλέπουμε τα ονόματα που θα έχουνε τα τμήματα όταν προσομοιωθούν στη συνέχεια ως VLAN.



Σε κάθε όροφο θα πρέπει να έχουμε ένα μεταγωγέα ο οποίος ουσιαστικά θα αποτελέσει το δίκτυο πρόσβασης πάνω στο οποίο θα συνδέονται όλα τα τερματικά. Το δίκτυο που αποτελεί τον πυρήνα της τοπολογίας θα είναι δύο άλλοι μεταγωγείς όπου σε περίπτωση που συμβεί κάποια δυσλειτουργία στον ένα μεταγωγέα θα αναλάβει την λειτουργία ο δεύτερος. Επιπλέον θα γίνει χρήση τριών εξυπηρετητών (servers) όπου ο κάθε ένας απ αυτούς θα συνδέεται σε δύο άλλους μεταγωγείς, όπου με τη σειρά τους θα συνδέονται στο πυρήνα του δικτύου για λόγους δυσλειτουργίας όπως αναφέρθηκε και πιο πάνω. Αυτοί θα είναι οι :

- Data Server
- Backup Server
- File Server

Με βάση τις απαιτήσεις των προσομοιώσεων δημιουργήθηκαν 9 τοπικά δίκτυα (LAN), τα οποία αναφέρονται στα τμήματα διάρθρωσης του οργανισμού. Ο σκοπός της τεχνικής υποδικτύωσης είναι να μην υπάρχει σπατάλη διευθύνσεων IP. Επιτακτική ανάγκη για την ανάλυση VLSM ορίζει επίσης το γεγονός ότι οι διευθύνσεις είναι IPv4. Ακόμη αξίζει να σημειωθεί ότι στον σχεδιασμό των δικτύων υπάρχει πρόβλεψη για ακόμα περαιτέρω διεύρυνση των χρηστών αφού έχουμε φροντίσει να υπάρχει διαθεσιμότητα σε IP διευθύνσεις. Η προσέγγιση βέβαια είναι θεωρητική στα πλαίσια των προσομοιώσεων που εκτελούμε.

Αρχικά θεωρούμε ότι ο εκάστοτε πάροχος μας διαθέτει την διεύθυνση 173.16.0.0/26. Για τον λόγο αυτό υπολογίσαμε την μετακίνηση της μάσκας υποδικτύου δεξιότερα από την αρχική της θέση που είναι στο 23^ο bit διότι η διαδικασία αυτή δημιουργεί υποδίκτυα.

| | | | | | | | | | | | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----|
| 15 ^ο | 14 ^ο | 13 ^ο | 12 ^ο | 11 ^ο | 10 ^ο | 9 ^ο | 8 ^ο | 7 ^ο | 6 ^ο | 5 ^ο | 4 ^ο | 3 ^ο | 2 ^ο | 1 ^ο | 0 |
| /17 | /18 | /19 | /20 | /21 | /22 | /23 | /24 | /25 | /26 | /27 | /28 | /29 | /30 | /31 | /32 |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Πίνακας 6-1 Πίνακας των bits και subnet mask

Θα ξεκινήσουμε με το τμήμα WLAN, διότι είναι το μεγαλύτερο σε χρήστες. Πιο συγκεκριμένα, θα αποτελείται από 50 work stations και 1 συσκευή, όπου συνολικά θα βρίσκονται 50 τερματικά.

Οπότε για το τμήμα WLAN οι διευθύνσεις δικτύου(network), του 1^{ου} χρήστη, του τελευταίου χρήστη και η broadcast θα είναι: 11 1111 → 32+16+8+4+2+1=63 άρα 63+0=63



Τμήμα WLAN

Network: 173.16.0.0/26

1^{ος} Χρήστης: 173.16.0.1/26

Τελευταίος Χρήστης: 173.16.0.62/26

Broadcast Address: 173.16.0.63/26

Το τμήμα Πληροφορικής απαρτίζεται από 7 work stations και 4 συσκευές, άρα συνολικά θα είναι 28 τερματικά. Ο υπολογισμός των διευθύνσεων IP σύμφωνα με τους hosts είναι: $63+31=94$

Τμήμα Πληροφορικής

Network: 173.16.0.64/27

1^{ος} Χρήστης: 173.16.0.65/27

Τελευταίος Χρήστης: 173.16.0.93/27

Broadcast Address: 173.16.0.94/27

Το υποδίκτυο Διαχείρισης VLAN αποτελείται από 18 work stations και 1 συσκευή, όπου συνολικά χρειάζονται 18 τερματικά. Ο υπολογισμός των διευθύνσεων IP σύμφωνα με τους hosts είναι: $31+95=126$

Τμήμα Διαχείρισης VLAN

Network: 173.16.0.95/27

1^{ος} Χρήστης: 173.16.0.96/27

Τελευταίος Χρήστης: 173.16.0.125/27

Broadcast Address: 173.16.0.126/27



Το τμήμα Διευθύνσεως Οικονομικού έχει 5 work stations και 3 συσκευές υπό την αιγίδα του, άρα θα χρειαστεί συνολικά 15 τερματικά. Ο υπολογισμός των διευθύνσεων IP σύμφωνα με τους hosts είναι: $15+127=142$

Τμήμα Διεύθυνσης Οικονομικού

Network: 173.16.0.127/28

1^{ος} Χρήστης: 173.16.0.128/28

Τελευταίος Χρήστης: 173.16.0.141/28

Broadcast Address: 173.16.0.142/28

Η γραμματειακή υποστήριξη έχει 2 work stations και 5 συσκευές από τις οποίες αποτελείται, έτσι τελικά χρειάζεται 10 τερματικά. Ο υπολογισμός των διευθύνσεων IP σύμφωνα με τους hosts είναι: $15+143=158$

Γραμματεία

Network: 173.16.0.143/28

1^{ος} Χρήστης: 173.16.0.144/28

Τελευταίος Χρήστης: 173.16.0.157/28

Broadcast Address: 173.16.0.158/28

Το τμήμα Διευθύνσεως Ανθρώπινων Πόρων έχει 4 work stations και 2 συσκευές. Επομένως χρειάζεται συνολικά 8 τερματικά. Ο υπολογισμός των διευθύνσεων IP σύμφωνα με τους hosts είναι: $7+159=167$

Τμήμα Διεύθυνσης Ανθρώπινων Πόρων

Network: 173.16.0.159/29

1^{ος} Χρήστης: 173.16.0.160/29

Τελευταίος Χρήστης: 173.16.0.165/29

Broadcast Address: 173.16.0.166/29



Το τμήμα Ανάλυσης & Σχεδιασμού έχει 3 work stations και 3 συσκευές ,γι' αυτό και χρειάζεται 9 τερματικά στο σύνολο. Ο υπολογισμός των διευθύνσεων IP σύμφωνα με τους hosts είναι: $15+167=182$

| |
|--|
| Τμήμα Ανάλυσης & Σχεδιασμού |
| Network: 173.16.0.167/29 |
| 1 ^{ος} Χρήστης: 173.16.0.168/29 |
| Τελευταίος Χρήστης: 173.16.0.181/29 |
| Broadcast Address: 173.16.0.182/29 |

Το τμήμα Ανάλυσης & Σχεδιασμού έχει 3 work stations και 5 συσκευές ,έτσι θα χρειαστεί 15 τερματικά συνολικά. Ο υπολογισμός των διευθύνσεων IP σύμφωνα με τους hosts είναι: $15+183=198$

| |
|--|
| Τμήμα Υποστήριξης Πελατών |
| Network: 173.16.0.183/30 |
| 1 ^{ος} Χρήστης: 173.16.0.184/30 |
| Τελευταίος Χρήστης: 173.16.0.197/30 |
| Broadcast Address: 173.16.0.198/30 |

Το υποδίκτυο των Εξυπηρετητών VLAN συνίσταται από 4 work stations και 2 συσκευές, όπου συνολικά χρειάζονται 8 τερματικά. Ο υπολογισμός των διευθύνσεων IP σύμφωνα με τους hosts είναι: $7+199=206$

| |
|--|
| Server LAN |
| Network: 173.16.0.199/30 |
| 1 ^{ος} Χρήστης: 173.16.0.200/30 |
| Τελευταίος Χρήστης: 173.16.0.205/30 |
| Broadcast Address: 173.16.0.206/30 |

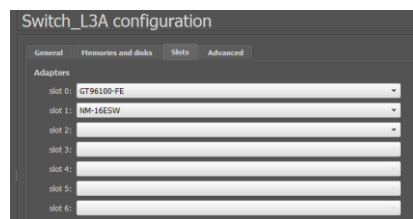


Ο σχεδιασμός ο οποίος έχει γίνει, αφήνει το περιθώριο για επέκταση των τμημάτων του οργανισμού μελλοντικά. Το είδος της δρομολόγησης (στατική ή δυναμική) τα πρωτόκολλα και οι κανόνες που ορίσαμε στο σενάριο μας θα παρουσιαστούν στη συνέχεια που θα προβούμε στην αναλυτική κατασκευή του δικτύου του οργανισμού που μελετήσαμε.

6.5 Υλοποίηση Τοπολογίας Οργανισμού

Στη συνέχεια προχωρούμε στην οπτικοποίηση και προσομοίωση του δικτύου που περιγράψαμε στην παραπάνω ενότητα. Θα μεταβούμε στο GUI του GNS3 και με drag and drop θα προσθέσουμε στον καμβά μας τον το c3725 IOS image που προσομοιώνει την λειτουργία ενός πραγματικού c3725 της εταιρείας Cisco.

Θα προσθέσουμε από την παλέτα μας στον καμβά το Switch_L3A που ουσιαστικά σε συνεργασία με το Switch_L3B(λειτουργεί σε περίπτωση δυσλειτουργίας) θα αποτελούν τον πυρήνα του δικτύου μας και εκεί θα συμβούν οι περισσότερες παραμετροποιήσεις όπως θα δούμε παρακάτω. Θα παραμετροποιήσουμε βέβαια και στους δύο μεταγωγείς τις κάρτες δικτύου θα βάλουμε και στους δύο NM-16ESW για να έχουμε 16 fastethernet interface και να επιτύχουμε ταχύτερη μετάδοση δεδομένων.



Εικόνα 6-59 Παραμετροποίηση slot 1 με κάρτα δικτύου NM-16 ESW

Στη συνέχεια θα κάνουμε start τον Switch_L3A και θα ξεκινήσουμε την παραμετροποίηση του. Θα αλλάξουμε το hostname σε Switch_L3A και θα θέσουμε κωδικό για τους χρήστες έτσι ώστε μόνο ο διαχειριστής του συστήματος να μπορεί να μπει και να το τροποποιήσει. Θα ενεργοποιήσουμε το ssh πρωτόκολλο για ασφαλέστερη μεταφορά απομακρυσμένων δεδομένων. Στο συγκεκριμένο σενάριο δε θα ασχοληθούμε σε πρακτικό επίπεδο τόσο με τις απομακρυσμένες συνδέσεις απλά στο τέλος θα δώσουμε μια μορφή ενός Firewall που σε πραγματικό χρόνο έχει σκοπό την προστασία με απομακρυσμένα sessions, το αφήνουμε για μελλοντική μελέτη.



```
Switch_L3A#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3A(config)#hostname Switch_L3A
Switch_L3A(config)#enable secret ccs1
Switch_L3A(config)#line vty 0 4
Switch_L3A(config-line)#password ccs1
Switch_L3A(config-line)#login
Switch_L3A(config-line)#transport input ssh
Switch_L3A(config-line)#exit
Switch_L3A(config)#line con 0
Switch_L3A(config-line)#password ccs1
Switch_L3A(config-line)#login
Switch_L3A(config-line)#exit
Switch_L3A(config)#exit
Switch_L3A#
*Mar  1 00:08:27.695: %SYS-5-CONFIG_I: Configured from console by console
```

Εικόνα 6-60 Βασικές παραμετροποιήσεις ασφαλείας του μεταγωγέα

Για να δούμε τις ρυθμίσεις που επιτύχαμε κάνουμε `sh run` και πηγαίνοντας στο CLI του μεταγωγέα βλέπουμε τον κωδικό πρόσβασης. Δεν το επιθυμούμε αυτό οπότε θα βάλουμε μια μορφή κρυπτογράφησης για να επιτύχουμε μεγαλύτερα επίπεδα ασφαλείας.

```
Switch_L3A
Switch_L3A#sh run
Building configuration...

Current configuration:
!
!
!
!
line con 0
  exec-timeout 0 0
  privilege level 15
  password ccs1
  logging synchronous
  login
line aux 0
  exec-timeout 0 0
  privilege level 15
  logging synchronous
line vty 0 4
  password ccs1
  login
  transport input ssh
```

Εικόνα 6-61 Βλέπουμε χωρίς κάποια μορφή κρυπτογράφησης ελεύθερα τον κωδικό

Με τις παρακάτω εντολές θα δώσουμε την κρυπτογράφηση και στην εικόνα 6-63 βλέπουμε το password ccs1 με κάποια μορφή hash.

```
Switch_L3A#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3A(config)#service password-encryption
Switch_L3A(config)#exit
Switch_L3A#write
*Mar  1 00:22:37.387: %SYS-5-CONFIG_I: Configured from console by console
Switch_L3A#write
Building configuration...
[OK]
```

Εικόνα 6-62 Εκτέλεση εντολών κρυπτογράφησης



```
line con 0
exec-timeout 0 0
privilege level 15
password 7 121A06041E
logging synchronous
login
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
line vty 0 4
password 7 104D0A0A0957
login
transport input ssh
```

Εικόνα 6-63 Κρυπτογραφημένος κωδικός

Επίσης στη συνέχεια θα κάνουμε ακόμη κάποιες παραμετροποιήσεις που αφορούν την ασφάλεια του δικτύου μας για να αποτρέψουμε την πρόσβαση των χρηστών σε HTTP servers για να αυξήσουμε την ασφάλεια του οργανισμού μας.

```
Switch_L3A#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3A(config)#no ip http server
Switch_L3A(config)#no ip http secure-server
Switch_L3A(config)#exit
```

Εικόνα 6-64 Αποτροπή πρόσβασης σε HTTP Servers

Έπειτα προχωρούμε στη ρύθμιση του VLAN του κάθε τμήματος αποδίδοντας του κάποιο αναγνωριστικό και χαρακτηριστικό όνομα για το καθένα. Για να γίνει βέβαια η αποθήκευση του VLAN θα πρέπει να δώσουμε την απαραίτητη μνήμη στο μεταγωγέα πρέπει να ρυθμιστεί στο συγκεκριμένο επίπεδο για να έχουμε επιτυχή αποθήκευση.

| General | Memories and disks | Slots | Advanced |
|---------------|--------------------|-------|----------|
| RAM size: | 128 MiB | | |
| NVRAM size: | 256 KiB | | |
| I/O memory : | 5 % | | |
| PCMCIA disk0: | 256 MiB | | |
| PCMCIA disk1: | 0 MiB | | |



```
Switch_L3A(vlan)#vlan 20 name IT
VLAN 20 modified:
  Name: IT
Switch_L3A(vlan)#vlan 30 name VLAN_admin
VLAN 30 modified:
  Name: VLAN_admin
Switch_L3A(vlan)#vlan 40 name economic_dep
VLAN 40 modified:
  Name: economic_dep
Switch_L3A(vlan)#vlan 50 name secretariat
VLAN 50 modified:
  Name: secretariat
Switch_L3A(vlan)#vlan 60 name human_resources
VLAN 60 modified:
  Name: human_resources
Switch_L3A(vlan)#vlan 70 name analysis_and_design
VLAN 70 modified:
  Name: analysis_and_design
Switch_L3A(vlan)#vlan 80 name customer_suport
VLAN 80 modified:
  Name: customer_suport
Switch_L3A(vlan)#vlan 90 name Server_LAN
VLAN 90 modified:
  Name: Server_LAN
```

Εικόνα 6-65 Δημιουργία και προσθήκη VLAN δικτύων

ΤΟ VLAN 10 λείπει σε αυτό το screenshot γιατί κάναμε κάποιες παραμετροποιήσεις και δεν το προσθέσαμε.

Μπαίνοντας παρακάτω στην βάση του VLAN βλέπουμε την επιτυχή προσθήκη των vlan και την ενεργή(active) τους κατάσταση.

```
Switch_L3A
VLAN 70 modified:
  Name: analysis_and_design
Switch_L3A(vlan)#vlan 80 name customer_suport
VLAN 80 modified:
  Name: customer suport
Switch_L3A(vlan)#vlan 90 name Server_LAN
VLAN 90 modified:
  Name: Server LAN
Switch_L3A(vlan)#exit
APPLY Completed.
Exiting...
Switch_L3A#show vlan-switch

VLAN Name                Status    Ports
-----
1    default                 active    Fa1/0, Fa1/1, Fa1/2, Fa1/3
                                           Fa1/4, Fa1/5, Fa1/6, Fa1/7
                                           Fa1/8, Fa1/9, Fa1/10, Fa1/11
                                           Fa1/12, Fa1/13, Fa1/14, Fa1/15
10   WLAN                   active
20   IT                     active
30   VLAN_admin            active
40   economic_dep         active
50   secretariat          active
60   human_resources      active
70   analysis_and_design  active
80   customer_suport     active
90   Server_LAN           active
```

Εικόνα 6-66 Επιτυχή προσθήκη VLANs

Έχοντας δημιουργήσει τα VLANs στη συνέχεια θα δώσουμε IP διεύθυνση αναλόγως με το subnetting που ακολουθήσαμε. Αφού έχουμε μπει σε configure terminal ακολουθούμε την παρακάτω αλληλουχία εντολών για την σωστή απόδοση IP στα VLANs. Με την επιτυχή ανάθεση των IP διευθύνσεων αποθηκεύουμε την λειτουργία του μεταγωγέα.



```
Switch_L3A#conf t
Enter Configuration commands, one per line. End with CNTL/Z.
Switch_L3A(config)#interface vlan 10
Switch_L3A(config-if)#ip add 172.16.0.2 255.255.255.192
Switch_L3A(config-if)#no shut
Switch_L3A(config-if)#interface vlan 20
Switch_L3A(config-if)#ip add 172.16.0.67 255.255.255.224
Switch_L3A(config-if)#no shut
Switch_L3A(config-if)#interface vlan 30
Switch_L3A(config-if)#ip add 172.16.0.97 255.255.255.224
Switch_L3A(config-if)#no shut
Switch_L3A(config-if)#interface vlan 40
Switch_L3A(config-if)#ip add 172.16.0.130 255.255.255.240
Switch_L3A(config-if)#no shut
Switch_L3A(config-if)#interface vlan 50
Switch_L3A(config-if)#ip add 172.16.0.147 255.255.255.240
Switch_L3A(config-if)#no shut
Switch_L3A(config-if)#interface vlan 60
Switch_L3A(config-if)#ip add 173.16.0.163 255.255.255.240
Switch_L3A(config-if)#no shut
Switch_L3A(config-if)#interface vlan 70
Switch_L3A(config-if)#ip add 173.16.0.177 255.255.255.240
Switch_L3A(config-if)#no shut
Switch_L3A(config-if)#interface vlan 80
Switch_L3A(config-if)#ip add 173.16.0.194 255.255.255.248
Switch_L3A(config-if)#no shut
Switch_L3A(config-if)#interface vlan 90
Switch_L3A(config-if)#ip add 173.16.0.203 255.255.255.248
Switch_L3A(config-if)#no shut
Switch_L3A(config-if)#exit
Switch_L3A(config)#exit
Switch_L3A#w
*Mar 1 15:08:14.432: %SYS-5-CONFIG_I: Configured from console by console
Switch_L3A#write
Building configuration...
[OK]
```

Εικόνα 6-67 Επιτυχής απόδοση διεύθυνσης IP στα VLANs

Θα μούμε στη συνέχεια στη βάση του VLAN και μπορούμε να δούμε κάποιες ενημερωτικές πληροφορίες(τεχνικά χαρακτηριστικά, αναγνωριστικό του εκάστοτε VLAN) που αφορούν τα VLAN, δεν θα προβούμε σε περαιτέρω ανάλυση σε αυτό το τεχνικό κομμάτι. Στο πρώτο VLAN που θα δούμε εκεί υπακούουν όλα τα ports.

```
Switch_L3A#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

Switch_L3A(vlan)#show current
VLAN ISL Id: 1
  Name: default
  Media Type: Ethernet
  VLAN 802.10 Id: 100001
  State: Operational
  MTU: 1500
  Translational Bridged VLAN: 1002
  Translational Bridged VLAN: 1003

VLAN ISL Id: 10
  Name: WLAN
  Media Type: Ethernet
  VLAN 802.10 Id: 100010
  State: Operational
  MTU: 1500

VLAN ISL Id: 20
  Name: IT
  Media Type: Ethernet
  VLAN 802.10 Id: 100020
  State: Operational
  MTU: 1500

VLAN ISL Id: 30
  Name: VLAN_admin
  Media Type: Ethernet
  VLAN 802.10 Id: 100030
  State: Operational
  MTU: 1500
```



```
VLAN ISL Id: 40
Name: economic_dep
Media Type: Ethernet
VLAN 802.10 Id: 100040
State: Operational
MTU: 1500

VLAN ISL Id: 50
Name: secretariat
Media Type: Ethernet
VLAN 802.10 Id: 100050
State: Operational
MTU: 1500

VLAN ISL Id: 60
Name: human_resources
Media Type: Ethernet
VLAN 802.10 Id: 100060
State: Operational
MTU: 1500

VLAN ISL Id: 70
Name: analysis_and_design
Media Type: Ethernet
VLAN 802.10 Id: 100070
State: Operational
MTU: 1500

VLAN ISL Id: 80
Name: customer_support
Media Type: Ethernet
VLAN 802.10 Id: 100080
State: Operational
MTU: 1500

VLAN ISL Id: 90
Name: Server_LAN
Media Type: Ethernet
VLAN 802.10 Id: 100090
State: Operational
MTU: 1500
```

Εικόνα 6-68 Είσοδος στη βάση του VLAN

Στη συνέχεια στο σενάριο μας θέλουμε να δημιουργήσουμε ένα vtp domain με vtp server τον Switch_L3A έτσι ώστε να μεταδίδονται τα VLANs και στα υπόλοιπα switches (clients). Σαν vtp domain ορίζουμε ένα αυθαίρετο του τύπου *porgee.gr*

```
Switch_L3A(vlan)#vtp server
Device mode already VTP SERVER.
Switch_L3A(vlan)#vtp domain porgee.gr
Changing VTP domain name from NULL to porgee.gr
Switch_L3A(vlan)#vtp v2-mode
V2 mode enabled.
Switch_L3A(vlan)#apply
APPLY completed.
Switch_L3A(vlan)#exit
APPLY completed.
Exiting...
Switch_L3A#
```

Εικόνα 6-69 Δημιουργία vtp_domain

Αυτό που πρέπει να υλοποιήσουμε μετά είναι το switchport trunk από εκεί δηλαδή που θα μετακινούνται τα VLANs αυτό θα επιτευχθεί με μία σύνδεση από το Switch_L3A προς το Switch_L3B διαμέσων της πόρτας f1/0 όπως θα δούμε στην παρακάτω εικόνα.

```
Switch_L3A#show inter status
Port      Name      Status      Vlan      Duplex  Speed  Type
-----
Fa1/0     notconnect 1          auto     auto    10/100BaseTX
Fa1/1     notconnect 1          auto     auto    10/100BaseTX
Fa1/2     notconnect 1          auto     auto    10/100BaseTX
Fa1/3     notconnect 1          auto     auto    10/100BaseTX
Fa1/4     notconnect 1          auto     auto    10/100BaseTX
Fa1/5     notconnect 1          auto     auto    10/100BaseTX
Fa1/6     notconnect 1          auto     auto    10/100BaseTX
Fa1/7     notconnect 1          auto     auto    10/100BaseTX
Fa1/8     notconnect 1          auto     auto    10/100BaseTX
Fa1/9     notconnect 1          auto     auto    10/100BaseTX
Fa1/10    notconnect 1          auto     auto    10/100BaseTX
Fa1/11    notconnect 1          auto     auto    10/100BaseTX
Fa1/12    notconnect 1          auto     auto    10/100BaseTX
Fa1/13    notconnect 1          auto     auto    10/100BaseTX
Fa1/14    notconnect 1          auto     auto    10/100BaseTX
Fa1/15    notconnect 1          auto     auto    10/100BaseTX
Switch_L3A#
```

Εικόνα 6-70 Πληροφορίες σχετικά με τα status των διεπαφών



```
Switch_L3A#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3A(config)#int fa1/0
Switch_L3A(config-if)#switchport mode trunk
Switch_L3A(config-if)#switchport trunk encapsulation dot1q
Switch_L3A(config-if)#no shut
Switch_L3A(config-if)#desv Sundersh me switch_L3_B
Switch_L3A(config-if)#exit
Switch_L3A(config)#exit
Switch_L3A#
Building configuration...
[OK]
Switch_L3A#
Mar 1 16:16:20.183: %SYS-5-CONFIG_I: Configured from console by console
Switch_L3A#
Switch_L3A#show int status

Port      Name               Status        Vlan    Duplex  Speed Type
-----
Fa1/0     Sundersh me switc notconnect   1       auto   auto 10/100BaseTX
Fa1/1     notconnect        1           auto   auto 10/100BaseTX
Fa1/2     notconnect        1           auto   auto 10/100BaseTX
Fa1/3     notconnect        1           auto   auto 10/100BaseTX
Fa1/4     notconnect        1           auto   auto 10/100BaseTX
Fa1/5     notconnect        1           auto   auto 10/100BaseTX
Fa1/6     notconnect        1           auto   auto 10/100BaseTX
Fa1/7     notconnect        1           auto   auto 10/100BaseTX
Fa1/8     notconnect        1           auto   auto 10/100BaseTX
Fa1/9     notconnect        1           auto   auto 10/100BaseTX
Fa1/10    notconnect        1           auto   auto 10/100BaseTX
Fa1/11    notconnect        1           auto   auto 10/100BaseTX
Fa1/12    notconnect        1           auto   auto 10/100BaseTX
Fa1/13    notconnect        1           auto   auto 10/100BaseTX
Fa1/14    notconnect        1           auto   auto 10/100BaseTX
Fa1/15    notconnect        1           auto   auto 10/100BaseTX
Switch_L3A#
```

Εικόνα 6-71 Επιτυχή σύνδεση με Fa1/0 που θα είναι ο διάυλος επικοινωνίας με τα άλλα switch

Πηγαίνουμε τώρα στο Switch_L3B έτσι ώστε να τον κάνω vtp client και να “τραβάει” τα δεδομένα του Switch_L3A δηλαδή να μεταφερθούν προς αυτόν τα VLANs.

```
Switch_L3B#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3B(config)#
Switch_L3B(config)#hostname Switch_L3B
Switch_L3B(config)#enable secret ccs1
Switch_L3B(config)#line vty 0 4
Switch_L3B(config-line)#password ccs1
Switch_L3B(config-line)#login
Switch_L3B(config-line)#transport input ssh
Switch_L3B(config-line)#exit
Switch_L3B(config)#line con 0
Switch_L3B(config-line)#password ccs1
Switch_L3B(config-line)#login
Switch_L3B(config-line)#exit
Switch_L3B(config)#exit
Switch_L3B#
Switch_L3B#sh run
```

Εικόνα 6-72 Εφαρμογή κωδικών ασφαλείας όπως και στο Switch_L3A

```
Switch_L3B#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

Switch_L3B(vlan)#vtp server
Device mode already VTP SERVER.
Switch_L3B(vlan)#vtp v2-mode
V2 mode enabled.
Switch_L3B(vlan)#vtp password ccs1
Setting device VLAN database password to ccs1.
Switch_L3B(vlan)#vtp domain porgee.gr
Changing VTP domain name from NULL to porgee.gr
Switch_L3B(vlan)#apply
APPLY completed.
Switch_L3B(vlan)#vtp client
Setting device to VTP CLIENT mode.
Switch_L3B(vlan)#exit
In CLIENT state, no apply attempted.
Exiting...
```

Εικόνα 6-73 Μετατροπή του Switch_L3B σε vtp client

Όπως είναι λογικό πρέπει να πάμε και στην πλευρά του Switch_L3B για να ενεργοποιήσουμε το switchport trunk και να υλοποιήσουμε τη σύνδεση(f1/0) με τον Switch_L3A.



```
Switch_L3B#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3B(config)#
Switch_L3B(config)#int fa1/0
Switch_L3B(config-if)#switchport mode trunk
Switch_L3B(config-if)#switchport trunk encapsulation dot1q
Switch_L3B(config-if)#no shut
Switch_L3B(config-if)#desc Sundesh me switchL3_A
Switch_L3B(config-if)#exit
Switch_L3B(config)#exit
Switch_L3B#wr
Switch_L3B#wr
*Mar 1 00:12:13.271: %SYS-5-CONFIG I: Configured from console by console
*Mar 1 00:12:13.739: %DTP-5-TRUNKPORTON: Port Fa1/0 has become dot1q trunk
Switch_L3B#wr
Building configuration...
[OK]
Switch_L3B#
```

Εικόνα 6-74 Ορισμός switchport trunk στον Switch_L3B στο f1/0

```
Switch_L3B#sh int status
Port      Name      Status      Vlan      Duplex  Speed  Type
Fa1/0     Sundesh me switchL connected  trunk    a-full  a-100  10/100BaseTX
Fa1/1     notconnect 1          auto      auto    10/100BaseTX
Fa1/2     notconnect 1          auto      auto    10/100BaseTX
Fa1/3     notconnect 1          auto      auto    10/100BaseTX
Fa1/4     notconnect 1          auto      auto    10/100BaseTX
Fa1/5     notconnect 1          auto      auto    10/100BaseTX
Fa1/6     notconnect 1          auto      auto    10/100BaseTX
Fa1/7     notconnect 1          auto      auto    10/100BaseTX
Fa1/8     notconnect 1          auto      auto    10/100BaseTX
Fa1/9     notconnect 1          auto      auto    10/100BaseTX
Fa1/10    notconnect 1          auto      auto    10/100BaseTX
Fa1/11    notconnect 1          auto      auto    10/100BaseTX
Fa1/12    notconnect 1          auto      auto    10/100BaseTX
Fa1/13    notconnect 1          auto      auto    10/100BaseTX
Fa1/14    notconnect 1          auto      auto    10/100BaseTX
Fa1/15    notconnect 1          auto      auto    10/100BaseTX
Switch_L3B#
```

Εικόνα 6-75 Επιτυχής σύνδεση με τον Switch_L3A

Έχοντας επιτύχει την διασύνδεση μεταξύ των δύο switches πηγαίνουμε στην βάση Vlan του Switch_L3B και βλέπουμε ότι έχουν παρθεί τα VLANs από τον Switch_L3A δυναμικά.

```
Switch_L3B#vlan database
Switch_L3B(vlan)#show current
VLAN ISL Id: 1
Name: default
Media Type: Ethernet
VLAN 802.10 Id: 100001
State: Operational
MTU: 1500
Translational Bridged VLAN: 1002
Translational Bridged VLAN: 1003

VLAN ISL Id: 10
Name: WLAN
Media Type: Ethernet
VLAN 802.10 Id: 100010
State: Operational
MTU: 1500

VLAN ISL Id: 20
Name: IT
Media Type: Ethernet
VLAN 802.10 Id: 100020
State: Operational
```




```
VLAN ISL Id: 30
Name: VLAN_admin
Media Type: Ethernet
VLAN 802.10 Id: 100030
State: Operational
MTU: 1500

VLAN ISL Id: 40
Name: economic_dep
Media Type: Ethernet
VLAN 802.10 Id: 100040
State: Operational
MTU: 1500

VLAN ISL Id: 50
Name: secretariat
Media Type: Ethernet
VLAN 802.10 Id: 100050
State: Operational
MTU: 1500

VLAN ISL Id: 60
Name: human_resources
Media Type: Ethernet
VLAN 802.10 Id: 100060
State: Operational
MTU: 1500

VLAN ISL Id: 70
Name: analysis_and_design
Media Type: Ethernet
VLAN 802.10 Id: 100070
State: Operational
MTU: 1500

VLAN ISL Id: 80
Name: customer_suport
Media Type: Ethernet
VLAN 802.10 Id: 100080
State: Operational
MTU: 1500

VLAN ISL Id: 90
Name: Server LAN
Media Type: Ethernet
VLAN 802.10 Id: 100090
```

Εικόνα 6-76 Επιτυχές πέρασμα των VLANs στον Switch_L3B

```
Switch_L3B#show vtp status
VTP Version          : 2
Configuration Revision : 2
Maximum VLANs supported locally : 68
Number of existing VLANs : 15
VTP Operating Mode   : Client
VTP Domain Name      : porgee.gr
VTP Pruning Mode     : Disabled
VTP V2 Mode          : Enabled
VTP Traps Generation : Disabled
MD5 digest           : 0xA9 0x62 0xF5 0x67 0x02 0xC3 0x15 0x31
```

Εικόνα 6-77 Πληροφορίες κατάστασης Vtp(Client)

Το επόμενο βήμα που πρέπει να κάνουμε είναι να δώσουμε IP στα VLANs έτσι ώστε να έχουμε επιτυχής επικοινωνία μεταξύ των δύο μεταγωγέων που έχουμε.



```
Switch_L3B#conf t
Enter Configuration commands, one per line. End with CNTRL/Z.
Switch_L3B(config)#interface vlan 10
Switch_L3B(config-if)#ip add 173.16.0.3 255.255.255.192
Switch_L3B(config-if)#no shut
Switch_L3B(config-if)#interface vlan 20
Switch_L3B(config-if)#ip add 173.16.0.68 255.255.255.224
Switch_L3B(config-if)#no shut
Switch_L3B(config-if)#interface vlan 30
Switch_L3B(config-if)#ip add 173.16.0.99 255.255.255.224
Switch_L3B(config-if)#no shut
Switch_L3B(config-if)#interface vlan 40
Switch_L3B(config-if)#ip add 173.16.0.131 255.255.255.240
Switch_L3B(config-if)#no shut
Switch_L3B(config-if)#interface vlan 50
Switch_L3B(config-if)#ip add 173.16.0.148 255.255.255.240
Switch_L3B(config-if)#no shut
Switch_L3B(config-if)#interface vlan 60
Switch_L3B(config-if)#ip add 173.16.0.164 255.255.255.240
Switch_L3B(config-if)#no shut
Switch_L3B(config-if)#interface vlan 70
Switch_L3B(config-if)#ip add 173.16.0.179 255.255.255.240
Switch_L3B(config-if)#no shut
Switch_L3B(config-if)#interface vlan 80
Switch_L3B(config-if)#ip add 173.16.0.195 255.255.255.248
Switch_L3B(config-if)#no shut
Switch_L3B(config-if)#interface vlan 90
Switch_L3B(config-if)#ip add 173.16.0.204 255.255.255.248
Switch_L3B(config-if)#no shut
Switch_L3B(config-if)#exit
Switch_L3B(config)#ex
```

Εικόνα 6-78 Απόδοση IP διευθύνσεων

Αυτό φυσικά που πρέπει να κάνουμε για να επιβεβαιωθεί η ορθή επικοινωνία είναι να στείλουμε ICMP πακέτα μέσω ping στις παρακάτω διευθύνσεις που αντιστοιχούν στα VLAN του Switch_L3A

```
Switch_L3A
```

| | | | | | |
|------------------|--------------|-----|--------|----|------|
| FastEthernet1/9 | unassigned | YES | unset | up | down |
| FastEthernet1/10 | unassigned | YES | unset | up | down |
| FastEthernet1/11 | unassigned | YES | unset | up | down |
| FastEthernet1/12 | unassigned | YES | unset | up | down |
| FastEthernet1/13 | unassigned | YES | unset | up | down |
| FastEthernet1/14 | unassigned | YES | unset | up | down |
| FastEthernet1/15 | unassigned | YES | unset | up | down |
| Vlan1 | unassigned | YES | NVRAM | up | up |
| Vlan10 | 173.16.0.2 | YES | manual | up | up |
| Vlan20 | 173.16.0.67 | YES | manual | up | up |
| Vlan30 | 173.16.0.97 | YES | manual | up | up |
| Vlan40 | 173.16.0.130 | YES | manual | up | up |
| Vlan50 | 173.16.0.147 | YES | manual | up | up |
| Vlan60 | 173.16.0.163 | YES | manual | up | up |
| Vlan70 | 173.16.0.177 | YES | manual | up | up |
| Vlan80 | 173.16.0.194 | YES | manual | up | up |
| Vlan90 | 173.16.0.203 | YES | manual | up | up |

Εικόνα 6-79 Διευθύνσεις που θα γίνουν ping από τον Switch_L3B



```
Switch_L3B#ping 173.16.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/29/40 ms
Switch_L3B#ping 173.16.0.67
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.67, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 20/28/36 ms
Switch_L3B#ping 173.16.0.97
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.97, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 20/29/36 ms
Switch_L3B#ping 173.16.0.130
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.130, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 24/29/40 ms
Switch_L3B#ping 173.16.0.147
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.147, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 28/32/36 ms
Switch_L3B#ping 173.16.0.163
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.163, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 24/30/40 ms
Switch_L3B#

Switch_L3B#ping 173.16.0.177
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.177, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 24/30/36 ms
Switch_L3B#ping 173.16.0.194
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.194, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 16/28/44 ms
Switch_L3B#ping 173.16.0.203
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.203, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 24/31/40 ms
Switch_L3B#
```

Εικόνα 6-80 Επιτυχείς προσπάθειες μετάδοσης πακέτων

Στη συνέχεια του σεναρίου που δουλεύουμε θέλουμε να κάνουμε τον ένα μεταγωγέα Switch_L3B να είναι ικανός να αναλάβει την λειτουργία του δικτύου αν συμβεί κάποια δυσλειτουργία στον Switch_L3A. Αυτό θα το επιτύχουμε με την υλοποίηση του *HSRP (Hot Standby Router Protocol)* το οποίο λειτουργεί ιδανικά σε τέτοιες καταστάσεις. Έτσι ο ένας μεταγωγέας θα είναι πάντα ο ενεργός και ο άλλος θα περιμένει. Θέτωνα priority 255 από το VLAN10 μέχρι το VLAN40 στο Switch_L3A το βάζουμε ενεργό switch για αυτά τα VLAN. Με την εντολή preempt το switch όταν διορθωθεί η δυσλειτουργία του αναλαμβάνει ξανά την κυριότητα. Έχοντας priority 1 τα VLAN 50 έως VLAN 90 στο Switch_L3A δεν αναλαμβάνουν προτεραιότητα. Το ακριβώς αντίθετο σενάριο θα ρυθμιστεί στον Switch_L3B στη συνέχεια έτσι ώστε να έχουμε κάλυψη στο θέμα των δυσλειτουργιών. Παραθέτουμε την ακολουθία του κώδικα που χρησιμοποιήσαμε για να το επιτύχουμε.



```
Switch_L3A(config-if)#standby 10 ip 173.16.0.1
Switch_L3A(config-if)#standby
*Mar 1 01:10:49.263: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Speak -> Standby
*Mar 1 01:10:49.763: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Standby -> Active
Switch_L3A(config-if)#standby 10 priority 255
Switch_L3A(config-if)#standby 10 preempt
Switch_L3A(config-if)#
Switch_L3A#
*Mar 1 01:12:06.367: %SYS-5-CONFIG_I: Configured from console by console
Switch_L3A#wr
Building configuration...
[OK]
Switch_L3A#conf t
Enter Configuration commands, one per line. End with CNTL/Z.
Switch_L3A(config)#interface vlan 20
Switch_L3A(config-if)#standby 20 ip 173.16.0.65
Switch_L3A(config-if)#standby 20 priority 255
Switch_L3A(config-if)#standby 20 preempt
Switch_L3A(config-if)#standby
Switch_L3A(config-if)#
% Incomplete command.

Switch_L3A(config-if)#
*Mar 1 01:17:01.847: %HSRP-5-STATECHANGE: Vlan20 Grp 20 state Speak -> Standby
*Mar 1 01:17:02.347: %HSRP-5-STATECHANGE: Vlan20 Grp 20 state Standby -> Active
```

Εικόνα 6-81 Standby λειτουργία προτεραιότητα 255

Βλέπουμε ότι ενεργοποιείται σωστά η λειτουργία standby του HSRP ορίζοντας μία IP από το range του εκάστοτε VLAN. Στην εικόνα φαίνεται για το VLAN 10 και για το VLAN 20, ομοίως θα ρυθμίσουμε τη λειτουργία standby και στα VLAN 30 και VLAN 40.

```
Switch_L3A(config)#interface vlan 80
Switch_L3A(config-if)#standby 80 ip 173.16.0.196
Switch_L3A(config-if)#standby 80 priority 1
Switch_L3A(config-if)#exit
Switch_L3A(config)#interface vlan 90
Switch_L3A(config-if)#standby 90 ip 173.16.0.205
Switch_L3A(config-if)#standby 90 priority 1
Switch_L3A(config-if)#
```

Εικόνα 6-82 Standby λειτουργία προτεραιότητα 1

Το ίδιο έχει συμβεί για τα VLAN 50,60,70 απλά επιλέγουμε να εμφανίσουμε αυτό το στιγμιότυπο οθόνης.

Τώρα πηγαίνουμε στο Switch_L3B για να κάνουμε την αντίθετη λειτουργία δηλαδή ορίζουμε priority 1 στα πρώτα 4 VLANS και στα υπόλοιπα 5 priority 255 για να γίνει ενεργό switch στην ομάδα με priority 255. Στο παρακάτω στιγμιότυπο φαίνεται για τα πρώτα δύο VLANs, ομοίως κάνουμε και για τα υπόλοιπα 2.



```
Switch_L3B#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3B(config)#interface vlan 10
Switch_L3B(config-if)#standby 10 ip 173.16.0.1
Switch_L3B(config-if)#standby 10 priority 1
Switch_L3B(config-if)#
*Mar 1 02:30:47.119: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Speak -> Standby
Switch_L3B(config-if)#exit
Switch_L3B(config)#interface vlan 20
Switch_L3B(config-if)#standby 20 ip 173.16.0.65
Switch_L3B(config-if)#standby 20 priority 1
Switch_L3B(config-if)#
*Mar 1 02:31:46.131: %HSRP-5-STATECHANGE: Vlan20 Grp 20 state Speak -> Standby
Switch_L3B(config-if)#
```

Εικόνα 6-83 Standby λειτουργία στον Switch_L3B προτεραιότητα 1

Εδώ έχοντας ρυθμίσει το priority 255 στα VLAN 50 έως VLAN 90 βλέπουμε ότι η κατάσταση Listen έχει αλλάξει σε active. Στο παρακάτω στιγμιότυπο φαίνεται για τα πρώτα δύο VLANs, ομοίως κάνουμε και για τα υπόλοιπα 3.

```
Switch_L3B(config)#interface vlan 50
Switch_L3B(config-if)#standby 50 ip 173.16.0.145
Switch_L3B(config-if)#standby 50 priority 255
Switch_L3B(config-if)#standby 50 preempt
Switch_L3B(config-if)#
*Mar 1 02:41:02.159: %HSRP-5-STATECHANGE: Vlan50 Grp 50 state Listen -> Active
Switch_L3B(config-if)#exit
Switch_L3B(config)#interface vlan 60
Switch_L3B(config-if)#standby 60 ip 173.16.0.161
Switch_L3B(config-if)#standby 60 priority 255
Switch_L3B(config-if)#standby 60 preempt
Switch_L3B(config-if)#
*Mar 1 02:42:20.107: %HSRP-5-STATECHANGE: Vlan60 Grp 60 state Listen -> Active
Switch_L3B(config-if)#
```

Εικόνα 6-84 Standby λειτουργία στον Switch_L3B προτεραιότητα 255

Πηγαίνοντας τώρα στο κάθε switch με την εκτέλεση της παρακάτω εντολής με την οποία εμφανίζονται τα standby interfaces, βλέπουμε ότι έχουμε επιτύχει το επιθυμητό όπου τα VLAN 10 έως VLAN 40 είναι local και ενεργά στο Switch_L3A ενώ είναι standby στις διευθύνσεις που έχουμε ορίσει στο Switch_L3B έτσι ώστε σε περίπτωση κάποιας ανεπιθύμητης λειτουργίας να μην χαθεί η επικοινωνία. Η αντίθετη λειτουργία γίνεται όπως έχουμε αναφέρει και πιο πάνω στο Switch_L3B. Το πρωτόκολλο λειτουργεί κανονικά και προχωρούμε παρακάτω στο σενάριο μας.

```
Switch_L3A#show standby brief
P indicates configured to preempt.
|
Interface Grp Prio P State Active Standby Virtual IP
V110 10 255 P Active local 173.16.0.3 173.16.0.1
V120 20 255 P Active local 173.16.0.68 173.16.0.65
V130 30 255 P Active local 173.16.0.99 173.16.0.98
V140 40 255 P Active local 173.16.0.131 173.16.0.129
V150 50 1 Standby 173.16.0.148 local 173.16.0.145
V160 60 1 Standby 173.16.0.164 local 173.16.0.161
V170 70 1 Standby 173.16.0.179 local 173.16.0.180
V180 80 1 Standby 173.16.0.195 local 173.16.0.196
V190 90 1 Standby 173.16.0.204 local 173.16.0.205
Switch_L3A#
```



```
Switch_L3A#show standby brief
P indicates configured to preempt.
 |
Interface Grp Prio P State Active Standby Virtual IP
V110 10 1 Standby 173.16.0.2 local 173.16.0.1
V120 20 1 Standby 173.16.0.67 local 173.16.0.65
V130 30 1 Standby 173.16.0.97 local 173.16.0.98
V140 40 1 Standby 173.16.0.130 local 173.16.0.129
V150 50 255 P Active local 173.16.0.147 173.16.0.145
V160 60 255 P Active local 173.16.0.163 173.16.0.161
V170 70 255 P Active local 173.16.0.177 173.16.0.180
V180 80 255 P Active local 173.16.0.194 173.16.0.196
V190 90 255 P Active local 173.16.0.203 173.16.0.205
Switch_L3A#
```

Εικόνα 6-85 Λειτουργία Standby (HSRP)

Το επόμενο βήμα το οποίο θα κάνουμε είναι να δημιουργήσουμε τα αντίστοιχα dhcp pools για το κάθε VLAN έτσι ώστε οι χρήστες να παίρνουν δυναμικά IP διεύθυνση. Εκτελώντας το παρακάτω τμήμα κώδικα σε όλα τα VLANs ανάλογα βεβαίως την IP που έχει το υποδίκτυο σύμφωνα με το subnetting που υλοποιήσαμε. Πιο κάτω φαίνεται ένα στιγμιότυπο οθόνης που εκτελείται το dhcp pool στο τμήμα WLAN.

```
Switch_L3A#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3A(config)#ip dhcp pool WLAN
Switch_L3A(dhcp-config)#network 172.16.0.0 255.255.255.192
Switch_L3A(dhcp-config)#default-router 172.16.0.1
Switch_L3A(dhcp-config)#exit
```

Εικόνα 6-86 DHCP pool στα VLANs

Στο παρακάτω screenshot βλέπουμε τα pools που έχουμε δημιουργήσει

```
Switch_L3A#sh ip dhcp pool
Pool WLAN :
  Utilization Marks (high/low) : 100 / 0
  Subnet size (first/next) : 0 / 0
  Total addresses : 64
  Leased addresses : 0
  Pending events : 0
  1 subnet is currently in the pool :
  Current index : IP address range : Leased addresses
  173.16.0.1 173.16.0.1 173.16.0.64 0
Pool IT :
  Utilization Marks (high/low) : 100 / 0
  Subnet size (first/next) : 0 / 0
  Total addresses : 30
  Leased addresses : 0
  Pending events : 0
  1 subnet is currently in the pool :
  Current index : IP address range : Leased addresses
  173.16.0.65 173.16.0.65 - 173.16.0.94 0
Pool ACCOUNTING :
  Utilization Marks (high/low) : 100 / 0
  Subnet size (first/next) : 0 / 0
  Total addresses : 14
  Leased addresses : 0
  Pending events : 0
  1 subnet is currently in the pool :
  Current index : IP address range : Leased addresses
  173.16.0.113 173.16.0.113 - 173.16.0.126 0
Pool SECRETARIAT :
  Utilization Marks (high/low) : 100 / 0
  Subnet size (first/next) : 0 / 0
  Total addresses : 14
  Leased addresses : 0
  Pending events : 0
  1 subnet is currently in the pool :
  Current index : IP address range : Leased addresses
  173.16.0.129 173.16.0.129 - 173.16.0.142 0
```

Εικόνα 6-87 Δημιουργημένα pools 1/2



```
pool human_resources :
Utilization mark (high/low) : 100 / 0
Subnet size (first/next) : 0 / 0
Total addresses : 14
Leased addresses : 0
Pending event : none
1 subnet is currently in the pool :
Current index IP address range Leased addresses
173.16.0.145 173.16.0.145 - 173.16.0.158 0

pool analysis_and_design :
Utilization mark (high/low) : 100 / 0
Subnet size (first/next) : 0 / 0
Total addresses : 14
Leased addresses : 0
Pending event : none
1 subnet is currently in the pool :
Current index IP address range Leased addresses
173.16.0.161 173.16.0.161 - 173.16.0.174 0

pool customer_support :
Utilization mark (high/low) : 100 / 0
Subnet size (first/next) : 0 / 0
Total addresses : 6
Leased addresses : 0
Pending event : none
1 subnet is currently in the pool :
Current index IP address range Leased addresses
173.16.0.177 173.16.0.177 - 173.16.0.182 0

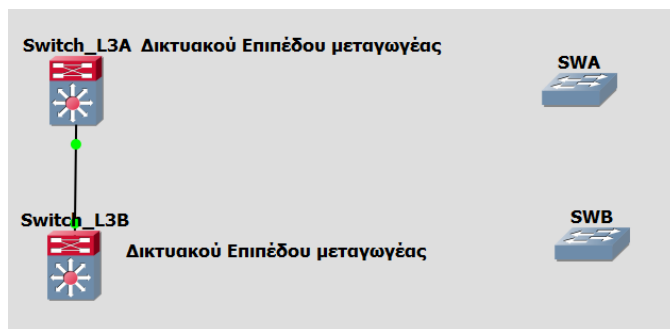
pool VLAN_admin :
Utilization mark (high/low) : 100 / 0
Subnet size (first/next) : 0 / 0
Total addresses : 0
Leased addresses : 0
Pending event : none
0 subnet is currently in the pool :

pool Server_IAM :
Utilization mark (high/low) : 100 / 0
Subnet size (first/next) : 0 / 0
Total addresses : 6
Leased addresses : 0
Pending event : none
1 subnet is currently in the pool :
Current index IP address range Leased addresses
173.16.0.193 173.16.0.193 - 173.16.0.198 0
switch_L3A
```

Εικόνα 6-88 Δημιουργημένα pools 2/2

Για να μην βγούμε από το σενάριο που ακολουθούμε γράφουμε ακριβώς τον ίδιο κώδικα και στον Switch_L3B που θα εργαστεί σε περίπτωση που συμβεί κάποια βλάβη.

Συνεχίζοντας θα προχωρήσουμε στην εγκατάσταση των μεταγωγέων προκειμένου να διαβιβαστούν τα δεδομένα προς τους servers. Θα χρειαστούν δύο μεταγωγείς για να καλύψουμε την περίπτωση ενδεχόμενης βλάβης που αναφέραμε πιο πάνω.



Εικόνα 6-89 Επίπεδο πρόσβασης



```
SWA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SWA (config)#hostname SWA
SWA (config)#enable secret ccs1
SWA (config)#line vty 0 1340
SWA (config-line)#password ccs1
SWA (config-line)#login
SWA (config-line)#transport input ssh
SWA (config-line)#exit
SWA (config)#line con 0
SWA (config-line)#password ccs1
SWA (config-line)#login
SWA (config-line)#exit
SWA (config)#exit
SWA (config)#exit
SWA#
SWA#sh run
```

Εικόνα 6-90 Βασική παραμετροποίηση SWA

Στη συνέχεια έχοντας ενεργοποιήσει το VLAN_Admin(VLAN 30) δίνουμε IP στο συγκεκριμένο VLAN.

```
SWA#vlan database
SWA (vlan)#vlan 30 name VLAN_admin
VLAN 30 added:
  Name: VLAN_admin
SWA (vlan)#vlan 30 state active
VLAN 30 modified:
  State ACTIVE

SWA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SWA (config)#interface vlan 30
SWA (config-if)#ip add 173.16.0.100 255.255.255.224
SWA (config-if)#no shut
SWA (config-if)#exit
SWA (config)#exit
```

Εικόνα 6-91 Προσθήκη και απόδοση IP στο VLAN_admin

Στη συνέχεια ενεργοποιούμε το vtr και κάνουμε πελάτη το SWA για να τραβήξει τις πληροφορίες των VLAN από το διακομιστή vtr που έχουμε ορίσει πιο πάνω, στο SWA trunk θα είναι η πόρτα fa1/0 και στο Switch_L3A θα είναι η fa1/1 που θα μεταβιβάσει τα VLAN.

| Interface | IP-Address | OK? | Method | Status | Protocol |
|------------------|--------------|-----|--------|-----------------------|----------|
| FastEthernet0/0 | unassigned | YES | unset | administratively down | down |
| FastEthernet0/1 | unassigned | YES | unset | administratively down | down |
| FastEthernet1/0 | unassigned | YES | unset | up | down |
| FastEthernet1/1 | unassigned | YES | unset | up | down |
| FastEthernet1/2 | unassigned | YES | unset | up | down |
| FastEthernet1/3 | unassigned | YES | unset | up | down |
| FastEthernet1/4 | unassigned | YES | unset | up | down |
| FastEthernet1/5 | unassigned | YES | unset | up | down |
| FastEthernet1/6 | unassigned | YES | unset | up | down |
| FastEthernet1/7 | unassigned | YES | unset | up | down |
| FastEthernet1/8 | unassigned | YES | unset | up | down |
| FastEthernet1/9 | unassigned | YES | unset | up | down |
| FastEthernet1/10 | unassigned | YES | unset | up | down |
| FastEthernet1/11 | unassigned | YES | unset | up | down |
| FastEthernet1/12 | unassigned | YES | unset | up | down |
| FastEthernet1/13 | unassigned | YES | unset | up | down |
| FastEthernet1/14 | unassigned | YES | unset | up | down |
| FastEthernet1/15 | unassigned | YES | unset | up | down |
| Vlan1 | unassigned | YES | unset | up | down |
| Vlan30 | 173.16.0.100 | YES | manual | up | down |

Εικόνα 6-92 Πληροφορίες των interfaces του SWA



```

SWA
Translational Bridged VLAN: 1003

VLAN ISL Id: 10
Name: WLAN
Media Type: Ethernet
VLAN 802.10 Id: 100010
State: Operational
MTU: 1500

VLAN ISL Id: 20
Name: IT
Media Type: Ethernet
VLAN 802.10 Id: 100020
State: Operational
MTU: 1500

VLAN ISL Id: 30
Name: VLAN admin
Media Type: Ethernet
VLAN 802.10 Id: 100030
State: Operational
MTU: 1500

VLAN ISL Id: 40
Name: economic dep
Media Type: Ethernet
VLAN 802.10 Id: 100040
State: Operational
MTU: 1500

VLAN ISL Id: 50
Name: secretariat
Media Type: Ethernet
VLAN 802.10 Id: 100050
State: Operational

```

Εικόνα 6-93 Επιτυχές πέρασμα των VLANs

Επίσης θα πρέπει να δημιουργήσουμε και ένα σύνδεσμο με το Switch_L3B που στην προκειμένη είναι το f1/1 στο SWA και στον Switch_L3B η ίδια δηλαδή η f1/1. Έτσι ώστε να επιτύχουμε σύνδεση και με τον μεταγωγέα που ουσιαστικά κρατάει backup.

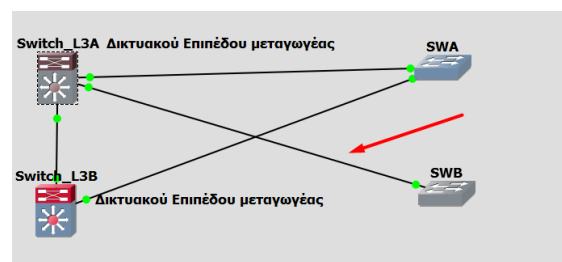
```

Switch_L3B#
Switch_L3B#
*Mar 1 00:00:37.727: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan60, changed state to down
*Mar 1 00:00:37.727: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan70, changed state to down
*Mar 1 00:00:37.727: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan80, changed state to down
*Mar 1 00:00:37.727: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan90, changed state to down
Switch_L3B#
*Mar 1 00:00:44.947: %HSRP-5-STATECHANGE: Vlan80 Grp 80 state Listen -> Active
*Mar 1 00:00:44.991: %HSRP-5-STATECHANGE: Vlan60 Grp 60 state Listen -> Active
*Mar 1 00:00:44.999: %HSRP-5-STATECHANGE: Vlan50 Grp 50 state Listen -> Active
*Mar 1 00:00:45.039: %HSRP-5-STATECHANGE: Vlan90 Grp 90 state Listen -> Active
*Mar 1 00:00:45.055: %HSRP-5-STATECHANGE: Vlan70 Grp 70 state Listen -> Active
Switch_L3B#sh interface status
Switch_L3B#sh interface status
Port Name Status Vlan Duplex Speed Type
Fa1/0 link with SWA connected trunk a-full a-100 10/100BaseTX
Fa1/1 link with SWL3B connected trunk a-full a-100 10/100BaseTX
Fa1/2 notconnect 1 auto auto 10/100BaseTX
Fa1/3 notconnect 1 auto auto 10/100BaseTX
Fa1/4 notconnect 1 auto auto 10/100BaseTX
Fa1/5 notconnect 1 auto auto 10/100BaseTX
Fa1/6 notconnect 1 auto auto 10/100BaseTX

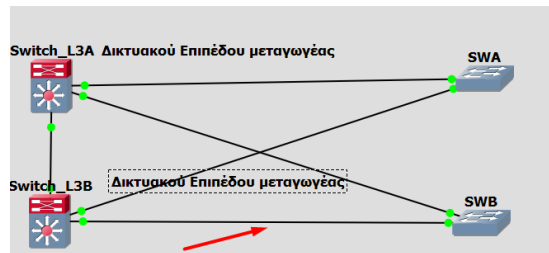
```

Εικόνα 6-94 Σύνδεση με τον backup μεταγωγέα

Έπειτα θα πρέπει να εργαστούμε στο ίδιο μοτίβο που υλοποιήσαμε τον μεταγωγέα SWA και στον μεταγωγέα SWB με την μόνη διαφορά ότι θα πρέπει να έχει IP 173.16.0.101 και subnet mask 255.255.255.224



Εικόνα 6-95 Δημιουργία σύνδεσης με τον Switch_L3A



Εικόνα 6-96 Δημιουργία σύνδεσης με τον Switch_L3B

```
SWB
User Access Verification
Password:
SWB#show interface status

```

| Port | Name | Status | Vlan | Duplex | Speed | Type |
|-------|-----------------|------------|-------|--------|-------|--------------|
| Ea1/0 | link with SWL3A | connected | trunk | a-full | a-100 | 10/100BaseTX |
| Ea1/1 | link with SWL3B | connected | trunk | a-full | a-100 | 10/100BaseTX |
| Ea1/2 | | notconnect | 1 | auto | auto | 10/100BaseTX |
| Ea1/3 | | notconnect | 1 | auto | auto | 10/100BaseTX |
| Ea1/4 | | notconnect | 1 | auto | auto | 10/100BaseTX |
| Ea1/5 | | notconnect | 1 | auto | auto | 10/100BaseTX |

Εικόνα 6-97 Σύνδεση με τα Switch level 3

Η σύνδεση έχει επιτύχει έχουμε κάνει πελάτη το SWB για να τραβήξει τις πληροφορίες των VLAN από το διακομιστή vtr που έχουμε ορίσει πιο πάνω.

Στη συνέχεια το βήμα που θα κάνουμε είναι να παραμετροποιήσουμε το δίκτυο χρησιμοποιώντας το πρωτόκολλο του *Spanning Tree*. Το πρωτόκολλο αυτό όπως έχει αναφερθεί και στο θεωρητικό υπόβαθρο στα παραπάνω κεφάλαια επιλέγει ρίζα by default την MAC address. Εμείς θα ορίσουμε root bridge με βάση το σενάριο που δουλεύουμε, δηλαδή θα κάνουμε root bridge για συγκεκριμένα VLANs τον SWITCH_L3A και για κάποια άλλα τον SWITCH_L3B ακολουθώντας την λογική του HSRP.

Πηγαίνουμε στον μεταγωγέα Switch_L3A και εκτελούμε τον παρακάτω κώδικα, στα VLAN που θα ορίσουμε τιμή 0 θέτουμε σαν ρίζα τον μεταγωγέα Switch_L3A. Οι τιμές που ορίζονται τα priority είναι από 0-61440. Επιλέγουμε να θέσουμε τα VLAN που δεν θα έχουνε ρίζα τον μεταγωγέα Switch_L3A να παίρνουν την τιμή 8192.

```
Switch_L3A#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3A(config)#spanning-tree vlan 10 priority 0
Switch_L3A(config)#spanning-tree vlan 20 priority 0
Switch_L3A(config)#spanning-tree vlan 30 priority 0
Switch_L3A(config)#spanning-tree vlan 40 priority 0
Switch_L3A(config)#spanning-tree vlan 50 priority 8192
Switch_L3A(config)#spanning-tree vlan 60 priority 8192
Switch_L3A(config)#spanning-tree vlan 70 priority 8192
Switch_L3A(config)#spanning-tree vlan 80 priority 8192
Switch_L3A(config)#spanning-tree vlan 90 priority 8192
Switch_L3A(config)#
```

Εικόνα 6-98 Spanning Tree priority SWL3A



Η αντίστροφη διαδικασία πρέπει να γίνει τώρα στην πλευρά του Switch_L3B

```
Switch_L3B#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3B(config)#spanning-tree vlan 10 priority 8192
Switch_L3B(config)#spanning-tree vlan 20 priority 8192
Switch_L3B(config)#spanning-tree vlan 30 priority 8192
Switch_L3B(config)#spanning-tree vlan 40 priority 8192
Switch_L3B(config)#spanning-tree vlan 50 priority 0
Switch_L3B(config)#spanning-tree vlan 60 priority 0
Switch_L3B(config)#spanning-tree vlan 70 priority 0
Switch_L3B(config)#spanning-tree vlan 80 priority 0
Switch_L3B(config)#spanning-tree vlan 90 priority 0
Switch_L3B(config)#exit
Switch_L3B#wr
Building configuration...
[OK]
```

Εικόνα 6-99 Spanning Tree priority SWL3B

Βλέπουμε τα νέα priority σε κάθε VLAN

```
VLAN30
Spanning tree enabled protocol ieee
Root ID    Priority    0
Address    c401.1fe4.0003
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

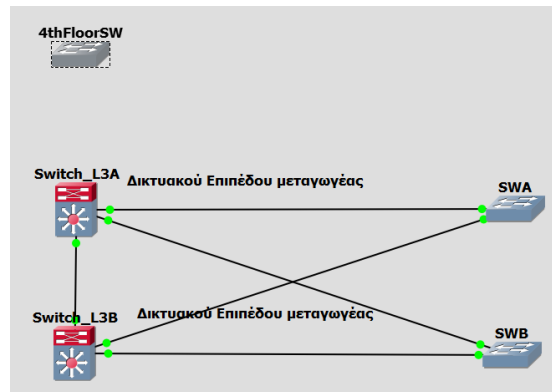
Bridge ID  Priority    0
Address    c401.1fe4.0003
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300

VLAN70
Spanning tree enabled protocol ieee
Root ID    Priority    0
Address    c402.3b70.0007
Cost       19
Port       41 (FastEthernet1/0)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID  Priority    8192
Address    c401.1fe4.0007
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
```

Εικόνα 6-100 Διαφορετικές προτεραιότητες στα VLANs

Το επίπεδο που θα δουλέψουμε τώρα είναι το *επίπεδο πρόσβασης* του οργανισμού μας. Θα πρέπει σε κάθε όροφο που έχουμε ορίσει να βάλουμε και εκεί ένα μεταγωγέα και ουσιαστικά να ρυθμίζουμε την κίνηση τα ranges δηλαδή των VLANs.



Εικόνα 6-101 Εισαγωγή μεταγωγέων πρόσβασης για τους ορόφους

4ος Όροφος(Customer Support & WAP)

Ακολουθούμε παρόμοια διαδικασία για την ρύθμιση κωδικών πρόσβασης όπως και στους υπόλοιπους μεταγωγείς και την διαδικασία του switchport mode trunk παρομοίως.

```
4thFloorSW#show interface status
Port      Name          Status      Vlan    Duplex  Speed  Type
Fa1/0     Sundesh me switchL connected   trunk   a-full a-100  10/100BaseTX
Fa1/1     Sundesh me switchL connected   trunk   a-full a-100  10/100BaseTX
Fa1/2     notconnect   1          auto    auto   10/100BaseTX
Fa1/3     notconnect   1          auto    auto   10/100BaseTX
Fa1/4     notconnect   1          auto    auto   10/100BaseTX
Fa1/5     notconnect   1          auto    auto   10/100BaseTX
```

Εικόνα 6-102 Σύνδεση με Switch_L3A και Switch_L3B

Έχοντας κάνει την σύνδεση και τον ορισμό του VTP σαν client προχωρούμε στη ρύθμιση των ranges. Στον 4ο όροφο σύμφωνα με το σενάριο διάρθρωσης της εταιρείας έχουμε το τμήμα Υποστήριξης πελατών και ένα σημείο πρόσβασης WLAN. Έτσι ανάλογα με τις απαιτήσεις που έχουμε θέσει προχωρούμε στη ρύθμιση των ranges όπως φαίνεται παρακάτω.

```
4thFloorSW(config)#interface range fa1/2
4thFloorSW(config-if-range)#switchport mode access
4thFloorSW(config-if-range)#switchport access vlan 10
4thFloorSW(config-if-range)#no shut
4thFloorSW(config-if-range)#exit
4thFloorSW(config)#exit
4thFloorSW#
*Mar 1 00:02:32.355: %SYS-5-CONFIG_I: Configured from console by console
4thFloorSW#conf t
Enter configuration commands, one per line. End with CNTL/Z.
4thFloorSW(config)#interface range fa1/3 - 6
4thFloorSW(config-if-range)#switchport mode access
4thFloorSW(config-if-range)#switchport access vlan 80
4thFloorSW(config-if-range)#no shut
4thFloorSW(config-if-range)#exit
4thFloorSW(config)#exit
```

Εικόνα 6-103 Καθορισμός ranges ανάλογα με τα τμήματα του κάθε ορόφου



Στις υπόλοιπες πόρτες που δεν θέλω να έχουν πρόσβαση τις κλείνουμε

```
4thFloorSW(config)#interface range fa1/7 - 15
4thFloorSW(config-if-range)#shutdown
4thFloorSW(config-if-range)#exit
4thFloorSW(config)#
*Mar 1 00:03:53.451: %LINK-5-CHANGED: Interface FastEthernet1/7, changed state to administratively down
*Mar 1 00:03:53.455: %LINK-5-CHANGED: Interface FastEthernet1/8, changed state to administratively down
*Mar 1 00:03:53.463: %LINK-5-CHANGED: Interface FastEthernet1/9, changed state to administratively down
*Mar 1 00:03:53.471: %LINK-5-CHANGED: Interface FastEthernet1/10, changed state to administratively down
*Mar 1 00:03:53.475: %LINK-5-CHANGED: Interface FastEthernet1/11, changed state to administratively down
*Mar 1 00:03:53.479: %LINK-5-CHANGED: Interface FastEthernet1/12, changed state to administratively down
4thFloorSW(config)#
*Mar 1 00:03:53.479: %LINK-5-CHANGED: Interface FastEthernet1/13, changed state to administratively down
*Mar 1 00:03:53.479: %LINK-5-CHANGED: Interface FastEthernet1/14, changed state to administratively down
*Mar 1 00:03:53.479: %LINK-5-CHANGED: Interface FastEthernet1/15, changed state to administratively down
4thFloorSW(config)#exit
```

Εικόνα 6-104 Περιορισμός πρόσβασης ανάλογα με το VLAN του κάθε ορόφου

```
interface FastEthernet1/1
description Sundesh me switchL3_B
switchport mode trunk
!
interface FastEthernet1/2
switchport access vlan 10
!
interface FastEthernet1/3
switchport access vlan 80
!
interface FastEthernet1/4
switchport access vlan 80
!
interface FastEthernet1/5
switchport access vlan 80
!
interface FastEthernet1/6
switchport access vlan 80
!
interface FastEthernet1/7
shutdown
!
interface FastEthernet1/8
shutdown
!
interface FastEthernet1/9
shutdown
```

Εικόνα 6-105 Πόρτες που έχουν πρόσβαση

Η διαδικασία που έγινε στον 4^ο όροφο πρέπει να γίνει και στους υπόλοιπους 3 ορόφους ανάλογα κάθε φορά με τα τμήματα που έχουμε στον κάθε όροφο. Θα ακολουθήσουν πιο συγκεντρωτικά στιγμιότυπα οθόνης από τον κάθε όροφο.

3^{ος} Όροφος(Analysis_and_design & IT&Server_LAN&WAP)



```
3rdFloorSW#conf t
Enter configuration commands, one per line. End with CNTL/Z.
3rdFloorSW(config)#interface range fal/2
3rdFloorSW(config-if-range)#switchport mode access
3rdFloorSW(config-if-range)#switchport access vlan 10
3rdFloorSW(config-if-range)#no shut
3rdFloorSW(config-if-range)#exit
3rdFloorSW(config)#conf t
^
% Invalid input detected at '^' marker.

3rdFloorSW(config)#interface range fal/3 - 7
3rdFloorSW(config-if-range)#switchport mode access
3rdFloorSW(config-if-range)#switchport access vlan 70
3rdFloorSW(config-if-range)#no shut
3rdFloorSW(config-if-range)#EXIT
3rdFloorSW(config)#interface range fal/8 - 15
3rdFloorSW(config-if-range)#switchport mode access
3rdFloorSW(config-if-range)#switchport access vlan 20
3rdFloorSW(config-if-range)#no shut
3rdFloorSW(config-if-range)#

3rdFloorSW(config)#interface range fa2/0 - 5
3rdFloorSW(config-if-range)#switchport mode access
3rdFloorSW(config-if-range)#switchport access vlan 90
3rdFloorSW(config-if-range)#no shut
3rdFloorSW(config-if-range)#
```

Εικόνα 6-106 Ranges 3^{ου} ορόφου ανάλογα με τα τμήματα

```
description Sundesh Hc Switch3_0
switchport mode trunk
!
interface FastEthernet1/2
switchport access vlan 10
!
interface FastEthernet1/3
switchport access vlan 70
!
interface FastEthernet1/4
switchport access vlan 70
!
interface FastEthernet1/5
switchport access vlan 70
!
interface FastEthernet1/6
switchport access vlan 70
!
interface FastEthernet1/7
switchport access vlan 70
!
interface FastEthernet1/8
switchport access vlan 20
!
interface FastEthernet1/9
switchport access vlan 20
!
interface FastEthernet1/10
switchport access vlan 20
!
interface FastEthernet1/11
switchport access vlan 20
!
interface FastEthernet1/12
switchport access vlan 20
!
interface FastEthernet1/13
switchport access vlan 20
!
interface FastEthernet1/14
switchport access vlan 20
!
interface FastEthernet1/15
switchport access vlan 20
!
interface FastEthernet2/0
switchport access vlan 90
!
interface FastEthernet2/1
shutdown
!
interface FastEthernet2/5
switchport access vlan 90
!
!
interface FastEthernet2/6
shutdown
!
interface FastEthernet2/7
shutdown
!
!
interface FastEthernet2/8
shutdown
!
!
interface FastEthernet2/9
shutdown
!
!
interface FastEthernet2/10
shutdown
!
!
interface FastEthernet2/11
shutdown
!
!
interface FastEthernet2/12
shutdown
!
!
interface FastEthernet2/13
shutdown
!
!
interface FastEthernet2/14
shutdown
!
!
interface FastEthernet2/15
shutdown
```

Εικόνα 6-107 Access ports

Το ίδιο γίνεται για τον 2^ο όροφο και για τον 1^ο όροφο ελέγχοντας πάντα τα range των τμημάτων και ποια VLAN θα είναι σε κάθε όροφο ακολουθώντας το σενάριο που προσομοιώνουμε.

2^{ος} Όροφος(Human Resources & Economic Department & WAP)



```
2ndFloorSW#conf t
Enter configuration commands, one per line. End with CNTL/Z.
2ndFloorSW(config)#interface range fa1/2
2ndFloorSW(config-if-range)#switchport mode access
2ndFloorSW(config-if-range)#switchport access vlan 10
2ndFloorSW(config-if-range)#no shut
2ndFloorSW(config-if-range)#exit
2ndFloorSW(config)#interface range fa1/3 - 8
2ndFloorSW(config-if-range)#switchport mode access
2ndFloorSW(config-if-range)#switchport access vlan 60
2ndFloorSW(config-if-range)#no shut
2ndFloorSW(config-if-range)#exit
2ndFloorSW(config)#interface range fa1/8 - 15
2ndFloorSW(config-if-range)#switchport mode access
2ndFloorSW(config-if-range)#switchport access vlan 40
2ndFloorSW(config-if-range)#no shut
2ndFloorSW(config-if-range)#exit
2ndFloorSW(config)#exit
2ndFloorSW#w
*Mar 1 00:07:57.579: %SYS-5-CONFIG_I: Configured from console by console
2ndFloorSW#wr
Building configuration...
[OK]
```

Εικόνα 6-108 Ranges 2^{ου} ορόφου ανάλογα με τα τμήματα

```
interface FastEthernet1/0
description Sundesh me switchL3_A
switchport mode trunk
!
interface FastEthernet1/1
description Sundesh me switchL3_B
switchport mode trunk
!
interface FastEthernet1/2
switchport access vlan 10
!
interface FastEthernet1/3
switchport access vlan 60
!
interface FastEthernet1/4
switchport access vlan 60
!
interface FastEthernet1/5
switchport access vlan 60
!
interface FastEthernet1/6
switchport access vlan 60
!
interface FastEthernet1/7
switchport access vlan 60
!
interface FastEthernet1/8
switchport access vlan 40
!
interface FastEthernet1/10
switchport access vlan 40
!
interface FastEthernet1/11
switchport access vlan 40
!
interface FastEthernet1/12
switchport access vlan 40
!
interface FastEthernet1/13
switchport access vlan 40
!
interface FastEthernet1/14
switchport access vlan 40
!
interface FastEthernet1/15
switchport access vlan 40
!
interface Vlan1
no ip address
!
interface Vlan30
ip address 173.16.0.103 255.255.255.224
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
```

Εικόνα 6-109 Access ports

1^{ος} Όροφος(Secretariat & VLAN_Admin & WAP)



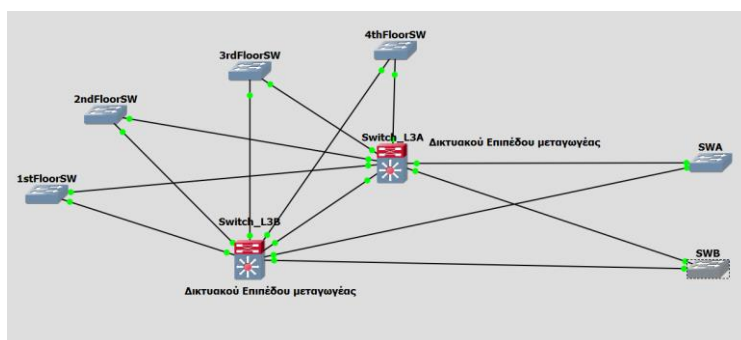
```
1stFloorSW#conf t
Enter configuration commands, one per line. End with CNTL/Z.
1stFloorSW(config)#interface range fal/2
1stFloorSW(config-if-range)#switchport mode access
1stFloorSW(config-if-range)#switchport access vlan 10
1stFloorSW(config-if-range)#no shut
1stFloorSW(config-if-range)#exit
1stFloorSW(config)#exit
1stFloorSW#
*Mar 1 00:03:49.031: %SYS-5-CONFIG_I: Configured from console by console
1stFloorSW#conf t
Enter configuration commands, one per line. End with CNTL/Z.
1stFloorSW(config)#interface range fal/3 - 5
1stFloorSW(config-if-range)#switchport mode access
1stFloorSW(config-if-range)#switchport access vlan 50
1stFloorSW(config-if-range)#no shut
1stFloorSW(config-if-range)#exit
1stFloorSW(config)#exit
1stFloorSW#
*Mar 1 00:04:06.463: %SYS-5-CONFIG_I: Configured from console by console
1stFloorSW#conf t
Enter configuration commands, one per line. End with CNTL/Z.
1stFloorSW(config)#interface range fal/6 - 15
1stFloorSW(config-if-range)#switchport mode access
1stFloorSW(config-if-range)#switchport access vlan 30
1stFloorSW(config-if-range)#no shut
1stFloorSW(config-if-range)#exit
1stFloorSW(config)#
1stFloorSW(config)#exit
1stFloorSW#wr
Building configuration...
*Mar 1 00:04:44.603: %SYS-5-CONFIG_I: Configured from console by console[OK]
```

Εικόνα 6-110 Ranges 1^{ος} ορόφου ανάλογα με τα τμήματα

```
interface FastEthernet1/0
description Sundesh me switchL3_A
switchport mode trunk
!
interface FastEthernet1/1
description Sundesh me switchL3_B
switchport mode trunk
!
interface FastEthernet1/2
switchport access vlan 10
!
interface FastEthernet1/3
switchport access vlan 50
!
interface FastEthernet1/4
switchport access vlan 50
!
interface FastEthernet1/5
switchport access vlan 50
!
interface FastEthernet1/6
switchport access vlan 30
!
interface FastEthernet1/7
switchport access vlan 30
!
interface FastEthernet1/8
switchport access vlan 30
!
interface FastEthernet1/9
switchport access vlan 30
!
interface FastEthernet1/10
switchport access vlan 30
!
interface FastEthernet1/11
switchport access vlan 30
!
interface FastEthernet1/12
switchport access vlan 30
!
interface FastEthernet1/13
switchport access vlan 30
!
interface FastEthernet1/14
switchport access vlan 30
!
interface FastEthernet1/15
switchport access vlan 30
!
interface Vlan1
no ip address
!
interface Vlan30
ip address 173.16.0.102 255.255.255.224
```

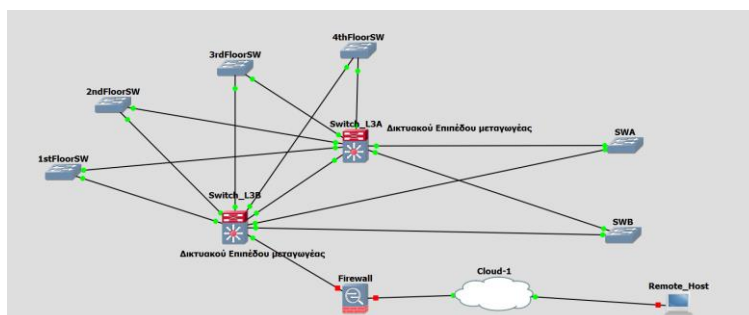
Εικόνα 6-111 Access ports

Μετά από αυτές τις παραμετροποιήσεις που έχουμε κάνει η τοπολογία του οργανισμού μας είναι σε αυτή τη μορφή



Εικόνα 6-112 Τοπολογία του οργανισμού

Το επόμενο βήμα στο σενάριο υλοποίησης που μελετάμε θα ήταν ένα Firewall που θα μας προστάτευε από εξωτερικές επιθέσεις, αυτό βέβαια προϋποθέτει αρκετή μελέτη και καλές γνώσεις στον τομέα της ασφάλειας των δικτύων, το οποίο το αφήνουμε για μελλοντική μελέτη σε ενδεχόμενη μεταπτυχιακή διπλωματική εργασία. Θεωρητικά προσεγγίζουμε το σενάριο της ασφάλειας με τον εξής τρόπο:



Εικόνα 6-113 Θεωρητική προσέγγιση Firewall

Σίγουρα θα υπάρχει κάποιο σύστημα που θα ζητά από τον εξωτερικό χρήστη username και password όπου αυτό θα πρέπει να έχει υλοποιηθεί στον δρομολογητή Firewall. Η σύνδεση του Firewall με τους μεταγωγείς επιπέδου 3 θα πρέπει να γίνεται με ssh χρησιμοποιώντας κάποιον αλγόριθμο κρυπτογράφησης για την μεταφορά των δεδομένων. Επίσης θα έπρεπε στο Firewall δρομολογητή να έχει διαφημιστεί με OSPF το δίκτυο που θέλει να έχει πρόσβαση στο δίκτυο μας. Θα έπρεπε να ορίσουμε κάποιο list που θα επέτρεπε μόνο hosts που έχουν σχέση με τον οργανισμό μας. Ο Remote host ας πούμε που βλέπουμε στην εικόνα είναι κάποιος ελεγκτής για συντήρηση κάποιου τμήματος.

6.5.1 Επιβεβαίωση λειτουργίας

Σε αυτό το σημείο έχοντας τελειώσει με τα switch που δίνουν πρόσβαση αυτό που θα πρέπει να κάνουμε είναι να πάμε στους μεταγωγείς που αποτελούν τον πυρήνα του

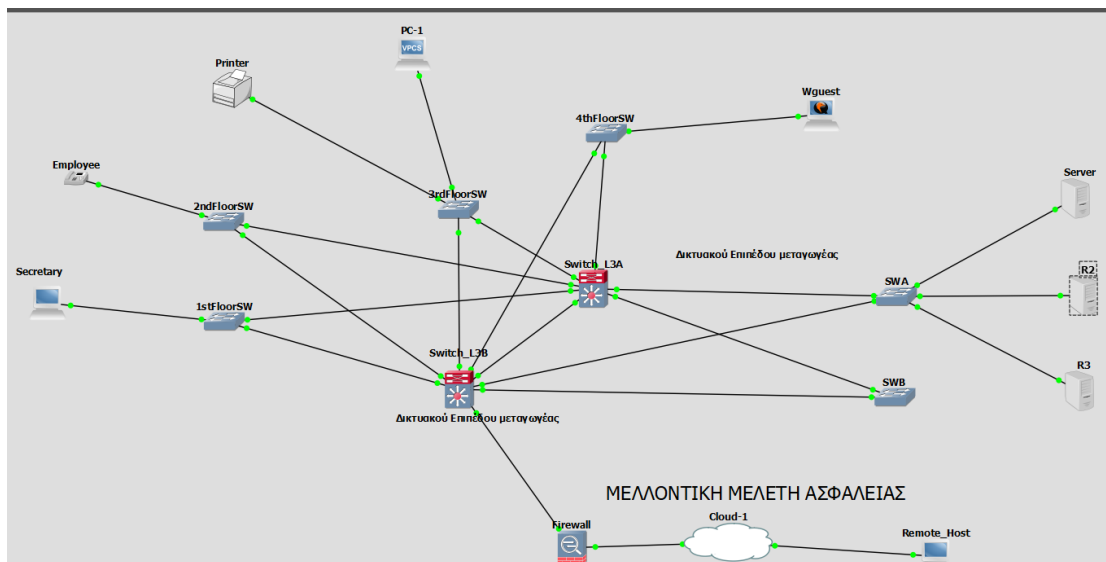


δικτύου μας(Switch_L3A, Switch L3B) και τρέχουμε το πρωτόκολλο OSPF όπως φαίνεται στο παρακάτω στιγμιότυπο οθόνης.

```
Switch_L3A#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch_L3A(config)#router ospf 10
Switch_L3A(config-router)#network 173.16.0.0 255.255.255.192 area 0
Switch_L3A(config-router)#network 173.16.0.64 255.255.255.224 area 0
Switch_L3A(config-router)#network 173.16.0.95 255.255.255.224 area 0
Switch_L3A(config-router)#network 173.16.0.127 255.255.255.240 area 0
Switch_L3A(config-router)#network 173.16.0.143 255.255.255.240 area 0
Switch_L3A(config-router)#network 173.16.0.159 255.255.255.240 area 0
Switch_L3A(config-router)#network 173.16.0.167 255.255.255.240 area 0
Switch_L3A(config-router)#network 173.16.0.183 255.255.255.248 area 0
Switch_L3A(config-router)#network 173.16.0.199 255.255.255.248 area 0
```

Εικόνα 6-114 Πρωτόκολλο διαφήμισης OSPF

Το ίδιο θα πρέπει να τρέξουμε και στον δρομολογητή που έχουμε για back up δηλαδή τον Switch_L3B σε περίπτωση βλάβης να αναλάβει αυτός λειτουργία. Σε αυτό το σημείο πρέπει να τονίσουμε ότι οι Hosts στο GNS3 πρέπει να προσομοιωθούν με κανονικά VMs από το Vbox με λειτουργικό Windows η Ubuntu έτσι ανοίγοντας τα παίρνουν με DHCP από τα POOL που έχουμε ορίσει στον κορμό του δικτύου μας IP διεύθυνση. Εμείς στο πείραμα μας θέλουμε σε κάθε όροφο ένα χρήστη και από ένα χρήστη στους 3 servers. Οπότε 8 VMs σε μία τοπολογία. Όπως θα παρουσιαστεί και σε παρακάτω ενότητα που θα μελετηθεί η συνπροσομοίωση με το Mininet θα δούμε ότι στο περιβάλλον του GNS3 μετά το 2^ο VM η CPU χτυπάει 100% και δεν μας επιτρέπει να πάρουμε τα αποτελέσματα που θέλουμε. Οπότε για να δούμε αν δουλεύει το δίκτυο που έχουμε σχεδιάσει σε κάθε όροφο βάζουμε ένα δρομολογητή που θα δουλεύει σαν host για να δοκιμάσουμε αν υπάρχει επικοινωνία μεταξύ των ορόφων μας. Έχοντας δώσει IP σε κάθε host ακολουθώντας το VLSM που υλοποιήσαμε πάμε να κάνουμε rings μεταξύ των host του δικτύου του οργανισμού μας.



Εικόνα 6-115 Τοπολογία Οργανισμού



- Επιχειρούμε ping από το 1^ο όροφο έχοντας τον χρήστη Secretary ping στον 2^ο όροφο που έχει τον Employee, μετά στον 3^ο όροφο που έχει ένα εκτυπωτή και στον 4^ο όροφο που υπάρχει ένα άλλος χρήστης

```
Topologia Organismou - GNS3
Secretary
Cisco IOS Software, 3700 Software (C3745)
Technical Support: http://www.cisco.com/
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Wed 18-Aug-10 08:18 by prod_rel_team
*Mar 1 00:00:08.783: %SNMP-5-COLDSTART:
*Mar 1 00:00:09.547: %LINEPROTO-5-UPDOWN:
*Mar 1 00:00:09.547: %LINEPROTO-5-UPDOWN:
Secretary#
*Mar 1 00:00:51.443: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
Secretary#
*Mar 1 00:00:54.135: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
Secretary#sh ip int brief
Secretary#ping 173.16.0.162
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.162, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 76/93/112 ms
Secretary#

Employee
*Mar 1 00:00:08.311: %SYS-5-CONFIG_I: Configured from console by Secretary
*Mar 1 00:00:08.463: %LINK-5-CHANGED: Interface FastEthernet0/0,
*Mar 1 00:00:08.463: %LINK-5-UPDOWN: Interface FastEthernet0/0,
*Mar 1 00:00:08.831: %SYS-5-RESTART: System restarted.
Cisco IOS Software, 3700 Software (C3745)
Technical Support: http://www.cisco.com/
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Wed 18-Aug-10 08:18 by prod_rel_team
*Mar 1 00:00:08.855: %SNMP-5-COLDSTART:
*Mar 1 00:00:09.463: %LINEPROTO-5-UPDOWN:
*Mar 1 00:00:09.463: %LINEPROTO-5-UPDOWN:
*Mar 1 00:00:53.731: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
*Mar 1 00:00:56.099: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
Employee#sh ip int brief
Employee#
```

Εικόνα 6-116 Ping από το 1^ο όροφο προς το 2^ο

```
Secretary
*Mar 1 00:00:09.547: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state
Secretary#
*Mar 1 00:00:51.443: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
Secretary#
*Mar 1 00:00:54.135: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
Secretary#sh ip int brief
Secretary#ping 173.16.0.162
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.162, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 88/207/628 ms
Secretary#

Printer
*Mar 1 00:00:08.607: %SYS-5-RESTART: System restarted.
Cisco IOS Software, 3700 Software (C3745)
Technical Support: http://www.cisco.com/
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Wed 18-Aug-10 08:18 by prod_rel_team
*Mar 1 00:00:08.627: %SNMP-5-COLDSTART:
*Mar 1 00:00:09.327: %LINEPROTO-5-UPDOWN:
*Mar 1 00:00:09.331: %LINEPROTO-5-UPDOWN:
*Mar 1 00:00:53.539: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
*Mar 1 00:00:56.315: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
Printer#sh ip int brief
Printer#
```

Εικόνα 6-117 Ping από το 1^ο όροφο προς το 3^ο

```
Secretary
*Mar 1 00:00:09.547: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state
Secretary#
*Mar 1 00:00:51.443: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
Secretary#
*Mar 1 00:00:54.135: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
Secretary#sh ip int brief
Secretary#ping 173.16.0.162
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.162, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/140/360 ms
Secretary#

Wquest
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Wed 18-Aug-10 08:18 by prod_rel_team
*Mar 1 00:00:08.791: %SNMP-5-COLDSTART: SNMP agent on host
*Mar 1 00:00:09.563: %LINEPROTO-5-UPDOWN: Line protocol on Interface
*Mar 1 00:00:09.563: %LINEPROTO-5-UPDOWN: Line protocol on Interface
*Mar 1 00:00:53.519: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
*Mar 1 00:00:56.275: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.150, State
ding Done
Wquest#sh ip int brief
Wquest#
```

Εικόνα 6-118 Ping από το 1^ο όροφο προς το 4^ο



```
Secretary
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/140/360 ms
Secretary#trace 173.16.0.162

Type escape sequence to abort.
Tracing the route to 173.16.0.162
  1 173.16.0.148 68 msec
    173.16.0.147 36 msec
    173.16.0.148 52 msec
  2 173.16.0.162 76 msec 72 msec 168 msec
Secretary#trace 173.16.0.70

Type escape sequence to abort.
Tracing the route to 173.16.0.70
  1 173.16.0.148 256 msec
    173.16.0.147 216 msec
    173.16.0.148 480 msec
  2 173.16.0.70 112 msec 72 msec 116 msec
Secretary#trace 173.16.0.10

Type escape sequence to abort.
Tracing the route to 173.16.0.10
  1 173.16.0.148 204 msec
    173.16.0.147 376 msec
    173.16.0.148 160 msec
  2 173.16.0.10 128 msec 72 msec 80 msec
Secretary#
```

Εικόνα 6-119 Trace προς τις IP των ορόφων

Διαμέσων του trace βλέπουμε ότι περνάει μέσα από το VLAN 50 που έχουμε καθορίσει για τον ορόφο αυτό.

- Επιχειρούμε ping από το 2^ο όροφο έχοντας τον χρήστη Employee ping στον 1^ο όροφο που έχει τον Secretary ,μετά στον 3^ο όροφο που έχει ένα εκτυπωτή(Printer) και στον 4^ο όροφο που υπάρχει ένα άλλος χρήστης(Vmware guest)

```
Employee
Cisco IOS Software, 3700 Software (C3745-...)
Technical Support: http://www.cisco.com/
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Wed 18-Aug-10 08:18 by prod_rel
*Mar 1 00:00:08.855: %SNMP-5-COLDSTART: Secretary#trace 173.16.0.10
*Mar 1 00:00:09.463: %LINEPROTO-5-UPDOWN: Type escape sequence to abort.
*Mar 1 00:00:09.463: %LINEPROTO-5-UPDOWN: Type escape sequence to abort.
*Mar 1 00:00:53.731: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.203 on FastEthernet0/0 from LOADING Done
*Mar 1 00:00:56.099: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.204 on FastEthernet0/0 from LOADING Done
Employee#sh ip int brief
Interface IP-Address
FastEthernet0/0 173.16.0.162
FastEthernet0/1 unassigned
Employee#
Employee#ping 173.16.0.150
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.150, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/224/420 ms
Employee#

Secretary
1 173.16.0.148 256 msec
  173.16.0.147 216 msec
  173.16.0.148 480 msec
2 173.16.0.70 112 msec 72 msec 116 msec
Secretary#sh ip int brief
Interface IP-Address
FastEthernet0/0 173.16.0.150
FastEthernet0/1 unassigned
Secretary#
```

Εικόνα 6-120 Ping από 2^ο όροφο προς 1^ο

```
Employee
*Mar 1 00:00:09.463: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
*Mar 1 00:00:53.731: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.204 on FastEthernet0/0 from LOADING Done
*Mar 1 00:00:56.099: %OSPF-5-ADJCHG: Process 1, Nbr 173.16.0.203 on FastEthernet0/0 from LOADING Done
Employee#sh ip int brief
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 173.16.0.162 YES NVRAM up up
FastEthernet0/1 unassigned YES NVRAM administratively down down
Employee#
Employee#ping 173.16.0.150
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.150, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/196/444 ms
Employee#

Printer
Printer#
Printer#sh ip int brief
Interface IP-Address
FastEthernet0/0 173.16.0.70
FastEthernet0/1 unassigned
Printer#
```

Εικόνα 6-121 Ping από 2^ο όροφο προς 3^ο



```

Employee
Interface IP-Address
FastEthernet0/0 173.16.0.162
FastEthernet0/1 unassigned
Employee#
Employee#ping 173.16.0.150
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.150:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/196/644 ms
Employee#ping 173.16.0.70
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.70:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/196/644 ms
Employee#ping 173.16.0.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 88/108/164 ms
  
```

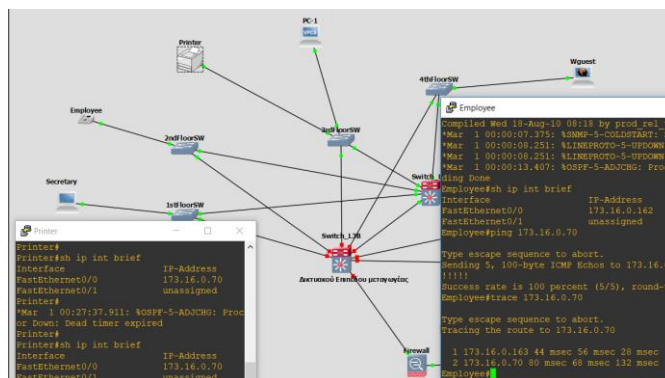
Εικόνα 6-122 Ping από 2^ο όροφο προς 4^ο

```

Employee
Employee#trace 173.16.0.150
Type escape sequence to abort.
Tracing the route to 173.16.0.150
 0 173.16.0.164 324 msec
 1 173.16.0.163 316 msec
 2 173.16.0.164 116 msec
 3 173.16.0.150 72 msec 68 msec 88 msec
Employee#trace 173.16.0.70
Type escape sequence to abort.
Tracing the route to 173.16.0.70
 0 173.16.0.164 124 msec
 1 173.16.0.163 44 msec
 2 173.16.0.164 72 msec
 3 173.16.0.70 92 msec 92 msec 92 msec
Employee#trace 173.16.0.10
Type escape sequence to abort.
Tracing the route to 173.16.0.10
 0 173.16.0.163 472 msec
 1 * επιλεγμένη ποσότητα trace του πούτzu destination
 2 173.16.0.164 236 msec
 3 173.16.0.10 72 msec 84 msec 72 msec
Employee#trace 173.16.0.10
Type escape sequence to abort.
Tracing the route to 173.16.0.10
 0 173.16.0.164 72 msec
 1 173.16.0.163 48 msec
 2 173.16.0.164 68 msec
 3 173.16.0.10 72 msec 72 msec 68 msec
Employee#
  
```

Εικόνα 6-123 Trace προς τις IP των ορόφων

Διαμέσων του trace βλέπουμε ότι περνάει μέσα από το VLAN 60 που έχουμε καθορίσει για τον όροφο αυτό πάει και στα δύο switches και όποιο έχει λιγότερη κίνηση επιλέγει αυτό. Επιλέγεται το δεύτερο. Παρακάτω βλέπουμε ότι αν κλείσουμε τον Switch_L3B πάει κατευθείαν διαμέσων του Switch_L3A,



Εικόνα 6-124 Trace με κλειστό τον Switch_L3B



- Επιχειρούμε ping από το 3^ο όροφο έχοντας τον χρήστη Printer ping στον 1^ο όροφο που έχει τον Secretary ,μετά στον 2^ο όροφο που έχει τον Employee και στον 4^ο όροφο που υπάρχει ένα άλλος χρήστης(Vmware guest).

```
Printer
ding Done
Printer#
Printer#sh ip int brief
^
% Invalid input detected at '^' marker
Printer#sh ip int brief
Interface          IP-Address
FastEthernet0/0    173.16.0.70
FastEthernet0/1    unassigned
Printer#ping 173.16.0.150
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.150:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/108/232 ms
Printer#
```

Εικόνα 6-125 Ping από 3^ο όροφο προς 1^ο

```
Printer
Printer#sh ip int brief
Interface          IP-Address
FastEthernet0/0    173.16.0.70
FastEthernet0/1    unassigned
Printer#ping 173.16.0.150
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.150:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/108/232 ms
Printer#ping 173.16.0.162
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.162, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/221/800 ms
Printer#
```

Εικόνα 6-126 Ping από το 3^ο προς 2^ο

```
Printer
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.150, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/108/232 ms
Printer#ping 173.16.0.162
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.162, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/108/232 ms
Printer#ping 173.16.0.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/74/88 ms
Printer#
```

Εικόνα 6-127 Ping από το 3^ο προς 4^ο



```

Tracing the route to 173.16.0.10
  1 173.16.0.68 516 msec
    173.16.0.67 244 msec
    173.16.0.68 172 msec
  2 173.16.0.10 164 msec 72 msec 68 msec
Printer#trace 173.16.0.162

Type escape sequence to abort.
Tracing the route to 173.16.0.162
  1 173.16.0.68 44 msec
    173.16.0.67 48 msec
    173.16.0.68 48 msec
  2 173.16.0.162 204 msec 120 msec 88 msec
Printer#trace 173.16.0.150

Type escape sequence to abort.
Tracing the route to 173.16.0.150
  1 173.16.0.68 52 msec
    173.16.0.67 48 msec
    173.16.0.68 52 msec
  2 173.16.0.150 72 msec 76 msec 96 msec
Printer#
  
```

Εικόνα 6-128 Trace προς τις IP των ορόφων

- Τέλος επιχειρούμε ping από το 4^ο όροφο έχοντας τον χρήστη Wguest ping στον 1^ο όροφο που έχει τον Secretary, μετά στον 2^ο όροφο που έχει τον Employee και στον 3^ο όροφο που υπάρχει ένα άλλος χρήστης Printer.

The screenshot shows a network simulation environment. On the left, a terminal window for 'Wguest' displays the results of ping tests to three destinations: 173.16.0.162 (Employee), 173.16.0.150 (Secretary), and 173.16.0.70 (Printer). All tests show a 100% success rate. On the right, three configuration windows are visible: 'Employee', 'Secretary', and 'Printer', each showing the 'ip int brief' command output. The Employee and Secretary configurations show FastEthernet0/0 with IP 173.16.0.162 and FastEthernet0/1 unassigned. The Printer configuration shows FastEthernet0/0 with IP 173.16.0.70 and FastEthernet0/1 unassigned. At the bottom, a network diagram shows a central switch labeled 'Δικτυακού Επίπεδου μεταγωγός' connected to three nodes: Employee, Secretary, and Printer.

Εικόνα 6-129 Pings προς όλους τους ορόφους

The screenshot shows a network simulation environment. On the left, a terminal window for 'Wguest' displays the results of trace tests to three destinations: 173.16.0.162 (Employee), 173.16.0.150 (Secretary), and 173.16.0.70 (Printer). The trace results show the path taken by the packets, including hop counts and delays. On the right, three configuration windows are visible: 'Employee', 'Secretary', and 'Printer', each showing the 'ip int brief' command output. The Employee and Secretary configurations show FastEthernet0/0 with IP 173.16.0.162 and FastEthernet0/1 unassigned. The Printer configuration shows FastEthernet0/0 with IP 173.16.0.70 and FastEthernet0/1 unassigned.

Εικόνα 6-130 Trace προς τις IP των ορόφων



- Επιχειρούμε ping από την πλευρά του ενός server προς τα δύο κεντρικά switches και βλέπουμε ότι επικοινωνεί επιτυχώς στα VLAN 90

```
Switch_L3A
FastEthernet1/15      unassigned      YES unset  up        down
Vlan1                unassigned      YES NVRAM  up        up
Vlan10              173.16.0.2      YES NVRAM  up        up
Vlan20              173.16.0.67     YES NVRAM  up        up
Vlan30              173.16.0.97     YES NVRAM  up        up
Vlan40              173.16.0.130    YES NVRAM  up        up
Vlan50              173.16.0.147    YES NVRAM  up        up
Vlan60              173.16.0.163    YES NVRAM  up        up
Vlan70              173.16.0.177    YES NVRAM  up        up
Vlan80              173.16.0.194    YES NVRAM  up        up
Vlan90              173.16.0.203    YES NVRAM  up        up
Switch_L3A#

Switch_L3B
Vlan20              173.16.0.68     YES NVRAM  up        up
Vlan30              173.16.0.99     YES NVRAM  up        up
Vlan40              173.16.0.131    YES NVRAM  up        up
Vlan50              173.16.0.148    YES NVRAM  up        up
Vlan60              173.16.0.164    YES NVRAM  up        up
Vlan70              173.16.0.179    YES NVRAM  up        up
Vlan80              173.16.0.195    YES NVRAM  up        up
Vlan90              173.16.0.204    YES NVRAM  up        up

Server
FastEthernet2/14      unassigned      YES unset  up        down
FastEthernet2/15      unassigned      YES unset  up        down
Vlan1                unassigned      YES NVRAM  up        down
Server# ping 173.16.0.203
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.203, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/148/424 ms
Server#
```

Εικόνα 6-131 Ping από το Server προς τον Switch_L3A

```
Switch_L3A
FastEthernet1/15      unassigned      YES unset  up        down
Vlan1                unassigned      YES NVRAM  up        up
Vlan10              173.16.0.2      YES NVRAM  up        up
Vlan20              173.16.0.67     YES NVRAM  up        up
Vlan30              173.16.0.97     YES NVRAM  up        up
Vlan40              173.16.0.130    YES NVRAM  up        up
Vlan50              173.16.0.147    YES NVRAM  up        up
Vlan60              173.16.0.163    YES NVRAM  up        up
Vlan70              173.16.0.177    YES NVRAM  up        up
Vlan80              173.16.0.194    YES NVRAM  up        up
Vlan90              173.16.0.203    YES NVRAM  up        up
Switch_L3A#

Switch_L3B
Vlan20              173.16.0.68     YES NVRAM  up        up
Vlan30              173.16.0.99     YES NVRAM  up        up
Vlan40              173.16.0.131    YES NVRAM  up        up
Vlan50              173.16.0.148    YES NVRAM  up        up
Vlan60              173.16.0.164    YES NVRAM  up        up
Vlan70              173.16.0.179    YES NVRAM  up        up
Vlan80              173.16.0.195    YES NVRAM  up        up
Vlan90              173.16.0.204    YES NVRAM  up        up

Server
Sending 5, 100-byte ICMP Echos to 173.16.0.203, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/148/424 ms
Server# ping 173.16.0.204
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 173.16.0.204, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/56/68 ms
Server#
```

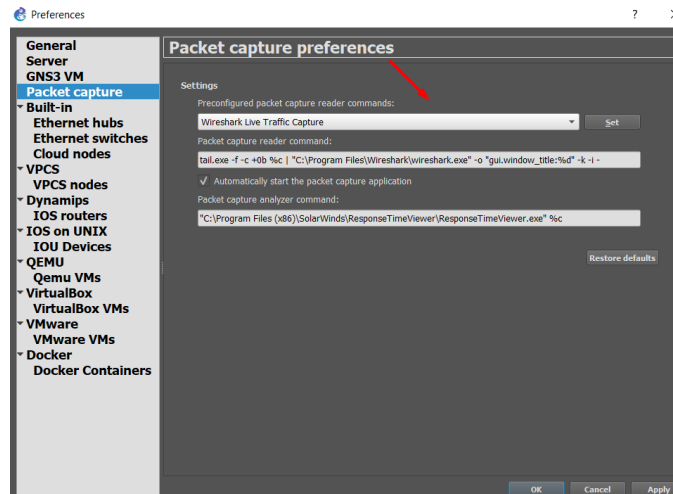
Εικόνα 6-132 Ping από το Server προς τον Switch_L3B



Μελέτη πακέτων διαμέσων του Ελεγκτή κυκλοφορίας Wireshark

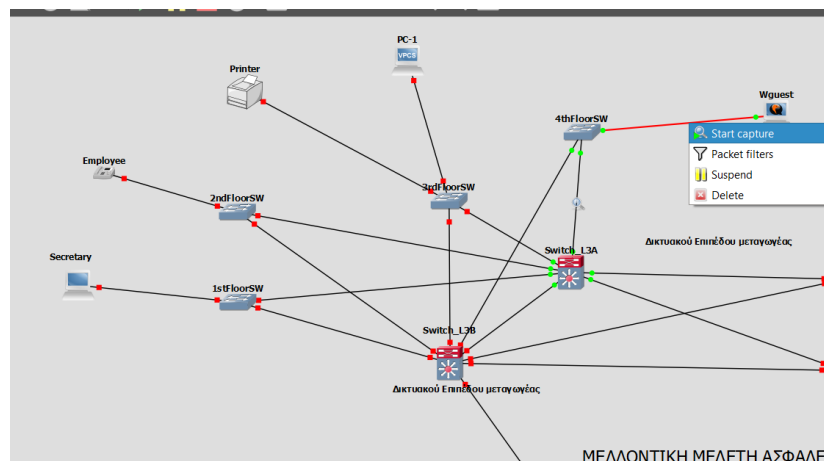
Επίσης έχοντας κατεβάσει την τελευταία έκδοση του Wireshark πηγαίνουμε να το φορτώσουμε στο γραφικό περιβάλλον του GNS3 αφού πρώτα έχουμε ανοίξει το GNS3 με τις εξής λειτουργίες:

Edit->Preferences->Packet Capture->Settings->Set και κάνουμε browse στην πρώτη στήλη που βλέπουμε και θα δούμε ότι έχει μόνο το Wireshark διαθέσιμο όποτε το επιλέγουμε και κάνουμε apply.



Εικόνα 6-133 Ενσωμάτωση του Wireshark στο GNS3

Το επόμενο βήμα στο οποίο πρέπει να προβούμε έτσι ώστε να κάνουμε sniff τα πακέτα είναι να φορτώσουμε ένα project και αφότου ανοίξει πάμε πάνω σε ένα link π.χ. όπως φαίνεται στο παρακάτω στιγμιότυπο οθόνης στην τοπολογία του οργανισμού μας

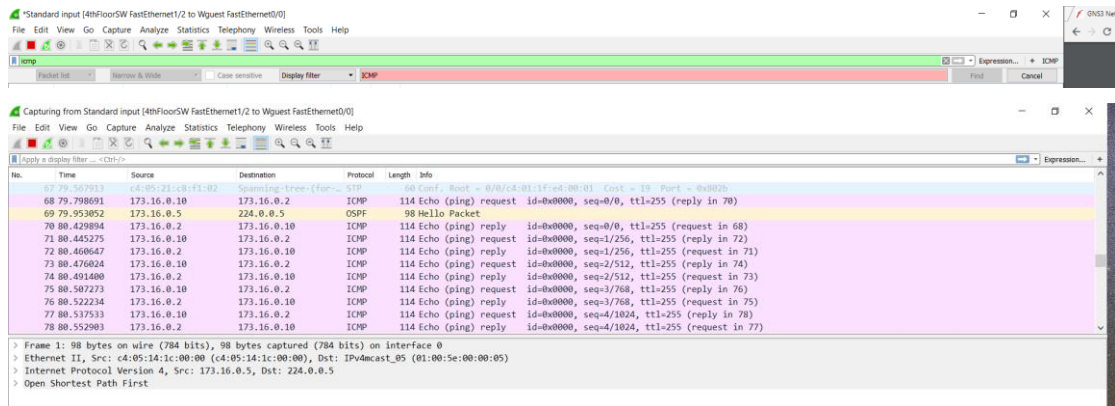


Εικόνα 6-134 Capturing μέσω Wireshark

Θέλουμε να στείλουμε 5 πακέτα από το Wguest προς τον Switch_L3A δηλαδή να κάνουμε ping στη διεύθυνση του VLAN 10 δηλαδή την 173.16.0.2. Πατάμε δεξί κλικ πάνω στο link που έχει κοκκινίσει->Start capture και μετά OK στην καρτέλα που μας εμφανίζεται και το Wireshark

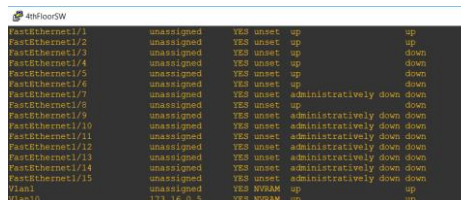


ανοίγει. Πηγαίνουμε στο Wireshark βάζουμε στο φίλτρο ICMP και πατάμε Enter και μας εμφανίζεται η εικόνα 6-134.



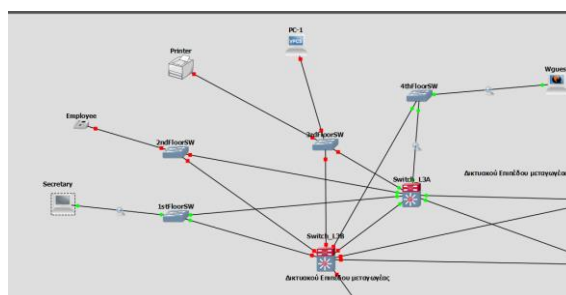
Εικόνα 6-135 ICMP πακέτα-Request από την πλευρά του wguest και reply από την πλευρά του Switch_L3A

Στην εικόνα 6-135 βλέπουμε ότι έχοντας βάλει σαν φίλτρο το ICMP πρωτόκολλο για την μεταφορά μηνυμάτων όπως έχουμε αναφέρει και στο θεωρητικό υπόβαθρο τα πακέτα που μας εμφανίζονται είναι δέκα. Βλέπουμε ότι στο Source είναι η διεύθυνση που θέλουμε να επικοινωνήσουμε και στο Destination από εκεί που το στέλνουμε. Όπως γνωρίζουμε και βλέπουμε από το παραπάνω στιγμιότυπο οθόνης το πρώτο πακέτο που στέλνεται είναι ένα ping request προς το Switch_L3A περνάει διαμέσων του 4thFloorSwitch στο οποίο υπάρχει το OSPF λόγω του trunk και που έχουμε κάνει και περνάει από το VLAN 10 με IP 173.16.0.5 παίρνοντας ένα μήνυμα Hello και στη συνέχεια παίρνει ένα reply από τον Switch_L3A με IP 173.16.0.2 και στην υπόλοιπα στέλνει τα υπόλοιπα 4 request και παίρνει τα αντίστοιχα reply από τον προορισμό. Με την χρήση του Wireshark μπορούμε να δούμε αναλυτικά την πορεία των πακέτων που στέλνουμε και των πρωτοκόλλων που υπάρχουν μέσα στην εκάστοτε τοπολογία που μελετάται.



Εικόνα 6-136 Πορεία του πακέτου διαμέσων του 4thFloorSW

Στέλνοντας τώρα ένα πακέτο από το Secretary με IP 173.16.0.150 προς το Wguest με IP 173.16.0.10 παίρνουμε το παρακάτω στιγμιότυπο ελέγχου από το Wireshark



Εικόνα 6-137 Διαδρομή πακέτου



*Standard input [1stFloorSW FastEthernet1/4 to Secretary FastEthernet0/0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|--------------|--------------|----------|--------|---|
| 16 | 15.043658 | 173.16.0.150 | 224.0.0.5 | OSPF | 94 | Hello Packet |
| 17 | 16.612489 | 173.16.0.150 | 173.16.0.10 | ICMP | 114 | Echo (ping) request id=0x0001, seq=0/0, ttl=255 (reply in 20) |
| 18 | 16.996993 | 173.16.0.147 | 173.16.0.150 | OSPF | 62 | Conf: Root = 8192/0/c4:01:1f:e4:00:05 Cost = 19 Port = 0x802d |
| 19 | 17.519671 | 173.16.0.147 | 224.0.0.5 | OSPF | 94 | Hello Packet |
| 20 | 17.672974 | 173.16.0.10 | 173.16.0.150 | ICMP | 114 | Echo (ping) reply id=0x0001, seq=0/0, ttl=254 (request in 17) |
| 21 | 17.688353 | 173.16.0.150 | 173.16.0.10 | ICMP | 114 | Echo (ping) request id=0x0001, seq=1/256, ttl=255 (reply in 23) |
| 22 | 17.688353 | 173.16.0.147 | 224.0.0.2 | HSRP | 62 | Hello (state Active) |
| 23 | 17.795981 | 173.16.0.10 | 173.16.0.150 | ICMP | 114 | Echo (ping) reply id=0x0001, seq=1/256, ttl=254 (request in 21) |
| 24 | 17.811357 | 173.16.0.150 | 173.16.0.10 | ICMP | 114 | Echo (ping) request id=0x0001, seq=2/512, ttl=255 (reply in 25) |
| 25 | 17.857486 | 173.16.0.10 | 173.16.0.150 | ICMP | 114 | Echo (ping) reply id=0x0001, seq=2/512, ttl=254 (request in 24) |
| 26 | 17.873318 | 173.16.0.150 | 173.16.0.10 | ICMP | 114 | Echo (ping) request id=0x0001, seq=3/768, ttl=255 (reply in 27) |

Εικόνα 6-138 Ping από το Secretary προς το wguest

Εδώ παρατηρούμε όπως και πριν ότι πηγαίνει ένα request προς την IP 173.16.0.10 βλέπουμε ότι περνάει διαμέσων του Switch_L3A δηλαδή διαμέσων του VLAN 50 στην 173.16.0.147 παίρνει ένα μήνυμα Hello από το OSPF που υπάρχει υλοποιημένο στους πίνακες του, παίρνει ένα reply άρα μιλάνε επιτυχώς και μετά παίρνει ακόμη ένα μήνυμα Hello από το HSRP που είναι ενεργοποιημένο(active) όπως είπαμε και πιο πάνω για περίπτωση βλάβης και μετά συνεχίζει κανονικά η επικοινωνία των δύο κόμβων όπου βλέπουμε ότι στα reply το TTL (Time to Live) μειώνεται στο 254 πράγμα που σημαίνει ότι συνέβη κάποια πολύ μικρή απώλεια κάπου μέσα στον διάυλο της επικοινωνίας.

Από την υλοποίηση του συγκεκριμένου σεναρίου προσπαθήσαμε να καταλάβουμε πως θα μπορούσε να στηθεί ένα δίκτυο στο πραγματικό κόσμο και τι προβλήματα μπορούν να δημιουργηθούν. Κατανοήσαμε ότι μεγάλο κομμάτι στο σχεδιασμό και στην υλοποίηση δικτυακών τοπολογιών παίζει η πρόβλεψη για μελλοντικές επεκτάσεις του δικτύου και φυσικά οι εναλλακτικές διαδρομές έτσι ώστε να μην επηρεάζεται η απόδοση του δικτύου. Επίσης έγινε μια πολύ μικρή αναφορά όσον αφορά την ασφάλεια ενός δικτύου και ορισμένες βασικές πρακτικές που πρέπει να έχει κάθε δίκτυο για την εύρυθμη λειτουργία του, ωστόσο δεν μελετήθηκε περαιτέρω.

6.6 Μελέτη και υλοποίηση τοπολογίας με χρήση του πρωτοκόλλου BGP σε συνεργασία με το EIGRP/OSPF με χρήση GRE Tunneling

Στην τοπολογία που μελετήσαμε θα γίνει προσομοίωση ενός σεναρίου tunneling όπου θα έχω 3 ISP routers, 2 customers routers και 2 τερματικά(hosts) που θα αναπαρίστανται από ελαφριά εικονικοποιημένα VMs,ένα στη μία πλευρά του tunnel και ένα στην άλλη.

Ο Πάροχος Υπηρεσιών Διαδικτύου (Internet Service Provider, ISP) είναι ένας οργανισμός, κερδοσκοπικός ή μη, που παρέχει στους συνδρομητές και χρήστες του, συχνά μέσω χρηματικού ή άλλου αντιτίμου, διάφορες υπηρεσίες, οι οποίες σχετίζονται με το Διαδίκτυο (ίντερνετ),όπως πρόσβαση σε ιστοσελίδες, ανταλλαγή ηλεκτρονικών μηνυμάτων (e-mail) αλλά και ανταλλαγή αρχείων (file sharing), επικοινωνία χρηστών σε πραγματικό χρόνο (chat) κ.λπ. Οι πάροχοι πρόσβασης ISP χρησιμοποιούν μια σειρά τεχνολογιών προκειμένου να συνδέσουν τους χρήστες στο δίκτυό τους. Οι διαθέσιμες τεχνολογίες με τις



οποίες οι πάροχοι ISP συνδέουν τους χρήστες στο διαδίκτυο είναι: οι τηλεφωνικές γραμμές, τα καλώδια τηλεόρασης (CATV), το ενσύρματο και ασύρματο Ethernet και οι οπτικές ίνες.[46]

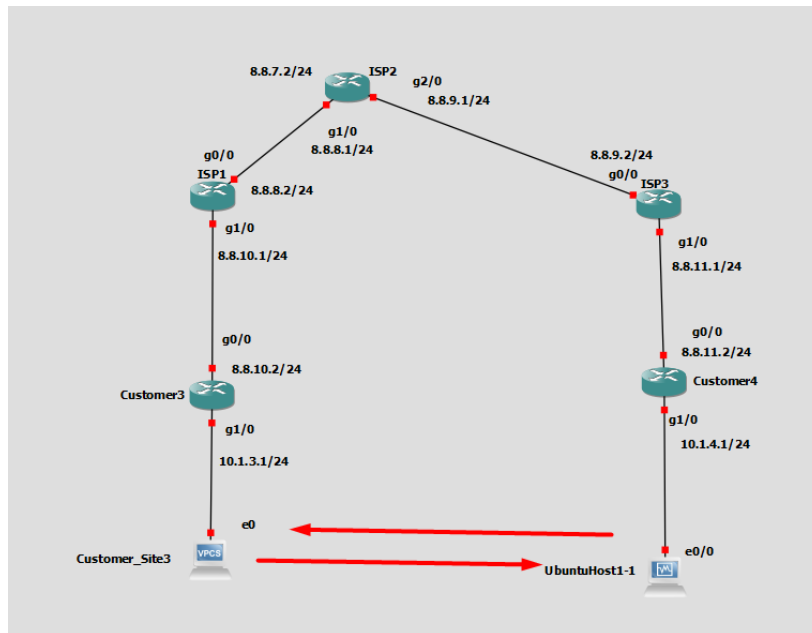
GRE Tunneling(Generic Routing Encapsulation)

Το GRE Tunneling είναι ένας μηχανισμός ενθυλάκωσης που τον δημιούργησε η Cisco και εκτελείται ανάμεσα στο δρομολογητή πηγής και το δρομολογητή προορισμού(router-to-router). Τα GRE tunnels παρέχουν ένα ειδικό μονοπάτι κατά μήκος μίας διαμοιραζόμενης υποδομής WAN που δεν ανήκει μόνο σε έναν χρήστη-πελάτη (για παράδειγμα Internet) και ενθυλακώνουν την κίνηση με νέες επικεφαλίδες πακέτου για να εξασφαλίσουν τη διανομή σε ένα συγκεκριμένο προορισμό. Μπορεί να ενσωματώσει και να μεταφέρει πακέτα από 20 διαφορετικά πρωτόκολλα. Τα πακέτα που πρόκειται να προωθηθούν κατά μήκος της διόδου ενθυλακώνονται με μία επικεφαλίδα GRE, μεταφέρονται κατά μήκος της διόδου και στο τέλος της αφαιρείται η επικεφαλίδα GRE.

Βασικά συστατικά του μηχανισμού GRE είναι:

- Passenger protocol: Το πρωτόκολλο που χρησιμοποιείται στο τοπικό δίκτυο και στη συνέχεια ενθυλακώνεται.
- Carrier protocol: Το πρωτόκολλο GRE που παρέχει την υπηρεσία.
- Transport protocol: Το πρωτόκολλο που χρησιμοποιείται για να μεταφερθούν τα άλλα δύο (μετά την ενθυλάκωση).[47]

Η τοπολογία που θα μελετήσουμε έχει την παρακάτω μορφή και σκοπός είναι τα interfaces G1/0 από την μέσα πλευρά των Customers να μην γνωρίζει ο έξω κόσμος την IP τους, στην προκειμένη οι πάροχοι ISP. Έτσι αυτά θα μπορούν να επικοινωνήσουν μεταξύ τους διαμέσω ενός tunnel interface που θα κατασκευάσουμε αλλά θα το βλέπουμε μόνο εμείς, έτσι με πιο απλά λόγια η εσωτερική IP θα ενθυλακώνεται πάνω στην εξωτερική δημόσια IP.



Εικόνα 6-139 Τοπολογία του GRE Tunneling

Το δίκτυο θα τρέχει με τον εξής τρόπο οι 3 ISP θα επικοινωνούν μεταξύ τους με το πρωτόκολλο γειτνίασης BGP ,επίσης θα χρησιμοποιήσουμε σε κάθε ISP και το loopback interface του γιατί δεν “πέφτει” ποτέ ουσιαστικά και θα το διαφημίσουμε στους άλλους ISP διαμέσων του OSPF πρωτοκόλλου που είναι IGP. Έτσι θα επιτύχω το tunneling που θέλω ενθυλακώνοντας την πλευρά του πελάτη. Οι λεπτομέρειες και ο κώδικας παρατίθενται στα παρακάτω στιγμιότυπα οθόνης.

Επιλέγουμε πρώτα να προγραμματίσουμε τον ISP1 όπου θα πρέπει να τρέξουμε τις παρακάτω εντολές.

```
ISP1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP1(config)#int lo0
ISP1(config-if)#ip add 1.1.1.1 255.255.255.255
ISP1(config-if)#no shut
ISP1(config-if)#exit
ISP1(config)#int g0/0
ISP1(config-if)#ip add 8.8.8.2 255.255.255.0
ISP1(config-if)#no shut
ISP1(config-if)#
```

Εικόνα 6-140 Προγραμματισμός interfaces lo0 και g0/0

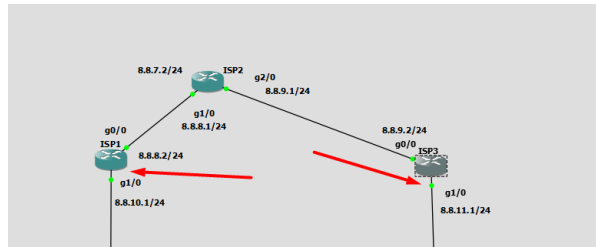
Πηγαίνουμε και δίνουμε IP στο loopback interface και στο g0/0 interface όπως φαίνεται στην εικόνα 6-140. Ακριβώς την ίδια διαδικασία πρέπει να κάνω και στους υπόλοιπους 2 ISP με lo0 στον ISP2 2.2.2.2 και στον ISP3 3.3.3.3. Πηγαίνουμε στον ISP3 και δίνουμε IP στο g0/0 8.8.9.2/24 όπως βλέπουμε και πιο πάνω.

Στη συνέχεια πρέπει να πάμε στον ISP2 που αυτός θα πραγματοποιήσει τη γειτνίαση προς τους άλλους δύο εσωτερικούς γείτονες και βάζω σαν loopback interface το 2.2.2.2 και στην μία πόρτα που είναι συνδεδεμένο με τον ISP1 (g1/0) βάζω την πρώτη IP του δικτύου 8.8.8.1. Ακολουθούμε



την ίδια ακριβώς λογική και από την πλευρά που είναι ο ISP3 στην gigabitethernet2/0 βάζω την πρώτη IP του δικτύου 8.8.9.1 και στην άλλη πλευρά έχουμε δώσει την .2.

Επίσης θα πρέπει να πάω να βάλω και στις παρακάτω πόρτες IP (g1/0) και έπειτα να τις διαφημίσω. Βάζουμε στον ISP1 IP ενός άλλου δικτύου του 8.8.10.0 που θα ακολουθεί στην πλευρά των πελατών και στον ISP3 του 8.8.11.0



Εικόνα 6-141 Απόδοση IP διεύθυνσης στις g1/0 θύρες των ISP

Έχοντας τελειώσει τις εξής ενέργειες είμαστε έτοιμοι να πάμε στο επόμενο βήμα ελέγχοντας ότι οι IP έχουν εκχωρηθεί σωστά. Οι διευθύνσεις όπου έχουν τονιστεί με κόκκινο βέλος δεν εξυπηρετούν κάτι σε αυτό το project είναι από μελέτη άλλης περίπτωσης και δεν τις λαμβάνουμε υπόψη.

```
ISP1
Interface IP-Address OMT Method Status Protocol
Ethernet0/0 unassigned YES NVRAM administratively down down
GigabitEthernet0/0 8.8.8.2 YES NVRAM up up
GigabitEthernet1/0 8.8.10.1 YES NVRAM up up
GigabitEthernet2/0 unassigned YES NVRAM administratively down down
GigabitEthernet3/0 unassigned YES NVRAM administratively down down
GigabitEthernet4/0 unassigned YES NVRAM administratively down down
Loopback0 1.1.1.1 YES NVRAM up up

ISP2
Interface IP-Address OMT Method Status Protocol
Ethernet0/0 unassigned YES NVRAM administratively down down
GigabitEthernet0/0 8.8.7.2 YES NVRAM up up
GigabitEthernet1/0 8.8.8.1 YES NVRAM up up
GigabitEthernet2/0 8.8.9.1 YES NVRAM up up
GigabitEthernet3/0 8.8.4.1 YES NVRAM up up
GigabitEthernet4/0 unassigned YES NVRAM administratively down down
Loopback0 2.2.2.2 YES NVRAM up up

ISP3
Interface IP-Address OMT Method Status Protocol
Ethernet0/0 unassigned YES NVRAM administratively down down
GigabitEthernet0/0 8.8.9.2 YES NVRAM up up
GigabitEthernet1/0 8.8.11.1 YES NVRAM up up
GigabitEthernet2/0 unassigned YES NVRAM administratively down down
GigabitEthernet3/0 unassigned YES NVRAM administratively down down
GigabitEthernet4/0 3.3.3.3 YES NVRAM up up
Loopback0 3.3.3.3 YES NVRAM up up
```

Εικόνα 6-142 Έλεγχος έγκυρης απόδοσης IP διεύθυνσης

Θα ξεκινήσουμε πηγαίνοντας στον ISP2 και θα τρέξουμε το BGP πρωτόκολλο γράφοντας τον παρακάτω κώδικα όπου καθορίζω ποιοι είναι οι γείτονες μου σύμφωνα με την τοπολογία χρησιμοποιώντας τα loopback interfaces τους (ISP1->1.1.1.1, ISP3->3.3.3.3 καθώς και τα δίκτυα που πρέπει να κάνουμε configure μαζί με το loopback του ISP2. Έτσι το BGP γνωρίζει σαν αρχικές διευθύνσεις τα loopback interfaces που βάλαμε.

```
ISP2#conf t
ISP2#router bgp 65000
ISP2#neighbor 1.1.1.1 remote-as 65000
ISP2#neighbor 3.3.3.3 remote-as 65000
ISP2#network 8.8.8.0 mask 255.255.255.0
ISP2#network 8.8.9.0 mask 255.255.255.0
ISP2#network 2.2.2.2 mask 255.255.255.255
```



Μετά από αυτό οι εσωτερικοί γείτονες γράφοντας την εντολή `show ip bgp sum` θα δούμε ότι οι εσωτερικοί γείτονες είναι idle.

| Neighbor | V | AS | MsgRcvd | MsgSent | TblVer | InQ | OutQ | Up/Down | State/PfxRcd |
|----------|---|-------|---------|---------|--------|-----|------|----------|--------------|
| 1.1.1.1 | 4 | 65000 | 244 | 246 | 11 | 0 | 0 | 03:38:50 | 3 |
| 3.3.3.3 | 4 | 65000 | 245 | 247 | 11 | 0 | 0 | 03:38:39 | 3 |

Εικόνα 6-143 Στιγμιότυπο που βλέπουμε τους εσωτερικούς γείτονες

Στην εικόνα 6-143 δεν φαίνονται idle διότι έχουμε τρέξει ήδη το σενάριο και το state είναι 3 prefixes(από αυτό το γείτονα) κανονικά αν το τρέξουμε για πρώτη φορά θα μας εμφανίζει idle. Αυτό συμβαίνει διότι δεν έχουμε κάνει configure το BGP σε αυτούς.

Πηγαίνουμε στο **ISP1** και γράφουμε τον παρακάτω κώδικα για το BGP

```
ISP1#conf t
ISP1#router bgp 65000
ISP1#neighbor 2.2.2.2 remote-as 65000
ISP1#neighbor 2.2.2.2 update-source loopback 0
ISP1#network 8.8.10.0 mask 255.255.255.0
ISP1#network 1.1.1.1 mask 255.255.255.255
ISP1#end
```

Αυτό που πρέπει να κάνουμε είναι να τρέξουμε ένα IGP πρωτόκολλο το OSPF γιατί η γειτνίαση δεν θα επιτευχθεί ποτέ δηλαδή δεν θα γίνει ποτέ 'up'. Θα τρέξουμε το OSPF μόνο στον **ISP1** και όχι στον Customer(CS1) δεν θέλουμε να ξέρει όλα τα μονοπάτια ο CS1 ουσιαστικά θα τον πηγαίνει ο ISP πάροχος.

```
ISP1# conf t
ISP1# router ospf 1
ISP1# network 8.8.8.2 0.0.0.0 area 0
ISP1# network 1.1.1.1 0.0.0.0 area 0
```

Θα πρέπει να κάνουμε το ίδιο πράγμα και στον **ISP2** για να δουλέψει η γειτνίαση των ISP

```
ISP2#conf t
ISP2#router ospf 1
ISP2#network 8.8.8.1 0.0.0.0 area 0
ISP2#network 8.8.9.1 0.0.0.0 area 0
ISP2#network 2.2.2.2 0.0.0.0 area 0
ISP2#end
```



Όταν γράφουμε τον κώδικα θα μας εμφανιστεί μήνυμα ότι μάθαμε το Loopback interface του ISP1 διαμέσων του OSPF πρωτοκόλλου.

```
ISP2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is 8.8.8.2 to network 0.0.0.0

S* 0.0.0.0/0 [1/0] via 8.8.8.2
O 1.0.0.0/32 is subnetted, 1 subnets
O 1.1.1.1 [110/2] via 8.8.8.2, 03:57:49, GigabitEthernet1/0

ISP2#sh ip bgp sum
BGP router identifier 2.2.2.2, local AS number 65000
BGP table version is 11, main routing table version 11
9 network entries using 1296 bytes of memory
12 path entries using 960 bytes of memory
3/3 BGP path/bestpath attribute entries using 408 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2664 total bytes of memory
BGP activity 9/0 prefixes, 12/0 paths, scan interval 60 secs

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down State/PfxRcd
1.1.1.1        4      65000   267   269     11    0    0 03:59:48      3
3.3.3.3        4      65000   269   270     11    0    0 03:59:36      3
```

Εικόνα 6-144 Επιτυχής λειτουργία του BGP διαμέσων του OSPF

Πηγαίνοντας στον ISP1 βλέπουμε ότι το BGP είναι UP και ξέρει σαν γείτονα τον ISP2, διαμέσων του loopback του.

```
ISP1#
*Aug  4 18:34:29.551: %BGP-5-ADJCHANGE: neighbor 2.2.2.2 Up
ISP1#

ISP1#sh ip bgp sum
BGP router identifier 1.1.1.1, local AS number 65000
BGP table version is 8, main routing table version 8
7 network entries using 1008 bytes of memory
8 path entries using 640 bytes of memory
2/2 BGP path/bestpath attribute entries using 272 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1920 total bytes of memory
BGP activity 7/0 prefixes, 8/0 paths, scan interval 60 secs

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down State/PfxRcd
2.2.2.2        4      65000     9     6     8     0    0 00:02:36      5
ISP1#

ISP1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is 8.8.8.1 to network 0.0.0.0

S* 0.0.0.0/0 [1/0] via 8.8.8.1
C 1.0.0.0/32 is subnetted, 1 subnets
C 1.1.1.1 is directly connected, Loopback0
O 2.0.0.0/32 is subnetted, 1 subnets
O 2.2.2.2 [110/2] via 8.8.8.1, 00:03:31, GigabitEthernet0/0
O 3.0.0.0/32 is subnetted, 1 subnets
O 3.3.3.3 [110/3] via 8.8.8.1, 00:03:31, GigabitEthernet0/0
O 8.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
B 8.8.8.0/24 [200/0] via 2.2.2.2, 00:02:14
B 8.8.7.0/24 [200/0] via 2.2.2.2, 00:02:14
C 8.8.8.0/24 is directly connected, GigabitEthernet0/0
E 8.8.8.2/32 is directly connected, GigabitEthernet0/0
O 8.8.9.0/24 [110/2] via 8.8.8.1, 00:03:31, GigabitEthernet0/0
O 8.8.10.0/24 is directly connected, GigabitEthernet1/0
L 8.8.10.1/32 is directly connected, GigabitEthernet1/0
ISP1#
```

Εικόνα 6-145 Ενημερωμένοι πίνακες του ISP1



Μετά πηγαίνουμε στον **ISP3** δρομολογητή για να τρέξουμε την ακόλουθη σχεδόν πανομοιότυπη διαδικασία. Τρέχουμε πρώτα το OSPF διαφημίζοντας το loopback του ISP3 ΚΑΙ την IP που έχει στην πόρτα g0/0 και βλέπουμε ότι μαθαίνει τα loopback interfaces των ISP2 και ISP1.

```
ISP3#conf t
```

```
ISP3#router ospf 1
```

```
ISP3#network 3.3.3.3 0.0.0.0 area 0
```

```
ISP3#network 8.8.9.2 0.0.0.0 area 0
```

```
ISP3
ISP3#SH IP ROUTE
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        + - replicated route, % - next hop override

Gateway of last resort is 8.8.9.1 to network 0.0.0.0

S* 0.0.0.0/0 [1/0] via 8.8.9.1
  1.0.0.0/32 is subnetted, 1 subnets
    O   1.1.1.1 [110/3] via 8.8.9.1, 00:12:21, GigabitEthernet0/0
  2.0.0.0/32 is subnetted, 1 subnets
    O   2.2.2.2 [110/2] via 8.8.9.1, 04:16:01, GigabitEthernet0/0
```

Εικόνα 6-146 Τα δίκτυα που έμαθε από το OSPF

Στην εικόνα 6-146 βλέπουμε τα δίκτυα που έμαθε με την χρήση του OSPF πρωτοκόλλου και τα loopback interfaces των γειτονικών ISP έμαθε διαμέσων της g2/0 με IP 8.8.9.1

```
ISP3#router bgp 65000
```

```
ISP3#neighbor 2.2.2.2 remote-as 65000
```

```
ISP3#neighbor 2.2.2.2 update-source loopback 0
```

```
ISP3#network 8.8.9.0 mask 255.255.255.0
```

```
ISP3#network 8.8.11.0 mask 255.255.255.0
```

```
ISP3#network 3.3.3.3 mask 255.255.255.255
```

```
ISP3#end
```



```
ISP3#sh run | sec bgp
router bgp 65000
  bgp log-neighbor-changes
  network 3.3.3.3 mask 255.255.255.255
  network 8.8.9.0 mask 255.255.255.0
  network 8.8.10.0 mask 255.255.255.0
  network 8.8.11.0 mask 255.255.255.0
  neighbor 2.2.2.2 remote-as 65000
  neighbor 2.2.2.2 update-source Loopback0
ISP3#
```

```
ISP1#sh run | sec bgp
router bgp 65000
  bgp log-neighbor-changes
  network 1.1.1.1 mask 255.255.255.255
  network 8.8.8.0 mask 255.255.255.0
  network 8.8.10.0 mask 255.255.255.0
  neighbor 2.2.2.2 remote-as 65000
  neighbor 2.2.2.2 update-source Loopback0
ISP1#
```

Εικόνα 6-147 Επιτυχής ενεργοποίηση του BGP στον ISP3 και ISP1

Κάνοντας reload τους ISP και ανοίγοντας τους βλέπουμε ότι το BGP είναι ενεργοποιημένο και οι γειτνίαση δουλεύει ορθώς.

```
ISP3#sh run | sec bgp
router bgp 65000
  bgp log-neighbor-changes
  network 3.3.3.3 mask 255.255.255.255
  network 8.8.9.0 mask 255.255.255.0
  network 8.8.10.0 mask 255.255.255.0
  network 8.8.11.0 mask 255.255.255.0
  neighbor 2.2.2.2 remote-as 65000
  neighbor 2.2.2.2 update-source Loopback0
ISP3#

ISP1#sh run | sec bgp
router bgp 65000
  bgp log-neighbor-changes
  network 1.1.1.1 mask 255.255.255.255
  network 8.8.8.0 mask 255.255.255.0
  network 8.8.10.0 mask 255.255.255.0
  neighbor 2.2.2.2 remote-as 65000
  neighbor 2.2.2.2 update-source Loopback0
ISP1#
```

Εικόνα 6-148 BGP πρωτόκολλο

```
ISP1#ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/24 ms
ISP1#ping 3.3.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/43/48 ms
ISP1#
```

Εικόνα 6-149 BGP log στον ISP1 και rings στα loopbacks των άλλων ISP

```
ISP2#show ip bgp sum
BGP router identifier 2.2.2.2, local AS number 65000
BGP table version is 10, main routing table version 10
9 network entries using 1296 bytes of memory
11 path entries using 880 bytes of memory
2/2 BGP path/bestpath attribute entries using 272 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2448 total bytes of memory
BGP activity 9/0 prefixes, 11/0 paths, scan interval 60 secs

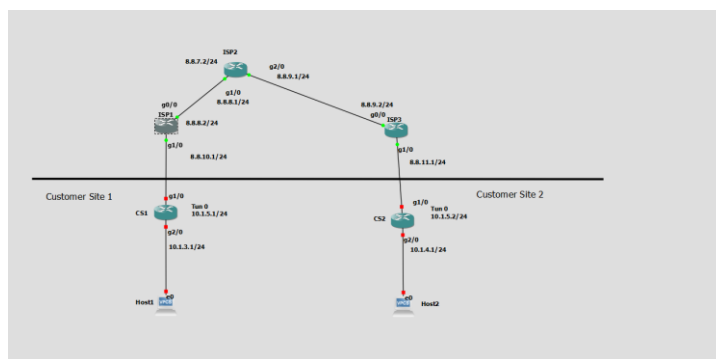
Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
1.1.1.1        4        65000    16     16      10    0  0 00:11:17      3
3.3.3.3        4        65000    16     16      10    0  0 00:11:18      3
```



```
ISP2#ping 3.3.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/18/44 ms
ISP2#ping 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/12/16 ms
ISP2#
```

Εικόνα 6-150 BGP log στον ISP2 και pings στα loopbacks των άλλων ISP

Αυτό που θα μελετήσουμε τώρα είναι το Tunneling που θέλουμε να κάνουμε μεταξύ των δύο hosts. Θα πρέπει οι δρομολογητές CS1 και CS2 στο interface g1/0 να παίρνουν IP με πρωτόκολλο DHCP και στο interface g2/0 βάζουμε εμείς στατικά IP με τον παρακάτω κώδικα.



Εικόνα 6-151 Μελέτη της πλευράς του Customer

Για να πάρουν φυσικά DHCP θα πρέπει να έχουμε δημιουργήσει ένα dhcp pool αυτό θα γίνει στους ISP1 και στους ISP3

```
ISP1(config)#ip dhcp pool POOL1
ISP1(dhcp-config)#network 8.8.10.0 255.255.255.0
ISP1(dhcp-config)#default-router 8.8.10.1
ISP1(dhcp-config)#dns-server 8.8.8.1
ISP1(dhcp-config)#
```

Εικόνα 6-152 DHCP POOL στον ISP1

Έπειτα πηγαίνουμε στον CS1 και γράφουμε τις παρακάτω εντολές για να πάρει ip με DHCP και για να εκχωρηθεί στατικά IP στην εσωτερική πλευρά του CS1(g2/0)

```
CS1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
CS1(config)#int g1/0
CS1(config-if)#ip address dhcp
CS1(config-if)#no shut
CS1(config-if)#ei
*Aug 4 13:42:52.931: %LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to up
*Aug 4 13:42:53.931: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state to up
CS1(config-if)#exit
*Aug 4 13:43:06.015: %DHCP-6-ADDRESS_ASSIGN: Interface GigabitEthernet1/0 assigned DHCP address 8.8.10.2, mas
k 255.255.255.0, hostname CS1
CS1(config-if)#exit
CS1(config)#exit
CS1#
*Aug 4 13:43:18.387: %SYS-5-CONFIG_I: Configured from console by console
CS1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
CS1(config)#int g2/0
CS1(config-if)#no shut
CS1(config-if)#ip add 10.1.3.1 255.255.255.0
CS1(config-if)#
*Aug 4 13:43:35.911: %LINK-3-UPDOWN: Interface GigabitEthernet2/0, changed state to up
*Aug 4 13:43:36.911: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet2/0, changed state to up
CS1(config-if)#
```

Εικόνα 6-153 Απόδοση διευθύνσεων



Από την εικόνα 6-152 βλέπουμε ότι τρέχοντας τον κώδικα για να πάρει αυτόματα DHCP παίρνει αυτόματα DHCP διαμέσων του POOL1 που έχουμε δημιουργήσει στον ISP1, πηγαίνοντας στο ISP1 και γράφοντας την εντολή `sh ip dhcp binding` βλέπουμε την IP που έχει δεσμεύσει ο CS1.

```
ISP1#sh ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/      Lease expiration    Type    State    Interface
Hardware address/
User name
8.8.10.2        0063.6973.636f.2d63.  Aug 05 2018 01:43 PM  Automatic  Active   GigabitEthernet1/0
6130.312e.3037.6163.
2e30.3031.632d.4769.
312f.30
```

Εικόνα 6-154 Δεσμευμένη IP από τον CS1

Με την εντολή `show ip route` μπορούμε να δούμε το routing table και μπορούμε να κάνουμε ping σε άλλα ISP devices στο "Internet" που προσομοιώνουμε.

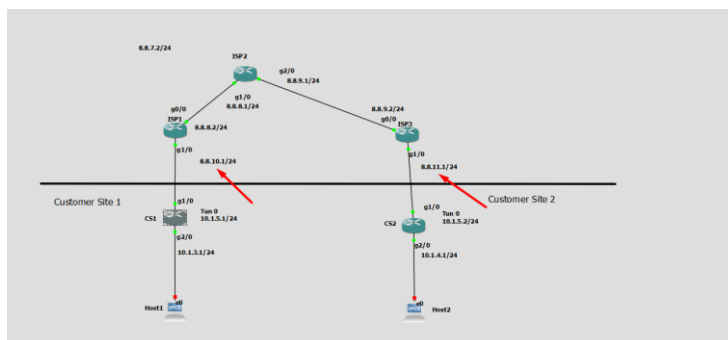
```
Gateway of last resort is 8.8.10.1 to network 0.0.0.0

S*  0.0.0.0/0 [254/0] via 8.8.10.1
    8.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
    C      8.8.10.0/24 is directly connected, GigabitEthernet1/0
    L      8.8.10.2/32 is directly connected, GigabitEthernet1/0
    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
    C      10.1.3.0/24 is directly connected, GigabitEthernet2/0
    L      10.1.3.1/32 is directly connected, GigabitEthernet2/0

CS1#ping 8.8.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.10.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/30/52 ms
CS1#ping 8.8.8.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/54/56 ms
CS1#ping 8.8.9.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.9.1, timeout is 2 seconds:
!!!!
```

Εικόνα 6-155 Επιτυχής επικοινωνία με τους ISP routers

Επίσης θα πρέπει να διαφημίσω στατικά αυτές τις διευθύνσεις προκειμένου οι δύο δρομολογητές να μπορούν να μιλήσουν, οπότε γράφοντας τον παρακάτω κώδικα στον CS1 και τον αντίστοιχο που χρειάζεται στον CS2 διαφημίζονται σωστά και ανταλλάσσουν πακέτα.



Εικόνα 6-156 IP που πρέπει να διαφημίσουμε



Για τον CS1

```
CS1#conf t
CS1#ip route 0.0.0.0 0.0.0.0 8.8.10.1
CS1#exit
CS1#wr
```

Για τον CS2

```
CS2#conf t
CS2#ip route 0.0.0.0 0.0.0.0 8.8.11.1
CS2#exit
CS2#wr
```

Έχοντας δημιουργήσει pool στον IPS3 όπως και στον ISP1 και δίνοντας στατικά IP στην εσωτερική πλευρά του CS2(g2/0) βλέπουμε ότι γράφοντας τις παρακάτω εντολές στο CS2 η απόδοση των IP γίνεται όπως και στον CS1.

```
conf t
int g1/0
ip address dhcp
no shut
int g2/0
no shut
ip add 10.1.4.1 255.255.255.0
```

```
Aug 4 14:42:10.903: %SYS-5-CONFIG_I: Configured from console by console
CS2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
CS2(config)#int g1/0
CS2(config-if)#ip address dhcp
CS2(config-if)#no shut
CS2(config-if)#int g2/0
CS2(config-if)#no shut
CS2(config-if)#ip add 10.1.4.1 255.255.255.0
CS2(config-if)#
Aug 4 14:42:15.639: %LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to up
Aug 4 14:42:15.781: %LINEPROTO-5-UPDOWN: Interface GigabitEthernet2/0, changed state to up
Aug 4 14:42:16.639: %LINEPROTO-5-UPDOWN: line protocol on Interface GigabitEthernet1/0, changed state to up
CS2(config-if)#
Aug 4 14:42:16.791: %LINEPROTO-5-UPDOWN: line protocol on Interface GigabitEthernet2/0, changed state to up
CS2(config-if)#exit
CS2(config)#exit
CS2#
Aug 4 14:42:21.147: %SYS-5-CONFIG_I: Configured from console by console
CS2#wr
Warning: Attempting to overwrite an MVRAM configuration previously written
by a different version of the system image.
Overwrite the previous MVRAM configuration?(confirm)
Building configuration...
[OK]
CS2#
Aug 4 14:42:29.851: %DHCP-6-ADDRESS_ASSIGN: Interface GigabitEthernet1/0 assigned DHCP address 8.8.11.2, mask
8 255.255.255.0, hostname CS2
CS2#
```

Εικόνα 6-157 Απόδοση διευθύνσεων στον CS2

Θα ελέγξουμε ότι οι δρομολογητές επικοινωνούν και θα τρέξουμε στον CS2 την εντολή *debug ip icmp* για να κάνει capture τα ICMP πακέτα που θα στείλουμε από τον CS1. Βλέπουμε ότι έχουμε επιτυχή επικοινωνία μεταξύ τους βλέπουμε τα echo reply που στέλνει ο CS2 με src την διεύθυνση του και dest τη διεύθυνση του CS1.



```
CS1
SI#ping 8.8.11.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.11.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/72/10
ms
SI#ping 10.1.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.4.1, timeout is 2 seconds:
....
Success rate is 0 percent (0/5)
SI#ping 8.8.11.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.11.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/56/60
ms
SI#ping 8.8.11.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.11.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/64/68
ms
SI#

CS2
Welcome to Virtual PC Simulator, version 0.6.1
Please use Ctrl-Alt-F1 to activate the console.
Type escape sequence to abort.
Verifying the previous NVRAM configuration[confirm]
Building configuration...
[OK]
CS2#
CS2#ping 8.8.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.10.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/44/48 ms
CS2#debug ip icmp
ICMP packet debugging is on
CS2#
*Aug 4 16:47:00.919: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, topology BASE, dsc
0 0 topoid 0
*Aug 4 16:47:00.983: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, topology BASE, dsc
0 0 topoid 0
*Aug 4 16:47:01.047: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, topology BASE, dsc
0 0 topoid 0
*Aug 4 16:47:01.115: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, topology BASE, dsc
0 0 topoid 0
*Aug 4 16:47:01.179: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, topology BASE, dsc
0 0 topoid 0
CS2#
```

Εικόνα 6-158 Επιτυχής επικοινωνία των δρομολογητών CS1 και CS2

Τώρα αυτό που πρέπει να κάνουμε για να επιτύχουμε το tunneling είναι δημιουργήσουμε DHCP pools στον CS1 και τον CS2 έτσι ώστε οι hosts να παίρνουν δυναμικά IP διεύθυνση. Ανοίγοντας τα host τερματικά των οποίων έχουμε φορτώσει τα IOS image με τον τρόπο που έχουμε αναφέρει στην εισαγωγή του GNS3 βλέπουμε ότι δεν έχουν κάποια IP διεύθυνση.

```
Host1
Dedicated to Dialing.
Build time: Jun 1 2015 11:42:32
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

Host1>
Host1> show

NAME IP/MASK LPORT RHOST:PORT MAC
Host1 0.0.0.0/0 0.0.0.0
0:79:66:68:01:10032 127.0.0.1:10033
fe80::250:79ff:fe66:6801/64

Host1>

Host2
Welcome to Virtual PC Simulator, version 0.6.1
Please use Ctrl-Alt-F1 to activate the console.
Type escape sequence to abort.
Verifying the previous NVRAM configuration[confirm]
Building configuration...
[OK]
Host2#
Host2#ping 8.8.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.10.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/44/48 ms
Host2#debug ip icmp
ICMP packet debugging is on
Host2#
*Aug 4 16:47:00.919: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, topology BASE, dsc
0 0 topoid 0
*Aug 4 16:47:00.983: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, topology BASE, dsc
0 0 topoid 0
*Aug 4 16:47:01.047: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, topology BASE, dsc
0 0 topoid 0
*Aug 4 16:47:01.115: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, topology BASE, dsc
0 0 topoid 0
*Aug 4 16:47:01.179: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, topology BASE, dsc
0 0 topoid 0
Host2#
```

Εικόνα 6-159 Άνοιγμα των Hosts (VPCs)

Δημιουργία DHCP pool host1 στον CS1

```
CS1
CS1(config)#ip dhcp pool Host1
CS1(dhcp-config)#network 10.1.3.0 255.255.255.0
CS1(dhcp-config)#default-router 10.1.3.1
CS1(dhcp-config)#end
CS1#wr
*Aug 4 19:52:37.714: %SYS-5-CONFIG_I: Configured from console by console
CS1#wr
```

Στη συνέχεια έχοντας ανοιχτό το host1 και γράφοντας την εντολή ip dhcp παίρνει αυτόματα IP διεύθυνση του δικτύου 10.1.3.0, την επόμενη από αυτή που έχει ο router.



```
Host1
      LPORT  RHOST:PORT
Host1  0.0.0.0/0      0.0.0.0      00:5
0:79:66:68:01  10032  127.0.0.1:10033
      fe80::250:79ff:fe66:6801/64

Host1> ip dhcp
DDORA IP 10.1.3.2/24 GW 10.1.3.1

Host1> save
Saving startup configuration to startup.vpc
. done

Host1> save startup.vpc
Saving startup configuration to startup.vpc
. done

Host1>
```

Εικόνα 6-160 Απόδοση IP διεύθυνσης με DHCP στον Host1

Δημιουργία DHCP pool host2 CS2

```
Filter
CS2
*Aug 4 16:47:01.115: ICMP: echo reply sent, src 8.8.11.2, dst 8.8.10.2, top
CS2#
CS2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
CS2(config)#ip dhcp pool Host1
CS2(dhcp-config)#network 10.1.4.0 255.255.255.0
CS2(dhcp-config)#default-router 10.1.4.1
CS2(dhcp-config)#end
CS2#wr
*Aug 4 20:05:52.299: %SYS-5-CONFIG_I: Configured from console by console
CS2#wr
Building configuration...
[OK]
CS2#

Host2
Executing the startup file

Host2>
Host2> show
NAME IP/MASK GATEWAY MAC LPORT RHOST:PORT
Host2 0.0.0.0/0 0.0.0.0 00:50:79:66:68:00 10034 127.0.0.1:10
      fe80::250:79ff:fe66:6800/64

Host2> IP DHCP
Bad command: "IP DHCP". Use ? for help.

Host2> ip dhcp
DDORA IP 10.1.4.2/24 GW 10.1.4.1
```

Εικόνα 6-161 Απόδοση IP διεύθυνσης με DHCP στον Host2

Show ip dhcp binding στον CS1

```
CS1#sh ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/      Lease expiration    Type      State      Interface
Hardware address/
User name
10.1.3.2        0100.5079.6668.01  Aug 05 2018 07:53 PM Automatic Active GigabitEthernet2/0
CS1#
```

Βλέπουμε τη διεύθυνση που δεσμεύτηκε και το ID του client που τη δέσμευσε

```
Host1
Source code and license can be found at vpcs.sf.net.
For more information, please visit Wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

D
DDORA IP 10.1.3.2/24 GW 10.1.3.1

Host1>
Host1> ping 10.1.3.1
64 bytes from 10.1.3.1 icmp_seq=1 ttl=255 time=9.920 ms
64 bytes from 10.1.3.1 icmp_seq=2 ttl=255 time=9.921 ms
64 bytes from 10.1.3.1 icmp_seq=3 ttl=255 time=0.992 ms
64 bytes from 10.1.3.1 icmp_seq=4 ttl=255 time=10.912 ms
64 bytes from 10.1.3.1 icmp_seq=5 ttl=255 time=8.928 ms

Host1>

Host2
Host2>
Host2> ip dhcp
DDORA IP 10.1.4.2/24 GW 10.1.4.1

Host2> save
Saving startup configuration to startup.vpc
. done

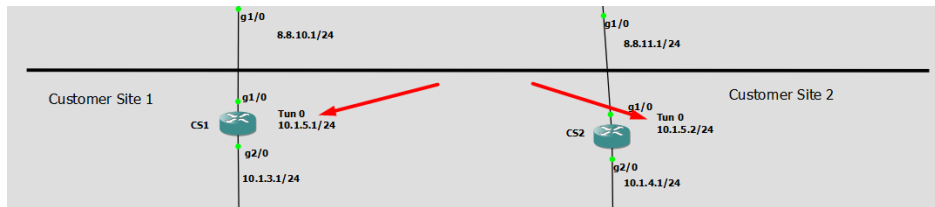
Host2> save startup.vpc
Saving startup configuration to startup.vpc
. done

Host2> ping 10.1.4.1
64 bytes from 10.1.4.1 icmp_seq=1 ttl=255 time=9.928 ms
64 bytes from 10.1.4.1 icmp_seq=2 ttl=255 time=9.920 ms
64 bytes from 10.1.4.1 icmp_seq=3 ttl=255 time=7.936 ms
64 bytes from 10.1.4.1 icmp_seq=4 ttl=255 time=7.935 ms
64 bytes from 10.1.4.1 icmp_seq=5 ttl=255 time=11.904 ms
```

Εικόνα 6-162 Pings των hosts στην default gateway



Το επόμενο βήμα το οποίο πρέπει να κάνουμε για να ολοκληρώσουμε το πείραμα μας είναι να φτιάξουμε το tunnel μεταξύ των δύο Customer routers. Έχουμε ορίσει σαν tunnel interface στον CS1 το tun 0 με ip 10.1.5.1/24 και για τον CS2 τον tun 0 με ip 10.1.5.2/24



Εικόνα 6-163 Tunnel interfaces

Πηγαίνουμε στον CS1 και ρυθμίζουμε το tunnel interface με τις παρακάτω εντολές. Δίνουμε IP στο tunnel interface του δικτύου 10.1.5.0. Του λέμε από που ξεκινάει το tunnel δηλαδή από την g1/0 του CS1 και έχει προορισμό την εξωτερική (public) ip διεύθυνση του CS2 8.8.11.2

```
CS1#conf t
```

```
CS1(config)#int tunnel 0
```

```
CS1(config-if)#
```

```
*Aug 5 11:20:50.239: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to down
```

```
CS1(config-if)#ip address 10.1.5.1 255.255.255.0
```

```
CS1(config-if)#tunnel source gigabitEthernet 1/0
```

```
CS1(config-if)#tunnel destination 8.8.11.2
```

```
CS1(config-if)#
```

```
*Aug 5 11:23:02.423: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up
```

```
CS1(config-if)#end
```

```
Aug 5 11:20:51.997: %SYS-5-CONFIG_I: Configured from console by console
CS1#sh ip int brief
Interface                IP-Address      OK? Method Status        Protocol
FastEthernet0/0          unassigned      YES NVRAM   administratively down  down
GigabitEthernet1/0       8.8.10.2        YES DHCP    up            up
GigabitEthernet2/0       10.1.3.1        YES NVRAM   up            up
GigabitEthernet3/0       unassigned      YES NVRAM   administratively down  down
GigabitEthernet4/0       unassigned      YES NVRAM   administratively down  down
Tunnel0                   10.1.5.1        YES manual up            up
CS1#
```

Εικόνα 6-164 Tunnel0 ενεργοποιημένο για τον CS1

Μέχρι στιγμής το tunnel δεν δουλεύει γιατί πρέπει να πάμε να κατασκευάσουμε tunnel και από την άλλη πλευρά δηλαδή στον CS2, ακολουθώντας την ίδια λογική στο source και στο destination για τον CS2 έχουμε τον παρακάτω κώδικα στον δρομολογητή



CS2#conf t

Enter configuration commands, one per line. End with CNTL/Z.

CS2(config)#int tun 0

CS2(config-if)#ip

*Aug 5 11:31:19.027: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to down

CS2(config-if)#ip add 10.1.5.2 255.255.255.0

CS2(config-if)#tunnel source gigabitEthernet 1/0

CS2(config-if)# tunnel destination 8.8.10.2

CS2(config-if)#

*Aug 5 11:33:27.075: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up

CS2(config-if)#end

```
CS2#sh ip int brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    unassigned      YES NVRAM   administratively down down
GigabitEthernet1/0 8.8.11.2        YES DHCP    up          up
GigabitEthernet2/0 10.1.4.1        YES NVRAM   up          up
GigabitEthernet3/0 unassigned      YES NVRAM   administratively down down
GigabitEthernet4/0 unassigned      YES NVRAM   administratively down down
Tunnel0            10.1.5.2        YES manual up          up
CS2#
```

Εικόνα 6-165 Tunnel0 ενεργοποιημένο για τον CS1

Τώρα θα πάμε να ελέγξουμε την επικοινωνία από τον ένα δρομολογητή στον άλλον διαμέσω του tunnel interface κάνοντας ping στα interfaces. Όπως βλέπουμε στην παρακάτω εικόνα 6-166 η επικοινωνία δουλεύει και από τις δύο πλευρές (CS1->CS2 , CS2->CS1)

```
CS1#sh ip int brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    unassigned      YES NVRAM   administratively down down
GigabitEthernet1/0 8.8.10.2        YES DHCP    up          up
GigabitEthernet2/0 10.1.3.1        YES NVRAM   up          up
GigabitEthernet3/0 unassigned      YES NVRAM   administratively down down
GigabitEthernet4/0 unassigned      YES NVRAM   administratively down down
Tunnel0            10.1.5.1        YES manual up          up
CS1#

CS1#ping 8.8.11.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.11.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 116/124/144 ms
CS1#ping 10.1.5.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.5.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/65/68 ms
CS1#

CS2#sh ip int brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    unassigned      YES NVRAM   administratively down down
GigabitEthernet1/0 8.8.11.2        YES DHCP    up          up
GigabitEthernet2/0 10.1.4.1        YES NVRAM   up          up
GigabitEthernet3/0 unassigned      YES NVRAM   administratively down down
GigabitEthernet4/0 unassigned      YES NVRAM   administratively down down
Tunnel0            10.1.5.2        YES manual up          up
CS2#

CS2#ping 8.8.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.10.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/119/128 ms
CS2#

CS2#ping 10.1.5.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.5.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/43/48 ms
CS2#
```

Εικόνα 6-166 Ping στα tunnel interfaces vice versa

Το νόημα είναι ότι ο πυρήνας του δικτύου, ο ISP2 δρομολογητής δηλαδή, δεν γνωρίζει για αυτό το tunnel του δικτύου 10.1.5.0 (.5.1 για τον CS1 και .5.2 για τον CS2) διότι αυτό ενθυλακώνεται πάνω στις public IP των δρομολογητών, αυτό το διαπιστώνουμε με την παρακάτω εντολή show ip route όπου στα routing tables του ISP2 δεν υπάρχει αυτό το δίκτυο άλλα η επικοινωνία λειτουργεί κανονικά.



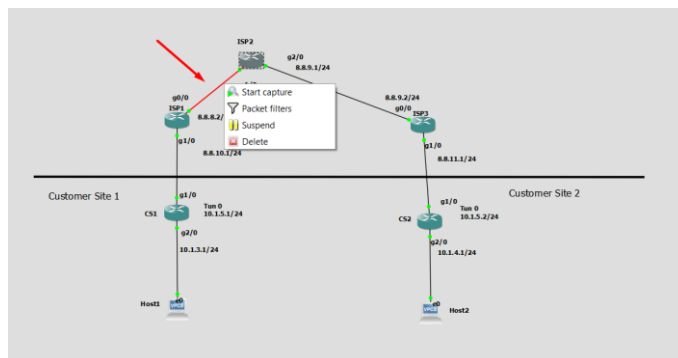
```
ISP2
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override
Gateway of last resort is 8.8.8.2 to network 0.0.0.0

S*  0.0.0.0/0 [1/0] via 8.8.8.2
O   1.0.0.0/32 is subnetted, 1 subnets
O   1.1.1.1 [110/2] via 8.8.8.2, 00:51:32, GigabitEthernet1/0
C   2.0.0.0/32 is subnetted, 1 subnets
C   2.2.2.2 is directly connected, Loopback0
O   3.0.0.0/32 is subnetted, 1 subnets
O   3.3.3.3 [110/2] via 8.8.9.2, 00:51:42, GigabitEthernet2/0
O   8.0.0.0/8 is variably subnetted, 10 subnets, 2 masks
C   8.8.4.0/24 is directly connected, GigabitEthernet3/0
C   8.8.4.1/32 is directly connected, GigabitEthernet3/0
C   8.8.7.0/24 is directly connected, GigabitEthernet0/0
C   8.8.7.2/32 is directly connected, GigabitEthernet0/0
C   8.8.8.0/24 is directly connected, GigabitEthernet1/0
C   8.8.8.1/32 is directly connected, GigabitEthernet1/0
C   8.8.9.0/24 is directly connected, GigabitEthernet2/0
C   8.8.9.1/32 is directly connected, GigabitEthernet2/0
B   8.8.10.0/24 [200/0] via 1.1.1.1, 00:50:25
B   8.8.11.0/24 [200/0] via 3.3.3.3, 00:50:25
```

Εικόνα 6-167 Routing table του ISP2 χωρίς την γνώση του δικτύου 10.1.5.0 (tunnel interfaces)

6.6.1 Έλεγχος της κίνησης του tunnel διαμέσων του Wireshark

Τώρα θα πάμε να ελέγξουμε αυτήν την κίνηση μέσα από το περιβάλλον του Wireshark και να δούμε ότι όντως οι IP των tunnel interfaces έχουν όντως ενθυλακωθεί στις public IP των δρομολογητών CS1 και CS2. Πηγαίνουμε στο link της παρακάτω εικόνας και κάνοντας δεξί κλικ στο link, πατώντας start capture ανοίγει το Wireshark που έχουμε εισάγει στο περιβάλλον του GNS3 με τον τρόπο που αναφέραμε και πιο πάνω στο σενάριο του οργανισμού που προσομοιώσαμε.



Εικόνα 6-168 Capture του link μέσα από το Wireshark

Έχοντας ανοίξει το Wireshark κάνουμε ping από τον CS1 στον CS2 στην IP του tunnel0 interface 10.1.5.2 για να δημιουργήσουμε κίνηση. Βάζοντας σαν φίλτρο στον Wireshark το ICMP για να δούμε τα Echo request και Echo reply βλέπουμε τα εξής.



Capturing from Standard input [ISP2 GigabitEthernet1/0 to ISP1 GigabitEthernet0/0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|-------------------|-------------------|----------|--------|---|
| 132 | 113.681443 | 8.8.8.1 | 224.0.0.5 | OSPF | 94 | Hello Packet |
| 133 | 114.226546 | 10.1.5.1 | 10.1.5.2 | ICMP | 138 | Echo (ping) request id=0x0003, seq=0/0, ttl=255 (reply in 134) |
| 134 | 114.270195 | 10.1.5.2 | 10.1.5.1 | ICMP | 138 | Echo (ping) reply id=0x0003, seq=0/0, ttl=255 (request in 133) |
| 135 | 114.302929 | 10.1.5.1 | 10.1.5.2 | ICMP | 138 | Echo (ping) request id=0x0003, seq=1/256, ttl=255 (reply in 136) |
| 136 | 114.346577 | 10.1.5.2 | 10.1.5.1 | ICMP | 138 | Echo (ping) reply id=0x0003, seq=1/256, ttl=255 (request in 135) |
| 137 | 114.379313 | 10.1.5.1 | 10.1.5.2 | ICMP | 138 | Echo (ping) request id=0x0003, seq=2/512, ttl=255 (reply in 138) |
| 138 | 114.421969 | 10.1.5.2 | 10.1.5.1 | ICMP | 138 | Echo (ping) reply id=0x0003, seq=2/512, ttl=255 (request in 137) |
| 139 | 114.454705 | 10.1.5.1 | 10.1.5.2 | ICMP | 138 | Echo (ping) request id=0x0003, seq=3/768, ttl=255 (reply in 140) |
| 140 | 114.498353 | 10.1.5.2 | 10.1.5.1 | ICMP | 138 | Echo (ping) reply id=0x0003, seq=3/768, ttl=255 (request in 139) |
| 141 | 114.531089 | 10.1.5.1 | 10.1.5.2 | ICMP | 138 | Echo (ping) request id=0x0003, seq=4/1024, ttl=255 (reply in 142) |
| 142 | 114.574737 | 10.1.5.2 | 10.1.5.1 | ICMP | 138 | Echo (ping) reply id=0x0003, seq=4/1024, ttl=255 (request in 141) |
| 143 | 115.499777 | ca:08:0c:c0:00:1c | ca:08:0c:c0:00:1c | LOOP | 60 | Reply |

> Frame 133: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface 0
> Ethernet II, Src: ca:09:04:b8:00:08 (ca:09:04:b8:00:08), Dst: ca:08:0c:c0:00:1c (ca:08:0c:c0:00:1c)
> Internet Protocol Version 4, Src: 8.8.10.2, Dst: 8.8.11.2
> Generic Routing Encapsulation (IP)
> Internet Protocol Version 4, Src: 10.1.5.1, Dst: 10.1.5.2
> Internet Control Message Protocol

Εικόνα 6-169 Κίνηση μέσα από το περιβάλλον του Wireshark

Από την παραπάνω εικόνα βλέπουμε την κίνηση που υπάρχει από τον CS1 προς τον CS2 το θέμα είναι ότι αυτή η κίνηση έχει ενθυλακωθεί σε αυτήν εδώ την κίνηση που βλέπουμε στο παρακάτω στιγμιότυπο οθόνης. Όπου φαίνεται ότι source και dest ip address που βλέπουν οι ISP είναι οι public IP διευθύνσεις των δρομολογητών (8.8.10.2 του CS1 και 8.8.11.2 του CCS2).

Capturing from Standard input [ISP2 GigabitEthernet1/0 to ISP1 GigabitEthernet0/0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

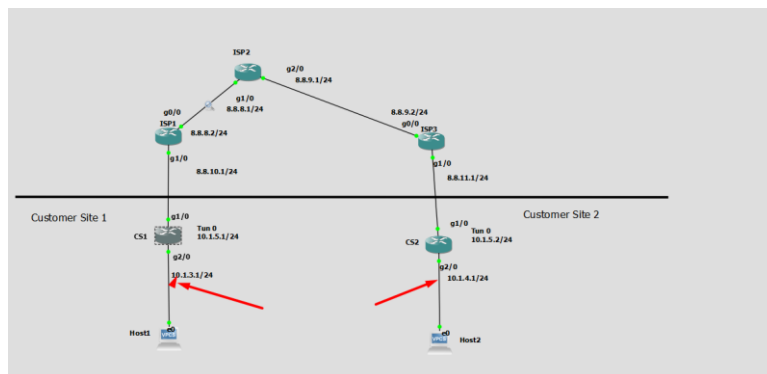
Apply a display filter ... <Ctrl-F>

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|-------------------|-------------------|----------|--------|---|
| 132 | 113.681443 | 8.8.8.1 | 224.0.0.5 | OSPF | 94 | Hello Packet |
| 133 | 114.226546 | 10.1.5.1 | 10.1.5.2 | ICMP | 138 | Echo (ping) request id=0x0003, seq=0/0, ttl=255 (reply in 134) |
| 134 | 114.270195 | 10.1.5.2 | 10.1.5.1 | ICMP | 138 | Echo (ping) reply id=0x0003, seq=0/0, ttl=255 (request in 133) |
| 135 | 114.302929 | 10.1.5.1 | 10.1.5.2 | ICMP | 138 | Echo (ping) request id=0x0003, seq=1/256, ttl=255 (reply in 136) |
| 136 | 114.346577 | 10.1.5.2 | 10.1.5.1 | ICMP | 138 | Echo (ping) reply id=0x0003, seq=1/256, ttl=255 (request in 135) |
| 137 | 114.379313 | 10.1.5.1 | 10.1.5.2 | ICMP | 138 | Echo (ping) request id=0x0003, seq=2/512, ttl=255 (reply in 138) |
| 138 | 114.421969 | 10.1.5.2 | 10.1.5.1 | ICMP | 138 | Echo (ping) reply id=0x0003, seq=2/512, ttl=255 (request in 137) |
| 139 | 114.454705 | 10.1.5.1 | 10.1.5.2 | ICMP | 138 | Echo (ping) request id=0x0003, seq=3/768, ttl=255 (reply in 140) |
| 140 | 114.498353 | 10.1.5.2 | 10.1.5.1 | ICMP | 138 | Echo (ping) reply id=0x0003, seq=3/768, ttl=255 (request in 139) |
| 141 | 114.531089 | 10.1.5.1 | 10.1.5.2 | ICMP | 138 | Echo (ping) request id=0x0003, seq=4/1024, ttl=255 (reply in 142) |
| 142 | 114.574737 | 10.1.5.2 | 10.1.5.1 | ICMP | 138 | Echo (ping) reply id=0x0003, seq=4/1024, ttl=255 (request in 141) |
| 143 | 115.499777 | ca:08:0c:c0:00:1c | ca:08:0c:c0:00:1c | LOOP | 60 | Reply |

> Frame 133: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface 0
> Ethernet II, Src: ca:09:04:b8:00:08 (ca:09:04:b8:00:08), Dst: ca:08:0c:c0:00:1c (ca:08:0c:c0:00:1c)
> Internet Protocol Version 4, Src: 8.8.10.2, Dst: 8.8.11.2
> Generic Routing Encapsulation (IP)
> Internet Protocol Version 4, Src: 10.1.5.1, Dst: 10.1.5.2
> Internet Control Message Protocol

Εικόνα 6-170 Ενθυλακωμένη κίνηση στις public ip των δρομολογητών

Έχοντας τελειώσει το κομμάτι του tunneling αυτό που θα κάνουμε τώρα είναι να ενεργοποιήσουμε το πρωτόκολλο EIGRP έτσι ώστε ο CS1 να μάθει για το εσωτερικό δίκτυο του CS2 10.1.4.1 και ο CS2 για το εσωτερικό δίκτυο του CS1 10.1.3.1



Εικόνα 6-171 Ενεργοποίηση πρωτοκόλλου EIGRP για την εκμάθηση των εσωτερικών δικτύων



Πηγαίνουμε στον CS1 και βλέπουμε τους πίνακες δρομολόγησης που έχει και δεν έχει το δίκτυο 10.1.4.0 και προσπάθειες για επικοινωνία είναι ανεπιτυχείς. Το ίδιο φυσικά συμβαίνει και από την πλευρά του CS2.

```
C      10.1.3.0/24 is directly connected, GigabitEthernet2/0
L      10.1.3.1/32 is directly connected, GigabitEthernet2/0
C      10.1.5.0/24 is directly connected, Tunnel0
L      10.1.5.1/32 is directly connected, Tunnel0
CS1# ping 10.1.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.4.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
CS1#
```

Εικόνα 6-172 Ανεπιτυχείς προσπάθειες επικοινωνίας με εσωτερικό δίκτυο του CS2

Γράφοντας τον παρακάτω κώδικα και στους δύο δρομολογητές ενεργοποιείται το πρωτόκολλο γειτνίασης EIGRP, όπου διαφημίζουμε το δίκτυο 10.0.0.0 που έχουν τα εσωτερικά μας, με το no auto-summary δεν θέλουμε να κάνει συνάθροιση τα υποδίκτυα, απλά θέλουμε να τα διαφημίσει όπως είναι. Αν θέλουμε σβήνουμε το no και τρέχουμε την εντολή ξανά .

```
CS1#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
CS1(config)#router eigrp 100
```

```
CS1(config-router)#network 10.0.0.0
```

```
CS1(config-router)#no auto-summary
```

```
CS1(config-router)#end
```

```
CS2#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
CS2(config)#router eigrp 100
```

```
CS2(config-router)#network 10.0.0.0
```

```
CS2(config-router)#no auto-summary
```

```
CS2(config-router)#end
```



```
CS1 10.1.3.1/32 is directly connected, GigabitEthernet2/0
C 10.1.5.0/24 is directly connected, Tunnel0
L 10.1.5.1/32 is directly connected, Tunnel0
CS1# ping 10.1.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.4.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
CS1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
CS1(config)#router eigrp 100
CS1(config-router)#network 10.0.0.0
CS1(config-router)#no auto-summary
*
% Invalid input detected at '^' marker.
CS1(config-router)#no auto-summary
CS1(config-router)#
CS1#
*Aug 5 12:31:31.035: %SYS-5-CONFIG_I: Configured from console by console
CS1#
*Aug 5 12:32:49.815: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.1.5.2 (Tunnel0) is up: new adjacency
CS1#

CS2 Success rate is 100 percent (5/5), round-trip min/avg/max = 112/119/128 ms
CS2#
CS2#sh ip int brief
Interface IP-Address OK? Method Status Prot
FastEthernet0/0 unassigned YES NVRAM administratively down down
GigabitEthernet1/0 8.8.11.2 YES DHCP up up
GigabitEthernet2/0 10.1.4.1 YES NVRAM up up
GigabitEthernet3/0 unassigned YES NVRAM administratively down down
GigabitEthernet4/0 unassigned YES NVRAM administratively down down
Tunnel0 10.1.5.2 YES manual up up
CS2#ping 10.1.5.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.5.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/43/48 ms
CS2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
CS2(config)#router eigrp 100
CS2(config-router)#network 10.0.0.0
CS2(config-router)#no auto-summary
*Aug 5 12:32:49.299: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.1.5.1 (Tunnel0) is up: new adjacency
CS2(config-router)#no auto-summary
CS2(config-router)#
```

Εικόνα 6-173 Ενεργοποίηση του EIGRP διαμέσω του tunnel 0

Παρατηρούμε ότι ο CS1 μαθαίνει το εσωτερικό δίκτυο του CS2 χρησιμοποιώντας το tun0 interface και το πρωτόκολλο γειτνίασης EIGRP.

Γράφοντας στον CS1 `sh ip eigrp neighbors` βλέπουμε ότι γνωρίζει σαν γείτονα το εσωτερικό δίκτυο του CS2 10.1.5.2

```
CS1#sh ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)
H Address Interface Hold Uptime SRTT RTO Q Se
q (sec) (ms) Cnt Nu
m
0 10.1.5.2 Tu0 10 00:07:21 66 1470 0 3
CS1#
```

Εικόνα 6-174 Ενεργοποίηση του EIGRP διαμέσω του tunnel 0

Επίσης γράφοντας τώρα `sh ip route` στον CS1 βλέπουμε ότι έμαθε το εσωτερικό δίκτυο του CS2 via 10.1.5.2 που είναι το tunnel interface χρησιμοποιώντας το EIGRP που τα καταλαβαίνουμε από την ένδειξη D που φαίνεται στο παρακάτω στιγμιότυπο.

```
CS1
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is 8.8.10.1 to network 0.0.0.0

S* 0.0.0.0/0 [1/0] via 8.8.10.1
8.8.10.0/8 is variably subnetted, 2 subnets, 2 masks
C 8.8.10.0/24 is directly connected, GigabitEthernet1/0
L 8.8.10.2/32 is directly connected, GigabitEthernet1/0
10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C 10.1.3.0/24 is directly connected, GigabitEthernet2/0
L 10.1.3.1/32 is directly connected, GigabitEthernet2/0
D 10.1.4.0/24 [90/26880256] via 10.1.5.2, 00:10:30, Tunnel0
C 10.1.5.0/24 is directly connected, Tunnel0
L 10.1.5.1/32 is directly connected, Tunnel0
CS1#
```

Εικόνα 6-175 Ανανεωμένος πίνακας δρομολόγησης

Στη συνέχεια επιβεβαιώνουμε ότι οι Customers μπορούν να επικοινωνήσουν και να στείλουν πακέτα στα εσωτερικά δίκτυα ό ένας στον άλλον



```

CS1
Gateway of last resort is 8.8.10.1 to network 0.0.0.0

S*  0.0.0.0/0 [1/0] via 8.8.10.1
C   8.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C   8.8.10.0/24 is directly connected, GigabitEthernet1/0
L   8.8.10.2/32 is directly connected, GigabitEthernet1/0
C   10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C   10.1.3.0/24 is directly connected, GigabitEthernet2/0
L   10.1.3.1/32 is directly connected, GigabitEthernet2/0
B   10.1.4.0/24 [90/26880256] via 10.1.5.2, 00:10:30, Tunnel0
C   10.1.5.0/24 is directly connected, Tunnel0
L   10.1.5.1/32 is directly connected, Tunnel0
CS1#ping 10.1.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.4.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/56/68 ms
CS1#

CS2
CS2(config-router)#no auto
*Aug 5 12:32:49.299: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.1.5.1 (
mne10) is up: new adjacency
CS2(config-router)#no auto-summary
CS2(config-router)#exit
CS2(config)#exit
CS2#vr
Building configuration...
[OK]
CS2#
*Aug 5 12:46:00.551: %SYS-5-CONFIG I: Configured from console by console
CS2#ping 10.1.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/51/56 ms
CS2#
  
```

Εικόνα 6-176 Επιτυχής επικοινωνία στα εσωτερικά δίκτυα των δρομολογητών

Το τελευταίο τεστ που πρέπει να κάνουμε είναι να δούμε ότι οι hosts μπορούν να επικοινωνήσουν μεταξύ τους, από το παρακάτω screenshot βλέπουμε ότι επικοινωνούν επιτυχώς χωρίς κάποιο πρόβλημα.

```

Host1
NAME IP/MASK GATEWAY MAC LPORT RHOST:IP
---
Host1 10.1.3.2/24 10.1.3.1 00:50:79:66:68:01 10034 127.0.0.1:10035
fe80::250:79ff:fe66:6800/64

Host1> ping 10.1.4.2
10.1.4.2 icmp_seq=1 timeout
10.1.4.2 icmp_seq=2 timeout
84 bytes from 10.1.4.2 icmp_seq=3 ttl=62 time=91.344 ms
84 bytes from 10.1.4.2 icmp_seq=4 ttl=62 time=77.376 ms
84 bytes from 10.1.4.2 icmp_seq=5 ttl=62 time=82.336 ms

Host1> ping 10.1.4.2
84 bytes from 10.1.4.2 icmp_seq=1 ttl=62 time=85.807 ms
84 bytes from 10.1.4.2 icmp_seq=2 ttl=62 time=88.288 ms
84 bytes from 10.1.4.2 icmp_seq=3 ttl=62 time=88.287 ms
84 bytes from 10.1.4.2 icmp_seq=4 ttl=62 time=90.272 ms
84 bytes from 10.1.4.2 icmp_seq=5 ttl=62 time=67.455 ms

Host1#

Host2
Host2> ping 10.1.4.1
84 bytes from 10.1.4.1 icmp_seq=1 ttl=255 time=9.928 ms
84 bytes from 10.1.4.1 icmp_seq=2 ttl=255 time=9.920 ms
84 bytes from 10.1.4.1 icmp_seq=3 ttl=255 time=7.936 ms
84 bytes from 10.1.4.1 icmp_seq=4 ttl=255 time=7.935 ms
84 bytes from 10.1.4.1 icmp_seq=5 ttl=255 time=11.904 ms

Host2> show
NAME IP/MASK GATEWAY MAC LPORT RHO
---
Host2 10.1.4.2/24 10.1.4.1 00:50:79:66:68:00 10032 127
fe80::250:79ff:fe66:6800/64

Host2> ping 10.1.3.2
84 bytes from 10.1.3.2 icmp_seq=1 ttl=62 time=87.296 ms
84 bytes from 10.1.3.2 icmp_seq=2 ttl=62 time=87.296 ms
84 bytes from 10.1.3.2 icmp_seq=3 ttl=62 time=66.464 ms
84 bytes from 10.1.3.2 icmp_seq=4 ttl=62 time=67.456 ms
84 bytes from 10.1.3.2 icmp_seq=5 ttl=62 time=76.384 ms

Host2#
  
```

Εικόνα 6-177 Επιτυχής επικοινωνία των hosts

Στέλνουμε ακόμη μία φορά πακέτο από το Host 1 μέσω της εντολής ping στην ip του Host 2 10.1.4.2 και θα πάμε να παρατηρήσουμε την κίνηση μέσα από το Wireshark

```

Standard Input [SP2 GigabitEthernet1/0 to SP1 GigabitEthernet0/0]
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp
Time Source Destination Protocol Length Info
---
7683 3711.640950 10.1.3.2 10.1.4.2 ICMP 122 Echo (ping) request id=0x81cc, seq=5/1280, ttl=63 (reply in 7684)
7684 3711.685683 10.1.4.2 10.1.3.2 ICMP 122 Echo (ping) reply id=0x81cc, seq=5/1280, ttl=63 (request in 7683)
7689 3715.755849 10.1.3.2 10.1.4.2 ICMP 122 Echo (ping) request id=0x85cc, seq=1/256, ttl=63 (reply in 7690)
7690 3715.804221 10.1.4.2 10.1.3.2 ICMP 122 Echo (ping) reply id=0x85cc, seq=1/256, ttl=63 (request in 7689)
7694 3716.855974 10.1.3.2 10.1.4.2 ICMP 122 Echo (ping) request id=0x86cc, seq=2/512, ttl=63 (reply in 7695)
7695 3716.918534 10.1.4.2 10.1.3.2 ICMP 122 Echo (ping) reply id=0x86cc, seq=2/512, ttl=63 (request in 7694)
7699 3717.954611 10.1.3.2 10.1.4.2 ICMP 122 Echo (ping) request id=0x87cc, seq=3/768, ttl=63 (reply in 7700)
7700 3718.009171 10.1.4.2 10.1.3.2 ICMP 122 Echo (ping) reply id=0x87cc, seq=3/768, ttl=63 (request in 7699)
7702 3719.044817 10.1.3.2 10.1.4.2 ICMP 122 Echo (ping) request id=0x88cc, seq=4/1024, ttl=63 (reply in 7703)
7703 3719.098881 10.1.4.2 10.1.3.2 ICMP 122 Echo (ping) reply id=0x88cc, seq=4/1024, ttl=63 (request in 7702)
7705 3720.143950 10.1.3.2 10.1.4.2 ICMP 122 Echo (ping) request id=0x89cc, seq=5/1280, ttl=63 (reply in 7706)
7706 3720.198310 10.1.4.2 10.1.3.2 ICMP 122 Echo (ping) reply id=0x89cc, seq=5/1280, ttl=63 (request in 7705)

Frame 7694: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on Interface 0
> Ethernet II, Src: ca:09:04:b8:00:08 (ca:09:04:b8:00:08), Dst: ca:08:0c:c0:00:1c (ca:08:0c:c0:00:1c)
> Internet Protocol Version 4, Src: 8.8.10.2, Dst: 8.8.11.2
> Generic Routing Encapsulation (IP)
> Internet Protocol Version 4, Src: 10.1.3.2, Dst: 10.1.4.2
> Internet Control Message Protocol
  
```

Εικόνα 6-178 Παρακολούθηση κίνησης διαμέσων Wireshark

Από το στιγμιότυπο 6-178 βλέπουμε τα ICMP πακέτα που στέλνει ο Host 1 στον Host 2 Source 10.1.3.2 dest 10.1.4.2. Στο layer2 το δεύτερο κόκκινο βέλος βλέπουμε τις MAC addresses των ISP δρομολογητών.



Επίσης βλέπουμε τις IPv4 διευθύνσεις που χρησιμοποιούνται στο Internet, όπου ο CS1 (8.8.10.2) στέλνει πακέτα κίνησης στον CS2 (8.8.11.2)

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-------------|----------|-------------|----------|--------|---|
| 7683 | 3711.640050 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x81cc, seq=5/1280, ttl=63 (reply in 7684) |
| 7684 | 3711.685683 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x81cc, seq=5/1280, ttl=63 (request in 7683) |
| 7689 | 3715.755849 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x85cc, seq=1/256, ttl=63 (reply in 7690) |
| 7690 | 3715.810421 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x85cc, seq=1/256, ttl=63 (request in 7689) |
| 7694 | 3716.855974 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x86cc, seq=2/512, ttl=63 (reply in 7695) |
| 7695 | 3716.910534 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x86cc, seq=2/512, ttl=63 (request in 7694) |
| 7699 | 3717.954611 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x87cc, seq=3/768, ttl=63 (reply in 7700) |
| 7700 | 3718.009171 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x87cc, seq=3/768, ttl=63 (request in 7699) |
| 7702 | 3719.044817 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x88cc, seq=4/1024, ttl=63 (reply in 7703) |
| 7703 | 3719.098881 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x88cc, seq=4/1024, ttl=63 (request in 7702) |
| 7705 | 3720.143950 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x89cc, seq=5/1280, ttl=63 (reply in 7706) |
| 7706 | 3720.198510 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x89cc, seq=5/1280, ttl=63 (request in 7705) |

> Frame 7694: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
> Ethernet II, Src: ca:09:04:b8:00:08 (ca:09:04:b8:00:08), Dst: ca:08:0c:c0:00:1c (ca:08:0c:c0:00:1c)
> Internet Protocol Version 4, Src: 8.8.10.2, Dst: 8.8.11.2
> Generic Routing Encapsulation (IP)
> Internet Protocol Version 4, Src: 10.1.3.2, Dst: 10.1.4.2
> Internet Control Message Protocol

Εικόνα 6-179 Παρακολούθηση IPv4 διευθύνσεων του Internet

Ακόμη βλέπουμε τις IP διευθύνσεις των hosts διαμέσων του GRE Tunnel και τα Echo requests και Echo reply

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-------------|----------|-------------|----------|--------|---|
| 7683 | 3711.640050 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x81cc, seq=5/1280, ttl=63 (reply in 7684) |
| 7684 | 3711.685683 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x81cc, seq=5/1280, ttl=63 (request in 7683) |
| 7689 | 3715.755849 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x85cc, seq=1/256, ttl=63 (reply in 7690) |
| 7690 | 3715.810421 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x85cc, seq=1/256, ttl=63 (request in 7689) |
| 7694 | 3716.855974 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x86cc, seq=2/512, ttl=63 (reply in 7695) |
| 7695 | 3716.910534 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x86cc, seq=2/512, ttl=63 (request in 7694) |
| 7699 | 3717.954611 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x87cc, seq=3/768, ttl=63 (reply in 7700) |
| 7700 | 3718.009171 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x87cc, seq=3/768, ttl=63 (request in 7699) |
| 7702 | 3719.044817 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x88cc, seq=4/1024, ttl=63 (reply in 7703) |
| 7703 | 3719.098881 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x88cc, seq=4/1024, ttl=63 (request in 7702) |
| 7705 | 3720.143950 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x89cc, seq=5/1280, ttl=63 (reply in 7706) |
| 7706 | 3720.198510 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x89cc, seq=5/1280, ttl=63 (request in 7705) |

> Frame 7694: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
> Ethernet II, Src: ca:09:04:b8:00:08 (ca:09:04:b8:00:08), Dst: ca:08:0c:c0:00:1c (ca:08:0c:c0:00:1c)
> Internet Protocol Version 4, Src: 8.8.10.2, Dst: 8.8.11.2
> Generic Routing Encapsulation (IP)
> Flags and Version: 0x0000
 Protocol Type: IP (0x0000)
> Internet Protocol Version 4, Src: 10.1.3.2, Dst: 10.1.4.2
> Internet Control Message Protocol

Εικόνα 6-180 Παρακολούθηση κίνησης του GRE Tunneling

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-------------|----------|-------------|----------|--------|---|
| 7683 | 3711.640050 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x81cc, seq=5/1280, ttl=63 (reply in 7684) |
| 7684 | 3711.685683 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x81cc, seq=5/1280, ttl=63 (request in 7683) |
| 7689 | 3715.755849 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x85cc, seq=1/256, ttl=63 (reply in 7690) |
| 7690 | 3715.810421 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x85cc, seq=1/256, ttl=63 (request in 7689) |
| 7694 | 3716.855974 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x86cc, seq=2/512, ttl=63 (reply in 7695) |
| 7695 | 3716.910534 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x86cc, seq=2/512, ttl=63 (request in 7694) |
| 7699 | 3717.954611 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x87cc, seq=3/768, ttl=63 (reply in 7700) |
| 7700 | 3718.009171 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x87cc, seq=3/768, ttl=63 (request in 7699) |
| 7702 | 3719.044817 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x88cc, seq=4/1024, ttl=63 (reply in 7703) |
| 7703 | 3719.098881 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x88cc, seq=4/1024, ttl=63 (request in 7702) |
| 7705 | 3720.143950 | 10.1.3.2 | 10.1.4.2 | ICMP | 122 | Echo (ping) request id=0x89cc, seq=5/1280, ttl=63 (reply in 7706) |
| 7706 | 3720.198510 | 10.1.4.2 | 10.1.3.2 | ICMP | 122 | Echo (ping) reply id=0x89cc, seq=5/1280, ttl=63 (request in 7705) |

> Frame 7694: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
> Ethernet II, Src: ca:09:04:b8:00:08 (ca:09:04:b8:00:08), Dst: ca:08:0c:c0:00:1c (ca:08:0c:c0:00:1c)
> Internet Protocol Version 4, Src: 8.8.10.2, Dst: 8.8.11.2
> Generic Routing Encapsulation (IP)
> Internet Protocol Version 4, Src: 10.1.3.2, Dst: 10.1.4.2
> Internet Control Message Protocol
 Type: 8 (Echo (ping) request)
 Code: 0
 Checksum: 0x993d [correct]
 [Checksum Status: Good]
 Identifier (BE): 8408 (0x81cc)
 Identifier (LE): 52358 (0xcc86)
 Sequence number (BE): 2 (0x0002)

Εικόνα 6-181 Παρακολούθηση ICMP πακέτων



Τέλος βλέπουμε ότι οι ISP routers δεν έχουν καμία ορατότητα στο δίκτυο 10.0.X.X έχουν ορατότητα μονάχα στο δίκτυο 8.8.X.X. Δεν διαφημίζουμε το εσωτερικό δίκτυο των Customers στο Internet. Οι ISP1 και ISP3 τρέχουν BGP και OSPF, οι CS1 και CS2 έχουν ένα static route στην εσωτερική πόρτα του δικτύου τους στην g2/0 δηλαδή, και παίρνουν με DHCP πρωτόκολλο διεύθυνση στην g1/0. Οι CS1 και CS2 τρέχουν μόνο το EIGRP για την γειτνίαση των εσωτερικών δικτύων, όχι BGP ούτε OSPF και η μόνη γειτνίαση που τρέχει είναι μεταξύ τους και με κανέναν άλλον δρομολογητή.

```
CS1
GigabitEthernet2/0 10.1.3.1 YES NVRAM up up
GigabitEthernet3/0 unassigned YES NVRAM administratively down down
GigabitEthernet4/0 unassigned YES NVRAM administratively down down
Tunnel0 10.1.5.1 YES manual up up

CS1#sh ip protocol
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
```

Εικόνα 6-182 Μόνο EIGRP στους Customer δρομολογητές(CS1 και CS2)

```
ISP2
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is 8.8.8.2 to network 0.0.0.0

S* 0.0.0.0/0 [1/0] via 8.8.8.2
  1.0.0.0/32 is subnetted, 1 subnets
O   1.1.1.1 [110/2] via 8.8.8.2, 02:21:10, GigabitEthernet1/0
  2.0.0.0/32 is subnetted, 1 subnets
C   2.2.2.2 is directly connected, Loopback0
O   3.0.0.0/32 is subnetted, 1 subnets
O   3.3.3.3 [110/2] via 8.8.9.2, 02:21:20, GigabitEthernet2/0
  8.0.0.0/8 is variably subnetted, 10 subnets, 2 masks
C   8.8.4.0/24 is directly connected, GigabitEthernet3/0
L   8.8.4.1/32 is directly connected, GigabitEthernet3/0
C   8.8.7.0/24 is directly connected, GigabitEthernet0/0
L   8.8.7.2/32 is directly connected, GigabitEthernet0/0
C   8.8.8.0/24 is directly connected, GigabitEthernet1/0
L   8.8.8.1/32 is directly connected, GigabitEthernet1/0
C   8.8.9.0/24 is directly connected, GigabitEthernet2/0
L   8.8.9.1/32 is directly connected, GigabitEthernet2/0
B   8.8.10.0/24 [200/0] via 1.1.1.1, 02:20:03
B   8.8.11.0/24 [200/0] via 3.3.3.3, 02:20:03

ISP2#
```

Εικόνα 6-183 Κεντρικός ISP πάροχος δεν γνωρίζει το δίκτυο των Customers 10.0.X.X



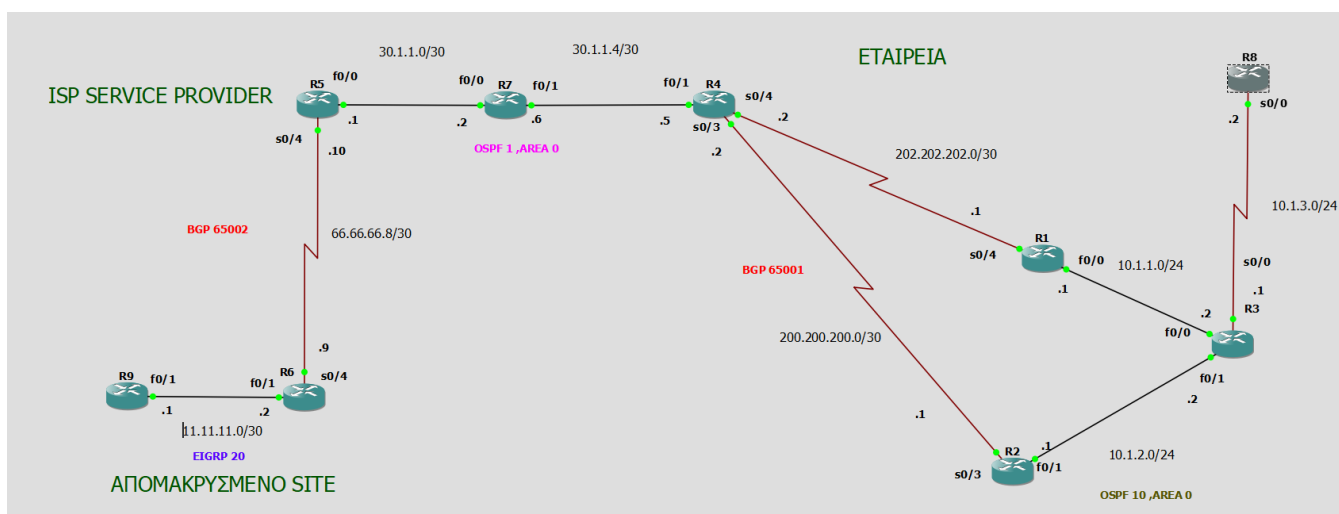
6.7 Redistribution(Ανακατανομή) μεταξύ EGP και IGP πρωτοκόλλων δρομολόγησης

Στην περίπτωση μεγάλων δικτυακών διατάξεων όπως για παράδειγμα στα μητροπολιτικά δίκτυα(MAN) και τα WAN, είναι ξεκάθαρο ότι δεν γίνεται να δουλεύουν όλα τα υποδίκτυα με τη χρήση του ίδιου πρωτοκόλλου δρομολόγησης. Συνεπώς καθίσταται επιτακτική ανάγκη η ύπαρξη μιας λειτουργίας που θα επιτρέπει την επικοινωνία των τερματικών, ανεξάρτητα με το που ανήκουν και με το είδος του πρωτοκόλλου που χρησιμοποιούν. Την απάντηση σε αυτό το πρόβλημα έρχεται να δώσει η ανακατανομή πρωτοκόλλων σε άλλα. Με την ενέργεια αυτή επιτυγχάνουμε την παραμετροποίηση δρομολογητών, που έχουν τη δυνατότητα να γνωρίζουν τις διαδρομές προς τον προορισμό τους, ακόμη και αν αυτές διασχίζουν περιοχές που λειτουργούν υπό διαφορετικό πρωτόκολλο δρομολόγησης.

Η ανακατανομή διαδρομών (Route Redistribution) δίνει τη δυνατότητα σε δρομολογητές να διαφημίσουν τις γνωστές σε αυτούς διαδρομές που τρέχουν ένα X πρωτόκολλο, σε ένα άλλο Y πρωτόκολλο. Το πρωτόκολλο που δέχεται αυτές τις διαδρομές, συνήθως τις καταγράφει ως εξωτερικές (external). Οι εξωτερικές διαδρομές συνήθως προτιμώνται λιγότερο από τις τοπικά καταγεγραμμένες(local directly). Αναγκαία είναι η δημιουργία ενός δρομολογητή ουσιαστικά που θα κάνει την αναδιανομή (Redistribution). Ο δρομολογητής αυτός πρέπει να τρέχει και τα δύο πρωτόκολλα δρομολόγησης.

Το σενάριο που θα δουλέψουμε παρακάτω υλοποιεί redistribute σε διαφορετικά πρωτόκολλα δρομολόγησης, η μορφή της τοπολογίας και τα αναλυτικά βήματα κατασκευής της δίνονται στη συνέχεια.

Μορφή τοπολογίας



Εικόνα 6-184 Τοπολογία Redistribution Διαφορετικών πρωτοκόλλων



Στην παραπάνω τοπολογία βλέπουμε τους δρομολογητές R9 και R6 που προσομοιώνουν ένα απομακρυσμένο site χρησιμοποιώντας πρωτόκολλο EIGRP. Η ΕΤΑΙΡΕΙΑ(δρομολογητές R1,R2,R3,R8) που προσομοιώνεται στη δεξιά πλευρά της τοπολογίας μας τρέχει πρωτόκολλο OSPF. Οι R5,R7,R4 δρομολογητές στην μέση της τοπολογίας που προσομοιώνουν τον ISP PROVIDER τρέχουν και αυτοί OSPF. Ο ISP PROVIDER(R5)επικοινωνεί με το ΑΠΟΜΑΚΡΥΣΜΕΝΟ SITE(R6) με το πρωτόκολλο συνοριακής πύλης BGP. Επίσης Ο ISP PROVIDER από την πλευρά της εταιρείας επικοινωνεί με την εταιρεία χρησιμοποιώντας το πρωτόκολλο BGP,πιο συγκεκριμένα BGP μεταξύ R4-R1 και BGP R4-R2 για να προβλέψουμε ενδεχόμενη βλάβη όπως είδαμε και στο σενάριο του οργανισμού μας στην παραπάνω ενότητα.

Αυτό που πρέπει να κάνουμε πρώτα πριν τρέξουμε όλα τα πρωτόκολλα και υλοποιήσουμε το redistribution πρέπει να δώσουμε στατικά IP στους δρομολογητές μας όπως ακριβώς βλέπουμε στην εικόνα 6-184 στις κατάλληλες πόρτες όπως φαίνεται στις σημειώσεις που έχουμε προσθέσει επάνω στις ζεύξεις. Η στατική απόδοση IP δεν παρουσιάζει κάποιο ιδιαίτερο ενδιαφέρον, την έχουμε τρέξει στους δρομολογητές απλά δεν το παρουσιάζουμε αυτή τη στιγμή, παρατίθεται βέβαια σε αρχείο txt όπως και όλοι οι κώδικες που έχουμε τρέξει. Έχοντας τα δίκτυα με /30 υποδηλώνουμε ότι θέλουμε δύο IP μόνο για δύο hosts εκεί δηλαδή που έχω point to point connection. Χρησιμοποιούμε serial interfaces με τον ISP PROVIDER όπως συνίσταται στους κανόνες της σωστής δικτύωσης για μεγαλύτερη ταχύτητα και αξιοπιστία στα δεδομένα μας.

Έχοντας τελειώσει με την στατική απόδοση των IP διευθύνσεων ξεκινούμε το redistribution από την πλευρά του απομακρυσμένου site όπου θα πάμε να ενεργοποιήσουμε το EIGRP μεταξύ των R6-R9 δρομολογητών.

- Πηγαίνουμε στο δρομολογητή R9 και τρέχουμε τον παρακάτω κώδικα ενεργοποίησης του EIGRP και μετά κάνουμε ένα στατικό route για διαφημίσουμε το FastEthernet0/1 προς τα έξω. Το 0.0.0.3 που έχουμε ορίσει είναι για το δίκτυο που έχουμε ορίσει /30 δύο hosts και η broadcast ip.

```
conf t
router eigrp 20
network 11.11.11.0 0.0.0.3
```

```
-----
ip route 0.0.0.0 0.0.0.0 FastEthernet0/1
```

- Μετά αυτό που πρέπει να κάνουμε είναι να πάμε στο δρομολογητή R6 για να εγκαθιδρύσουμε την EIGRP γειτνίαση με τον R9 δρομολογητή.

```
conf t
router eigrp 20
network 11.11.11.0 0.0.0.3
```

```
R6(config)# router eigrp 20
R6(config-router)# network 11.11.11.0 0.0.0.3
R6(config-router)#
*Mar  1 01:54:35.191: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 20: Neighbor 11.11.11.1 (FastEthernet0/1) is up: new adjacency
R6(config-router)#
```

Εικόνα 6-185 Επιτυχημένη γειτνίαση μεταξύ των R9-R6



- Επόμενη κίνηση που πρέπει να κάνουμε στον R6 είναι να γίνει το Redistribute χρησιμοποιώντας το exterior gateway protocol BGP. Αυτό θα το επιτύχουμε με τον παρακάτω κώδικα

*router bgp 65002 /*65002 είναι το AS στην αριστερή πλευρά της τοπολογίας η πλευρά με το απομακρυσμένο site*/*

redistribute connected / με αυτή την εντολή θα εμφανιστούν μόνο τα άμεσα συνδεδεμένα δίκτυα με αυτό (directly connected) δεν χρειάζεται να τα βάλω manual. Δεν θα μάθει τα άλλα local interfaces που είναι συνδεδεμένα στον δρομολογητή μόνο αυτό που γίνεται η σύνδεση. */*

*neighbor 66.66.66.10 remote-as 65002 /*θέτουμε την συγκεκριμένη ip σαν bgp γείτονα*/*

neighbor 66.66.66.10 next-hop-self / δεξ αυτή τη συγκεκριμένη διεύθυνση σαν next hop και καμία άλλη*/*

Αυτό που θα κάνουμε τώρα είναι να διαφημίσουμε στατικά το serial0/4. Έχοντας τρέξει το route στον R6 κάνουμε sh ip route και βλέπουμε τα δίκτυα, ο R9 δεν διαφημίζει κάτι οπότε δεν εμφανίζεται στον πίνακα και συνεχίζουμε παρακάτω.

ip route 0.0.0.0 0.0.0.0 Serial0/4

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
   66.0.0.0/30 is subnetted, 1 subnets
C       66.66.66.8 is directly connected, Serial0/4
   11.0.0.0/30 is subnetted, 1 subnets
C       11.11.11.0 is directly connected, FastEthernet0/1
S*     0.0.0.0/0 is directly connected, Serial0/4
R6#
```

Εικόνα 6-186 Routing Table του R6

- Έπειτα θα πάμε στον δρομολογητή R5 και θα κάνουμε redistribute το BGP από την πλευρά που είναι συνδεδεμένο με τον R6 στο OSPF από την πλευρά που είναι συνδεδεμένο με τον R7. Κάθε φορά ο ενδιάμεσος δρομολογητής θα κάνει την ανακατανομή.



```
router ospf 1
 redistribute connected subnets
 redistribute bgp 65002 subnets
 network 30.1.1.0 0.0.0.3 area 0
```

```
conf t
router bgp 65002
bgp redistribute-internal /* με την εντολή αυτή του λέμε ότι routes έμαθες
προώθησε τα αυτό δεν γίνεται αυτόματα γιατί δεν περνούν τα routes από exterior
(BGP) σε interior gateway protocols(OSPF)*/
neighbor 66.66.66.9 remote-as 65002 /* remote γείτονας ο R6*/
neighbor 66.66.66.9 next-hop-self /* δεξ αυτή τη συγκεκριμένη διεύθυνση σαν next
hop και καμία άλλη*/
```

Αυτό που θα κάνουμε τώρα είναι να διαφημίσουμε στατικά το serial0/4 *ip route 0.0.0.0 0.0.0.0 serial0/4* και να δούμε ότι έχει επιτευχθεί η γειτνίαση

```
R5#
*Mar 1 03:17:48.987: %BGP-5-ADJCHANGE: neighbor 66.66.66.9 Up
R5#
*Mar 1 03:17:50.251: %SYS-5-CONFIG_I: Configured from console by console
R5#
```

Εικόνα 6-187 Επιτυχής γειτνίαση R5-R6

Κάνοντας show ip route βλέπουμε ότι ο R5 έμαθε ένα δίκτυο διαμέσων του 66.66.66.9 χρησιμοποιώντας το BGP.

```
66.0.0.0/30 is subnetted, 1 subnets
C    66.66.66.8 is directly connected, Serial0/4
B    11.0.0.0/30 is subnetted, 1 subnets
     11.11.11.0 [200/0] via 66.66.66.9, 00:09:29
C    30.0.0.0/30 is subnetted, 1 subnets
     30.1.1.0 is directly connected, FastEthernet0/0
R5#
```

Εικόνα 6-188 Επιτυχής εκμάθηση δικτύου με το BGP



- Στη συνέχεια με την ίδια λογική πηγαίνω στο δρομολογητή **R7** εκεί βέβαια δεν χρειάζεται να τρέξουμε κάτι άλλο πέρα από το OSPF διαφημίζοντας τα δύο γειτονικά του δίκτυα

```
router ospf 1
network 30.1.1.0 0.0.0.3 area 0
network 30.1.1.4 0.0.0.3 area 0
```

```
R7(config)#router ospf 1
R7(config-router)# network 30.1.1.0 0.0.0.3 area 0
R7(config-router)# network 30.1.1.4 0.0.0.3 area 0
R7(config-router)#
*Mar  1 03:41:35.267: %OSPF-5-ADJCHG: Process 1, Nbr 66.66.66.10 on FastEthernet0/0 from LOADING to FULL, Loading Done
R7(config-router)#
```

```

 66.0.0.0/30 is subnetted, 1 subnets
O E2   66.66.66.8 [110/20] via 30.1.1.1, 00:00:50, FastEthernet0/0
 11.0.0.0/30 is subnetted, 1 subnets
O E2   11.11.11.0 [110/1] via 30.1.1.1, 00:00:50, FastEthernet0/0
 30.0.0.0/30 is subnetted, 2 subnets
C      30.1.1.4 is directly connected, FastEthernet0/1
C      30.1.1.0 is directly connected, FastEthernet0/0
R7#
```

Εικόνα 6-189 Routes που έχουν γίνει redistributed στο OSPF

Η ένδειξη OE2 δίπλα στις νέες διαδρομές που έμαθε ο δρομολογητής R7 μας δείχνει ότι το OSPF έμαθε αυτές τα routes από εξωτερικές(external) διαδρομές.

- Συνεχίζουμε στην λογική της αναδιανομής και στον δρομολογητή **R4** ο οποίος από την εσωτερική πλευρά που συνδέεται με τον R7 τρέχει το OSPF και από την άλλη γίνεται γείτονας διαμέσων του BGP με άλλα δύο δίκτυα ένα του R1 και του R2 όπως θα δούμε και στον κώδικα πιο κάτω. Οπότε εδώ έχουμε redistribution από OSPF1 area σε BGP 65001 autonomous system και redistribution από το BGP 65001 autonomous system στο OSPF1 area. Άρα ο R4 κάνει δύο αναδιανομές κάνει match τις internal διαδρομές με τις external.

```
router ospf 1
 redistribute connected subnets
 redistribute bgp 65001 subnets
 network 30.1.1.4 0.0.0.3 area 0
-----
router bgp 65001
 bgp redistribute-internal
 redistribute ospf 1 match internal external 1 external 2
 neighbor 200.200.200.1 remote-as 65001
```



```
neighbor 200.200.200.1 next-hop-self
neighbor 202.202.202.1 remote-as 65001
neighbor 202.202.202.1 next-hop-self
```

Προσοχή πρέπει να δοθεί σε αυτή τη γραμμή κώδικα **redistribute ospf 1 match internal external 1 external 2**, την γράφουμε διότι το κομμάτι του δικτύου OSPF1 area0 ο ISP Service Provider δηλαδή, έχει μάθει όπως είπαμε και στην εικόνα 6-189 δύο εξωτερικές διαδρομές διαμέσων του OSPF. Αν θέλουμε να τις διαφημίσουμε αυτές τις διαδρομές στην ΕΤΑΙΡΕΙΑ απαραίτητη προϋπόθεση για την πλήρη διασύνδεση του δικτύου είναι αυτή η γραμμή κώδικα που τρέχουμε στο R7 κάνοντας match τις internal external διαδρομές του OSPF1 area 0 με τις external διαδρομές της εταιρείας, που είπαμε ότι κάνει την αναδιανομή από τον ISP Service Provider στην ΕΤΑΙΡΕΙΑ και από την ΕΤΑΙΡΕΙΑ στον ISP Service Provider.

- Επόμενος σταθμός στην τοπολογία μας είναι ο δρομολογητής **R1** και αυτός θα κάνει αναδιανομή το δίκτυο με OSPF 10 με το BGP 65001.

```
router ospf 10
```

redistribute connected subnets/ με αυτή την εντολή θα εμφανιστούν μόνο τα άμεσα συνδεδεμένα δίκτυα με αυτό (directly connected) δεν χρειάζεται να τα βάλω manual. Δεν θα μάθει τα άλλα local interfaces που είναι συνδεδεμένα στον δρομολογητή μόνο αυτό που γίνεται η σύνδεση.*/**

```
redistribute bgp 65001 subnets
network 10.1.1.0 0.0.0.255 area 0
```

```
router bgp 65001
```

bgp redistribute-internal/ με την εντολή αυτή του λέμε ότι routes έμαθες προώθησε τα αυτό δεν γίνεται αυτόματα γιατί δεν περνούν τα routes από exterior (BGP) σε interior gateway protocols(OSPF), να τα προωθήσει δηλαδή πίσω στον ISP*/**

```
redistribute connected
redistribute ospf 10
neighbor 202.202.202.2 remote-as 65001
neighbor 202.202.202.2 next-hop-self
```



- Με την ακριβώς ίδια λογική με τον R1 τρέχω τον αντίστοιχο κώδικα στον R2 με διαφορετικό δίκτυο βέβαια, που λειτουργεί σαν backup του R1.

```
router ospf 10
```

*redistribute connected subnets/** με αυτή την εντολή θα εμφανιστούν μόνο τα άμεσα συνδεδεμένα δίκτυα με αυτό (*directly connected*) δεν χρειάζεται να τα βάλω *manual*. Δεν θα μάθει τα άλλα *local interfaces* που είναι συνδεδεμένα στον δρομολογητή μόνο αυτό που γίνεται η σύνδεση.*/

```
redistribute bgp 65001 subnets  
network 10.1.2.0 0.0.0.255 area 0
```

```
router bgp 65001
```

*bgp redistribute-internal/** με την εντολή αυτή του λέμε ότι *routes* έμαθες προώθησε τα αυτό δεν γίνεται αυτόματα γιατί δεν περνούν τα *routes* από *exterior (BGP)* σε *interior gateway protocols(OSPF)*, να τα προωθήσει δηλαδή πίσω στον *ISP**/

```
redistribute connected  
redistribute ospf 10  
neighbor 200.200.200.2 remote-as 65001  
neighbor 200.200.200.2 next-hop-self
```

- Πηγαίνοντας στον δρομολογητή R3 το μόνο που κάνουμε είναι να τρέξουμε το OSPF με id 10 μιας και εδώ δεν έχω αναδιανομή απλά διαφημίζω τις διαδρομές(δίκτυα) που είναι “συνδεδεμένα” στον R3 με το OSPF πρωτόκολλο

```
router ospf 10  
network 10.1.1.0 0.0.0.255 area 0  
network 10.1.2.0 0.0.0.255 area 0  
network 10.1.3.0 0.0.0.255 area 0
```

- Τέλος πηγαίνουμε στον R8 που έχει μόνο ένα δίκτυο και τρέχουμε και εκεί το OSPF

```
router ospf 10  
network 10.1.3.0 0.0.0.255 area 0
```



6.7.1 Επαλήθευση λειτουργίας του δικτύου

Πηγαίνουμε στον R8 τελευταίο δρομολογητή της τοπολογίας που ανήκει στην πλευρά της εταιρείας μας και τρέχουμε την εντολή `show ip route`

```
Gateway of last resort is not set

  200.200.200.0/30 is subnetted, 1 subnets
O E2   200.200.200.0 [110/20] via 10.1.3.1, 00:25:43, Serial0/0
  202.202.202.0/30 is subnetted, 1 subnets
O E2   202.202.202.0 [110/20] via 10.1.3.1, 00:25:43, Serial0/0
  66.0.0.0/30 is subnetted, 1 subnets
O E2   66.66.66.8 [110/1] via 10.1.3.1, 00:25:43, Serial0/0
  10.0.0.0/24 is subnetted, 3 subnets
C      10.1.3.0 is directly connected, Serial0/0
O      10.1.2.0 [110/74] via 10.1.3.1, 00:25:43, Serial0/0
O      10.1.1.0 [110/74] via 10.1.3.1, 00:25:45, Serial0/0
  11.0.0.0/30 is subnetted, 1 subnets
O E2   11.11.11.0 [110/1] via 10.1.3.1, 00:25:45, Serial0/0
  30.0.0.0/30 is subnetted, 2 subnets
O E2   30.1.1.4 [110/1] via 10.1.3.1, 00:25:46, Serial0/0
O E2   30.1.1.0 [110/1] via 10.1.3.1, 00:25:46, Serial0/0
R8#
```

Εικόνα 6-190 Routing table του R8

Από την εικόνα 6-190 βλέπουμε ότι έχει μάθει τα δίκτυα με το κόκκινο βέλος από το πρωτόκολλο OSPF για αυτό και η ένδειξη O. Τα δίκτυα με ένδειξη O E2 της εικόνας 6-191 τα έχει μάθει από τα external OSPF δηλαδή από το Redistribution των R1 και R2 BGP to OSPF.

```
Gateway of last resort is not set

  200.200.200.0/30 is subnetted, 1 subnets
O E2   200.200.200.0 [110/20] via 10.1.3.1, 00:28:40, Serial0/0
  202.202.202.0/30 is subnetted, 1 subnets
O E2   202.202.202.0 [110/20] via 10.1.3.1, 00:28:40, Serial0/0
  66.0.0.0/30 is subnetted, 1 subnets
O E2   66.66.66.8 [110/1] via 10.1.3.1, 00:28:40, Serial0/0
  10.0.0.0/24 is subnetted, 3 subnets
C      10.1.3.0 is directly connected, Serial0/0
O      10.1.2.0 [110/74] via 10.1.3.1, 00:28:40, Serial0/0
O      10.1.1.0 [110/74] via 10.1.3.1, 00:28:41, Serial0/0
  11.0.0.0/30 is subnetted, 1 subnets
O E2   11.11.11.0 [110/1] via 10.1.3.1, 00:28:41, Serial0/0
  30.0.0.0/30 is subnetted, 2 subnets
O E2   30.1.1.4 [110/1] via 10.1.3.1, 00:28:42, Serial0/0
O E2   30.1.1.0 [110/1] via 10.1.3.1, 00:28:42, Serial0/0
R8#
```

Εικόνα 6-191 Routing table του R8

Και αυτή η ένδειξη έχει μάθει από το redistribution που ξεκινά από τον `R6->R5->R7` συγκεντρώνει όλα αυτά τα δίκτυα και τα κάνει redistribute από το δίκτυο OSPF στο επόμενο BGP με AS 65001 `R4->R1->R3->R8` ή εναλλακτικά `R4->R2->R3->R8`

```
  66.0.0.0/30 is subnetted, 1 subnets
O E2   66.66.66.8 [110/1] via 10.1.3.1, 00:28:40, Serial0/0
  10.0.0.0/24 is subnetted, 3 subnets
C      10.1.3.0 is directly connected, Serial0/0
O      10.1.2.0 [110/74] via 10.1.3.1, 00:28:40, Serial0/0
O      10.1.1.0 [110/74] via 10.1.3.1, 00:28:41, Serial0/0
  11.0.0.0/30 is subnetted, 1 subnets
O E2   11.11.11.0 [110/1] via 10.1.3.1, 00:28:41, Serial0/0
  30.0.0.0/30 is subnetted, 2 subnets
O E2   30.1.1.4 [110/1] via 10.1.3.1, 00:28:42, Serial0/0
O E2   30.1.1.0 [110/1] via 10.1.3.1, 00:28:42, Serial0/0
R8#
```

Εικόνα 6-192 Routing table του R8



Ping από τον R8 προς τον R9

```
R8#ping 11.11.11.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.11.11.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/88/104 ms
R8#
```

Εικόνα 6-193 Επιτυχής επικοινωνία του R8 προς τον R9

Traceroute για εύρεση της διαδρομής R8-R9

```
R8#traceroute 11.11.11.1
Type escape sequence to abort.
Tracing the route to 11.11.11.1

 0 10.1.3.1 28 msec 24 msec 24 msec
 1 10.1.2.1 48 msec 48 msec 48 msec
 2 200.200.200.2 52 msec 48 msec 48 msec
 3 30.1.1.6 92 msec 72 msec 68 msec
 4 30.1.1.1 100 msec 92 msec 72 msec
 5 66.66.66.9 68 msec 88 msec 68 msec
 6 11.11.11.1 92 msec 96 msec 88 msec
R8#
```

Εικόνα 6-194 Διαδρομή που ακολούθησε το πακέτο διαμέσων του R2

Επιλέγει το 10.1.2.1 προς τον δρομολογητή R2 και μετά 200.200.200.2 του ser0/3 του R2.

Στη συνέχεια πηγαίνουμε κλείνουμε την πόρτα ser0/3 του R2 και βλέπουμε ότι ακολουθεί την εναλλακτική διαδρομή που προσφέρει ο R1 κάνοντας ξανά traceroute.

```
R2#CONF T
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#INT SER0/3
R2(config-if)#SHUT
R2(config-if)#
*Mar 1 01:14:03.095: %LINK-5-CHANGED: Interface Serial0/3, changed state to administratively down
*Mar 1 01:14:04.095: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/3, changed state to down
R2(config-if)#
```



```
Tracing the route to 11.11.11.1

  1 10.1.3.1 28 msec 24 msec 28 msec
  2 10.1.1.1 32 msec 28 msec 28 msec
  3 202.202.202.2 36 msec 32 msec 28 msec
  4 30.1.1.6 56 msec 64 msec 60 msec
  5 30.1.1.1 76 msec 76 msec 72 msec
  6 66.66.66.9 72 msec 72 msec 72 msec
  7 11.11.11.1 92 msec 96 msec 96 msec
R8#
```

Εικόνα 6-195 Διαδρομή που ακολούθησε το πακέτο διαμέσον του R1 κλείνοντας την πόρτα του R2

Ping από τον R9 προς τον R8

```
R9#ping 10.1.3.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/76/96 ms
R9#
```

Εικόνα 6-196 Επιτυχής μετάδοση πακέτου από τον R9-R8

Πηγαίνοντας στην πλευρά που έχουμε το ΑΠΟΜΑΚΡΥΣΜΕΝΟ SITE R9 και R6 routers βλέπουμε το παρακάτω

```
R9#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

R#
11.0.0/30 is subnetted, 1 subnets
C    11.11.11.0 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 is directly connected, FastEthernet0/1
R9#

R6#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

R6#
66.0.0/30 is subnetted, 1 subnets
C    66.66.66.8 is directly connected, Serial10/4
C    11.0.0/30 is subnetted, 1 subnets
C    11.11.11.0 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 is directly connected, Serial10/4
R6#
```

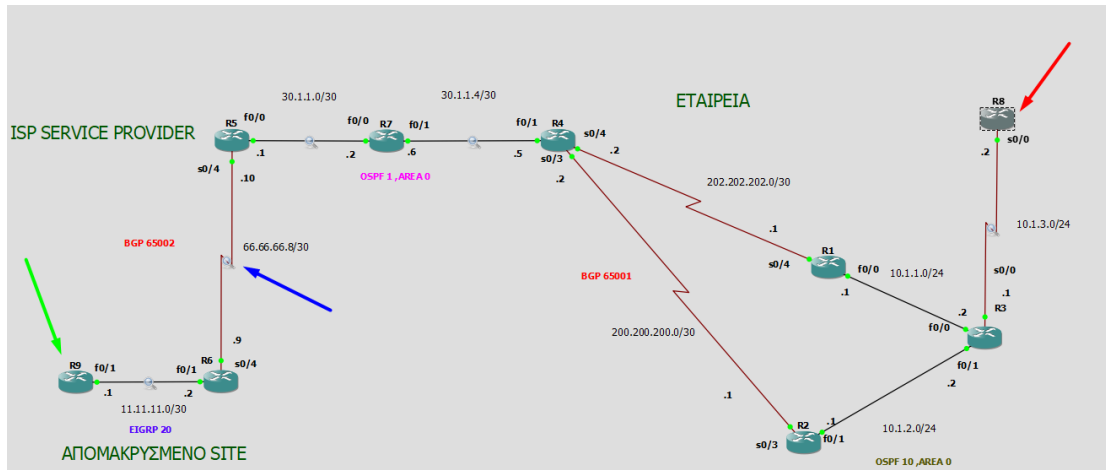
Εικόνα 6-197 Routing tables του ΑΠΟΜΑΚΡΥΣΜΕΝΟΥ SITE

Από την εικόνα 6-197 βλέπουμε ότι ο δρομολογητής R9 γνωρίζει μόνο τον γειτονικό του δρομολογητή R6 και ο R6 τον γείτονα του R9 και μόνο τον γείτονα R5 που έχουμε τρέξει το BGP άρα ο R6 στέλνει στον R5 πακέτα και μόνο αυτός γνωρίζει τις διαδρομές όχι ο R6. Ο R6 γνωρίζει τον αμέσως επόμενο γείτονα του R5 και τίποτα άλλο extra.



6.7.2 Έλεγχος της κυκλοφορίας με το Wireshark

Αυτό που θα κάνουμε τώρα είναι να παρατηρήσουμε την σύνδεση μεταξύ του R5-R6 δρομολογητή. Θα στείλουμε ένα μήνυμα 5 πακέτων από τον R8->R9 δρομολογητή και θα αρχίσουμε το Capture.



Εικόνα 6-198 Capture της ser0/4 & αποστολή πακέτου R8-R9

Έχοντας κάνει capture το link στέλνουμε τα 5 ICMP πακέτα και μεταφερόμαστε στο Wireshark.

Capturing from Standard input [R5 Serial0/4 to R6 Serial0/4]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl+>

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|-------------|-------------|----------|--------|--|
| 1 | 0.000000 | 66.66.66.10 | 66.66.66.9 | BGP | 63 | KEEPALIVE Message |
| 2 | 0.001978 | N/A | N/A | SLARP | 24 | Line keepalive, outgoing sequence 23, returned sequence 21 |
| 3 | 0.325368 | 66.66.66.9 | 66.66.66.10 | TCP | 44 | 29720 → 179 [ACK] Seq=1 Ack=20 Win=16225 Len=0 |
| 4 | 9.147200 | 10.1.3.2 | 11.11.11.1 | ICMP | 104 | Echo (ping) request id=0x0004, seq=0/0, ttl=250 (reply in 5) |
| 5 | 9.177455 | 11.11.11.1 | 10.1.3.2 | ICMP | 104 | Echo (ping) reply id=0x0004, seq=0/0, ttl=254 (request in 4) |
| 6 | 9.269711 | 10.1.3.2 | 11.11.11.1 | ICMP | 104 | Echo (ping) request id=0x0004, seq=1/256, ttl=250 (reply in 7) |
| 7 | 9.300468 | 11.11.11.1 | 10.1.3.2 | ICMP | 104 | Echo (ping) reply id=0x0004, seq=1/256, ttl=254 (request in 6) |
| 8 | 9.393218 | 10.1.3.2 | 11.11.11.1 | ICMP | 104 | Echo (ping) request id=0x0004, seq=2/512, ttl=250 (reply in 9) |
| 9 | 9.423971 | 11.11.11.1 | 10.1.3.2 | ICMP | 104 | Echo (ping) reply id=0x0004, seq=2/512, ttl=254 (request in 8) |
| 10 | 9.522100 | 10.1.3.2 | 11.11.11.1 | ICMP | 104 | Echo (ping) request id=0x0004, seq=3/768, ttl=250 (reply in 11) |
| 11 | 9.562776 | 11.11.11.1 | 10.1.3.2 | ICMP | 104 | Echo (ping) reply id=0x0004, seq=3/768, ttl=254 (request in 10) |
| 12 | 9.649076 | 10.1.3.2 | 11.11.11.1 | ICMP | 104 | Echo (ping) request id=0x0004, seq=4/1024, ttl=250 (reply in 13) |

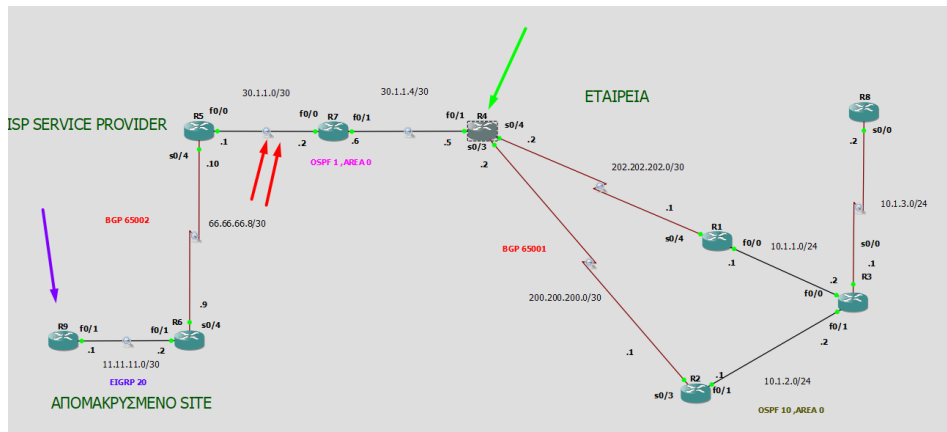
Εικόνα 6-199 Πληροφορίες της ser0/4

Στην εικόνα 6-199 βλέπουμε τα keepralive μηνύματα που μεταδίδονται από τον R5 προς τον R6 όπως ορίζει το BGP πρωτόκολλο ανά 60 sec. Επίσης αυτό που βλέπουμε είναι το BGP session που όπως αναφέρθηκε και στο θεωρητικό κομμάτι είναι ένα session που κάνει χρήση TCP πρωτοκόλλου ανάμεσα από BGP neighbors που εναλλάσσονται δεδομένα δρομολόγησης με τη χρήση του BGP στην πόρτα 179. Οι γείτονες διαχειρίζονται το state του session με την αποστολή keepralive μηνυμάτων ανά 30 sec, αυτό το βλέπουμε μεταξύ του R6(66.66.66.9) και του R5(66.66.66.10). Ο Αριθμός Επιβεβαίωσης (Acknowledgment-ACK), χρησιμοποιείται για να εξασφαλιστεί ότι κάθε τμήμα έχει φτάσει στον προορισμό του. Όταν ο παραλήπτης στο άλλο άκρο λάβει το τμήμα στέλνει ένα νέο τμήμα (ACK- επιβεβαίωσης) του οποίου το πεδίο Αριθμός επιβεβαίωσης, είναι συμπληρωμένο. Όπως βλέπουμε στην εικόνα το session έχει εγκατασταθεί



επιτυχώς. Στην συνέχεια μεταδίδοντα ICMP πακέτα χρησιμοποιώντας πρώτα το Echo request από τον 10.1.3.2 (R8) προς τον 11.11.11.1(R9) και αυτός απαντά με Echo reply.

Μέσω του Wireshark μπορούμε να δούμε και τα άλλα πακέτα που μεταδίδονται και στις άλλες ζεύξεις στην τοπολογία μας στέλνοντας ένα μήνυμα από τον R4->R9 παρακολουθώντας την f0/0 σύνδεση όπως φαίνεται παρακάτω



Εικόνα 6-200 Capture της F0/0 & αποστολή πακέτων R4-R9

Capturing from Standard input [R5 FastEthernet0/0 to R7 FastEthernet0/0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|------------|-------------|----------|--------|--|
| 501 | 2029.818120 | 30.1.1.1 | 224.0.0.5 | OSPF | 94 | Hello Packet |
| 502 | 2031.463924 | 30.1.1.2 | 224.0.0.5 | OSPF | 94 | Hello Packet |
| 503 | 2035.288289 | 30.1.1.5 | 11.11.11.1 | ICMP | 114 | Echo (ping) request id=0x0002, seq=0/0, ttl=254 (no response found!) |
| 504 | 2039.004424 | 30.1.1.5 | 11.11.11.1 | ICMP | 114 | Echo (ping) request id=0x0002, seq=1/256, ttl=254 (reply in 505) |
| 505 | 2039.050223 | 11.11.11.1 | 30.1.1.5 | ICMP | 114 | Echo (ping) reply id=0x0002, seq=1/256, ttl=253 (request in 504) |
| 506 | 2039.096682 | 30.1.1.5 | 11.11.11.1 | ICMP | 114 | Echo (ping) request id=0x0002, seq=2/512, ttl=254 (reply in 507) |
| 507 | 2039.142470 | 11.11.11.1 | 30.1.1.5 | ICMP | 114 | Echo (ping) reply id=0x0002, seq=2/512, ttl=253 (request in 506) |
| 508 | 2039.188936 | 30.1.1.5 | 11.11.11.1 | ICMP | 114 | Echo (ping) request id=0x0002, seq=3/768, ttl=254 (reply in 509) |
| 509 | 2039.234614 | 11.11.11.1 | 30.1.1.5 | ICMP | 114 | Echo (ping) reply id=0x0002, seq=3/768, ttl=253 (request in 508) |
| 510 | 2039.281192 | 30.1.1.5 | 11.11.11.1 | ICMP | 114 | Echo (ping) request id=0x0002, seq=4/1024, ttl=254 (reply in 511) |
| 511 | 2039.326964 | 11.11.11.1 | 30.1.1.5 | ICMP | 114 | Echo (ping) reply id=0x0002, seq=4/1024, ttl=253 (request in 510) |

Εικόνα 6-201 Πληροφορίες της F0/0

Εδώ βλέπουμε τα Hello πακέτα που στέλνει το OSPF από τον R5&R7 που τρέχουν το OSPF και στην συνέχεια την ανταλλαγή πακέτων μεταξύ των R4 και R9.[48][49]



7 Μελέτη του Εξομοιωτή Mininet

Στο κεφάλαιο αυτό θα δώσουμε πλήρη περιγραφή του εξομοιωτή και ορισμένων σεναρίων που τρέξαμε σε περιβάλλον Linux όπου εκεί υπάρχει γραφικό περιβάλλον για το Mininet, ενώ στα Windows δυστυχώς δεν υπάρχει. Αυτό που θα κάνουμε στα Windows για να πετύχουμε τη διασύνδεση με το GNS3 όπου θα αναφερθεί στο κεφάλαιο 9 είναι να κατεβάσουμε μία precompiled εικονική μηχανή που τρέχει το Mininet και από εκεί θα δουλεύουμε τις εντολές μας.

7.1 Περιγραφή του Mininet

Το Mininet όπως αναφέραμε και στο κεφάλαιο 6 είναι ένα πρόγραμμα εικονικοποίησης δικτύων που έχει υλοποιηθεί σε Python και τρέχει σε λειτουργικό Linux. Με το Mininet μπορούμε να εξομοιώσουμε πολύ μεγάλες δικτυακές τοπολογίες εξαιτίας της αρχιτεκτονικής που χρησιμοποιεί, δηλαδή της ελαφριάς εικονικοποίησης. Ο χρήστης του Mininet χρησιμοποιεί εντολές σε python για να εκκινήσει τις τοπολογίες του οι οποίες όπως προαναφέραμε μπορεί να είναι εκατοντάδες switches τα οποία είναι αδύνατο να τρέξουν σε ένα συμβατικό υπολογιστή. Χρησιμοποιεί τον πυρήνα του Linux και τον κατακερματίζει σε πολλά μικρά εικονικά μηχανήματα τα οποία μοιράζονται αυτόν τον ενιαίο πυρήνα και χρησιμοποιούν κοινά αρχεία έχοντας πολύ απλές λειτουργίες των Unix systems πράγμα που τα κάνει εύχρηστα και “ελαφριά”. Με πιο απλά λόγια εξαιτίας του μικρού χώρου που καταλαμβάνουν τρέχουν σε μεγάλες ταχύτητες και είναι πολύ πρακτικά για πολλαπλές δοκιμές και σε τοπολογίες τεράστιες σε όγκο. Το Mininet έχει τη δυνατότητα δημιουργίας πραγματικών εικονικών δικτύων μέσα από τις εντολές που τρέχουμε στην κεντρική εικονική μηχανή. Όπως θα δούμε και παρακάτω δημιουργεί τοπολογίες με hosts, switches και routers. Οι routers λειτουργούν σαν hosts άλλα έχουν και την λειτουργία ip_forwarding.

Επίσης το Mininet είναι κατά κάποιο τρόπο συνδεδεμένο με το πρωτόκολλο Openflow καθώς οι μεταγωγείς του Mininet το χρησιμοποιούν στο οποίο θα αναφερθούμε επιγραμματικά διότι έχει ένα μεγάλο εύρος παραμέτρων που πρέπει να αναφέρουμε που δεν θα τα προσεγγίσουμε στην εν λόγω μελέτη. Πιο συγκεκριμένα το πρωτόκολλο Openflow δημιουργήθηκε για να μπορούμε να προγραμματίζουμε τους πίνακες ροής των switches και routers προκειμένου να εκτελούν λειτουργίες (π.χ. διαφορετική διαδρομή για να φτάσει τον πακέτο στον προορισμό του) ανάλογα με την επιθυμία του εκάστοτε διαχειριστή του συστήματος, χωρίς να επηρεάζεται βέβαια η συνολική λειτουργία του υπόλοιπου δικτύου. Το πρωτόκολλο OpenFlow συγκροτεί ένα πρωτόκολλο δικτύου που μπορούμε να παραμετροποιήσουμε και παρήχθη με επιδίωξη την διαχείριση και την κατεύθυνση της κίνησης των δεδομένων μεταξύ των δρομολογητών (router), των μεταγωγέων (switch) και των επαναληπτών (hub). Κάθε Openflow μεταγωγέας που έχουμε στο Mininet συνδέεται με ένα remote (απομακρυσμένο) controller ο οποίος είτε μπορεί να είναι στον ίδιο πυρήνα Linux με το Mininet η να τρέχει σε διαφορετική εικονική μηχανή η οποία έχει μία IP διεύθυνση και ακούει σε ένα server στο διαδίκτυο, όπου εμείς μπορούμε να συνδεόμαστε και να παρακολουθούμε την κίνηση του δικτύου στο Mininet. Παρακάτω θα μελετήσουμε τον τρόπο υλοποίησης της παραπάνω πρότασης. Ο Openflow μεταγωγέας ακούγοντας τον controller που έχουμε ορίσει μπορεί να προωθεί συγκεκριμένα πακέτα που αυτός επιθυμεί ή να απορρίψει κάποια πακέτα για λόγους ασφαλείας, ακούγοντας πάντα τον ελεγκτή.[44]

7.2 Εγκατάσταση του Mininet

Το Mininet κατά βάση τρέχει στο λειτουργικό Linux όπου εκεί χρησιμοποιώντας το πρόγραμμα Miniedit μπορούμε να έχουμε οπτικοποίηση των τοπολογιών που τρέχουμε. Ακόμη



μπορούμε με drag and drop να προσθέσουμε στην παλέτα μας switches και hosts. Στο λειτουργικό Windows όπως είπαμε και στην εισαγωγή κατεβάζουμε ένα προκατασκευασμένο VM που είναι συμπεριεμένο σε ένα OVA file. Αυτό θα μελετηθεί στη συνέχεια.

Πειραματιστήκαμε στο Mininet και στα δύο λογισμικά. Για να καταλάβουμε πλήρως την λειτουργία του πρέπει να το εγκαταστήσουμε και βήμα-βήμα θα εξηγήσουμε τα χαρακτηριστικά του και τις δυνατότητές του. Για να εγκαταστήσουμε το Mininet μιας και χρησιμοποιούμε λειτουργικό Windows θα δημιουργήσουμε ένα εικονικό μηχάνημα διαμέσων του Virtual Box όπου θα εγκαταστήσουμε τα Ubuntu και μετά θα κατεβάσουμε από το github το precompiled image του Mininet που περιέχει το miniedit ένα tool που μας δίνει μια μορφή γραφικού για να έχουμε και οπτικά αποτελέσματα.

- Install του git για να μπορούμε να τραβάμε και να κλωνοποιήσουμε το Mininet από το repository του Github

```
sudo apt install linux-headers-$(uname -r) build-essential dkms
```

```
sudo apt get install git
```

```
git clone git://github.com/mininet/mininet
```

```
cd mininet
```

```
git tag git checkout -b 2.2.1 2.2.1
```

```
cd
```

*mininet/util/install.sh -a /*Ετσι κατεβάζουμε όλα τα αρχεία μαζί και όλα τα διαθέσιμα scripts που υπάρχουν για αυτό όπως το miniedit και το packet sniffer wireshark */*

Έχοντας τελειώσει με αυτές τις εντολές πηγαίνουμε στο λειτουργικό μας και τρέχουμε την πιο απλή εντολή του Mininet

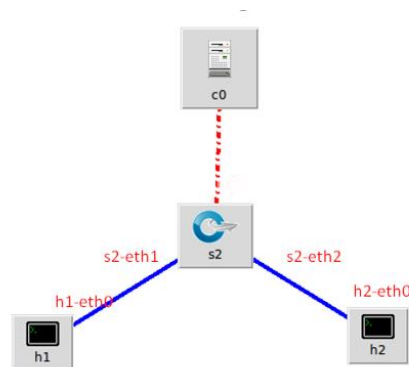
- *sudo mn* Με την εντολή αυτή δημιουργούμε μια απλή τοπολογία που αποτελείται από δύο hosts h1,h2 ένα μεταγωγέα s1 και το ελεγκτή controller. Στη συνέχεια γράφουμε την εντολή *pingall* και τεστάρουμε την επικοινωνία κάνει αυτόματα ping σε όλη την τοπολογία και γενικά ότι το mininet δουλεύει.



```
cg@cg-VirtualBox:~$ sudo mn
[sudo] password for cg:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

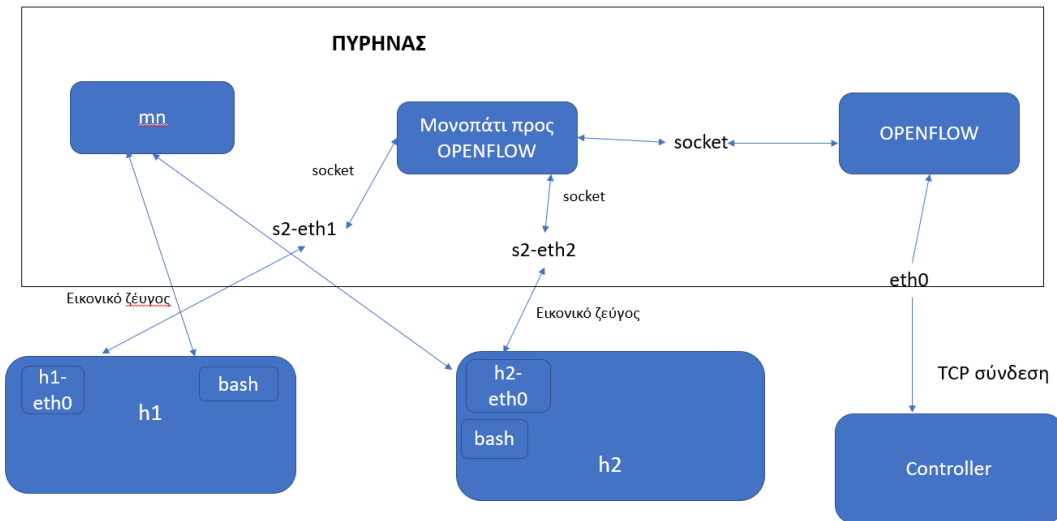
Εικόνα 7-1 Επιτυχής επικοινωνία στο περιβάλλον του Mininet

Η τοπολογία που δημιουργήθηκε με αυτό το script που τρέξαμε είναι της μορφής αυτής



Εικόνα 7-2 Μορφή τοπολογίας που δημιουργεί το mininet

Ουσιαστικά αυτό που γίνεται με την εντολή mn είναι ότι δημιουργεί δύο bash(κέλυφος) ένα για τον host1 και ένα για τον host 2 που υπάρχουν μέσα στον ίδιο πυρήνα και υπακούν σε αυτόν. Έπειτα κατασκευάζει δύο ζευγάρια από εικονικά interfaces το ένα είναι το *s2-eth1—h1-eth0* και το άλλο *s2-eth2—h2-eth0*. Σκεφτόμαστε ότι έχουμε 3 ‘χώρους εργασίας’ έναν το κεντρικό που βρίσκεται το s2 έναν του host1 και ένα του host2. Έπειτα τα s2-eth1 και s2-eth2 διασυνδέονται μέσω sockets σε ένα μονοπάτι που αυτό συνδέεται με ένα άλλο socket με το openflow πρωτόκολλο. Η παραπάνω παράγραφος θα εκφραστεί και με ένα σχήμα για την καλύτερη επεξήγηση της.[51]



Εικόνα 7-3 Αρχιτεκτονική του Mininet

Όπως βλέπουμε το Mininet έχει το δικό του τερματικό και μπορεί από κει και δίνουμε εντολές. Επίσης μπορούμε να τρέξουμε εντολές σε κάθε τερματικό ξεχωριστά (π.χ. ping σε άλλον host.) Αυτό το επιτυγχάνουμε με τον emulator xterm που κατεβαίνει μαζί με το mininet και ανοίγει ξεχωριστά παράθυρα εκτέλεσης εντολών (CLI) για κάθε host γράφοντας την εντολή xterm h1 h2.

```

[~] cp@cg-VirtualBox: ~
[sudo] password for cg:
Sorry, try again.
[sudo] password for cg:
Sorry, try again.
[sudo] password for cg:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1...
*** Starting CLI:
mininet> xterm h1 h2
mininet> xterm h1 h2
mininet>
  
```

Εικόνα 7-4 Δημιουργία ξεχωριστών CLI για κάθε host

Ορισμένες εντολές που χρησιμοποιούμε στο περιβάλλον του Mininet είναι οι παρακάτω:

- **help**, όπου μας δείχνει τις δυνατότητες ή τις επιλογές που έχουμε στο περιβάλλον του εξομοιωτή

```

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

Εικόνα 7-5 Διαθέσιμες επιλογές στο mininet

- **links**, όπου βλέπουμε τις διαθέσιμες συνδέσεις στην εκάστοτε τοπολογία που τρέχουμε



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

Εικόνα 7-6 Links που φαίνονται στην τοπολογία mn

- `dump`, αναλυτικές πληροφορίες για την εκάστοτε τοπολογία[50]

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=8971>
<Host h2: h2-eth0:10.0.0.2 pid=8973>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=8978>
<Controller c0: 127.0.0.1:6653 pid=8964>
mininet>
```

Εικόνα 7-7 Αναλυτικές πληροφορίες τοπολογίας mn

Επίσης γράφοντας στο περιβάλλον κάποιου host έχοντας ανοίξει πάντα ένα xterm τερματικό την εντολή `ifconfig` βλέπουμε τα διαθέσιμα network interfaces. Βλέπουμε στο interface h1-eth0 του h1 που με αυτό έχει εικονικό ζεύγος με το switch να ανατίθεται by default IP διεύθυνση δικτύου 10.0.0.0 που υποδεικνύει ότι είναι virtual network.

```
root@ecg-VirtualBox:~# ifconfig
h1-eth0  Link encap:Ethernet  HWaddr 8a:b6:66:e0:bd:dd
         inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
         inet6 addr: fe80::88b6:66ff:fee0:bd:dd/64  Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:51 errors:0 dropped:0 overruns:0 frame:0
         TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:6389 (6.3 KB)  TX bytes:1216 (1.2 KB)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128  Scope:Host
         UP LOOPBACK RUNNING  MTU:65536  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Εικόνα 7-8 ifconfig στο περιβάλλον του h1

Ακόμη έχοντας τα host τερματικά IP τρέχοντας την default τοπολογία mn με την εντολή `h1 ping -c 3 h2(10.0.0.2)` στέλνουμε 3 πακέτα ICMP προς τον h2



```
cg@cg-VirtualBox:~$ sudo mininet
*** Shutting down stale tunnels
bkill -9 -f mininet:
*** Creating network
bkill -9 -f Tunnel=Ethernet
bkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
cg@cg-VirtualBox:~$ sudo mininet
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> xterm h1 h2
mininet>

"Node: h2"
root@cg-VirtualBox:~# ifconfig
h2-eth0  Link encap:Ethernet  HWaddr ea:6a:29:52:60:b6
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::e9a:129f:1422:4906:64 Scope:link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:28  errors:0  dropped:0  overruns:0  frame:0
          TX packets:19  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:3715 (3.7 KB)  TX bytes:726 (0.8 B)

          Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0  errors:0  dropped:0  overruns:0  frame:0
          TX packets:0  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@cg-VirtualBox:~#

"Node: h1"
root@cg-VirtualBox:~# ifconfig
h1-eth0  Link encap:Ethernet  HWaddr ea:6a:29:52:60:b6
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::e9a:129f:1422:4906:64 Scope:link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:28  errors:0  dropped:0  overruns:0  frame:0
          TX packets:19  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:3715 (3.7 KB)  TX bytes:726 (0.8 B)

          Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0  errors:0  dropped:0  overruns:0  frame:0
          TX packets:0  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@cg-VirtualBox:~# ping -c 3 h2
ping: unknown host h2
root@cg-VirtualBox:~# ping -c 3 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=18.4 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.457 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.050 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2018ms
rtt min/avg/max/mdev = 0.060/6.534/15.486/6.534 ms
root@cg-VirtualBox:~#
```

Εικόνα 7-9 ping από τον h1 προς τον h2

Το Mininet όπως αναφέραμε και πιο πάνω είναι γραμμένο στη προγραμματιστική γλώσσα Python τρέχοντας όλες τις τοπολογίες που θα μελετήσουμε σε scripts. Πιο κάτω θα αναφέρουμε ορισμένες βασικές εντολές που χρησιμοποιήσαμε προκειμένου να παραμετροποιήσουμε κάποια ήδη υπάρχοντα scripts για τις δικά μας πειράματα καθώς και ορισμένες που χτίσαμε από την αρχή μόνοι μας. Παρακάτω βλέπουμε ένα default script του εξομοιωτή που θα εξηγήσουμε ορισμένες βασικές εντολές για την δημιουργία της εκάστοτε τοπολογίας. Στην εικόνα 7-10 βλέπουμε ένα script που δημιουργεί δύο switches με ένα τερματικό το κάθε switch.

```
from mininet.topo import Topo

class MyTopo( Topo ):

    def __init__( self ):

        Topo.__init__( self )

        leftHost = self.addHost( 'h1' )
        rightHost = self.addHost( 'h2' )
        leftSwitch = self.addSwitch( 's3' )
        rightSwitch = self.addSwitch( 's4' )

        self.addLink( leftHost, leftSwitch )
        self.addLink( leftSwitch, rightSwitch )
        self.addLink( rightSwitch, rightHost )

topos = { 'mytopo': ( lambda: MyTopo() ) }
```

Εικόνα 7-10 Python Script δυο switch με ένα host το καθένα

- class MyTopo(Topo): η κλάση της τοπολογίας και την ονομάζουμε Topo
- def __init__(self): Δημιουργία της δικής μας τοπολογίας
- Topo.__init__(self): Αρχικοποιούμε την τοπολογία που έχουμε

/*εδώ προσθέτουμε η αφαιρούμε hosts και switches*/



```
leftHost = self.addHost( 'h1' )
rightHost = self.addHost( 'h2' )
leftSwitch = self.addSwitch( 's3' )
rightSwitch = self.addSwitch( 's4' )
```

```
/*Εδώ προσθέτουμε τις συνδέσεις μεταξύ των hosts και των switches*/
```

```
self.addLink( leftHost, leftSwitch )
self.addLink( leftSwitch, rightSwitch )
self.addLink( rightSwitch, rightHost )
```

Το Mininet όπως είπαμε και πιο πάνω δίνει τη δυνατότητα δημιουργίας μεγάλων τοπολογιών που υλοποιούνται σε μερικά μόλις λεπτά, τρέχοντας πάντα ένα python script. Πιο κάτω τρέχουμε ένα script που υπάρχει στο φάκελο mininet/examples με όνομα tree1024.py που βλέπουμε ότι δημιουργεί 1024 hosts και 33 μεταγωγείς ένα controller, όπου κάθε μεταγωγέας συνδέεται με κάθε μεταγωγέα και κάθε host. Όπου η δημιουργία του διήρκεσε λιγότερο από 3 λεπτά.

```
cg@cg-VirtualBox:~$ cd mininet/examples
cg@cg-VirtualBox:~/mininet/examples$ sudo python tree1024.py
[sudo] password for cg:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h
43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h
63 h64 h65 h66 h67 h68 h69 h70 h71 h72 h73 h74 h75 h76 h77 h78 h79 h80 h81 h82 h
83 h84 h85 h86 h87 h88 h89 h90 h91 h92 h93 h94 h95 h96 h97 h98 h99 h100 h101 h10
2 h103 h104 h105 h106 h107 h108 h109 h110 h111 h112 h113 h114 h115 h116 h117 h11
```

```
cg@cg-VirtualBox: ~/mininet/examples
4 h775 h776 h777 h778 h779 h780 h781 h782 h783 h784 h785 h786 h787 h788 h789 h79
9 h791 h792 h793 h794 h795 h796 h797 h798 h799 h800 h801 h802 h803 h804 h805 h80
5 h807 h808 h809 h810 h811 h812 h813 h814 h815 h816 h817 h818 h819 h820 h821 h82
2 h823 h824 h825 h826 h827 h828 h829 h830 h831 h832 h833 h834 h835 h836 h837 h83
8 h839 h840 h841 h842 h843 h844 h845 h846 h847 h848 h849 h850 h851 h852 h853 h85
4 h855 h856 h857 h858 h859 h860 h861 h862 h863 h864 h865 h866 h867 h868 h869 h87
0 h871 h872 h873 h874 h875 h876 h877 h878 h879 h880 h881 h882 h883 h884 h885 h88
5 h887 h888 h889 h890 h891 h892 h893 h894 h895 h896 h897 h898 h899 h900 h901 h90
2 h903 h904 h905 h906 h907 h908 h909 h910 h911 h912 h913 h914 h915 h916 h917 h91
8 h919 h920 h921 h922 h923 h924 h925 h926 h927 h928 h929 h930 h931 h932 h933 h93
4 h935 h936 h937 h938 h939 h940 h941 h942 h943 h944 h945 h946 h947 h948 h949 h95
0 h951 h952 h953 h954 h955 h956 h957 h958 h959 h960 h961 h962 h963 h964 h965 h96
5 h967 h968 h969 h970 h971 h972 h973 h974 h975 h976 h977 h978 h979 h980 h981 h98
2 h983 h984 h985 h986 h987 h988 h989 h990 h991 h992 h993 h994 h995 h996 h997 h99
8 h999 h1000 h1001 h1002 h1003 h1004 h1005 h1006 h1007 h1008 h1009 h1010 h1011 h
1012 h1013 h1014 h1015 h1016 h1017 h1018 h1019 h1020 h1021 h1022 h1023 h1024
*** Starting controller
c0
*** Starting 33 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15 s16 s17 s18 s19 s20 s21 s22 s
23 s24 s25 s26 s27 s28 s29 s30 s31 s32 s33 ...
*** Running test
*** Starting CLI:
mininet>
```

Εικόνα 7-11 Δημιουργία τοπολογίας με 1024 host

Έχοντας δημιουργήσει αυτή την τοπολογία τρέχουμε την εντολή `pingall` και γίνεται ping από τον ένα host σε όλους τους υπόλοιπους, για να ελέγξουμε ότι το δίκτυο μας δουλεύει σωστά και οι hosts επικοινωνούν μεταξύ τους. Αυτό βέβαια χρειάζεται αρκετή ώρα για να επιβεβαιωθεί αν σκεφτεί κανείς τον όγκο των hosts σε συνδυασμό με όλα τα pings που πρέπει να εκτελέσει ο κάθε host.



Οπότε βλέπουμε μια άλλη τοπολογία αρκετά μικρότερη σε όγκο hosts που μπορούμε να δημιουργήσουμε με την παρακάτω εντολή `sudo mn --topo=linear,50` όπου δημιουργεί 50 hosts και 50 openflow switches,κάθε host συνδέεται με ένα switch και τα switch μεταξύ τους σε σειρά δημιουργώντας ουσιαστικά μία *bus topology* μέσα σε μερικά δευτερόλεπτα.

```
cg@cg-VirtualBox:~$ sudo mn --topo=linear,50
[sudo] password for cg:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h
43 h44 h45 h46 h47 h48 h49 h50
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15 s16 s17 s18 s19 s20 s21 s22 s
23 s24 s25 s26 s27 s28 s29 s30 s31 s32 s33 s34 s35 s36 s37 s38 s39 s40 s41 s42 s
43 s44 s45 s46 s47 s48 s49 s50
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (h5, s5) (h6, s6) (h7, s7) (h8, s8) (h9, s9)
(h10, s10) (h11, s11) (h12, s12) (h13, s13) (h14, s14) (h15, s15) (h16, s16) (h
17, s17) (h18, s18) (h19, s19) (h20, s20) (h21, s21) (h22, s22) (h23, s23) (h24,
s24) (h25, s25) (h26, s26) (h27, s27) (h28, s28) (h29, s29) (h30, s30) (h31, s3
1) (h32, s32) (h33, s33) (h34, s34) (h35, s35) (h36, s36) (h37, s37) (h38, s38)
(h39, s39) (h40, s40) (h41, s41) (h42, s42) (h43, s43) (h44, s44) (h45, s45) (h4
6, s46) (h47, s47) (h48, s48) (h49, s49) (h50, s50) (s2, s1) (s3, s2) (s4, s3) (
s5, s4) (s6, s5) (s7, s6) (s8, s7) (s9, s8) (s10, s9) (s11, s10) (s12, s11) (s13
, s12) (s14, s13) (s15, s14) (s16, s15) (s17, s16) (s18, s17) (s19, s18) (s20, s
19) (s21, s20) (s22, s21) (s23, s22) (s24, s23) (s25, s24) (s26, s25) (s27, s26)
(s28, s27) (s29, s28) (s30, s29) (s31, s30) (s32, s31) (s33, s32) (s34, s33) (s
35, s34) (s36, s35) (s37, s36) (s38, s37) (s39, s38) (s40, s39) (s41, s40) (s42,
s41) (s43, s42) (s44, s43) (s45, s44) (s46, s45) (s47, s46) (s48, s47) (s49, s4
8) (s50, s49)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h
43 h44 h45 h46 h47 h48 h49 h50
*** Starting controller
c0
*** Starting 50 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15 s16 s17 s18 s19 s20 s21 s22 s
23 s24 s25 s26 s27 s28 s29 s30 s31 s32 s33 s34 s35 s36 s37 s38 s39 s40 s41 s42 s
43 s44 s45 s46 s47 s48 s49 s50 ...
*** Starting CLI:
mininet>
```

Εικόνα 7-12 Δημιουργία τοπολογίας bus

Με την εντολή `dump` βλέπουμε πληροφορίες σύνδεσης του κάθε host

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=11208>
<Host h2: h2-eth0:10.0.0.2 pid=11210>
<Host h3: h3-eth0:10.0.0.3 pid=11212>
<Host h4: h4-eth0:10.0.0.4 pid=11214>
<Host h5: h5-eth0:10.0.0.5 pid=11216>
<Host h6: h6-eth0:10.0.0.6 pid=11218>
<Host h7: h7-eth0:10.0.0.7 pid=11220>
<Host h8: h8-eth0:10.0.0.8 pid=11222>
<Host h9: h9-eth0:10.0.0.9 pid=11224>
<Host h10: h10-eth0:10.0.0.10 pid=11226>
<Host h11: h11-eth0:10.0.0.11 pid=11228>
<Host h12: h12-eth0:10.0.0.12 pid=11230>
<Host h13: h13-eth0:10.0.0.13 pid=11232>
<Host h14: h14-eth0:10.0.0.14 pid=11234>
<Host h15: h15-eth0:10.0.0.15 pid=11236>
<Host h16: h16-eth0:10.0.0.16 pid=11238>
<Host h17: h17-eth0:10.0.0.17 pid=11240>
<Host h18: h18-eth0:10.0.0.18 pid=11242>
<Host h19: h19-eth0:10.0.0.19 pid=11244>
<Host h20: h20-eth0:10.0.0.20 pid=11246>
<Host h21: h21-eth0:10.0.0.21 pid=11248>
<Host h22: h22-eth0:10.0.0.22 pid=11250>
<Host h23: h23-eth0:10.0.0.23 pid=11252>
<Host h24: h24-eth0:10.0.0.24 pid=11254>
<Host h25: h25-eth0:10.0.0.25 pid=11256>
<Host h26: h26-eth0:10.0.0.26 pid=11258>
<Host h27: h27-eth0:10.0.0.27 pid=11260>
<Host h28: h28-eth0:10.0.0.28 pid=11262>
<Host h29: h29-eth0:10.0.0.29 pid=11264>
<Host h30: h30-eth0:10.0.0.30 pid=11266>
<Host h31: h31-eth0:10.0.0.31 pid=11268>
<Host h32: h32-eth0:10.0.0.32 pid=11270>
<Host h33: h33-eth0:10.0.0.33 pid=11272>
<Host h34: h34-eth0:10.0.0.34 pid=11274>
<Host h35: h35-eth0:10.0.0.35 pid=11276>
<Host h36: h36-eth0:10.0.0.36 pid=11278>
<Host h37: h37-eth0:10.0.0.37 pid=11280>
<Host h38: h38-eth0:10.0.0.38 pid=11282>
<Host h39: h39-eth0:10.0.0.39 pid=11284>
<Host h40: h40-eth0:10.0.0.40 pid=11286>
<Host h41: h41-eth0:10.0.0.41 pid=11288>
<Host h42: h42-eth0:10.0.0.42 pid=11290>
<Host h43: h43-eth0:10.0.0.43 pid=11292>
<Host h44: h44-eth0:10.0.0.44 pid=11294>
<Host h45: h45-eth0:10.0.0.45 pid=11296>
<Host h46: h46-eth0:10.0.0.46 pid=11298>
<Host h47: h47-eth0:10.0.0.47 pid=11300>
<Host h48: h48-eth0:10.0.0.48 pid=11302>
<Host h49: h49-eth0:10.0.0.49 pid=11304>
<Host h50: h50-eth0:10.0.0.50 pid=11306>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=11311>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=11314>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=11317>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None pid=11320>
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None,s5-eth3:None pid=11323>
<OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth2:None,s6-eth3:None pid=11326>
<OVSSwitch s7: lo:127.0.0.1,s7-eth1:None,s7-eth2:None,s7-eth3:None pid=11329>
<OVSSwitch s8: lo:127.0.0.1,s8-eth1:None,s8-eth2:None,s8-eth3:None pid=11332>
<OVSSwitch s9: lo:127.0.0.1,s9-eth1:None,s9-eth2:None,s9-eth3:None pid=11335>
<OVSSwitch s10: lo:127.0.0.1,s10-eth1:None,s10-eth2:None,s10-eth3:None pid=11338>
<OVSSwitch s11: lo:127.0.0.1,s11-eth1:None,s11-eth2:None,s11-eth3:None pid=11341>
<OVSSwitch s12: lo:127.0.0.1,s12-eth1:None,s12-eth2:None,s12-eth3:None pid=11344>
<OVSSwitch s13: lo:127.0.0.1,s13-eth1:None,s13-eth2:None,s13-eth3:None pid=11347>
<OVSSwitch s14: lo:127.0.0.1,s14-eth1:None,s14-eth2:None,s14-eth3:None pid=11350>
<OVSSwitch s15: lo:127.0.0.1,s15-eth1:None,s15-eth2:None,s15-eth3:None pid=11353>
<OVSSwitch s16: lo:127.0.0.1,s16-eth1:None,s16-eth2:None,s16-eth3:None pid=11356>
<OVSSwitch s17: lo:127.0.0.1,s17-eth1:None,s17-eth2:None,s17-eth3:None pid=11359>
<OVSSwitch s18: lo:127.0.0.1,s18-eth1:None,s18-eth2:None,s18-eth3:None pid=11362>
<OVSSwitch s19: lo:127.0.0.1,s19-eth1:None,s19-eth2:None,s19-eth3:None pid=11365>
<OVSSwitch s20: lo:127.0.0.1,s20-eth1:None,s20-eth2:None,s20-eth3:None pid=11368>
```

Εικόνα 7-13 Πληροφορίες σύνδεσης των hosts και των switches



Με την εντολή *pingall* ελέγχουμε την επικοινωνίας των hosts και βλέπουμε ότι το δίκτυο μας μετά από 12 περίπου λεπτά ολοκληρώνει επιτυχώς όλα τα pings και τελειώνει η προσομοίωση μας έχοντας στείλει 2450 πακέτα.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28
h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28
h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28
h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27
h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h50
h50 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27
h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49
*** Results: 0% dropped (2450/2450 received)
mininet>
```

Εικόνα 7-14 Pingall στους hosts

Πιο κάτω ορίζουμε κάποιες ακόμη βασικές εντολές του Mininet

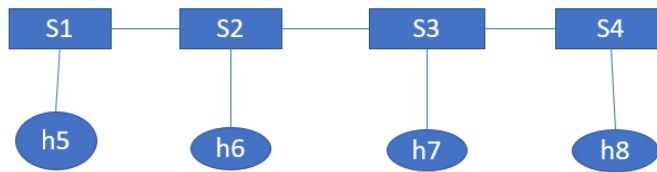
- *--topo* – ορίζει μια συγκεκριμένη τοπολογία όταν το mininet ξεκινά
- *--switch* – ορίζει το switch που θα χρησιμοποιηθεί. By default χρησιμοποιείται το OVS switch
- *--controller* – ο ελεγκτής που θα προσθέσουμε και στη συνέχεια που μπορούμε να βλέπουμε τις τοπολογίες που δημιουργήσαμε και τους πίνακες των hosts.

7.3 Python scripts που υλοποιήθηκαν

Μιας και το mininet τρέχει python scripts, η πιο απλή του εντολή *sudo mn* τρέχει στο παρασκήνιο python, παρακάτω θα δούμε κάποια απλά python scripts που δημιουργήσαμε όπου τρέχοντας τα δημιουργούνται κάποιες τοπολογίες. Το παρακάτω script δημιουργεί 4 switch στη σειρά από ένα host το καθένα σχηματικά αναπαρίσταται στην εικόνα 7-15

1^ο 4swINROW.PY (4switch 4 host)

```
#!/usr/bin/env python
from mininet.net import Mininet /* πακέτα που γίνονται import σε python*/
from mininet.topo import LinearTopo /* πακέτα που γίνονται import σε python*/
Linear = LinearTopo(k=4) /* κάνε την τοπολογία linear και δημιούργησε 4 κόμβους */
net = Mininet(topo=Linear) /* πέρασμα της τοπολογίας στο Mininet*/
net.start() /* ξεκινάμε τον εξομοιωτή Mininet*/
net.pingAll() /* εκτέλεση της εντολής pingall*/
net.stop()/* σταματάμε το mininet*/
```



Εικόνα 7-15 Σχηματική αναπαράσταση του κώδικα

Έπειτα για να τρέξουμε το script που έχουμε δημιουργήσει δίνουμε το path που το έχουμε αποθηκεύσει και τρέχοντας *sudo python όνομα script.py* φορτώνεται στο mininet και βλέπουμε ότι οι hosts επικοινωνούν κανονικά.

```
cg@cg-VirtualBox:~/mininet/custom/ΜΥΤΟΡΟΣ sudo python 4swINROW.py
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
```

Εικόνα 7-16 4swINROW.py

2° 2HONESW.PY (2 hosts σε ένα switch)

```
#!/usr/bin/env python
```

```
from mininet.net import Mininet /*import τα απαραίτητα modules για τρέξει το script μας*/
```

```
net=Mininet() /*δημιουργία instance(στιγμιότυπο) ενός Mininet emulator */
```

```
/*Δημιουργία των κόμβων της τοπολογίας*/
```

```
c0=net.addController() /*Δημιουργία controller*/
```

```
h0=net.addHost('myH0') /*Δημιουργία host με όνομα myH0*/
```

```
s0=net.addSwitch('myS0') /*Δημιουργία switch με όνομα myS0*/
```

```
h1=net.addHost('myH1') /* Δημιουργία host με όνομα myH1 */
```

```
/*Δημιουργία ζεύξεων μεταξύ των κόμβων του δικτύου*/
```

```
net.addLink(h0,s0) /*Δημιουργία ζεύξης με την μέθοδο addLink μεταξύ h0-s0*/
```

```
net.addLink(h1,s0) /*Δημιουργία ζεύξης με την μέθοδο addLink μεταξύ h1-s0*/
```

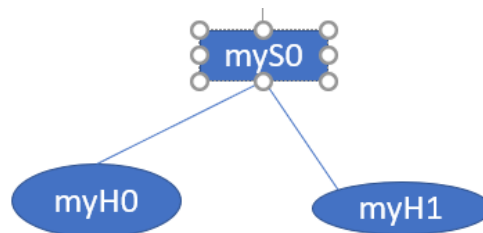
```
h0.setIP('192.168.1.1',24) /*ανάθεση IP διευθύνσεων στον h0 με την μέθοδο setIP
```

```
*/
```



```
h1.setIP('192.168.1.2',24) /*ανάθεση IP διεύθυνσης στον h1 με την μέθοδο setIP */
```

```
net.start()/* ξεκινάμε τον εξομοιωτή Mininet*/  
net.pingAll()/* εκτέλεση της εντολής pingall*/  
net.stop()/* σταματάμε τον mininet εξομοιωτή*/
```



Εικόνα 7-17 Σχηματική αναπαράσταση του κώδικα

Όπως και πριν φορτώνουμε το script και βλέπουμε ότι τρέχει κανονικά και το ping δουλεύει και από τις δύο πλευρές

```
cg@cg-VirtualBox:~/mininet/custom/MYTOPO$ sudo python 2HONESW.py  
[sudo] password for cg:  
*** Ping: testing ping reachability  
myH0 -> myH1  
myH1 -> myH0  
*** Results: 0% dropped (2/2 received)
```

Εικόνα 7-18 2HONESW.py

3^ο 4host1Switch (4host συνδεδεμένοι σε ένα switch)

```
#!/usr/bin/env python  
/*Απαιράιτητα imports για να δουλέψει ο κώδικας μας*/  
from mininet.topo import Topo  
from mininet.net import Mininet  
from mininet.util import dumpNodeConnections  
from mininet.log import setLogLevel
```



```
class SingleSwitchTopo(Topo): /*Υποκλάση της κλάσης Topo*/
def __init__(self, n=2, **opts): /* Constructor της κλάσης SingleSwitchTopo*/

    Topo.__init__(self, **opts) /*Καλούμε τον constructor της υπερκλάσης*/
    switch = self.addSwitch('S1') /*Δημιουργούμε το πρώτο switch με όνομα S1*/

    for h in range(n): /*συνθήκη for που παίρνει μία τιμή n η οποία περνιέται με τον
    constructor και έχει μία default τιμή 2 και ανάλογα με αυτή την τιμή μας λέει πόσες φορές
    θα κάνουμε επανάληψη*/
        host=self.addHost('h%s' % (h + 1)) /*Για κάθε επανάληψη προστίθεται ένας
        host*/
        self.addLink(host, switch) /*Για κάθε host προστίθεται ένα Link που συνδέει τον
        κάθε host με το switch

def simpleTest(): /*Μέθοδος που την ονομάζουμε simpleTest()
    topo = SingleSwitchTopo(n=4) /*καλούμε τον constructor και βάζουμε n=4 για να
    δημιουργήσουμε 4 hosts*/
    net = Mininet(topo) /*δημιουργία instance(στιγμιότυπο) ενός Mininet emulator */
    net.start() /* ξεκινάμε τον εξομοιωτή Mininet*/
    dumpNodeConnections(net.hosts) /* μέθοδος που μας δείχνει αναλυτικές πληροφορίες για
    τα interfaces που είναι συνδεδεμένοι οι hosts και το switch*/
    net.pingAll() /* εκτέλεση της εντολής pingall για να κάνουν ping όλοι οι hosts μεταξύ τους*/

    h1 = net.get('h1') /* εμφανίζονται οι πληροφορίες του interface του h1*/
    result = h1.cmd('ifconfig')
    print result
    net.stop() /* σταματάμε τον mininet εξομοιωτή*/
    /*Καλούμε την μέθοδο simpleTest()*/

if __name__ == '__main__':
    setLogLevel('info')
    simpleTest()
```




```
cg@cg-VirtualBox:~/mininet/custom/MYTOP0$ sudo python 4host1Switch.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
S1
*** Adding links:
(h1, S1) (h2, S1) (h3, S1) (h4, S1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
S1 ...
Dumping host connections
h1 h1-eth0:S1-eth1
h2 h2-eth0:S1-eth2
h3 h3-eth0:S1-eth3
h4 h4-eth0:S1-eth4
Test tou diktuou
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
h1-eth0  Link encap:Ethernet  HWaddr b6:c4:b0:a3:e0:cc
        inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
        inet6 addr: fe80:b4c4:b0ff:fea3:e0cc/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:19  errors:0  dropped:0  overruns:0  frame:0
        TX packets:13  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0  txqueuelen:1000
        RX bytes:1454 (1.4 KB)  TX bytes:1022 (1.0 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128  Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:0  errors:0  dropped:0  overruns:0  frame:0
        TX packets:0  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0  txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

*** Stopping 1 controllers
c0
*** Stopping 4 links
...
*** Stopping 1 switches
S1
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
```

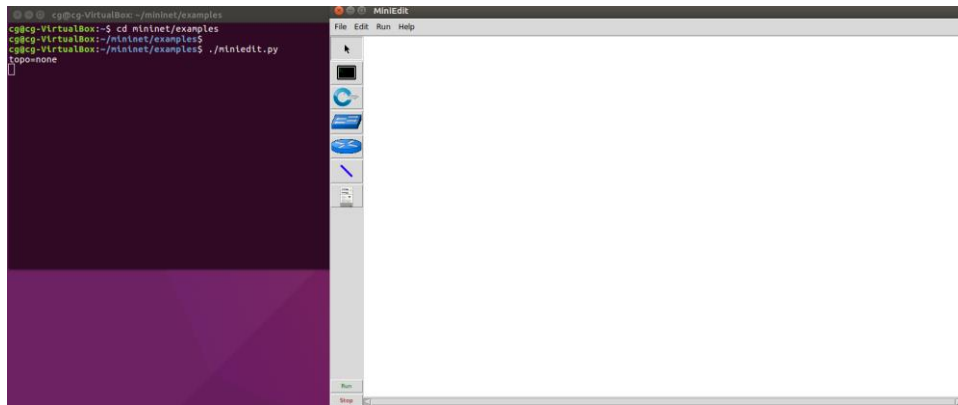
Εικόνα 7-19 4host1Switch.py

Βλέπουμε ότι ο κώδικας μας τρέχει σωστά δημιουργώντας τον controller προσθέτοντας 4 hosts όπως έχουμε γράψει στο script, 1 switch, links από το S1 προς όλους τους hosts. Εκκινεί τον controller και το switch μέσα από την συνάρτηση dump μας εμφανίζει πληροφορίες για τη σύνδεση, το ping δουλεύει επιτυχώς από όλους τους host. Εμφανίζει τις πληροφορίες του interface του host1 και τέλος σταματάει ότι δημιουργήσαμε.

7.4 Δημιουργία τοπολογιών διαμέσω του script Miniedit

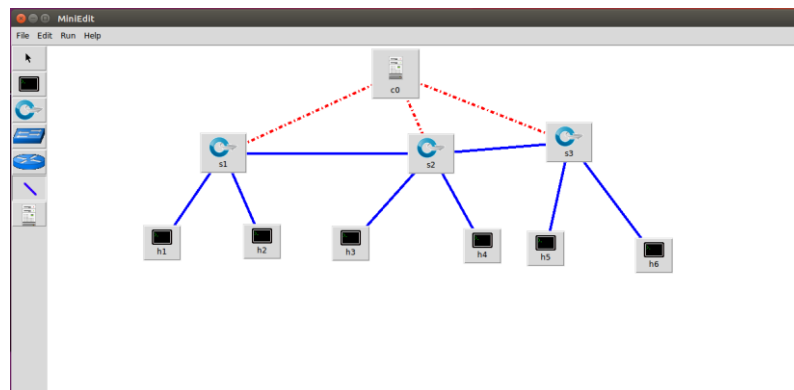
Το Mininet εξαιτίας του open source χαρακτήρα του μπορεί να προσεγγιστεί από πολλές πλευρές. Η κοινότητα του Mininet δημιούργησε ένα πειραματικό script το οποίο αποσκοπεί σε μια μορφή εικονοποίησης της εκάστοτε τοπολογίας που προσομοιώνεται. Οι δυνατότητες του είναι περιορισμένες αν σκεφτεί κανείς άλλες πλατφόρμες με γραφικό περιβάλλον ωστόσο είναι αρκετά λειτουργικό για το πειραματικό επίπεδο που το δουλεύουμε εμείς.

Για να δούμε τις λειτουργίες του Miniedit εντοπίζουμε το path που βρίσκεται το script και το τρέχουμε πάντα με *sudo* για να μπορούμε στη συνέχεια να το κάνουμε edit αλλιώς θα μπορούμε μόνο να το κάνουμε view και μας εμφανίζεται το παρακάτω γραφικό. Ουσιαστικά εμείς κάνουμε drag and drop hosts, switches, controller και αυτά τρέχουν στο παρασκήνιο και εμείς μπορούμε να τα παρακολουθούμε



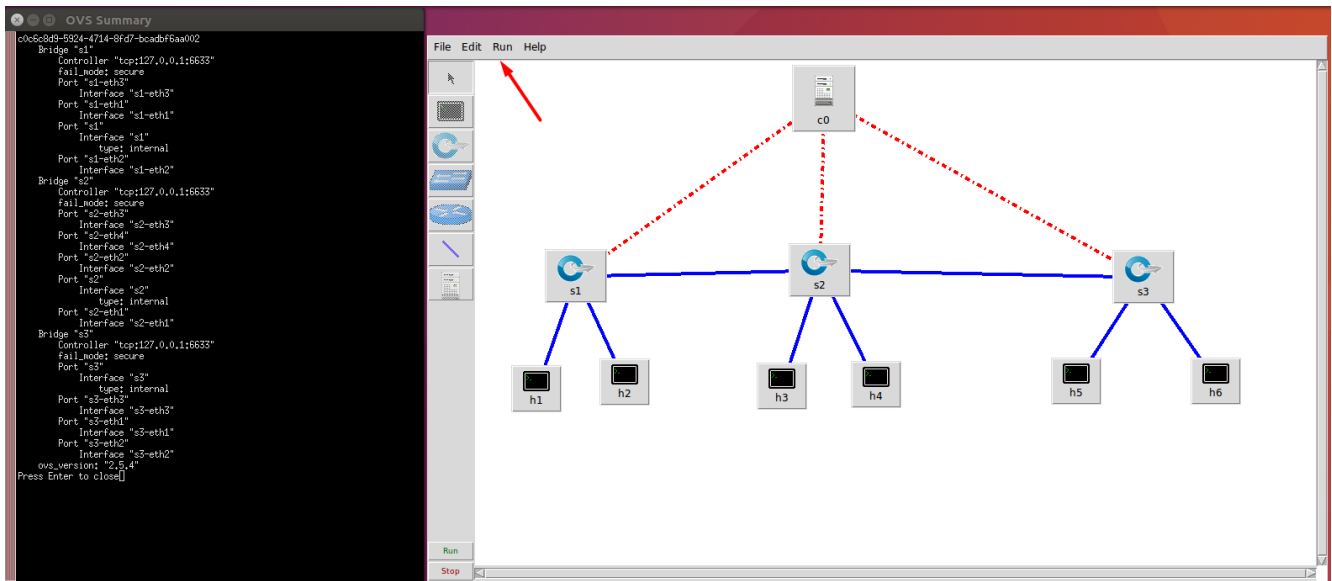
Εικόνα 7-20 Γραφικό περιβάλλον του Miniedit script

Αυτό που θα κάνουμε είναι φτιάξουμε μία τοπολογία που θα διαθέτει 6 hosts και 3 switches και έναν ελεγκτή. Αφού τα έχουμε κάνει drag and drop στον καμβά μας πρέπει να πατήσουμε το κουμπί *Select* που έχει στην παλέτα και μετά το κουμπί *NetLink*. Τα συνδέουμε μεταξύ τους, κάθε switch να έχει δύο hosts, τα switch να συνδέονται μεταξύ τους και μετά και το κάθε switch να συνδέεται στον controller.



Εικόνα 7-21 Τοπολογία στο Miniedit

Στη συνέχεια αυτό που πρέπει να κάνουμε είναι να το κάνουμε export σαν script την εν λόγω τοπολογία που δημιουργήσαμε με drag and drop έτσι ώστε να μπορούμε να την τρέξουμε ξανά. Έπειτα τρέχουμε την τοπολογία στην καρτέλα που λέει *Run* στην παλέτα μας και στην συνέχεια πατάμε το *show ovs-summary* στην ίδια καρτέλα και βλέπουμε τα bridges που έχουν δημιουργηθεί στο εκάστοτε switch και τα interfaces που είναι συνδεδεμένο με τους host και με το επόμενο switch. Βλέπουμε ότι όσο η τοπολογία τρέχει δεν μπορούμε να προσθέσουμε κάτι άλλο σε αυτή, είναι σε executive mode.



Εικόνα 7-22 Bridges της τοπολογίας του Miniedit

Από την 7.22 βλέπουμε όπως αναφέραμε και πιο πάνω 3 γέφυρες την s1,s2,s3 σε κάθε γέφυρα τις εκάστοτε πόρτες που δημιουργούνται και φυσικά τον controller που σε αυτό το σημείο είναι στο localhost,θα ασχοληθούμε με το remote monitoring του ελεγκτή στην επόμενη ενότητα.

Μετά πατάμε quit στην τοπολογία μας από την παλέτα του Miniedit, *File->Quit* και βλέπουμε τι είχε τρέξει στο παρασκήνιο μέχρι στιγμής. Βλέπουμε ότι εμφανίζονται ορθώς οι κινήσεις που κάναμε όπως να ορίσουμε 6 hosts ,3 switch και ένα controller καθώς και το σταμάτημα αυτών.

```
topo=None
Getting Hosts and Switches.
Getting controller selection:ref
Getting Links.
*** Configuring hosts
h3 h4 h2 h6 h1 h5
**** Starting 1 controllers
c0
**** Starting 3 switches
s1 s3 s2
No NetFlow targets specified.
No sFlow targets specified.
*** Stopping 1 controllers
c0
*** Stopping 8 links
*****
*** Stopping 3 switches
s1 s3 s2
*** Stopping 6 hosts
h3 h4 h2 h6 h1 h5
*** Done
```

Εικόνα 7-23 Background δεδομένα της τοπολογίας

Καθαρίζουμε το περιβάλλον του Mininet με την εντολή *sudo mn -c* σε περίπτωση που έχει μείνει κάποιον ‘απομεινάρι’ προηγούμενης τοπολογίας. Βρίσκουμε το path που έχουμε το script και πάντα με sudo το τρέχουμε. Βλέπουμε ότι το script που δημιουργήσαμε στο Miniedit, εδουλεύει με επιτυχία καθώς το τρέχουμε στο περιβάλλον του Mininet.



```
*** Cleanup complete.
cg@cg-VirtualBox:~/mininet/examples$ ls
baresshd.py      cpu.py           mobility.py      scratchnet.py
bind.py          emptynet.py     multilink.py   scratchnetuser.py
clustercli.py    first.py        multiping.py   simpleperf.py
clusterdemo.py  hwintf.py      multipoll.py   sshd.py
clusterperf.py  __init__.py    multitest.py  test
cluster.py       intfoptions.py natnet.py      tree1024.py
clusterSanity.py limit.py        nat.py         treeping64.py
consoles.py     linearbandwidth.py numberedports.py vlanhost.py
controllers2.py linuxrouter.py  popenpoll.py
controllers.py  MINIEDITMYCG.py popen.py
controlnet.py  mntedit.py     README.md
cg@cg-VirtualBox:~/mininet/examples$ sudo python ./MINIEDITMYCG.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h3 h4 h2 h6 h1 h5
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet>
```

Εικόνα 7-24 Επιτυχής φόρτωση δημιουργημένου script από το Miniedit στο Mininet

Τρέχοντας τις εντολές *nodes*, *net*, *dump* ελέγχουμε ότι το script μας είναι λειτουργικό. Παρατηρούμε ότι υπάρχουν 10 κόμβοι στην τοπολογία μας όπως έχουμε ορίσει, βλέπουμε την διασύνδεση τους μέσω της εντολής *net* ποια interfaces έχουν συνδεθεί που και με το *dump* βλέπουμε τις εικονικές ip των hosts.

```
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 s1 s2 s3
mininet> net
h3 h3-eth0:s2-eth2
h4 h4-eth0:s2-eth3
h2 h2-eth0:s1-eth2
h6 h6-eth0:s3-eth3
h1 h1-eth0:s1-eth1
h5 h5-eth0:s3-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:s2-eth1
s3 lo: s3-eth1:s2-eth4 s3-eth2:h5-eth0 s3-eth3:h6-eth0
s2 lo: s2-eth1:s1-eth3 s2-eth2:h3-eth0 s2-eth3:h4-eth0 s2-eth4:s3-eth1
c0
mininet> dump
<Host h3: h3-eth0:10.0.0.3 pid=5698>
<Host h4: h4-eth0:10.0.0.4 pid=5700>
<Host h2: h2-eth0:10.0.0.2 pid=5702>
<Host h6: h6-eth0:10.0.0.6 pid=5704>
<Host h1: h1-eth0:10.0.0.1 pid=5706>
<Host h5: h5-eth0:10.0.0.5 pid=5708>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=5687>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=5690>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2-eth4:None
pid=5693>
<Controller c0: 127.0.0.1:6633 pid=5679>
mininet>
```

Εικόνα 7-25 Πληροφορίες της τοπολογίας MiniEdit



Αφού μιλάμε για Open vSwitch, με άλλα λόγια μεταγωγείς που μπορούμε να παρατηρήσουμε τους πίνακες ροής τους θα ελέγξουμε την κίνηση και στη δική μας τοπολογία. Ένας μεταγωγέας OpenFlow περιέχει πίνακες ροής (flow tables) και πίνακες ομάδας (group tables), που επιτελούν χειρισμούς όπως αναζήτηση και προώθηση πακέτων διαμέσων ενός διαύλου OpenFlow που είναι συνδεδεμένο με το controller. Ο ελεγκτής (controller) διευθύνει το switch χρησιμοποιώντας το πρωτόκολλο OpenFlow. Με τη χρήση του πρωτόκολλου, ο controller προσθέτει, ενημερώνει και διαγράφει καταχωρήσεις που εντοπίζονται στα flow tables.

Πηγαίνουμε στην τοπολογία που έχουμε φορτώσει και τρέχει στο Mininet έτσι ώστε να ενεργοποιήσουμε τους μεταγωγείς για να ξεκινήσουν τον εντοπισμό των δεδομένων. Γράφουμε την εντολή `dpctl dump-flows` και παρατηρούμε ότι οι μεταγωγείς είναι ανοιχτοί και περιμένουν κάποια κίνηση μέσα στο δίκτυο.

```
mininet> dpctl dump-flows
*** s1 -----
NXST_FLOW reply (xid=0x4):
*** s3 -----
NXST_FLOW reply (xid=0x4):
*** s2 -----
NXST_FLOW reply (xid=0x4):
mininet>
```

Εικόνα 7-26 Ενεργοποίηση των Open vSwitches

Για να δημιουργήσουμε κίνηση στο δίκτυο μας θα στείλουμε 3 πακέτα από τον h1 προς τον h2 που βρίσκονται στο ίδιο OpenvSwitch το S1 και βλέπουμε ότι δημιουργείται κίνηση σε αυτόν τον μεταγωγέα.

```
mininet> h1 ping -c 3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.73 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.441 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.065 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2019ms
rtt min/avg/max/mdev = 0.065/1.748/4.739/2.120 ms
mininet> dpctl dump-flows
*** s1 -----
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=5.229s, table=0, n_packets=1, n_bytes=42, idle_timeout=60,
 idle_age=5, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=3e:d4:d1:09:d2:
 b4,dl_dst=ba:88:f3:5f:f1:a7,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions=0
 output:1
 cookie=0x0, duration=0.139s, table=0, n_packets=1, n_bytes=42, idle_timeout=60,
 idle_age=0, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=3e:d4:d1:09:d2:
 b4,dl_dst=ba:88:f3:5f:f1:a7,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=1 actions=0
 output:1
 cookie=0x0, duration=0.137s, table=0, n_packets=1, n_bytes=42, idle_timeout=60,
 idle_age=0, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=ba:88:f3:5f:f1:
 a7,dl_dst=3e:d4:d1:09:d2:b4,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions=0
 output:2
 cookie=0x0, duration=5.226s, table=0, n_packets=3, n_bytes=294, idle_timeout=60
 , idle_age=3, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=ba:88:f3:5f:f
 1:a7,dl_dst=3e:d4:d1:09:d2:b4,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type
 =8,icmp_code=0 actions=output:2
 cookie=0x0, duration=5.227s, table=0, n_packets=3, n_bytes=294, idle_timeout=60
 , idle_age=3, priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=3e:d4:d1:09:d
 2:b4,dl_dst=ba:88:f3:5f:f1:a7,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type
 =0,icmp_code=0 actions=output:1
*** s3 -----
NXST_FLOW reply (xid=0x4):
*** s2 -----
NXST_FLOW reply (xid=0x4):
mininet>
```

Εικόνα 7-27 Επιτυχή μετάδοση και παρακολούθηση από h1 σε h2

Από την εικόνα 7-27 βλέπουμε ότι μεταδώσαμε σωστά τα πακέτα από τον host 1 στον host 2, όπως ήταν αναμενόμενο πρώτα εμφανίζονται οι ARP αιτήσεις.



7.4.1 ARP αιτήσεις

Όταν το ARP πάρει μία IP διεύθυνση, ψάχνει τον ARP πίνακα για να δει αν υπάρχει εγγραφή που να αντιστοιχεί σε αυτή την IP διεύθυνση. Αν υπάρχει εγγραφή, τότε επιστρέφει την αντίστοιχη φυσική διεύθυνση, διαφορετικά αν δεν υπάρχει, στέλνει ένα ειδικό μήνυμα που *ονομάζεται ARP αίτηση* σε όλες τις συσκευές του δικτύου. Εάν μία συσκευή αναγνωρίσει στην IP διεύθυνση προορισμού της αίτησης, τη δική της IP διεύθυνση, στέλνει μία ARP απάντηση με τη δική της φυσική διεύθυνση στη συσκευή που έστειλε την συσκευή που έστειλε την ARP αίτηση. Η ARP απάντηση έχει την ίδια μορφή με ARP αίτηση, με τη διαφορά, ότι έχει ανάποδα τα πεδία αποστολέα και προορισμού και προορισμού. Η συσκευή που δημιούργησε την ARP αίτηση, δημιουργεί μια νέα εγγραφή στον ARP πίνακά της και βάζει εκεί τη διεύθυνση που μόλις έλαβε. Όταν χρειαστεί ξανά να βρει τη φυσική διεύθυνση αυτής της συσκευής, θα διαβάσει απλώς τον πίνακα και δεν θα χρειαστεί να δημιουργηθεί νέο αίτημα ARP. [19]

Αφού έχουν αντιστοιχιστεί οι διευθύνσεις βλέπουμε μετά διαμέσων του s1 ότι περνάνε τα ICMP πακέτα με πρώτο το ECHO request από τον h1 και μετά το ECHO reply από τον h2. Θα κάνουμε αντίστοιχο πείραμα από τον h1 στον h4 του S2 στέλνοντας 3 πακέτα και μετά να δούμε την κίνηση που θα δημιουργηθεί.

```
mininet> h1 ping -c 3 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data:
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=8.53 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.766 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.071 ms

--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.071/3.125/8.538/3.838 ms
mininet> dpctl dump-flows
*** s1 ***
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=6.205s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=6, priority=65535,arp,in_port=3,vlan_tci=0x0000,dl_src=42:9f:47:92:9c:88,dl_dst=ba:88:f3:5f:f1:a7,arp_spa=10.0.0.4,arp_tpa=10.0.0.1,arp_op=2 actions=output:1
 cookie=0x0, duration=0.978s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=0, priority=65535,arp,in_port=3,vlan_tci=0x0000,dl_src=42:9f:47:92:9c:88,dl_dst=ba:88:f3:5f:f1:a7,arp_spa=10.0.0.4,arp_tpa=10.0.0.1,arp_op=1 actions=output:1
 cookie=0x0, duration=0.975s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=0, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=ba:88:f3:5f:f1:a7,dl_dst=42:9f:47:92:9c:88,arp_spa=10.0.0.1,arp_tpa=10.0.0.4,arp_op=2 actions=output:3
 cookie=0x0, duration=6.204s, table=0, n_packets=3, n_bytes=294, idle_timeout=60, idle_age=4, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=ba:88:f3:5f:f1:a7,dl_dst=42:9f:47:92:9c:88,nw_src=10.0.0.1,nw_dst=10.0.0.4,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:3
 cookie=0x0, duration=6.202s, table=0, n_packets=3, n_bytes=294, idle_timeout=60, idle_age=4, priority=65535,icmp,in_port=3,vlan_tci=0x0000,dl_src=42:9f:47:92:9c:88,dl_dst=ba:88:f3:5f:f1:a7,nw_src=10.0.0.4,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:1
*** s3 ***
NXST_FLOW reply (xid=0x4):
*** s2 ***
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=6.213s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=6, priority=65535,arp,in_port=3,vlan_tci=0x0000,dl_src=42:9f:47:92:9c:88,dl_dst=ba:88:f3:5f:f1:a7,arp_spa=10.0.0.4,arp_tpa=10.0.0.1,arp_op=2 actions=output:1
 cookie=0x0, duration=0.986s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=0, priority=65535,arp,in_port=3,vlan_tci=0x0000,dl_src=42:9f:47:92:9c:88,dl_dst=ba:88:f3:5f:f1:a7,arp_spa=10.0.0.4,arp_tpa=10.0.0.1,arp_op=1 actions=output:1
 cookie=0x0, duration=0.978s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=0, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=ba:88:f3:5f:f1:a7,dl_dst=42:9f:47:92:9c:88,arp_spa=10.0.0.1,arp_tpa=10.0.0.4,arp_op=2 actions=output:3
 cookie=0x0, duration=6.209s, table=0, n_packets=3, n_bytes=294, idle_timeout=60, idle_age=4, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=ba:88:f3:5f:f1:a7,dl_dst=42:9f:47:92:9c:88,nw_src=10.0.0.1,nw_dst=10.0.0.4,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:3
 cookie=0x0, duration=6.208s, table=0, n_packets=3, n_bytes=294, idle_timeout=60, idle_age=4, priority=65535,icmp,in_port=3,vlan_tci=0x0000,dl_src=42:9f:47:92:9c:88,dl_dst=ba:88:f3:5f:f1:a7,nw_src=10.0.0.4,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:1
mininet>
```

Εικόνα 7-28 Επιτυχή μετάδοση και παρακολούθηση πακέτων από τον h1 στον h4

Παρατηρούμε ότι στέλνοντας πακέτα από τον h1->h4 δημιουργείται η ίδια ακριβώς κίνηση που υπήρχε στον S1 και στο μεταγωγέα S2 στέλνοντας πρώτα τις αιτήσεις ARP για γνωστοποίηση και μετά τα ICMP πακέτα με source όπως βλέπουμε την 10.0.0.1 του h1 και dest 10.0.0.4 του h4.



Στέλνοντας τώρα πακέτα από τον h1 στον h6 του S3 βλέπουμε ότι οι ARP αιτήσεις πέφτουν μόνο σε μία καθώς έχει γίνει γνωστοποίηση των άλλων δύο μεταγωγέων μεταξύ τους. Με κόκκινο βέλος βλέπουμε τη μετάδοση πακέτου από h1->h6 και τους μεταγωγείς που ελέγχουν την κίνηση. Με κίτρινο βέλος παρακολουθούμε τις arp αιτήσεις σε κάθε μεταγωγέα και με πράσινο τα ICMP πακέτα με source όπως βλέπουμε την 10.0.0.1 του h1 και dest 10.0.0.6 του h6 και το αντίστροφο στο echo reply μήνυμα.

```
mininet> h1 ping -c 3 h6
PING 10.0.0.6 (10.0.0.6) 56(84) bytes of data.
64 bytes from 10.0.0.6: icmp_seq=1 ttl=64 time=13.6 ms
64 bytes from 10.0.0.6: icmp_seq=2 ttl=64 time=0.896 ms
64 bytes from 10.0.0.6: icmp_seq=3 ttl=64 time=0.083 ms

--- 10.0.0.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.083/4.891/13.696/6.234 ms
mininet> dpctl dump-flows
*** s1 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=3.792s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=3, priority=65535, arp, in_port=3, vlan_tci=0x0000, dl_src=12:0c:98:dc:93:c3, dl_dst=ba:88:f3:5f:f1:a7, arp_spa=10.0.0.6, arp_tpa=10.0.0.1, arp_op=2 actions=output:1
  cookie=0x0, duration=3.789s, table=0, n_packets=3, n_bytes=294, idle_timeout=60, idle_age=1, priority=65535, icmp, in_port=1, vlan_tci=0x0000, dl_src=ba:88:f3:5f:f1:a7, dl_dst=12:0c:98:dc:93:c3, nw_src=10.0.0.1, nw_dst=10.0.0.6, nw_tos=0, icmp_type=8, icmp_code=0 actions=output:3
  cookie=0x0, duration=3.782s, table=0, n_packets=3, n_bytes=294, idle_timeout=60, idle_age=1, priority=65535, icmp, in_port=3, vlan_tci=0x0000, dl_src=12:0c:98:dc:93:c3, dl_dst=ba:88:f3:5f:f1:a7, nw_src=10.0.0.6, nw_dst=10.0.0.1, nw_tos=0, icmp_type=0, icmp_code=0 actions=output:1
*** s3 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=3.794s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=3, priority=65535, arp, in_port=3, vlan_tci=0x0000, dl_src=12:0c:98:dc:93:c3, dl_dst=ba:88:f3:5f:f1:a7, arp_spa=10.0.0.6, arp_tpa=10.0.0.1, arp_op=2 actions=output:1
  cookie=0x0, duration=3.790s, table=0, n_packets=3, n_bytes=294, idle_timeout=60, idle_age=1, priority=65535, icmp, in_port=1, vlan_tci=0x0000, dl_src=ba:88:f3:5f:f1:a7, dl_dst=12:0c:98:dc:93:c3, nw_src=10.0.0.1, nw_dst=10.0.0.6, nw_tos=0, icmp_type=8, icmp_code=0 actions=output:3
  cookie=0x0, duration=3.788s, table=0, n_packets=3, n_bytes=294, idle_timeout=60, idle_age=1, priority=65535, icmp, in_port=3, vlan_tci=0x0000, dl_src=12:0c:98:dc:93:c3, dl_dst=ba:88:f3:5f:f1:a7, nw_src=10.0.0.6, nw_dst=10.0.0.1, nw_tos=0, icmp_type=0, icmp_code=0 actions=output:1
*** s2 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=3.797s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=3, priority=65535, arp, in_port=4, vlan_tci=0x0000, dl_src=12:0c:98:dc:93:c3, dl_dst=ba:88:f3:5f:f1:a7, arp_spa=10.0.0.6, arp_tpa=10.0.0.1, arp_op=2 actions=output:1
  cookie=0x0, duration=3.793s, table=0, n_packets=3, n_bytes=294, idle_timeout=60, idle_age=1, priority=65535, icmp, in_port=1, vlan_tci=0x0000, dl_src=ba:88:f3:5f:f1:a7, dl_dst=12:0c:98:dc:93:c3, nw_src=10.0.0.1, nw_dst=10.0.0.6, nw_tos=0, icmp_type=8, icmp_code=0 actions=output:4
  cookie=0x0, duration=3.790s, table=0, n_packets=3, n_bytes=294, idle_timeout=60, idle_age=1, priority=65535, icmp, in_port=4, vlan_tci=0x0000, dl_src=12:0c:98:dc:93:c3, dl_dst=ba:88:f3:5f:f1:a7, nw_src=10.0.0.6, nw_dst=10.0.0.1, nw_tos=0, icmp_type=0, icmp_code=0 actions=output:1
mininet>
```

Εικόνα 7-29 Επιτυχή μετάδοση και παρακολούθηση από h1 σε h6

7.4.2 Capture πακέτων διαμέσων του Wireshark

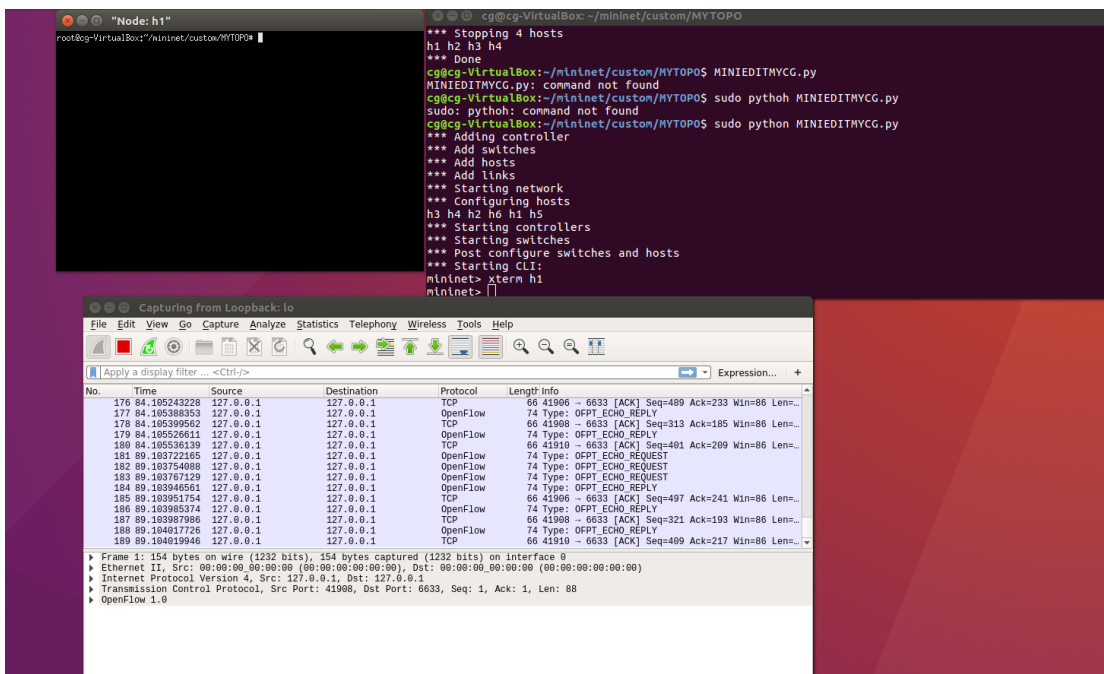
Το επόμενο που θα κάνουμε είναι να κάνουμε capture κάποια δεδομένα μέσω από το περιβάλλον του Wireshark. Έχοντας κατεβάσει επιτυχώς το wireshark στο περιβάλλον των Ubuntu 16.04 LTS που χρησιμοποιούμε θα φορτώσουμε την προηγούμενη τοπολογία που δημιουργήσαμε στο Mininet δίνοντας πάντα το σωστό path και super user δικαιώματα.



```
cg@cg-VirtualBox:~/mininet/custom/MYTOPOS$ sudo python MINIEDITMYCG.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h3 h4 h2 h6 h1 h5
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
```

Εικόνα 7-30 Φόρτωση του python script

Έχοντας φορτώσει επιτυχώς το script θα ανοίξουμε ένα ξεχωριστό CLI για τον h1 με την εντολή xterm h1 και θα μεταδώσω πακέτα προς τον h2. Πριν υλοποιήσω αυτό τρέχω από ένα terminal στα ubuntu sudo wireshark και ανοίγουμε το packet sniffer wireshark. Η εικόνα που θα δούμε θα είναι αυτής της μορφής



Εικόνα 7-31 Παρακολούθηση τοπολογίας μέσω του Wireshark

Έχοντας βάλει το Wireshark να κάνει capture στο Loopback interface 127.0.0.1 βλέπουμε τα πρωτόκολλα που χρησιμοποιούνται στην τοπολογία μας που είναι το Openflow και το TCP που χρησιμοποιείται για την μεταφορά των μηνυμάτων στο δίκτυο. Πηγαίνοντας στο terminal του h1 στέλνω 3 πακέτα στον h2 που μεταδίδονται όπως βλέπουμε διαμέσων του OpenFlow πρωτοκόλλου.



| No. | Time | Source | Destination | Protocol | Length | Info |
|------|---------------|-------------------|-------------------|----------|--------|---|
| 1175 | 597.099250301 | 127.0.0.1 | 127.0.0.1 | OpenFlow | 90 | Type: OFPT_PACKET_OUT |
| 1176 | 597.099254761 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 41910 → 6633 [ACK] Seq=2177 Ack=1257 Win=86 Len=0 TSval=574065837 TSecr=574065837 |
| 1177 | 597.100507505 | aa:b7:d0:b1:7b:d6 | da:bc:e0:2d:db:fc | OpenFlow | 126 | Type: OFPT_PACKET_IN |
| 1178 | 597.100621066 | 127.0.0.1 | 127.0.0.1 | OpenFlow | 146 | Type: OFPT_FLOW_MOD |
| 1179 | 597.101714553 | da:bc:e0:2d:db:fc | Broadcast | OpenFlow | 126 | Type: OFPT_PACKET_IN |
| 1180 | 597.101792910 | 127.0.0.1 | 127.0.0.1 | OpenFlow | 90 | Type: OFPT_PACKET_OUT |
| 1181 | 597.101796380 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 41908 → 6633 [ACK] Seq=2177 Ack=1257 Win=86 Len=0 TSval=574065839 TSecr=574065839 |
| 1182 | 597.102547623 | 10.0.0.1 | 10.0.0.2 | OpenFlow | 182 | Type: OFPT_PACKET_IN |
| 1183 | 597.102613863 | 127.0.0.1 | 127.0.0.1 | OpenFlow | 146 | Type: OFPT_FLOW_MOD |
| 1184 | 597.103869911 | 10.0.0.2 | 10.0.0.1 | OpenFlow | 182 | Type: OFPT_PACKET_IN |
| 1185 | 597.103939445 | 127.0.0.1 | 127.0.0.1 | OpenFlow | 146 | Type: OFPT_FLOW_MOD |
| 1186 | 597.147993397 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 41906 → 6633 [ACK] Seq=2469 Ack=1497 Win=86 Len=0 TSval=574065886 TSecr=574065842 |
| 1187 | 601.104359982 | 127.0.0.1 | 127.0.0.1 | OpenFlow | 74 | Type: OFPT_ECHO_REQUEST |
| 1188 | 601.104671705 | 127.0.0.1 | 127.0.0.1 | OpenFlow | 74 | Type: OFPT_ECHO_REPLY |

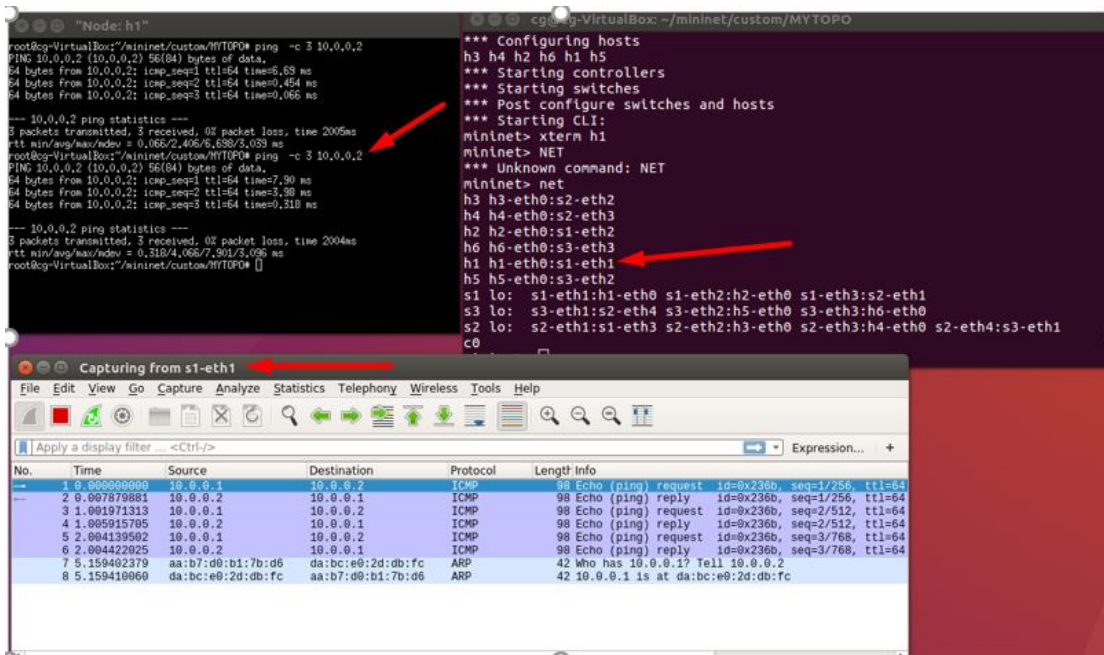
Εικόνα 7-32 Παρατήρηση πακέτων στην ροή του LOOPBACK

Φιλτράροντας τα γράφοντας ICMP, απομονώνουμε τα ICMP πακέτα και βλέπουμε τον αποστολέα και τον παραλήπτη και το πρωτόκολλο που χρησιμοποίησαν για να επιτύχουν την επικοινωνία.

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|---------------|----------|-------------|----------|--------|----------------------|
| 1182 | 597.102547623 | 10.0.0.1 | 10.0.0.2 | OpenFlow | 182 | Type: OFPT_PACKET_IN |
| 1184 | 597.103869911 | 10.0.0.2 | 10.0.0.1 | OpenFlow | 182 | Type: OFPT_PACKET_IN |

Εικόνα 7-33 Φιλτράρισμα ICMP packets

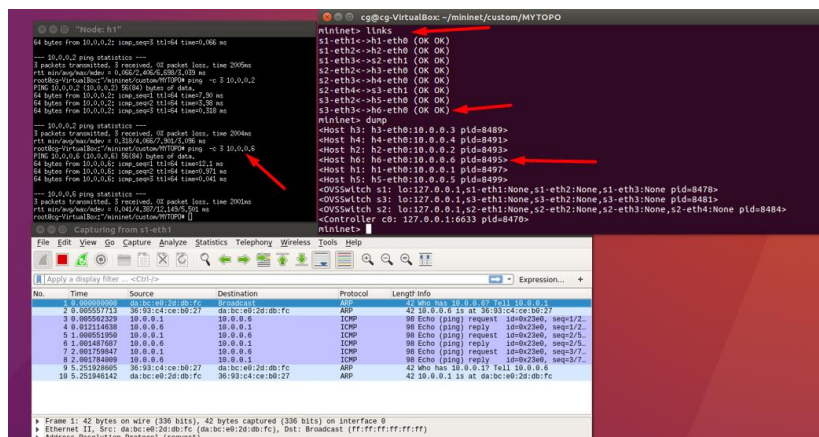
Επίσης αν επιθυμούμε περαιτέρω απομόνωση των πακέτων θα πάμε στο wireshark το interface s1-eth1 την πόρτα ουσιαστικά που χρησιμοποιεί ο h1 για να μεταδώσει τα πακέτα του. Θα πάμε στο τερματικό που τρέχουμε το mininet και γράφοντας `net` βλέπουμε την πόρτα του για να την ρυθμίσουμε σωστά στο Wireshark



Εικόνα 7-34 Αναλυτικά τα ICMP και ARP πακέτα στον h1

Στέλνουμε τα 3 πακέτα ξανά από τον h1 στον h2 έχοντας ρυθμίσει να κάνουμε capture στο κατάλληλο interface (s1-eth1) και βλέπουμε το ECHO request του h1(10.0.0.1) προς τον h1(10.0.0.2) και τα αντίστοιχα ICMP που ανταλλάσσουν. Επίσης βλέπουμε και την ARP αίτηση και την απάντηση που στέλνεται σε μορφή ethernet διεύθυνσης, όπου έπρεπε να γίνει για να εγκαθιδρυθεί η επικοινωνία.

Μετάδοση πακέτου από h1->h6 του S3. Ελέγχουμε ότι τα links δουλεύουν σωστά βλέπουμε την IP του h6 και στέλνουμε πάλι 3 πακέτα.



Εικόνα 7-35 Αναλυτικά τα ICMP και ARP πακέτα του h1 που στέλνει στον h6

Έχοντας επικοινωνήσει στη σωστή πόρτα βλέπουμε την Broadcast ARP αίτηση που στέλνει ο h1 προς τον h6 μετά τα αντίστοιχα ICMP που μεταδίδουν και μετά τις ARP αιτήσεις του h6 έτσι ώστε να “γνωριστούν”.



8 Integration των δικτυακών προσομοιωτών GNS3 & Mininet

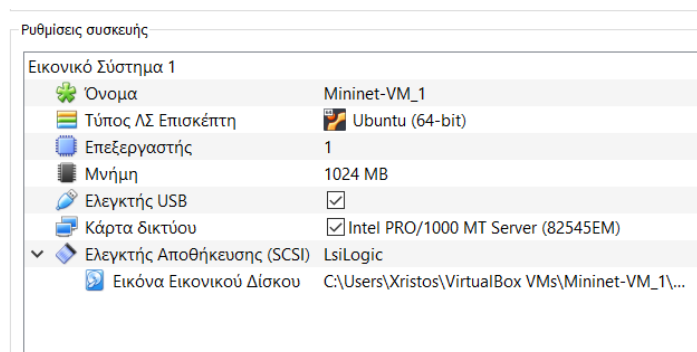
Έχοντας ασχοληθεί με διάφορα δικτυακά πρωτόκολλα και σενάρια στον GNS3 προσομοιωτή ως επακόλουθο έχουμε τη διασύνδεση με τον εξομοιωτή δικτύου Mininet. Το Mininet όπως έχει προαναφερθεί έχει την δυνατότητα δημιουργίας μιας εικονικής τοπολογίας δικτύου, όπου έχουμε τη δυνατότητα ταυτόχρονης εκτέλεσης τερματικών, μεταγωγέων, δρομολογητών, ελεγκτών μέσα σε έναν ενιαίο πυρήνα (kernel) Linux.

8.1 Εγκατάσταση του Mininet

Για να επιτευχθεί η διασύνδεση του εξομοιωτή Mininet αναγκαία προϋπόθεση είναι να κατεβάσουμε από το Github μια precompiled εικονική μηχανική (Virtual Machine) του Mininet την οποία εγκαταστήσαμε και δουλέψαμε στην πλατφόρμα Virtual Box. Η ενέργεια αυτή έγινε διότι χρησιμοποιούμε Windows 10 λειτουργικό σύστημα και το Mininet δεν υποστηρίζει γραφικό περιβάλλον για το συγκεκριμένο λειτουργικό. Επιπλέον επιτακτική ανάγκη για την οπτικοποίηση των εξαγόμενων αποτελεσμάτων του εξομοιωτή είναι η χρήση του Aruba HP SDN Controller. Όπως έχει αναφερθεί και πιο πάνω ο συγκεκριμένος ελεγκτής είναι επίσης μία precompiled εικονική μηχανή την οποία εγκαταστήσαμε χρησιμοποιώντας τα ίδια βήματα όπως του Mininet στο Virtual Box.[55]

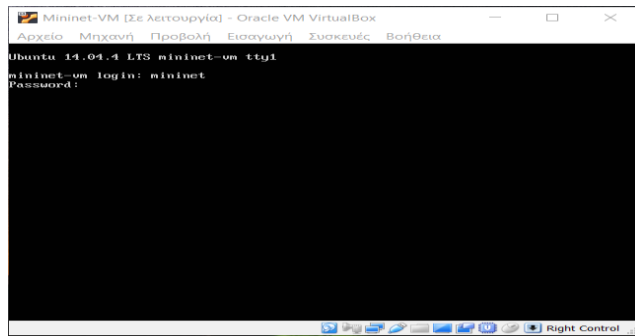
8.2 Βήματα εγκατάστασης των εικονικών μηχανών

Έχοντας κατεβάσει με επιτυχία τα virtual machines ακολουθούμε μια σειρά βημάτων που φαίνεται στα παρακάτω στιγμιότυπα οθόνης. Πηγαίνοντας στην κεντρική οθόνη του Virtual Box ακολουθούμε την διαδρομή Αρχείο->Εισαγωγή Συσκευής->Επιλογή Ova file (στον φάκελο που έχει γίνει η λήψη και δίνουμε τα χαρακτηριστικά που θέλουμε να έχει η μηχανή μας.



Εικόνα 8-1 Χαρακτηριστικά εικονικής μηχανής

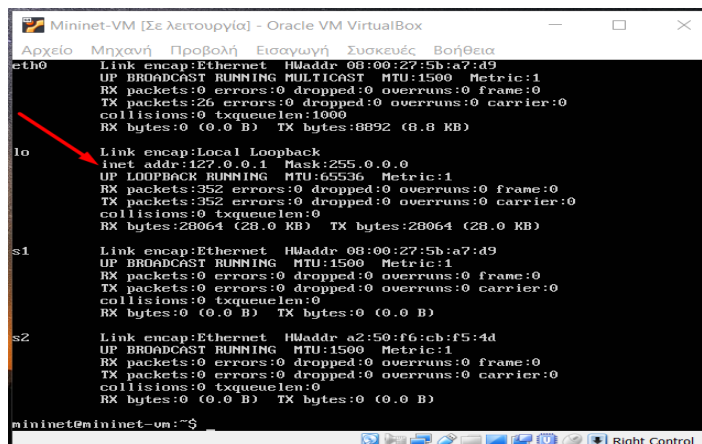
Στη συνέχεια όταν έχει ολοκληρωθεί η εγκατάσταση, μας εμφανίζεται στην κύρια οθόνη της μηχανής ένα παράθυρο για να εισάγουμε credentials για να πραγματοποιήσουμε την είσοδο στο Mininet.



Εικόνα 8-2 Είσοδος στο Mininet

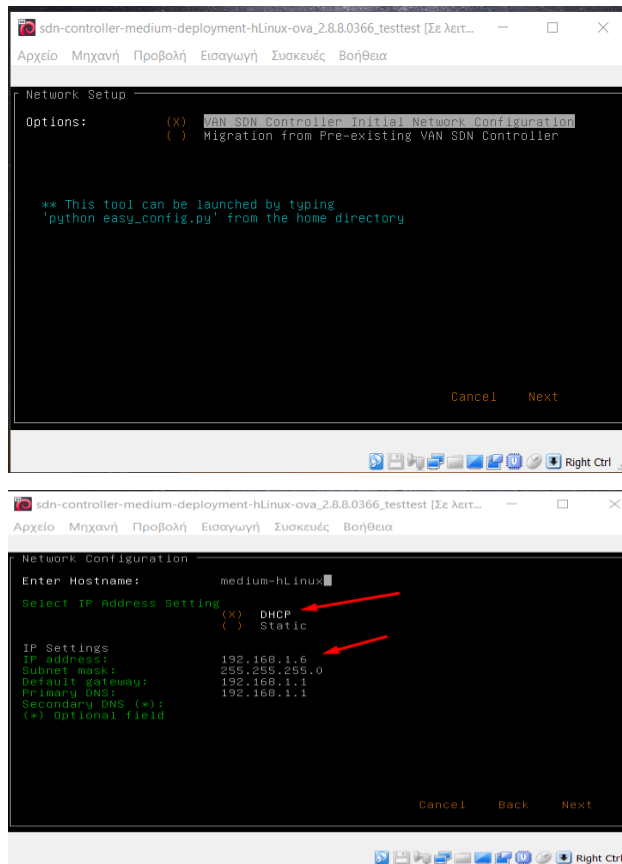
Πληκτρολογώντας για username:*mininet* και σαν password:*mininet* έχουμε επιτυχή είσοδο στον εξομοιωτή.

Εκτελώντας τώρα την εντολή **ifconfig** για να ελέγξουμε τα network interfaces βλέπουμε ότι προκαθορισμένα δεν έχει ανατεθεί κάποια IP διεύθυνση εκτός από το Loopback interface που έχει την localhost IP.



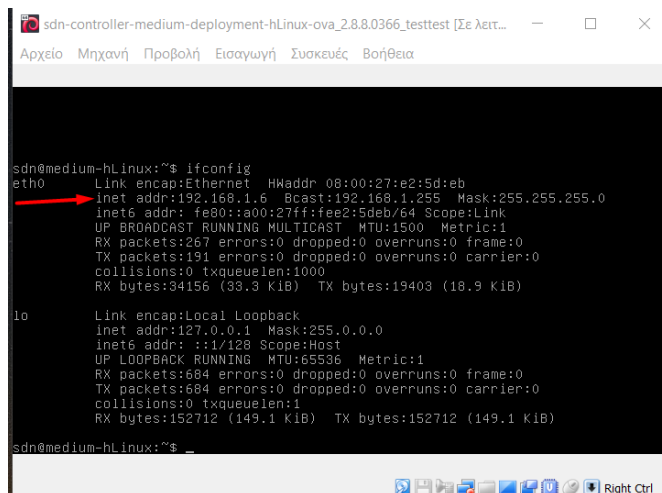
Εικόνα 8-3 Προβολή των διαθέσιμων interfaces

Έχοντας προβεί στην επιτυχή εγκατάσταση του Mininet συνεχίζουμε με τον ίδιο τρόπο για την εγκατάσταση και του ελεγκτή Aruba HP SDN Controller. Εφόσον έχουμε κάνει λήψη την precompiled εικονική μηχανή του SDN Controller και ακολουθώντας την αλληλουχία των παραπάνω βημάτων φτάνουμε στο σημείο που βάζοντας σαν username:*sdn* και password:*skyline* έχουμε πρόσβαση στο περιβάλλον της μηχανής. Στη συνέχεια θα χρειαστεί να κάνουμε παραμετροποίηση έτσι ώστε να πάρει με DHCP IP διεύθυνση ο ελεγκτής.



Εικόνα 8-4 Ανάθεση IP με DHCP πρωτόκολλο

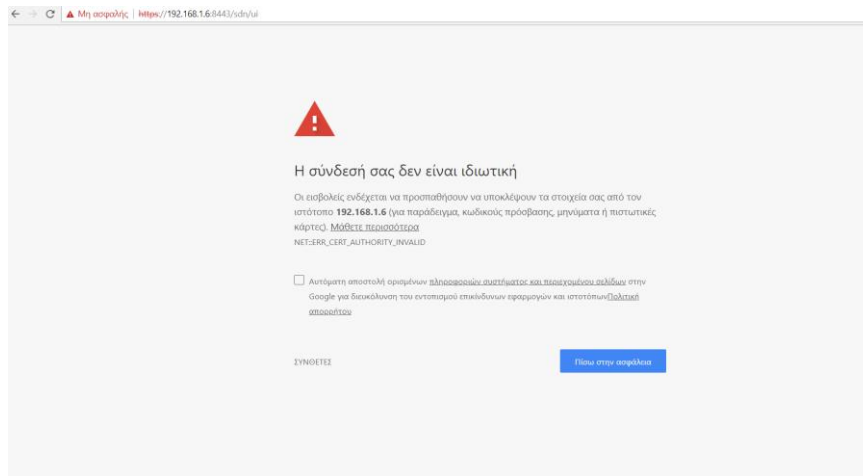
Βλέπουμε ότι η διεύθυνση της IP θα γίνει μέσω του DHCP πρωτοκόλλου και θα ανατεθεί διεύθυνση που ανήκει στο δικό μας τοπικό δίκτυο. Συνεχίζουμε γράφοντας την εντολή *ifconfig* και βλέπουμε ότι έχει ανατεθεί επιτυχώς IP στον SDN.



Εικόνα 8-5 IP του SDN Controller μέσω DHCP

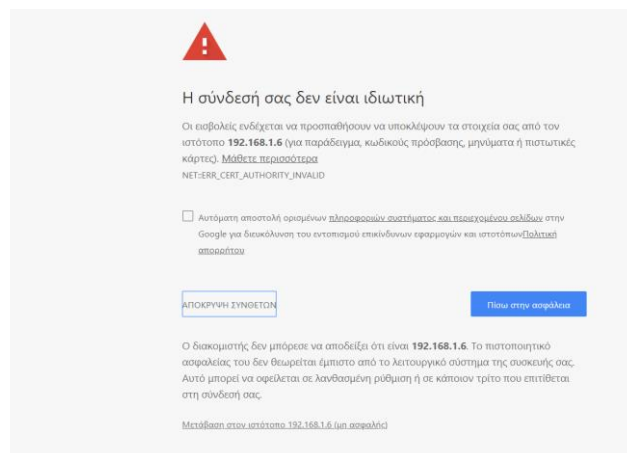


Για να δούμε την λειτουργία του ανοίγουμε ένα browser και πληκτρολογούμε την συγκεκριμένη διεύθυνση <https://192.168.1.12:8443/sdn/ui>

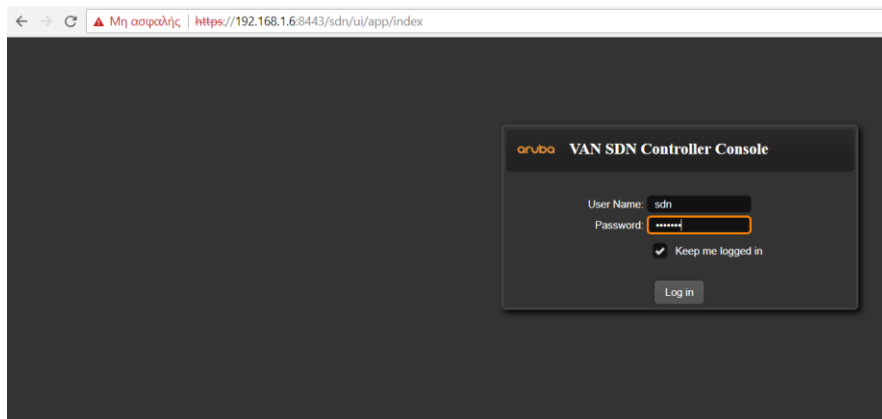


Εικόνα 8-6 Μετάβαση στον ιστότοπο του Aruba SDN Controller

Πατώντας ΣΥΝΘΕΤΕΣ και στην συνέχεια μετάβαση στον ιστότοπο θα μεταβούμε στο γραφικό περιβάλλον του ελεγκτή.

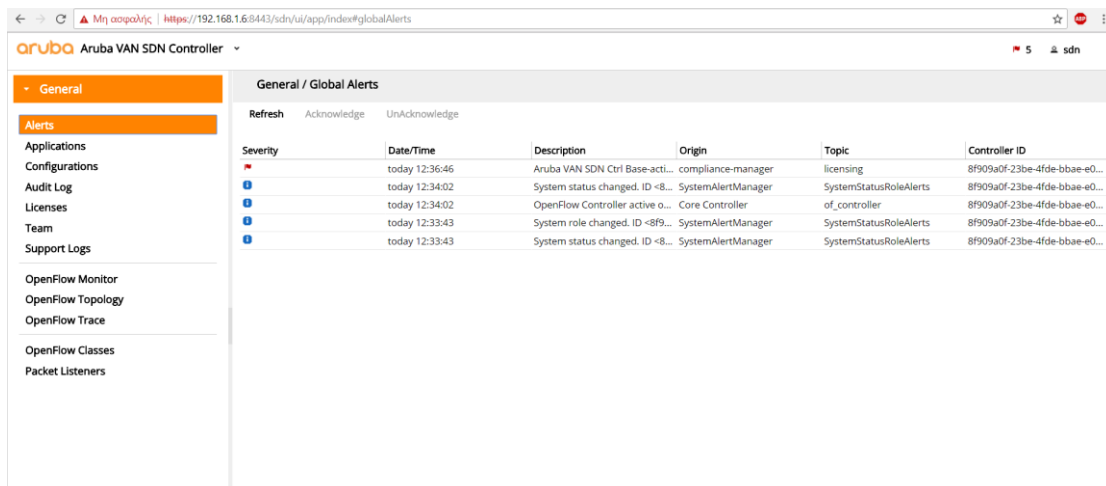


Εικόνα 8-7 Μετάβαση στον ιστότοπο του Aruba SDN Controller



Εικόνα 8-8 Είσοδος στο περιβάλλον του Aruba SDN Controller

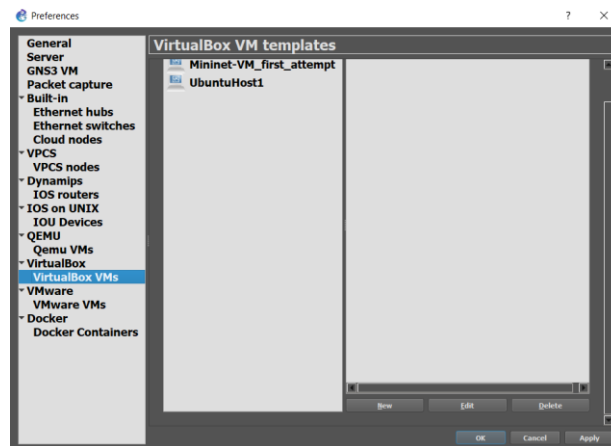
Εισάγοντας τα παραπάνω credentials όπου password: *skyline* βλέπουμε το main gui της web εφαρμογής, παρατηρώντας στα αριστερά μας ορισμένες καρτέλες όπως το OpenFlow Topology, Monitor, Trace τα οποία θα χρησιμοποιήσουμε παρακάτω για να έχουμε οπτική επαφή των τοπολογιών που θα δουλέψουμε στο Mininet.



Εικόνα 8-9 Γραφικό περιβάλλον του Aruba SDN Controller

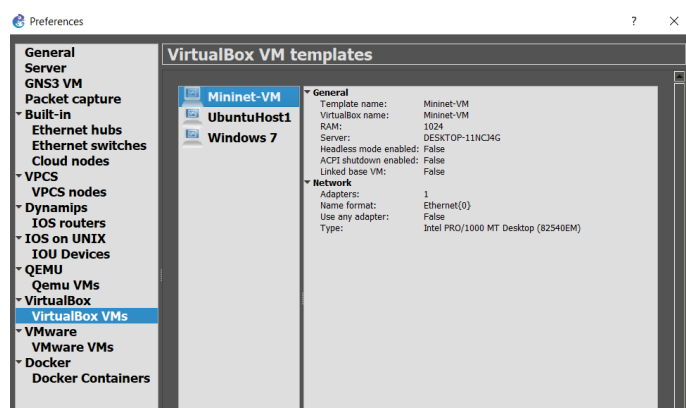
Στη συνέχεια δουλέψαμε την διασύνδεση και των τριών αυτών προσομοιωτών (GNS3, Mininet, Aruba HP Van SDN Controller). Ανοίγοντας το GNS3 δημιουργούμε ένα καινούργιο project το οποίο θα περιέχει και το MininetVM μέσα. Αυτό θα επιτευχθεί εκτελώντας τα παρακάτω βήματα.

Edit->Preferences->VirtualBox VMS->New επιλέγουμε από τη λίστα που μας εμφανίζεται το precompiled Mininet VM



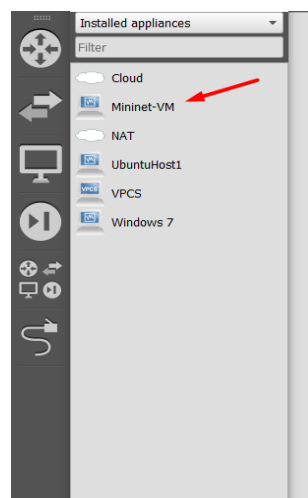
Εικόνα 8-10 Εισαγωγή Mininet στο GNS3

Βλέπουμε ότι το έχουμε προσθέσει με επιτυχία στο GNS3 και συνεχίζουμε για να ολοκληρώσουμε την παραμετροποίηση του.



Εικόνα 8-11 Εισαγωγή Mininet στο GNS3

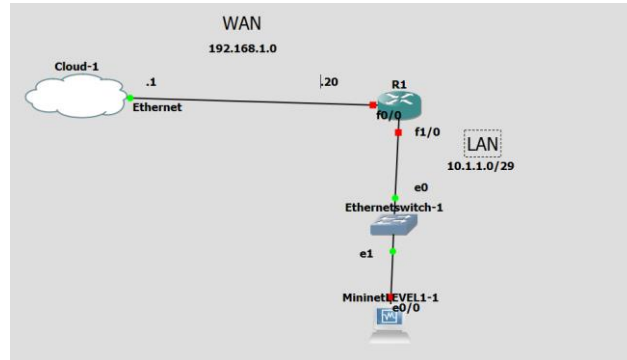
Όπως αναφέραμε και πιο πάνω έχουμε δημιουργήσει ένα νέο project και τώρα εισάγουμε το Mininet image μέσα σε αυτό κάνοντας το drag and drop από την παλέτα των end devices.



Εικόνα 8-12 Εισαγωγή Mininet στο GNS3



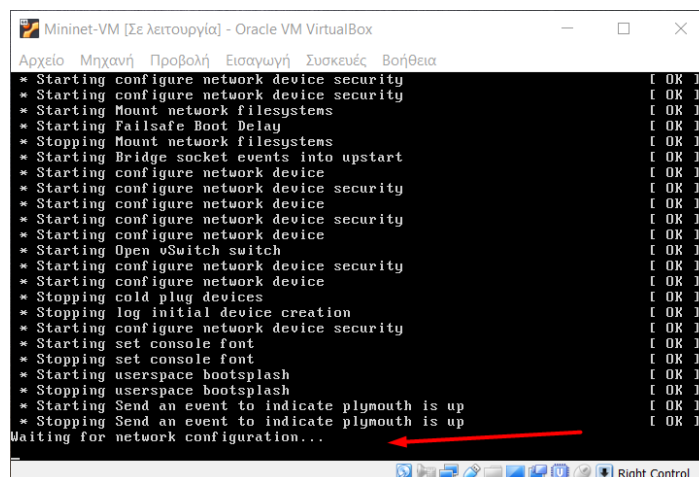
8.3 Δημιουργία πρώτης τοπολογίας διασύνδεσης με την χρήση του NAT πρωτοκόλλου



Εικόνα 8-13 Πρώτη τοπολογία GNS3-Mininet

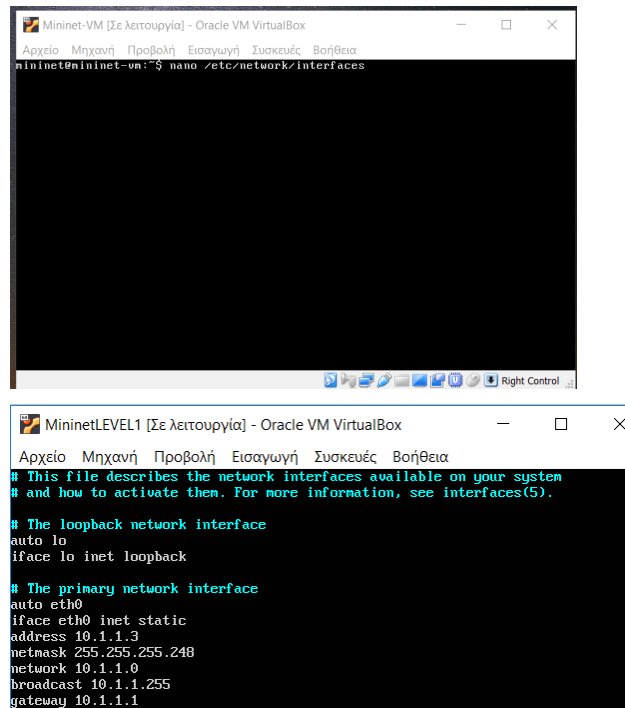
Στη συγκεκριμένη τοπολογία έχουμε έναν δρομολογητή (router3600), ένα cloud image το οποίο στη συνέχεια θα μας βοηθήσει στην επικοινωνία του Mininet με τον SDN Controller διότι απαιτείται σύνδεση στο διαδίκτυο για την επιτυχή υλοποίηση του σεναρίου.

Έχοντας εισάγει την εικονική μηχανή στο project κάνουμε start όλους τους κόμβους και πηγαίνουμε στο τερματικό του Mininet VM για ορισμένες παραμετροποιήσεις που χρειάζεται να γίνουν για την σωστή IP διεύθυνση. Αφού έχουμε ξεκινήσει το project το VM του Mininet ανοίγει από μόνο του αφού το έχουμε ενσωματώσει στο γραφικό περιβάλλον του GNS3. Καθώς το Mininet ανοίγει βλέπουμε ότι μας εμφανίζεται ένα μήνυμα που μας λέει ότι περιμένει ο εξομοιωτής να πάρει δίκτυο. Εμείς στη συνέχεια πρέπει να πάμε στο αρχείο interfaces του Mininet έτσι ώστε να βάλουμε στατικά IP διεύθυνση στο interface eth0 του Mininet έτσι ώστε να μπορεί να επικοινωνήσει με τον δρομολογητή



Εικόνα 8-14 Πρόβλημα με την απόδοση IP

Αυτό θα το κάνουμε ακολουθώντας την παρακάτω διαδρομή με την συγκεκριμένη εντολή, αφότου έχουμε κάνει login.



Εικόνα 8-15 Interfaces path

Και γράφουμε στο eth0 interface τον παρακάτω κώδικα δίνοντας την IP 10.1.1.3 που στο σενάριο μας είναι το τοπικό δίκτυο LAN και προεπιλεγμένη πύλη 10.1.1.1 του δρομολογητή R1

auto eth0

iface eth0 inet static

address 10.1.1.3

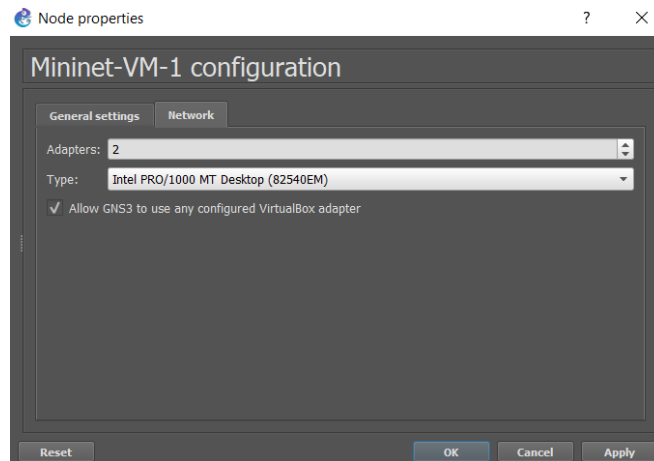
netmask 255.255.255.248

network 10.1.1.0

broadcast 10.1.1.255

gateway 10.1.1.1

Έπειτα πηγαίνοντας στο project που έχουμε στο GNS3 πατάμε δεξί κλικ στο image του MininetVM->Configure->Network και προσθέτουμε adapters έτσι ώστε να συνδεθούμε με τον δρομολογητή R1, επιλέγοντας *Allow* έτσι ώστε το GNS3 να έχει πρόσβαση στον adapter που δουλεύει η εικονική μηχανή του Mininet. Σε αυτό το σημείο το Mininet έχει IP και μπορεί να συνδεθεί με τον R1 στο τοπικό δίκτυο 10.1.1.0.



Εικόνα 8-16 Προσθήκη adapters στο Mininet

Αυτό που θα κάνουμε σε αυτό το σημείο είναι να ασχοληθούμε με το LAN δίκτυο και στη συνέχεια με το WAN δίκτυο που θα μας χρησιμεύσει στη σύνδεση με το διαδίκτυο και κατ' επέκταση με τον SDN ελεγκτή που αναφέραμε και πιο πάνω.

Ανοίγουμε τον δρομολογητή R1 και ξεκινάμε την παραμετροποίησή του, δίνοντας IP στο LAN δίκτυο **10.1.1.0/29**

Router1(3600)

conf t

int f1/0

ip add 10.1.1.1 255.255.255.248

no shut

exit

wr



Πηγαίνουμε στο περιβάλλον του Mininet και βλέπουμε ότι έχει καταχωρηθεί επιτυχώς IP διεύθυνση.

```
mininet@mininet-um:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:7e:6c:ee
          inet addr:10.1.1.3  Bcast:10.1.1.255  Mask:255.255.255.248
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:288 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:17280 (17.2 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:123 errors:0 dropped:0 overruns:0 frame:0
          TX packets:123 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11349 (11.3 KB)  TX bytes:11349 (11.3 KB)

mininet@mininet-um:~$
```

Εικόνα 8-17 IPv4 Mininet-eth0

Ping από το περιβάλλον του Mininet στο δρομολογητή R1 στο f1/0 και αντίστροφα

```
FastEthernet1/0    10.1.1.1    YES NVRAM  up        up
NV10               unassigned  NO  unset   up        up
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#ping 10.1.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/12/16 ms
R1#
```

```
MininetLEVEL1 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
mininet@mininet-um:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:7e:6c:ee
          inet addr:10.1.1.3  Bcast:10.1.1.255  Mask:255.255.255.248
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:288 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:17280 (17.2 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:123 errors:0 dropped:0 overruns:0 frame:0
          TX packets:123 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11349 (11.3 KB)  TX bytes:11349 (11.3 KB)

mininet@mininet-um:~$ ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data:
64 bytes from 10.1.1.1: icmp_seq=2 ttl=255 time=0.95 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=255 time=7.03 ms
64 bytes from 10.1.1.1: icmp_seq=4 ttl=255 time=5.00 ms
64 bytes from 10.1.1.1: icmp_seq=5 ttl=255 time=13.3 ms
64 bytes from 10.1.1.1: icmp_seq=6 ttl=255 time=11.9 ms
64 bytes from 10.1.1.1: icmp_seq=7 ttl=255 time=9.90 ms
^C
--- 10.1.1.1 ping statistics ---
 7 packets transmitted, 6 received, 14% packet loss, time 6017ms
 rtt min/avg/max/ndev = 5.007/9.375/13.388/2.825 ms
mininet@mininet-um:~$
```

Εικόνα 8-18 Επικοινωνία Router 1 -> Mininet

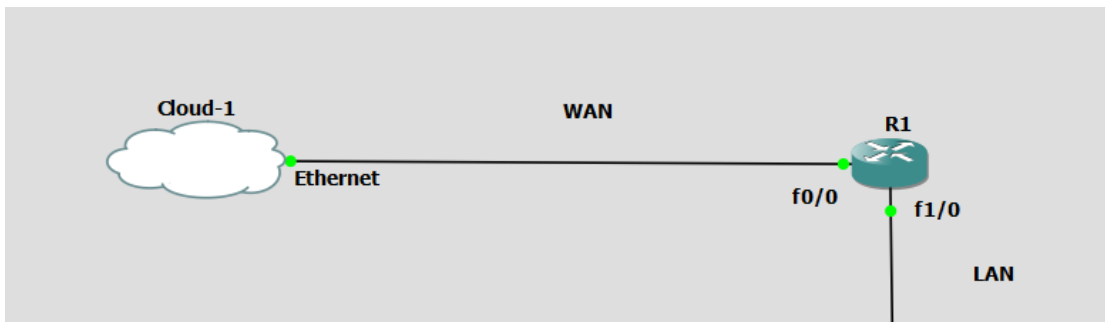
Βλέπουμε ότι η επικοινωνία μέσω ping από το Mininet στην IP 10.1.1.1 του Router1 και από τον Router1 στην IP του Mininet 10.1.1.3 δουλεύουν επιτυχώς έχουμε εκατό τοις εκατό μετάδοση πακέτων.

Μετά την επιτυχή επικοινωνία του Router1 του GNS3 και του Mininet θα πρέπει να στήσουμε στη συνέχεια και το WAN δίκτυο. Στήνοντας το WAN δίκτυο θα έχουμε την δυνατότητα διασύνδεσης μεταξύ του Mininet και του ελεγκτή Aruba HP VAN SDN Controller στον οποίο έχουμε πρόσβαση στην IP <https://192.168.1.12:8443/sdn/ui/app/index>. Έπειτα θα έχουμε την δυνατότητα να δούμε τις τοπολογίες που θα έχουμε δημιουργήσει στο Mininet, τους διάφορους host που θα χρησιμοποιήσουμε και γενικότερα την κίνηση του εικονικού δικτύου.



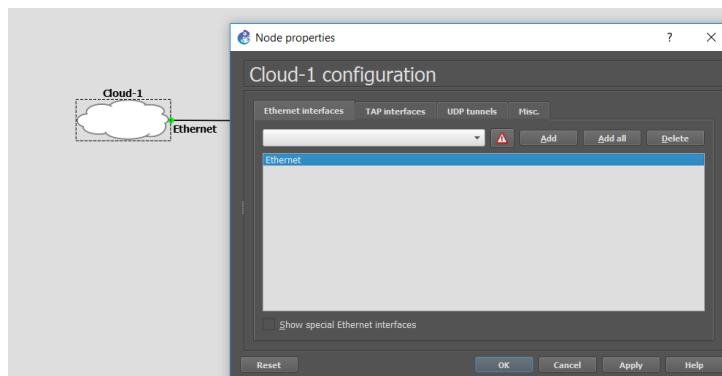
Για να στήσουμε το WAN δίκτυο θα πρέπει να προσθέσουμε στο project μας το Cloud IOS image έτσι ώστε να έχουμε πρόσβαση στο διαδίκτυο. Αυτό θα γίνει κατεβάζοντας το image cloud από την σελίδα του GNS3 και προσθέτοντάς το στο γραφικό περιβάλλον της πλατφόρμας μετα εξής απλά βήματα.

Edit->Preferences->Cloud nodes->New και επιλέγουμε το image που έχουμε κατεβάσει. Αφότου έχουμε προσθέσει το Cloud Node το κάνουμε drag and drop στο πρότζεκτ μας και το συνδέουμε με το δρομολογητή μας R1(3600).



Εικόνα 8-19 Προσθήκη Cloud

Πηγαίνουμε τώρα στο τερματικό του Cloud κάνοντας *δεξί κλικ->Configure* επιλέγουμε στα Ethernet interfaces την κάρτα δικτύου ethernet και την κάνουμε add στον κόμβο μας. Έχουμε επιλέξει την κάρτα ethernet έτσι ώστε να παίρνει την IP του τοπικού μας δικτύου.



Εικόνα 8-20 Παραμετροποίηση Cloud - Προσθήκη Ethernet

Στη συνέχεια πηγαίνουμε στον δρομολογητή μας και τον παραμετροποιούμε έτσι ώστε να παίρνει στατικά IP διεύθυνση, του Ethernet δικτύου(192.168.1.0). Αυτό θα το επιτύχουμε με τις παρακάτω εντολές όπου θα δώσουμε IP στον δρομολογητή μας την 192.168.1.20 και θα διαφημίσουμε την default gateway του δικτύου Ethernet.

Router1(3600)

```
conf t
```

```
int fa0/0
```

```
ip add 192.168.1.20 255.255.255.0
```

```
no shut
```



ip domain-lookup

exit

exit

wr

```
Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : Home
Link-local IPv6 Address . . . . . : fe80::605d:5ce4:1409:a267%19
IPv4 Address. . . . . : 192.168.1.3
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1
```

Εικόνα 8-21 Default gateway Ethernet δικτύου

Έχοντας κάνει σωστά την συνδεσμολογία ελέγχουμε την λειτουργία της επικοινωνίας του δρομολογητή μας με το διαδίκτυο κάνοντας ping στη google και στη cisco παίρνοντας 100% επιτυχή μετάδοση πακέτων με την πρώτη προσπάθεια στη google και στη cisco.

```
R1#ping google.com
Translating "google.com"...domain server (8.8.8.8) [OK]
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 216.58.205.142, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/84/84 ms
R1#ping cisco.com
Translating "cisco.com"...domain server (8.8.8.8) [OK]
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 72.163.4.185, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 176/178/184 ms
R1#
```

Εικόνα 8-22 Ping στη google και στην cisco

Αυτό που θα υλοποιήσουμε στη συνέχεια για να επιτύχουμε την διασύνδεση των τριών προγραμμάτων είναι η εφαρμογή του NAT πρωτοκόλλου στο WAN και LAN δίκτυο μας. Χρησιμοποιώντας στον δρομολογητή μας την παρακάτω ακολουθία εντολών θα επιτύχουμε την επικοινωνία του WAN και LAN δικτύου μας. Θα πρέπει να διαφημίσουμε στο υπόλοιπο δίκτυο μας στατικά την default gateway του δικτύου Ethernet.

Router1(3600)

conf t

ip route 0.0.0.0 0.0.0.0 192.168.1.1

exit

conf t

int f0/0



```
ip nat outside /* Θέτουμε την f0/0 σαν public gateway*/
```

```
exit
```

```
int f1/0
```

```
ip nat inside /* Θέτουμε την f1/0 σαν εσωτερική gateway*/
```

```
exit
```

```
access-list 1 permit 192.168.1.0 0.0.0.7 /* πρόσβαση στο διαδίκτυο θα έχουν μόνο 7 συσκευές*/
```

```
ip nat inside source list 1 int f0/0 overload
```

Εφόσον έχουν εκτελεστεί σωστά οι παραπάνω εντολές τώρα θα πρέπει να πάμε στο περιβάλλον του Mininet και να ορίσουμε DNS-Server 8.8.8.8(αυτό της google) που είναι αρκετά αξιόπιστο έτσι ώστε να έχουμε πρόσβαση στο διαδίκτυο.

```
MininetLEVEL1 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 10.1.1.3
netmask 255.255.255.248
network 10.1.1.0
broadcast 10.1.1.255
gateway 10.1.1.1
dns-nameservers 8.8.8.8
```

Εικόνα 8-23 Προσθήκη DNS-server στο Mininet

Εφόσον έχουμε εκτελέσει σωστά τις παραπάνω ενέργειες εξετάζουμε αν λειτουργεί σωστά η διαφήμιση(route) από το LAN δηλαδή το Mininet στο WAN δίκτυο στο interface f0/0. Παρατηρούμε ότι μεταδίδονται με επιτυχία τα 6 πακέτα που στέλνει στο Mininet προς την IP του WAN δικτύου του δρομολογητή μας χωρίς καμία απώλεια και σε πολύ μικρό χρονικό διάστημα.

```
mininet@mininet-vm:~$ ping 192.168.1.20
PING 192.168.1.20 (192.168.1.20) 56(84) bytes of data:
64 bytes from 192.168.1.20: icmp_seq=1 ttl=255 time=9.71 ms
64 bytes from 192.168.1.20: icmp_seq=2 ttl=255 time=7.35 ms
64 bytes from 192.168.1.20: icmp_seq=3 ttl=255 time=5.30 ms
64 bytes from 192.168.1.20: icmp_seq=4 ttl=255 time=14.7 ms
64 bytes from 192.168.1.20: icmp_seq=5 ttl=255 time=13.3 ms
64 bytes from 192.168.1.20: icmp_seq=6 ttl=255 time=12.9 ms
^C
--- 192.168.1.20 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 5.309/10.567/14.776/3.408 ms
mininet@mininet-vm:~$
```

Εικόνα 8-24 Επικοινωνία LAN - WAN

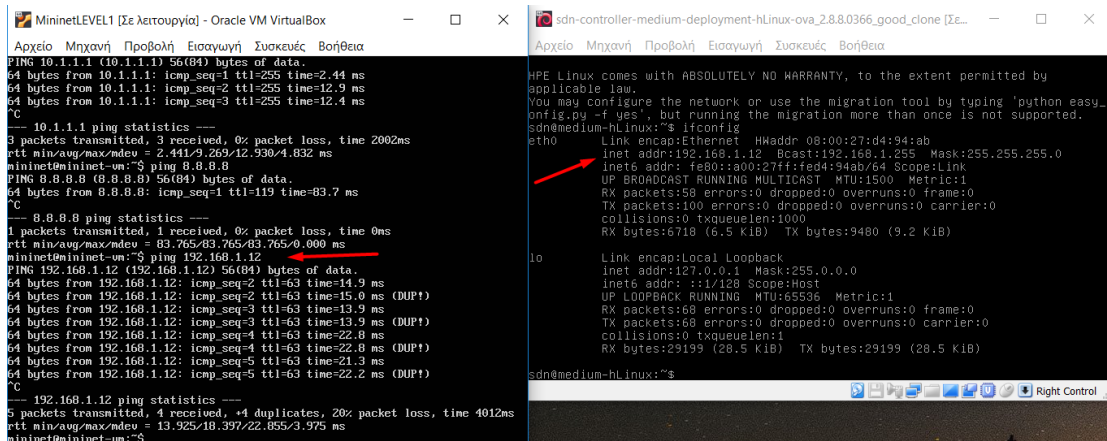
Εν συνεχεία θα ελέγξουμε τη διασύνδεση του Mininet με το διαδίκτυο κάνοντας ping στο server της google ο οποίος έχει IP 8.8.8.8.



```
mininet@mininet-vm:~$ ping google.com
PING google.com (216.58.208.46) 56(84) bytes of data:
64 bytes from fra15s12-in-f46.1e100.net (216.58.208.46): icmp_seq=1 ttl=53 time=82.5 ms
64 bytes from fra15s12-in-f46.1e100.net (216.58.208.46): icmp_seq=2 ttl=53 time=81.3 ms
64 bytes from fra15s12-in-f46.1e100.net (216.58.208.46): icmp_seq=3 ttl=53 time=80.8 ms
64 bytes from fra15s12-in-f46.1e100.net (216.58.208.46): icmp_seq=4 ttl=53 time=70.4 ms
64 bytes from fra15s12-in-f46.1e100.net (216.58.208.46): icmp_seq=5 ttl=53 time=79.3 ms
64 bytes from fra15s12-in-f46.1e100.net (216.58.208.46): icmp_seq=6 ttl=53 time=78.2 ms
64 bytes from fra15s12-in-f46.1e100.net (216.58.208.46): icmp_seq=7 ttl=53 time=76.9 ms
^C
--- google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6008ms
rtt min/avg/max/mdev = 70.413/78.528/82.566/3.759 ms
mininet@mininet-vm:~$
```

Εικόνα 8-25 Επικοινωνία Mininet - google.com

Βλέποντας ότι το Mininet επικοινωνεί με το διαδίκτυο τώρα θα πάμε να ελέγξουμε αν επικοινωνεί με τον SDN Controller, που τρέχει σε άλλη εικονική μηχανή όπως έχουμε αναφέρει και πιο πάνω και παίρνει IP του τοπικού δικτύου μας με DHCP πρωτόκολλο. Παρατηρούμε ότι τα πακέτα στέλνονται επιτυχώς και τα τρία προγράμματα είναι διασυνδεδεμένα.



Εικόνα 8-26 Επιτυχής διασύνδεση και των τριών πλατφορμών

Πιο κάτω θα προχωρήσουμε στην χρήση και τη λειτουργία του Mininet εξομοιωτή χρησιμοποιώντας και τα τρία εργαλεία μαζί. Πηγαίνοντας τώρα στο τερματικό του Mininet θα δούμε την λειτουργία του χρησιμοποιώντας ορισμένες εντολές που θα δημιουργήσουν δύο ξεχωριστούς hosts οι οποίοι θα έχουν από ένα μεταγωγέα ο καθένας.

```
mininet@mininet-vm:sudo mn --controller=remote,ip=192.168.1.12 --topo=linear,2  
--switch=ovsk,protocols=OpenFlow13 --mac
```

Με αυτή την εντολή δημιουργείται η τοπολογία με τους hosts και τους μεταγωγείς χρησιμοποιώντας σαν απομακρυσμένη διεύθυνση αυτή του ελεγκτή SDN Controller για να έχουμε τα αποτελέσματα σε γραφικό περιβάλλον, σαν IP διεύθυνση βάσης αυτή του LAN δικτύου, επιλέγοντας γραμμική τοπολογία και χρησιμοποιώντας το OpenFlow πρωτόκολλο που υλοποιεί ο Aruba Controller. Πιο κάτω βλέπουμε ότι το Mininet δημιουργεί επιτυχώς το δίκτυο συνδέεται με τον remote ελεγκτή προσθέτει τους hosts και τα switches και δημιουργεί τις συνδέσεις μεταξύ h1-s1 και h2-s2.



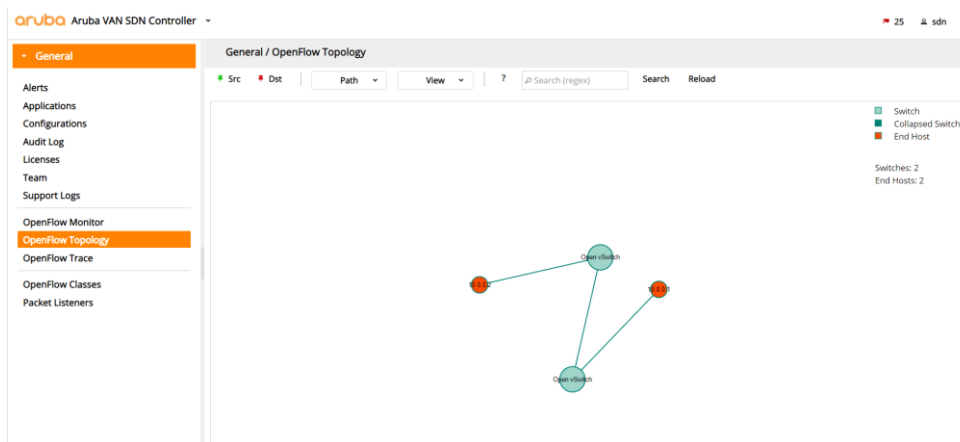
```
*** Creating network
*** Adding controller
Unable to contact the remote controller at 192.168.1.12:6653
Connecting to remote controller at 192.168.1.12:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s2) (s2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet>
```

Εικόνα 8-27 Δημιουργία τοπολογίας

Εκτελώντας την εντολή `pingall` βλέπουμε ότι έχουμε πλήρη επικοινωνία μεταξύ των host `h1`, `h2` και πηγαίνοντας στο γραφικό περιβάλλον της σελίδας του SDN Controller έχουμε το αποτέλεσμα της τοπολογίας που τρέξαμε.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
```

Εικόνα 8-28 Pings των hosts



Εικόνα 8-29 Τοπολογία στον Aruba Controller

Στη συνέχεια αυτό που θέλουμε να κάνουμε είναι και οι hosts του Mininet να επικοινωνούν με το διαδίκτυο οπότε θα πρέπει να πάμε να ενεργοποιήσουμε το `nat` και στους host του Mininet. Αυτό θα το επιτύχουμε προσθέτοντας στο script όπου υπάρχουν τα interfaces την παρακάτω εντολή.



```
MininetLEVEL1 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 10.1.1.3
netmask 255.255.255.248
network 10.1.1.0
broadcast 10.1.1.255
gateway 10.1.1.1
dns-nameservers 8.8.8.8

iface nat0-eth0 inet manual
```

Εικόνα 8-30 Προσθήκη NAT στο interface eth0 του Mininet

Τρέχουμε ξανά στην προηγούμενη τοπολογία στο Mininet με την προσθήκη του NAT
`sudo mn --controller=remote,ip=192.168.1.12 --topo=linear,2`
`--switch=ovsk,protocols=OpenFlow13 --mac --nat`

```
Connecting to remote controller at 192.168.1.12:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s2) (s2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 nat0
h2 -> h1 nat0
nat0 -> h1 h2
*** Results: 0% dropped (6/6 received)
```

Εικόνα 8-31 Προσθήκη NAT πρωτοκόλλου

Με την προσθήκη του nat στο eth0 ουσιαστικά προστέθηκε ένας ακόμη host που είναι υπεύθυνος για την λειτουργία του NAT πρωτοκόλλου μέσα στο περιβάλλον του Mininet με τον οποίο μπορούν και επικοινωνούν οι άλλοι δύο hosts. Έπειτα δοκιμάζουμε ping προς το διαδίκτυο και από τους δύο host και βλέπουμε ότι είμαστε πλήρως συνδεδεμένοι καθώς ο h1 επικοινωνεί με την google και ο h2 με την cisco.



```
mininet> h1 ping -c 3 google.com
PING google.com (172.217.16.206) 56(84) bytes of data.
64 bytes from fra16s08-in-f206.1e100.net (172.217.16.206): icmp_seq=1 ttl=52 time=82.5 ms
64 bytes from fra16s08-in-f206.1e100.net (172.217.16.206): icmp_seq=2 ttl=52 time=90.8 ms
64 bytes from fra16s08-in-f206.1e100.net (172.217.16.206): icmp_seq=3 ttl=52 time=89.8 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 82.579/87.780/90.878/3.700 ms
mininet> h2 ping -c 3 cisco.com
PING cisco.com (72.163.4.185) 56(84) bytes of data.
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=1 ttl=242 time=186 ms
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=2 ttl=242 time=185 ms
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=3 ttl=242 time=185 ms

--- cisco.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 185.122/186.004/186.900/0.725 ms
```

Εικόνα 8-32 Επιτυχής επικοινωνία hosts με το διαδίκτυο

Ουσιαστικά καταφέραμε να αποκρύψουμε ένα ολόκληρο δίκτυο πίσω από μία public IP διεύθυνση που είναι προσβάσιμη από όλους, ενώ το εσωτερικό δίκτυο παραμένει στην αφάνεια. Για την επιβεβαίωση αυτού του ισχυρισμού θα ανοίξουμε το Wireshark και θα κάνουμε capture την κίνηση που περνά από το Router 1 στο f0/0 που δίνει πρόσβαση με το διαδίκτυο και θα στείλουμε πακέτα προς την cisco από τον h1 που έχει IP 10.0.0.1

```
mininet> h1 ping -c 5 cisco.com
PING cisco.com (72.163.4.185) 56(84) bytes of data.
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=1 ttl=242 time=186 ms
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=2 ttl=242 time=185 ms
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=3 ttl=242 time=184 ms
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=4 ttl=242 time=195 ms
64 bytes from redirect-ns.cisco.com (72.163.4.185): icmp_seq=5 ttl=242 time=194 ms

--- cisco.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 184.553/189.546/195.471/4.709 ms
mininet>
```

Εικόνα 8-33 Αποστολή πακέτων προς την Cisco

*Standard input [Cloud-1 Ethernet to R1 FastEthernet0/0]

| No. | icmpv6 | Source | Destination | Protocol | Length | Info |
|-----|-----------|--------------|--------------|----------|--------|---|
| 245 | 37.401442 | 192.168.1.20 | 72.163.4.185 | ICMP | 98 | Echo (ping) request id=0x0c67, seq=1/256, ttl=62 (reply in 246) |
| 246 | 37.572659 | 72.163.4.185 | 192.168.1.20 | ICMP | 98 | Echo (ping) reply id=0x0c67, seq=1/256, ttl=244 (request in 245) |
| 270 | 38.401902 | 192.168.1.20 | 72.163.4.185 | ICMP | 98 | Echo (ping) request id=0x0c67, seq=2/512, ttl=62 (reply in 271) |
| 271 | 38.571502 | 72.163.4.185 | 192.168.1.20 | ICMP | 98 | Echo (ping) reply id=0x0c67, seq=2/512, ttl=244 (request in 270) |
| 272 | 39.402405 | 192.168.1.20 | 72.163.4.185 | ICMP | 98 | Echo (ping) request id=0x0c67, seq=3/768, ttl=62 (reply in 273) |
| 273 | 39.572697 | 72.163.4.185 | 192.168.1.20 | ICMP | 98 | Echo (ping) reply id=0x0c67, seq=3/768, ttl=244 (request in 272) |
| 274 | 40.404248 | 192.168.1.20 | 72.163.4.185 | ICMP | 98 | Echo (ping) request id=0x0c67, seq=4/1024, ttl=62 (reply in 276) |
| 276 | 40.581605 | 72.163.4.185 | 192.168.1.20 | ICMP | 98 | Echo (ping) reply id=0x0c67, seq=4/1024, ttl=244 (request in 274) |
| 277 | 41.404680 | 192.168.1.20 | 72.163.4.185 | ICMP | 98 | Echo (ping) request id=0x0c67, seq=5/1280, ttl=62 (reply in 278) |
| 278 | 41.580077 | 72.163.4.185 | 192.168.1.20 | ICMP | 98 | Echo (ping) reply id=0x0c67, seq=5/1280, ttl=244 (request in 277) |

Εικόνα 8-34 Κίνηση που παρατηρείται στο Wireshark

Από την εικόνα 8-34 παρατηρούμε ότι καθώς στέλνουμε πακέτα από την IP του h1=10.0.0.1 προς την cisco φαίνεται μονάχα η δημόσια IP του R1 (192.168.1.20) που έχουμε ορίσει ως NAT εξωτερικό και όχι το υπόλοιπο εσωτερικό δίκτυο που υπάρχει στην τοπολογία συμπεραίνοντας ότι το πρωτόκολλο λειτουργεί με επιτυχία.



8.4 Υλοποίηση Δεύτερη Τοπολογίας Διασύνδεσης μεταξύ GNS3-Mininet-OpenFlowController

Σε αυτό το σενάριο που θα δουλέψουμε θα χρησιμοποιήσουμε πάλι το precompiled Mininet-VM που έχουμε κατεβάσει και τον OpenFlow controller που χρησιμοποιήσαμε και στην παραπάνω συνπροσομοίωση. Θα πρέπει να δημιουργήσω ξανά δύο καινούργια μηχανήματα ένα Mininet και ένα controller δεν πρέπει να χρησιμοποιήσω τα ίδια διότι χρησιμοποιούν δεδομένα της προηγούμενης τοπολογίας και θα υπάρξουν πολλά bugs. Θα ονομάσω τα καινούργια machines

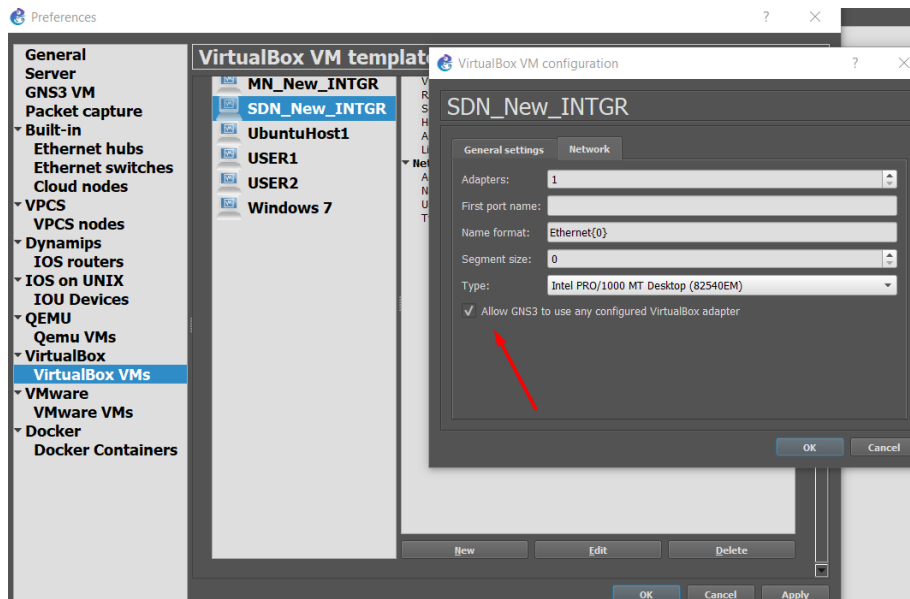


Εικόνα 8-35 Νέα ονόματα pre-built machines

Στο σενάριο αυτό θα δουλέψουμε και τα δύο machines μέσα στο GNS3 ενώ πριν είχαμε δουλέψει το Mininet μέσα και τον SDN έξω και τους κάναμε να επικοινωνούν διαμέσων πρωτοκόλλου NAT.

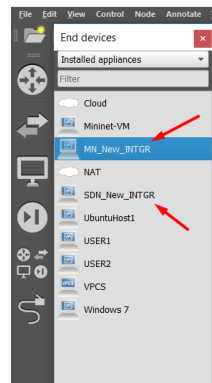
8.4.1 Βήματα Integrate

Στο περιβάλλον του GNS3 πηγαίνουμε και εισάγουμε τις εικονικές μας μηχανές με τα βήματα που έχουμε αναφέρει και στο προηγούμενο σενάριο τα αναφέρουμε σε πιο γρήγορο ρυθμό. Στο γραφικό περιβάλλον του GNS3 πηγαίνουμε Edit->Preferences->VirtualBoxVMS->New και επιλέγουμε το ονα file για το κάθε VM που έχουμε κατεβάσει και το έχουμε κάνει save σε ένα φάκελο στον υπολογιστή, κάνουμε browse το επιλέγουμε και πατάμε OK. Όταν το έχουμε δημιουργήσει πατάμε Edit πάνω του και στην καρτέλα Network κάνουμε επιλογή το παρακάτω. Αυτό το κάνουμε και για το Mininet και για τον SDN.



Εικόνα 8-36 Διασύνδεση εξομοιωτών με το GNS3

Όταν τελειώσουμε αυτή την διαδικασία πηγαίνοντας στην καρτέλα Browse end Devices βλέπουμε τα νέα machines που εγκαταστήσαμε. Ένα από τα βασικά χαρακτηριστικά του GNS3 open source platform είναι ότι επιτρέπει στον χρήστη να προσθέτει και να παραμετροποιεί εργαλεία και νέα χαρακτηριστικά.



Εικόνα 8-37 Προσθήκη Mininet και SDN controller

Στη συνέχεια τα κάνουμε drop στον καμβά και τα ενεργοποιούμε πατώντας start στο GNS3. Αυτό το πράγμα που πρέπει να κάνουμε και στις δύο precompiled εικονικές μηχανές είναι να πάμε και να γράψουμε ένα μικρό script έτσι ώστε να παίρνουν IP με DHCP. Όταν κάνουν boot up οι μηχανές μας συνδεόμαστε στο Mininet βάζοντας τα απαραίτητα credentials.

Όνομα χρήστη: mininet

Κωδικός πρόσβασης: mininet

Έχοντας συνδεθεί όπως έχουμε αναφέρει και πιο πάνω πάμε και βρίσκουμε το αρχείο με τα network interfaces του Mininet και γράφουμε το παρακάτω script ακολουθώντας αυτό το path

nano/etc/network/interfaces



```
MN_New_INTGR [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
GNU nano 2.2.6 File: /etc/network/interfaces

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

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
```

Εικόνα 8-38 Απόδοση IP διεύθυνσης με DHCP

Λέμε δηλαδή στο Mininet στο interface eth0 να περιμένει να πάρει IP με DHCP πρωτόκολλο. Την ίδια δουλειά πρέπει να κάνουμε αυτή τη φορά και στον SDN αφού θα τρέχει στο ίδιο πρότζεκτ με το Mininet και όχι από έξω όπως στο προηγούμενο Intergration.

Έχοντας γίνει boot up και ο SDN βάζοντας τα απαραίτητα credentials συνδεόμαστε.

Όνομα χρήστη: *sdn*

Κωδικός πρόσβασης: *skyline*

Έχοντας συνδεθεί όπως έχουμε αναφέρει και πιο πάνω πάμε και βρίσκουμε το αρχείο με τα network interfaces του SDN και γράφουμε το παρακάτω script ακολουθώντας αυτό το path

nano/etc/network/interfaces

```
SDN_New_INTGR [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
GNU nano 2.2.6 File: /etc/network/interfaces

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

# The loopback network interface
auto lo
iface lo inet loopback

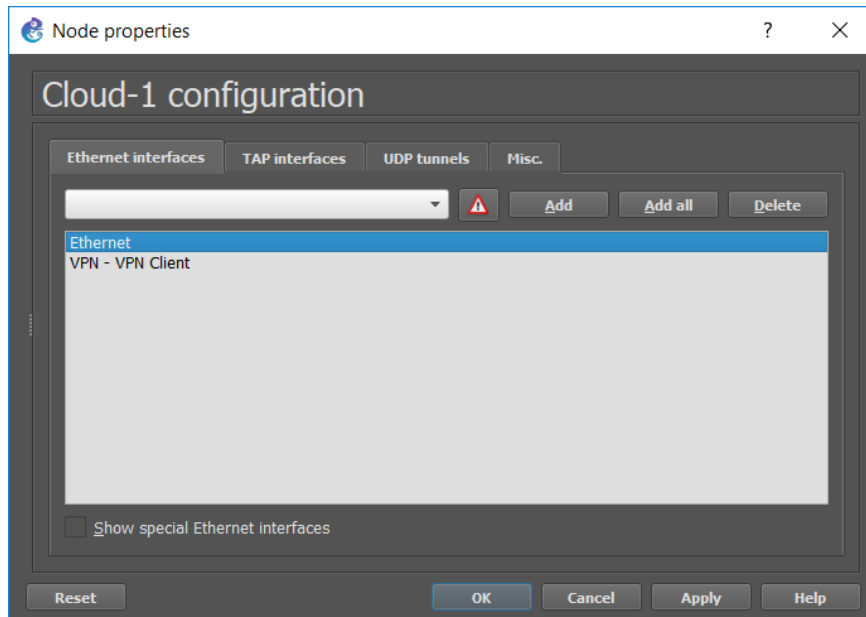
# The sw-mgmt network interface
# auto eth1
# iface eth1 inet dhcp
#Configured from installation
#The mgmt network interfaces
auto eth0
iface eth0 inet dhcp
```

Εικόνα 8-39 Απόδοση IP διεύθυνσης με DHCP

Αφού έχουμε εκτελέσει αυτό το script και στις δύο μηχανές περιμένουν να πάρουν IP διεύθυνση. Το επόμενο βήμα που πρέπει να υλοποιήσουμε είναι να δώσουμε πρόσβαση στο τοπικό μας



δίκτυο μέσω του Cloud image. Κάνουμε drag and drop το cloud στον καμβά μας κάνουμε add to ethernet.

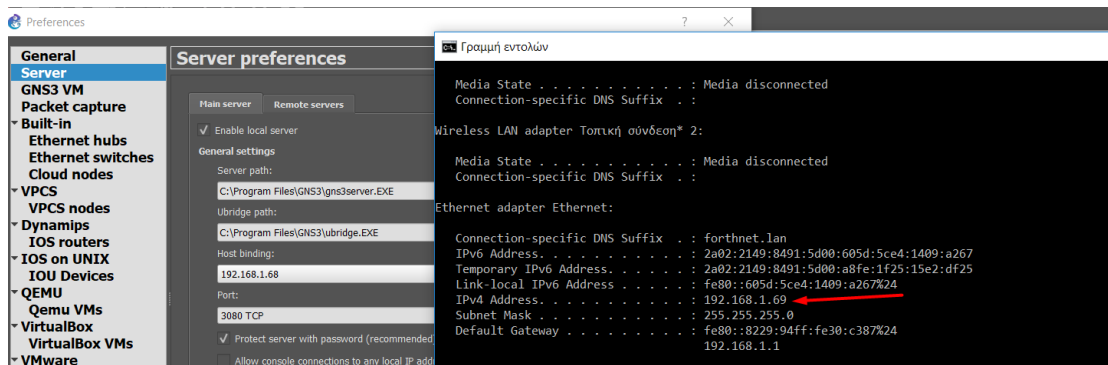


Εικόνα 8-40 Προσθήκη Ethernet

Για να επιτευχθεί η διασύνδεση μας θα πρέπει να είμαστε συνδεδεμένοι στο διαδίκτυο με καλώδιο και όχι με Wifi

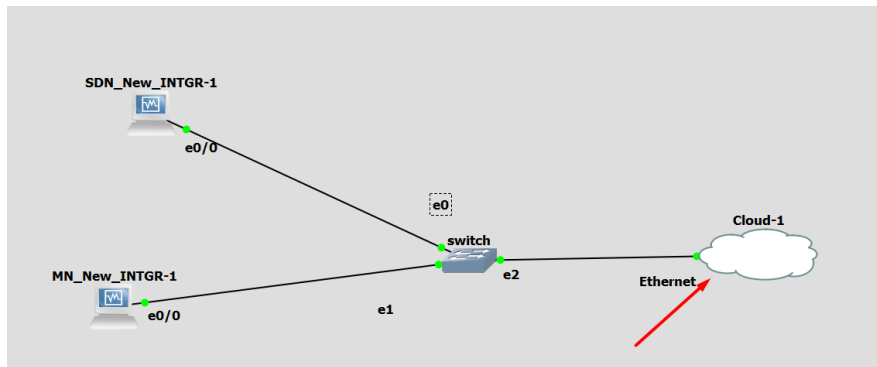
Προσοχή

Θα πρέπει να είμαστε συνδεδεμένοι στην IP διεύθυνση του ethernet στον server του GNS3. Δηλαδή πρέπει να πάμε Edit->Preferences->Server και να επιλέξουμε την IP που δίνει ο πάροχος στο ethernet. Ανοίγουμε ένα cmd στα Windows κάνουμε *ipconfig* και βλέπουμε την διεύθυνση που έχει αποδοθεί στο ethernet και ανάλογα αυτή επιλέγουμε το server στο GNS3.



Εικόνα 8-41 IP του Ethernet και του GNS3

Έχοντας υλοποιήσει αυτές τις κινήσεις δημιουργούμε την παρακάτω τοπολογία όπως φαίνεται στην παρακάτω εικόνα, συνδέουμε ένα switch με το cloud στην πόρτα Ethernet της τοπικής μας σύνδεσης και έτσι διαμέσων του switch θα δώσουμε πρόσβαση στο δίκτυο και IP στις δύο παρακάτω πλατφόρμες (Mininet, SDN). Όπως είπαμε και πιο πάνω οι πλατφόρμες περιμένουν IP κάνουμε ένα reload το πρότζεκτ και ανοίγουμε ξανά τις πλατφόρμες.



Εικόνα 8-42 Τοπολογία που γίνεται Integrate

Γράφοντας ifconfig και στις δύο βλέπουμε ότι έχουν πάρει IP του τοπικού μας δικτύου.

```
sdn@medium-hl.inux:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:14:9f:82
          inet addr:192.168.1.72  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe14:9f82/64 Scope:Link
          RX packets:772 errors:0 dropped:0 overruns:0 frame:0
          TX packets:471 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:75493 (73.7 KiB)  TX bytes:40513 (39.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1:128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1554 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1554 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:425055 (415.0 KiB)  TX bytes:425055 (415.0 KiB)

sdn@medium-hl.inux:~$

mininet@mininet-on-3:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:2d:ae:67
          inet addr:192.168.1.74  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:741 errors:0 dropped:0 overruns:0 frame:0
          TX packets:516 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:60625 (60.6 KB)  TX bytes:42892 (42.8 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:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1280 (1.2 KB)  TX bytes:1280 (1.2 KB)

mininet@mininet-on-3:~$
```

Εικόνα 8-43 Απόδοση IP DHCP από το τοπικό μας δίκτυο

Ping από το Mininet στον SDN και το αντίστροφο

SDN IP: 192.168.1.72

Mininet IP: 192.168.1.74



```

SDN_New_INTGR [Εξ λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Ύψοκευές Βοήθεια

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:772 errors:0 dropped:0 overruns:0 frame:0
TX packets:471 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:75493 (73.7 KiB) TX bytes:40513 (39.5 KiB)

lo
  Link encap:Local Loopback
  inet addr:127.0.0.1 Mask:255.0.0.0
  inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:154 errors:0 dropped:0 overruns:0 frame:0
TX packets:154 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:425055 (415.0 KiB) TX bytes:425055 (415.0 KiB)

sdn@medium-hl-linux:~$ ping 192.168.1.74
PING 192.168.1.74 (192.168.1.74) 56(84) bytes of data:
64 bytes from 192.168.1.74: icmp_seq=1 ttl=64 time=1.32 ms
64 bytes from 192.168.1.74: icmp_seq=2 ttl=64 time=1.39 ms
64 bytes from 192.168.1.74: icmp_seq=3 ttl=64 time=1.79 ms
^C
--- 192.168.1.74 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 1.326/1.670/1.890/0.251 ms
sdn@medium-hl-linux:~$

MN_New_INTGR [Εξ λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Ύψοκευές Βοήθεια

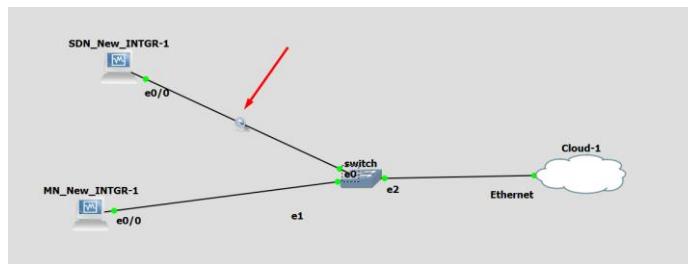
eth0
  Link encap:Ethernet  Hwaddr 98:80:27:2d:ae:67
  inet addr:192.168.1.74 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:741 errors:0 dropped:0 overruns:0 carrier:0
TX packets:516 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:60625 (60.6 KB) TX bytes:42892 (42.8 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:29 errors:0 dropped:0 overruns:0 frame:0
TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:1280 (1.2 KB) TX bytes:1280 (1.2 KB)

mininet@mininet-om:~$ ping 192.168.1.72
PING 192.168.1.72 (192.168.1.72) 56(84) bytes of data:
64 bytes from 192.168.1.72: icmp_seq=1 ttl=64 time=1.31 ms
64 bytes from 192.168.1.72: icmp_seq=2 ttl=64 time=2.15 ms
64 bytes from 192.168.1.72: icmp_seq=3 ttl=64 time=1.95 ms
64 bytes from 192.168.1.72: icmp_seq=4 ttl=64 time=2.02 ms
64 bytes from 192.168.1.72: icmp_seq=5 ttl=64 time=2.05 ms
64 bytes from 192.168.1.72: icmp_seq=6 ttl=64 time=1.91 ms
64 bytes from 192.168.1.72: icmp_seq=7 ttl=64 time=1.99 ms
^C
--- 192.168.1.72 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 1.316/1.915/2.153/0.258 ms
  
```

Εικόνα 8-44 Επιτυχής Επικοινωνία των δύο εξομοιωτών

Στην εικόνα 8-44 βλέπουμε ότι οι δύο πλατφόρμες επικοινωνούν επιτυχώς και μπορούν και ανταλλάζουν πακέτα. Έπειτα κάνουμε capture την παρακάτω σύνδεση και στέλνουμε πάλι πακέτα από τον SDN(src 192.168.1.72)->Mininet(dest:192.168.1.74) και βλέπουμε τα ICMP πακέτα που ανταλλάσσουν.



Standard input [SDN_New_INTGR-1 Ethernet0 to switch Ethernet0]

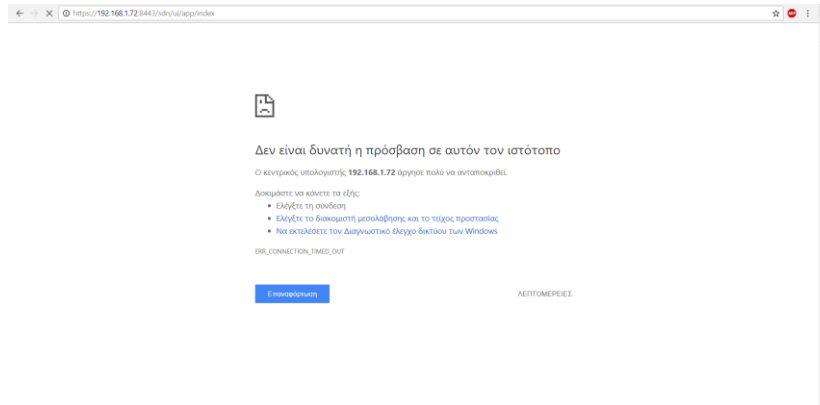
| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|--------------|--------------|----------|--------|--|
| 98 | 26.164389 | 192.168.1.72 | 192.168.1.74 | ICMP | 98 | Echo (ping) request id=0x0b5a, seq=1/256, ttl=64 (reply in 99) |
| 99 | 26.165338 | 192.168.1.74 | 192.168.1.72 | ICMP | 98 | Echo (ping) reply id=0x0b5a, seq=1/256, ttl=64 (request in 98) |
| 102 | 27.164858 | 192.168.1.72 | 192.168.1.74 | ICMP | 98 | Echo (ping) request id=0x0b5a, seq=2/512, ttl=64 (reply in 103) |
| 103 | 27.165835 | 192.168.1.74 | 192.168.1.72 | ICMP | 98 | Echo (ping) reply id=0x0b5a, seq=2/512, ttl=64 (request in 102) |
| 104 | 28.175856 | 192.168.1.72 | 192.168.1.74 | ICMP | 98 | Echo (ping) request id=0x0b5a, seq=3/768, ttl=64 (reply in 105) |
| 105 | 28.176816 | 192.168.1.74 | 192.168.1.72 | ICMP | 98 | Echo (ping) reply id=0x0b5a, seq=3/768, ttl=64 (request in 104) |
| 106 | 29.177313 | 192.168.1.72 | 192.168.1.74 | ICMP | 98 | Echo (ping) request id=0x0b5a, seq=4/1024, ttl=64 (reply in 107) |
| 107 | 29.178235 | 192.168.1.74 | 192.168.1.72 | ICMP | 98 | Echo (ping) reply id=0x0b5a, seq=4/1024, ttl=64 (request in 106) |
| 108 | 30.178761 | 192.168.1.72 | 192.168.1.74 | ICMP | 98 | Echo (ping) request id=0x0b5a, seq=5/1280, ttl=64 (reply in 109) |
| 109 | 30.179696 | 192.168.1.74 | 192.168.1.72 | ICMP | 98 | Echo (ping) reply id=0x0b5a, seq=5/1280, ttl=64 (request in 108) |
| 114 | 31.181943 | 192.168.1.72 | 192.168.1.74 | ICMP | 98 | Echo (ping) request id=0x0b5a, seq=6/1536, ttl=64 (reply in 115) |
| 115 | 31.182915 | 192.168.1.74 | 192.168.1.72 | ICMP | 98 | Echo (ping) reply id=0x0b5a, seq=6/1536, ttl=64 (request in 114) |

Εικόνα 8-45 Capture από το περιβάλλον του Wireshark



Είσοδος στον SDN Controller

Επόμενο βήμα που θα υλοποιήσουμε είναι να μπούμε στην IP διεύθυνση του SDN Controller στο διαδίκτυο έτσι ώστε να μπορούμε να έχουμε οπτική επαφή και εποπτεία των τοπολογιών του Mininet. Σημειώνεται ότι η IP του SDN Controller είναι 192.168.1.72. Τρέχοντας στον browser μας βλέπουμε το παρακάτω



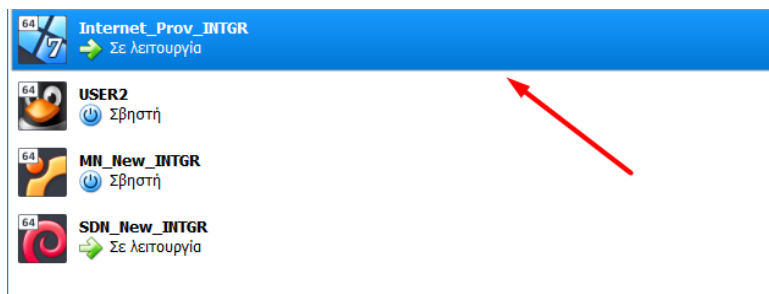
Εικόνα 8-46 Αποτυχία πρόσβασης στον controller

8.4.2 Προσθήκη εσωτερικού Browser

Μετά από ορισμένη μελέτη και περαιτέρω κατανόηση οδηγηθήκαμε στο συμπέρασμα ότι ο GNS3 δεν επέτρεπε στον Controller από την στιγμή που το είχαμε εισάγει μέσα στο project μας, στην τοπολογία μέσα δηλαδή να ακούει στην IP διεύθυνση <https://192.168.1.72:8443/sdn/ui/app/index> από κάποιον εξωτερικό browser.

- Αυτό που έπρεπε να γίνει για να πετύχουμε την οπτικοποίηση των αποτελεσμάτων που θέλαμε και να έχουμε πρόσβαση στην IP του SDN Controller είναι να έχουμε πρόσβαση από έναν εσωτερικό browser μέσα από την τοπολογία που ήδη τρέχουμε. Η λύση που υλοποιήσαμε βασίζεται στον open source χαρακτήρα του GNS3. Προσθέσαμε ένα ακόμη VM στην τοπολογία μας ένα εικονικό μηχάνημα που τρέχει Windows7 που είναι ελαφριά και έχουν και γραφικό περιβάλλον.

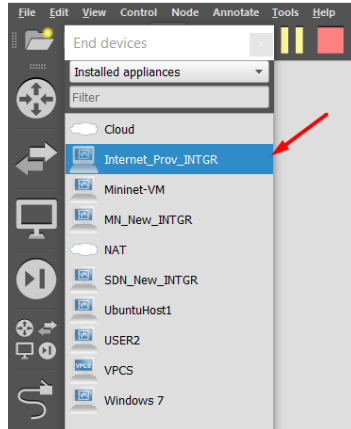
Η προσθήκη των Windows 7 μας έδωσε την λύση στην πρόσβαση σε ένα browser από την υπάρχουσα τοπολογία. Η προσθήκη της εικονικής μηχανής Windows 7 στην τοπολογία μας προϋποθέτει να διαθέτουμε το λειτουργικό Windows 7 και να το κάνουμε boot στο Virtual Box.



Εικόνα 8-47 Boot του Virtual Box

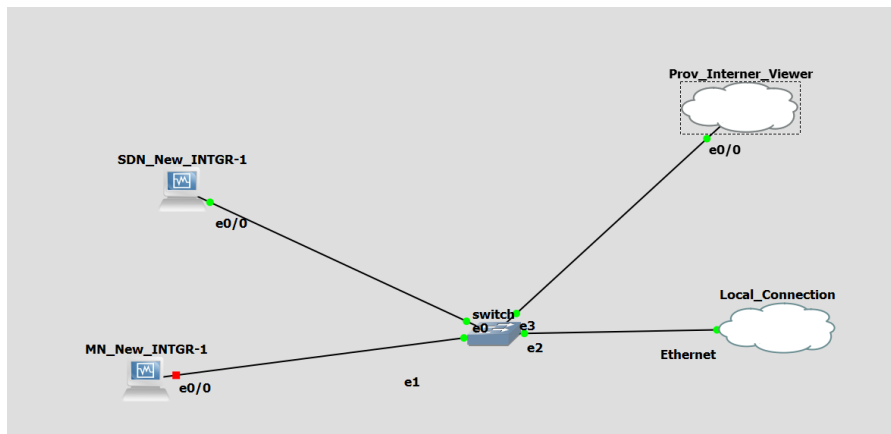


Έχοντας κάνει αυτό το βήμα με τα παραδοσιακά βήματα προσθέσαμε στα διαθέσιμα devices το Windows 7 machine το οποίο το ονομάσαμε *Internet_Prov*



Εικόνα 8-48 Προσθήκη στα devices

Στη συνέχεια αυτό που έπρεπε να κάνουμε ήταν να το συνδέσουμε στο switch που έχουμε για να μεταβιβάζει πακέτα το Internet_Prov μηχανήμα έτσι ώστε να πάρει IP από το τοπικό μας δίκτυο και να έχει πρόσβαση στο Internet.



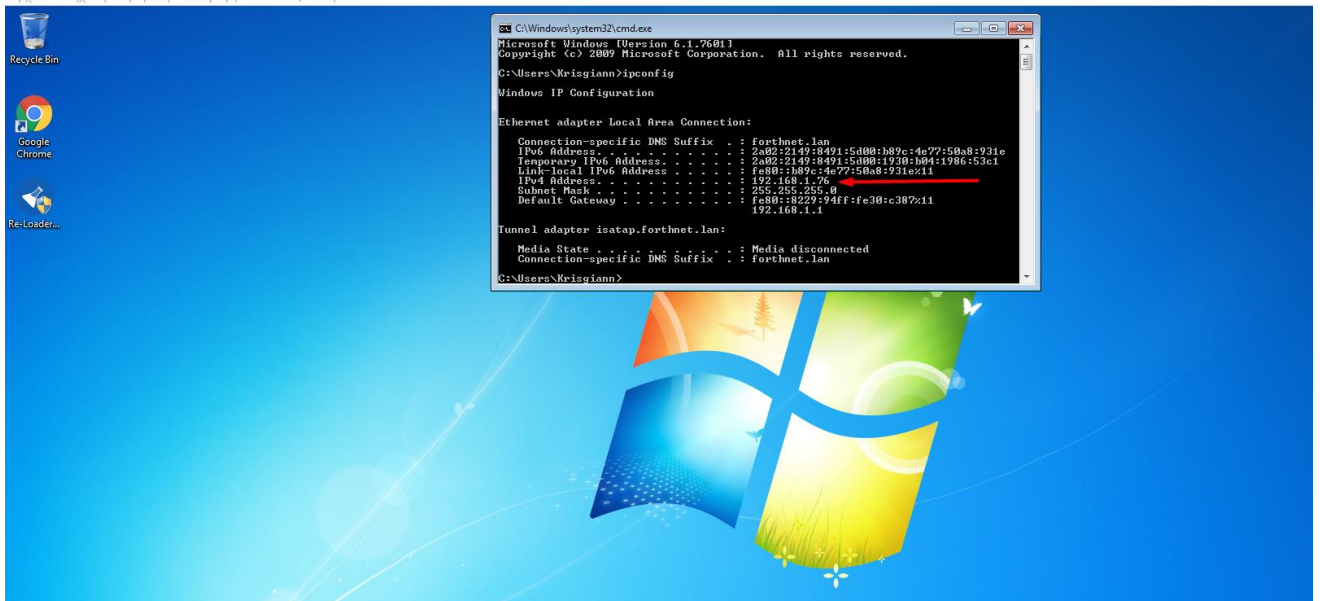
Εικόνα 8-49 Νέα μορφή της τοπολογίας μας

Κάνοντας boot το νέο μηχανήμα και πατώντας *ipconfig* βλέπουμε τα παρακάτω, ότι έχουμε πάρει IP της τοπικής μας σύνδεσης.

IP WINDOWS MACHINE :192.168.1.76

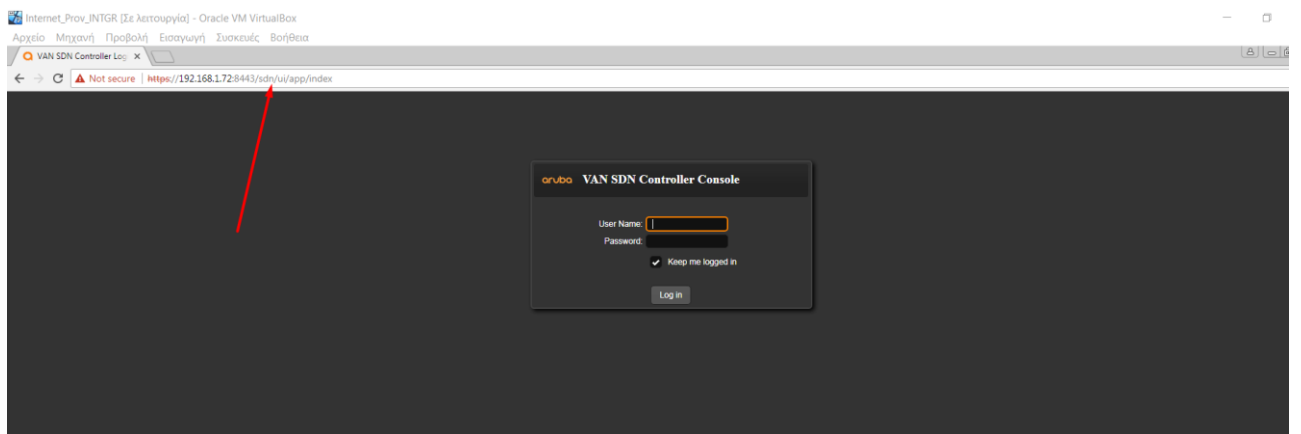


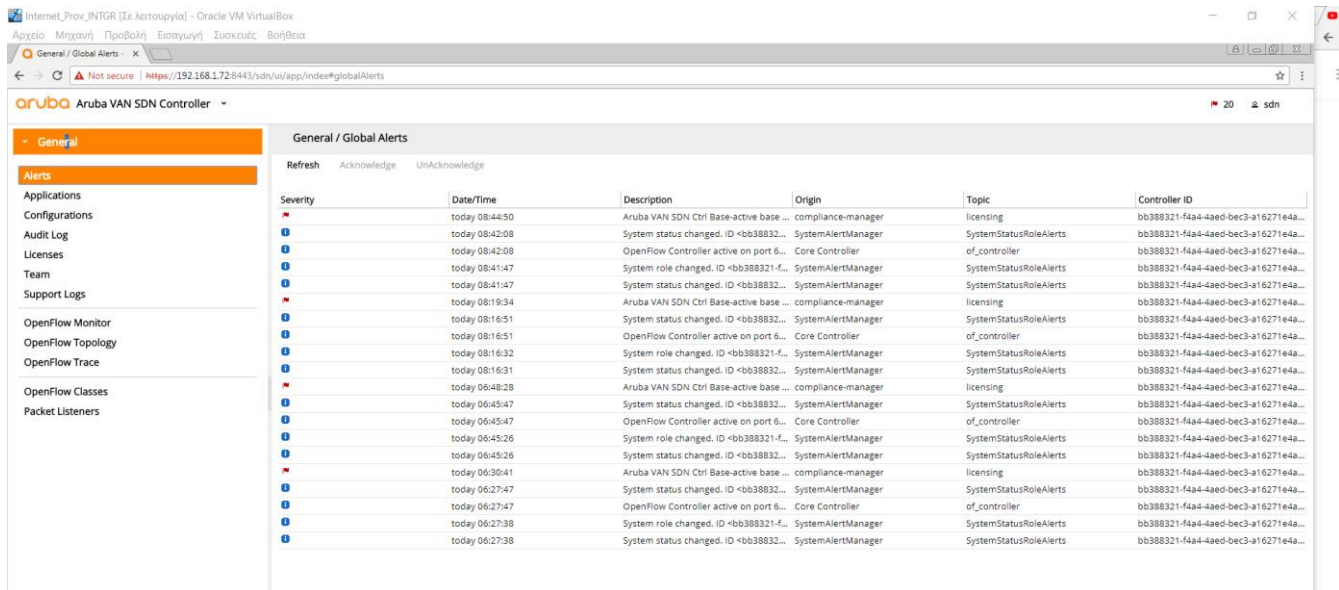
Internet_Prov_INTGR [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια



Εικόνα 8-50 Επιτυχή απόδοση IP διεύθυνσης στο Windows 7 machine

Τώρα θα ανοίξουμε ένα browser και θα προσπαθήσουμε να συνδεθούμε στην πόρτα του sdn μέσα από τον εσωτερικό browser της τοπολογίας μας. Έχοντας ανοίξει τον browser και πληκτρολογώντας την IP του SDN και την πόρτα 8443 που ακούει επιτυχώς μεταφερόμαστε στην σελίδα του ελεγκτή. Πληκτρολογώντας τα κατάλληλα credentials μπαίνουμε μέσα στον ελεγκτή και έχουμε τις πλατφόρμες όλες διασυνδεδεμένες μεταξύ τους.

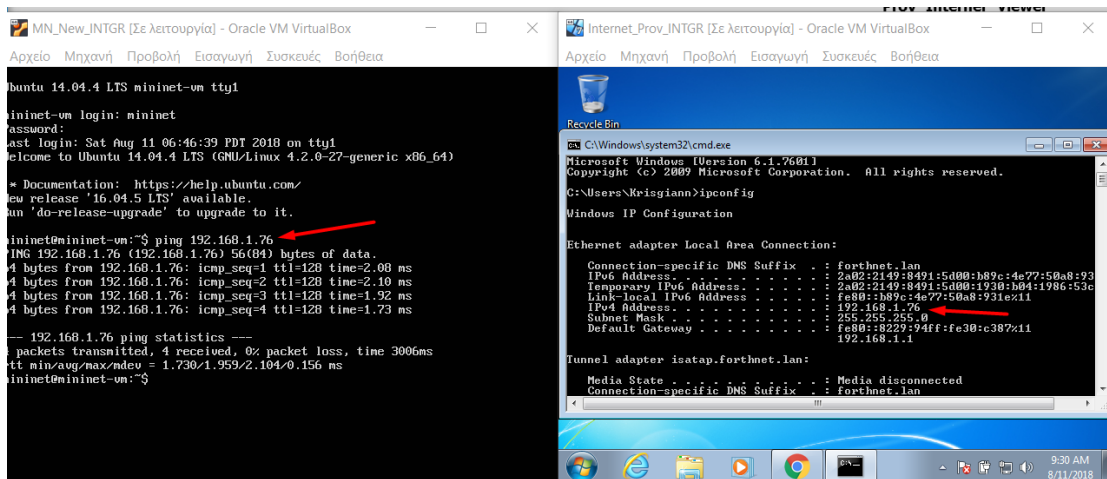




Εικόνα 8-51 Περιβάλλον του SDN Controller

Αυτή τη στιγμή ο Openflow Controller δεν παρακολουθεί κάποια κίνηση διότι δεν τον έχουμε ορίσει από κάποια πλατφόρμα ως remote controller πράγμα που θα πάμε στη συνέχεια να κάνουμε από το περιβάλλον του Mininet.

Βλέπουμε και ότι το Mininet επικοινωνεί με τον Windows machine οπότε ολόκληρη η τοπολογία μας είναι διασυνδεδεμένη.



Εικόνα 8-52 Επιτυχής επικοινωνία Mininet-Windows machine

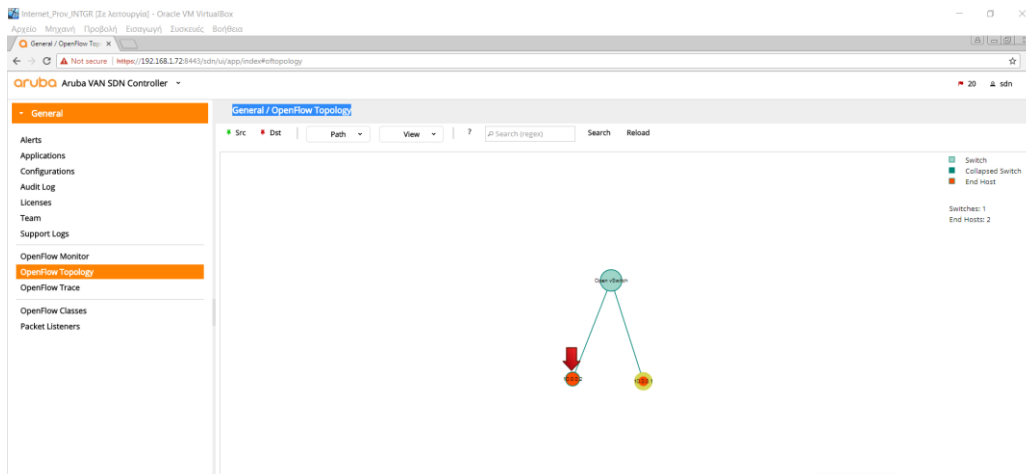
Πηγαίνουμε τώρα στο Mininet και δίνουμε συγκεκριμένο controller τον δικό μας SDN Controller έτσι ώστε να επιτευχθεί η διασύνδεση και η οπτικοποίηση των τοπολογιών του Mininet από τον SDN. Κάνουμε ένα απλό τεστ απλά βάζουμε remote controller την IP του SDN και δημιουργούμε μία απλή τοπολογία ένα switch με δύο hosts και μετά κάνουμε pingall για να δημιουργηθεί.



```
mininet@mininet-vm:~$ sudo mn --controller=remote,ip=192.168.1.72
*** Creating network
*** Adding controller
Unable to contact the remote controller at 192.168.1.72:6653
Connecting to remote controller at 192.168.1.72:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

Εικόνα 8-53 Σύνδεση με τον remote controller

Πηγαίνουμε στον εσωτερικό browser που δουλεύει στα Windows 7 κάνουμε ένα reload του ελεγκτή μας και βλέπουμε το παρακάτω, βλέπουμε το δημιουργημένο Open switch και τους δύο hosts πράγμα που σημαίνει ότι έχουμε πλήρη διασύνδεση στις πλατφόρμες μας.



Εικόνα 8-54 Τοπολογία που μας εμφανίζεται στον ελεγκτή Aruba SDN Controller



8.5 Πειράματα που υλοποιήθηκαν στον διασυνδεδεμένο μηχανισμό

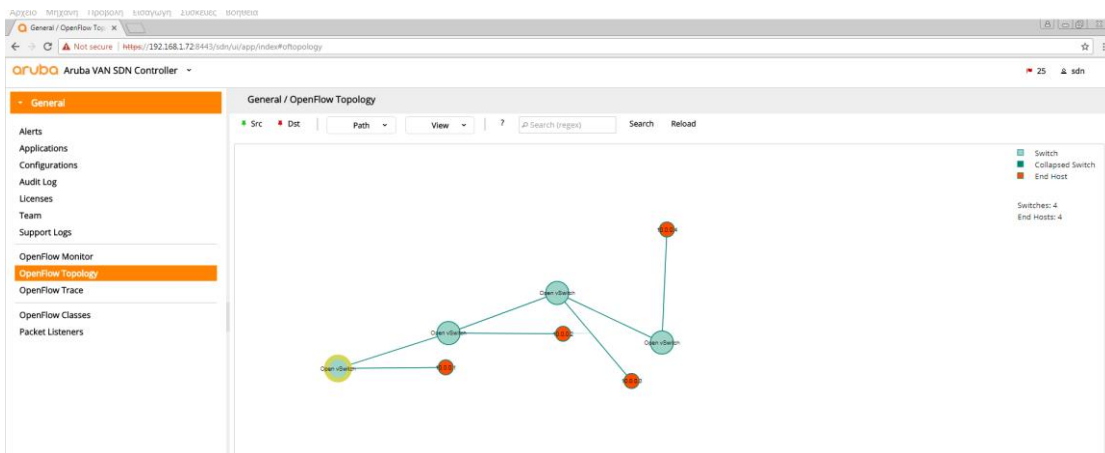
8.5.1 1^ο πείραμα

Παρακάτω θα δούμε μια τοπολογία όπου γράφοντας στο Mininet τον παρακάτω κώδικα δημιουργούμε μια γραμμική τοπολογία με 4 hosts και 4 openflow switch προσδιορίζοντας πάντα την IP του Controller και το πρωτόκολλο OpenFlow1.3 η νεότερη έκδοση που χρησιμοποιεί ο SDN μας.

Mininet Command

```
sudo mn --controller=remote,ip=192.168.1.72 --switch=ovsk,protocols=OpenFlow13 --mac --topo=linear,4
```

Έχοντας γράψει την εντολή στο Mininet και έχοντας τρέξει και την εντολή pingall για να επικοινωνήσουν οι hosts μεταξύ τους βλέπουμε στον Openflow controller που βλέπουμε διαμέσων του εσωτερικού browser που έχουμε δημιουργήσει την εν λόγω τοπολογία



Εικόνα 8-55 4hosts-linear-4switch

Στην καρτέλα Openflow Monitor βλέπουμε την διεύθυνση των switches που είναι αυτή του Mininet 192.168.1.74

| Data Path ID | Address | Negotiated Version | Manufacturer | HW Version | SW Version | Serial # |
|----------------------|--------------|--------------------|--------------|--------------|------------|----------|
| 00:00:00:00:00:00:01 | 192.168.1.74 | 1.3.0 | Nicira, Inc. | Open vSwitch | 2.0.2 | None |
| 00:00:00:00:00:00:02 | 192.168.1.74 | 1.3.0 | Nicira, Inc. | Open vSwitch | 2.0.2 | None |
| 00:00:00:00:00:00:03 | 192.168.1.74 | 1.3.0 | Nicira, Inc. | Open vSwitch | 2.0.2 | None |
| 00:00:00:00:00:00:04 | 192.168.1.74 | 1.3.0 | Nicira, Inc. | Open vSwitch | 2.0.2 | None |

Εικόνα 8-56 Monitoring της τοπολογίας

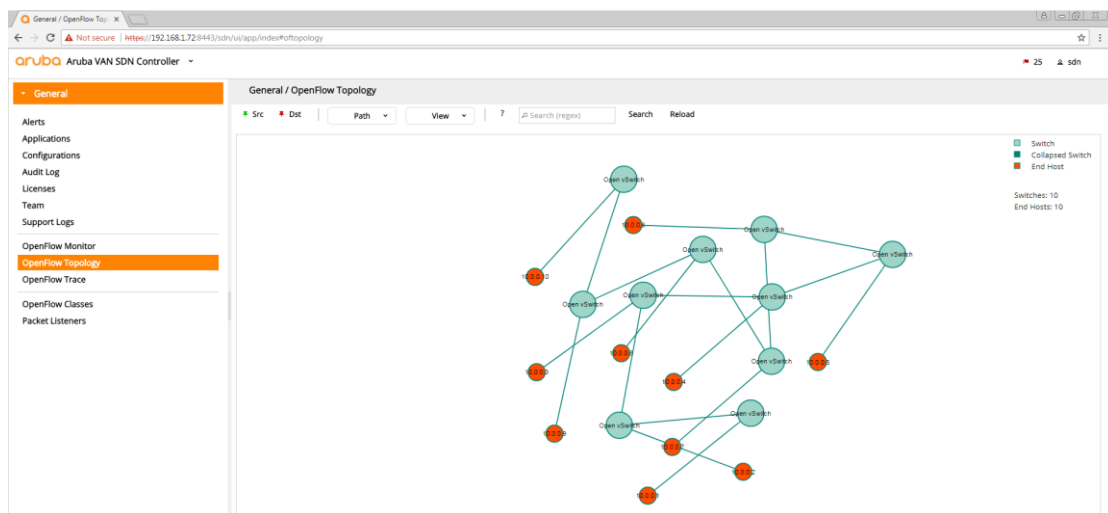


Τίδια λογική με 10 hosts και 10 switches

Τίδια λογική με 10 hosts και 10 switches

Mininet Command

`sudo mn --controller=remote,ip=192.168.1.72 --switch=ovsk,protocols=OpenFlow13 --mac --topo=linear,10`



Εικόνα 8-57 Γραμμική τοπολογία 10hosts-10OpenVswitch

Δοκιμάζουμε κάτι πολύ μεγάλο 100 hosts-100 switches και καταλαβαίνουμε έτσι τη δύναμη του Mininet να κατασκευάζει πολύ γρήγορα μεγάλες δικτυακές διατάξεις.

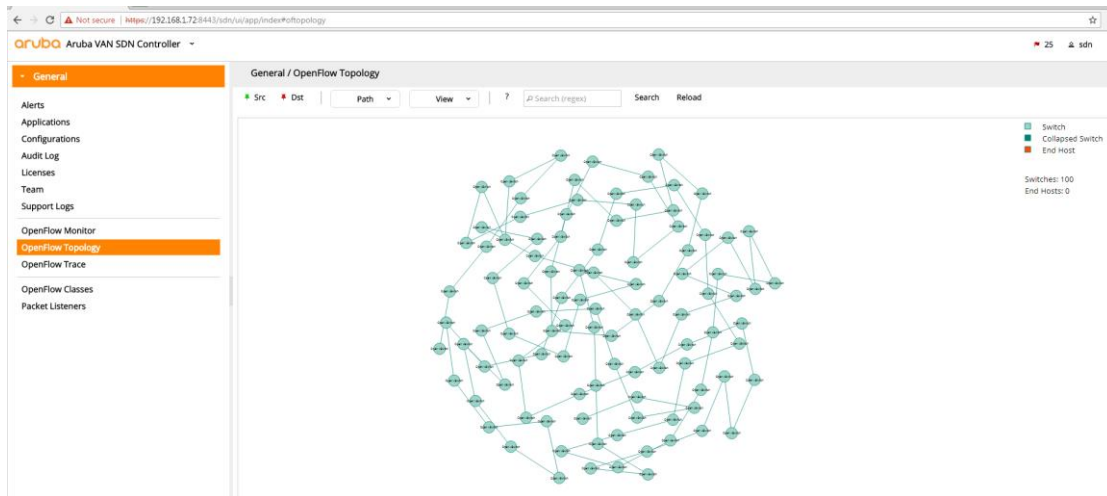
Mininet Command

`sudo mn --controller=remote,ip=192.168.1.72 --switch=ovsk,protocols=OpenFlow13 --mac --topo=linear,100`

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

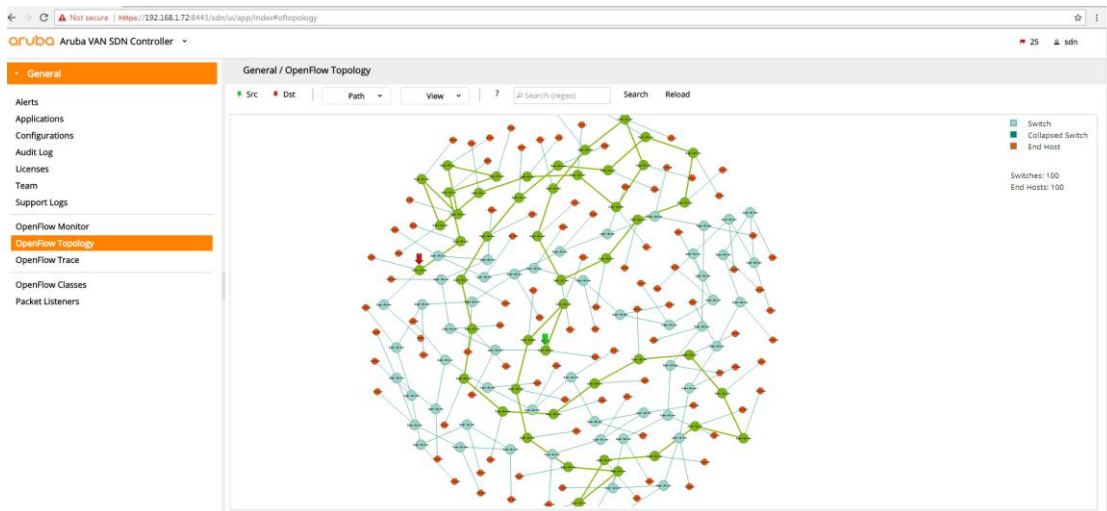
s7, s97) (h98, s98) (h99, s99) (h100, s100) (s2, s1) (s3, s2) (s4, s3) (s5, s4)
(s6, s5) (s7, s6) (s8, s7) (s9, s8) (s10, s9) (s11, s10) (s12, s11) (s13, s12) (
s14, s13) (s15, s14) (s16, s15) (s17, s16) (s18, s17) (s19, s18) (s20, s19) (s21
, s20) (s22, s21) (s23, s22) (s24, s23) (s25, s24) (s26, s25) (s27, s26) (s28, s
27) (s29, s28) (s30, s29) (s31, s30) (s32, s31) (s33, s32) (s34, s33) (s35, s34)
(s36, s35) (s37, s36) (s38, s37) (s39, s38) (s40, s39) (s41, s40) (s42, s41) (s
43, s42) (s44, s43) (s45, s44) (s46, s45) (s47, s46) (s48, s47) (s49, s48) (s50,
s49) (s51, s50) (s52, s51) (s53, s52) (s54, s53) (s55, s54) (s56, s55) (s57, s5
5) (s58, s57) (s59, s58) (s60, s59) (s61, s60) (s62, s61) (s63, s62) (s64, s63)
(s65, s64) (s66, s65) (s67, s66) (s68, s67) (s69, s68) (s70, s69) (s71, s70) (s7
2, s71) (s73, s72) (s74, s73) (s75, s74) (s76, s75) (s77, s76) (s78, s77) (s79,
s78) (s80, s79) (s81, s80) (s82, s81) (s83, s82) (s84, s83) (s85, s84) (s86, s85
) (s87, s86) (s88, s87) (s89, s88) (s90, s89) (s91, s90) (s92, s91) (s93, s92) (
s94, s93) (s95, s94) (s96, s95) (s97, s96) (s98, s97) (s99, s98) (s100, s99)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h
43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h
63 h64 h65 h66 h67 h68 h69 h70 h71 h72 h73 h74 h75 h76 h77 h78 h79 h80 h81 h82 h
83 h84 h85 h86 h87 h88 h89 h90 h91 h92 h93 h94 h95 h96 h97 h98 h99 h100
*** Starting controller
s0
*** Starting 100 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15 s16 s17 s18 s19 s20 s21 s22 s
23 s24 s25 s26 s27 s28 s29 s30 s31 s32 s33 s34 s35 s36 s37 s38 s39 s40 s41 s42 s
43 s44 s45 s46 s47 s48 s49 s50 s51 s52 s53 s54 s55 s56 s57 s58 s59 s60 s61 s62 s
63 s64 s65 s66 s67 s68 s69 s70 s71 s72 s73 s74 s75 s76 s77 s78 s79 s80 s81 s82 s
83 s84 s85 s86 s87 s88 s89 s90 s91 s92 s93 s94 s95 s96 s97 s98 s99 s100 ...
*** Starting CLI:
mininet> -
```

Εικόνα 8-58 Δημιουργία 100 switches-100hosts



Εικόνα 8-59 100 switches στον ελεγκτή

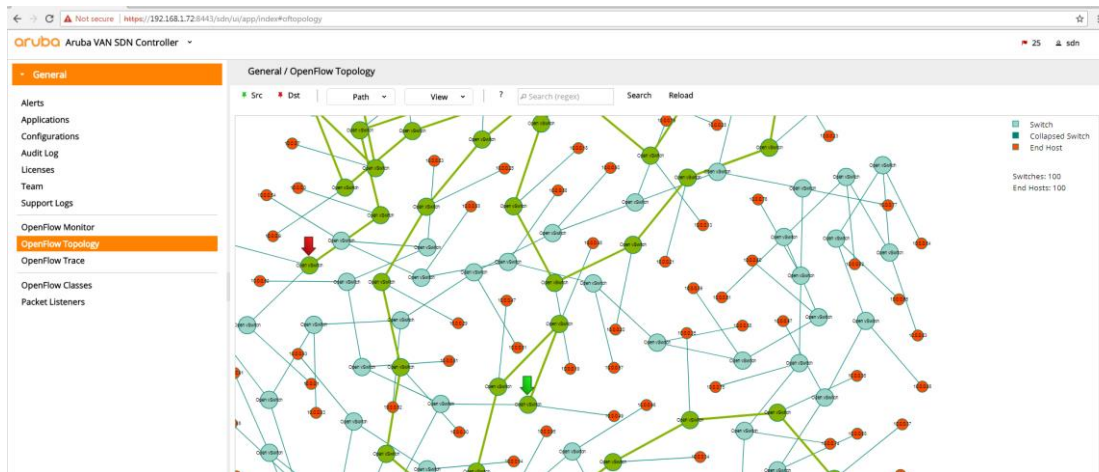
Για να εμφανιστούν και οι hosts πρέπει να κάνουμε κάποιο ping, αν εκτελέσουμε pingall θα επικοινωνήσουν όλοι με όλους τους host αλλά θα περιμένουμε ένα διάστημα για να τρέξει όλη η εξομοίωση. Παρακάτω βλέπουμε τους hosts που έχουν εμφανιστεί. Αυτό που μπορούμε να κάνουμε είναι να ορίσουμε ένα source και ένα destination στα switch και να δούμε το shortest μονοπάτι που θα ακολουθήσει.



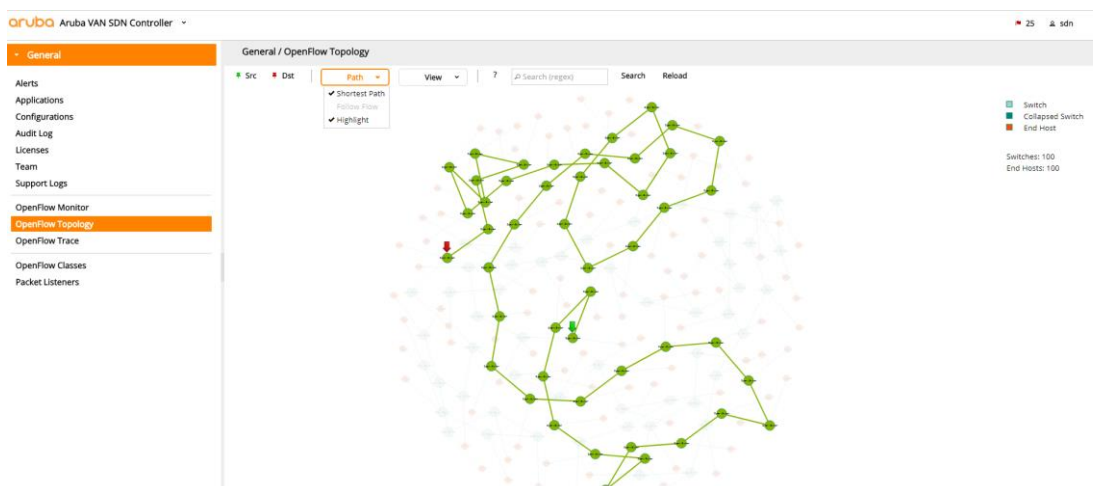
Εικόνα 8-60 Εμφάνιση hosts



Με κόκκινο βέλος ορίζουμε το dest και με πράσινο το source όπου ο ελεγκτής μας δίνει τη δυνατότητα να επιλέξουμε εμείς με τις καρφίτσες *src* και *dest* που διαθέτει.



Εικόνα 8-61 Επιλογή μικρότερου μονοπατιού



Εικόνα 8-62 Απομόνωση του καλύτερου μονοπατιού

2^ο πείραμα

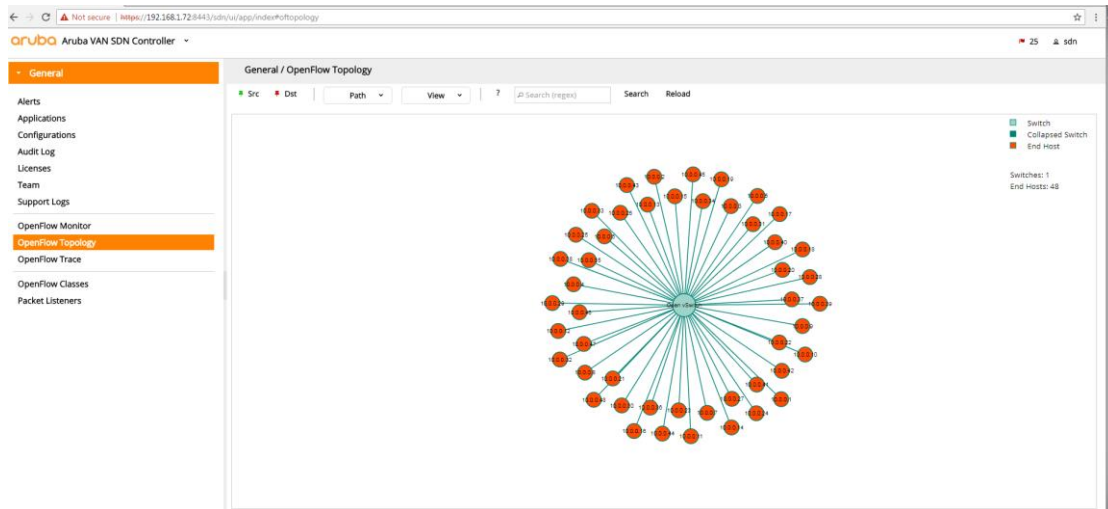
Στο πρώτο πείραμα είδαμε την δυνατότητα του Mininet να κατασκευάζει πάρα πολλά switch και φυσικά η διαχείριση 100 μεταγωγών είναι ένα αρκετά δύσκολο και περίπλοκο πράγμα. Εδώ θα δούμε ένα switch το οποίο θα έχει 48 ports και κατ' επέκταση 48 hosts.



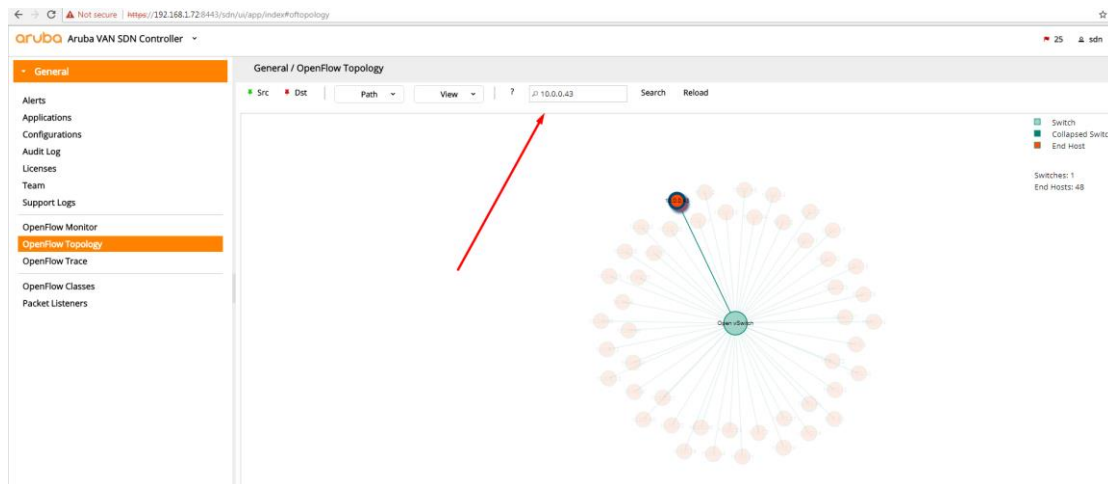
Mininet Command

```
sudo mn --controller=remote,ip=192.168.1.72 --switch=ovsk,protocols=OpenFlow13 --mac --topo=star,48
```

Star Topology κάθε host απέχει ένα hop από τον άλλον. Επίσης υπάρχει η δυνατότητα απομόνωσης του κάθε host όπου στο search βάζοντας την IP του host που επιθυμούμε τον παγώνει σε real time χρόνο και μπορούμε να ασχοληθούμε μαζί του ενώ η τοπολογία τρέχει.



Εικόνα 8-63 Star Topology



Εικόνα 8-64 Απομόνωση Host με IP 10.0.0.43

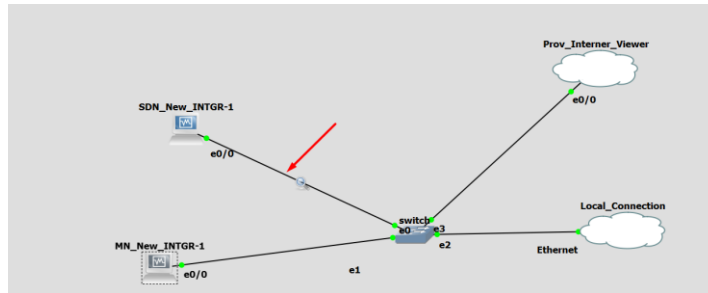
3^ο πείραμα



Ένα switch με 4 ports και ορισμένα capture από το Wireshark

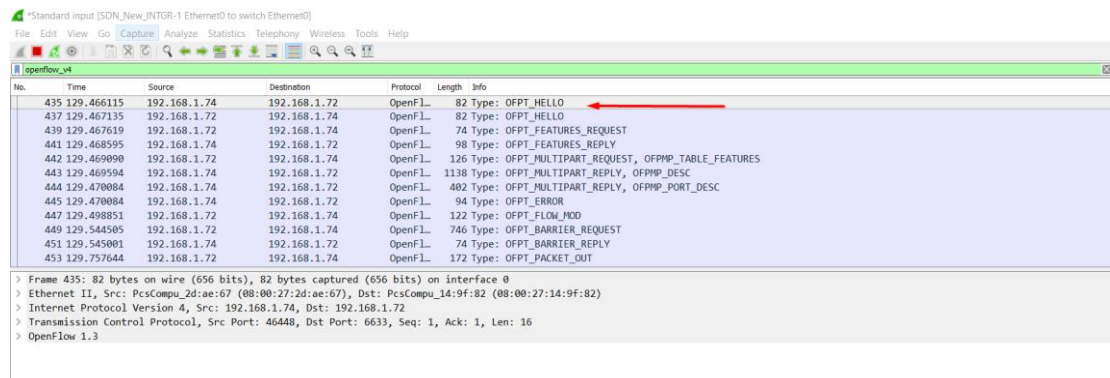
Mininet Command

```
sudo mn --controller=remote,ip=192.168.1.72 --switch=ovsk,protocols=OpenFlow13 --mac --topo=single,4
```



Εικόνα 8-65 Capture του link e0/0

Έχοντας ανοίξει το Wireshark όντας έτοιμο να κάνει capture τρέχουμε την τοπολογία στο Mininet και πηγαίνουμε έπειτα στο Wireshark και βάζουμε φίλτρο το openflow_v4 το οποίο είναι το OpenFlow13 πρωτόκολλο που χρησιμοποιούμε.



Εικόνα 8-66 Hello messages

Αυτό που βλέπουμε εδώ είναι ένα Hello message που στέλνει το switch προς τον controller, την απάντηση του controller πίσω στο switch, βλέπουμε ότι ο controller ζητά κάποια features από το switch. Το switch απαντά με κάποια features όπως το ότι δεν διαθέτει πρωτόκολλο spanning tree μέχρι στιγμής.



| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|--------------|--------------|-----------|--------|--|
| 435 | 129.466115 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 82 | Type: OFPT_HELLO |
| 437 | 129.467135 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 82 | Type: OFPT_HELLO |
| 439 | 129.467619 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 74 | Type: OFPT_FEATURES_REQUEST |
| 441 | 129.468595 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 98 | Type: OFPT_FEATURES_REPLY |
| 442 | 129.469090 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 126 | Type: OFPT_MULTIPART_REQUEST, OFPMP_TABLE_FEATURES |
| 443 | 129.469594 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 1138 | Type: OFPT_MULTIPART_REPLY, OFPMP_DESC |
| 444 | 129.470084 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 402 | Type: OFPT_MULTIPART_REPLY, OFPMP_PORT_DESC |
| 445 | 129.470084 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 94 | Type: OFPT_ERROR |
| 447 | 129.498851 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 122 | Type: OFPT_FLOW_MOD |
| 449 | 129.544505 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 746 | Type: OFPT_BARRIER_REQUEST |
| 451 | 129.545001 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 74 | Type: OFPT_BARRIER_REPLY |
| 453 | 129.757644 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 172 | Type: OFPT_PACKET_OUT |

n_buffers: 256
n_tables: 254
auxiliary_id: 0
Pad: 0
capabilities: 0x00000047
.....1 = OFPC_FLOW_STATS: True
.....1 = OFPC_TABLE_STATS: True
.....1 = OFPC_PORT_STATS: True
.....0... = OFPC_GROUP_STATS: False
.....0... = OFPC_IP_REASM: False
.....1... = OFPC_QUEUE_STATS: True
.....0... = OFPC_PORT_BLOCKED: False
Reserved: 0x00000000

Εικόνα 8-67 Απάντηση switch με κάποια features

Ουσιαστικά του λέει ποια ports ξέρει και πληροφορίες για τα 4 ports του μεταγωγέα και ένα ακόμη local port του μεταγωγέα.

*Standard input [SDN_New_INTGR-1 Ethernet0 to switch Ethernet0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|--------------|--------------|-----------|--------|--|
| 435 | 129.466115 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 82 | Type: OFPT_HELLO |
| 437 | 129.467135 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 82 | Type: OFPT_HELLO |
| 439 | 129.467619 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 74 | Type: OFPT_FEATURES_REQUEST |
| 441 | 129.468595 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 98 | Type: OFPT_FEATURES_REPLY |
| 442 | 129.469090 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 126 | Type: OFPT_MULTIPART_REQUEST, OFPMP_TABLE_FEATURES |
| 443 | 129.469594 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 1138 | Type: OFPT_MULTIPART_REPLY, OFPMP_DESC |
| 444 | 129.470084 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 402 | Type: OFPT_MULTIPART_REPLY, OFPMP_PORT_DESC |
| 445 | 129.470084 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 94 | Type: OFPT_ERROR |
| 447 | 129.498851 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 122 | Type: OFPT_FLOW_MOD |
| 449 | 129.544505 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 746 | Type: OFPT_BARRIER_REQUEST |
| 451 | 129.545001 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 74 | Type: OFPT_BARRIER_REPLY |
| 453 | 129.757644 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 172 | Type: OFPT_PACKET_OUT |

.....0... = OFPMP_REPLY_MORE: 0x0
Pad: 00000000
Port
Port no: 3
Pad: 00000000
Hw addr: 6a:56:d6:fc:3c:88 (6a:56:d6:fc:3c:88)
Pad: 0000
Name: s1-eth3
Config: 0x00000000
State: 0x00000000
.....0... = OFPMP_REPLY_MORE: 0x0
Pad: 00000000
Port
Port no: 3
Pad: 00000000
Hw addr: 6a:56:d6:fc:3c:88 (6a:56:d6:fc:3c:88)
Pad: 0000
Name: s1-eth3
Config: 0x00000000
State: 0x00000000
Current: 0x00000000
Advertised: 0x00000000

*Standard input [SDN_New_INTGR-1 Ethernet0 to switch Ethernet0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|--------------|--------------|-----------|--------|--|
| 435 | 129.466115 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 82 | Type: OFPT_HELLO |
| 437 | 129.467135 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 82 | Type: OFPT_HELLO |
| 439 | 129.467619 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 74 | Type: OFPT_FEATURES_REQUEST |
| 441 | 129.468595 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 98 | Type: OFPT_FEATURES_REPLY |
| 442 | 129.469090 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 126 | Type: OFPT_MULTIPART_REQUEST, OFPMP_TABLE_FEATURES |
| 443 | 129.469594 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 1138 | Type: OFPT_MULTIPART_REPLY, OFPMP_DESC |
| 444 | 129.470084 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 402 | Type: OFPT_MULTIPART_REPLY, OFPMP_PORT_DESC |
| 445 | 129.470084 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 94 | Type: OFPT_ERROR |
| 447 | 129.498851 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 122 | Type: OFPT_FLOW_MOD |
| 449 | 129.544505 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 746 | Type: OFPT_BARRIER_REQUEST |
| 451 | 129.545001 | 192.168.1.74 | 192.168.1.72 | OpenFL... | 74 | Type: OFPT_BARRIER_REPLY |
| 453 | 129.757644 | 192.168.1.72 | 192.168.1.74 | OpenFL... | 172 | Type: OFPT_PACKET_OUT |

> Peer: 0x00000000
Curr speed: 10000000
Max speed: 0
Port
Port no: OFPP_LOCAL (4294967294)
Pad: 00000000
Hw addr: a6:db:bb:f6:ec:44 (a6:db:bb:f6:ec:44)
Pad: 0000
Name: s1
Config: 0x00000000
State: 0x00000000
Current: 0x00000000
Advertised: 0x00000000

Εικόνα 8-68 Port description



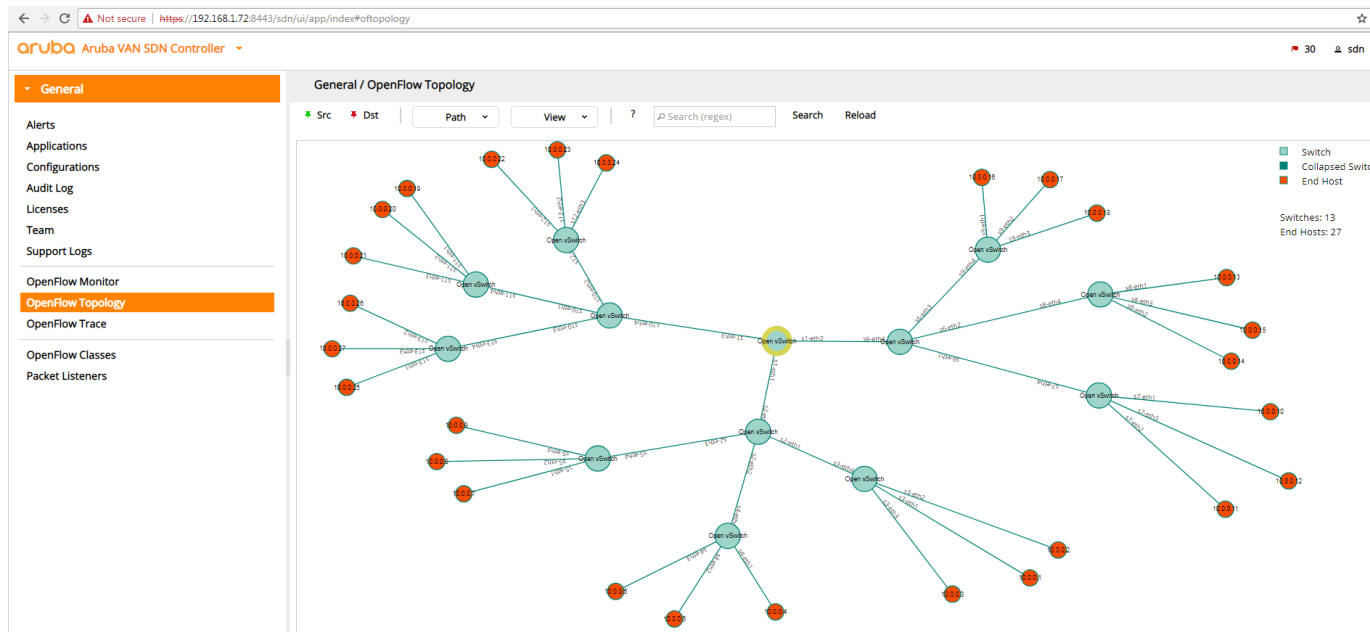
4^ο πείραμα

Εδώ θα μελετήσουμε μία τοπολογία σε μορφή δέντρου με βάθος 3 και αριθμό hosts ανά στρώμα=3

Mininet Command

```
sudo mn --controller=remote,ip=192.168.1.72 --switch=ovsk,protocols=OpenFlow13 --mac --topo=tree,depth=3,fanout=3
```

Έχουμε ένα κεντρικό switch που είναι συνδεδεμένο με άλλα 3 switch που αυτά τα 3 switch έχουν από 3 switch το καθένα και κάθε τελικό switch έχει 3 hosts



Εικόνα 8-69 Tree topology fanout=3 depth=3



Πηγαίνοντας στην καρτέλα OpenFlowMonitor μπορούμε και βλέπουμε τα πακέτα που περνούν από το κεντρικό switch με mac address **00:00:00:00:00:00:01** από τα port1 ,port 2 και port 3.

| Table ID | Flow Count | Table Name | Priority | Packets | Bytes | Match | Actions/Instructions | Flow Class ID |
|----------|------------|------------|----------|---------|--------|---|---|-------------------------|
| 60000 | 8 | | 60000 | | 528 | eth_type: bddp | apply_actions: output: CONTROLLER | com.hp.sdn.bddp.steal |
| 34000 | 642 | | 34000 | 642 | 26964 | in_port: 1 eth_type: arp | apply_actions: output: NORMAL | com.hp.sdn.infra.filter |
| 34000 | 638 | | 34000 | 638 | 26796 | in_port: 2 eth_type: arp | apply_actions: output: NORMAL | com.hp.sdn.infra.filter |
| 34000 | 287 | | 34000 | 287 | 12054 | in_port: 3 eth_type: arp | apply_actions: output: NORMAL | com.hp.sdn.infra.filter |
| 31500 | 0 | | 31500 | 0 | 0 | eth_type: ipv4 ip_proto: udp udp_src: 67 udp_dst: 68 | apply_actions: output: CONTROLLER output: NORMAL | com.hp.sdn.dhcp.copy |
| 31500 | 0 | | 31500 | 0 | 0 | eth_type: ipv4 ip_proto: udp udp_src: 68 udp_dst: 67 | apply_actions: output: CONTROLLER output: NORMAL | com.hp.sdn.dhcp.copy |
| 31000 | 0 | | 31000 | 0 | 0 | eth_type: arp | apply_actions: output: CONTROLLER output: NORMAL | com.hp.sdn.arp.copy |
| 0 | 1952 | | 0 | 1952 | 191296 | | apply_actions: output: NORMAL | com.hp.sdn.normal |

Εικόνα 8-70 Πακέτα που περνούν από το κεντρικό switch

Επίσης στην καρτέλα trace μπορούμε να δούμε τα request και τα reply που περνούν από κάθε switch βλέπουμε τις αιτήσεις από τους υπόλοιπους κόμβους που περνούν διαμέσων του κεντρικού switch με mac 00:00:00:00:00:00:01. Με Rx βλέπουμε τις αιτήσεις που λαμβάνει(receive) ο Controller και με Tx αυτές που στέλνει (transmit) ο controller.

| Time | Event | Data Path ID | Message |
|--------------|-------|----------------------|-------------------------------|
| 08:30:53.877 | Ctrl | | Recording started [10s] |
| 08:30:54.867 | Rx | 00:00:00:00:00:00:05 | {ofm}[V_1_3.ECHO_REQUEST.8.0] |
| 08:30:54.867 | Rx | 00:00:00:00:00:00:08 | {ofm}[V_1_3.ECHO_REQUEST.8.0] |
| 08:30:54.868 | Tx | 00:00:00:00:00:00:05 | {ofm}[V_1_3.ECHO_REPLY.8.0] |
| 08:30:54.868 | Tx | 00:00:00:00:00:00:08 | {ofm}[V_1_3.ECHO_REPLY.8.0] |
| 08:30:54.868 | Rx | 00:00:00:00:00:00:09 | {ofm}[V_1_3.ECHO_REQUEST.8.0] |
| 08:30:54.868 | Tx | 00:00:00:00:00:00:09 | {ofm}[V_1_3.ECHO_REPLY.8.0] |
| 08:30:54.869 | Rx | 00:00:00:00:00:00:03 | {ofm}[V_1_3.ECHO_REQUEST.8.0] |
| 08:30:54.869 | Tx | 00:00:00:00:00:00:03 | {ofm}[V_1_3.ECHO_REPLY.8.0] |
| 08:30:54.869 | Rx | 00:00:00:00:00:00:07 | {ofm}[V_1_3.ECHO_REQUEST.8.0] |
| 08:30:54.869 | Tx | 00:00:00:00:00:00:07 | {ofm}[V_1_3.ECHO_REPLY.8.0] |
| 08:30:54.869 | Rx | 00:00:00:00:00:00:04 | {ofm}[V_1_3.ECHO_REQUEST.8.0] |
| 08:30:54.869 | Tx | 00:00:00:00:00:00:04 | {ofm}[V_1_3.ECHO_REPLY.8.0] |
| 08:30:54.869 | Rx | 00:00:00:00:00:00:00 | {ofm}[V_1_3.ECHO_REQUEST.8.0] |
| 08:30:54.869 | Tx | 00:00:00:00:00:00:00 | {ofm}[V_1_3.ECHO_REPLY.8.0] |
| 08:30:54.869 | Rx | 00:00:00:00:00:00:0c | {ofm}[V_1_3.ECHO_REQUEST.8.0] |
| 08:30:54.870 | Tx | 00:00:00:00:00:00:0c | {ofm}[V_1_3.ECHO_REPLY.8.0] |
| 08:30:54.870 | Rx | 00:00:00:00:00:00:06 | {ofm}[V_1_3.ECHO_REQUEST.8.0] |
| 08:30:54.870 | Tx | 00:00:00:00:00:00:06 | {ofm}[V_1_3.ECHO_REPLY.8.0] |

Εικόνα 8-71 Echo Request και echo reply



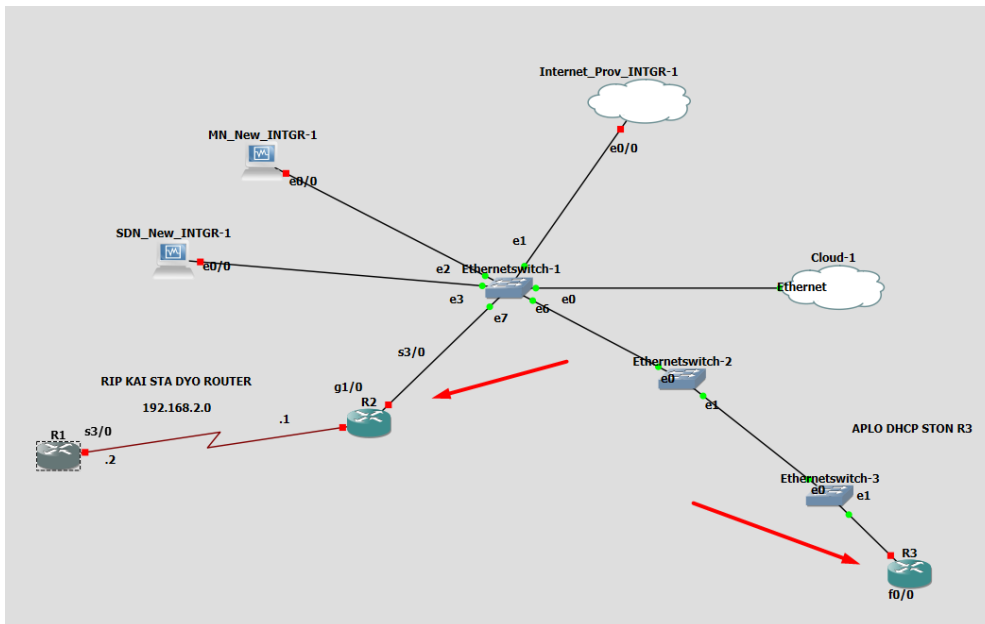
| Flow Class ID | Priority | Cookie | Match Fields | Actions | Description |
|-------------------------|----------|---------|-----------------------------------|--------------|--|
| com.hp.sdn.arp.copy | 31000 | 0xfff | ETH_TYPE | COPY | Copies ARP requests for node location |
| com.hp.sdn.arp.filter | 34000 | 0xffff | IN_PORT,ETH_TYPE | FORWARD | Filter ARP copies from infrastructure ports |
| com.hp.sdn.bddp.steal | 60000 | 0xffff | ETH_TYPE | STEAL | Steal BDDP packets to the controller for link discovery |
| com.hp.sdn.dhcp.copy | 31500 | 0xffff | ETH_TYPE,IP_PROTO,UDP_SRC,UDP_DST | COPY | Copies DHCP packets for node location |
| com.hp.sdn.infra.filter | 34000 | 0xffff | IN_PORT | GOTO | Filter traffic from infrastructure ports |
| com.hp.sdn.initial | 65500 | 0xffff | | FORWARD | Initial flow sent during pipeline setup |
| com.hp.sdn.ip.normal | 1 | 0xffff | | FORWARD | Default treatment of IPv4 flow misses via normal switch-based forwarding |
| com.hp.sdn.macgrp.dst | 30901 | 0xffff1 | ETH_DST | GOTO | Writes metadata for packet matching source or destination Mac address in the Mac ... |
| com.hp.sdn.macgrp.src | 30900 | 0xffff0 | ETH_SRC | GOTO | Writes metadata for packet matching source or destination Mac address in the Mac ... |
| com.hp.sdn.normal | 0 | 0xffff | | FORWARD,GOTO | Default treatment of flow misses via normal switch-based forwarding |

Εικόνα 8-72 Πεδίο Openflow classes

Στο πεδίο Openflow classes βλέπουμε τις κινήσεις που μπορεί να κάνει ο controller. Έχει τη δυνατότητα COPY να αντιγράψει ARP αιτήσεις από κάποιο κόμβο που θέλει να επικοινωνήσει.

8.5.2 Προσθήκη Δρομολογητή που επικοινωνεί με τον διασυνδεδεμένο μηχανισμό

Το επόμενο βήμα που κάναμε ήταν να προσθέσουμε δύο δρομολογητές στο κεντρικό switch που διαμοιράζει την ip του τοπικού δικτύου μας. Ο δρομολογητής R2 είναι συνδεδεμένος directly στο κεντρικό switch και ο δρομολογητής R3 παίρνει την IP του τοπικού δικτύου μέσα από δύο άλλους μεταγωγείς. Αυτό που ουσιαστικά θέλουμε σε αυτό το σημείο δεν είναι να επικοινωνεί μόνο με την IP του Mininet αλλά και με τους εσωτερικούς hosts του Mininet που δημιουργούνται κάθε φορά για την εκάστοτε τοπολογία. Το μόνο ζήτημα είναι ότι κλείνοντας την τοπολογία στο Mininet δεν αποθηκεύει τις αλλαγές που θα κάνουμε για να πετύχουμε την επικοινωνία. Ως επακόλουθο τρέχουμε κάθε φορά μία αλληλουχία εντολών για να επιτύχουμε το επιθυμητό αποτέλεσμα.



Εικόνα 8-73 Τοπολογία με προσθήκη δύο δρομολογητών

Αυτό που έπρεπε να κάνουμε και στους δύο δρομολογητές για να επιτύχουμε την απόδοση IP διεύθυνσης από το τοπικό δίκτυο είναι να ενεργοποιήσουμε την λειτουργία του DHCP πρωτοκόλλου έτσι ώστε να δεσμευτούν δυναμικά IP διευθύνσεις.

Γράφοντας τον παρακάτω κώδικα και στους δύο δρομολογητές περιμένουν να πάρουν IP διεύθυνση δυναμικά, την οποία και αποκτούν.

Δρομολογητής R2

```
conf t
int g1/0
ip address dhcp
no shut
exit
exit
wr
```

Δρομολογητής R3

```
conf t
int f0/0
```



ip address dhcp

no shut

exit

exit

wr

Έχοντας εκτελέσει τον αντίστοιχο κώδικα παρατηρούμε ότι παίρνουν IP διαμέσων DHCP χρησιμοποιώντας την IP του τοπικού μας δικτύου.

```
R3
*Aug 14 12:31:54.859: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Aug 14 12:31:55.959: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up[OK]
R3#
*Aug 14 12:32:07.119: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP address 192.168.1.17, ma
sk 255.255.255.0, hostname R3

R2
[conf]
Building configuration...
[OK]
R2#
*Aug 14 12:31:08.955: %DHCP-6-ADDRESS_ASSIGN: Interface GigabitEthernet1/0 assigned DHCP address 192.168.1.16,
mask 255.255.255.0, hostname R2
R2#ping 192.168.1.15
```

Εικόνα 8-74 Δυναμική απόδοση IP διευθύνσεων

8.5.2.1 Έλεγχος επικοινωνίας δρομολογητών με τα άλλα εικονικά μηχανήματα που έχουμε εισάγει

Ανοίγουμε όλα τα μηχανήματα περιμένουν να πάρουν IP από τον κεντρικό μεταγωγέα και στην συνέχεια επιχειρούμε pings.



Πings από R2->όλο το δίκτυο και το αντίθετο

The image shows two Oracle VM VirtualBox windows. The left window, titled 'SDN_INTGR [Σε λειτουργία] - Oracle VM VirtualBox', displays the configuration of the 'eth0' interface on a Linux host. The configuration includes the MAC address '08:00:27:08:df:4d', IP address '192.168.1.14', and subnet mask '255.255.255.0'. The right window, titled 'MN_INTGR [Σε λειτουργία] - Oracle VM VirtualBox', shows the configuration of the 'eth0' interface on a Windows host with IP address '192.168.1.12' and subnet mask '255.255.255.0'. Below these windows, a console window for 'R2' shows the results of ping tests: 'R2#ping 192.168.1.14' (Success rate is 100 percent (5/5), round-trip min/avg/max = 4/9/12 ms), 'R2#ping 192.168.1.15' (Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/12 ms), and 'R2#ping 192.168.1.12' (Success rate is 100 percent (5/5), round-trip min/avg/max = 8/12/20 ms). A network diagram on the right shows 'Eth0netswitch-3' connected to 'R3'.

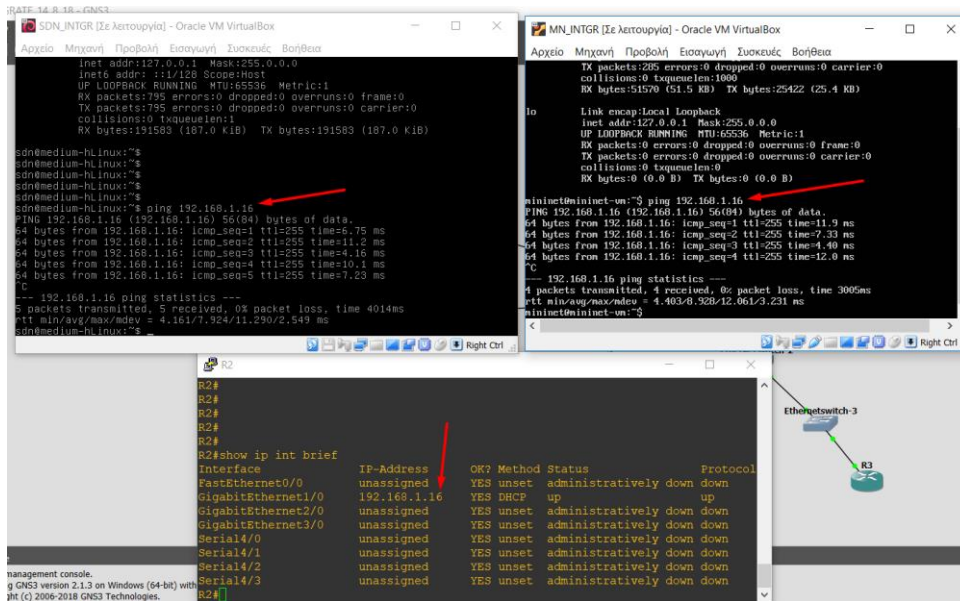
Internet_ProvINTGR [Σε λειτουργία] - Oracle VM VirtualBox

Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια

The image shows a Windows command prompt window titled 'C:\Windows\system32\cmd.exe'. The user has entered the command 'ipconfig'. The output shows the configuration for the 'Ethernet adapter Local Area Connection:' with the following details: 'Connection-specific DNS Suffix : Home', 'Link-local IPv6 Address : Fe80::b89c:4e77:50a8:931e%11', 'IPv4 Address : 192.168.1.12', 'Subnet Mask : 255.255.255.0', and 'Default Gateway : 192.168.1.1'. A red arrow points to the IPv4 address '192.168.1.12'. The output for the 'Tunnel adapter isatap.Home:' shows 'Media State : Media disconnected' and 'Connection-specific DNS Suffix : Home'.

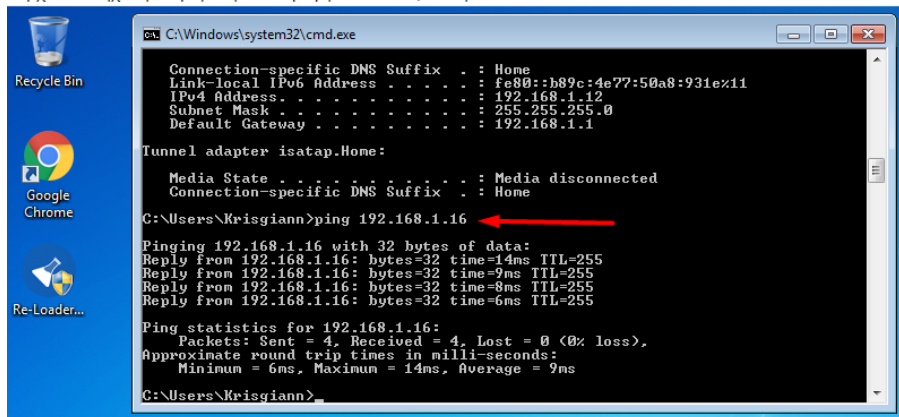
Εικόνα 8-75 Pings R2 δρομολογήτη προς τις εικονικές μηχανές

Στην παραπάνω εικόνα εκτελούμε pings από τον δρομολογητή R2 προς την IP των μηχανών που έχουν IP του τοπικού μας δικτύου.



Internet_ProvINTGR [Σε λειτουργία] - Oracle VM VirtualBox

Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια



Εικόνα 8-76 Ping από τις εικονικές μηχανές προς τον R2

Στην παραπάνω εικόνα εκτελούμε pings προς τον δρομολογητή R2 από τις εικονικές μηχανές



Pings από R3->όλο το δίκτυο και το αντίθετο

The screenshot shows three Oracle VM VirtualBox windows. The top-left window is titled 'SDN_INTGR [Σε λειτουργία] - Oracle VM VirtualBox' and displays network configuration for 'sdn@medium-hl.linux:~\$'. It shows the configuration for 'eth0' (IP: 192.168.1.14) and 'lo' (IP: 127.0.0.1). The top-right window is titled 'MIN_INTGR [Σε λειτουργία] - Oracle VM VirtualBox' and displays network configuration for 'mininet@mininet-um:~\$'. It shows the configuration for 'eth0' (IP: 192.168.1.15) and 'lo' (IP: 127.0.0.1). The bottom window is titled 'R3' and shows the output of ping commands from R3 to 192.168.1.14, 192.168.1.15, and 192.168.1.12. All pings are successful with a 100% success rate and a round-trip time of approximately 8ms.

Internet_ProvINTGR [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εξαγωγή Συσκευές Βοήθεια

The screenshot shows a Windows command prompt window titled 'C:\Windows\system32\cmd.exe'. The user has entered the command 'ipconfig'. The output shows the configuration for the 'Ethernet adapter Local Area Connection':
Connection-specific DNS Suffix . : Home
Link-local IPv6 Address : fe80:b89c:4e77:50a8:931e:x11
IPv4 Address. : 192.168.1.12
Subnet Mask : 255.255.255.0
Default Gateway : 192.168.1.1
The configuration for the 'Tunnel adapter isatap.Home' is also shown, indicating it is disconnected.

Εικόνα 8-77 Pings R3 δρομολογητή προς τις εικονικές μηχανές

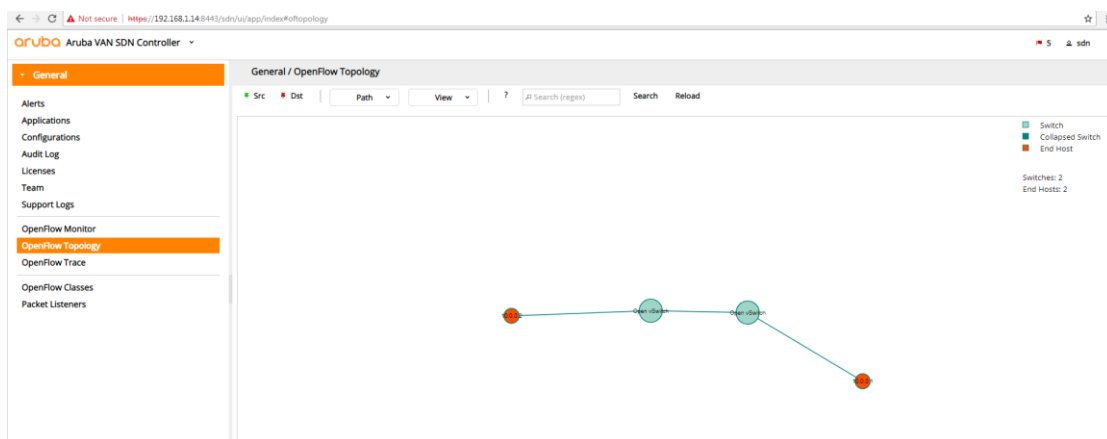
Στην παραπάνω εικόνα εκτελούμε pings από τον δρομολογητή R3 προς την IP των μηχανών που έχουν IP του τοπικού μας δικτύου.



```
sudo mn --controller=remote,ip=192.168.1.14 --topo=linear,2  
--switch=ovsk,protocols=OpenFlow13 --mac
```

```
MN_INTGR-1  
*** Adding controller  
Unable to contact the remote controller at 192.168.1.14:6653  
Connecting to remote controller at 192.168.1.14:6633  
*** Adding hosts:  
h1 h2  
*** Adding switches:  
s1 s2  
*** Adding links:  
(h1, s1) (h2, s2) (s2, s1)  
*** Configuring hosts  
h1 h2  
*** Starting controller  
c0  
*** Starting 2 switches  
s1 s2 ...  
*** Starting CLI:  
mininet> dump  
<Host h1: h1-eth0:10.0.0.1 pid=1393>  
<Host h2: h2-eth0:10.0.0.2 pid=1395>  
<OVSSwitch{'protocols': 'OpenFlow13'} s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=1400>  
<OVSSwitch{'protocols': 'OpenFlow13'} s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None pid=1403>  
<RemoteController{'ip': '192.168.1.14'} c0: 192.168.1.14:6633 pid=1385>  
mininet> pingall  
*** Ping: testing ping reachability  
h1 -> h2  
h2 -> h1  
*** Results: 0% dropped (2/2 received)  
mininet>
```

Εικόνα 8-79 Δημιουργία τοπολογίας στον Mininet



Εικόνα 8-80 Οπτικοποίηση τοπολογίας στον Aruba SDN



2^ο Βήμα Δημιουργία bridge σε διαφορετικό παράθυρο putty

*sudo ovs-vsctl add-port s1 eth0 /*Με την εντολή αυτή γεφυρώνουμε τον h1 με το s1*/*

```
MN_INTGR [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
o Link encap:Local Loopback
  inet addr:127.0.0.1 Mask:255.0.0.0
  UP LOOPBACK RUNNING MTU:65536 Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

ininet@mininet-vm:~$ ping 192.168.1.17
PING 192.168.1.17 (192.168.1.17) 56(84) bytes of data:
4 bytes from 192.168.1.17: icmp_seq=1 ttl=255 time=10.8 ms
4 bytes from 192.168.1.17: icmp_seq=2 ttl=255 time=7.67 ms
4 bytes from 192.168.1.17: icmp_seq=3 ttl=255 time=3.71 ms
4 bytes from 192.168.1.17: icmp_seq=4 ttl=255 time=11.6 ms
4 bytes from 192.168.1.17: icmp_seq=5 ttl=255 time=8.68 ms
4 bytes from 192.168.1.17: icmp_seq=6 ttl=255 time=5.75 ms
C
-- 192.168.1.17 ping statistics --
  packets transmitted, 6 received, 0% packet loss, time 5008ms
  tt min/avg/max/mdev = 3.710/8.048/11.612/2.749 ms
ininet@mininet-vm:~$
ininet@mininet-vm:~$ sudo ovs-vsctl add-port s1 eth0
ininet@mininet-vm:~$
```

Εικόνα 8-81 Γεφύρωση του h1 με το s1

3^ο Βήμα Παραμετροποίηση IP στο h1 από άλλο παράθυρο putty

Πριν κάνουμε αυτήν την παραμετροποίηση βλέπουμε ότι ο h1 με ip 10.0.0.1 κάνει ping στον h2 με ip(10.0.0.2) μετά την παραμετροποίηση δεν θα μπορεί

```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.368 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.044 ms
^C
```

Εικόνα 8-82 Επικοινωνία πριν την παραμετροποίηση

Έπειτα θα τρέξουμε αυτές τις δύο εντολές όπου θα θέσουμε στατικά την IP του h1 σε 192.168.1.20 του τοπικού μας δικτύου. Στην συνέχεια θα δώσουμε default-gateway και θα διαφημίσουμε τον h1



Pings R2 προς h1(IP=192.168.1.20)

Στην παρακάτω εικόνα βλέπουμε ότι ο h1 με τον R1 είναι πλήρως διασυνδεδεμένοι και επικοινωνούν .

```
R2
R2#
R2#show ip int brief
Interface      IP-Address      OK? Method Status          Protocol
FastEthernet0/0 unassigned      YES unset  administratively down down
GigabitEthernet1/0 192.168.1.16   YES DHCP    up              up
GigabitEthernet2/0 unassigned      YES unset  administratively down down
GigabitEthernet3/0 unassigned      YES unset  administratively down down
Serial4/0         unassigned      YES unset  administratively down down
Serial4/1         unassigned      YES unset  administratively down down
Serial4/2         unassigned      YES unset  administratively down down
Serial4/3         unassigned      YES unset  administratively down down
R2#
R2#ping 192.168.1.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.20, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
R2#ping 192.168.1.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.20, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
R2#
```

Εικόνα 8-85 Pings R2->h1



IP που θα γίνουν Ping

SDN Controller: 192.168.1.14

R3 Δρομολογητής: 192.168.1.17

Εσωτερικός browser: 192.168.1.12

```
mininet> h1 ping 192.168.1.17
PING 192.168.1.17 (192.168.1.17) 56(84) bytes of data.
 64 bytes from 192.168.1.17: icmp_seq=1 ttl=255 time=9.75 ms
 64 bytes from 192.168.1.17: icmp_seq=2 ttl=255 time=5.61 ms
 64 bytes from 192.168.1.17: icmp_seq=3 ttl=255 time=12.3 ms
^C
--- 192.168.1.17 ping statistics ---
 3 packets transmitted, 3 received, 0% packet loss, time 2004ms
 rtt min/avg/max/mdev = 5.614/9.238/12.350/2.774 ms
mininet> h1 ping 192.168.1.12
PING 192.168.1.12 (192.168.1.12) 56(84) bytes of data.
 64 bytes from 192.168.1.12: icmp_seq=1 ttl=128 time=1.34 ms
 64 bytes from 192.168.1.12: icmp_seq=2 ttl=128 time=1.25 ms
 64 bytes from 192.168.1.12: icmp_seq=3 ttl=128 time=1.06 ms
^C
--- 192.168.1.12 ping statistics ---
 3 packets transmitted, 3 received, 0% packet loss, time 2002ms
 rtt min/avg/max/mdev = 1.066/1.222/1.349/0.120 ms
mininet> h1 ping 192.168.1.14
PING 192.168.1.14 (192.168.1.14) 56(84) bytes of data.
 64 bytes from 192.168.1.14: icmp_seq=1 ttl=64 time=2.05 ms
 64 bytes from 192.168.1.14: icmp_seq=2 ttl=64 time=1.19 ms
 64 bytes from 192.168.1.14: icmp_seq=3 ttl=64 time=1.05 ms
^C
--- 192.168.1.14 ping statistics ---
 3 packets transmitted, 3 received, 0% packet loss, time 2002ms
 rtt min/avg/max/mdev = 1.051/1.431/2.052/0.444 ms
```

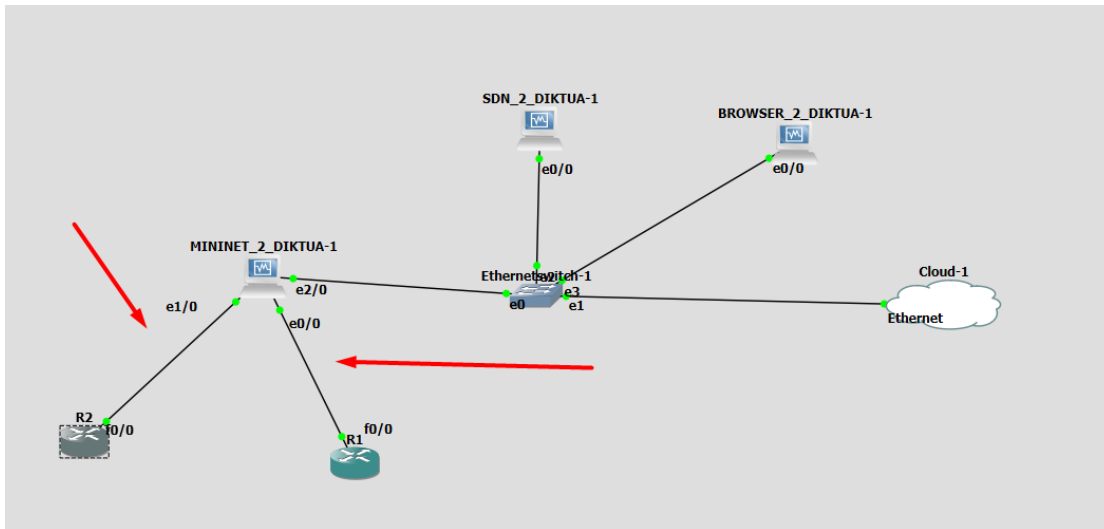
Εικόνα 8-86 Επιτυχής επικοινωνία του εσωτερικού host με το υπόλοιπο δίκτυο

Παρατήρηση

Αυτό που είδαμε σε αυτήν την τοπολογία ήταν ότι καθώς βλέπαμε ότι αλλάζαμε κανονικά την IP του h1 έτσι ώστε να επικοινωνούμε με το υπόλοιπο δίκτυο χανόταν η τοπολογία στο γραφικό περιβάλλον του SDN Controller. Θέλαμε να βελτιώσουμε και να τελειοποιήσουμε τον μηχανισμό μας οπότε δουλέψαμε το επόμενο επίπεδο σε αυτήν την τοπολογία που θα παρουσιαστεί στην παρακάτω ενότητα.

8.6 Υλοποίηση 3^{ης} Τοπολογίας Διασυνδεδεμένου μηχανισμού

Σε αυτό το σημείο θα δημιουργήσουμε παρόμοια τοπολογία με παραπάνω με κάποιες ζωτικής σημασίας όμως αλλαγές.



Εικόνα 8-87 3^η τοπολογία διασύνδεση Mininet με δύο δρομολογητές

Σε αυτήν την τοπολογία βλέπουμε ότι έχουμε στο Mininet 3 θύρες αυτή την φορά την e2/0, e1/0, e0/0. Τα υπόλοιπα εικονικά μηχανήματα δεν παρουσιάζουν κάποια αλλαγή.

Τα βήματα που ακολουθήσαμε για να πετύχουμε μια πιο ολοκληρωμένη διασύνδεση του Mininet με το περιβάλλον του GNS3 και σε οπτικό επίπεδο φαίνονται παρακάτω

1^ο Βήμα

Πρώτα έπρεπε να πάμε στο περιβάλλον του GNS3 και προσθέσουμε επιπλέον adapters στην εικονική μηχανή του Mininet. Αυτό που κάναμε είναι να πάμε στο device του Mininet να κάνουμε δεξί κλικ->Configure->Network->Adapters και βάζουμε τον αριθμό 4.

2^ο Βήμα

Αυτό που έπρεπε να κάνουμε στη συνέχεια ήταν να πάμε στο περιβάλλον του Mininet και να γράψουμε ορισμένα scripts τα οποία θα καθόριζαν ποιες πόρτες θα είχαν στατική IP για να επικοινωνούν με τους routers και ποια πόρτα θα έπαιρνε IP με DHCP για να συνδεθεί με το physical network.

Γράφοντας στο παρακάτω αρχείο με path όπως έχουμε αναφέρει και πιο πάνω:

```
sudo vi /etc/network/interfaces
```



```
MININET_2_DIKTUA [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth2
iface eth2 inet dhcp

auto eth0
iface eth0 inet static
address 10.0.0.5
netmask 255.255.255.0
network 10.0.0.0
broadcast 10.0.0.255

auto eth1
iface eth1 inet static
address 10.0.0.6
netmask 255.255.255.0
network 10.0.0.0
broadcast 10.0.0.255
^C
^C
^C
"/etc/network/interfaces" 24L, 492C                               1,1          all
```

Εικόνα 8-88 Script διασύνδεσης με router

Στο παραπάνω script όπως βλέπουμε και από την τοπολογία στην eth2 θύρα λέμε στο Mininet να εκτελεί DHCP Client λειτουργία. Στις άλλες δύο πόρτες eth0 και eth1 δίνουμε στατική IP του δικτύου 10.0.0.0 που τρέχει by default το Mininet έτσι ώστε να επιτευχθεί η ζεύξη με τους δρομολογητές R1 και R2 αντίστοιχα. Βάζουμε IP για το eth0 την 10.0.0.5 και για το eth1 10.0.0.6 και βέβαια network 10.0.0.0. Βάζουμε αυτές τις δύο διότι το Mininet χρησιμοποιεί τις πρώτες δύο στους h1 και h2 αντίστοιχα.

3^ο Βήμα

Στην συνέχεια κάνουμε ένα reboot το Mininet για να ενημερωθεί από τις αλλαγές που κάναμε και στη συνέχεια τρέχουμε την παρακάτω εντολή για να δημιουργήσουμε μία τοπολογία που χρησιμοποιεί δύο switches συνδεδεμένα μεταξύ τους, έχοντας έναν host ο καθένας. Σαν IP του Controller βάζουμε αυτή που δίνει η εικονική μηχανή του Aruba SDN καθώς ανοίγει, στην προκειμένη είναι 192.168.1.22.

```
sudo mn --controller=remote,ip=192.168.1.22 --topo=linear,2
--switch=ovsk,protocols=OpenFlow13 --mac
```



4^ο Βήμα

Εδώ θα πρέπει να δημιουργήσουμε μία γεφύρωση του κάθε switch στο εικονικό περιβάλλον με την πόρτα στο φυσικό περιβάλλον όπου είναι η *eth0* και η *eth1*. Αυτό θα το υλοποιήσουμε με την παρακάτω εντολή σε κάποιο άλλο session καθώς στο κύριο session τρέχει η τοπολογία μας και δεν μπορούμε να τρέξουμε εντολή *sudo*.

Πρόσθεση *eth0* στο Mininet switch 1

```
sudo ovs-vsctl add-port s1 eth0
```

Πρόσθεση *eth1* στο Mininet switch 2

```
sudo ovs-vsctl add-port s2 eth1
```

Τρέχοντας την εντολή *sudo ovs-vsctl show* βλέπουμε την επιτυχής γεφύρωση με τις πόρτες *eth0* και *eth1*

```
Bridge "s2"  
  Controller "ptcp:6655"  
  Controller "tcp:192.168.1.22:6633"  
    is_connected: true  
  fail_mode: secure  
  Port "eth1"  
    Interface "eth1"  
  Port "s2-eth1"  
    Interface "s2-eth1"  
  Port "s2-eth2"  
    Interface "s2-eth2"  
  Port "s2"  
    Interface "s2"  
      type: internal  
Bridge "s1"  
  Controller "tcp:192.168.1.22:6633"  
    is_connected: true  
  Controller "ptcp:6654"  
  fail_mode: secure  
  Port "s1-eth2"  
    Interface "s1-eth2"  
  Port "s1-eth1"  
    Interface "s1-eth1"  
  Port "eth0"  
    Interface "eth0"  
  Port "s1"  
    Interface "s1"  
      type: internal  
  ovs_version: "2.0.2"  
mininet@mininet-vm:~$
```

Εικόνα 8-89 Επιτυχής γεφύρωση εικονικών switches με physical ports



5^ο Βήμα

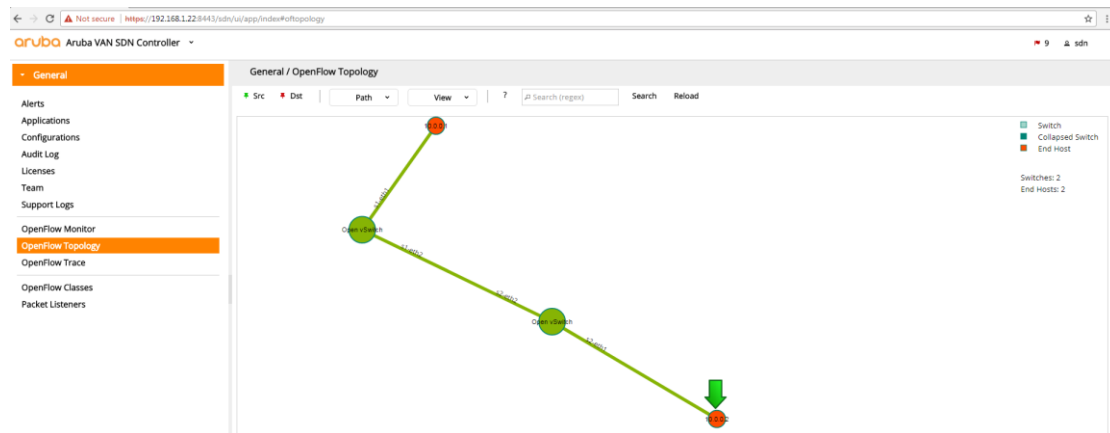
Σε αυτό το βήμα συνδέουμε τους δρομολογητές επάνω στο Mininet όπως ακριβώς φαίνεται στην εικόνα της τοπολογίας μας και δίνουμε στατικά IP στις fa0/0 πόρτες των δρομολογητών του δικτύου 10.0.0.0

IP R1: 10.0.0.3

IP R2:10.0.0.4

6^ο Βήμα

Σε αυτό το σημείο θα ελέγξουμε την λειτουργία του δικτύου μας. Πριν γίνουν pings με τους εξωτερικούς routers R1&R2 η τοπολογία του Mininet μας εμφανίζεται στον controller όπως φαίνεται στην παρακάτω εικόνα



Εικόνα 8-90 Εικόνα από Aruba SDN πριν μάθει τους εξωτερικούς δρομολογητές



Έπειτα θα πάμε στο putty που τρέχουμε την τοπολογία και θα στείλουμε πακέτα προς τους εξωτερικούς routers. Στην παρακάτω εικόνα στέλνουμε 3 πακέτα προς τον δρομολογητή R3, στον R2 και στον εσωτερικό host 2 και παραδίδονται επιτυχώς.

```
mininet> h1 ping -c 3 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=255 time=2.87 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=255 time=10.0 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=255 time=8.18 ms

--- 10.0.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 2.877/7.021/10.003/3.024 ms
mininet> h1 ping -c 3 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=255 time=10.5 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=255 time=7.99 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=255 time=3.30 ms

--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 3.306/7.294/10.587/3.014 ms
mininet> h1 ping -c 3 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.517 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.042 ms

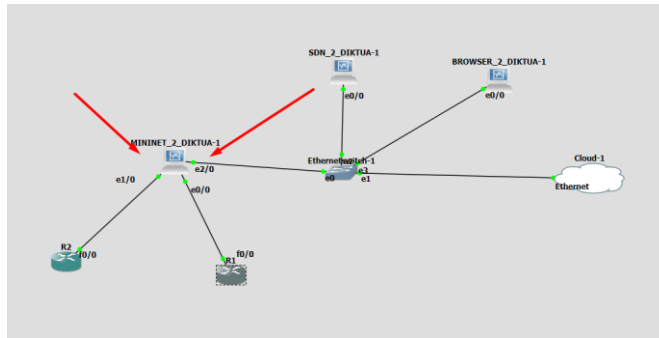
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.041/0.200/0.517/0.224 ms
mininet>
```

Εικόνα 8-91 Επιτυχής επικοινωνία των εσωτερικών hosts με τους εξωτερικούς routers

Pings R1 προς h1,h2 και διαμέσων του Mininet στον R2

```
R1#ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/30/104 ms
R1#ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
R1#ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/9/12 ms
R1#ping 10.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/16/28 ms
R1#
```

Εικόνα 8-92 Επιτυχής επικοινωνία του R1 ->h1,h2 και R1->R2



Εικόνα 8-93 Πέρασμα πακέτων διαμέσων του Mininet

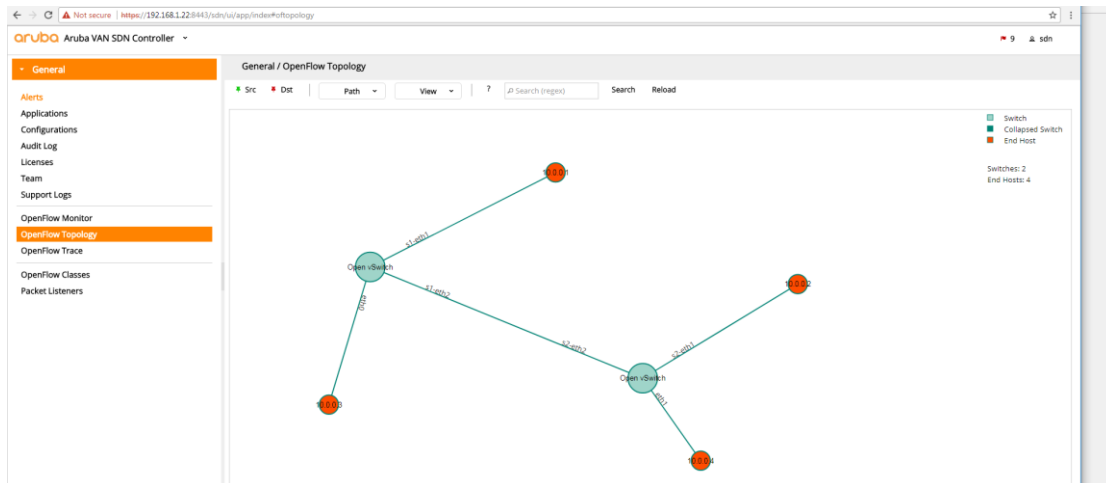
Σε αυτό το σημείο τονίζουμε ότι επικοινωνία R1->R2 και R2->R1 γίνεται διαμέσων του εξομοιωτή Mininet

Pings R2 προς h1,h2 και διαμέσων του Mininet στον R1

```
R2#ping 10.0.0.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/11/12 ms
R2#ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/12 ms
R2#ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/12 ms
R2#
```

Εικόνα 8-94 Επιτυχής επικοινωνία του R2 ->h1,h2 και R2->R1

Τέλος κάνοντας ένα reload στον Controller που τρέχει στον εσωτερικό browser της τοπολογίας μας βλέπουμε ότι η τοπολογία μας ανανεώθηκε και ο Controller έμαθε και τους εξωτερικούς δρομολογητές του φυσικού μας δικτύου αλλάζοντας τον αριθμό των host του σε 4,προσθέτοντας στο s1 τον δρομολογητή R1 και στον s2 τον δρομολογητή R2,πράγμα που μας ενημερώνει ότι ο μηχανισμός μας δουλεύει σε ακόμη καλύτερο επίπεδο από ότι στις προηγούμενες τοπολογίες.



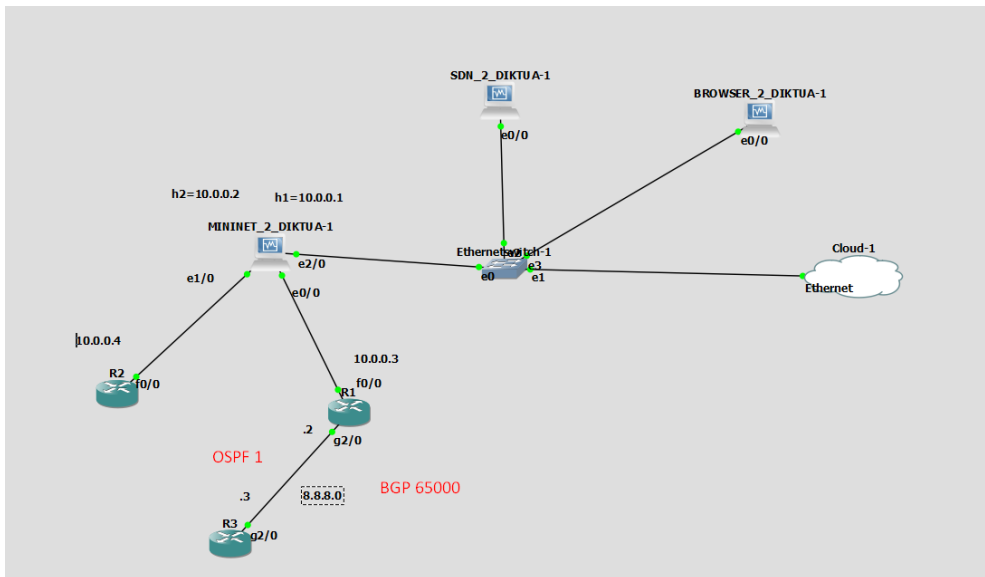
Εικόνα 8-95 Ενημερωμένη τοπολογία με 4 hosts στον Controller

8.6.1 Συνέχεια Τοπολογίας Προσθήκη Δρομολογητή που συνδέεται διαμέσων ενός άλλου με το Mininet και τους hosts.

Θέλοντας να προχωρήσουμε ένα βήμα την τοπολογία και να δούμε μέχρι που μπορεί να επεκταθεί κατά κάποιο τρόπο προσθέσαμε ένα ακόμη δρομολογητή κάτω από τον R1 και προσπαθήσαμε μέσω κάποιων πρωτοκόλλων να διαφημίσουμε αυτόν τον δρομολογητή στους εσωτερικούς hosts του Mininet. Η νέα μορφή της τοπολογίας φαίνεται στην παρακάτω εικόνα

Αυτό που έπρεπε να κάνουμε είναι να τρέξουμε το BGP και το OSPF πρωτόκολλο γειτνίασης έτσι ώστε να μάθει ο ένας δρομολογητής τα δίκτυα του άλλου.

Θα πρέπει να εκχωρήσουμε στατικά τις IP 8.8.8.2 και 8.8.8.3 στους δρομολογητές R2 και R3 αντίστοιχα.



Εικόνα 8-96 Μορφή τοπολογίας με ένα ακόμη δρομολογητή

Στην τοπολογία τώρα θα πρέπει να ενεργοποιήσουμε τα πρωτόκολλα γειτνίασης που αναφέραμε και πιο πάνω. Θα πάμε πρώτα στον δρομολογητή R1 που ουσιαστικά θα κάνει το redistribution έτσι ώστε να επιτευχθεί επικοινωνία με τους host του Mininet.

Διαδικασία Ενεργοποίησης BGP-OSPF

R1 Δρομολογητής

```
conf t
```

```
int lo0
```

```
ip add 1.1.1.1 255.255.255.255 /*Προσθήκη loopback για να μένει σταθερό και να μην επηρεάζεται η κατάσταση του*/
```

```
no shut
```

```
exit
```

```
int g2/0
```

```
ip add 8.8.8.2 255.255.255.0
```

```
no shut
```

```
end
```



R3 Δρομολογητής

```
conf t
int lo0
ip add 3.3.3.3 255.255.255.255 /*Προσθήκη loopback για δεν αλλάζει ποτέ την
κατάσταση του*/
no shut
exit
int g2/0
ip add 8.8.8.3 255.255.255.0
no shut
end
```

R1 Δρομολογητής

```
conf t
router bgp 65000
neighbor 3.3.3.3 remote-as 65000
network 8.8.8.0 mask 255.255.255.0
end
```

R3 Δρομολογητής

```
conf t
router bgp 65000
neighbor 1.1.1.1 remote-as 65000
neighbor 1.1.1.1 update-source loopback 0
network 1.1.1.1 mask 255.255.255.255 /*Μόνο το loopback διαφημίζω*/
```



OSPF Πρωτόκολλο

R3 Δρομολογητής

```
conf t
router ospf 1
network 8.8.8.3 0.0.0.0 area 0
network 3.3.3.3 0.0.0.0 area 0
```

R1 Δρομολογητής

```
conf t
router ospf 1
network 8.8.8.2 0.0.0.0 area 0
network 1.1.1.1 0.0.0.0 area 0
network 10.0.0.3 0.0.0.0 area 0
end
```

Έχοντας τρέξει τα πρωτόκολλα αυτά έχουμε επιτύχει διασύνδεση του δρομολογητή R3 και στα δύο δίκτυα που έχει ο R1, πιο αναλυτικά ο R3 κάνει ping και στην g2/0 με ip 8.8.8.2 και στην f0/0 με IP 10.0.0.3

```
R3#ping 10.0.0.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/18/40 ms
R3#ping 8.8.8.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/12/16 ms
R3#
```

Εικόνα 8-97 Επικοινωνία R3-R1



Στη συνέχεια αυτό που θέλαμε να κάνουμε είναι να επιτύχουμε επικοινωνία με τους host του Mininet(h1,h2). Αυτό που σκεφτήκαμε να κάνουμε είναι να διαφημίσουμε τις ip των hosts του μέσα από το Mininet.

Γνωρίζοντας ότι οι hosts έχουν IP του δικτύου 10.0.0.0/24 κάναμε τα παρακάτω static routes για να επιτύχουμε επικοινωνία με τον R3.

```
h1 route add default gw 10.0.0.1 h1-eth0
```

```
h2 route add default gw 10.0.0.2 h2-eth0
```

Έχοντας κάνει αυτή τη δρομολόγηση θα πάμε να ελέγξουμε την επικοινωνία των hosts του Mininet πρώτα στην πόρτα του R1 που έχει δίκτυο 8.8.8.2 και έπειτα στην πόρτα του R3 που έχει δίκτυο 8.8.8.3.

Στην παρακάτω εικόνα βλέπουμε ότι ο h1 επικοινωνεί επιτυχώς με τους R1 και R3 που τρέχουν μεταξύ τους πρωτόκολλα BGP και OSPF.

```
mininet> h1 ping -c 8.8.8.2
Usage: ping [-aAbBdDfhInOqrRUvV] [-c count] [-i interval] [-I interface]
          [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
          [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
          [-w deadline] [-W timeout] [hop1 ...] destination
mininet> h1 ping -c 3 8.8.8.2
PING 8.8.8.2 (8.8.8.2) 56(84) bytes of data.
64 bytes from 8.8.8.2: icmp_seq=1 ttl=255 time=28.5 ms
64 bytes from 8.8.8.2: icmp_seq=2 ttl=255 time=4.02 ms
64 bytes from 8.8.8.2: icmp_seq=3 ttl=255 time=10.8 ms

--- 8.8.8.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 4.024/14.465/28.500/10.310 ms
mininet> h1 ping -c 3 8.8.8.3
PING 8.8.8.3 (8.8.8.3) 56(84) bytes of data.
64 bytes from 8.8.8.3: icmp_seq=1 ttl=254 time=40.7 ms
64 bytes from 8.8.8.3: icmp_seq=2 ttl=254 time=26.5 ms
64 bytes from 8.8.8.3: icmp_seq=3 ttl=254 time=22.6 ms

--- 8.8.8.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 22.628/29.986/40.737/7.773 ms
mininet>
```

Εικόνα 8-98 Επικοινωνία h1 ->R1 & h1->R3



Στην παρακάτω εικόνα βλέπουμε ότι ο h2 επικοινωνεί επιτυχώς με τους R1 και R3 που τρέχουν μεταξύ τους πρωτόκολλα BGP και OSPF.

```
mininet> h2 ping -c 3 8.8.8.3
PING 8.8.8.3 (8.8.8.3) 56(84) bytes of data.
64 bytes from 8.8.8.3: icmp_seq=1 ttl=254 time=30.3 ms
64 bytes from 8.8.8.3: icmp_seq=2 ttl=254 time=26.6 ms
64 bytes from 8.8.8.3: icmp_seq=3 ttl=254 time=32.4 ms

--- 8.8.8.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 26.608/29.801/32.487/2.435 ms
mininet> h2 ping -c 3 8.8.8.2
PING 8.8.8.2 (8.8.8.2) 56(84) bytes of data.
64 bytes from 8.8.8.2: icmp_seq=1 ttl=255 time=9.67 ms
64 bytes from 8.8.8.2: icmp_seq=2 ttl=255 time=7.10 ms
64 bytes from 8.8.8.2: icmp_seq=3 ttl=255 time=4.12 ms

--- 8.8.8.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 4.123/6.968/9.672/2.267 ms
mininet>
```

Εικόνα 8-99 Επικοινωνία h2 ->R1 & h2->R3

Στην παρακάτω εικόνα βλέπουμε ότι ο R3 επικοινωνεί επιτυχώς με τους h1 και h2 που τους μαθαίνει διαμέσων του δρομολογητή R1 ανταλλάσσοντας πίνακες δρομολόγησης μεταξύ τους με τα πρωτόκολλα BGP και OSPF.

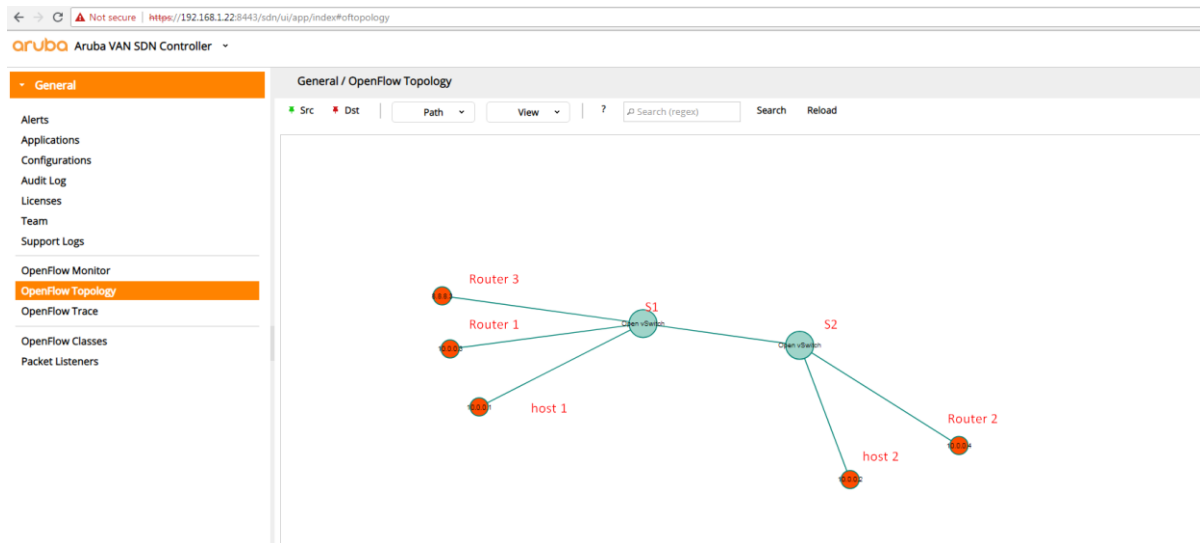
```
R3#ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
R3#ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
R3#
```

Εικόνα 8-100 Επικοινωνία R3->h1 & R3->h2

Τέλος θα πάμε στον εσωτερικό browser της τοπολογίας μας όπου υπάρχει ο SDN Controller και θα ελέγξουμε ότι το Mininet έχει προσθέσει στο switch 1 τον Router 1 που αρχικά είχε γίνει η



διασύνδεση του με τους hosts και τον Router 3 που είναι διαφορετικού δικτύου και επικοινωνεί με τους hosts διαμέσων του R1.

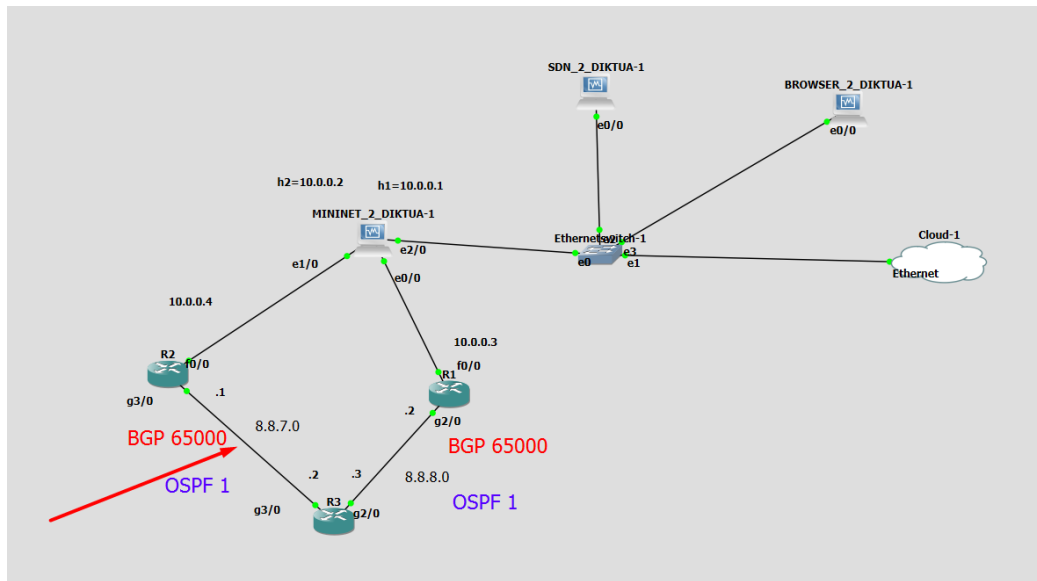


Εικόνα 8-101 Τοπολογία στον SDN με δρομολογητή διαφορετικού δικτύου

8.6.2 Προσθήκη επιπλέον ζεύξης-δικτύου για πλήρη επικοινωνία R2-R3

Η παρατήρηση που κάναμε στην προηγούμενη τοπολογία ήταν ότι ο Δρομολογητής R2 δεν επικοινωνούσε με τον δρομολογητή R3. Για αυτό το λόγο προσθέσαμε μία επιπλέον ζεύξη και ένα ακόμη δίκτυο και τρέξαμε και στους 3 δρομολογητές μας το BGP και το OSPF για την πλήρη αναδιανομή των δικτύων και την απόλυτη επικοινωνία του δικτύου της τοπολογίας μας.

Αυτό που κάνουμε ήταν να προσθέσουμε στατικά IP του δικτύου 8.8.7.0 στις πόρτες g3/0 των δρομολογητών R3 και R2 όπως φαίνεται στην παρακάτω εικόνα της τοπολογίας μας, καθώς και να δώσουμε IP στο loopback του R2 όπως κάναμε και πιο πάνω.



Εικόνα 8-102 Νέα ζεύξη R2-R3

BGP-OSPF στους δρομολογητές R1-R2-R3

Δρομολογητής R3

```
router bgp 65000
neighbor 2.2.2.2 remote-as 65000
neighbor 2.2.2.2 update-source loopback 0
neighbor 1.1.1.1 remote-as 65000
network 8.8.8.0 mask 255.255.255.0
network 8.8.7.0 mask 255.255.255.0
network 3.3.3.3 mask 255.255.255.255
end
```

Δρομολογητής R2

```
router bgp 65000
neighbor 3.3.3.3 remote-as 65000
neighbor 3.3.3.3 update-source loopback 0
network 10.0.0.0 mask 255.255.255.0
network 2.2.2.2 mask 255.255.255.255
```



end

conf t

router ospf 1

network 8.8.7.1 0.0.0.0 area 0

network 2.2.2.2 0.0.0.0 area 0

end

Δρομολογητής R3

conf t

router ospf 1

network 8.8.7.2 0.0.0.0 area 0

network 8.8.8.3 0.0.0.0 area 0

network 3.3.3.3 0.0.0.0 area 0

end

Δρομολογητής R1

router ospf 1

network 1.1.1.1 0.0.0.0 area 0

network 8.8.8.2 0.0.0.0 area 0

end

Έχοντας γράψει τον παραπάνω κώδικα στους δρομολογητές θα ελέγχξουμε την επικοινωνία όλου του δικτύου



8.6.2.1 Έλεγχος επικοινωνίας του δικτύου

R3: $g2/0 = 8.8.8.3/24$

$g3/0 = 8.8.7.2/24$

R2: $f0/0 = 10.0.0.4/24$

$g3/0 = 8.8.7.1/24$

R1: $f0/0 = 10.0.0.3/24$

$g2/0 = 8.8.8.2/24$

Mininet: $h1 = 10.0.0.1$

$H2 = 10.0.0.2$

Pings R3 προς όλες τις IP του δικτύου



```
R3#
R3#
R3#
R3#ping 8.8.7.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.7.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/22/28 ms
R3#ping 8.8.8.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/20/24 ms
R3#ping 8.8.8.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R3#ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/30/32 ms
R3#ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/30/32 ms
R3#ping 10.0.0.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/56 ms
R3#ping 10.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms
R3#
```

Εικόνα 8-103 R3 προς όλες τις IP του δικτύου

Pings R1 προς όλες τις IP του δικτύου

```
R1#ping 8.8.8.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/33/108 ms
R1#ping 8.8.7.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.7.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/21/24 ms
R1#ping 10.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
R1#ping 8.8.7.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.7.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
R1#ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/12 ms
R1#ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
R1#
```

Εικόνα 8-104 R1 προς όλες τις IP του δικτύου



Pings R2 προς όλες τις IP του δικτύου

```
R2#ping 8.8.8.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/58/208 ms
R2#ping 8.8.7.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.7.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/18/24 ms
R2#ping 10.0.0.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/19/24 ms
R2#ping 8.8.8.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/22/28 ms
R2#ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/9/16 ms
R2#ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
R2#
```

Εικόνα 8-105 R2 προς όλες τις IP του δικτύου

Pings h1 του Mininet προς όλες τις IP του δικτύου

```
mininet> h1 ping -c 3 8.8.8.3
PING 8.8.8.3 (8.8.8.3) 56(84) bytes of data:
64 bytes from 8.8.8.3: icmp_seq=1 ttl=254 time=29.1 ms
64 bytes from 8.8.8.3: icmp_seq=2 ttl=254 time=24.5 ms
64 bytes from 8.8.8.3: icmp_seq=3 ttl=254 time=29.5 ms

--- 8.8.8.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 24.582/27.775/29.549/2.262 ms
mininet> h1 ping -c 2 8.8.7.2
PING 8.8.7.2 (8.8.7.2) 56(84) bytes of data:
64 bytes from 8.8.7.2: icmp_seq=1 ttl=254 time=23.0 ms
64 bytes from 8.8.7.2: icmp_seq=2 ttl=254 time=26.4 ms

--- 8.8.7.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 23.018/24.756/26.495/1.745 ms
mininet> h1 ping -c 2 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data:
64 bytes from 10.0.0.4: icmp_seq=1 ttl=255 time=9.50 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=255 time=3.10 ms

--- 10.0.0.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 3.104/6.306/9.508/3.202 ms
mininet> h1 ping -c 2 8.8.7.1
PING 8.8.7.1 (8.8.7.1) 56(84) bytes of data:
64 bytes from 8.8.7.1: icmp_seq=1 ttl=255 time=26.2 ms
64 bytes from 8.8.7.1: icmp_seq=2 ttl=255 time=28.9 ms

--- 8.8.7.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 26.210/27.597/28.984/1.387 ms
mininet> h1 ping -c 2 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data:
64 bytes from 10.0.0.3: icmp_seq=1 ttl=255 time=4.42 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=255 time=11.8 ms

--- 10.0.0.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 4.420/8.159/11.899/3.740 ms
mininet> h1 ping -c 2 8.8.8.2
PING 8.8.8.2 (8.8.8.2) 56(84) bytes of data:
64 bytes from 8.8.8.2: icmp_seq=1 ttl=255 time=6.78 ms
64 bytes from 8.8.8.2: icmp_seq=2 ttl=255 time=9.99 ms

--- 8.8.8.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 6.787/8.390/9.994/1.606 ms
mininet>
```

Εικόνα 8-106 h1 προς όλες τις IP του δικτύου



Pings h2 του Mininet προς όλες τις IP του δικτύου

```
mininet> h2 ping -c 2 8.8.8.3
PING 8.8.8.3 (8.8.8.3) 56(84) bytes of data.
64 bytes from 8.8.8.3: icmp_seq=1 ttl=254 time=27.7 ms
64 bytes from 8.8.8.3: icmp_seq=2 ttl=254 time=32.4 ms

--- 8.8.8.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 27.744/30.112/32.480/2.368 ms
mininet> h2 ping -c 2 8.8.7.2
PING 8.8.7.2 (8.8.7.2) 56(84) bytes of data.
64 bytes from 8.8.7.2: icmp_seq=1 ttl=254 time=28.6 ms
64 bytes from 8.8.7.2: icmp_seq=2 ttl=254 time=23.7 ms

--- 8.8.7.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 23.774/26.204/28.635/2.435 ms
mininet> h2 ping -c 2 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=255 time=9.39 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=255 time=6.03 ms

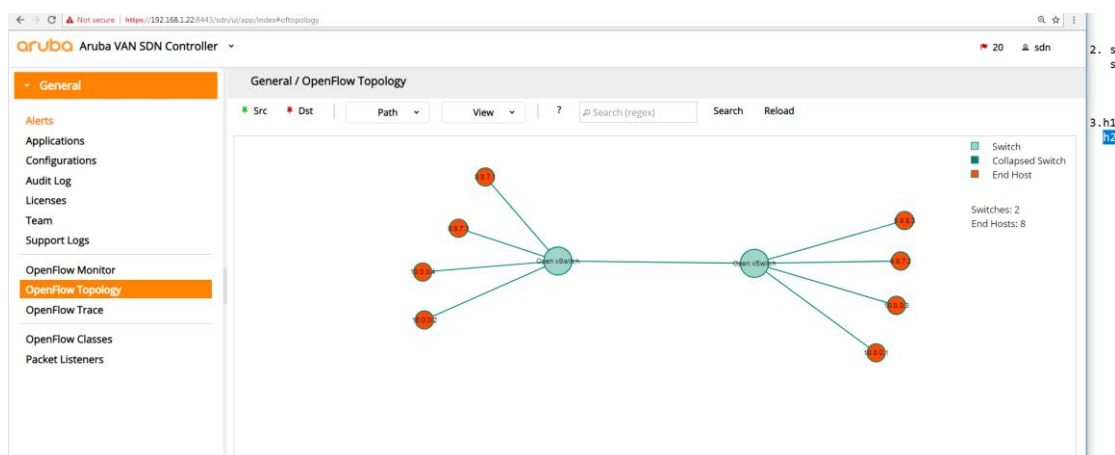
--- 10.0.0.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 6.032/7.712/9.393/1.682 ms
mininet> h2 ping -c 2 8.8.7.1
PING 8.8.7.1 (8.8.7.1) 56(84) bytes of data.
64 bytes from 8.8.7.1: icmp_seq=1 ttl=255 time=8.30 ms
64 bytes from 8.8.7.1: icmp_seq=2 ttl=255 time=7.02 ms

--- 8.8.7.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 7.025/7.662/8.300/0.643 ms
mininet> h2 ping -c 2 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=255 time=11.4 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=255 time=7.00 ms

--- 10.0.0.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 7.004/9.229/11.455/2.227 ms
mininet> h2 ping -c 2 8.8.8.2
PING 8.8.8.2 (8.8.8.2) 56(84) bytes of data.
64 bytes from 8.8.8.2: icmp_seq=1 ttl=255 time=8.70 ms
64 bytes from 8.8.8.2: icmp_seq=2 ttl=255 time=3.15 ms
```

Εικόνα 8-107 h2 προς όλες τις IP του δικτύου

Παρατηρούμε ότι όλα τα πακέτα μεταδίδονται επιτυχώς σε όλη την τοπολογία του δικτύου και αυτό το παρατηρούμε καθώς μπαίνουμε και στον SDN Controller όπου βλέπουμε ότι και το Mininet έμαθε όλους τους δρομολογητές και τους τοποθέτησε στα εικονικά οvs switch που έχει.



Εικόνα 8-108 Εικόνα τοπολογίας μέσα από τον Aruba SDN Controller



9 Εγκατάσταση και Βασική Παραμετροποίηση του πειραματικού δικτυακού εξομοιωτή Core

9.1 Εισαγωγή στο Core

Ο κοινός ανοιχτός ερευνητικός εξομοιωτής (CORE) είναι ένα εργαλείο για την οικοδόμηση εικονικών δικτύων. Ως εξομοιωτής, το CORE χτίζει μια αναπαράσταση ενός πραγματικού δικτύου υπολογιστών που τρέχει σε πραγματικό χρόνο, σε αντίθεση με την προσομοίωση, όπου χρησιμοποιούνται αφηρημένα μοντέλα. Η εξομοίωση ζωντανής μετάδοσης μπορεί να συνδεθεί με φυσικά δίκτυα και δρομολογητές. Παρέχει ένα περιβάλλον για την εκτέλεση πραγματικών εφαρμογών και πρωτοκόλλων, εκμεταλλευόμενοι την εικονικοποίηση που παρέχεται από το *Linux* ή το *FreeBSD* λειτουργικά συστήματα. Εμείς δουλέψαμε στο περιβάλλον των *Linux*.

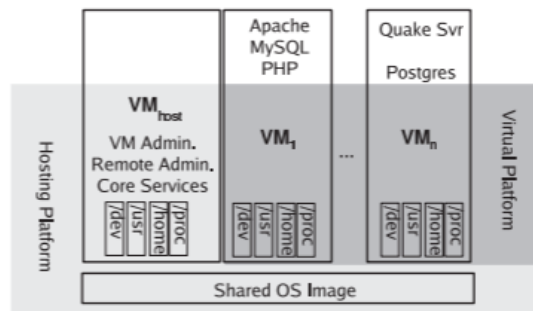
Μερικά από τα βασικά χαρακτηριστικά του είναι:

- αποτελεσματικότητα και επεκτασιμότητα
- εκτελεί εφαρμογές και πρωτόκολλα χωρίς τροποποίηση
- εύκολο στη χρήση GUI
- εξαιρετικά προσαρμόσιμο

Το CORE χρησιμοποιείται συνήθως για έρευνα δικτύων και πρωτοκόλλων, επιδείξεις, δοκιμές εφαρμογών και πλατφορμών, αξιολόγηση σεναρίων δικτύωσης, μελέτες ασφάλειας και αύξηση του μεγέθους των δικτύων με φυσικές δοκιμές(physical testing).[51]

9.1.1 Αρχιτεκτονική του Core

Ένα σύστημα container based όπως θα δούμε και πιο κάτω περιλαμβάνει μία κοινή εικόνα λειτουργικού που περιέχει ένα σύστημα που διαχειρίζεται αρχεία, ένα σύστημα δημόσιων βιβλιοθηκών και κοινών εκτελέσιμων αρχείων . Κάθε εικονική μηχανή μπορεί να ξεκινήσει ή να τερματιστεί σαν όλα τα γνωστά υπολογιστικά συστήματα . Το μέγεθος του δίσκου, ο αριθμός της CPU και το μέγεθος της μνήμης εισάγονται σε κάθε μηχανή τη στιγμή που κατασκευάζεται και υπάρχει δυνατότητα παραμετροποίησής της σε πραγματικές συνθήκες. Από τη μεριά του χρήστη τα OS images διακρίνονται ως ξεχωριστά μηχανήματα, έτσι π.χ. κάθε βλάβη που μπορεί να προκληθεί είναι εύκολο να απομονωθεί και να περιοριστεί σε κάθε VM. Επίσης κάθε VM δεν επηρεάζει τους πόρους άλλης εικονικής μηχανής(CPU, μνήμη, δίσκος).Τέλος κάθε μηχανή έχει τους δικούς της κανόνες όσον αφορά τον τομέα της ασφάλειας και δεν εξαρτάται από τους κανόνες που ορίζει μια άλλη εικονική μηχανή.



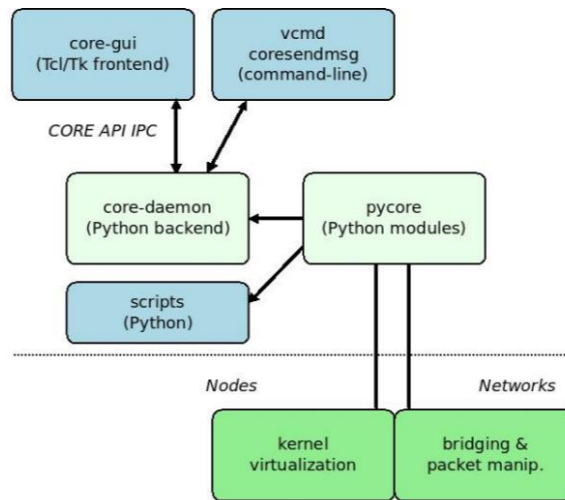
Εικόνα 9-1 Container based αρχιτεκτονική

Τα κύρια συστατικά του CORE εμφανίζονται στην παρακάτω εικόνα που μας δείχνει την αρχιτεκτονική του. Ένας CORE daemon (που δουλεύει backend) διαχειρίζεται τα session της εξομοίωσης. Χτίζει τα δίκτυα εξομοίωσης χρησιμοποιώντας εικονικοποίηση kernel (πυρήνα) για εικονικούς κόμβους και κάποια μορφή γεφύρωσης και επεξεργασίας πακέτων για εικονικά δίκτυα. Οι κόμβοι και τα δίκτυα έρχονται μαζί διαμέσων διεπαφών που είναι εγκατεστημένες στους κόμβους.

Ο daemon ελέγχεται διαμέσων του γραφικού περιβάλλοντος χρήστη, το CORE GUI (frontend). Ο daemon χρησιμοποιεί Python modules τα οποία μπορούν να εισαχθούν απευθείας από Python scripts. Το GUI και ο daemon επικοινωνούν χρησιμοποιώντας ένα προσαρμοσμένο, ασύγχρονο και βασισμένο σε sockets API, γνωστό και ως API CORE. Η διακεκομμένη γραμμή στο σχήμα απεικονίζει κατανοητά το χώρο του χρήστη και τον διαχωρισμό του με τον πυρήνα kernel. Τα συστατικά στοιχεία με τα οποία αλληλοεπιδρά ο χρήστης είναι με μπλε χρώμα και είναι GUI, σενάρια ή εργαλεία γραμμής εντολών.

Το σύστημα είναι αρθρωτό αποτελείται δηλαδή από διάφορα μέρη για να επιτρέπει ένα συνδυασμό διαφορετικών στοιχείων (components). Για παράδειγμα το στοιχείο των εικονικών δικτύων μπορεί να υλοποιηθεί και με άλλους προσομοιωτές- εξομοιωτές δικτύου όπως το ns-3 και το EMANE. Υπάρχει η δυνατότητα υποστήριξης διαφορετικού τύπου kernel εικονικοποίησης. Ένα άλλο παράδειγμα είναι ο τρόπος σχεδιασμού και εκκίνησης ενός session στο γραφικό περιβάλλον /etc/init.d/core-daemon και πως θα συνεχίσει να τρέχει σε "headless" λειτουργία με κλειστό το GUI. Το API του Core είναι βασισμένο σε sockets για να υπάρχει η δυνατότητα εκτέλεσης διαφορετικών components σε διαφορετικά μηχανήματα.

Το Core ξεκινά χρησιμοποιώντας την linux εντολή *core-gui*. Ο core δαίμονας συμβολίζεται ως *core-daemon* και συνήθως ξεκινά μέσω του script αρχικοποίησης (/etc/init.d/core-daemon ή μέσω της εντολής *core-daemon.service* αναλόγως την κάθε πλατφόρμα). Ο core δαίμονας διαχειρίζεται τα sessions των εικονικών κόμβων και δικτύων εκ των οποίων άλλα scripts και utilities (βοηθητικά προγράμματα) μπορούν να χρησιμοποιηθούν για περαιτέρω έλεγχο.



Εικόνα 9-2 Αρχιτεκτονική του Core emulator

9.1.2 Τρόπος Λειτουργίας του Core

Ο εξομοιωτής Core όπως είπαμε και πιο πάνω έχει ένα γεμάτο GUI όπου ο χρήστης πάνω στην παλέτα του γραφικού έχει τη δυνατότητα να χρησιμοποιήσει δομικά στοιχεία μιας δικτυακής τοπολογίας όπως *switches, routers, hosts* τα οποία μπορεί με drag and drop να τα τοποθετήσει στο σχεδιαστικό περιβάλλον. Η τοπολογία έχει τη δυνατότητα αρχικοποίησης καθώς εκτελείται. Ανοίγοντας ένα κόμβο της τοπολογίας μπορούμε να δώσουμε εντολές σε πραγματικό χρόνο. Σε αντίθεση με τις κλασσικές εικονικές μηχανές, τα OS images έχουν μέρος του λειτουργικού συστήματος που λειτουργεί πάνω στο host μηχανήμα. Έτσι λειτουργούν πάνω στον ίδιο kernel πυρήνα και διαμοιράζονται το ίδιο σύστημα αρχείων και υπολογιστικούς πόρους. Έτσι δημιουργούνται τοπολογίες με πολλαπλά εικονικά μηχανήματα.[44][52]

9.2 Οδηγίες και βήματα εγκατάστασης του Core εξομοιωτή

Για την εγκατάσταση του Core όπως και των υπόλοιπων εξομοιωτών (Mininet, Imunes) πρέπει να χρησιμοποιήσουμε την πλατφόρμα Virtual Box όπου και εκτελέσαμε τα περισσότερα από τα πειράματά μας. Έχοντας εγκαταστήσει τα Ubuntu 16.04 LTS ξεκινούμε την διαδικασία λήψης των απαραίτητων πακέτων και ενημερώσεων έτσι ώστε ο εξομοιωτής να τρέχει σωστά. Πηγαίνουμε στο repository του Github και κλωνοποιούμε τον κώδικα για να τον χρησιμοποιήσουμε στο δικό μας εικονικό μηχανήμα.[53]



Εγκατάσταση του Core από το Github

```
$ sudo apt-get install core-network  
$ sudo apt-get update  
$ sudo apt-get install git  
$ sudo apt-get install bash bridge-utils ebtables  
$ sudo apt-get install iproute libev-dev python tcl8.5 tk8.5 libtk-img  
$ sudo apt-get install autoconf automake gcc libev-dev make python-dev  
$ sudo apt-get install libreadline-dev pkg-config imagemagick help2man
```

Κλωνοποίηση του πηγαίου κώδικα του Core από το Github

```
$ cd  
$ git clone https://github.com/coreemu/core.git  
$ cd core  
$ ./bootstrap.sh  
$ ./configure  
$ make  
$ sudo make install
```

Στη συνέχεια κάνουμε επανεκκίνηση το σύστημα μας γράφοντας την εντολή reboot. Αν χρειαστούμε στο μέλλον κάποια ενημέρωση στην έκδοση του Core γράφουμε το παρακάτω script που τραβάει τις ενημερώσεις από το Github.

```
$ cd  
$ cd core  
$ git pull  
$ ./bootstrap.sh  
$ ./configure  
$ make  
$ sudo make install
```

Για να εκτελέσουμε τα πειράματά μας πρέπει να εγκαταστήσουμε ορισμένες υπηρεσίες δικτύου που ενδέχεται να εκτελούνται στους host που θα προσομοιώνουν κόμβους δικτύου στο CORE.

```
$ sudo apt-get install quagga quagga-doc
```



```
$ sudo apt-get install openssh-server isc-dhcp-server isc-dhcp-client  
$ sudo apt-get install vsftpd apache2 tcpdump radvd at ucarp openvpn  
$ sudo apt-get install ipsec-tools racoon traceroute mgen wireshark  
$ sudo apt-get install iperf3 tshark snmpd snmptrapd openssh-client
```

Επίσης κάνουμε εγκατάσταση και παραμετροποίηση το Wireshark έτσι ώστε οι χρήστες να μπορούν να κάνουν capture τα δεδομένα τους.

```
$ sudo setcap 'CAP_NET_RAW+eip CAP_NET_ADMIN+eip' /usr/bin/dumpcap  
$ sudo adduser $USER wireshark
```

Κάνουμε ακόμη ένα reboot για να ενεργοποιηθούν οι αλλαγές μας. Συνεχίζοντας ρυθμίζουμε το *Quagga* το οποίο είναι μια σουίτα λογισμικού δρομολόγησης, που υποστηρίζει OSPFv2, OSPFv3, RIP v1 και v2, RIPng και BGP-4 για πλατφόρμες Unix και ιδιαίτερα για το FreeBSD, το Linux, το Solaris και το NetBSD, το οποίο υπάρχει μέσα στις βιβλιοθήκες του Core και επιτρέπει στους δρομολογητές να εκτελούν πρωτόκολλα δρομολόγησης.

```
$ sudo touch /etc/quagga/zebra.conf  
$ sudo touch /etc/quagga/ospfd.conf  
$ sudo touch /etc/quagga/ospf6d.conf  
$ sudo touch /etc/quagga/ripd.conf  
$ sudo touch /etc/quagga/ripngd.conf  
$ sudo touch /etc/quagga/isisd.conf  
$ sudo touch /etc/quagga/pimd.conf  
$ sudo touch /etc/quagga/vtysh.conf  
$ sudo chown quagga.quaggavty /etc/quagga/*.conf  
$ sudo chmod 666 /etc/quagga/*.conf
```

Πηγαίνουμε στο αρχείο daemons file του Quagga και παραμετροποιούμε τα παρακάτω πεδία τα οποία είναι το zebra και το ospfd. Το πεδίο zebra μας επιτρέπει να υποστηρίξουμε routing protocols όποτε το αλλάζουμε σε **yes** και το **ospfd** για να υποστηρίζεται το OSPF και όλες του οι εκδόσεις.



```
GNU nano 2.5.3 File: /etc/quagga/daemons
# the daemon will not be started by /etc/init.d/quagga. The permissions $
# be u=rw,g=r,o=.
# When using "vtysh" such a config file is also needed. It should be own$
# group "quaggavty" and set to ug=rw,o= though. Check /etc/pam.d/quagga,$
#
# The watchquagga daemon is always started. Per default in monitoring-on$
# that can be changed via /etc/quagga/debian.conf.
#
zebra=no
bgpd=no
ospfd=no
ospf6d=no
ripd=no
ripngd=no
isisd=no
babeld=no
```

Εικόνα 9-3 παραμετροποίηση των πεδίων zebra και ospfd

```
GNU nano 2.5.3 File: /etc/quagga/daemons Modified
# the daemon will not be started by /etc/init.d/quagga. The permissions $
# be u=rw,g=r,o=.
# When using "vtysh" such a config file is also needed. It should be own$
# group "quaggavty" and set to ug=rw,o= though. Check /etc/pam.d/quagga,$
#
# The watchquagga daemon is always started. Per default in monitoring-on$
# that can be changed via /etc/quagga/debian.conf.
#
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
babeld=no
```

Εικόνα 9-4 Αλλαγή των πεδίων με την τιμή yes

Ακόμη για να μην έχουμε πρόβλημα με την προσομοίωση των εικονικών δρομολογητών που θα χρησιμοποιήσουμε στη συνέχεια, παραδείγματος χάριν μαύρη οθόνη στο τερματικό του κάθε δρομολογητή, κάνουμε παραμετροποίηση στους φακέλους του **Quagga daemon** που όπως είπαμε και πιο πάνω παίζει καθοριστικό ρόλο στην προσομοίωση μας.

```
$ sudo bash -c 'echo "export VTYSH_PAGER=more" /etc/bash.bashrc'
```

```
$ sudo bash -c 'echo "VTYSH_PAGER=more" /etc/environment'
```

Έχοντας εκτελέσει αυτή τη σειρά των βημάτων είμαστε έτοιμοι να τρέξει με το Core. Πάντα πρώτα τρέχουμε τον daemon του Core το οποίο τρέχει backend και διεκπαιρεύνει τα sessions



που εκτελούνται κάθε φορά και μετά το γραφικό Core gui όπως έχουμε αναφέρει και πιο πάνω στην αρχιτεκτονική του.

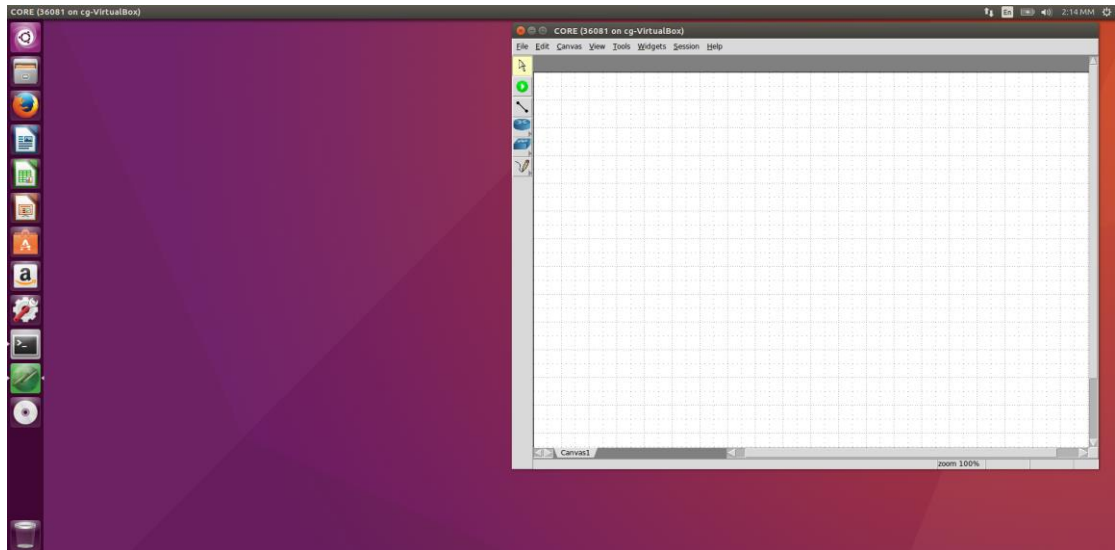
Εκκίνηση του Core daemon

```
$ sudo service core-daemon start
```

Εκκίνηση του Core Gui


```
$ core-gui
```


Και μας εμφανίζεται το γραφικό περιβάλλον του Core όπου θα δουλέψουμε ορισμένα πειράματα.



Εικόνα 9-5 Γραφικό περιβάλλον του Core

9.3 Πειράματα που υλοποιήθηκαν

Το Core GUI έχει δύο βασικές λειτουργίες την Επεξεργασία(Edit) και την εκτέλεση(Execute). Με την επιλογή  selection tool κάνουμε drag and drop τους κόμβους στον καμβά.

Μόλις τελειώσουμε το Editing πατώντας την επιλογή  start button αρχικοποιείται η τοπολογία εντός του πυρήνα Linux και εισέρχεται σε λειτουργία εκτέλεσης(Execute mode). Αφότου το project τρέξει ο χρήστης μπορεί να παραμετροποιήσει κάποια πράγματα που αφορούν την εκτέλεση όχι όμως να παραμετροποιήσει Editing λειτουργίες. Έχει τη δυνατότητα να ανοίξει διαφορετικό παράθυρο xterm για κάθε κόμβο. Κάνοντας stop την τοπολογία μπορούμε να έρθουμε πάλι σε execute mode.

Αυτό που είδαμε και στις προηγούμενες ενότητες είναι ότι τα Open source εργαλεία και λογισμικά μας δίνουν δυνατότητες παρατήρησης και κατανόησης του δικτύου σε πραγματικές συνθήκες.



9.3.1 Υλοποίηση Σεναρίου Έξυπνων Σπιτιών Στη Σάμο

9.3.1.1 Smart Enviroment

Η ραγδαία εξέλιξη της τεχνολογίας και η αύξηση της μικρογράφησης της τεχνολογίας έχουν βάλει στο προσκήνιο μικροσκοπικούς αισθητήρες και επεξεργαστές για να ενσωματωθούν σε καθημερινά αντικείμενα. Αυτή η πρόοδος του smart environment έχει υποστηριχτεί από τις τεράστιες εξελίξεις στα εξής: φορητές συσκευές, ασύρματη δικτύωση αισθητήρων, ασύρματες κινητές επικοινωνίες, μηχανική μάθηση βασισμένη στη λήψη αποφάσεων, υποστήριξη IPv6 και διάφορα άλλα ευφυή συστήματα.

Ένα smart environment είναι κατά κάποιο τρόπο ένας μικρόκοσμος όπου διάφορες διασυνδεδεμένες συσκευές που χρησιμοποιούν αισθητήρες κάνουν πιο εύκολη τη ζωή των ανθρώπων. Ο όρος έξυπνος αναφέρεται στην ικανότητα να αποκτούν αυτόνομα και να εφαρμόζουν τη γνώση, και ο όρος περιβάλλον αναφέρεται σε αυτά που το περιβάλλουν. Ως εκ τούτου, ένα έξυπνο περιβάλλον είναι ικανό να αποκτήσει γνώση και να εφαρμόζει την γνώση που έμαθε σύμφωνα με τις ανάγκες των χρηστών κάνοντας πιο χρηστικό το περιβάλλον που εφαρμόζεται. Οι δυνατότητες των έξυπνων αντικειμένων ενισχύονται με τη διασύνδεση τους με άλλα αντικείμενα που χρησιμοποιούν ασύρματες τεχνολογίες. Σε αυτό το κομμάτι το IPv6 διαδραματίζει ζωτικό ρόλο εξαιτίας ορισμένων χαρακτηριστικών του όπως μηχανισμοί ασφαλείας, scalability όσων αφορά τις IP διευθύνσεις δίνοντας ip σε δισεκατομμυρίων συνδεδεμένες συσκευές, καταργώντας το NAT. Τη σύνδεση των έξυπνων αντικειμένων με το Διαδίκτυο ήταν ιδέα του Kevin Ashton ως το "Διαδίκτυο των Πράγματα" (IoT). Σήμερα, το IoT λαμβάνει προσοχή σε πολλούς τομείς, όπως η υγειονομική περίθαλψη, τις μεταφορές και τη βιομηχανία.

Η Cisco αναφέρει ότι 50 δισεκατομμύρια αντικείμενα και συσκευές θα έχουν συνδεθεί στο Διαδίκτυο έως το 2020. Ωστόσο, περισσότερο από το 99 τοις εκατό των διαθέσιμων συσκευών σήμερα τα στον κόσμο παραμένουν δίχως σύνδεση στο διαδίκτυο. Σύμφωνα με μια έκθεση έρευνας του Navigant, ο αριθμός των εγκατεστημένων έξυπνων μετρητών σε όλο τον κόσμο θα αυξηθεί σε 1,1 δισεκατομμύρια μέχρι το 2022. Μια άλλη έκθεση από το Automotive News αναφέρει ότι ο αριθμός των αυτοκινήτων που είναι συνδεδεμένα με το διαδίκτυο σε όλο τον κόσμο θα αυξηθεί από 23 εκατομμύρια που ήταν το 2013 σε 152 εκατομμύρια το 2020. Η πρόβλεψη αυτής της σημαντικής ανάπτυξης δείχνει ότι το IoT θα γίνει ο πυρήνας των σύγχρονων κοινωνιών έτσι ώστε να γίνει κατανοητό το όραμα των έξυπνων περιβαλλόντων. Έχουν διεξαχθεί αρκετές ερευνητικές προσπάθειες να ενσωματωθεί το IOT σε έξυπνο περιβάλλον. Η διασύνδεση του IOT με ένα έξυπνο περιβάλλον επεκτείνει τις δυνατότητες των έξυπνων αντικειμένων επιτρέποντας στον χρήστη να παρακολουθεί το περιβάλλον από απομακρυσμένες τοποθεσίες. Το IoT μπορεί να διασυνδεθεί με διαφορετικά έξυπνα περιβάλλοντα βασιζόμενο πάντα στις απαιτήσεις της εφαρμογής. Οι τομείς του smart environment ανάλογα με το που εφαρμόζεται ταξινομούνται σε : έξυπνες πόλεις, έξυπνες κατοικίες, smart grid, έξυπνα κτίρια, έξυπνες μεταφορές, έξυπνες Εφαρμογές υγείας και την έξυπνη βιομηχανία.



9.3.1.2 Smart Homes

Όσον αφορά τα έξυπνα περιβάλλοντα που στα σπίτια, αυτό που προτάθηκε είναι μια οικιακή λύση που βασίζεται σε σύννεφο για την ανίχνευση ενός σφάλματος στο έξυπνο οικιακό περιβάλλον με βάση την SDN αρχιτεκτονική (επιτρέπει στα δίκτυα να χειρίζονται πολλές διαφορετικές υπηρεσίες).

Ένας software defined networking (SDN) ελεγκτής συλλέγει πληροφορίες από τα πακέτα που περνούν από τους SDN switches και πραγματοποιεί γράφημα σχετικά με την κατάσταση της κάθε συσκευής μέσα στο σπίτι. Ένα ελεγκτής SDN που χρησιμοποιεί cloud αρχικοποιεί αυτόματα και τους 4 τομείς (IoT physical space, IoT Service, IoT Network, και IoT IoT, μειώνοντας την επιβάρυνση των χρηστών και των παρόχων.

Η προτεινόμενη λύση είναι επωφελής τόσο για τους χρήστες όσο και για τους ISP του σπιτιού.. Η προτεινόμενη αρχιτεκτονική έχει πολλαπλά επίπεδα, όπως ένα στρώμα πυρήνα (core layer) και ένα service layer που διαδραματίζουν σημαντικό ρόλο στη λήψη αποφάσεων.

Το core επίπεδο περιλαμβάνει web services που διασυνδέουν το σύστημα με τις υπηρεσίες του. Μια home gateway σε έξυπνα σπίτια δίνει πρόσβαση σε εξωτερικά δίκτυα.

Για να επιτύχουμε αυτονομία στις συσκευές χρειάζεται προσθήκη νέων IOT συστημάτων ή υπηρεσιών, νέες διεπαφές και stub modules που χρησιμοποιούνται για τον κατακερματισμό των συσκευών. Οι web service υπηρεσίες χρησιμοποιούν stateless πρωτόκολλα και δεν είναι κατάλληλες για μεγάλες σε διάρκεια συνεδρίες (sessions). [56]

9.3.1.3 Υλοποίηση σεναρίου στο Core

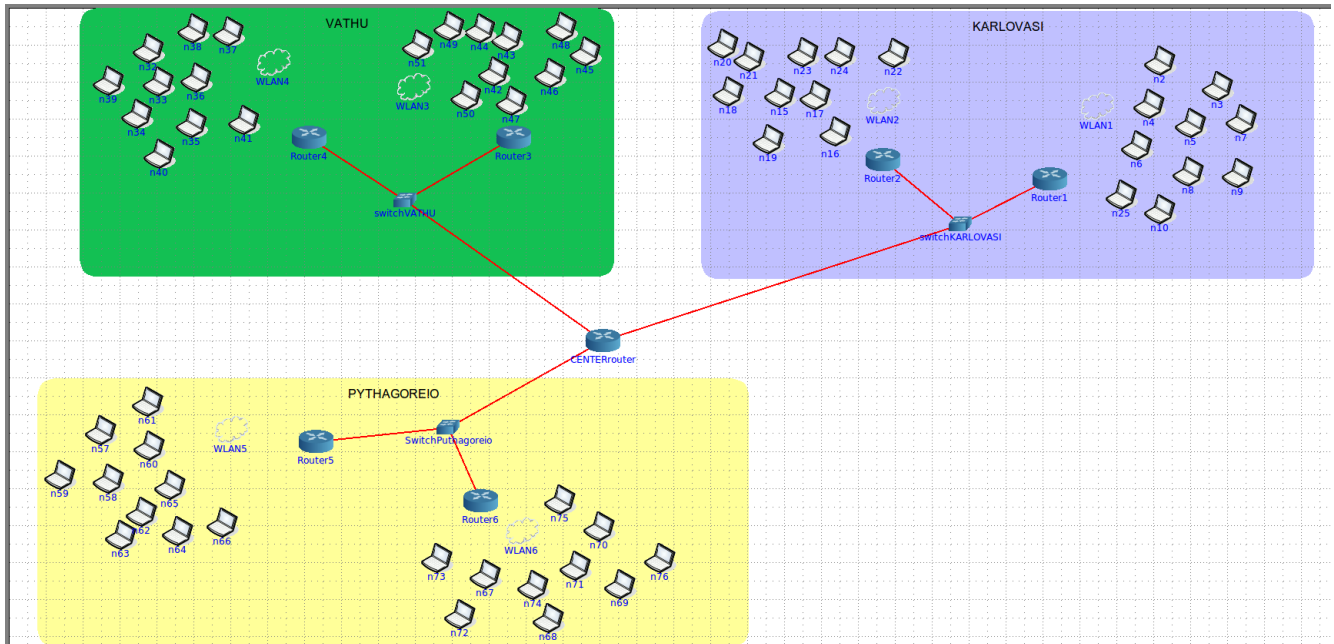
Στο παρακάτω σενάριο που θα υλοποιήσουμε στον εξομοιωτή Core θα προσομοιώσουμε την λειτουργία ενός δικτύου με έξυπνα σπίτια στην περιοχή της Σάμου. Πιο συγκεκριμένα θα έχουμε δύο έξυπνα σπίτια στο Νέο Καρλόβασι, δύο έξυπνα σπίτια στο Βαθύ και δύο έξυπνα σπίτια στο Πυθαγόρειο.

Το κάθε σπίτι που θα υπάρχει σε κάθε πόλη θα περιέχει δέκα έξυπνες συσκευές που θα επικοινωνούν με τον δρομολογητή του σπιτιού. Εκμεταλλευόμενοι τη δυνατότητα που μας δίνει το Core, έτσι ώστε λειτουργίες όπως η απόδοση των IP διευθύνσεων, η ενεργοποίηση πρωτοκόλλων δρομολόγησης και η μετάδοση πακέτων αποτελούν αυτοματοποιημένες διαδικασίες που τρέχουν στο background scripts που υλοποιούν αυτές τις εντολές ουσιαστικά. Η λειτουργία του Core θα γίνει πιο κατανοητή με την υλοποίηση και επεξήγηση του παρακάτω σεναρίου.

Ο εξομοιωτής Core σαν open source λογισμικό επιτρέπει στους χρήστες να παραμετροποιούν το περιβάλλον του ανάλογα με τις απαιτήσεις τους. Έτσι προσθέσαμε τις δικές



μας εικόνες για τις έξυπνες συσκευές και τις προσόψεις των σπιτιών. Η τοπολογία που ακολουθήσαμε έχει τον αντίστοιχο σχεδιασμό που φαίνεται στην παρακάτω εικόνα.



Εικόνα 9-6 Τοπολογία Έξυπνων σπιτιών στη Σάμο

Περίληπτικά η τοπολογία μας θα διαθέτει 60 hosts, 7 Routers, 3 switches και 6 WLAN κόμβους που θα ρυθμίζουν την εμφάνιση των wireless smart devices. Στη συνέχεια θα δούμε αναλυτικά πως κατασκευάσαμε το παραπάνω δίκτυο και τι πρωτόκολλα και λειτουργίες ενεργοποιήσαμε.

9.3.2 Έξυπνα Σπίτια Καρλοβάσσου

Όπως αναφέρθηκε και πιο πάνω το κάθε σπίτι έχει δέκα Wireless συσκευές που επικοινωνούν με τον router του σπιτιού. Χρησιμοποιήσαμε IPV6 διευθύνσεις για να αποφύγουμε τις διπλότυπες διευθύνσεις και για να ταιριάζει με το OSPFv3 που χρησιμοποιήσαμε για την επικοινωνία μεταξύ των δρομολογητών. Επίσης το Core δίνει την δυνατότητα συνύπαρξης IPV4 και IPV6 στο ίδιο δίκτυο πράγμα που πρέπει να εφαρμοστεί στο μέλλον και στον πραγματικό κόσμο σε ευρύτερη κλίμακα.

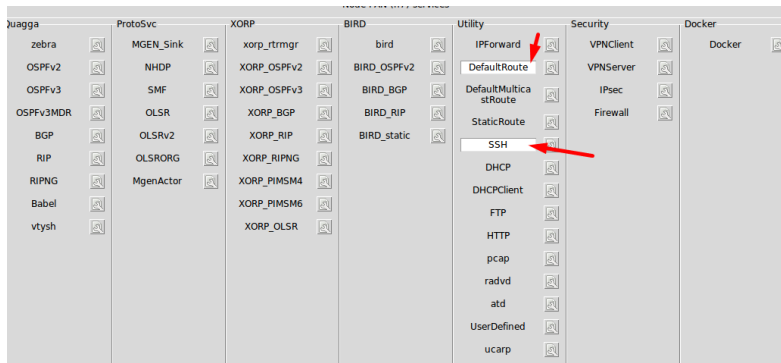


Πιο αναλυτικά για το *σπίτι νούμερο 1* στο Καρλόβασι χρησιμοποιήσαμε τις εξής συσκευές που εισήγαμε χειροκίνητα στο περιβάλλον του Core

| ID | Interface | NAME | IPv4 | IPv6 |
|----|-----------|-------------------|-----------------|---------------|
| 1 | eth0 | Router 1 | 192.168.0.1/24 | 2001:1::1/64 |
| 1 | eth1 | Router 1 | 192.168.1.1/24 | 2001::1/64 |
| 2 | eth0 | switchL3Karlovasi | 192.168.1.2/24 | 2001::2/64 |
| 2 | eth1 | switchL3Karlovasi | 192.168.2.1/24 | 2001:2::1/64 |
| 2 | eth2 | switchL3Karlovasi | 192.168.3.1/24 | 2001:3::1/64 |
| 3 | eth0 | Garage | 192.168.1.2/24 | 2001:1::2/64 |
| 4 | eth0 | CellingFan | 192.168.1.3/24 | 2001:1::3/64 |
| 5 | eth0 | Lamp | 192.168.1.4/24 | 2001:1::4/64 |
| 6 | eth0 | Smart Lock | 192.168.1.5/24 | 2001:1::5/64 |
| 7 | eth0 | Theromostat | 192.168.1.6/24 | 2001:1::6/64 |
| 8 | eth0 | Windows | 192.168.1.7/24 | 2001:1::7/64 |
| 9 | eth0 | Lawn Sprinkler | 192.168.1.8/24 | 2001:1::8/64 |
| 10 | eth0 | Camera | 192.168.1.9/24 | 2001:1::9/64 |
| 11 | eth0 | Movement Sensor | 192.168.1.10/24 | 2001:1::9/64 |
| 12 | eth0 | Humidifier | 192.168.1.11/24 | 2001:1::10/64 |

Πίνακας 9-1 IP διευθύνσεις smart home No1

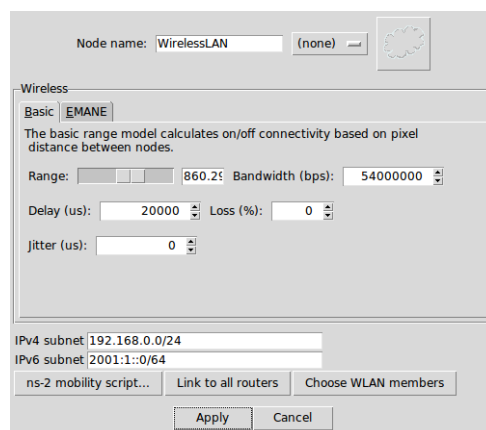
Αυτό που έπρεπε να ρυθμίσουμε στις συσκευές για να λειτουργούν ουσιαστικά μόνο σαν τερματικά ήταν οι εξής επιλογές στην καρτέλα Configure->Services. Αυτό που θα κάνουν είναι μονάχα Default route και SSH.



Εικόνα 9-7 Λειτουργία των Έξυπνων συσκευών

Το επόμενο βήμα που εκτελέσαμε ήταν να βάλουμε τον Wireless LAN κόμβο έτσι ώστε να προσθέσουμε μια ασύρματη πρόσβαση διαμέσων κεραίας στον οικιακό δρομολογητή. Κάνοντας drag and drop τον κόμβο WLAN προσθέσαμε τα εξής χαρακτηριστικά έτσι ώστε να επιτύχουμε την επιθυμητή επικοινωνία.

Πηγαίνοντας *WLAN->Choose WLAN Members* δημιουργήσαμε ένα group που ουσιαστικά προσθέσαμε τις έξυπνες συσκευές και φυσικά τον δρομολογητή που θα απέδιδε σε αυτές IP διευθύνσεις. Τα δίκτυα που θα χρησιμοποιήσουμε σε IPv4 είναι το 192.168.0.0/24 και σε IPv6 είναι το 2001:1::0/64.



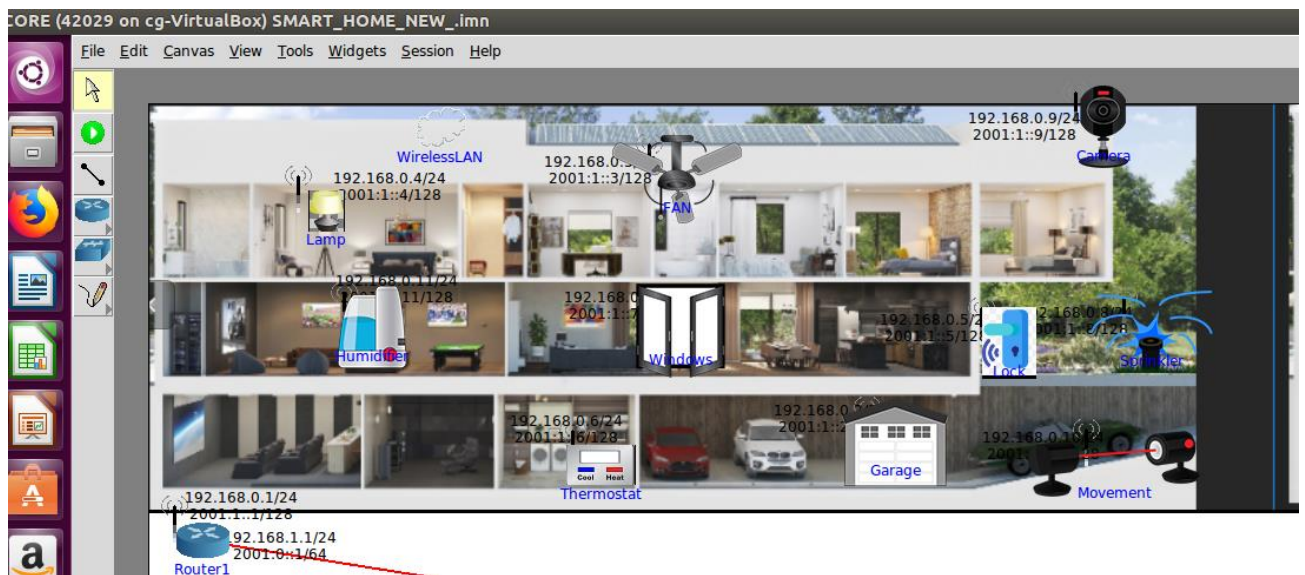
Εικόνα 9-8 Δίκτυα που θα τρέχουν στο δίκτυο του Smart Home 1



Συνεπώς στο WLAN group του κάθε σπιτιού θα είχαμε 11 κόμβους συν δηλαδή τον δρομολογητή. Ο δρομολογητής (*Router 1*) θα παίρνει την πρώτη IP του δικτύου, έτσι στο σενάριο μας στο πρώτο σπίτι θα έχει την *2001:1:1/128*.

Επίσης θα πρέπει να ορίσουμε και το κατάλληλο range για τις κεραίες των συσκευών έτσι ώστε να επικοινωνούν μεταξύ τους και φυσικά με τον δρομολογητή. Πηγαίνοντας *WLAN* κάνουμε edit το range που βασίζεται στην απόσταση των pixels μεταξύ των κόμβων.

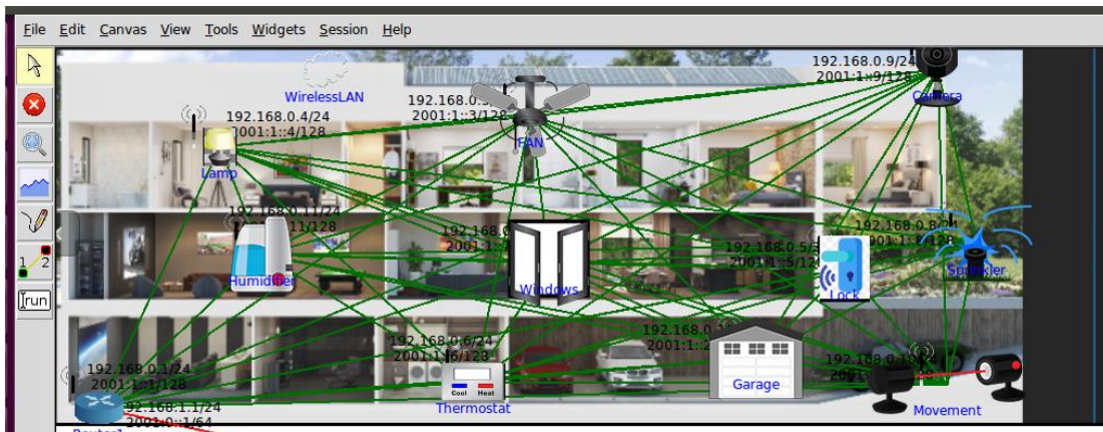
Παρακάτω βλέπουμε μία εικόνα από το πρώτο έξυπνο σπίτι στο Καρλόβασι που προσομοιώνουμε



Εικόνα 9-9 Τοπολογία Σπιτιού Νο1 στο Καρλόβασι

Στην παραπάνω εικόνα βλέπουμε τις συσκευές που έχουμε προσθέσει μέσα στο σπίτι τις IP που αποδίδονται και σε v4 και σε v6 για την διευκόλυνση στην κατανόηση, τις κεραίες που έχουν τοποθετηθεί σε όλες τις συσκευές καθώς και το Wireless κόμβο που είναι υπεύθυνος για την ασύρματη επικοινωνία.

Πατώντας start the session η τοπολογία μας ξεκινάει και βλέπουμε ότι όλες οι συσκευές είναι συνδεδεμένες μεταξύ τους, σχηματίζοντας διαδρομές η μία προς την άλλη.



Εικόνα 9-10 Τοπολογία Σπιτιού No1 στο Καρλόβασι

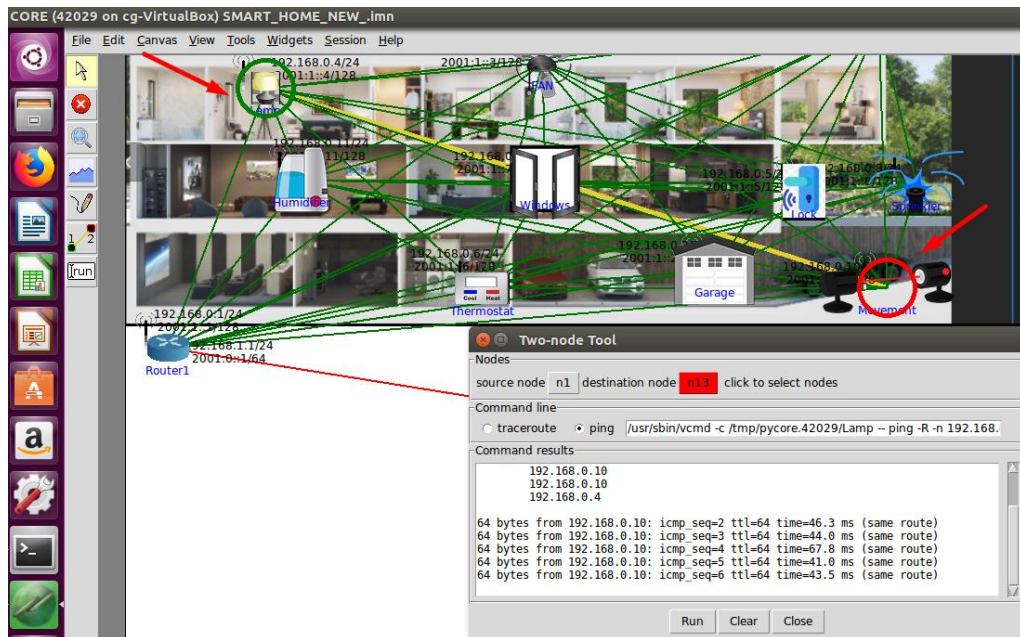
Επίσης στον δρομολογητή *Router 1* βλέπουμε ότι υπάρχουν δύο δίκτυα έτσι ώστε να επιτευχθεί η διασύνδεση και με τα υπόλοιπα σπίτια των άλλων πόλεων. Ο δρομολογητής έχει ενεργοποιημένα τα πρωτόκολλα OSPFv2 για IPv4, OSPFv3 για IPv6 και το IPForward.

9.3.3 Έλεγχος επικοινωνίας των Smart Devices

Όπως αναφέρθηκε και πιο πάνω οι λειτουργίες του ping, traceroute είναι αυτοματοποιημένες διαδικασίες που βοηθούν τον ερευνητή να κάνει πιο γρήγορα και αποδοτικά τις δοκιμές του χωρίς να σπαταλά αρκετές ώρες στο troubleshooting. Φυσικά δίνεται και η δυνατότητα χρήσης χειροκίνητων εντολών. Στις δοκιμές που θα ακολουθήσουμε θα δούμε την διευκόλυνση που παρέχει το Core συγκριτικά με τις άλλες πλατφόρμες που δουλέψαμε στην έρευνα μας.



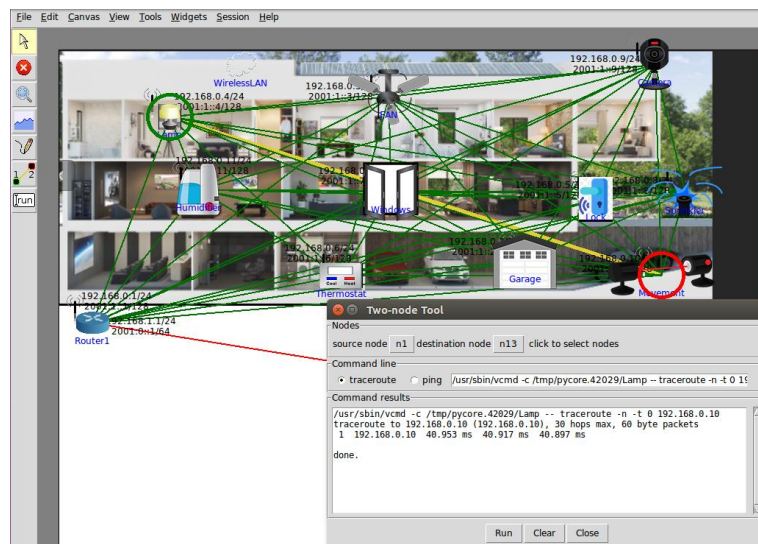
Ping από την Lamp->Sensor Movement



Εικόνα 9-11 Ping Lamp->Sensor Movement

Παρατηρούμε ότι οι δύο συσκευές επικοινωνούν επιτυχώς μεταξύ τους και βλέπουμε και την οπτική επαφή που έχουν με χρώμα κίτρινο.

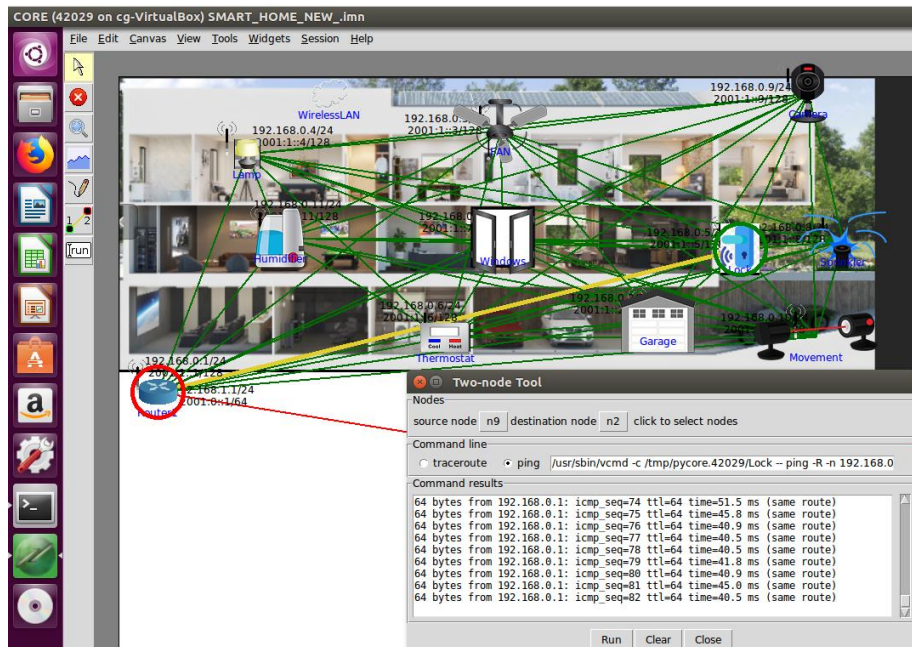
Κάνοντας στην συνέχεια *traceroute* παρατηρούμε ότι τα Hop όπως είναι αναμενόμενο είναι μονάχα 1.



Εικόνα 9-12 TraceRoute από Lamp->Sensor Management



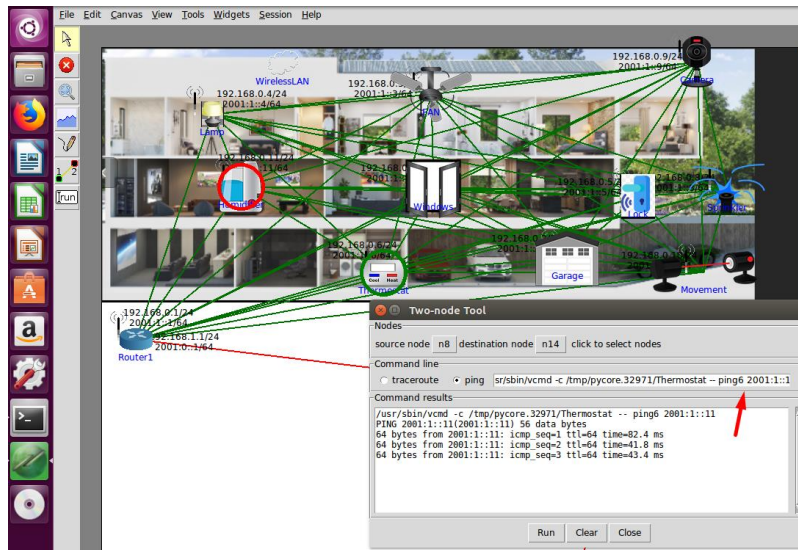
Ping από το Smart Lock->Router 1



Εικόνα 9-13 Ping από το Smart Lock->Router 1

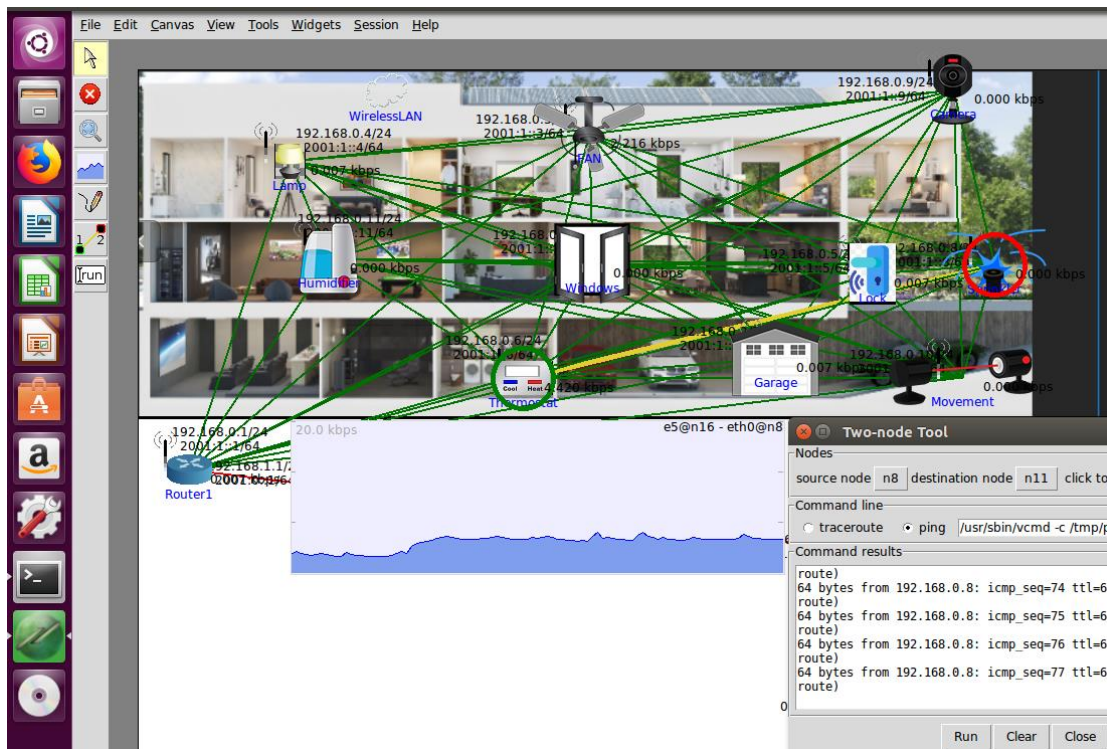
Ping6 από το Thermostat->Humidifier

Βλέπουμε ότι ο εξομοιωτής μας υποστηρίζει IPv4 και IPv6 επικοινωνία μεταξύ των έξυπνων συσκευών μας. Παραμετροποιούμε στην εντολή το ping, προσθέτοντας το ping6 και βλέπουμε ότι έχουμε επιτυχή επικοινωνία θερμοστάτη-αφυγραντήρας.



Εικόνα 9-14 Ping6 από τον Thermostat->Humidifier

Επίσης έχουμε την δυνατότητα να παρατηρούμε την κίνηση της κάθε ζεύξης σε πραγματικό χρόνο. Πιο κάτω δημιουργούμε κάποια κίνηση από τον θερμοστάτη προς την μηχανή αυτόματου ποτισμού και βλέπουμε το γράφημα της κίνησης σε kbps. Από το γράφημα φαίνεται η ασύρματη σύνδεση του Thermostat στην eth0 με το Wireless LAN ελεγκτή στην πόρτα e5.

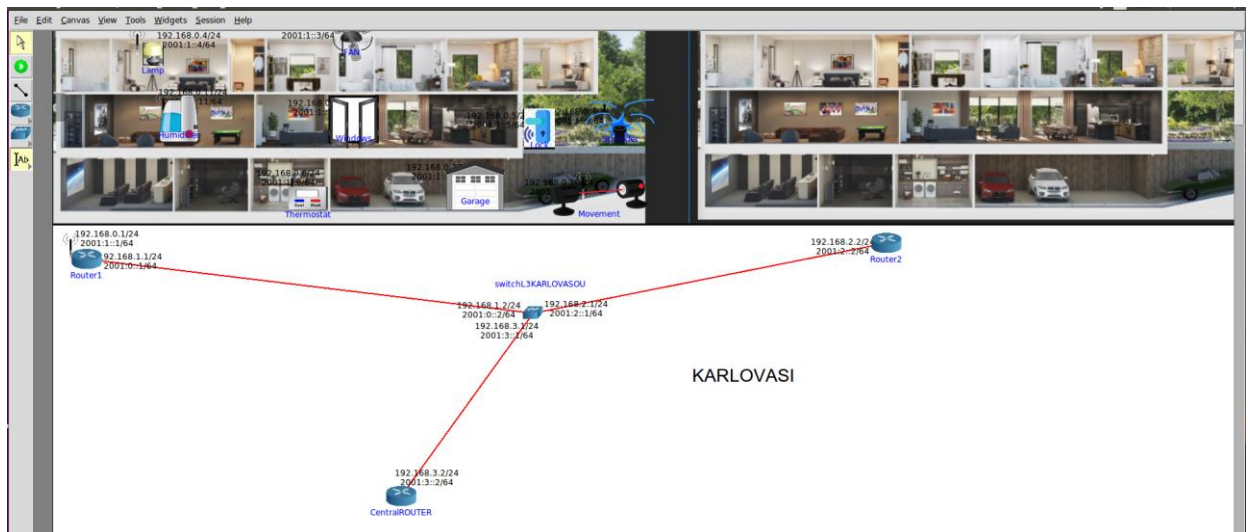


Εικόνα 9-15 Διάγραμμα κίνησης σε kbps



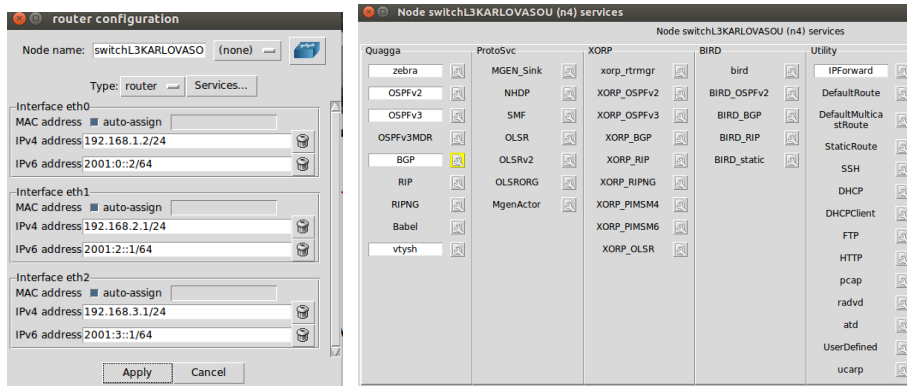
Έχοντας τελειώσει το πρώτο σπίτι στο Καρλόβασι στη συνέχεια θα πάμε να κατασκευάσουμε το δεύτερο σπίτι που θα έχει τις ίδιες συσκευές χρησιμοποιώντας διαφορετικό δίκτυο.

Αυτό ουσιαστικά που θα πρέπει να κάνουμε σύμφωνα πάντα με την τοπολογία του δικτύου μας είναι να έχουμε *ένα Switch Επιπέδου 3* για το Καρλόβασι όπως και για κάθε πόλη που θα συνδέει τους δρομολογητές των σπιτιών μεταξύ τους και θα είναι συνδεδεμένο *με τον κεντρικό δρομολογητή της Σάμου*.



Εικόνα 9-16 Δίκτυα στην πόλη του Καρλοβάσου

- Έτσι ο *Router 1* θα έχει δύο δίκτυα όπως βλέπουμε από την παραπάνω εικόνα ένα που συνδέεται με το σπίτι και ένα που συνδέεται με το Switch Επιπέδου 3 που το ονομάζουμε switchL3KARLOVASOU.
- Ακόμη το SwitchL3KARLOVASOU θα συνδέεται με τον δρομολογητή *Router 2* του δεύτερου σπιτιού στην πόλη με άλλο δίκτυο, καθώς και με τον κεντρικό δρομολογητή της Σάμου σε διαφορετικό δίκτυο.
- Το πρωτόκολλα που θα τρέχει το SwitchL3KARLOVASOU για να επιτύχουμε την ορθή επικοινωνία θα είναι το *BGP και OSPFv2, OSPFv3 για IPv4, IPv6* αντίστοιχα.
- Πηγαίνουμε στο Configure Tab του μεταγωγέα και ενεργοποιούμε τα συγκεκριμένα πρωτόκολλα για να επιτύχουμε την επιθυμητή γειτνίαση.



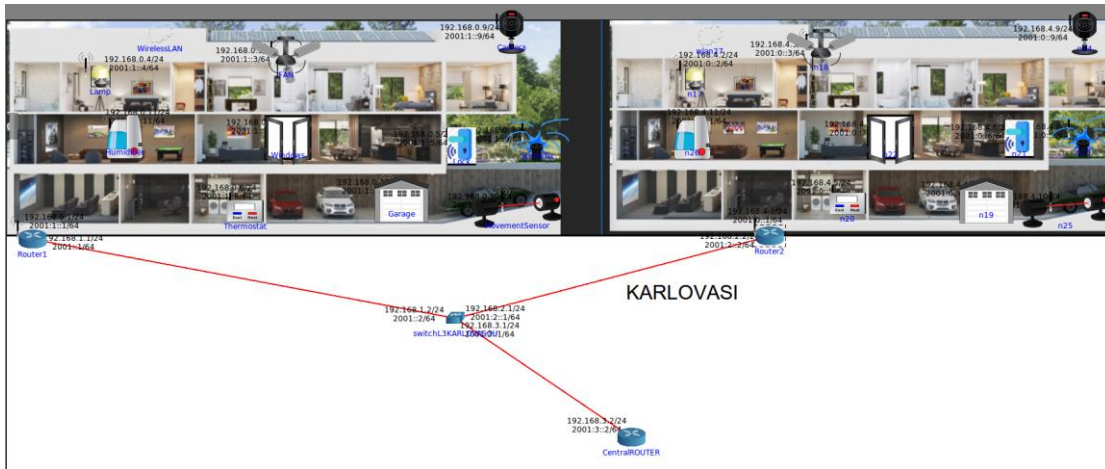
Εικόνα 9-17 Διαθέσιμα Δίκτυα και Διαθέσιμα Services του μεταγωγέα Καρλοβάσου

- Πηγαίνουμε στο σπίτι νούμερο 2 και υλοποιούμε παρόμοιες ενέργειες με το σπίτι νούμερο 1

Πιο αναλυτικά για το *σπίτι νούμερο 2* στο Καρλόβασι χρησιμοποιήσαμε τις εξής συσκευές που εισήγαμε χειροκίνητα στο περιβάλλον του Core

| ID | Interface | NAME | IPv4 | IPv6 |
|----|-----------|-----------------|-----------------|--------------|
| 1 | eth0 | Router 2 | 192.168.2.2/24 | 2001:2::2/64 |
| 1 | eth1 | Router 2 | 192.168.4.1/24 | 2001::1/64 |
| 2 | eth0 | Lamp | 192.168.4.2/24 | 2001::2/64 |
| 3 | eth0 | CellingFan | 192.168.4.3/24 | 2001::3/64 |
| 4 | eth0 | Garage | 192.168.4.4/24 | 2001::4/64 |
| 5 | eth0 | Thermostat | 192.168.4.5/24 | 2001::5/64 |
| 6 | eth0 | Smart Lock | 192.168.4.6/24 | 2001::6/64 |
| 7 | eth0 | Windows | 192.168.4.7/24 | 2001::7/64 |
| 8 | eth0 | Lawn Sprinkler | 192.168.4.8/24 | 2001::8/64 |
| 9 | eth0 | Camera | 192.168.4.9/24 | 2001::9/64 |
| 10 | eth0 | Movement Sensor | 192.168.4.10/24 | 2001::10/64 |
| 11 | eth0 | Humidifier | 192.168.4.11/24 | 2001::11/64 |

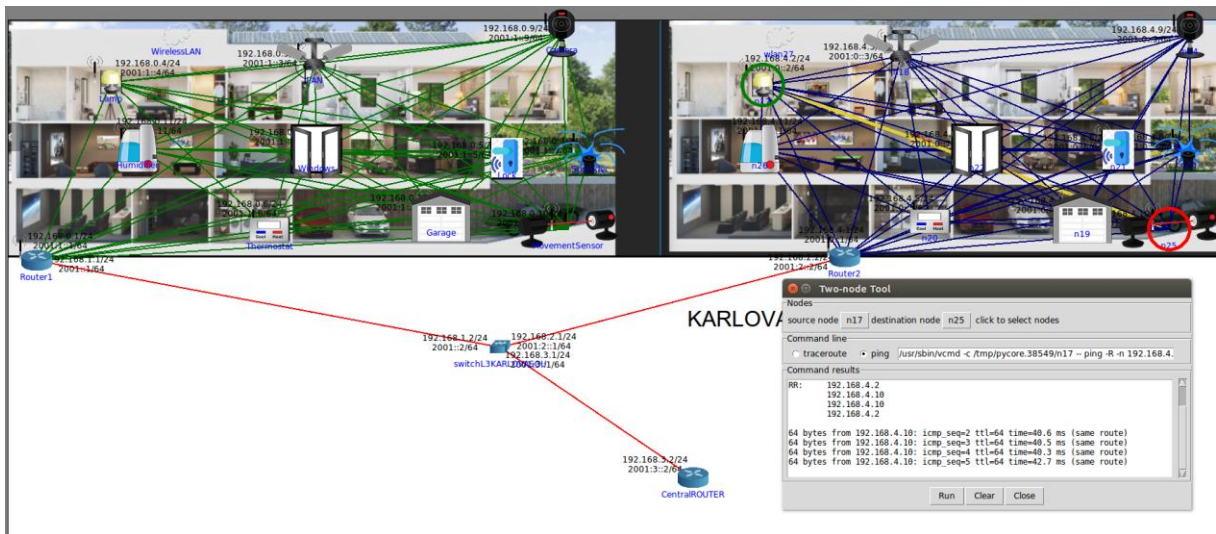
Πίνακας 9-2 IP διευθύνσεις smart home No2



Εικόνα 9-18 Τοπολογία σπιτιών Καρλοβάσου

Έχοντας τοποθετήσει τις αντίστοιχες συσκευές και στο σπίτι νούμερο 2, φροντίζοντας να έχουν την κατάλληλη IP διεύθυνση δοκιμάζουμε και εδώ ορισμένα rings για να δούμε ότι λειτουργεί σωστά η δικτύωση του σπιτιού.

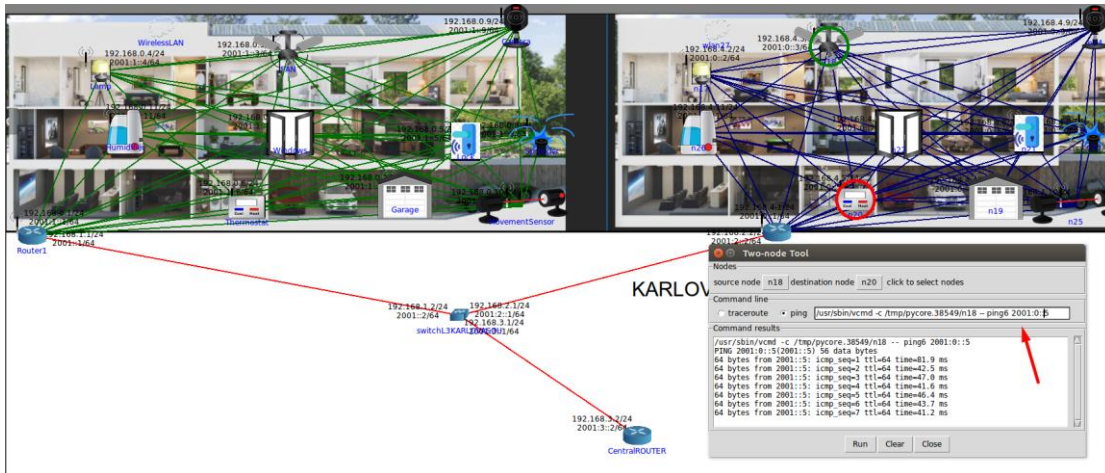
Ring από την Lamp του δεύτερου σπιτιού->Movement Sensor



Εικόνα 9-19 Ring σπιτιού νούμερο 2



Ping6 CielingFAN->Thermostat

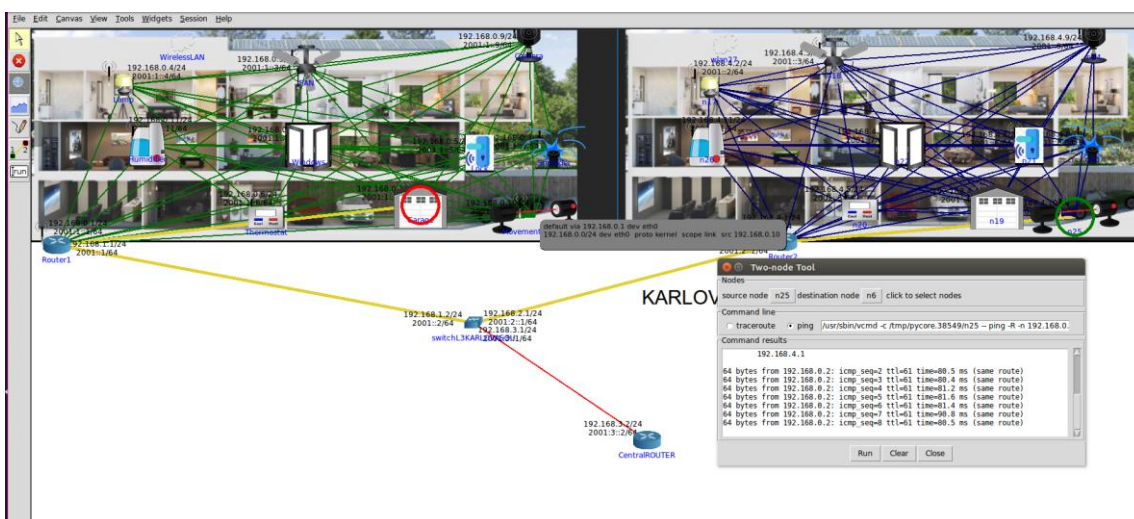


Εικόνα 9-20 Ping6 σπιτιού νοούμερο 2

Ενεργοποιώντας την λειτουργία του *Observer* ενώ η τοπολογία μας τρέχει μπορούμε αφού επιλέξουμε IPv4 ή IPv6 routes στις συσκευές μας, να δούμε τα διαθέσιμα routes στην κάθε συσκευή.

- Θα πάμε στο Switch του Καρλοβάσου και θα δούμε ότι γνωρίζει τα διαθέσιμα δίκτυα των σπιτιών και του κάθε δρομολογητή

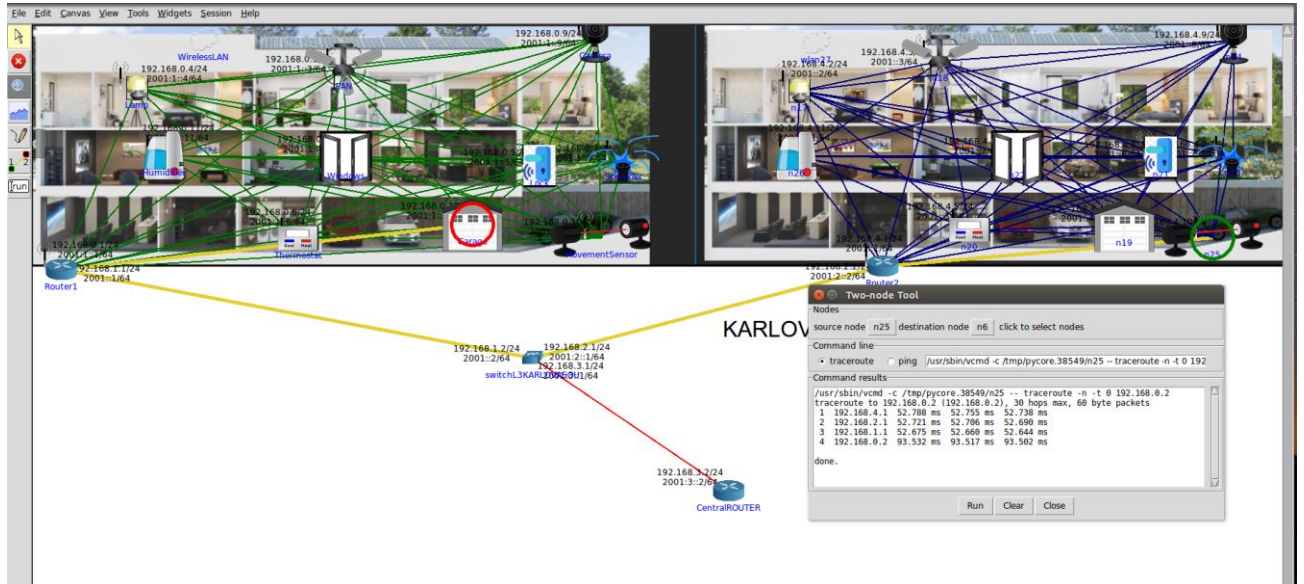
Ping από τον Movement Sensor του σπιτιού νοούμερο 2->Garage του σπιτιού νοούμερο 1



Εικόνα 9-21 Επικοινωνία Devices Διαφορετικών σπιτιών



Traceroute από τον Movement Sensor του σπιτιού νούμερο 2->Garage του σπιτιού νούμερο 1



Εικόνα 9-22 Εύρεση μονοπατιού Movement Sensor του σπιτιού νούμερο 2->Garage του σπιτιού νούμερο 1

9.3.4 Capture Πακέτων διαμέσων του Wireshark

Ανοίγοντας το Wireshark για το link του *Router 1 eth1* πριν δημιουργήσουμε κάποια κίνηση βλέπουμε τα εξής πακέτα. Παρατηρούμε τα hello πακέτα που ανταλλάσσονται από τους γειτονικούς δρομολογητές διαμέσων του OSPFv2 πρωτοκόλλου που χρησιμοποιούν. Επίσης βλέπουμε τις ενημερώσεις από το MDNS πρωτόκολλο που ουσιαστικά μετατρέπει τα host names σε IP addresses. Ακόμη βλέπουμε τις ARP αιτήσεις που στέλνονται για να γνωστοποιηθούν οι δρομολογητές.

| | | | | | | |
|----|-------------|---------------------|-------------------|--------|-----|---|
| 1 | 0.000000000 | 192.168.1.2 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 2 | 0.000120421 | 192.168.1.1 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 3 | 0.038338382 | fe80::200:ff:feaa:c | ff02::5 | OSPF | 94 | Hello Packet |
| 4 | 0.038449268 | fe80::200:ff:feaa:b | ff02::5 | OSPF | 94 | Hello Packet |
| 5 | 1.007929332 | fe80::f477:c0ff:fed | ff02::fb | MDNS | 203 | Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question PTR _ipps._tcp.local, " |
| 6 | 2.594724144 | fe80::d074:b4ff:feb | ff02::fb | MDNS | 203 | Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question PTR _ipps._tcp.local, " |
| 7 | 2.625183567 | fe80::f477:c0ff:fed | ff02::2 | ICMPv6 | 70 | Router Solicitation from d2:74:b4:be:55:71 |
| 8 | 2.625309405 | fe80::d074:b4ff:feb | ff02::2 | ICMPv6 | 70 | Router Solicitation from d2:74:b4:be:55:71 |
| 9 | 9.993474800 | 00:00:00:aa:00:0b | Broadcast | ARP | 42 | who has 192.168.1.2? Tell 192.168.1.1 |
| 10 | 9.993522881 | 00:00:00:aa:00:0c | 00:00:00:aa:00:0b | ARP | 42 | 192.168.1.2 is at 00:00:00:aa:00:0c |

Εικόνα 9-23 Capturing πακέτων στο R1-eth1



Δημιουργία κίνησης από Thermostat (σπίτι2)->Lawn Sprinkler(σπίτι1)

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|-------------|-------------|----------|--------|--|
| 1 | 0.000000000 | 192.168.4.5 | 192.168.0.8 | ICMP | 138 | Echo (ping) request id=0x0027, seq=13/3328, ttl=62 (reply in 2) |
| 2 | 0.041149794 | 192.168.0.8 | 192.168.4.5 | ICMP | 138 | Echo (ping) reply id=0x0027, seq=13/3328, ttl=63 (request in 1) |
| 3 | 1.001844499 | 192.168.4.5 | 192.168.0.8 | ICMP | 138 | Echo (ping) request id=0x0027, seq=14/3584, ttl=62 (reply in 4) |
| 4 | 1.045244318 | 192.168.0.8 | 192.168.4.5 | ICMP | 138 | Echo (ping) reply id=0x0027, seq=14/3584, ttl=63 (request in 3) |
| 5 | 2.003492057 | 192.168.4.5 | 192.168.0.8 | ICMP | 138 | Echo (ping) request id=0x0027, seq=15/3840, ttl=62 (reply in 6) |
| 6 | 2.044273976 | 192.168.0.8 | 192.168.4.5 | ICMP | 138 | Echo (ping) reply id=0x0027, seq=15/3840, ttl=63 (request in 5) |
| 7 | 3.004463937 | 192.168.4.5 | 192.168.0.8 | ICMP | 138 | Echo (ping) request id=0x0027, seq=16/4096, ttl=62 (reply in 8) |
| 8 | 3.045414469 | 192.168.0.8 | 192.168.4.5 | ICMP | 138 | Echo (ping) reply id=0x0027, seq=16/4096, ttl=63 (request in 7) |
| 9 | 4.004867534 | 192.168.4.5 | 192.168.0.8 | ICMP | 138 | Echo (ping) request id=0x0027, seq=17/4352, ttl=62 (reply in 10) |
| 10 | 4.046543781 | 192.168.0.8 | 192.168.4.5 | ICMP | 138 | Echo (ping) reply id=0x0027, seq=17/4352, ttl=63 (request in 9) |
| 11 | 5.000798790 | 192.168.4.5 | 192.168.0.8 | ICMP | 138 | Echo (ping) request id=0x0027, seq=18/4608, ttl=62 (reply in 12) |
| 12 | 5.047227558 | 192.168.0.8 | 192.168.4.5 | ICMP | 138 | Echo (ping) reply id=0x0027, seq=18/4608, ttl=63 (request in 11) |
| 13 | 6.011012197 | 192.168.4.5 | 192.168.0.8 | ICMP | 138 | Echo (ping) request id=0x0027, seq=19/4864, ttl=62 (reply in 14) |
| 14 | 6.052808233 | 192.168.0.8 | 192.168.4.5 | ICMP | 138 | Echo (ping) reply id=0x0027, seq=19/4864, ttl=63 (request in 13) |

Εικόνα 9-24 ICMP πακέτα

Ping6 από Camera(σπίτι1)->Lamp(σπίτι2)

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|-------------------|-------------------|----------|--------|---|
| 15 | 2.171759826 | 2001::9 | 2001::2 | ICMPv6 | 118 | Echo (ping) request id=0x0043, seq=9, hop limit=63 (reply in 16) |
| 16 | 2.171780337 | 2001::2 | 2001::9 | ICMPv6 | 118 | Echo (ping) reply id=0x0043, seq=9, hop limit=64 (request in 15) |
| 17 | 2.210682971 | 00:00:00:aa:00:0b | 00:00:00:aa:00:0c | ARP | 42 | Who has 192.168.1.2? Tell 192.168.1.1 |
| 18 | 2.210713526 | 00:00:00:aa:00:0c | 00:00:00:aa:00:0b | ARP | 42 | 192.168.1.2 is at 00:00:00:aa:00:0c |
| 19 | 3.002344520 | 192.168.4.4 | 192.168.0.11 | ICMP | 138 | Echo (ping) request id=0x0034, seq=195/49920, ttl=62 (no response found!) |
| 20 | 3.049747084 | 192.168.0.11 | 192.168.4.4 | ICMP | 138 | Echo (ping) reply id=0x0034, seq=195/49920, ttl=63 (request in 19) |
| 21 | 3.176488400 | 2001::9 | 2001::2 | ICMPv6 | 118 | Echo (ping) request id=0x0043, seq=10, hop limit=63 (reply in 22) |
| 22 | 3.176517620 | 2001::2 | 2001::9 | ICMPv6 | 118 | Echo (ping) reply id=0x0043, seq=10, hop limit=64 (request in 21) |
| 23 | 4.017069214 | 192.168.4.4 | 192.168.0.11 | ICMP | 138 | Echo (ping) request id=0x0034, seq=196/50176, ttl=62 (reply in 24) |
| 24 | 4.076481844 | 192.168.0.11 | 192.168.4.4 | ICMP | 138 | Echo (ping) reply id=0x0034, seq=196/50176, ttl=63 (request in 23) |
| 25 | 4.178550012 | 2001::9 | 2001::2 | ICMPv6 | 118 | Echo (ping) request id=0x0043, seq=11, hop limit=63 (no response found!) |

Εικόνα 9-25 ICMPv6 πακέτα

Στην παραπάνω εικόνα βλέπουμε το ICMPv6 request που στέλνει η κάμερα(2001::9) προς την λάμπα(2001::2) και το ICMPv6 reply της κάμερας. Έπειτα βλέπουμε και τις ARP αιτήσεις που ανταλλάσσουν για την επιτυχή αναγνώριση τους.

9.4 Έξυπνα Σπίτια Βαθύ

Για την υλοποίηση των έξυπνων σπιτιών στο Βαθύ κινηθήκαμε στο ίδιο μοτίβο με το Καρλόβασι δημιουργώντας σπίτια με αντίστοιχες συσκευές και με ίδια αρχιτεκτονική δικτύου.

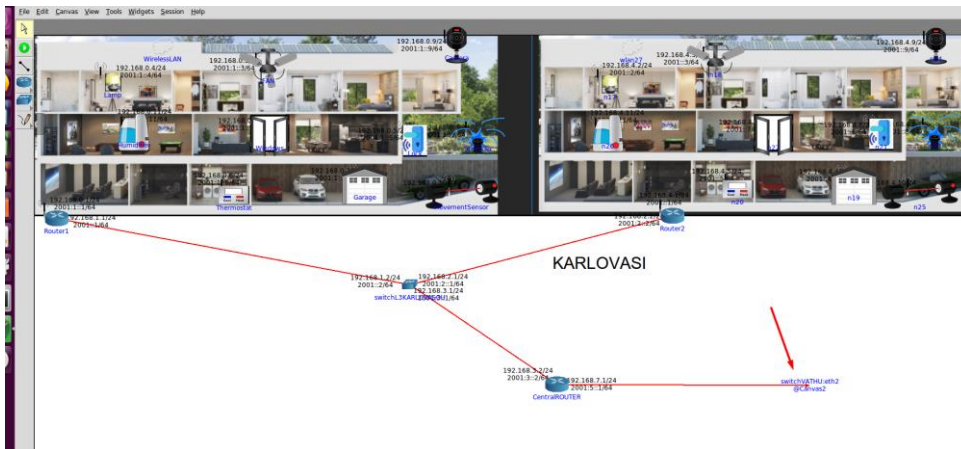
Αυτό που έπρεπε να κάνουμε ήταν να δημιουργήσουμε έναν δεύτερο καμβά όπου εκεί θα βάζαμε την νέα μας τοπολογία και θα την συνδέαμε με τον πρώτο καμβά.

Όπως έχει αναλυθεί και πιο πάνω όλες οι πόλεις θα παίρνουν από έναν Central Router όπου εμείς για ευκολία των έχουμε τοποθετήσει στον καμβά νούμερο 1 όπου υπάρχουν και τα έξυπνα σπίτια στο Καρλόβασι. Έτσι αυτός θα ήταν ο σύνδεσμος(link) του καμβά 1 προς τον καμβά 2 (Έξυπνα σπίτια Βαθύ).

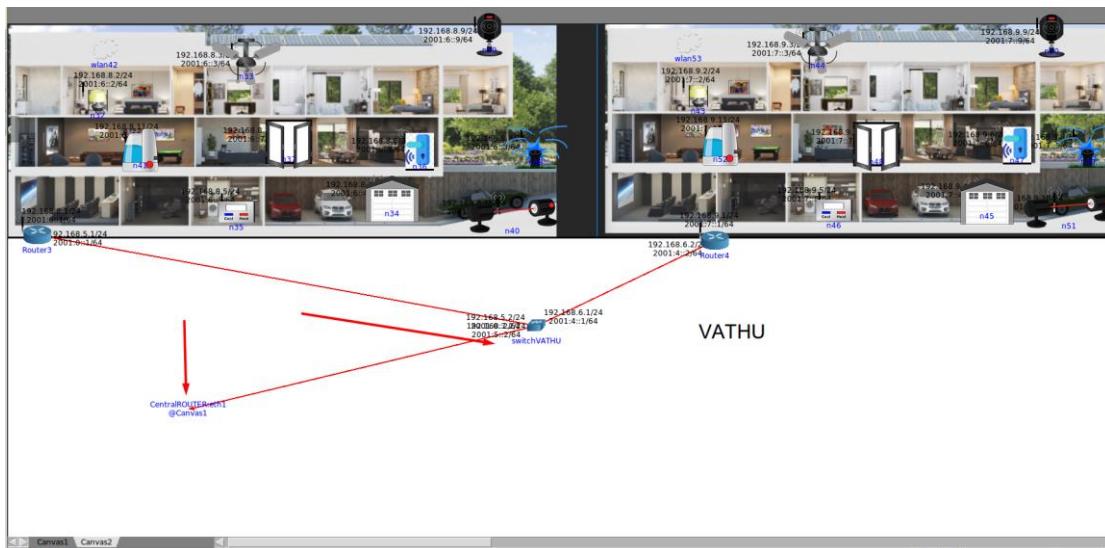
Το Core έχει τη δυνατότητα να τρέχει διαφορετικούς καμβάδες ταυτόχρονα, έτσι δημιουργήσαμε σύνδεση από τον Central Router<->switchVATHU για την διασύνδεση των δύο πόλεων, όπως



βλέπουμε στα παρακάτω στιγμιότυπα οθόνης. Πατώντας πάνω στο link που έχει τονιστεί με κόκκινο βέλος μεταφερόμαστε στο δεύτερο καμβά όπου υπάρχει η πόλη του Βαθίου.



Εικόνα 9-26 Link Καρλόβασι-Βαθύ



Εικόνα 9-27 Link Βαθύ-Καρλόβασι

Συνεπώς έχουμε τα εξής δίκτυα στους δρομολογητές της πόλης της Σάμου:

Central Router : 192.168.3.0/24 -> Σύνδεση με switchKarlovasi

192.168.7.0/24->Σύνδεση με switchVathu

Router3(Smart home No3 Βαθύ) : 192.168.5.0/24->Σύνδεση με switchVathu

192.168.8.0/24->Σύνδεση με το έξυπνο σπίτι Νο3

Router4(Smart home No4 Βαθύ) : 192.168.6.0/24->Σύνδεση με switchVathu

192.168.9.0/24->Σύνδεση με το έξυπνο σπίτι Νο4



SwitchL3VATHU: 192.168.5.0/24->Σύνδεση με Router3

192.168.6.0/24->Σύνδεση με Router4

192.168.7.0/24->Σύνδεση με Central Router

- Τα πρωτόκολλα που τρέχουν οι δρομολογητές όπως έχουμε αναφέρει και για τα σπίτια του Καρλοβάσου είναι *OSPFv2, OSPFv3* και *BGP* στο switch της κάθε πόλης.

Πιο κάτω παραθέτουμε τις IP διευθύνσεις που έχουν ανατεθεί στις συσκευές μέσα στο σπίτι χρησιμοποιώντας το WLAN manager για την ομαδοποίηση ουσιαστικά του Router με τις συσκευές του σπιτιών Νο3, Νο4.

Έξυπνο σπίτι Νο3 ΒΑΘΥ IP διευθύνσεις

| ID | Interface | NAME | IPv4 | IPv6 |
|----|-----------|-----------------|-----------------|---------------|
| 1 | eth0 | Router 3 | 192.168.5.1/24 | 2001::1/64 |
| 1 | eth1 | Router 3 | 192.168.8.1/24 | 2001:6::1/64 |
| 2 | eth0 | SwitchL3VATHU | 192.168.5.2/24 | 2001::2/64 |
| 2 | eth1 | SwitchL3VATHU | 192.168.6.1/24 | 2001:4::1/64 |
| 2 | eth2 | SwitchL3VATHU | 192.168.7.2/24 | 2001:5::2/64 |
| 3 | eth0 | Lamp | 192.168.8.2/24 | 2001:6::2/64 |
| 4 | eth0 | CellingFan | 192.168.8.3/24 | 2001:6::3/64 |
| 5 | eth0 | Garage | 192.168.8.4/24 | 2001:6::4/64 |
| 6 | eth0 | Thermostat | 192.168.8.5/24 | 2001:6::5/64 |
| 7 | eth0 | Smart Lock | 192.168.8.6/24 | 2001:6::6/64 |
| 8 | eth0 | Windows | 192.168.8.7/24 | 2001:6::7/64 |
| 9 | eth0 | Lawn Sprinkler | 192.168.8.8/24 | 2001:6::8/64 |
| 10 | eth0 | Camera | 192.168.8.9/24 | 2001:6::9/64 |
| 11 | eth0 | Movement Sensor | 192.168.8.10/24 | 2001:6::10/64 |
| 12 | eth0 | Humidifier | 192.168.8.11/24 | 2001:6::11/64 |

Πίνακας 9-3 IP διευθύνσεις smart home Νο3



Έξυπνο σπίτι Νο4 ΒΑΘΥ IP διευθύνσεις

| ID | Interface | NAME | IPv4 | IPv6 |
|----|-----------|-----------------|-----------------|---------------|
| 1 | eth0 | Router 4 | 192.168.6.2/24 | 2001:4::2/64 |
| 1 | eth1 | Router 4 | 192.168.9.1/24 | 2001:7::1/64 |
| 2 | eth0 | Lamp | 192.168.9.2/24 | 2001:7::2/64 |
| 3 | eth0 | CellingFan | 192.168.9.3/24 | 2001:7::3/64 |
| 4 | eth0 | Garage | 192.168.9.4/24 | 2001:7::4/64 |
| 5 | eth0 | Thermostat | 192.168.9.5/24 | 2001:7::5/64 |
| 6 | eth0 | Smart Lock | 192.168.9.6/24 | 2001:7::6/64 |
| 7 | eth0 | Windows | 192.168.9.7/24 | 2001:7::7/64 |
| 8 | eth0 | Lawn Sprinkler | 192.168.9.8/24 | 2001:7::8/64 |
| 9 | eth0 | Camera | 192.168.9.9/24 | 2001:7::9/64 |
| 10 | eth0 | Movement Sensor | 192.168.9.10/24 | 2001:7::10/64 |
| 11 | eth0 | Humidifier | 192.168.9.11/24 | 2001:7::11/64 |

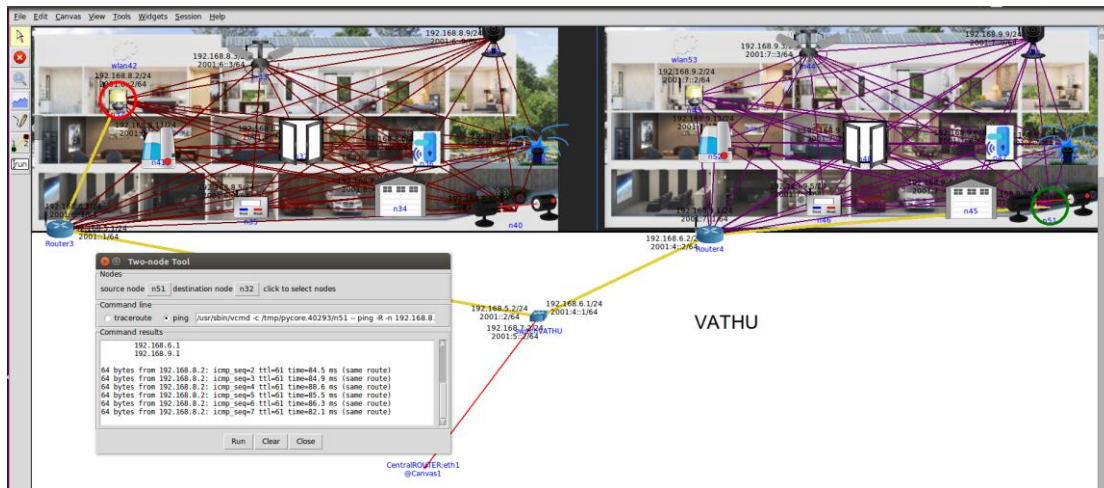
Πίνακας 9-4 IP διευθύνσεις smart home Νο4

9.4.1 Έλεγχος επικοινωνίας των Smart Devices

Έχοντας υλοποιήσει σωστά τις απαραίτητες λειτουργίες στα έξυπνα σπίτια στο ΒΑΘΥ θα πάμε να ελέγξουμε την λειτουργία μεταξύ τους και με τα άλλα έξυπνα σπίτια στο Καρλόβασι.

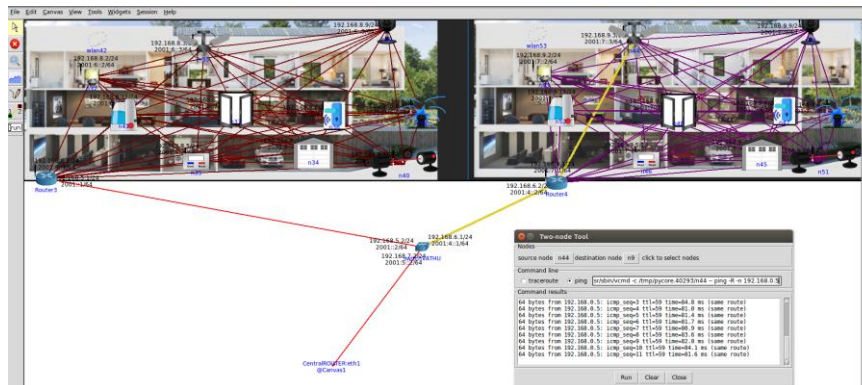


Pings από Sensor Movement smart home 4->Lamp του smart home 3



Εικόνα 9-28 Επικοινωνία των smart home στο Βαθύ

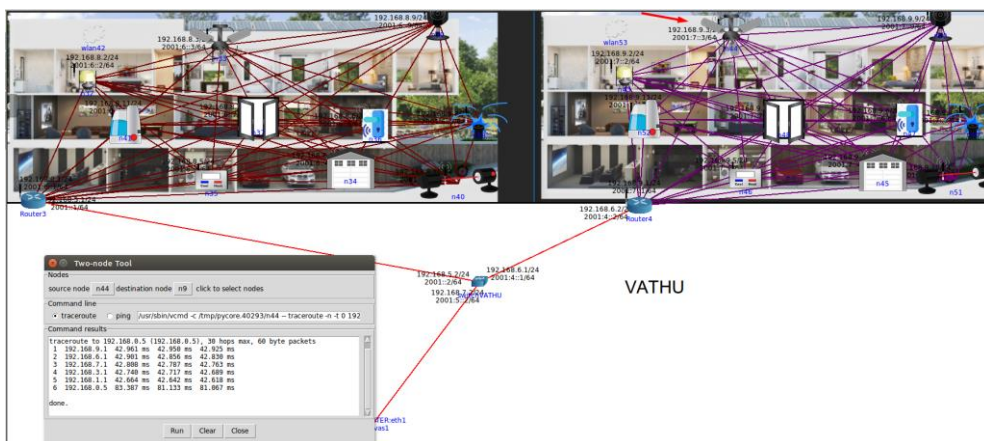
Pings από Ceiling FAN smart home 4->Smart Lock του smart home 1



Εικόνα 9-29 Επικοινωνία των smart home Βαθύ-Καρλόβασι



Traceroute από Ceiling FAN smart home 4->Smart Lock του smart home 1

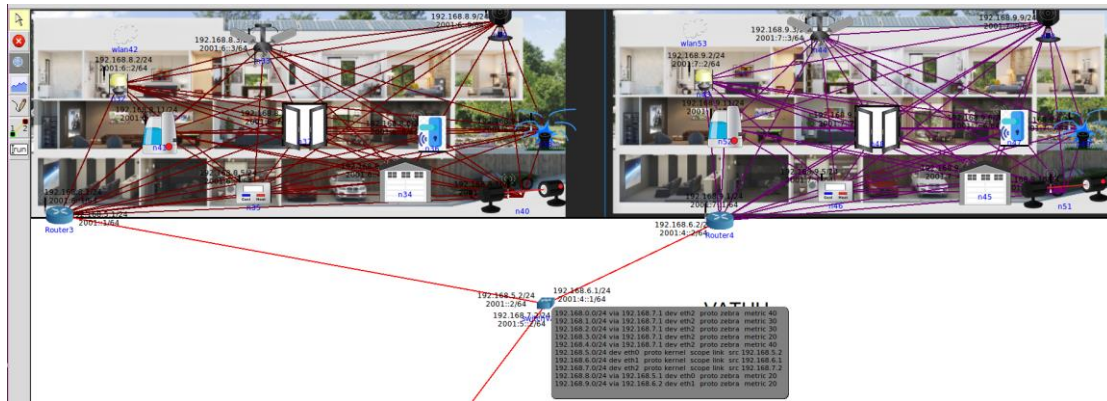


Εικόνα 9-30 Εύρεση Διαδρομής από ανεμιστήρα (home No4)- κλειδαριά(home No1)

Από την παραπάνω εικόνα παρατηρούμε την διαδρομή που διένυσαν τα πακέτα από τον ανεμιστήρα του σπιτιού Νο4 στο Βαθύ μέχρι την έξυπνη κλειδαριά στο σπίτι Νο1 στο Καρλόβασι. Βλέπουμε ότι πραγματοποιήσε 6 hops μέχρι να φτάσει στον προορισμό που θέσαμε.

Router4->switchVATHU->Central_Router->switchKarlovasi->Router1->Lock

Επίσης βλέπουμε ότι ο switchL3VATHU γνωρίζει όλα τα δίκτυα και από τις δύο πόλεις και έτσι επιτυγχάνουμε την ορθή επικοινωνία.



Εικόνα 9-31 IPv4 Routes switchL3VATHU

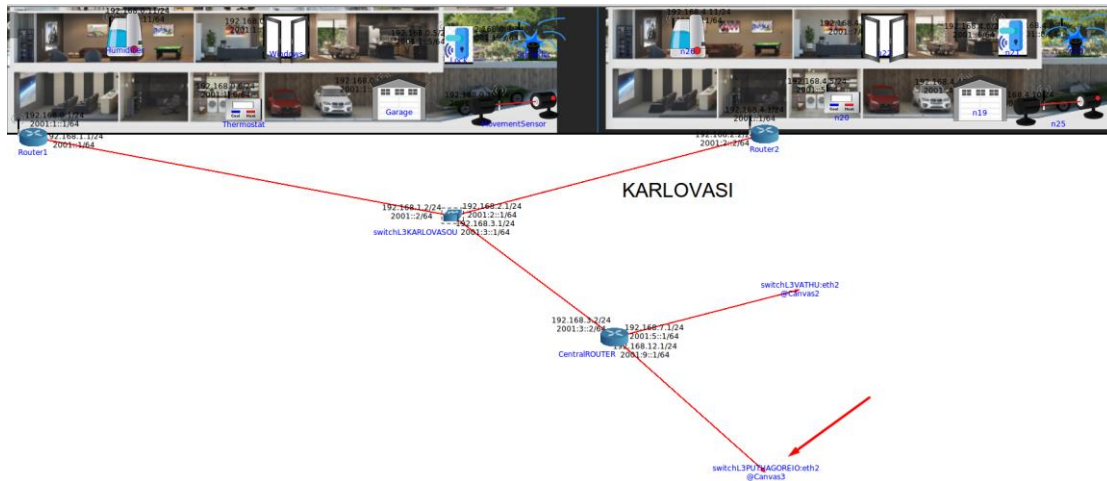
9.5 Έξυπνα Σπίτια Πυθαγορείου

Για την υλοποίηση των έξυπνων σπιτιών στο Πυθαγόρειο κινηθήκαμε στο ίδιο μοτίβο με το Καρλόβασι και το Βαθύ δημιουργώντας σπίτια με αντίστοιχες συσκευές και με ίδια αρχιτεκτονική δικτύου.

Αυτό που έπρεπε να κάνουμε ήταν να δημιουργήσουμε έναν τρίτο καμβά όπου εκεί θα βάζαμε την νέα μας τοπολογία και θα την συνδέαμε με τον πρώτο καμβά.

Όπως έχει αναλυθεί και πιο πάνω όλες οι πόλεις θα παίρνουν από έναν Central Router όπου εμείς για ευκολία των έχουμε τοποθετήσει στον καμβά νούμερο 1 όπου υπάρχουν και τα έξυπνα σπίτια στο Καρλόβασι. Έτσι αυτός θα ήταν ο σύνδεσμος(link) του καμβά 1 προς τον καμβά 3 (Έξυπνα σπίτια Πυθαγόρειο).

Το Core έχει τη δυνατότητα να τρέχει διαφορετικούς καμβάδες ταυτόχρονα, έτσι δημιουργήσαμε σύνδεση από τον Central Router<->switchL3PUTHAGOREIO για την διασύνδεση και των τριών πόλεων, όπως βλέπουμε στα παρακάτω στιγμιότυπα οθόνης. Πατώντας πάνω στο link που έχει τονιστεί με κόκκινο βέλος μεταφερόμαστε στον τρίτο καμβά όπου υπάρχει η πόλη του Πυθαγορείου.



Εικόνα 9-32 Διασύνδεση και των τριών πόλεων

Συνεπώς έχουμε τα εξής δίκτυα στους δρομολογητές της πόλης του Πυθαγορείου:

Central Router : 192.168.3.0/24 ->Σύνδεση με switchKarlovasi

192.168.7.0/24->Σύνδεση με switchVathu

192.168.12.0/24->Σύνδεση με switchPuthagoreio

Router5(Smart home No5 Πυθαγόρειο) : 192.168.10.0/24->Σύνδεση με switchPuthagoreio

192.168.13.0/24->Σύνδεση με το έξυπνο σπίτι No5

Router6(Smart home No6 Πυθαγόρειο) : 192.168.11.0/24->Σύνδεση με switchPuthagoreio

192.168.14.0/24->Σύνδεση με το έξυπνο σπίτι No6

SwitchL3Puthagoreio: 192.168.10.0/24->Σύνδεση με Router5

192.168.11.0/24->Σύνδεση με Router6

192.168.12.0/24->Σύνδεση με Central Router

- Τα πρωτόκολλα που τρέχουν οι δρομολογητές όπως έχουμε αναφέρει και για τα σπίτια του Καρλοβάσου είναι **OSPFv2**, **OSPFv3** και **BGP** στο switch της κάθε πόλης.

Πιο κάτω παραθέτουμε τις IP διευθύνσεις που έχουν ανατεθεί στις συσκευές μέσα στο σπίτι χρησιμοποιώντας το WLAN manager για την ομαδοποίηση ουσιαστικά των Routers με τις συσκευές του σπιτιών No5, No6.



Έξυπνο σπίτι Νο5 ΠΥΘΑΓΟΡΕΙΟ IP διευθύνσεις

| ID | Interface | NAME | IPv4 | IPv6 |
|----|-----------|---------------------|------------------|----------------|
| 1 | eth0 | Router 5 | 192.168.10.1/24 | 2001:0::1/64 |
| 1 | eth1 | Router 5 | 192.168.13.1/24 | 2001:10::1/64 |
| 2 | eth0 | SwitchL3PUTHAGOREIO | 192.168.10.2/24 | 2001:0::2/64 |
| 2 | eth1 | SwitchL3PUTHAGOREIO | 192.168.11.1/24 | 2001:8::1/64 |
| 2 | eth2 | SwitchL3PUTHAGOREIO | 192.168.12.2/24 | 2001:9::2/64 |
| 3 | eth0 | Lamp | 192.168.12.2/24 | 2001:10::2/64 |
| 4 | eth0 | CellingFan | 192.168.13.3/24 | 2001:10::3/64 |
| 5 | eth0 | Garage | 192.168.13.4/24 | 2001:10::4/64 |
| 6 | eth0 | Thermostat | 192.168.13.5/24 | 2001:10::5/64 |
| 7 | eth0 | Smart Lock | 192.168.13.6/24 | 2001:10::6/64 |
| 8 | eth0 | Windows | 192.168.13.7/24 | 2001:10::7/64 |
| 9 | eth0 | Lawn Sprinkler | 192.168.13.8/24 | 2001:10::8/64 |
| 10 | eth0 | Camera | 192.168.13.9/24 | 2001:10::9/64 |
| 11 | eth0 | Movement Sensor | 192.168.13.10/24 | 2001:10::10/64 |
| 12 | eth0 | Humidifier | 192.168.13.11/24 | 2001:10::11/64 |

Πίνακας 9-5 IP διευθύνσεις smart home Νο5

Έξυπνο σπίτι Νο6 ΠΥΘΑΓΟΡΕΙΟ IP διευθύνσεις

| ID | Interface | NAME | IPv4 | IPv6 |
|----|-----------|------------|-----------------|---------------|
| 1 | eth0 | Router 6 | 192.168.11.2/24 | 2001:8::2/64 |
| 1 | eth1 | Router 6 | 192.168.14.1/24 | 2001:11::1/64 |
| 2 | eth0 | Lamp | 192.168.14.2/24 | 2001:11::2/64 |
| 3 | eth0 | CellingFan | 192.168.14.3/24 | 2001:11::3/64 |



| | | | | |
|----|------|-----------------|------------------|----------------|
| 4 | eth0 | Garage | 192.168.14.4/24 | 2001:11::4/64 |
| 5 | eth0 | Thermostat | 192.168.14.5/24 | 2001:11::5/64 |
| 6 | eth0 | Smart Lock | 192.168.14.6/24 | 2001:11::6/64 |
| 7 | eth0 | Windows | 192.168.14.7/24 | 2001:11::7/64 |
| 8 | eth0 | Lawn Sprinkler | 192.168.14.8/24 | 2001:11::8/64 |
| 9 | eth0 | Camera | 192.168.14.9/24 | 2001:11::9/64 |
| 10 | eth0 | Movement Sensor | 192.168.14.10/24 | 2001:11::10/64 |
| 11 | eth0 | Humidifier | 192.168.14.11/24 | 2001:11::11/64 |

Πίνακας 9-6 IP διευθύνσεις smart home Νο6

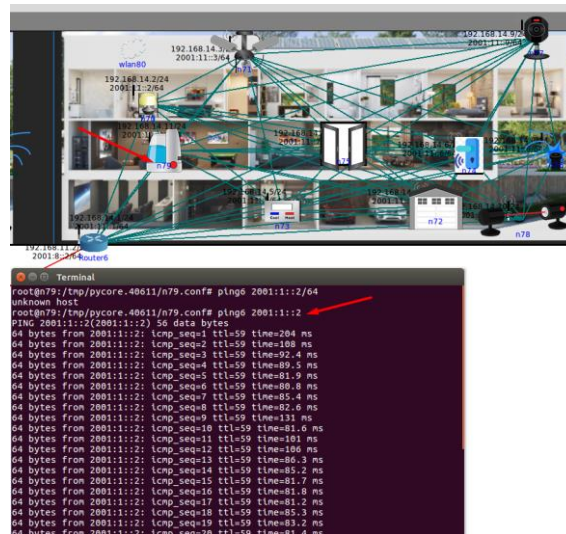
9.5.1 Έλεγχος επικοινωνίας των Smart homes

Θα ελέγξουμε την πλήρη επικοινωνία των σπιτιών από διαφορετικές πόλεις, θα παρατηρήσουμε ότι οι 3 πόλεις που έχουμε επικοινωνούν χάρη στο Central_Router που ενώνει τους 3 καμβάδες και κατ' επέκταση και τις πόλεις. Άρα όλη η κυκλοφορία του δικτύου μας περνά από εκεί για να φτάσει από την μία πόλη στην άλλη.

Ping6 Humidifier(Smart home Νο6 ΠΥΘΑΓΟΡΕΙΟ)->Garage(Smart home Νο1 ΚΑΡΛΟΒΑΣΙ)

Humidifier->IPv6=2001:11::11/64

Garage->IPv6= 2001:1::2/64

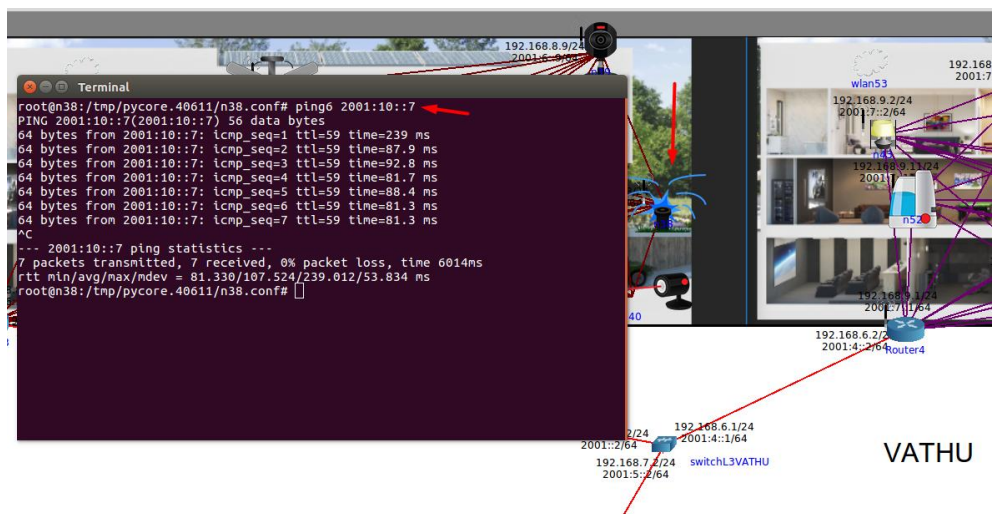


Εικόνα 9-33 ping6 Humidifier(No6)->Garage(No1)

Ping6 Sprinkler(Smart home No3 Βαθύ)->Windows(Smart home No5 Ποθαγόρειο)

Sprinkler->IPv6=2001:6::8/64

Windows->IPv6= 2001:10::7/64



Εικόνα 9-34 ping6 Sprinkler(No3)->Windows(No5)



Traceroute Sprinkler(Smart home No3 Βαθύ)->Windows(Smart home No5 Πυθαγόρειο)

```
root@n38:/tmp/pycore.40611/n38.conf# traceroute 2001:10::7
traceroute to 2001:10::7 (2001:10::7), 30 hops max, 80 byte packets
 1 2001:6::1 (2001:6::1)  97.918 ms  97.836 ms  97.821 ms
 2 2001::2 (2001::2)  97.808 ms  97.795 ms  97.781 ms
 3 2001:5::1 (2001:5::1)  97.765 ms  97.751 ms  97.737 ms
 4 2001:9::2 (2001:9::2)  97.725 ms  97.712 ms  97.699 ms
 5 2001::1 (2001::1)  97.686 ms  97.672 ms  97.655 ms
 6 2001:10::7 (2001:10::7)  231.063 ms  131.017 ms  130.956 ms
root@n38:/tmp/pycore.40611/n38.conf#
```

Εικόνα 9-35 traceroute Sprinkler(No3)->Windows(No5)

Διαδρομή που ακολούθησε το ICMPv6 πακέτο

Router3->switchL3VATHU->Central_Router->switchL3PUTHAGOREIO->Router5->Windows



10 Εγκατάσταση και βασική Παραμετροποίηση με τον πειραματικό δικτυακό εξομοιωτή IMUNES

10.1 Εισαγωγή στο IMUNES

Το IMUNES αποτελεί ακρωνύμιο *Intergrated Multiprotocol Network Emulator/Simulator*. Το Multi-protocol δικαιολογείται από την συνύπαρξη των πρωτοκόλλων IPv4 και IPv6 κατά την εξομοίωση καθώς και για την υποστήριξη πολλαπλών πρωτόκολλων δρομολόγησης και του multicast routing. Το IMUNES όπως και υπόλοιποι εξομοιωτές που δουλέψαμε πιο πάνω είναι βασισμένοι στην ιδέα της ελαφριάς εικονικοποίησης που περιεγράφηκε στο κεφάλαιο 6. Επίσης το IMUNES καθώς τα πακέτα διακινούνται στην εκάστοτε τοπολογία που εξομοιώνουμε, εκτελεί zero copying, δηλαδή η CPU δεν κάνει αντιγραφή δεδομένων από την μία περιοχή μνήμης στην άλλη, αποθηκεύοντας έτσι κύκλους CPU και memory bandwidth. Είναι βασισμένο και δουλεύει σε ένα τροποποιημένο Linux πυρήνα και δίνει τη δυνατότητα στους κόμβους που εξομοιώνονται κάθε φορά να χρησιμοποιούν τυπικές εφαρμογές του UNIX. Τα βασικά πλεονεκτήματα αυτού του εργαλείου είναι η επεκτασιμότητα (scalability), η απόδοση (performance) και η εμπιστοσύνη (fidelity) που προσφέρει. Αναπτύσσει ένα καταναμημένο δίκτυο προσομοίωσης αυξάνοντας την επεκτασιμότητα (scalability) επιτρέποντας τμήματα του cluster που εξομοιώνεται να αναπτύσσονται αυτόνομα. Η αποκεντρωμένη διαχείριση του cluster που εξομοιώνεται βελτιώνει την διαθεσιμότητα και τη ευρωστία του συστήματος. Υποστηρίζει πολυχρηστικό και πειραματικό περιβάλλον έτσι ώστε να εκμεταλλευτεί στο μέγιστο τους συνεχώς αυξανόμενους πόρους (resources). [54]

Στη συνέχεια αναλύουμε λίγο περισσότερο τα βασικά πλεονεκτήματα του εργαλείου που αναφέραμε και πιο πάνω:

- *Βελτιωμένη δυνατότητα κλιμάκωσης* – Όπως αναφέραμε και πιο πάνω αυτό αποτελεί ίσως και το πιο σημαντικό ζήτημα. Η επεκτασιμότητα στο IMUNES επιτυγχάνεται με την χρήση του μοντέλου της ελαφριάς εικονικοποίησης και έτσι δεν υπερκαταναλώνει τους πόρους του host machine. Επίσης το scalability βελτιώθηκε περαιτέρω με τη διέλευση των πακέτων από το ένα VM στο άλλο απλά περνώντας το δείκτη του εκάστοτε πακέτου. Αναμένουμε ότι η κλιμάκωση θα βελτιωθεί ανάλογα με τον αριθμό των καταναμημένων hosts.
- *Βελτιωμένη Ευρωστία* – Η κατάσταση του cluster διατηρείται ενημερωμένη για κάθε host του cluster. Ακόμη και σε περίπτωση βλάβης κάποιου host η κατάσταση του cluster παραμένει διαθέσιμη και ακριβής.
- *Βελτίωση στη χρησιμοποίηση των πόρων (Improved utilization)* – Όλοι οι διαθέσιμοι πόροι του cluster πρέπει να χρησιμοποιούνται με αποτελεσματικό τρόπο. Αυτό σημαίνει ότι επιτρέπεται σε περισσότερα από ένα άτομα να χρησιμοποιούν το cluster και να προσομοιώνουν περισσότερες από μία τοπολογίες στον ίδιο χρόνο.
- *Βελτιωμένη φορητότητα* – Με την έννοια αυτή αναφερόμαστε στη δυνατότητα των χρηστών να λειτουργούν άλλο λειτουργικό σύστημα για να χρησιμοποιήσουν τις δυνατότητες του IMUNES.
- *Διατήρηση της απόδοσης* – Διατηρούνται οι ιδιότητες της προσομοίωσης ακόμη και όταν η προσομοίωση είναι χωρισμένη στους hosts που έχουμε στο cluster.
- *Βελτιωμένη διαθεσιμότητα* – Παρέχεται απομακρυσμένη πρόσβαση στο cluster.
- *Αύξηση της εμπιστοσύνης (fidelity)* – Η εμπιστοσύνη αυξάνεται από το γεγονός ότι όλες οι εικονικές μηχανές διαμένουν στον πυρήνα, οπότε η μετάδοση ενός πακέτου



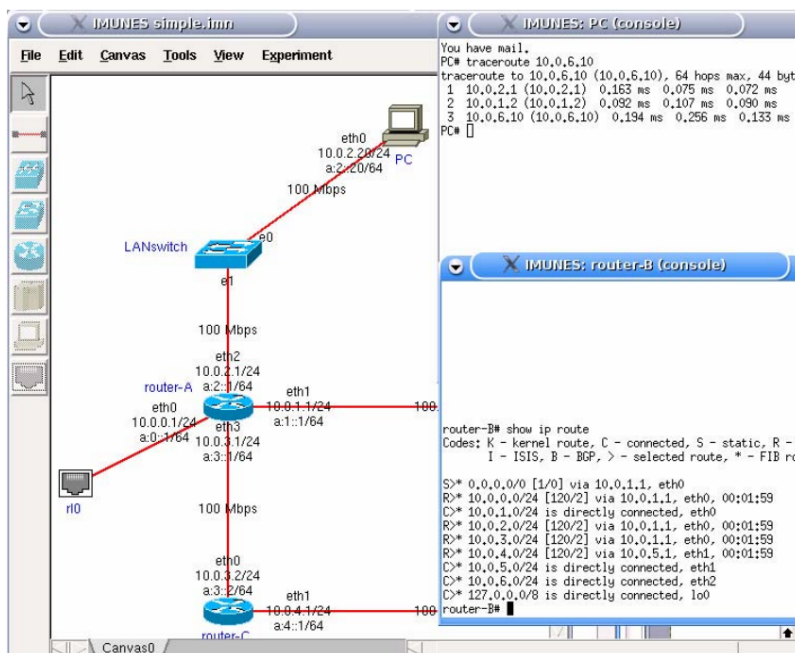
χρησιμοποιεί κλήσεις πραγματικού συστήματος (*real system calls*) που υποβάλλονται σε επεξεργασία μόνο σε ένα πλαίσιο μεταγωγής, έτσι ουσιαστικά βγαίνει ένα αίτημα προς τον πυρήνα.

10.2 Τοπολογία του IMUNES

Το IMUNES χρησιμοποιεί ένα γραφικό περιβάλλον (GUI) για τον σχεδιασμό και την εποπτεία της εκάστοτε τοπολογίας. Η τοπολογία στο IMUNES αποτελείται από δύο βασικές μονάδες : *κόμβους* και *συνδέσμους* (*links*). Οι κόμβοι μπορούν να υπάρχουν ανεξάρτητα ενώ οι σύνδεσμοι πρέπει πάντα να συνδέουν δύο διαφορετικούς κόμβους. Οι κόμβοι ταξινομούνται περαιτέρω σε *link layer nodes* και *network layer nodes*.

Οι κόμβοι *link layer* είναι LAN switch, hub (διανομέας) και physical interfaces και παρέχουν μόνο λειτουργικότητα διασύνδεσης(απλά περνάνε ή απορρίπτουν πακέτα). Δεν υλοποιούνται σαν εικονικές μηχανές είναι απλά netgraph nodes (κόμβοι γράφοι).

Οι κόμβοι *network layer* είναι αυτοί που ασχολούνται με την επεξεργασία του δικτύου και υλοποιούνται στον πυρήνα (kernel) σαν εικονικές μηχανές. Οι κόμβοι *network layers* είναι οι hosts , τα PCs και τα router,έχουν την δική τους TCP/IP στοίβα και παρέχουν την πλήρη λειτουργικότητα του Linux συστήματος. Η διαφορά που υπάρχει στους network layer nodes είναι στη διαδικασία του boot. Οι hosts και τα PCs δεν προωθούν πακέτα και τα routes είναι static, ενώ ο δρομολογητής έχει τη δυνατότητα προώθησης πακέτων χρησιμοποιώντας τις διαδρομές (routes) που λαμβάνει από τα πρωτόκολλα δυναμικής δρομολόγησης που διατίθενται διαμέσων της σουίτας λογισμικού Quagga όπως αναφέραμε και στον εξομοιωτή Core. Ο host αρχίζει κανονικά μέσω του inetd(*internet service daemon) σε αντίθεση με τα PCs. Στους εικονικούς κόμβους δικτύου (network layer) ένα UNIX shell μπορεί να ανοίξει και οποιαδήποτε διαθέσιμη εφαρμογή μπορεί να ξεκινήσει.

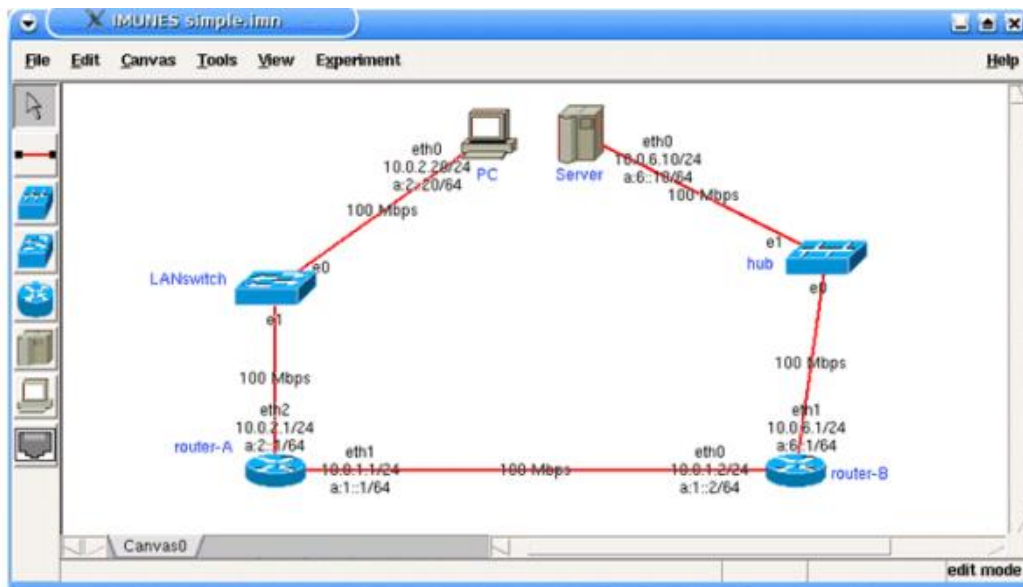


Εικόνα 10-1 Unix shell που ανοίγουμε για τους κόμβους του δικτύου

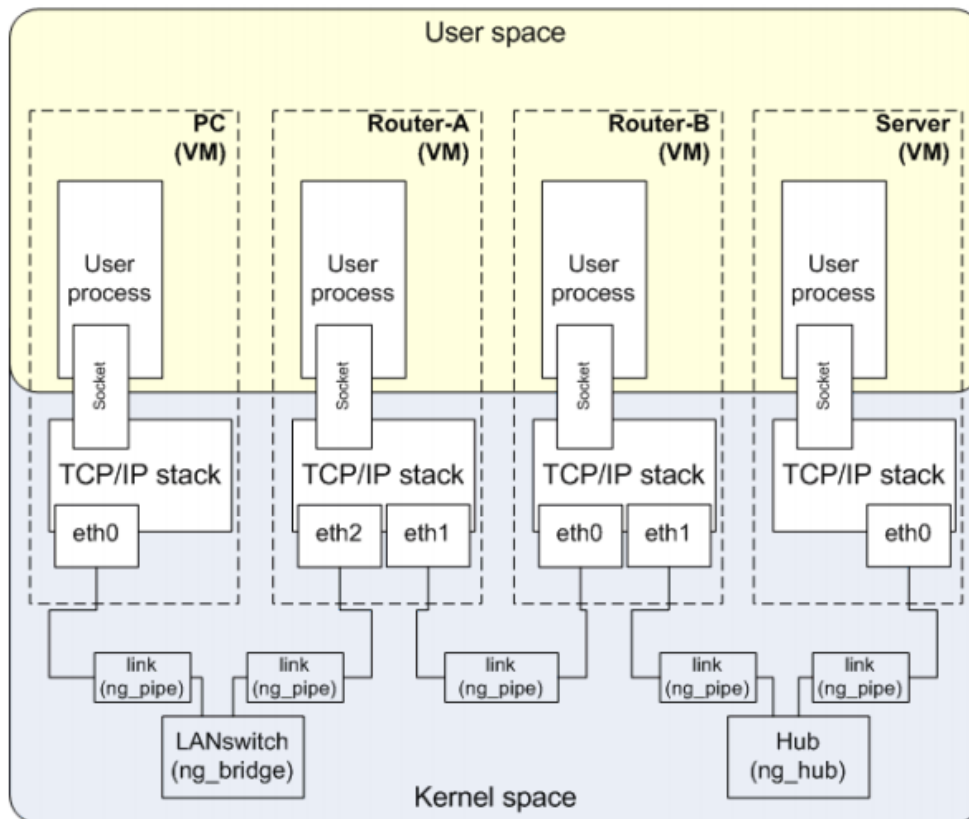


*Το *inetd* (internet service daemon – under the hood program) είναι ένας *super server daemon* σε πολλά Unix συστήματα που παρέχει υπηρεσίες δικτύου, ακούει σε καθορισμένα ports που χρησιμοποιούνται από Internet services όπως FTP, POP3 και telnet.

Στο γραφικό περιβάλλον του IMUNES μπορεί να δημιουργηθεί ένας σύνδεσμος μεταξύ δύο κόμβων. Προεπιλεγμένα εμφανίζονται συνδέσεις τύπου Ethernet αλλά μπορεί να υποστηρίξει και σειριακές (*serial*). Όλοι οι σύνδεσμοι (links) δουλεύουν πλήρως αμφίδρομα (full duplex). Οι σύνδεσμοι τοποθετούνται σε netgraph nodes καθώς και σε link layer nodes. Η συσχέτιση μεταξύ της κάθε τοπολογίας και της κατάλληλης δομής πυρήνα φαίνεται στις παρακάτω εικόνες.



Εικόνα 10-2 Τοπολογία όπως καθορίζεται στο IMUNES



Εικόνα 10-3 Τοπολογία όπως αναπτύσσεται στον πυρήνα (kernel)

Εκτός βεβαίως από τη δημιουργία μονάχα κόμβων και συνδέσμων στο IMUNES, οι ιδιότητες των κόμβων μπορούν να αλλάξουν διαμέσω του GUI. Με αυτόν τον τρόπο μπορούμε να προσομοιώσουμε ένα δίκτυο στο οποίο ορισμένοι κόμβοι ξεκινούν να υλοποιούν DNS services και άλλοι ξεκινούν web servers. Ορισμένοι σύνδεσμοι (links) μπορούν να έχουν υψηλό BER (bit error rate) και κάποιοι άλλοι χαμηλό bandwidth. Η διαθεσιμότητα που προσφέρει το IMUNES σε τυπικές δικτυακές εφαρμογές (πρωτόκολλα και κάποιες υπηρεσίες) και ο τρόπος που προσομοιώνει τις δικτυακές τοπολογίες το καθιστά κατάλληλο για χρήση ως *testbed*.

Επίσης στους κόμβους δικτύου (network layer nodes) του IMUNES είναι διαθέσιμες όλες οι εφαρμογές που χρησιμοποιεί ένα standard UNIX σύστημα όποτε η διαδικασία δοκιμής ενός νέου πρωτοκόλλου είναι απλή και δεν χρειάζεται πολλές αλλαγές στον κώδικα.[54]



10.3 Βήματα εγκατάστασης του εξομοιωτή IMUNES

Για την εγκατάσταση του *IMUNES* όπως και των υπόλοιπων εξομοιωτών που δουλέψαμε (Mininet, Core) χρησιμοποιήσαμε την πλατφόρμα Virtual Box όπου και εκτελέσαμε τα περισσότερα από τα πειράματά μας. Σε αυτόν το εξομοιωτή η έκδοση 16.04 των Ubuntu δεν υποστήριζε κάποια χαρακτηριστικά του εξομοιωτή και κάναμε λήψη την 14.04 η οποία δούλεψε σε αρκετά καλά επίπεδα αν σκεφτούμε ότι χρησιμοποιούμε εικονική μηχανή μέσα σε εικονική μηχανή.[53]

- Πρώτα θα χρειαστεί να κάνουμε λήψη τα απαραίτητα πακέτα που χρειάζεται για να δουλέψει το εργαλείο μας π.χ. (openvswitch, xterm, wireshark, ImageMagick και της γλώσσας tcl)

```
sudo apt-get install openvswitch-switch xterm wireshark ImageMagick tcl tcllib tk user-mode-linux
```

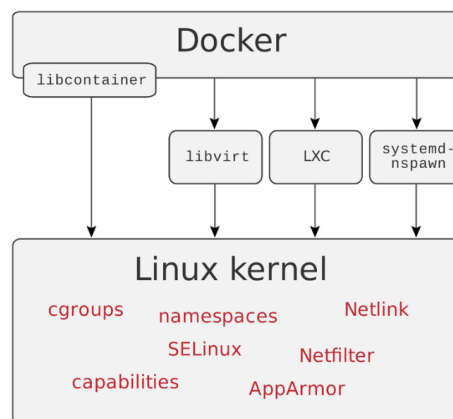
- Στη συνέχεια πρέπει να κάνουμε λήψη την τελευταία έκδοση του Docker.

```
$ wget -qO- https://get.docker.com/ | sh
```

- Εκκίνηση της υπηρεσίας Docker στο σύστημα

```
$ sudo service docker start
```

Το Docker αποτελεί έναν μηχανισμό ανοιχτού κώδικα, με σκοπό την αυτοματοποίηση και την διανομή εφαρμογών μέσα σε δοχεία λογισμικού και την αυτοματοποίηση της εικονικοποίησης των μικρο-υπηρεσιών του Linux. Ουσιαστικά βασίζεται στη λειτουργικότητα του πυρήνα του Linux και απομονώνει πόρους που χρειάζεται όπως έχουμε αναφέρει και στην εικονικοποίηση δικτύου. Έτσι χρησιμοποιούμε το Docker για να έχουμε με πιο απλά λόγια τις δυνατότητες ενός Unix συστήματος στους κόμβους του δικτύου.



Εικόνα 10-4 Docker υποδοχέας (Container)



- Έπειτα θα πρέπει να κατεβάσουμε το εργαλείο *nsenter* η ακόμη πιο συγκεκριμένα μία precompiled βιβλιοθήκη του *nsenter*, που ουσιαστικά θα μας δώσει πρόσβαση στον Docker container που εξομοιώνει τους κόμβους που τρέχουν στην εκάστοτε τοπολογία. Στις νεότερες εκδόσεις των Ubuntu το *nsenter* υπάρχει ήδη στο *util-package*.

```
$ sudo docker run -v /usr/local/bin:/target jpetazzo/nsenter
```

- Τέλος πηγαίνουμε στο repository του Github και κλωνοποιούμε τον κώδικα για να τον χρησιμοποιήσουμε στο δικό μας εικονικό μηχάνημα.

```
$ sudo git clone https://github.com/imunes/imunes.git
```

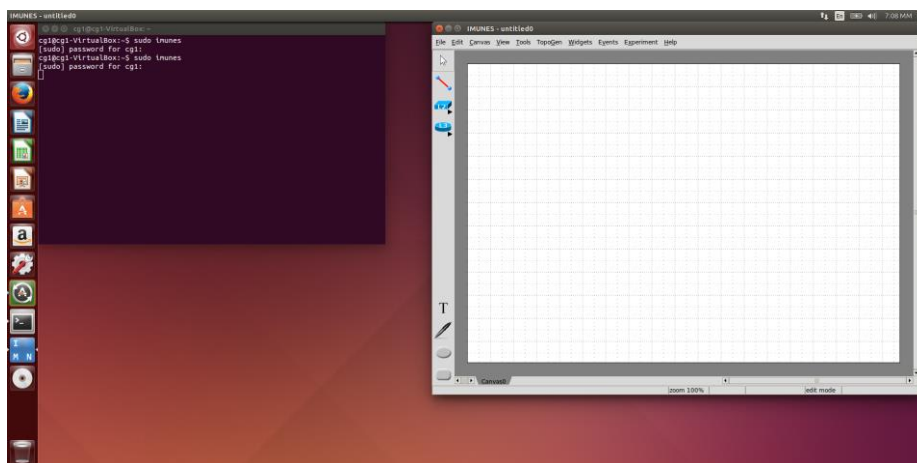
```
$ cd imunes
```

```
$ sudo make install
```

```
$ sudo imunes -p
```

- Εκκίνηση του IMUNES αν δεν τρέχει ο Docker Container πρέπει να τον εκκινήσουμε

```
$ sudo imunes
```



Εικόνα 10-5 Γραφικό Περιβάλλον του IMUNES




10.4 Τοπολογία Διασύνδεσης IMUNES & GNS3 εξομοιωτή

Σε αυτήν την τοπολογία εκμεταλλευόμενοι τον Open Source χαρακτήρα και των δύο εξομοιωτών, διασυνδέσαμε τις δύο πλατφόρμες σε ένα σενάριο έξυπνων σπιτιών που θα επικοινωνούν μεταξύ τους διαμέσων του GNS3. Πιο αναλυτικά κινηθήκαμε σε παρόμοιο μοτίβο με αυτό των διασυνδέσεων που μελετήσαμε και στις παραπάνω ενότητες, με την διαφορά ότι σε αυτό το σενάριο το IMUNES δεν είναι precompiled εικονικό μηχανήμα όπως π.χ. το Mininet, αλλά είναι εγκατεστημένο σε Ubuntu 14.04 με πλήρη γραφικό χαρακτήρα.

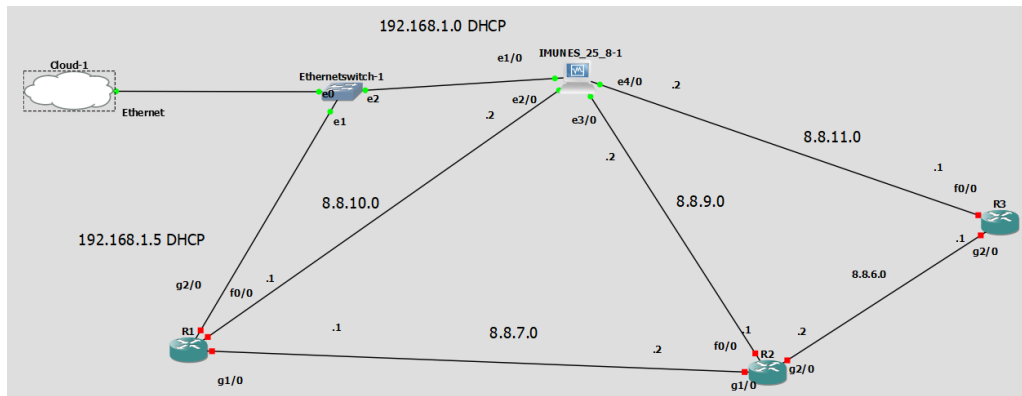
Στην τοπολογία που υλοποιήσαμε έχουμε 3 έξυπνα σπίτια τα οποία τρέχουν στο περιβάλλον του IMUNES, όπου το κάθε σπίτι είναι συνδεδεμένο με έναν κεντρικό δρομολογητή που τρέχει στο GNS3. Η σύνδεση των δρομολογητών στο GNS3 έχει γίνει με την χρήση του BGP και του OSPF πρωτοκόλλου καθώς και της τεχνικής του Redistribution.

Όπως είδαμε και πιο πάνω η διασύνδεση των πλατφορμών σε precompiled εικονικές μηχανές είναι μια διαδικασία γεφύρωσης (bridging), χρησιμοποιώντας τα διαθέσιμα interfaces της εικονικής μηχανής δίνοντας τους IP διεύθυνση είτε με στατικό είτε με δυναμικό τρόπο. Το ζήτημα στην τοπολογία που μελετήσαμε ήταν να γίνει η γεφύρωση σε δύο γραφικά περιβάλλοντα (GNS3-IMUNES). Ουσιαστικά αυτό που έπρεπε να γίνει στην πλευρά που δουλεύουμε το IMUNES ήταν να έχουμε πρόσβαση από το γραφικό περιβάλλον στα διαθέσιμα interfaces της εικονικής μηχανής, έτσι ώστε να επιτύχουμε τη διασύνδεση με την τοπολογία στο GNS3.

Τη λύση σε αυτήν την περίπτωση μας την έδωσε ένα tool που υπάρχει ενσωματωμένο στο γραφικό περιβάλλον του IMUNES το physical interface (RJ45)  External interface, ένα tool που παρέχει τη δυνατότητα σύνδεσης ενός εικονικού κόμβου που υλοποιείται στο IMUNES με ένα physical interface. Έτσι με την χρήση αυτού του tool μπορέσαμε να “πατήσουμε” πάνω στα physical interfaces του IMUNES από το γραφικό περιβάλλον και να τα συνδέσουμε με τον εξομοιωτή GNS3.

10.4.1 Βήματα Διασύνδεσης (Integration)

Η μορφή της τοπολογίας που μελετήσαμε στο GNS3 φαίνεται στο παρακάτω στιγμιότυπο οθόνης καθώς και τα απαραίτητα δίκτυα που έπρεπε να ρυθμίσουμε έτσι ώστε να επιτευχθεί η σύνδεση.



Εικόνα 10-6 Τοπολογία διασύνδεσης GNS3-IMUNES

Στην εικόνα 10-6 βλέπουμε τοποθετημένο τον προσομοιωτή IMUNES που τρέχει σε εικονικό μηχάνημα στο Virtual Box ενσωματωμένο στο project μας. Τα βήματα για την εισαγωγή της εικονικής μηχανής παραλείπονται μιας και έχουν αναφερθεί λεπτομερώς στην ενότητα 6. Προσοχή πρέπει να δοθεί στο αριθμό των interfaces όπου στην προκειμένη περίπτωση πρέπει να είναι 4. Παρατηρούμε ότι το interface e1/0 είναι συνδεδεμένο με τον κόμβο Cloud 1 όπου αντιστοιχεί στην τοπική μας σύνδεση με δίκτυο 192.168.1.0. Λειτουργεί με DHCP πρωτόκολλο το οποίο έχουμε ρυθμίσει από το περιβάλλον του εικονικού μηχανήματος που τρέχει το IMUNES. Τα υπόλοιπα 3 interfaces θα τα ρυθμίσουμε με στατικό τρόπο και θα θέσουμε ανάλογο δίκτυο στο καθένα σύμφωνα με την τοπολογία που μελετάμε. Ο κώδικας για την ρύθμιση των interfaces θα παρατεθεί στη συνέχεια.

Πηγαίνοντας στη συνέχεια στο περιβάλλον του IMUNES ρυθμίζουμε τα interfaces με τον παρακάτω κώδικα, δίνοντας το κατάλληλο path όπου βρίσκονται τα δικτυακά interfaces, όπως έχουμε αναφέρει και στις παραπάνω ενότητες.

`sudo nano /etc/network/interfaces`

```
GNU nano 2.5.3 File: /etc/network/interfaces
Interfaces(5) File used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto eth1
iface eth1 inet dhcp

auto eth2
iface eth2 inet static
address 8.8.10.2
netmask 255.255.255.0
network 8.8.10.0
broadcast 8.8.10.255

auto eth3
iface eth3 inet static
address 8.8.9.2
netmask 255.255.255.0
network 8.8.9.0
broadcast 8.8.9.255

auto eth4
iface eth4 inet static
address 8.8.11.2
netmask 255.255.255.0
network 8.8.11.0
broadcast 8.8.11.255
```

Εικόνα 10-7 Παραμετροποίηση interfaces του IMUNES machine

Στην εικόνα 10-7 βλέπουμε ότι στα interfaces δίνουμε την δεύτερη IP διεύθυνση του εκάστοτε δικτύου καθώς έχουμε δώσει την πρώτη στους δρομολογητές που τρέχουν στο GNS3



όπως βλέπουμε στην εικόνα 10-7. Έπειτα από αυτό το βήμα έχουμε επιτύχει την επικοινωνία μεταξύ *R1-eth2*, *R2-eth3*, *R3-eth4*

```
R1
R1#ping 8.8.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.10.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/10/12
R1#
R1#
R1#
R1#
R1#
R1#

R2
R2#ping 8.8.9.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.9.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/8/12 ms
R2#
R2#
R2#
R2#
R2#

R3
R3#ping 8.8.11.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.11.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/9/12 ms
R3#
R3#
R3#
R3#
R3#
```

Εικόνα 10-8 Επιτυχής επικοινωνία GNS3-IMUNES machine

10.4.2 Redistribution των δρομολογητών με BGP&OSPF πρωτόκολλο

Έχοντας υλοποιήσει με ορθό τρόπο αυτήν τη σύνδεση το επόμενο βήμα που θα κάνουμε είναι να εγκαθιδρύσουμε την επικοινωνία μεταξύ των τριών δρομολογητών (R1,R2,R3). Βλέποντας την εικόνα της τοπολογίας μας, εικόνα 10-6,βλέπουμε ότι ο κεντρικός δρομολογητής R2 είναι αυτός που θα υλοποιήσει ουσιαστικά το Redistribution μεταξύ R1&R3 δρομολογητή. Θα ρυθμίσουμε στατικά τις αντίστοιχες πόρτες όπως βλέπουμε από την τοπολογία και θα θέσουμε σε κάθε δρομολογητή loopback interface ανάλογα με τον αριθμό του π.χ.(R1-> 1.1.1.1). Στη συνέχεια παρατίθεται ο κώδικας που τρέξαμε στους δρομολογητές με την κατάλληλη σειρά για την κατάλληλη ενεργοποίηση των πρωτοκόλλων.



Δρομολογητής R2

```
conf t
```

```
int f0/0
```

```
ip add 8.8.9.1 255.255.255.0
```

```
no shut
```

```
exit
```

```
int g2/0
```

```
ip add 8.8.6.2 255.255.255.0
```

```
no shut
```

```
exit
```

```
int lo0
```

```
ip add 2.2.2.2 255.255.255.255
```

```
no shut
```

```
exit
```

```
int g1/0
```

```
ip add 8.8.7.2 255.255.255.0
```

```
no shut
```

```
end
```

Δρομολογητής R1

```
conf t
```

```
int f0/0
```

```
ip add 8.8.10.1 255.255.255.0
```

```
no shut
```

```
exit
```

```
int g2/0
```

```
ip add 192.168.1.11 255.255.255.0
```



no shut

int lo0

ip add 1.1.1.1 255.255.255.255

no shut

exit

int g1/0

ip add 8.8.7.1 255.255.255.0

no shut

end

Δρομολογητής R3

conf t

int f0/0

ip add 8.8.11.1 255.255.255.0

no shut

exit

int lo0

ip add 3.3.3.3 255.255.255.255

no shut

exit

int g2/0

ip add 8.8.6.1 255.255.255.0

no shut

end



Αυτό που θα κάνουμε στη συνέχεια μετά την απόδοση των παραπάνω διευθύνσεων είναι να υλοποιήσουμε το BGP πρωτόκολλο σε συνδυασμό με το OSPF για την επικοινωνία των δρομολογητών όπως είδαμε και στην ενότητα 8.

Δρομολογητής R2

```
conf t
router bgp 65000
neighbor 1.1.1.1 remote-as 65000
neighbor 3.3.3.3 remote-as 65000
network 8.8.6.0 mask 255.255.255.0
network 8.8.7.0 mask 255.255.255.0
network 2.2.2.2 mask 255.255.255.255
end
```

Δρομολογητής R1

```
conf t
router bgp 65000
neighbor 2.2.2.2 remote-as 65000
neighbor 2.2.2.2 update-source loopback 0
network 8.8.10.0 mask 255.255.255.0
network 192.168.1.0 mask 255.255.255.0 /* Το δίκτυο αυτό εξυπηρετεί μελλοντικούς σκοπούς
που θα αναφέρουμε στο τέλος της ενότητας*/
network 1.1.1.1 mask 255.255.255.255
end
```

Έχοντας διαφημίσει τις εξωτερικές(external) διαδρομές με το πρωτόκολλο BGP θα τρέξουμε το OSPF πρωτόκολλο για την διαφήμιση των εσωτερικών(internal) διαδρομών, για την πλήρη γειτνίαση και επικοινωνία των δρομολογητών.

Δρομολογητής R1

```
conf t
```



```
router ospf 1
network 8.8.7.1 0.0.0.0 area 0
network 1.1.1.1 0.0.0.0 area 0
end
wr
```

Δρομολογητής R2

```
conf t
router ospf 1
network 8.8.7.2 0.0.0.0 area 0
network 8.8.9.1 0.0.0.0 area 0
network 8.8.6.2 0.0.0.0 area 0
network 2.2.2.2 0.0.0.0 area 0
end
```

Στη συνέχεια θα πάμε να τρέξουμε την ίδια διαδικασία από την πλευρά του δρομολογητή R3.

Δρομολογητής R3

```
conf t
router bgp 65000
neighbor 2.2.2.2 remote-as 65000
neighbor 2.2.2.2 update-source loopback 0
network 8.8.6.0 mask 255.255.255.0
network 8.8.11.0 mask 255.255.255.0
network 3.3.3.3 mask 255.255.255.255
end
```

```
-----
conf t
router ospf 1
network 3.3.3.3 0.0.0.0 area 0
network 8.8.6.1 0.0.0.0 area 0
```



end

wr

Αυτό που πρέπει να κάνουμε για να επιτύχουμε την επικοινωνία των R1-R3 είναι να κάνουμε redistribute το OSPF διαμέσων του κεντρικού δρομολογητή R2.

Δρομολογητής R2

conf t

router bgp 65000

redistribute ospf 1

end

wr

10.4.3 Πειράματα επικοινωνίας των δρομολογητών

Έπειτα θα κάνουμε ένα reload όλη την τοπολογία μας και καθώς θα ανοίξουμε το CLI των δρομολογητών θα εμφανιστούν μηνύματα ότι το BGP και το OSPF είναι ενεργοποιημένο και θα διαπιστώσουμε ότι οι 3 δρομολογητές επικοινωνούν μεταξύ τους.

```
*Aug 28 11:33:00.582: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
*Aug 28 11:33:00.587: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet2/0, changed state to up
*Aug 28 11:33:00.859: %SYS-5-CONFIG_I: Configured from memory by console
*Aug 28 11:33:04.083: %SYS-5-RESTART: System restarted -
Cisco IOS Software, 7200 Software (C7200-ADVISERVICEXK9-M), Version 15.2(4)85, RELEASE SOFTWARE © Cisco Systems, Inc.
Compiled Thu 20-Feb-14 06:53 by prod_rel_team
*Aug 28 11:33:04.403: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on GigabitEthernet1/0 is Done
*Aug 28 11:33:45.163: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on GigabitEthernet2/0 is Done
*Aug 28 11:33:49.579: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
*Aug 28 11:34:11.803: %BGP-5-ADJCHANGE: neighbor 3.3.3.3 Up
*Aug 28 11:33:01.479: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Aug 28 11:33:01.515: %LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to up
*Aug 28 11:33:01.527: %LINK-3-UPDOWN: Interface GigabitEthernet2/0, changed state to up
*Aug 28 11:33:01.823: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
*Aug 28 11:33:02.591: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*Aug 28 11:33:02.595: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state to up
*Aug 28 11:33:02.599: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet2/0, changed state to up
*Aug 28 11:33:02.727: %SYS-5-CONFIG_I: Configured from memory by console
*Aug 28 11:33:02.955: %SYS-5-RESTART: System restarted -
Cisco IOS Software, 7200 Software (C7200-ADVISERVICEXK9-M), Version 15.2(4)85, RELEASE SOFTWARE © Cisco Systems, Inc.
Compiled Thu 20-Feb-14 06:53 by prod_rel_team
*Aug 28 11:33:12.387: %BGP-4-ADJCHG_ASSGN: Interface GigabitEthernet2/0 assigned BGP address mask 255.255.0.0, hostname R1
*Aug 28 11:33:43.639: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet1/0 from LDR: 2.2.2.2 is Done
*Aug 28 11:33:48.847: %BGP-5-ADJCHANGE: neighbor 2.2.2.2 Up
*Aug 28 11:33:04.591: %SYS-5-RESTART: System restarted -
Cisco IOS Software, 7200 Software (C7200-ADVISERVICEXK9-M), Version 15.2(4)85, RELEASE SOFTWARE © Cisco Systems, Inc.
Compiled Thu 20-Feb-14 06:53 by prod_rel_team
*Aug 28 11:33:05.515: %LINK-5-CHANGED: Interface GigabitEthernet1/0, changed state to up
*Aug 28 11:33:05.607: %LINK-5-CHANGED: Interface GigabitEthernet3/0, changed state to up
*Aug 28 11:33:06.897: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state to up
*Aug 28 11:33:06.897: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet3/0, changed state to up
*Aug 28 11:33:44.787: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet2/0 is Done
*Aug 28 11:34:11.363: %BGP-5-ADJCHANGE: neighbor 2.2.2.2 Up
```

Εικόνα 10-9 Μηνύματα ενημέρωσης της επιτυχούς λειτουργίας των πρωτοκόλλων

Pings R1-R3 & R3-R1

| | | |
|------|----|----------------|
| g2/0 | R1 | 192.168.1.5/24 |
| g1/0 | R1 | 8.8.7.1/24 |
| f0/0 | R1 | 8.8.10.1/24 |
| f0/0 | R3 | 8.8.11.1/24 |



g2/0 R3 8.8.6.1/24

```
R1#ping 8.8.6.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.6.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/40/44 ms
R1#ping 8.8.11.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.11.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/41/44 ms
R1#
```

Εικόνα 10-10 Επιτυχής επικοινωνία R1->R3

```
R3#ping 8.8.7.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.7.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/32 ms
R3#ping 8.8.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.10.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/40/44 ms
R3#ping 192.168.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/39/44 ms
R3#
```

Εικόνα 10-11 Επιτυχής επικοινωνία R3->R1

Traceroute R1-R3 & R3-R1

```
R1#traceroute 8.8.11.1
Type escape sequence to abort.
Tracing the route to 8.8.11.1
VRF info: (vrf in name/id, vrf out name/id)
 1 8.8.7.2 12 msec 20 msec 20 msec
 2 8.8.6.1 32 msec 32 msec 32 msec
R1#
```


Εικόνα 10-12 Εύρεση διαδρομής διαμέσων του R2 από τον R1

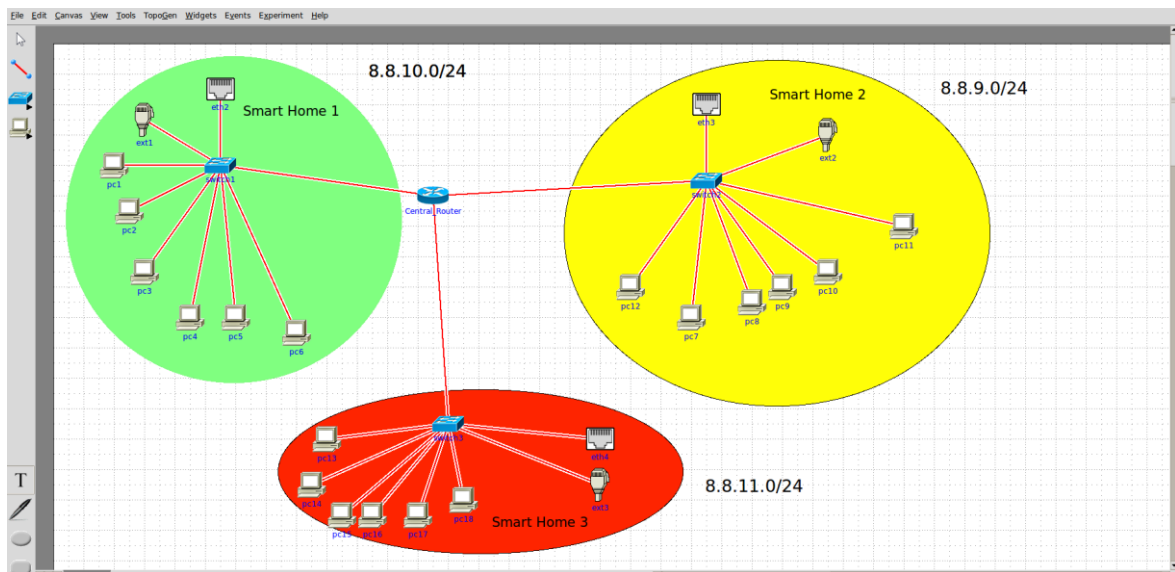
```
R3#traceroute 192.168.1.5
Type escape sequence to abort.
Tracing the route to 192.168.1.5
VRF info: (vrf in name/id, vrf out name/id)
 1 8.8.6.2 20 msec 20 msec 24 msec
 2 8.8.7.1 40 msec 40 msec 44 msec
R3#
```

Εικόνα 10-13 Εύρεση διαδρομής διαμέσων του R2 από τον R3



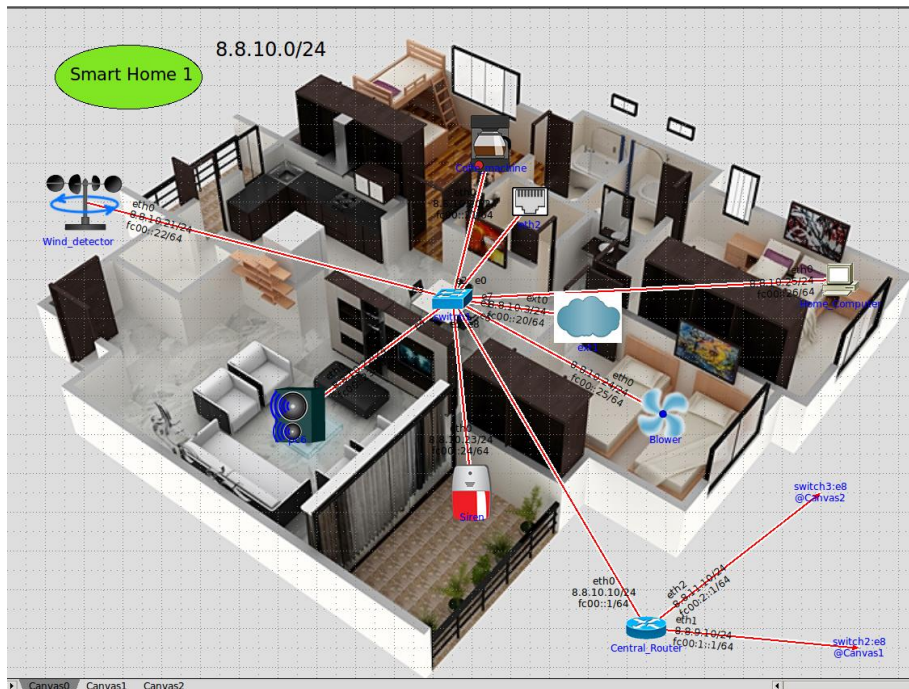
10.4.4 Τοπολογία έξυπνων σπιτιών στο περιβάλλον του IMUNES

Στο περιβάλλον του IMUNES κινηθήκαμε σε παρόμοιο μοτίβο με αυτή του εξομοιωτή Core, μιας και το IMUNES είναι ουσιαστικά ο πρόγονος του Core. Δημιουργήσαμε 3 έξυπνα σπίτια με 6 συσκευές στο κάθε σπίτι, όπου όλες οι συσκευές είναι συνδεδεμένες με ένα κεντρικό switch που διαθέτει το κάθε σπίτι και κάθε switch είναι συνδεδεμένο με το αντίστοιχο RJ45 tool που ουσιαστικά είναι η αντίστοιχη ethernet port που συνδέει κάθε σπίτι που τρέχει στο IMUNES με τον αντίστοιχο δρομολογητή που είναι συνδεδεμένος σε αυτό ethernet port στο GNS3. Επίσης σε κάθε σπίτι έχουμε και το tool  External connection που ουσιαστικά καθορίζει τις IP που θα δοθούν στις συσκευές του δικτύου, δηλαδή ποιου δικτύου την IP θα έχουν, είναι μία μορφή controller που χρησιμοποιεί ο εξομοιωτής. Τέλος τα switch των έξυπνων σπιτιών θα συνδέονται μεταξύ τους σε ένα κεντρικό δρομολογητή ο οποίος θα τρέχει το πρωτόκολλο RIPv2, RIPng(IPv6) και θα συνδέει τα σπίτια μεταξύ τους. Στη συνέχεια παρουσιάζεται η μορφή της τοπολογίας στο IMUNES για την καλύτερη κατανόηση της, χωρίς την παραμετροποίηση που θα δώσουμε στο background και στις εικόνες των τερματικών.




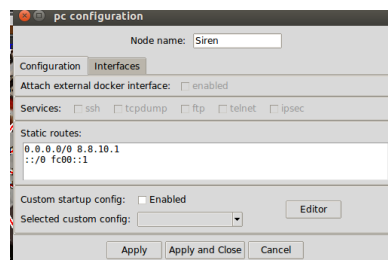
Εικόνα 10-13 Τοπολογία Έξυπνων σπιτιών στο IMUNES

Επιπλέον αυτό που κάναμε είναι η εισαγωγή νέων εικόνων για τα τερματικά που θα προσομοιάσουμε και ενός νέου background για τον καμβά του εργαλείου όπου θα παρουσιάζεται η κάτοψη ενός σπιτιού έχοντας μια κλίση. Επίσης κινηθήκαμε με παρόμοιο τρόπο όπως και με το Core μόνο που εδώ δεν μπορούμε να έχουμε wireless συνδέσεις παρά μόνο wired.



Εικόνα 10-14 Τοπολογία Smart Home 1

Εκμεταλλευόμενοι τον open-source χαρακτήρα του εργαλείου, χρησιμοποιήσαμε ορισμένες αυτοματοποιημένες ενέργειες σαν την αυτόματη ρύθμιση της IP διεύθυνσης των τερματικών. Καθώς κάνουμε drag and drop τα τερματικά τρέχει στο background ένα script που τους δίνει IP σύμφωνα πάντα με τον Manager(External connection tool) που έχουμε θέσει στο σενάριο. Εμείς προσομοιώνουμε το external connection tool με Cloud . Έτσι έχοντας συνδέσει τον Manager με το switch του κάθε σπιτιού και θέτοντας IP 8.8.9.3/24 με δεξιά κλικ->Configure, όπου την πρώτη IP την έχει ο αντίστοιχος δρομολογητής (R1) στο GNS3 και την δεύτερη το eth2 του IMUNES-VM, δημιουργούμε τον κορμό του δικτύου σε κάθε σπίτι. Στη συνέχεια συνδέοντας τα τερματικά στο switch παίρνουν αυτόματα IP του δικτύου 8.8.9.0/24 ξεκινώντας από την .20 που ορίζει το script που τρέχει ο RJ45. Επίσης γίνεται και η προσθήκη ενός static route στον configuration table του κάθε τερματικού που αντιστοιχεί με την ethernet port που συνδέεται ο RJ45.



Εικόνα 10-15 Static route ethernet port



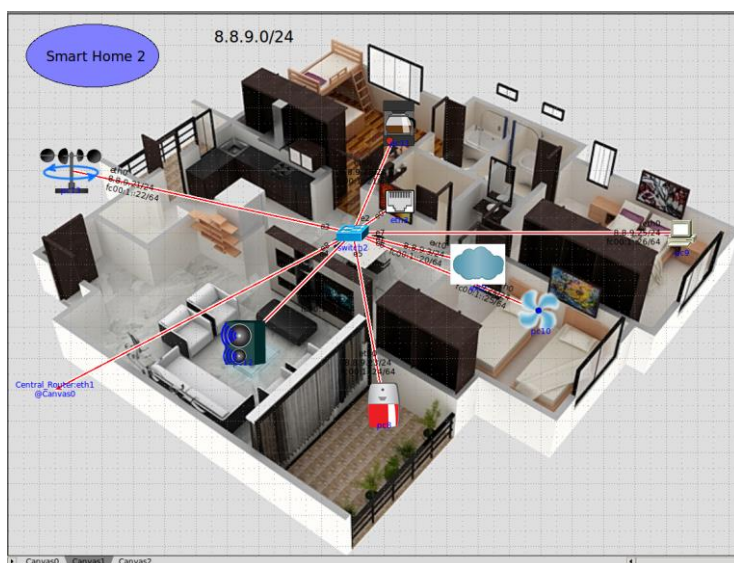
Παρακάτω βλέπουμε τις IP που έχουμε στο Smart Home 1, καθώς και τις IP που έχει ο Central_Router που συνδέει τα σπίτια μεταξύ τους διαφημίζοντας τα δίκτυα με το πρωτόκολλο RIP.

Smart Home 1

| ID | Interface | NAME | IPv4 | IPv6 |
|----|-----------|------------------------------|--------------|-----------------------------|
| 1 | eth2 | IMUNES-VM | 8.8.10.2/24 | fe80::a00:27ff:fe1a:5005/64 |
| 2 | ext0 | Manager(External Connection) | 8.8.10.3/24 | fc00::20/64 |
| 3 | eth0 | Central_Router | 8.8.10.10/24 | fc00::1/64 |
| 3 | eth1 | Central_Router | 8.8.9.10/24 | fc00:1::1/64 |
| 3 | eth2 | Central_Router | 8.8.11.10/24 | fc00:2::1/64 |
| 4 | eth0 | Coffee_machine | 8.8.10.20/24 | fc00::21/64 |
| 5 | eth0 | WindDetector | 8.8.10.21/24 | fc00::22/64 |
| 6 | eth0 | Speaker | 8.8.10.22/24 | fc00::23/64 |
| 7 | eth0 | Siren | 8.8.10.23/24 | fc00::24/64 |
| 8 | eth0 | Blower | 8.8.10.24/24 | fc00::25/64 |
| 9 | eth0 | HomeComputer | 8.8.10.25/24 | fc00::26/64 |

Πίνακας 10-1 Διευθύνσεις IP του Smart Home 1

Με ίδιο τρόπο αλλά σε διαφορετικό καμβά έχουμε δουλέψει για το Smart Home 2 και το Smart Home 3 όπου η σύνδεση των καμβάδων γίνεται διαμέσων του Central_Router με δεξιά κλικ -> create link to->canvas 1,canvas2.



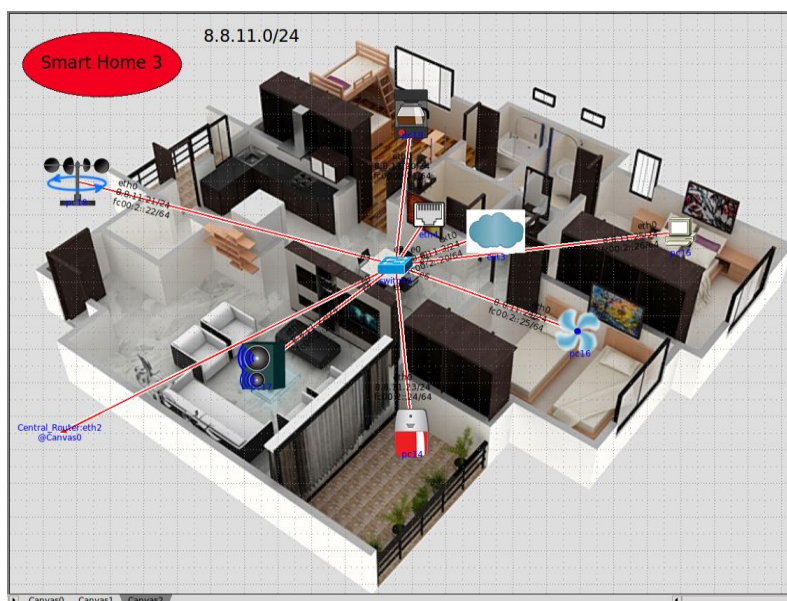
Εικόνα 10-16 Smart Home 2



Smart Home 2

| ID | Interface | NAME | IPv4 | IPv6 |
|----|-----------|------------------------------|-------------|-----------------------------|
| 1 | eth3 | IMUNES-VM | 8.8.9.2/24 | fe80::a00:27ff:fe27:f57a/64 |
| 2 | ext0 | Manager(External Connection) | 8.8.9.3/24 | fc00:1::20/64 |
| 3 | eth0 | Coffee_machine | 8.8.9.20/24 | fc00:1::21/64 |
| 4 | eth0 | WindDetector | 8.8.9.21/24 | fc00:1::22/64 |
| 5 | eth0 | Speaker | 8.8.9.22/24 | fc00:1::23/64 |
| 6 | eth0 | Siren | 8.8.9.23/24 | fc00:1::24/64 |
| 7 | eth0 | Blower | 8.8.9.24/24 | fc00:1::25/64 |
| 8 | eth0 | HomeComputer | 8.8.9.25/24 | fc00:1::26/64 |

Πίνακας 10-2 Διευθύνσεις IP του Smart Home 2



Εικόνα 10-17 Smart Home 3



Smart Home 3

| ID | Interface | NAME | IPv4 | IPv6 |
|----|-----------|------------------------------|--------------|-----------------------------|
| 1 | eth4 | IMUNES-VM | 8.8.11.2/24 | fe80::a00:27ff:fed2:b3b8/64 |
| 2 | ext0 | Manager(External Connection) | 8.8.11.3/24 | fc00:2::20/64 |
| 3 | eth0 | Coffee_machine | 8.8.11.20/24 | fc00:2::21/64 |
| 4 | eth0 | WindDetector | 8.8.11.21/24 | fc00:2::22/64 |
| 5 | eth0 | Speaker | 8.8.11.22/24 | fc00:2::23/64 |
| 6 | eth0 | Siren | 8.8.11.23/24 | fc00:2::23/64 |
| 7 | eth0 | Blower | 8.8.11.24/24 | fc00:2::24/64 |
| 8 | eth0 | HomeComputer | 8.8.11.25/24 | fc00:2::24/64 |

Πίνακας 10-3 Διευθύνσεις IP του Smart Home 3

Ακόμη βλέπουμε και τις διαθέσιμες IP διευθύνσεις που έχουμε στους δρομολογητές που τρέχουν στο GNS3 και στη συνέχεια θα ελέγξουμε ότι οι τοπολογίες μας επικοινωνούν ορθώς.

| ID | Interface | NAME | IPv4 |
|----|-----------|-----------|----------------|
| 1 | g2/0 | R1 | 192.168.1.5/24 |
| 1 | g1/0 | R1 | 8.8.7.1/24 |
| 1 | f0/0 | R1 | 8.8.10.1/24 |
| 2 | g1/0 | R2 | 8.8.7.2/24 |
| 2 | f0/0 | R2 | 8.8.9.1/24 |
| 2 | g2/0 | R2 | 8.8.6.2/24 |
| 3 | f0/0 | R3 | 8.8.11.1/24 |
| 3 | g2/0 | R3 | 8.8.6.1/24 |
| 4 | eth1 | IMUNES-VM | 192.168.1.7/24 |
| 4 | eth2 | IMUNES-VM | 8.8.10.2/24 |
| 4 | eth3 | IMUNES-VM | 8.8.9.2/24 |
| 4 | eth4 | IMUNES-VM | 8.8.11.2/24 |

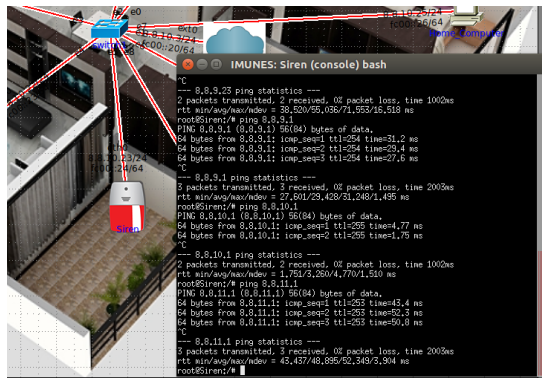
Πίνακας 10-4 Διευθύνσεις IP GNS3 project



10.5 Έλεγχος Επικοινωνίας της διασυνδεδεμένης τοπολογίας

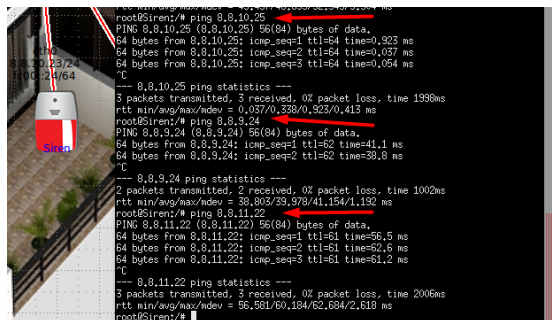
Θα επιχειρήσουμε σε αυτό το σημείο Pings από μία συσκευή σε κάθε σπίτι προς τις IP των συσκευών που βρίσκονται σε άλλα σπίτια και προς όλους τους δρομολογητές που τρέχουν στο GNS3 καθώς και pings από τους δρομολογητές του GNS3 προς τις έξυπνες συσκευές που τρέχουν στο IMUNES.

Pings Siren(Smart Home 1)->R1-R2-R3



Εικόνα 10-18 Pings Siren(SH1)->R1,R2,R3

Pings Siren(Smart Home 1)->Home Computer-Blower(Smart Home 2)-R3(Speaker)



Εικόνα 10-19 Pings Siren(SH1)->Home Computer,Blower(SH2),Speaker(SH3)



Pings Blower(Smart Home 2)->R1,R2,R3

```
IMUNES: pc10 (console) bash
root@pc10:~# ping 8.8.9.1
PING 8.8.9.1 (8.8.9.1) 56(84) bytes of data:
64 bytes from 8.8.9.1: icmp_seq=1 ttl=255 time=7.83 ms
64 bytes from 8.8.9.1: icmp_seq=2 ttl=255 time=5.22 ms
^C
--- 8.8.9.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/ndev = 5.226/6.528/7.831/1.305 ms
root@pc10:~# ping 8.8.10.1
PING 8.8.10.1 (8.8.10.1) 56(84) bytes of data:
64 bytes from 8.8.10.1: icmp_seq=1 ttl=254 time=17.2 ms
64 bytes from 8.8.10.1: icmp_seq=2 ttl=254 time=15.2 ms
^C
--- 8.8.10.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/ndev = 15.234/16.250/17.267/1.024 ms
root@pc10:~# ping 8.8.11.1
PING 8.8.11.1 (8.8.11.1) 56(84) bytes of data:
64 bytes from 8.8.11.1: icmp_seq=1 ttl=254 time=29.0 ms
64 bytes from 8.8.11.1: icmp_seq=2 ttl=254 time=28.0 ms
^C
--- 8.8.11.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/ndev = 28.085/28.591/29.097/0.506 ms
root@pc10:~#
```

Εικόνα 10-19 Pings Blower(SH2)->R1,R2,R3

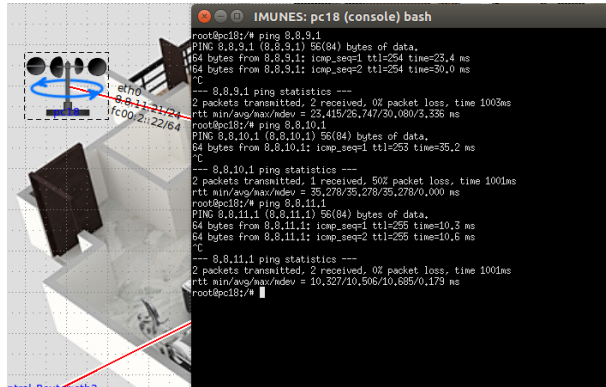
Pings Blower(Smart Home 2)->Coffe_machine-WindDetector(Smart Home 1)-Siren(Smart Home 3)

```
IMUNES: pc10 (console) bash
^C
--- 8.8.11.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/ndev = 28.085/28.591/29.097/0.506 ms
root@pc10:~# ping 8.8.9.20
PING 8.8.9.20 (8.8.9.20) 56(84) bytes of data:
64 bytes from 8.8.9.20: icmp_seq=1 ttl=64 time=0.784 ms
64 bytes from 8.8.9.20: icmp_seq=2 ttl=64 time=0.038 ms
^C
--- 8.8.9.20 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/ndev = 0.038/0.411/0.784/0.373 ms
root@pc10:~# ping 8.8.10.21
PING 8.8.10.21 (8.8.10.21) 56(84) bytes of data:
64 bytes from 8.8.10.21: icmp_seq=1 ttl=62 time=40.1 ms
64 bytes from 8.8.10.21: icmp_seq=2 ttl=62 time=40.9 ms
^C
--- 8.8.10.21 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1019ms
rtt min/avg/max/ndev = 40.179/40.561/40.944/0.432 ms
root@pc10:~# ping 8.8.11.23
PING 8.8.11.23 (8.8.11.23) 56(84) bytes of data:
64 bytes from 8.8.11.23: icmp_seq=1 ttl=62 time=77.3 ms
64 bytes from 8.8.11.23: icmp_seq=2 ttl=62 time=41.8 ms
64 bytes from 8.8.11.23: icmp_seq=3 ttl=62 time=39.4 ms
^C
--- 8.8.11.23 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/ndev = 39.467/52.883/77.334/17.317 ms
root@pc10:~#
```

Εικόνα 10-20 Pings Blower(SH2)->Coffe_machine-WindDetector(SH1)-Siren(SH3)

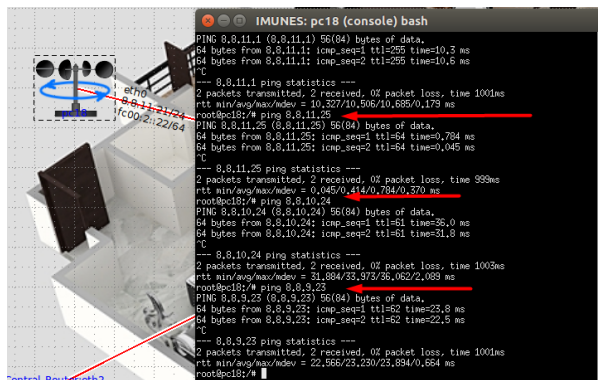


Pings WindDetector(Smart Home 3)->R1,R2,R3



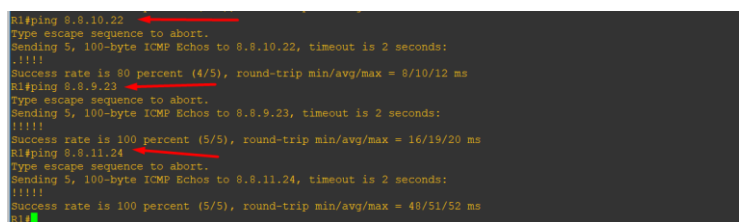
Εικόνα 10-21 Pings WindDetector(SH3)->R1,R2,R3

Pings WindDetector(Smart Home 3)->Home_Computer-Blower (Smart Home 1)-Siren(Smart Home 2)



Εικόνα 10-22 Pings WindDetector(SH3)->Home_Computer-Blower(SH1)-Siren(SH2)

Pings R1->Speaker(Smart Home 1)-Siren(Smart Home 2)-Blower(Smart Home 3)



Εικόνα 10-23 Pings R1->Speaker(SH1)-Siren(SH2)-Blower(SH3)



Pings R2->Coffe_machine(Smart Home 1)-WindDetector(Smart Home 3)-Speaker(Smart Home 2)

```
R2#ping 8.8.10.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.10.20, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/39/72 ms
R2#ping 8.8.11.21
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.11.21, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms
R2#ping 8.8.9.22
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.9.22, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/9/12 ms
R2#
```

Εικόνα 10-24 Pings R2->Coffe_Machine(SH1)-WindDetector(SH3)-Speaker(SH2)

Pings R3->HomeComputer (Smart Home 2)-Siren(Smart Home 1)-Coffe_machine(Smart Home 3)

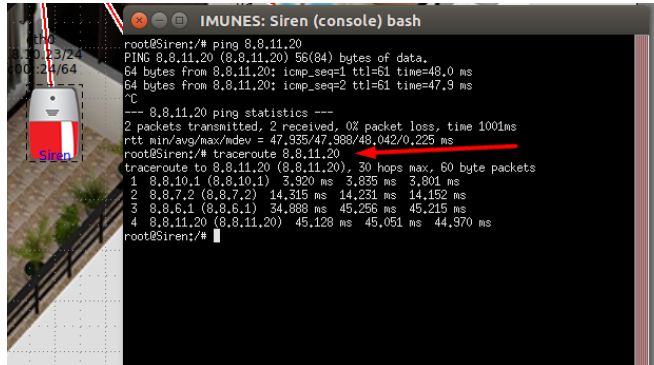
```
R3#ping 8.8.9.25
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.9.25, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/25/48 ms
R3#ping 8.8.10.23
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.10.23, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/32 ms
R3#ping 8.8.11.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.11.20, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/12 ms
R3#
```

Εικόνα 10-25 Pings R3->HomeComputer (SH2)-Siren(SH1)-Coffe_machine(SH3)

Από τα παραπάνω στιγμιότυπα οθόνης παρατηρούμε ότι ο μηχανισμός που δημιουργήσαμε λειτουργεί σωστά και έχουμε πλήρης επικοινωνία μεταξύ των τοπολογιών που έχουμε υλοποιήσει χωρίς κάποια απώλεια στα πακέτα που στέλνουμε. Θα δούμε και ορισμένα *traceroutes* από τις συσκευές που τρέχουν στο IMUNES προς ορισμένες IP άλλων συσκευών διαφορετικών σπιτιών και θα δούμε ότι κάθε πακέτο επιλέγει την διαδρομή που συνδέεται με το RJ45(eth2,eth3,eth4) και στη συνέχεια διασχίζει την τοπολογία των δρομολογητών στο GNS3 και από την αντίστοιχη ethernet port εισέρχεται στη συσκευή του σπιτιού.

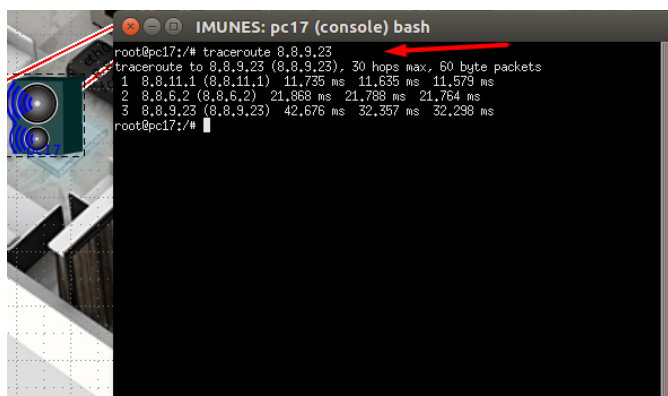


Traceroute Siren(Smart Home 1)-Coffe machine(Smart Home 3)



Εικόνα 10-26 Traceroute Siren(SH1)-Coffe machine(SH3)

Traceroute Speaker(Smart Home 3)-Siren (Smart Home 2)



Εικόνα 10-27 Traceroute Speaker(SH3)-Siren (SH2)

Από τις εικόνες 10-26 και 10-27 βλέπουμε τις διαδρομές που επιλέγουν οι συσκευές διαμέσων των ethernet port των δρομολογητών που τρέχουμε στο GNS3. Οι δρομολογητές του GNS3 για



να έχουν πρόσβαση σε συσκευές σπιτιών από διαφορετικό δίκτυο από ότι το δικό τους επιλέγουν διαδρομές των γειτονικών τους δρομολογητών.

10.5.1 Wireshark Captures

Θα ανοίξουμε το Wireshark packet sniffer στον κόμβο Central_Router με *δεξιά κλικ* > *Wireshark* > *επιλογή ethernet port* και θα δούμε τα μηνύματα που ανταλλάσσονται από το πρωτόκολλο RIPv2 για IPv4, το RIPng για IPv6 και το OSPF πακέτο που τρέχει στο GNS3 στον R2 δρομολογητή ο οποίος υλοποιεί το Redistribution και περνάει στο περιβάλλον του IMUNES διαμέσων του ethernet port 2.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|-----------------------|-------------|----------|--------|--|
| 1 | 0.000000 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 86 | Command Request, Version 1 |
| 2 | 0.000041 | 8.8.10.10 | 224.0.0.9 | RIPv2 | 66 | Request |
| 3 | 0.000042 | 8.8.10.10 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0.0.9 for any sources |
| 4 | 0.739556 | 8.8.10.10 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0.0.9 for any sources |
| 5 | 16.371602 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 6 | 32.967749 | 8.8.10.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 7 | 33.388657 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 8 | 60.982348 | 8.8.10.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 9 | 69.399982 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 10 | 90.984590 | 8.8.10.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 11 | 103.431126 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 12 | 126.420992 | 8.8.10.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 13 | 131.675924 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 14 | 154.821716 | 8.8.10.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 15 | 176.468983 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 16 | 186.809390 | 8.8.10.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 17 | 204.515920 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 18 | 221.068438 | 8.8.10.10 | 224.0.0.9 | RIPv2 | 86 | Response |

Εικόνα 10-28 RIP αιτήσεις στο ethernet port 0 του δικτύου 8.8.10.0

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|-----------------------|-------------|----------|--------|--|
| 1 | 0.000000 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 86 | Command Request, Version 1 |
| 2 | 0.000013 | 8.8.11.10 | 224.0.0.9 | RIPv2 | 66 | Request |
| 3 | 0.000411 | 8.8.11.10 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0.0.9 for any sources |
| 4 | 0.470213 | 8.8.11.10 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0.0.9 for any sources |
| 5 | 10.383882 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 6 | 24.383901 | 8.8.11.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 7 | 48.874412 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 8 | 53.413615 | 8.8.11.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 9 | 82.414054 | 8.8.11.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 10 | 88.914724 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 11 | 115.410076 | 8.8.11.10 | 224.0.0.9 | RIPv2 | 86 | Response |

Εικόνα 10-29 RIP αιτήσεις στο ethernet port 2 του δικτύου 8.8.11.0

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|-----------------------|-------------|----------|--------|--|
| 1 | 0.000000 | 8.8.9.10 | 224.0.0.5 | OSPF | 90 | Hello Packet |
| 2 | 0.201529 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 86 | Command Request, Version 1 |
| 3 | 0.201548 | 8.8.9.10 | 224.0.0.9 | RIPv2 | 66 | Request |
| 4 | 0.206862 | 8.8.9.10 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0.0.9 for any sources |
| 5 | 0.738230 | 8.8.9.10 | 224.0.0.22 | IGMPv3 | 54 | Membership Report / Join group 224.0.0.9 for any sources |
| 6 | 1.000410 | 8.8.9.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 7 | 9.026596 | 8.8.9.1 | 224.0.0.5 | OSPF | 90 | Hello Packet |
| 8 | 11.370157 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 9 | 18.068244 | 8.8.9.1 | 224.0.0.5 | OSPF | 90 | Hello Packet |
| 10 | 28.294424 | 8.8.9.1 | 224.0.0.5 | OSPF | 90 | Hello Packet |
| 11 | 31.115958 | 8.8.9.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 12 | 37.868482 | 8.8.9.1 | 224.0.0.5 | OSPF | 90 | Hello Packet |
| 13 | 47.318382 | 8.8.9.1 | 224.0.0.5 | OSPF | 90 | Hello Packet |
| 14 | 55.606185 | fe80::4000:aaff:fee:: | ff02::9 | RIPng | 126 | Command Response, Version 1 |
| 15 | 56.921363 | 8.8.9.1 | 224.0.0.5 | OSPF | 90 | Hello Packet |
| 16 | 60.145372 | 8.8.9.10 | 224.0.0.9 | RIPv2 | 86 | Response |
| 17 | 68.290369 | 8.8.9.1 | 224.0.0.5 | OSPF | 90 | Hello Packet |
| 18 | 75.565528 | 8.8.9.1 | 224.0.0.5 | OSPF | 90 | Hello Packet |

Εικόνα 10-30 OSPF hello μηνύματα από Router2(GNS3) & RIP αιτήσεις στο ethernet port 1 του δικτύου 8.8.9.0



10.5.2 Μελλοντική Έρευνα

Όπως βλέπουμε στην τοπολογία που υπάρχει στο GNS3 στον δρομολογητή R1 έχουμε συνδεθεί σε ένα μεταγωγέα που μας δίνει πρόσβαση στο Internet. Αυτό που θέλουμε να επιτύχουμε είναι να αποκρύψουμε ουσιαστικά όλο το δίκτυο που έχουμε δημιουργήσει στις δύο τοπολογίες αυτές πίσω από μία δημόσια IP διεύθυνση (NAT), πράγμα που έχουμε υλοποιήσει εν μέρει και σε άλλες τοπολογίες που μελετήσαμε. Το δίκτυο αυτό θα επικοινωνεί με κάποιο remote monitoring μηχανισμό έτσι ώστε να παρατηρούμε και να διαχειριζόμαστε τις τοπολογίες απομακρυσμένα και να δίνουμε εντολές ή να προγραμματίζουμε ενδεχόμενες λειτουργίες των συσκευών και να παρατηρούμε την απόδοσή τους. Τέλος αυτό που επιθυμούμε σαν μελλοντικό στόχο είναι να φέρουμε τις τοπολογίες μας ένα βήμα πιο κοντά στο IoT και στα έξυπνα περιβάλλοντα.



11 Βιβλιογραφία

1. [https://en.wikipedia.org/wiki/Point-to-point_\(telecommunications\)](https://en.wikipedia.org/wiki/Point-to-point_(telecommunications))
2. https://www.webopedia.com/quick_ref/topologies.asp
3. <http://www.atis.org/glossary/definition.aspx?id=4250>
4. <http://www.atis.org/glossary/definition.aspx?id=2879>
5. https://el.wikipedia.org/wiki/%CE%A4%CE%BF%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1_%CE%B4%CE%B9%CE%BA%CF%84%CF%8D%CE%BF%CF%85
6. <https://www.computerhope.com/jargon/b/bustopol.htm>
7. https://www.webopedia.com/TERM/R/ring_network.html
8. <https://www.techopedia.com/definition/17045/bus-topology>
9. <https://www.computerhope.com/jargon/s/startopo.htm>
10. <https://www.computerhope.com/jargon/t/treetopo.htm>
11. <https://techspirited.com/hybrid-topology>
12. Δίκτυα υπολογιστών 5η Αμερικανική Έκδοση”, Tanenbaum, Wetherall
13. https://www.cs.jhu.edu/~cs647/intro_adhoc.pdf
14. https://el.wikipedia.org/wiki/%CE%94%CE%AF%CE%BA%CF%84%CF%85%CE%BF_%CE%B5%CF%85%CF%81%CE%B5%CE%AF%CE%B1%CF%82_%CF%80%CE%B5%CF%81%CE%B9%CE%BF%CF%87%CE%AE%CF%82
15. <https://www.techrepublic.com/article/understanding-the-differences-between-client-server-and-peer-to-peer-networks/>
16. <https://www.bbc.com/bitesize/guides/zh4whyc/revision/1>
17. An Integrated NAN Architecture for Smart Energy Grid
18. [https://en.wikipedia.org/wiki/Router_\(computing\)](https://en.wikipedia.org/wiki/Router_(computing))
19. Δικτύωση Υπολογιστών 6η Έκδοση Προσέγγιση από πάνω προς τα κάτω



20. Τεχνολογία δικτύων επικοινωνιών, Εκδ. Ο.Ε.Δ.Β.
21. Werner Feibel, The Encyclopedia of Networking (2nd Edition), SYBEX 1996, ISBN 0-7821-1829-1
22. [https://en.wikipedia.org/wiki/Bridging_\(networking\)](https://en.wikipedia.org/wiki/Bridging_(networking))
23. https://en.wikipedia.org/wiki/Ethernet_hub
24. <https://www.techopedia.com/definition/13538/wireless-access-point-wap>
25. <https://www.digitalocean.com/community/tutorials/an-introduction-to-networking-terminology-interfaces-and-protocols>
26. CCIE Professional Development Routing TCP/IP, Volume I, Second Edition
27. Running IPv6, Ijitsch van Beijnum, εκδόσεις Apress.
28. Ασύρματα Δίκτυα Υπολογιστών Ασφάλεια και Απόδοση των Πρωτοκόλλων TCP/IP
29. <http://www.ipv6now.com.au/IPv6AddressBasics-IPv6Now.pdf>
30. [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc781068\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc781068(v=ws.10))
31. IPv4 Vs IPv6 QoS: A challenge in MANET
32. https://en.wikipedia.org/wiki/IPv6_address
33. Neighbor Discovery for IP version 6 (IPv6)
34. <https://tools.ietf.org/html/rfc4861>
35. Marin Dunmore: 6net An IPv6 Deployment Guide, The 6net Consortium, 2005
36. https://en.wikipedia.org/wiki/Network_address_translation
37. https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-nat-overview.html
38. https://en.wikipedia.org/wiki/Cisco_Discovery_Protocol
39. An Introduction to IGRP Cisco
40. https://www.cisco.com/c/en/us/td/docs/ios/12_2/ip/configuration/guide/fipr_c/1cfigrp.pdf
41. Enhanced Interior Gateway Routing Protocol Cisco



42. <https://study-ccna.com/eigrp-overview/>
43. Introduction to EIGRP Cisco
44. Reproducible Network Experiments Using Container-Based Emulation Nikhil Handigol , Brandon Heller , Vimalkumar Jeyakumar , Bob Lantz , Nick McKeown
45. Container-based Operating System Virtualization: A Scalable, High-performance Alternative to Hypervisors Stephen Soltesz, Herbert Pötz, Marc E. Fiuczynski, Andy Bavier, Larry Peterson
46. https://en.wikipedia.org/wiki/Internet_service_provider
47. Σχεδιασμός Εικονικών Δικτύων Ενότητα 5: Εικονικά Ιδιωτικά Δίκτυα Επιπέδου
48. Understanding Redistribution of OSPF Routes into BGP Cisco
49. Redistributing Routing Protocols Cisco
50. <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>
51. A Network in a Laptop: Rapid Prototyping for Software-Defined Networks Bob Lantz, Brandon Heller, Nick McKeown
52. CORE Documentation Release 4.8
53. <http://www.brianlinkletter.com/open-source-network-simulators/>
54. IMUNES Based Distributed Network Emulator Z. Puljiz and M. Mikuc Faculty of Electrical Engineering and Computing/Department of Telecommunications, Zagreb, Croatia
55. Aruba VAN SDN Controller 2.8 Administrator Guide
56. Internet-of-Things-Based Smart Environments: State of the Art, Taxonomy, and Open Research Challenges Ejaz Ahmed, Ibrar Yaqoob, Abdullah Gani, Muhammad Imran, and Mohsen Guizani
57. GNS3 Network Simulation Guide
58. Understanding VLAN Trunk Protocol (VTP) Cisco
59. Catalyst 3560 Software Configuration Guide, Release 12.2(52)SE
60. Understanding and Configuring Spanning Tree Protocol (STP) on Catalyst Switches
61. Spanning Tree Protocol Problems and Related Design Considerations