



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

## **Εικονικός μεταγωγέας - Virtual Switch Ανάλυση απαιτήσεων, σχεδιασμός και υλοποίηση**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
του  
ΜΠΡΕΓΙΑΝΝΗ ΝΙΚΟΛΑΟΥ

**Επιβλέπων:** Βασίλειος Μάγκλαρης  
Καθηγητής Ε.Μ.Π

Εργαστήριο Διαχείρισης & Βέλτιστου Σχεδιασμού Δικτύων  
Αθήνα, Ιούλιος 2012





# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ & ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ

## **Εικονικός μεταγωγέας - Virtual Switch Ανάλυση απαιτήσεων, σχεδιασμός και υλοποίηση**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
ΤΟΥ  
ΜΠΡΕΓΙΑΝΝΗ ΝΙΚΟΛΑΟΥ

**Επιβλέπων:** Βασίλειος Μάγκλαρης  
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή επιτροπή την 24<sup>η</sup> Ιουλίου, 2012

.....  
Βασίλειος Μάγκλαρης  
Καθηγητής Ε.Μ.Π

.....  
Συμεών Παπαβασιλείου  
Αν. Καθηγητής Ε.Μ.Π

.....  
Δημήτριος Καλογεράς  
Ερευνητής ΕΠΙΣΕΥ

.....  
Μπρέγιαννης Νικόλαος  
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

**Copyright © - All rights reserved** Μπρέγιαννης Νικόλαος, 2012

Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται στο συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

# Περίληψη

Σκοπός της διπλωματικής εργασίας ήταν η ανάλυση των απαιτήσεων και ο σχεδιασμός ενός εικονικού μεταγωγέα (layer-2 switch), που να έχει τη δυνατότητα να δημιουργεί τοπικά και εικονικά τοπικά δίκτυα, με κόμβους εικονικές μηχανές (virtual machines). Οι εικονικές μηχανές λειτουργούν σε πραγματικές υποδομές και χρησιμοποιήθηκαν για την πραγματοποίηση πειραμάτων σε πραγματικά δίκτυα.

Συγκεκριμένα έγινε χρήση του ανοιχτού λογισμικού Open vSwitch, το οποίο υλοποιεί λειτουργίες μεταγωγής σε εικονικοποιημένα περιβάλλοντα. Έγινε ο σχεδιασμός του δικτυακού συστήματος και στη συνέχεια ελέγχθηκε η απόδοση του εικονικού μεταγωγέα τόσο σε καταστάσεις απλής μεταγωγής πακέτων και δεδομένων όσο και σε περιπτώσεις streaming ήχου και βίντεο όπου η ποιότητα της παρεχόμενης υπηρεσίας είναι ιδιαίτερα σημαντική και απαιτητική. Τέλος για κάθε πείραμα που εκτελέστηκε, καταγράφηκαν σε γραφικές παραστάσεις τα αποτελέσματα της χρήσης του δικτύου.

Η μεθοδολογία αυτή μπορεί να γίνει οδηγός για την πραγματοποίηση διαφορετικών πειραμάτων αλλά και για τη χρήση του συγκεκριμένου ανοιχτού λογισμικού σε ιδιαίτερα απαιτητικές δικτυακές δομές καθώς έχουν καταγραφεί όλα τα βήματα για την εγκατάσταση και τη λειτουργία του.

## Λέξεις Κλειδιά

Εικονικοποίηση, επίπεδο ζεύξης δεδομένων, εικονικός μεταγωγέας, εικονικό δίκτυο, αναμετάδοση ήχου, αναμετάδοση βίντεο



# Abstract

The scope of this thesis was to analyze the requirements and the design of a virtual switch (layer-2 switch), which has the potential to create local and virtual local area networks (VLANs), using as nodes virtual machines. These virtual machines run in real infrastructure and can be used for experiments on real networks.

The open source software that was selected to implement the switching function in the visualized environments was Open vSwitch. First aim was the design and installation of the network system. Second aim was the evaluation of the performance of the virtual switch both in simple packet-switching and data situations and in cases of sound and video streaming, where quality of service is of great importance. Finally, for each experiment performed, the results of the Network traffic were recorded and displayed in graphs.

This methodology can be a guide for conducting new experiments as much as using the selected software in highly demanding network virtualized structures due to the fact that every step for installation and operation are recorded analytically.

## Keywords

Virtualization, data link layer, virtual switch, virtual network, sound streaming, video streaming

# Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Βασίλειο Μάγκλαρη και τον κ. Λεωνίδα Λυμπερόπουλο για την καθοδήγησή και την βοήθεια τους σε κάθε φάση της δημιουργίας και εκπόνησης της παρούσας διπλωματικής εργασίας. Επίσης θα ήθελα να ευχαριστήσω τους κ. Χρήστο Αργυρόπουλο, κ. Δημήτρη Καλογερά και κ. Αλέξανδρο Σιούγγαρη, οι οποίοι με τις γνώσεις τους και τη συνεισφορά τους βοήθησαν στην επιτυχή ολοκλήρωση της εργασίας. Παράλληλα, θα ήθελα να ευχαριστήσω όλα τα μέλη του Εργαστηρίου Διαχείρισης και Βέλτιστου Σχεδιασμού Δικτύων για τη διάθεση τους να βοηθήσουν σε οποιοδήποτε πρόβλημα ή απορία εμφανιζόταν.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, η οποία με στήριξε και με στηρίζει σε κάθε βήμα μου, καθώς και τους φίλους μου για την υπομονή, την κατανόηση και την υποστήριξή τους όλα αυτά τα χρόνια.

Μπρέγιαννης Νικόλαος



# Περιεχόμενα

Περίληψη .....	5
Abstract.....	7
Ευχαριστίες .....	8
Περιεχόμενα.....	9
Κατάλογος Εικόνων .....	13
<b>ΚΕΦΑΛΑΙΟ 1 .....</b>	<b>14</b>
<b>ΠΡΟΛΟΓΟΣ .....</b>	<b>14</b>
<b>ΚΕΦΑΛΑΙΟ 2 .....</b>	<b>15</b>
<b>ΘΕΩΡΗΤΙΚΗ ΕΙΣΑΓΩΓΗ .....</b>	<b>15</b>
2.1 Το Δικτυακό Πρότυπο OSI.....	15
2.2 Δικτύωση στο Layer 2 .....	17
2.2.1 Γέφυρες και Μεταγωγείς Δικτύων.....	17
2.2.2 L2 Virtual Private Networks .....	18
2.2.2.1 OSI Layer 2 VPN Services.....	18
2.2.2.2 Tunelling with Layer 2 payload .....	19
2.3 Overlay Transport Virtualization (OVT).....	20
2.3.1 Nexus 7000 Switch .....	21
2.4 ΕΙΚΟΝΙΚΟΠΟΙΗΣΗ (VIRTUALIZATION) .....	22
2.4.1 Εικονικοποίηση Υλικού (Hardware Virtualization) .....	23
2.4.1.1 Πλήρης Εικονικοποίηση (Full virtualization).....	23
2.4.1.2 Μερική εικονικοποίηση (Partial virtualization) .....	24
2.4.1.3 Παραεικονικοποίηση (Paravirtualization) .....	24
2.4.1.4 Εικονικοποίηση σε επίπεδο λειτουργικού συστήματος (Operating system-level virtualization) .....	25
2.4.2 Εικονικοποίηση δικτύων (Network Virtualization) .....	26
2.4.2.1 Συστατικά Εικονικών Δικτύων .....	26
2.4.2.2 Τεχνικές Εικονικοποίησης Δικτύων .....	28
<b>ΚΕΦΑΛΑΙΟ 3 .....</b>	<b>29</b>

<b>ΠΛΑΤΦΟΡΜΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ .....</b>	<b>29</b>
3.1 Hardware .....	29
3.2 Software .....	30
3.2.1 Linux Kernel.....	30
3.2.2 Ubuntu/Linux Distribution .....	31
3.2.3 Oracle VM VirtualBox.....	32
3.2.3.1 Επισκόπηση του VirtualBox .....	32
3.2.3.2 Χαρακτηριστικά του VirtualBox .....	33
3.2.3.3 Virtual Networking στο VirtualBox .....	33
3.2.3.4 Συνοπτική παρουσίαση του Bridged Networking .....	34
3.2.4 VLC Player .....	35
3.2.3.1 Επισκόπηση του VLC Player .....	35
3.2.3.2 Χαρακτηριστικά του VLC Player.....	35
3.2.4 sFlow Trend.....	36
3.2.4.1 Επισκόπηση του sFlow Trend .....	36
3.2.4.2 Αρχιτεκτονική sFlow Trend .....	37
3.2.4 Εικονικός Μεταγωγές - Open vSwitch .....	39
3.2.4.1 Επισκόπηση του Open vSwitch.....	39
3.2.4.2 Αρχιτεκτονική του Open vSwitch.....	41
<b>ΚΕΦΑΛΑΙΟ 4 .....</b>	<b>43</b>
<b>ΑΝΑΛΥΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....</b>	<b>43</b>
4.1 Επισκόπηση του συστήματος .....	43
4.2 Στάδιο 1 <sup>ο</sup> : Εγκατάσταση λειτουργικού συστήματος και απαραίτητων λογισμικών. ....	44
4.3 Στάδιο 2 <sup>ο</sup> : Εγκατάσταση του εικονικού μεταγωγέα (Open vSwitch).....	44
4.4 Στάδιο 3 <sup>ο</sup> : Σχεδιασμός και Λειτουργικότητα Τοπολογίας Δικτύου .....	46
4.5 Στάδιο 4 <sup>ο</sup> : Απόδοση συστήματος σε streaming ήχου και εικόνας .....	48
4.6 Στάδιο 5 <sup>ο</sup> : Απεικόνιση μετρήσεων με το sFlow Trend .....	48
<b>ΚΕΦΑΛΑΙΟ 5 .....</b>	<b>49</b>
<b>ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....</b>	<b>49</b>
5.1 Υλοποίηση πρώτου σταδίου. Εγκατάσταση λειτουργικού συστήματος και λογισμικών που θα χρησιμοποιηθούν .....	49
5.2 Υλοποίηση δεύτερου σταδίου. Εγκατάσταση εικονικού μεταγωγέα στη θέση του linux bridge. ....	50

5.3 Υλοποίηση τρίτου σταδίου. Εγκατάσταση τοπολογίας δικτύου και έλεγχος λειτουργικότητας .....	52
5.3.1 Εγκαθίδρυση συνδέσεων μεταξύ των μερών του συστήματος .....	53
5.3.1.1 Ενσωμάτωση επιπλέον κίνησης πακέτων στην απλή τοπολογία δικτύου και τροποποίηση της ποιότητας της υπηρεσίας (QoS) .....	57
5.3.2 Δημιουργία τοπολογίας δικτύου με χρήση VLANs.....	60
5.3.2.1 Ενσωμάτωση επιπλέον κίνησης πακέτων στην τοπολογία δικτύου με VLAN και τροποποίηση της ποιότητας της υπηρεσίας (QoS) .....	61
5.4 Υλοποίηση 4ου σταδίου: Έλεγχος απόδοσης συστήματος σε streaming ήχου και βίντεο με χρήση VLC .....	63
5.4.1 Έλεγχος απόδοσης συστήματος σε streaming ήχου στην απλή τοπολογία.....	63
5.4.2 Έλεγχος απόδοσης συστήματος σε streaming ήχου στην απλή τοπολογία μαζί με δικτυακή συμφόρηση.....	65
5.4.3 Έλεγχος απόδοσης συστήματος σε streaming ήχου στην τοπολογία με χρήση VLAN .....	66
5.4.4 Έλεγχος απόδοσης συστήματος σε streaming ήχου στην τοπολογία με VLANs μαζί με δικτυακή συμφόρηση.....	69
5.4.5 Έλεγχος απόδοσης συστήματος σε streaming βίντεο στην απλή τοπολογία .....	70
5.4.6 Έλεγχος απόδοσης συστήματος σε streaming βίντεο στην απλή τοπολογία με δικτυακή συμφόρηση.....	72
5.4.7 Έλεγχος απόδοσης συστήματος σε streaming βίντεο στην τοπολογία με χρήση VLAN .....	73
5.4.8 Έλεγχος απόδοσης συστήματος σε streaming ήχου στην τοπολογία με VLANs μαζί με δικτυακή συμφόρηση.....	76
5.9 Υλοποίηση 5ου σταδίου: απεικόνιση των λειτουργιών και των επιμέρους μετρήσεων με το λογισμικό sFlow Trend.....	77
5.9.1 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου.....	79
5.9.1.1 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε απλή τοπολογία χωρίς δικτυακή συμφόρηση.....	79
5.9.1.2 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε απλή τοπολογία με δικτυακή συμφόρηση.....	81
5.9.1.3 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε απλή τοπολογία με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείου.....	83
5.9.1.4 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε τοπολογία VLAN χωρίς δικτυακή συμφόρηση.....	85
5.9.1.5 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε τοπολογία VLAN με δικτυακή συμφόρηση.....	87

5.9.1.6 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε τοπολογία VLAN με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείων. ....	89
5.5.10 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο. ....	91
5.5.10.1 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο στην απλή τοπολογία χωρίς δικτυακή συμφόρηση.....	91
5.5.10.2 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε απλή τοπολογία με δικτυακή συμφόρηση.....	93
5.5.10.3 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο σε απλή τοπολογία με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείου.....	95
5.5.10.4 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο σε τοπολογία VLAN χωρίς δικτυακή συμφόρηση. ....	97
5.5.10.5 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο σε τοπολογία VLAN με δικτυακή συμφόρηση.....	99
5.5.10.6 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο σε τοπολογία VLAN με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείων. ....	101
<b>ΚΕΦΑΛΑΙΟ 6</b> .....	<b>104</b>
<b>ΕΠΙΛΟΓΟΣ</b> .....	<b>104</b>
6.1 Συμπεράσματα.....	104
6.2 Προτάσεις για μελλοντικές επεκτάσεις και βελτιώσεις .....	104
<b>ΠΑΡΑΡΤΗΜΑ Α</b> .....	<b>106</b>
<b>ΟΔΗΓΟΣ ΕΓΚΑΤΑΣΤΑΣΗΣ OPEN VSWITCH ΜΕΣΩ SCRIPTS</b> .....	<b>106</b>
A.1 Βασικά χαρακτηριστικά Open vSwitch.....	106
A.2 Script για την αυτοματοποιημένη δημιουργία απλής τοπολογίας και VLAN τοπολογία.....	106
<b>ΠΑΡΑΡΤΗΜΑ Β'</b> .....	<b>109</b>
<b>ΟΔΗΓΟΣ ΔΗΜΙΟΥΡΓΙΑΣ ΣΥΜΦΟΡΗΣΗ ΚΑΙ OPENFLOW ΡΟΗΣ</b> .....	<b>109</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b> .....	<b>110</b>

## Κατάλογος Εικόνων

Εικόνα 1: Δομή του πρωτοκόλλου OSI .....	15
Εικόνα 2: Επικοινωνία Εικονικών Δικτύων μέσω του OVT .....	20
Εικόνα 3: Δομή πλήρους εικονικοποίησης .....	23
Εικόνα 4: Δομή παραεικονικοποίησης .....	24
Εικόνα 5: Δομή εικονικοποίησης σε επίπεδο λειτουργικού συστήματος .....	25
Εικόνα 6: Χάρτης πυρήνα Linux (Linux Kernel Map).....	30
Εικόνα 7: Τυπικό περιβάλλον Ubuntu 10.04 LTS.....	31
Εικόνα 8: Γραφικό περιβάλλον VirtualBox .....	32
Εικόνα 9: Γραφικό περιβάλλον VLC Player .....	35
Εικόνα 10: Γραφικό περιβάλλον sFlow Trend .....	36
Εικόνα 11: Αρχιτεκτονική sFlow Trend .....	38
Εικόνα 12: Τρόπος λειτουργίας sFlow Trend.....	38
Εικόνα 13: Open vSwitch .....	39
Εικόνα 14: Δομή Open vSwitch.....	39
Εικόνα 15: Απλή τοπολογία δίκτυο με τέσσερα εικονικά μηχανήματα και τον host συνδεδεμένα στον εικονικό μεταγωγέα br0 .....	46
Εικόνα 16: Τοπολογία VLAN με συμφόρηση .....	47
Εικόνα 17: Παραδείγματα απεικόνισης γραφημάτων στο sFlow Trend .....	48
Εικόνα 18: Στιγμιότυπο εγκατάστασης Ubuntu.....	49
Εικόνα 19: Ορισμός τύπου των διαμορφωμένων τομέων .....	50
Εικόνα 20: Στιγμιότυπο του VirtualBox μετά την εγκατάσταση των εικονικών μηχανών .....	53
Εικόνα 21: Clone machine και αλλαγή MAC address .....	55
Εικόνα 22: Streaming ήχου μέσω VLC interface – απλή τοπολογία.....	64
Εικόνα 23: Εντολή που εκτελείται από το VLC – απλή τοπολογία .....	64
Εικόνα 24: Αναπαραγωγή του streaming ήχου από VLC interface – απλή τοπολογία .....	65
Εικόνα 25: Streaming ήχου μέσω VLC interface – VLAN .....	67
Εικόνα 26: Εντολή που εκτελείται από το VLC – VLAN.....	67
Εικόνα 27: Εντολή που εκτελείται από VLC - VLAN .....	68
Εικόνα 28: Streaming ήχου μέσω VLC interface - VLAN .....	68
Εικόνα 29: Streaming βίντεο μέσω VLC interface - απλή τοπολογία .....	71
Εικόνα 30: Εντολή που εκτελείται από VLC - απλή τοπολογία .....	71
Εικόνα 31: Εντολή δημιουργίας streaming - VLAN.....	74
Εικόνα 32: Αναπαραγωγή μέσω VLC interface - VLAN .....	74
Εικόνα 33: Εντολή δημιουργίας streaming - VLAN.....	75
Εικόνα 34: Αναπαραγωγή μέσω VLC interface - VLAN .....	75

# ΚΕΦΑΛΑΙΟ 1

## ΠΡΟΛΟΓΟΣ

Τα δίκτυα, είτε αυτά συνδέονται διαδικτυακά, είτε τοπικά, αρχίζουν να εμφανίζουν περιορισμούς λόγω των όλο και μεγαλύτερων αναγκών σε πόρους που απαιτούνται από τους χρήστες. Η δομή των σύγχρονων δικτύων βασίζεται πλέον σε μεγάλο βαθμό σε τεχνολογίες εικονικοποίησης, οι οποίες έχουν αποκτήσει ένα μεγάλο μερίδιο στην εγκατάσταση νέων τοπολογιών. Ο λόγος είναι ότι ένα και μόνο φυσικό μηχάνημα, μπορεί πλέον να φιλοξενήσει ένα πολύ μεγάλο αριθμό εικονικών μηχανημάτων που λειτουργούν μέσα σε αυτό. Σε τοπικό επίπεδο τα δίκτυα αυτά μπορούν να ομαδοποιηθούν σε μία κατηγορία η οποία ονομάζεται 'εικονικά δίκτυα', ενώ σε επίπεδο διαδικτύου αναφερόμαστε σε αυτά ως 'νέφη υπολογιστών' (cloud networking).

Σημαντικό ρόλο στη διαχείριση αυτών των δικτύων, παίζει ο μεταγωγέας και συγκεκριμένα όταν αναφερόμαστε σε εικονικοποιημένα περιβάλλοντα ο εικονικός μεταγωγέας (virtual switch). Όπως ακριβώς στα φυσικά δίκτυα, έτσι και στα εικονικά, απαιτείται η σωστή διαχείριση και προώθηση των πακέτων μέσα σε αυτά με βέλτιστο τρόπο και με όσο το δυνατόν μικρότερες απώλειες. Επιπλέον με τη χρήση εικονικού μεταγωγέα, επιτρέπονται δυνατότητες L2 learning table, L3 προωθήσεις, πολιτικές τροποποίησης της ποιότητας της υπηρεσίας (QoS) καθώς και διαδικασίες παρακολούθησης της δικτυακής σύνθεσης μέσω τρίτων προγραμμάτων που συνδέονται σε αυτόν.

Για να μπορέσει να λειτουργήσει ο εικονικός μεταγωγέας σε πραγματικά δίκτυα απαιτείται η γνώση και η ανάλυση των εργαλείων που τον διαχειρίζονται. Παράλληλα με τη δημιουργία ενός περιβάλλοντος στο οποίο υπάρχει η δυνατότητα πειραμάτων, δύναται να ελεγχθεί η λειτουργικότητα και η απόδοση του σε καταστάσεις που ανταποκρίνονται σε καθημερινές, απαιτητικές ή μη απαιτητικές δομές δικτύων. Το ενδιαφέρον στην παρούσα διπλωματική εστιάζεται κυρίως σε δυο διαφορετικές τοπολογίες, στην μεταξύ τους συνδεσιμότητα αλλά και στην απόδοση λειτουργιών αναπαραγωγής πολυμέσων μέσα στο δίκτυο.

## ΚΕΦΑΛΑΙΟ 2

### ΘΕΩΡΗΤΙΚΗ ΕΙΣΑΓΩΓΗ

#### 2.1 Το Δικτυακό Πρότυπο OSI (1)

Το δικτυακό πρότυπο OSI (εικόνα 1), αναπτύχθηκε ως ένα μοντέλο για το χαρακτηρισμό και την τυποποίηση των λειτουργιών συστημάτων επικοινωνιών, σε όρους επιπέδων (layers). Σύμφωνα με αυτό, παρόμοιες λειτουργίες, ομαδοποιούνται σε λογικά επίπεδα. Ένα επίπεδο εξυπηρετεί τόσο το επίπεδο που βρίσκεται πάνω από αυτό, όσο και το επίπεδο που βρίσκεται κάτω από αυτό. Για παράδειγμα, ένα επίπεδο που προσφέρει επικοινωνία σε ένα δίκτυο χωρίς λάθη, παρέχει τη διαδρομή που χρειάζεται το ακριβώς ανώτερο επίπεδο, ενώ καλεί το χαμηλότερο επίπεδο, το οποίο είναι υπεύθυνο για τη μετάδοση των πακέτων στο δίκτυο. Τα 2 διαφορετικά επίπεδα συνδέονται με το μεσαίο με μία οριζόντια σύνδεση.

Το πρωτόκολλο OSI αποτελείται από 7 επίπεδα (layers) τα οποία αναλύονται ακολούθως:

OSI Model			
	Data unit	Layer	Function
Host layers	Data	7. Application	Network process to application
		6. Presentation	Data representation, encryption and decryption, convert machine dependent data to machine independent data
		5. Session	Interhost communication, managing sessions between applications
	Segments	4. Transport	End-to-end connections, reliability and flow control
Media layers	Packet/Datagram	3. Network	Path determination and logical addressing
	Frame	2. Data link	Physical addressing
	Bit	1. Physical	Media, signal and binary transmission

Εικόνα 1: Δομή του πρωτοκόλλου OSI

- **Layer 1:** Το φυσικό επίπεδο ασχολείται με τη μετάδοση ανεπεξέργαστων δυαδικών ψηφίων μέσω ενός καναλιού επικοινωνίας. Η σχεδίασή του βασίζεται στην εξασφάλιση του ότι όταν μία πλευρά στέλνει το bit 0 θα πρέπει να λαμβάνεται από την άλλη πλευρά ως bit 0 και όχι ως bit 1. Καθορίζει τη σχέση σύνδεσης μεταξύ της συσκευής που θέλει να μεταδώσει την πληροφορία και του τρόπου με τον οποίο θα γίνει η μετάδοση. Δηλαδή αν θα γίνει η μετάδοση μέσω χαλκού, οπτικού καλωδίου, αέρα κτλ.
- **Layer 2:** Το επίπεδο συνδέσμου ζεύξης δεδομένων παρέχει τα λειτουργικά και διαδικαστικά μέσα για τη μεταφορά δεδομένων από μια συσκευή ενός τοπικού δικτύου σε άλλη, αλλά και για την ανίχνευση και διόρθωση σφαλμάτων που συμβαίνουν στο φυσικό επίπεδο. Οι μη

ιεραρχημένες διευθύνσεις των συσκευών εδώ είναι οι φυσικές (π.χ. MAC διευθύνσεις), δηλαδή είναι προκαθορισμένες και αποθηκευμένες στις κάρτες δικτύου των επικοινωνούντων κόμβων από το εργοστάσιο. ο πιο γνωστό πρότυπο αυτού του επιπέδου είναι το Ethernet, για τοπικά δίκτυα. Άλλα παραδείγματα πρωτοκόλλων ζεύξης δεδομένων αποτελούν τα HDLC και ADCCP, για συνδέσεις από-σημείο-σε-σημείο (αγγλ. point-to-point) και το 802.11, για ασύρματα τοπικά δίκτυα.

- **Layer 3:** Το επίπεδο δικτύου παρέχει τα λειτουργικά και διαδικαστικά μέσα για τη μεταφορά στοιχειωσειρών δεδομένων μεταβλητού μήκους από μια προέλευση σε έναν προορισμό, μέσα από ένα ή περισσότερα ενδιάμεσα δίκτυα, ενώ διατηρεί την ποιότητα εξυπηρέτησης που απαιτεί το επίπεδο μεταφοράς. Το επίπεδο δικτύου εκτελεί λειτουργίες δρομολόγησης, με πιθανές κατατμήσεις / αποτμηματοποιήσεις, και αναφέρει σφάλματα σχετικά με την παράδοση των πακέτων. Οι δρομολογητές (routers) λειτουργούν στο επίπεδο αυτό διακινώντας δεδομένα σε διασυνδεδεμένα δίκτυα. Υπάρχουν και δικτυακοί διακόπτες (switch) που σχετίζονται με τις διευθύνσεις (IP). Το πλέον αναγνωρίσιμο παράδειγμα πρωτοκόλλου δικτύου είναι το Πρωτόκολλο Διαδικτύου (Internet Protocol, IP).
- **Layer 4:** Το επίπεδο μεταφοράς διεκπεραιώνει τη μεταφορά των δεδομένων από χρήστη σε χρήστη, απαλλάσσοντας έτσι τα ανώτερα επίπεδα από κάθε φροντίδα να προσφέρουν αξιόπιστη μεταφορά δεδομένων από το ένα άκρο της επικοινωνίας στο άλλο. Το επίπεδο μεταφοράς ελέγχει την αξιοπιστία ενός χρησιμοποιούμενου καναλιού με έλεγχο ροής (αγγλ. flow control), κατάτμηση και αποτμηματοποίηση (segmentation / desegmentation), καθώς και έλεγχο σφαλμάτων (error control). Ορισμένα πρωτόκολλα καταγράφουν καταστάσεις και συνδέσεις, οπότε κρατούν λογαριασμό των πακέτων και επανεκπέμπουν αυτά που δεν παρελήφθησαν σωστά. Τα διάφορα πρωτόκολλα μορφοποιούν διαφορετικά τα εκπεμπόμενα πακέτα πληροφοριών, αλλά τα προς αποστολή δεδομένα παραλαμβάνονται αρχικά από τα ανώτερα επίπεδα. Το συνηθέστερο παράδειγμα πρωτοκόλλου μεταφοράς είναι το TCP (2) (Transmission Control Protocol, πρωτόκολλο ελέγχου μετάδοσης). Άλλα πρωτόκολλα μεταφοράς είναι τα UDP (3) (User Datagram Protocol, πρωτόκολλο για ασυνδεσμική αποστολή δεδομένων, SCTP (Stream Control Transmission Protocol, πρωτόκολλο ελέγχου της ροής μετάδοσης).
- **Layer 5:** Το επίπεδο συνόδου ελέγχει τις συνόδους (δηλαδή τις ανταλλαγές δεδομένων) μεταξύ δύο υπολογιστών, του A και του B. Ξεκινά, διαχειρίζεται και τερματίζει τη σύνδεση μεταξύ μιας τοπικής και μιας απομακρυσμένης εφαρμογής. Αντιμετωπίζει λειτουργίες FDX (*full duplex, οι A και B μιλούν ταυτόχρονα από δύο κανάλια*) ή HDX (*half-duplex, μιλάει ο A και μετά απαντάει ο B από το ένα διαθέσιμο κανάλι*), ενώ υποστηρίζει διαδικασίες αποθήκευσης κατάστασης (checkpoint), αναβολής (adjournment), τερματισμού (termination) και επανεκκίνησης (restart). Αυτό το επίπεδο είναι υπεύθυνο για το ομαλό κλείσιμο της συνόδου (που είναι ιδιότητα του TCP) και επίσης για την *αποθήκευση και ανάκτηση κατάστασης*, λειτουργίες οι οποίες δεν χρησιμοποιούνται στην στοίβα πρωτοκόλλων του Διαδικτύου.
- **Layer 6:** Το επίπεδο παρουσίασης μετασχηματίζει τα δεδομένα σε τυπική μορφή που την αναμένει το επίπεδο εφαρμογών. Στο επίπεδο αυτό τα δεδομένα υφίστανται κρυπτογράφηση, συμπίεση, κωδικοποίηση MIME και όποια άλλη διαμόρφωση απαιτεί η μορφή δεδομένων ή ο



σχεδιαστής του πρωτοκόλλου. Παραδείγματα αποτελούν η μετατροπή αρχείων από κώδικα EBCDIC σε κώδικα ASCII και η μετατροπή της δομής των δεδομένων σε μορφή XML ή αντίστροφα (π.χ. από XML σε έγγραφο τύπου DOC).

- **Layer 7: Το επίπεδο εφαρμογών** παρέχει στον χρήστη έναν τρόπο να προσπελάσει μέσω μιας εφαρμογής τις πληροφορίες ενός δικτύου. Αυτό το επίπεδο είναι η κύρια διασύνδεση του χρήστη με την εφαρμογή και, συνεπώς, με το δίκτυο. Στο επίπεδο αυτό γίνεται η διαχείριση των καταναμημένων εφαρμογών, η αποστολή του ηλεκτρονικού ταχυδρομείου κλπ. Παραδείγματα πρωτοκόλλων επιπέδου εφαρμογών αποτελούν τα Telnet, FTP, SFTP, http.

## 2.2 Δικτύωση στο Layer 2

Το επίπεδο που παρουσιάζει το μεγαλύτερο ενδιαφέρον στα εικονικοποιημένα περιβάλλοντα είναι το δεύτερο. Στο επίπεδο 2 (Layer 2, L2) του δικτυακού προτύπου OSI, όπως αναφέρθηκε, βρίσκεται το επίπεδο Ζεύξης Δεδομένων (Data Link Layer). Εκεί βρίσκονται όλες οι λειτουργίες για την μεταφορά δεδομένων μεταξύ δικτυακών οντοτήτων, καθώς και η δυνατότητα αναγνώρισης και πιθανής διόρθωσης σφαλμάτων του Φυσικού επιπέδου (Physical Layer, L1). Παραδείγματα L2 πρωτοκόλλων είναι το Ethernet που αφορούν κυρίως στα τοπικά δίκτυα, το πρωτόκολλο Point-to-Point (PPP, RFC 1661) κ.α. Το επίπεδο Data Link έχει σχέση με την προώθηση και παράδοση πλαισίων (Data Link frames) σε τοπικό περιβάλλον, μεταξύ συσκευών που ανήκουν στο ίδιο τοπικό δίκτυο. Τα πλαίσια αυτά δεν εξέρχονται από τα όρια του τοπικού δικτύου. Τόσο η δρομολόγηση στο διαδίκτυο, όσο και η καθολική διευθυνσιοδότηση είναι λειτουργίες υψηλότερου επιπέδου. Τα L2 πρωτόκολλα ασχολούνται αποκλειστικά με το τοπικό επίπεδο μεταφοράς, διευθυνσιοδότησης και ελέγχου όσων βρίσκονται συνδεδεμένοι σε αυτό.

### 2.2.1 Γέφυρες και Μεταγωγείς Δικτύων

Τα κύρια μέρη ενός δικτύου επιπέδου 2, είναι οι γέφυρες (bridges) και οι μεταγωγείς (switches). Αναλύοντας τον όρο γεφύρωση (bridging), αποτελεί μία τεχνική προώθησης σε δίκτυα μεταγωγής πακέτων. Σε αντίθεση με τη δρομολόγηση (routing), η γεφύρωση δεν κάνει υποθέσεις σχετικά με το που βρίσκεται μία συγκεκριμένη διεύθυνση στο δίκτυο. Αντίθετα, βασίζεται στην τεχνική της πλημμύρας (flooding) και στον έλεγχο της διεύθυνσης προέλευσης στην επικεφαλίδα των πλαισίων που έχουν ληφθεί με σκοπό τον εντοπισμό άγνωστων συσκευών. Λόγω αυτής της εξάρτησης από την τεχνική της πλημμύρας, η γεφύρωση χρησιμοποιείται κυρίως στα τοπικά δίκτυα.

Δικτυακός μεταγωγέας (network switch) είναι μία συσκευή δικτύου η οποία λειτουργεί ως διανομέας, δηλαδή επαναλαμβάνει το σήμα που λαμβάνει, με την εξαίρεση ότι εγγράφει τις διευθύνσεις MAC των υπολογιστών που είναι συνδεδεμένοι σε αυτόν σε ένα πίνακα (MAC address table). Όταν λαμβάνει κάποιο μήνυμα, το στέλνει στον παραλήπτη για τον οποίον προορίζεται. Οι μεταγωγείς δικτύου περιορίζουν την άχρηστη δικτυακή κυκλοφορία και συνδράμουν στη δημιουργία ενός δικτύου υψηλής απόδοσης. Αυτό συμβαίνει διότι δε γίνονται τροποποιήσεις στα πακέτα δεδομένων αλλά μόνο στα πλαίσια που τα ενθυλακώνει και μόνο στην περίπτωση διάδοσης σε ανόμοια μέσα. Μπορούν να λειτουργούν σε ένα ή περισσότερα επίπεδα του μοντέλου OSI (multilayer switch).

Οι γέφυρες είναι ηλεκτρονικές συσκευές που υλοποιούν τη διασύνδεση - επικοινωνία μεταξύ τοπικών δικτύων υπολογιστών στο επίπεδο σύνδεσης (data link layer) του μοντέλου OSI. Οι γέφυρες κάνουν χρήση των διευθύνσεων υλικού (MAC address) των σταθμών εργασίας του τοπικού δικτύου, για να μεταδώσουν τα πλαίσια δεδομένων (data frames) μεταξύ των δικτύων που συνδέουν. Αρχικά συνέδεαν μόνο ομοειδή δίκτυα, ενώ αργότερα εμφανίστηκαν γέφυρες με δυνατότητα σύνδεσης και μεταξύ ετερογενών δικτύων. Οι σημερινές γέφυρες έχουν επιπλέον χαρακτηριστικά, όπως δυνατότητα φιλτραρίσματος και υψηλό ρυθμό μετάδοσης δεδομένων μεταξύ των δικτύων που συνδέουν. Με το πέρασμα των χρόνων η δυνατότητα γεφύρωσης ενσωματώθηκε στους δρομολογητές (routers).

### 2.2.2 L2 Virtual Private Networks (4)

Ένα εικονικό ιδιωτικό δίκτυο (VPN) είναι ένα δίκτυο που χρησιμοποιεί κατά κύριο λόγο μη ασφαλή επικοινωνία, όπως το Διαδίκτυο, και δίνει τη δυνατότητα σύνδεσης σε απομακρυσμένες ιδιωτικές δικτυακές τοποθεσίες (LAN). Ένα VPN συνήθως απαιτεί από τους απομακρυσμένους χρήστες του δικτύου πιστοποίηση, και συχνά ασφαλίσει τα δεδομένα με τεχνολογίες κρυπτογράφησης για να αποτραπεί η διάδοση των ιδιωτικών πληροφοριών σε μη εξουσιοδοτημένους χρήστες. Για να επιτευχθεί το παραπάνω, το πρωτόκολλο VPN ενθυλακώνει τα πακέτα χρησιμοποιώντας μια ασφαλή κρυπτογράφηση μεταξύ των συσκευών που δεν είναι μέλη του ίδιου ιδιωτικού δικτύου. Ένα VPN μπορεί να γίνεται για τη λειτουργικότητα του δικτύου που βρίσκεται σε οποιοδήποτε δίκτυο, όπως η κοινή χρήση των δεδομένων και η πρόσβαση σε πόρους δικτύου, εκτυπωτές, βάσεις δεδομένων, ιστοσελίδες, κλπ. Η τεχνολογία VPN μειώνει το κόστος, επειδή δεν χρειάζεται φυσική μισθωμένη γραμμή για τη σύνδεση απομακρυσμένων χρηστών σε ένα intranet.

Τα συστήματα VPN μπορούν να ταξινομηθούν από:

- Τα πρωτόκολλα που χρησιμοποιούνται για την σήραγγα της κυκλοφορίας.
- Το τερματικό σημείο της σήραγγας, δηλαδή, την άκρη πελάτη ή την άκρη του δικτύου παροχής.
- Το εάν προσφέρουν από σελίδα-σε-σελίδα ή απομακρυσμένη σύνδεση πρόσβασης.
- Τα επίπεδα ασφαλείας που παρέχονται.
- Το στρώμα του OSI που παρουσιάζουν για τη σύνδεση του δικτύου, όπως κυκλώματα επιπέδου 2 ή επιπέδου 3 σύνδεσης με το δίκτυο.

Στην παρούσα διπλωματική εργασία θα ασχοληθούμε και θα αναλύσουμε τις υπηρεσίες VPN που λειτουργούν στο επίπεδο 2.

#### 2.2.2.1 OSI Layer 2 VPN Services

Τα βασικά Layer 2 VPN services του μοντέλου OSI είναι:

- Virtual LAN: Τα Virtual LAN είναι μία L2 τεχνική που επιτρέπει την συνύπαρξη πολλαπλών LAN broadcast domains, διασυνδεδεμένα με ένα σύνολο από multipoint links (trunks), κάνοντας χρήση του IEEE 802.1Q trunking πρωτοκόλλου.

- Virtual private LAN service (VPLS): Το VPLS είναι ένα L2 Point-to-Point Virtual Private Network (PPVPN) προσομοιώνοντας όλες τις λειτουργίες ενός LAN. Από την προοπτική του χρήστη, ένα VPLS παρέχει την δυνατότητα της διασύνδεσης κάποιων τμημάτων LAN (LAN segments) πάνω από μία υποδομή δικτύου μεταγωγής πακέτων. Η υποδομή αυτή είναι αόρατη στον χρήστη με αποτέλεσμα τα απομακρυσμένα LAN segments να συμπεριφέρονται σαν ένα LAN. Στο VPLS, ο παροχέας της υποδομής προσομοιώνει επιπλέον και ένα bridge που υποστηρίζει learning και, προαιρετικά, VLAN.
- Pseudo wire (PW): Το PW είναι παρόμοια υπηρεσία με το VPWS παροχή point-to-point προσομοίωση L1 κυκλωμάτων χωρίς data link δομή, αλλά μπορεί επιπλέον να παρέχει διαφορετικά L2 πρωτόκολλα στα δύο άκρα. Συνήθως, το interface του είναι ένα πρωτόκολλο WAN, όπως τα Asynchronous Transfer Mode (ATM) και Frame Relay. Όταν έχουμε σαν σκοπό την παρουσίαση ενός συνεχούς (contiguous) LAN μεταξύ δύο ή περισσότερων τοποθεσιών είναι πιο κατάλληλα τα VPLS και IPLS.
- IP-only LAN-like service (IPLS): Το IPLS είναι ένα υποσύνολο του VPLS με δυνατότητες L3. Χρησιμοποιεί πακέτα και όχι πλαίσια.

### 2.2.2.2 Tunelling with Layer 2 payload

Η υλοποίηση των VPN services βασίζονται σε υποδομές που υλοποιούνται από tunneling πρωτόκολλα. Όταν αναφερόμαστε σε πρωτόκολλο tunneling στα δίκτυα υπολογιστών, εννοούμε ένα δικτυακό πρωτόκολλο (το delivery protocol) μέσα στο οποίο ενθυλακώνεται ένα άλλο πρωτόκολλο (το payload protocol). Με την χρήση του tunneling μπορούμε για παράδειγμα να μεταφέρουμε ένα πακέτο πρωτοκόλλου μη συμβατού με το δίκτυο παράδοσης, είτε να δημιουργήσουμε ένα ασφαλές κανάλι μέσα σε ένα μη αξιόπιστο δίκτυο. Ιδιαίτερο ενδιαφέρον, παρουσιάζουν οι τρόποι δημιουργίας VPLS που παρέχει υπηρεσία Ethernet, λόγω της ευρείας χρήσης του Ethernet στα τοπικά δίκτυα. Το VPLS μπορεί να προκύψει με την χρήση tunneling πρωτοκόλλων που δέχονται για payload protocol το Ethernet. Ακολουθούν οι βασικότερες τεχνολογίες ενθυλάκωσης πλαισίων Ethernet.

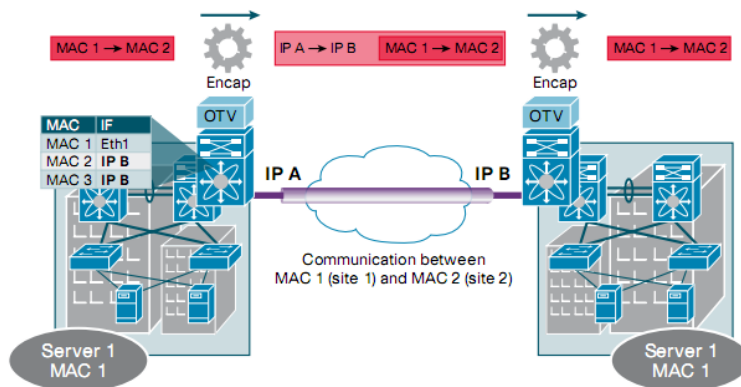
- Layer 2 Tunneling Protocol (L2TP): Το L2TP είναι ένα tunneling πρωτόκολλο, αναπτυγμένο από την Cisco Systems<sup>1</sup>, που χρησιμοποιείται για την δημιουργία VPNs. Δεν παρέχει κρυπτογράφηση ή εμπιστευτικότητα από μόνο του. Βασίζεται σε κάποιο πρωτόκολλο κρυπτογράφησης που μεταφέρεται μέσα στο tunnel για να παρέχει ασφάλεια. Το L2TP ενεργεί σαν L2 πρωτόκολλο αλλά τυπικά ανήκει στο Επίπεδο Συνόδου (Session Layer, L5). Βασίζεται στα παλαιότερα πρωτόκολλα Layer 2 Forwarding Protocol (L2F) και Point-to-Point Tunneling Protocol (PPTP). Η τελευταία έκδοση L2TPv3 παρέχει επιπλέον επιλογές ασφαλείας, βελτιωμένη ενθυλάκωση και δυνατότητα να μεταφέρει πλαίσια διαφορετικά του PPP πάνω από ένα διαδίκτυο IP, όπως Ethernet, Frame Relay, ATM κ.τ.λ. Το L2TPv3 μπορεί, λοιπόν, να χρησιμοποιηθεί ως υποδομή ενός VPLS.
- Multiprotocol Label Switching (MPLS): Το MPLS είναι ένας υψηλά επεκτάσιμος, ανεξάρτητος από κάθε τηλεπικοινωνιακό πρωτόκολλο μηχανισμός μεταφοράς πακέτων. Σε ένα δίκτυο MPLS

<sup>1</sup> <http://www.cisco.com>

ανατίθενται ετικέτες στα πακέτα δεδομένων. Οι αποφάσεις προώθησης των πακέτων γίνονται αποκλειστικά με βάση το περιεχόμενο των ετικετών. Έτσι δημιουργούνται "κυκλώματα" από άκρη σε άκρη (end-to-end) ανεξαρτήτως του τύπου του μέσου μεταφοράς και του πρωτοκόλλου. Το πλεονέκτημα του MPLS είναι ότι εξαλείφει κάθε εξάρτηση σε συγκεκριμένη τεχνολογία Data Link Layer, καθώς και την ανάγκη πολλαπλών δικτύων L2 για την ικανοποίηση διαφορετικών τύπων κίνησης. Το MPLS αποτελεί, ουσιαστικά, έναν τρόπο δημιουργίας εικονικών ζεύξεων (virtual links) μεταξύ απομακρυσμένων κόμβων και μπορεί να ενθυλακώσει πολλά διαφορετικά είδη πρωτοκόλλων.

- **Generic Routing Encapsulation (GRE):** Το GRE είναι ένα tunneling πρωτόκολλο αναπτυγμένο από την Cisco Systems που μπορεί να δεχτεί σαν payload μία μεγάλη ποικιλία από πρωτόκολλα, μεταφέροντας τα μέσα από εικονικά point-to-point links πάνω από ένα διαδίκτυο πρωτοκόλλου IP. Όπως και το L2TP, το GRE δεν δημιουργεί ασφαλή tunnels αλλά επιτρέπει την ενθυλάκωση πρωτοκόλλων που παρέχουν ασφάλεια, όπως το IPsec. Το GRE, όπως λέει και το όνομά του, έχει ως σκοπό να αποτελέσει ένα γενικό πρωτόκολλο, ικανό να ενθυλακώσει μία μεγάλη ποικιλία από πρωτόκολλα. Ένα από τα πρωτόκολλα που μπορούν να αποτελέσουν το payload protocol είναι και το Ethernet. Η τεχνολογία αυτή ονομάζεται EoGRE και μπορεί να χρησιμοποιηθεί για να στηρίξει ένα VPLS.

## 2.3 Overlay Transport Virtualization (OTV)<sup>2</sup>



Εικόνα 2: Επικοινωνία Εικονικών Δικτύων μέσω του OTV

Το τελευταίο διάστημα γίνεται μεγάλη συζήτηση για μια νέα τεχνολογία της Cisco η οποία ονομάζεται Overlay Transport Virtualization (OTV). Το OTV είναι μια διεύθυνση MAC στην IP τεχνολογία, που υποστηρίζει L2 VPNs πάνω από οποιοδήποτε μέσο μεταφοράς είτε αυτό είναι επιπέδου 2, είτε επιπέδου

3. Χρησιμοποιώντας τις αρχές της δρομολόγησης διευθύνσεων MAC,

το OTV παρέχει ένα επίπεδο που ενεργοποιεί την συνδεσιμότητα σε L2, μεταξύ δύο τελείως ξεχωριστών domain τα οποία έχουν με τη σειρά τους τοπολογία βασισμένη σε L2, ενώ ταυτόχρονα, διατηρεί το πλεονέκτημα της απομόνωσης των σφαλμάτων, που παρέχει μια διασύνδεση βασισμένη στο IP. Οι βασικές αρχές που διέπουν το OTV είναι, η χρήση ενός πρωτοκόλλου ελέγχου που αφορά στην παροχή των απαραίτητων πληροφοριών για να γίνονται γνωστές οι διευθύνσεις MAC (σε αντίθεση με ένα πρωτόκολλο data-plane learning) καθώς και η μεταγωγή πακέτων του IP πρωτοκόλλου ενθυλακωμένες σε L2 κίνηση (για την αποφυγή χρησιμοποίησης κυκλωματικής μεταγωγής).

<sup>2</sup> <http://www.cisco.com/en/US/netsol/ns1153/index.html>

### 2.3.1 Nexus 7000 Switch<sup>3</sup>

Για την κατανόηση του OVT θα χρησιμοποιήσουμε ως παράδειγμα, ένα από τα προϊόντα της Cisco Systems που χρησιμοποιεί τη συγκεκριμένη τεχνολογία. Το συγκεκριμένο προϊόν ονομάζεται Nexus 7000 switch. Πρόκειται για ένα μεταγωγέα (switch) ο οποίος χρησιμοποιείται για την επικοινωνία δύο ή περισσότερων απομονωμένων δικτύων.

Μέσω του OVT δημιουργείται ένα επίπεδο ελέγχου δια της ταυτοποίησης των ενδιαφερόμενων πλευρών (γνωστών ως edge nodes). Αφού δημιουργηθεί και εγκατασταθεί η σύνδεση, είναι εφικτή η δρομολόγηση, ενθυλακώνοντας την πληροφορία και αποστέλλοντας την μέσω της υποδομή του IP πρωτοκόλλου. Η δρομολόγηση καθορίζεται συνδέοντας μια διεύθυνση MAC με την Next-Hop IP Address. Η διαδικασία αυτή είναι πλήρως δυναμική και δεν απαιτείται η δημιουργία και η διαχείριση σήραγγας (tunnel) ή εικονικού καλωδίου (virtual wire). Με τον τρόπο αυτό απλοποιείται η διαχείριση και η επίβλεψη, ενώ ταυτόχρονα ο χρήστης έχει τη δυνατότητα να εκμεταλλεύεται πλήρως τον πυρήνα IP έχοντας, για παράδειγμα, βέλτιστη δρομολόγηση και σημαντικά χαρακτηριστικά όπως εξισορρόπηση φορτίου (load balancing), πολλαπλή αναπαραγωγή κίνησης (multicast traffic replication) και γρήγορη ανάκαμψη από τυχόν σφάλματα (fast failover).

Μερικά από τα βασικά πλεονεκτήματα που επιτυγχάνονται μέσω του OVT είναι:

- Η μηδενική διαφοροποίηση στις τοπολογίες δικτύων που έχουν δημιουργηθεί. Η τεχνολογία OVT αποτελεί μια αόρατη σύνδεση μεταξύ των ενδιαφερόμενων πλευρών η οποία συνδέει διαφορετικές τεχνολογίες και πρωτόκολλα μέσω του επιπέδου 2 (point-to-cloud model).
- Η ενσωματωμένες πληροφορίες που διαθέτει, καθιστά μη αναγκαία τη χρήση βοηθητικών πρωτοκόλλων όπως το VPLS.
- Η απομόνωση τυχόν αστοχιών καθώς δεν βασίζεται στο μηχανισμό των πλημμύρων για να διαδώσει πληροφορίες προσβασιμότητας για τις διευθύνσεις MAC. Αντ' αυτού, χρησιμοποιείται ένα πρωτόκολλο ελέγχου για τη διανομή των πληροφοριών ενώ οι ενδιαφερόμενες πλευρές παραμένουν ανεξάρτητες μεταξύ τους και οι αποτυχίες δεν διαδίδονται πέρα από την άκρη της συσκευής του OTV.
- Η δυναμική του φύση αποφεύγει την εκθετική περιπλοκότητα των εικονικών καλωδίων (virtual wires) καθώς και το ρίσκο μιας πιθανής πλημμύρας (flooding).
- Η αξιοποίηση των δυνατοτήτων του IP multicast για βέλτιστη αναπαραγωγή της κυκλοφορίας.
- Η ενοποίηση της κυκλοφορίας από πολλαπλά VLANs καθώς υποστηρίζει πάνω από 4000 VLAN ID σε ένα μόνο 802.1Q domain.
- Η διαμόρφωση χωρίς μεγάλη δυσκολία για το χρήστη.

---

<sup>3</sup> <http://www.cisco.com/en/US/products/ps9402/index.html>

## 2.4 ΕΙΚΟΝΙΚΟΠΟΙΗΣΗ (VIRTUALIZATION) (5)

Η σημερινή δομή των δικτύων υπολογιστών καθώς και το διαδίκτυο με τη μορφή που έχει, αποτελεί τροχοπέδη στην ανάπτυξη και υλοποίηση νέων αρχιτεκτονικών, τεχνολογιών και υπηρεσιών. Για την αντιμετώπιση των περισσότερων προβλημάτων οι περισσότερες μελέτες έστρεψαν το βλέμμα τους σε νέες πλατφόρμες και νέες μεθόδους. Τα εικονικά δίκτυα για τα οποία γίνεται λόγος, υφίστανται αρκετά χρόνια στον χώρο των τεχνολογιών πληροφορικής και των επικοινωνιών, προσφέροντας απομόνωση μεταξύ των εκτελούμενων υπηρεσιών και αρχιτεκτονικών. Η τεχνική της εικονικοποίησης (virtualization) δίνει μία πιθανή λύση στο αδιέξοδο που έχει οδηγηθεί το διαδίκτυο. Με τον όρο εικονικοποίηση ορίζουμε ένα μηχανισμό σύμφωνα με τον οποίο φυσικά συστήματα μετατρέπονται σε ιδεατά. Κάθε φυσικός πόρος (επεξεργαστική ισχύς, μνήμη, δίκτυο, αποθηκευτικός χώρος κλπ.) γίνεται ένας ενιαίος πόρος και μοιράζεται ταυτόχρονα σε πολλά εικονικά συστήματα. Μπορούμε λοιπόν μέσω του virtualization να πετύχουμε έναν, ή και περισσότερους, από τους παρακάτω σκοπούς: δημιουργία ενός επιπέδου αφαίρεσης, απόκρυψη χαρακτηριστικών και τέλος δημιουργία απομονωμένων περιβαλλόντων.

Μερικά από τα πλεονεκτήματα του virtualization είναι τα παρακάτω:

- Οι εικονικές μηχανές (virtual machines ή VMs) άμεσο αποτέλεσμα του virtualization, λειτουργούν ως ένα ασφαλές περιβάλλον, για την εκτέλεση αγνώστων ή λιγότερο αξιόπιστων εφαρμογών.
- Ενοποίηση του φόρτου εργασίας πολλαπλών υποχρησιμοποιούμενων "υπολογιστών" σε λιγότερους με σκοπό την εξοικονόμηση hardware, λειτουργίες ελέγχου και διαχείρισης των υποδομών.
- Δημιουργία πολλαπλών ανεξάρτητων περιβαλλόντων εκτέλεσης και αύξηση της ποιότητας της υπηρεσίας, εξασφαλίζοντας συγκεκριμένη ορισμένη "ποσότητα" πόρων.
- Παροχή μη διαθέσιμου υλικού που το κόστος τους και η διαθεσιμότητά τους σε πραγματικά υλικά θα δημιουργούσε προβλήματα (π.χ. εικονικούς προσαρμογείς (virtual adapters) Ethernet, εικονικούς μεταγωγείς (virtual switches) Ethernet κ.τ.λ.).
- Διευκολύνει την μετανάστευση (migration) του λογισμικού, παρέχοντας έτσι ευκινησία (mobility) μεταξύ των ενδιαφερόμενων πλευρών.
- Βοηθάει στην παραγωγή δοκιμαστικών και άγνωστων για την κατάληξή τους, σεναρίων που είναι δύσκολο να παραχθούν στην πραγματικότητα, διευκολύνοντας έτσι την διαδικασία των δοκιμών.
- Σημαντική μείωση στο κόστος που θα προοριζόταν για την αγορά φυσικών πόρων και μηχανημάτων
- Συμβάλλει στην εξοικονόμηση ενέργειας και άρα πρόκειται για μία «πράσινη» τεχνολογία αφού η χρήση πολλαπλών μηχανημάτων και πόρων περιορίζεται σε μεγάλο βαθμό.

Σήμερα έχουν αναπτυχθεί αρκετές τεχνικές εικονικοποίησης και πλέον μπορούν να κατηγοριοποιηθούν κατάλληλα. Οι σημαντικότερες από αυτές τις κατηγορίες, δίνονται ακολούθως:

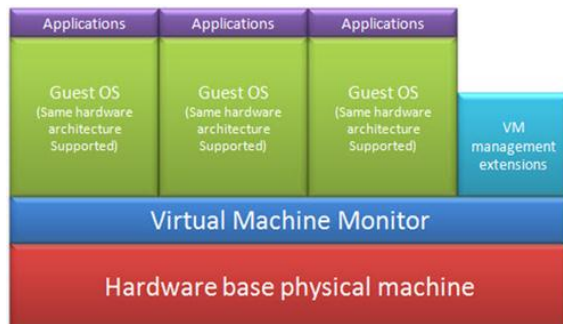
## Κατηγορίες Εικονικοποίησης

1. Υλικού
2. Λογισμικού
3. Δικτύων
4. Μνήμης
5. Αποθήκευσης
6. Δεδομένων

### 2.4.1 Εικονικοποίηση Υλικού (Hardware Virtualization)

Η εικονικοποίηση υλικού, αναφέρεται στη δημιουργία μιας εικονικής μηχανής, η οποία συμπεριφέρεται ακριβώς όπως ένας υπολογιστής με λειτουργικό σύστημα. Το λογισμικό το οποίο εκτελείται στις συγκεκριμένες εικονικές μηχανές, διαχωρίζεται από τους πόρους που διαθέτει το υλικό του υποκείμενου υλικού. Για παράδειγμα ένας υπολογιστής που διαθέτει συγκεκριμένο λειτουργικό σύστημα (Microsoft Windows), μπορεί να δημιουργήσει μια εικονική μηχανή που να διαθέτει λειτουργικό σύστημα με τελείως διαφορετική αρχιτεκτονική (Ubuntu Linux) και οι εφαρμογές της εικονικής μηχανής να λειτουργούν απροβλημάτιστα. Στην εικονικοποίηση υλικού, το φυσικό μηχάνημα (host) είναι αυτό στο οποίο λαμβάνει χώρα η εικονικοποίηση ενώ το φιλοξενούμενο μηχάνημα (guest) είναι το εικονικό μηχάνημα. Το λογισμικό που δημιουργεί την εικονική μηχανή καλείται “επόπτης” ή Hypervisor.

#### 2.4.1.1 Πλήρης Εικονικοποίηση (Full virtualization)



Εικόνα 3: Δομή πλήρους εικονικοποίησης

Η πλήρης εικονικοποίηση είναι μια τεχνική εικονικοποίησης η οποία χρησιμοποιείται για να παρέχει συγκεκριμένο τύπο περιβάλλοντος στις εικονικές μηχανές. Στο συγκεκριμένο τρόπο εικονικοποίησης, όλα τα βασικά χαρακτηριστικά του υλικού (hardware) αντανακλώνται στο VM συμπεριλαμβανομένων και όλων των εντολών καθοδήγησης, λειτουργίες εισόδου εξόδου, πρόσβαση

στη μνήμη καθώς και όλα εκείνα τα στοιχεία που λειτουργούν στο φυσικό μηχάνημα και οφείλουν να

λειτουργούν και στο εικονικό μηχάνημα. Σε ένα τέτοιο περιβάλλον οποιοδήποτε λογισμικό έχει τη δυνατότητα να τρέξει στον host πρέπει να μπορεί να τρέξει και στην εικονική μηχανή. Μία δοκιμή για την πλήρη εικονικοποίηση, είναι ένα λειτουργικό σύστημα που έχει τη δυνατότητα να εγκατασταθεί στον host, να μπορεί με επιτυχία να εγκατασταθεί και να λειτουργήσει στην εικονική μηχανή. Τέλος βασική προϋπόθεση για την πλήρη εικονικοποίηση είναι ότι η εκτέλεση συγκεκριμένων λειτουργιών, πρέπει να μένουν στην εικονική μηχανή και να μην επηρεάζουν την κατάσταση οποιασδήποτε άλλης εικονικής μηχανής. Ενώ ορισμένες εντολές μηχανής μπορούν να υλοποιηθούν κατευθείαν από το υλικό, άλλες εντολές δεν επιτρέπεται να εκτελεστούν απευθείας και πρέπει να εξομοιωθούν από το

λογισμικό εικονικοποίησης, τέτοιες εντολές είτε έχουν πρόσβαση, είτε επηρεάζουν καταστάσεις που βρίσκονται έξω από την εικονική μηχανή. Η πλήρης εικονικοποίηση θεωρείται επιτυχημένη για λειτουργίες όπως:

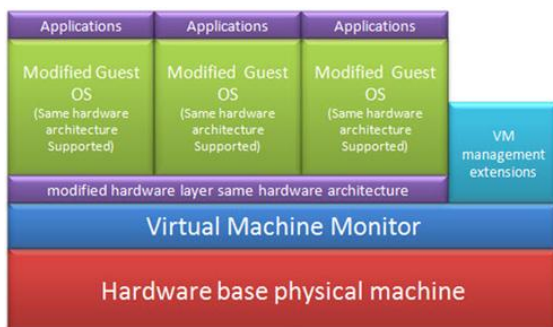
1. Ο διαμοιρασμός ενός υπολογιστικού συστήματος μεταξύ πολλαπλών χρηστών.
2. Η απομόνωση των χρηστών μεταξύ τους.
3. Η προσομοίωση καινούργιου hardware για την επίτευξη αυξημένης αξιοπιστίας, ασφάλειας και παραγωγικότητας.

#### 2.4.1.2 Μερική εικονικοποίηση (Partial virtualization)

Στη μερική εικονικοποίηση, η εικονική μηχανή εξομοιώνει πολλαπλά στιγμιότυπα ενός μεγάλου μέρους του υποκείμενου hardware περιβάλλοντος, και συγκεκριμένα χώρους διευθύνσεων (address spaces). Αυτό σημαίνει ότι δεν υποστηρίζεται η εγκατάσταση ενός πλήρους λειτουργικού συστήματος στην εικονική μηχανή, όπως στην πλήρη εικονικοποίηση που αναφερθήκαμε στο προηγούμενο κεφάλαιο, αλλά αντίθετα μπορούν να εκτελεστούν πολλές εφαρμογές. Μία σημαντική μορφή μερικής εικονικοποίησης είναι η εικονικοποίηση χώρου διευθύνσεων (address space virtualization), στην οποία το εικονικό μηχάνημα αποτελείται από ένα ανεξάρτητο address space. Αυτή η δυνατότητα απαιτεί hardware που να υποστηρίζει μετάθεση διευθύνσεων (address relocation).

Η μερική εικονικοποίηση αποτέλεσε ένα πολύ σημαντικό βήμα για την επίτευξη της πλήρους εικονικοποίησης και είναι σημαντικά πιο εύκολο να υλοποιηθεί. Παρέχει την δυνατότητα για την δημιουργία αξιόλογων virtual machines, δυνατά να υποστηρίξουν σημαντικές εφαρμογές. Ταυτόχρονα χρησιμοποιείται με επιτυχία για τον διαμοιρασμό υπολογιστικών πόρων μεταξύ πολλαπλών χρηστών. Ωστόσο, σε σύγκριση με την πλήρη εικονικοποίηση, μειονεκτεί σε καταστάσεις που απαιτούν συμβατότητα προς τα πίσω (backward compatibility) ή φορητότητα (portability). Εφόσον υπάρχει δυσκολία στην ακριβή πρόβλεψη των χαρακτηριστικών που χρησιμοποιούνται από μία δεδομένη εφαρμογή, όταν συγκεκριμένα χαρακτηριστικά του hardware δεν έχουν εξομοιωθεί οποιοδήποτε λογισμικό τα χρησιμοποιεί αποτυγχάνει.

#### 2.4.1.3 Παραεικονικοποίηση (Paravirtualization)



Εικόνα 4: Δομή παραεικονικοποίησης

Είναι μία τεχνική εικονικοποίησης που παρέχει στα εικονικά μηχανήματα μια διεπαφή λογισμικού (software interface) που είναι παρόμοια αλλά όχι πανομοιότυπη με αυτή του υποκείμενου hardware.

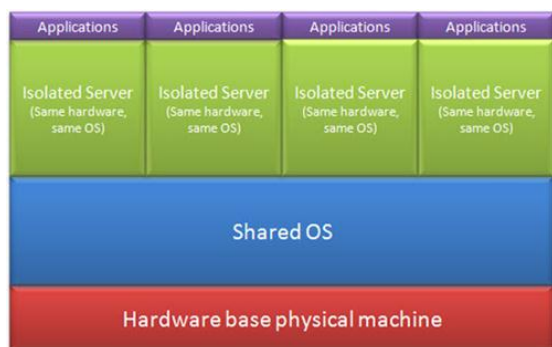
Ο στόχος της τροποποιημένης διεπαφής (interface) είναι να μειώσει το μερίδιο χρόνου εκτέλεσης λειτουργιών του φιλοξενούμενου (guest) που είναι συγκριτικά πιο δύσκολη σε ένα εικονικό περιβάλλον σε σύγκριση με ένα μη-εικονικό περιβάλλον. Η



παραεικονικοποίηση παρέχει ειδικά καθορισμένα "άγκιστρα" (hooks) που επιτρέπουν στους guests και στον host να ζητούν και να αναγνωρίζουν αυτές τις διεργασίες, που σε άλλη περίπτωση θα εκτελούνταν στον εικονικό τομέα (virtual domain) όπου η απόδοση είναι χειρότερη. Μία επιτυχημένη πλατφόρμα επιτρέπει στο διαχειριστή των εικονικών μηχανών (virtual machine monitor, VMM) να κάνει τις διαδικασίες απλούστερες, μεταφέροντας την εκτέλεση των κρίσιμων διεργασιών από το virtual domain στο host domain και μειώνοντας την συνολική μειωμένη απόδοση των εκτελέσεων στον guest.

Η παραεικονικοποίηση απαιτεί το λειτουργικό σύστημα του guest να είναι συμβατό με την διεπαφή προγραμματισμού της εφαρμογής (application programming interface, API) του paravirtualization. Ένα συμβατικό λειτουργικό που δεν αναγνωρίζει τη διαδικασία της παραεικονικοποίησης, δεν μπορεί να εκτελεστεί πάνω σε έναν αντίστοιχης τεχνολογίας VMM. Αντίθετα, σε περιπτώσεις όπου το λειτουργικό δεν μπορεί να τροποποιηθεί, γίνεται να υπάρχουν διαθέσιμα συστατικά που επιτρέπουν πολλά από τα σημαντικά πλεονεκτήματα επίδοσης του paravirtualization.

#### 2.4.1.4 Εικονικοποίηση σε επίπεδο λειτουργικού συστήματος (Operating system-level virtualization)



Εικόνα 5: Δομή εικονικοποίησης σε επίπεδο λειτουργικού συστήματος.

Είναι μία μέθοδος server virtualization όπου ο πυρήνας (kernel) του λειτουργικού συστήματος, επιτρέπει την δημιουργία πολλαπλών απομονωμένων στιγμιότυπων του σε επίπεδο χρήστη (user-space) αντί για ένα μοναδικό. Τέτοια στιγμιότυπα, που συχνά αποκαλούνται δοχεία (containers) ή φυλακές (jails), εμφανίζονται σαν πραγματικοί servers από την πλευρά των χρηστών τους. Σε συστήματα Unix, αυτή η τεχνολογία μπορεί να ταυτιστεί σε ένα βαθμό, με τον τυπικό chroot μηχανισμό. Εκτός από τους μηχανισμούς απομόνωσης (isolation), ο πυρήνας

παρέχει δυνατότητες διαχείρισης πόρων, ώστε να περιορίζει την επιρροή των δραστηριοτήτων του ενός container στα υπόλοιπα.

Η εικονικοποίηση σε επίπεδο λειτουργικού συστήματος, χρησιμοποιείται συνήθως σε περιβάλλοντα εικονικής φιλοξενίας (virtual hosting environments), όπου χρησιμεύει στην δέσμευση πόρων hardware ανάμεσα σε έναν μεγάλο αριθμό αμοιβαία μη εμπιστευόμενων (mutually-distrusting) χρηστών. Χρησιμοποιείται επίσης, σε μικρότερη κλίμακα, για την ενοποίηση του hardware του server μετακινώντας υπηρεσίες (services) απομονωμένων hosts σε έναν server. Παράλληλα, υπάρχουν περιπτώσεις που περιλαμβάνουν εφαρμογές σε διαχωρισμένα containers για αυξημένη ασφάλεια, ανεξαρτησία του hardware και επιπρόσθετες λειτουργίες διαχείρισης πόρων. Τέλος, υπάρχουν περιπτώσεις OS-level virtualization που παρέχουν ενεργή μετανάστευση (live migration) και μπορούν να χρησιμοποιηθούν για δυναμική εξισορρόπηση φόρτου (load balancing) μεταξύ κόμβων σε μία συστοιχία υπολογιστών (cluster). Μερικά από τα πλεονεκτήματα είναι ότι χρησιμοποιεί μηχανισμούς αποθήκευσης δεδομένων οι οποίοι καθιστούν τη διαδικασία αντιγράφων ασφαλείας και την επαναφορά του συστήματος σε προηγούμενη κατάσταση ακόμα ευκολότερη. Παράλληλα αυτή η

μορφή εικονικοποίησης, συνήθως δεν παρουσιάζει ιδιαίτερες απαιτήσεις σε πόρους, καθώς δε απαιτεί τη χρήση του hardware ούτε προσομοιώνει διαδικασίες για να εκτελέσει προγράμματα. Αντίθετα ένα από τα μειονεκτήματα του είναι ότι δεν παρουσιάζει μεγάλη ευελιξία όπως άλλες υλοποιήσεις εικονικοποίησης, αφού δε μπορεί να φιλοξενήσει λειτουργικό σύστημα διαφορετικό από αυτό του host ούτε και διαφορετικό πυρήνα (ο guest).

## 2.4.2 Εικονικοποίηση δικτύων (Network Virtualization) (6)

Εικονικοποίηση δικτύων (Network Virtualization) είναι η διαδικασία κατά την οποία συνδυάζονται δικτυακοί πόροι υλικού και λογισμικού καθώς και δικτυακή λειτουργικότητα, σε μία διαχειριστική οντότητα λογισμικού η οποία ονομάζεται εικονικό δίκτυο (virtual network). Ο σκοπός της εικονικοποίησης δικτύων είναι ο διαμοιρασμός δικτυακών πόρων σε συστήματα και χρήστες με αποδοτικό, ελεγχόμενο και ασφαλή τρόπο. Τα εικονικά δίκτυα χωρίζονται σε δύο μεγάλες κατηγορίες, στα εσωτερικά και στα εξωτερικά.

Τα εξωτερικά εικονικά δίκτυα αποτελούνται από ένα ή περισσότερα τοπικά δίκτυα τα οποία είτε συνδυάστηκαν, είτε χωρίστηκαν με απώτερο στόχο τη βελτίωση της αποτελεσματικότητας του προηγούμενου δικτύου. Τα κύρια συστατικά ενός εξωτερικού εικονικού δικτύου, είναι τα VLANs και οι μεταγωγείς δικτύου (switch). Με τη χρησιμοποίηση των VLAN και της τεχνολογίας μεταγωγέων, ο διαχειριστής, μπορεί να διαμορφώσει συστήματα τα οποία είναι συνδεδεμένα με φυσικό τρόπο στο ίδιο τοπικό δίκτυο, έτσι ώστε να ανήκουν σε διαφορετικά μεταξύ τους εικονικά δίκτυα.

Ένα εσωτερικό εικονικό δίκτυο αποτελείται από ένα σύστημα που χρησιμοποιεί εικονικές μηχανές ή παρεμφερείς τεχνολογίες προσαρμοσμένο πάνω σε μία τουλάχιστον ψευδο-δικτυακή διεπαφή (pseudo-network interface). Τα συγκεκριμένα virtual machines, μπορούν να επικοινωνούν μεταξύ τους σαν να συνυπήρχαν στο ίδιο τοπικό δίκτυο, δημιουργώντας έτσι ένα εικονικό δίκτυο. Τα συστατικά στοιχεία του είναι εικονικές δικτυακές διεπαφές (virtual network interface) ή virtual NICs (vNICs10) και εικονικοί μεταγωγείς (virtual switches).

### 2.4.2.1 Συστατικά Εικονικών Δικτύων

Ένα virtual network στηρίζεται πάνω σε 4 βασικά στοιχεία:

- virtual hosts, που εκτελούν λογισμικό και προωθούν πακέτα,
- virtual links, που μεταφέρουν τα πακέτα μεταξύ των virtual hosts,
- virtual switches, που εκτελούν λειτουργίες για την δημιουργία εικονικών τοπικών δικτύων
- logical routers, που εκτελούν τις υπηρεσίες δρομολόγησης σε εικονικό διαδικτυακό επίπεδο.

Ο virtual host παρέχει την ψευδαίσθηση ενός αφιερωμένου φυσικού οικοδεσπότη (physical host), ενώ στην πραγματικότητα μπορεί να υπάρχουν ταυτόχρονα και άλλοι virtual hosts στο ίδιο physical hardware. Ένας virtual host έχει αφιερωμένους φυσικούς πόρους (physical resources) ή λογικούς πόρους (logical resources). Παραδείγματα των physical resources είναι η CPU, η μνήμη και το bandwidth, ενώ τα logical resources είναι πόροι που υλοποιούνται από το OS, όπως ο πίνακας διεργασιών (process table), ο πίνακας σελίδων (page table), ο πίνακας προώθησης IPv4 (IPv4

forwarding table) και η ενδιάμεση μνήμη (memory buffer). Στην πραγματικότητα, όλοι αυτοί οι πόροι διατίθενται από ένα virtualization layer που υλοποιεί την αφαίρεση του virtual host. Το virtualization layer δημιουργεί εικονικούς πόρους από τους φυσικούς χρησιμοποιώντας μηχανισμούς δέσμευσης πόρων και χρονοδρομολόγησης, έτσι ώστε κάθε αφηρημένος virtual host να δέχεται τους αναμενόμενους πόρους που ζητήθηκαν. Ακόμη, η αφαίρεση του virtual host οριοθετεί το εύρος των logical resources ώστε κάθε virtual host μπορεί να διαχειριστεί τα δικά του logical resources. Το virtualization layer μπορεί να μην εικονικοποιεί όλους τους διαθέσιμους πόρους, με πιθανό αποτέλεσμα την ύπαρξη πόρων στο φυσικό μηχάνημα. Αυτοί οι πόροι είτε θα είναι μη διαθέσιμοι, είτε θα υποστηρίζουν περιορισμένη "αλληλεπίδραση" με το εσωτερικό του virtual host, είτε χρησιμοποιούνται και από άλλες εφαρμογές ταυτόχρονα.

Οι virtual hosts μπορούν να επιτύχουν δύο είδη isolation:

- **Resource isolation:** Εξασφαλίζει ότι κανένας virtual host δεν μπορεί έχει πρόσβαση στους πόρους άλλου virtual host. Οι resource allocators στο virtualization layer πολυπλέκουν και προγραμματίζουν τους φυσικούς πόρους ώστε να παρέχουν εικονικούς πόρους μέσα σε έναν virtual host.
- **Namespace isolation:** Εξασφαλίζει ότι κάθε virtual host μπορεί να ονομάσει και να αναφερθεί σε πόρους (όπως διεργασίες, αρχεία, μνήμη, δικτυακά interface, δικτυακές διευθύνσεις, forwarding tables) και δεν δύναται να αναφερθεί σε πόρους άλλων πλαισίων. Για παράδειγμα, μία εφαρμογή σε έναν virtual host δεν μπορεί να προσθέσει διαδρομές στην Βάση Πληροφοριών Προώθησης (Forward Information Base, FIB) ενός άλλου virtual host, και δύο ή περισσότεροι virtual hosts μπορούν να χρησιμοποιούν την ίδια διεύθυνση IP για να ονομάσουν διαφορετικά εικονικά δικτυακά interfaces.

Ένα virtual link φαίνεται σαν ένα physical link, αλλά πολλά virtual links μπορεί να μοιράζονται ένα physical link. Από την άλλη, ένα virtual link μπορεί να αποτελείται από πολλά hops στο υποκείμενο φυσικό δίκτυο.

Τα virtual switches είναι software που επιτρέπει την επικοινωνία μεταξύ πολλών VMs. Όπως και ένα φυσικό Ethernet switch, ένα virtual switch μπορεί να κάνει περισσότερα από την απλή προώθηση πακέτων. Μπορεί να διαχειριστεί έξυπνα την δικτυακή επικοινωνία ελέγχοντας τα πακέτα πριν τα προωθήσει. Τα virtual switches είτε ενσωματώνονται στο λογισμικό για virtualization, είτε μπορεί να περιέχονται στο firmware ενός φυσικού server.

Οι λογικοί δρομολογητές (logical routers) χωρίζουν έναν φυσικό router σε πολλαπλούς logical routers που διατηρούν τους δικούς τους πίνακες δρομολόγησης (routing tables), interfaces, policies, και routing protocols. Το βασικό ζητούμενο των logical routers είναι η ενοποίηση πολλαπλών δικτυακών στοιχείων σε μία συσκευή hardware, απλοποιώντας έτσι τις ρυθμίσεις σε φυσικό επίπεδο (π.χ. καλώδια και racks) και μειώνοντας τον απαιτούμενο φυσικό χώρο καθώς και τα κόστη. Επιπλέον, η ύπαρξη των logical routers επιτρέπει προσαρμοσμένες ρυθμίσεις δρομολόγησης ή και διαφορετικά πρωτόκολλα δρομολόγησης για συγκεκριμένες εφαρμογές.

### 2.4.2.2 Τεχνικές Εικονικοποίησης Δικτύων

Η έννοια των πολλαπλών συνυπαρχόντων λογικών δικτύων είναι αυτή που περιγράφει, ουσιαστικά τον όρο εικονικό δίκτυο. Τα εικονικά δίκτυα μπορούν κατηγοριοποιηθούν σε τέσσερις βασικές κλάσεις ως εξής:

- **Virtual Local Area Networks (VLANs):** Ένα VLAN είναι ένα σύνολο από λογικά δικτυωμένους hosts με μοναδικό broadcast domain ανεξάρτητα της φυσικής συνδεσιμότητας, σύμφωνα με το πρωτόκολλο 802.11. Όλα τα πλαίσια σε ένα VLAN περιέχουν ένα VLAN ID στην επικεφαλίδα MAC, και οι μεταγωγείς με δυνατότητες VLAN χρησιμοποιούν τόσο την διεύθυνση MAC όσο και το VLAN ID για να προωθήσουν τα πλαίσια. Εφόσον τα VLANs βασίζονται σε λογικές αντί για φυσικές ζεύξεις η διαχείριση του δικτύου, ο έλεγχος του και η επαναρύθμιση των VLANs είναι ευκολότερη από αυτές των αντίστοιχων φυσικών. Επιπρόσθετα, τα VLANs παρέχουν αυξημένα επίπεδα απομόνωσης.
- **Virtual Private Networks (VPNs):** Ένα VPN είναι ένα dedicated δίκτυο που συνδέει πολλαπλές τοποθεσίες χρησιμοποιώντας ιδιωτικές και ασφαλείς σήραγγες (tunnels) τοποθετημένες πάνω σε δημόσια δίκτυα επικοινωνίας όπως το Ίντερνετ. Στις περισσότερες περιπτώσεις τα VPNs ενώνουν απομακρυσμένες γεωγραφικά τοποθεσίες μίας εταιρικής επιχείρησης. Κάθε τοποθεσία VPN περιέχει μία ή περισσότερες συσκευές άκρου πελάτη(customer edge) που συνδέονται με έναν ή περισσότερους δρομολογητές άκρου παροχέα (provider edge).
- **Active and Programmable Networks:** Η έρευνα πάνω στα Active and Programmable Networks πυροδοτήθηκε από την ανάγκη για δημιουργία, ανάπτυξη και διαχείριση νέων υπηρεσιών γρήγορα με βάση τις ανάγκες των χρηστών. Εκτός από προγραμματισμότητα, προσφέρουν έννοιες όπως απομονωμένα περιβάλλοντα που επιτρέπουν σε πολλαπλούς χρήστες να εκτελούν πιθανώς συγκρουόμενους κώδικες στα ίδια δικτυακά στοιχεία χωρίς να προκαλούν δικτυακές αστάθειες.
- **Overlay Networks:** Αποτελούν λογικά δίκτυα τοποθετημένα πάνω σε ένα ή περισσότερα υπαρκτά φυσικά δίκτυα. Το ίδιο το Ίντερνετ ξεκίνησε σαν overlay πάνω από το τηλεπικοινωνιακό δίκτυο. Στο σημερινό Ίντερνετ τα overlays δημιουργούνται, κυρίως, στο επίπεδο εφαρμογής. Υπάρχουν, όμως, και υλοποιήσεις σε χαμηλότερα επίπεδα της δικτυακής στοίβας (network stack). Τα overlay δεν χρειάζονται, ούτε προκαλούν, αλλαγές στο underlay δίκτυο. Σαν αποτέλεσμα, έχουν χρησιμοποιηθεί εκτενώς ως σχετικά απλός και ανέξοδος τρόπος για την υλοποίηση νέων χαρακτηριστικών και διόρθωση λαθών στο Ίντερνετ.

Ένα πλήθος overlay αρχιτεκτονικών έχουν προταθεί τα τελευταία χρόνια με σκοπό την επίλυση ποικίλων προβλημάτων όπως: δυνατότητα multicasting, παροχή υπηρεσιών QoS, διαθεσιμότητα δρομολόγησης στο Ίντερνετ, προστασία από επιθέσεις DOS(Denial of Service), δίκτυα για Content Distribution (CDNs) και διαμοιρασμό αρχείων. Ακόμα μία χρήση τους είναι στην δημιουργία testbed με σκοπό τον σχεδιασμό και την αξιολόγηση νέων αρχιτεκτονικών.

## ΚΕΦΑΛΑΙΟ 3

### ΠΛΑΤΦΟΡΜΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

Για την εκπόνηση της συγκεκριμένης διπλωματικής εργασίας θα στηριχτούμε σε τρεις πλατφόρμες, κάθε μια από τις οποίες επιτελεί διαφορετικό σκοπό. Η πρώτη πλατφόρμα εκτελεί καθήκοντα Physical Host πάνω στον οποίο θα γίνουν τα πειράματα και οι μετρήσεις και ο οποίος έχει επάρκεια σε επεξεργαστική ισχύ, μνήμη και αποθηκευτικό χώρο. Η δεύτερη πλατφόρμα παρέχει το απαραίτητο λογισμικό για τη δημιουργία της τοπολογίας εικονικών δικτύων. Η τελευταία πλατφόρμα αφορά την τεχνολογία που υλοποιεί το virtual switching. Ακολουθώς αναλύονται οι κύριες πλατφόρμες που χρησιμοποιήθηκαν. Παράλληλα θα γίνει μια παρουσίαση των επιμέρους εργαλείων που ήταν απαραίτητα για τις μετρήσεις και τα αποτελέσματα

#### 3.1 Hardware

Για την τοπολογία δικτύου που θα μελετήσουμε στη συγκεκριμένη διπλωματική, χρησιμοποιήθηκε για τα πειράματα, φορητός υπολογιστής<sup>4</sup> ο οποίος διαθέτει το παρακάτω υλικό.

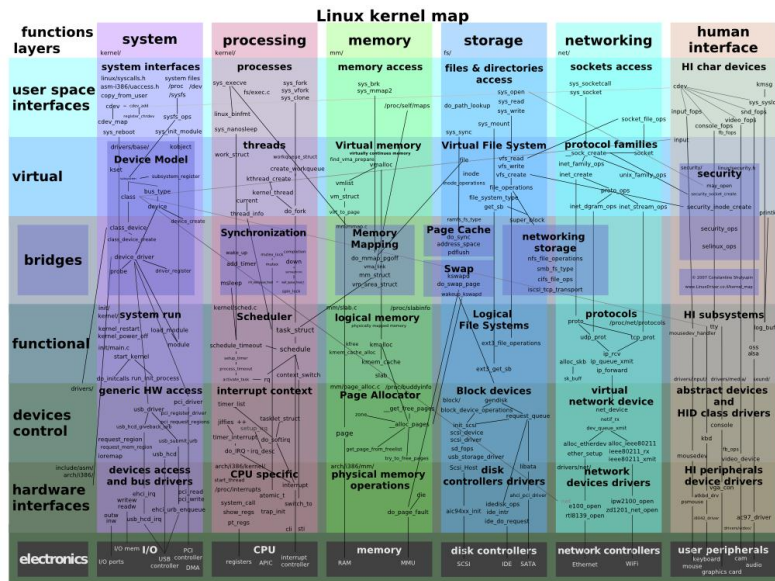
Processors:	1 (2Cores)		
Processor Speed:	2.8 GHz	Processor Type:	Core 2 Duo (T9600)
System Bus Speed:	1066 MHz	Cache Bus Speed:	2.8 GHz (Built-in)
ROM/Firmware Type:	EFI	EFI Architecture:	64-Bit
L1 Cache:	32k/32k	L2/L3 Cache:	6 MB (on chip)
RAM Type:	PC3-8500 DDR3	RAM Speed:	1333 MHz
RAM:	8 GB		
Hard Drive:	SSD 256GB	Int. HD Interface:	Serial ATA (3 Gb/s)
Standard Optical:	8X DL "SuperDrive"	Standard Disk:	None
Standard AirPort:	802.11a/b/g/n	Standard Ethernet:	10/100/1000Base-T
USB Ports:	2 (2.0)	Firewire Ports:	1 (800)
Battery Type:	73 W h Li-Poly	Battery Life:	7 Hours
Dimensions:	0.95 x 14.35 x 9.82	Weight:	5.5 lbs (2.49 kg)

Πίνακας 1: Υλικό του φυσικού μηχανήματος που χρησιμοποιήθηκε

<sup>4</sup> <http://www.apple.com>

## 3.2 Software

### 3.2.1 Linux Kernel



Εικόνα 6: Χάρτης πυρήνα Linux (Linux Kernel Map)

Ο πυρήνας Linux είναι ο πυρήνας της οικογένειας λειτουργικών συστημάτων GNU/Linux. Πρόκειται για ένα παγκόσμιο εγχείρημα το οποίο άλλαξε τον τρόπο με τον οποίο δομούνται τα λειτουργικά συστήματα και έδωσε το δικαίωμα σε κάθε χρήστη που έχει την απαραίτητη γνώση να συμβάλει στην ανάπτυξη και στην βελτίωση του.

Είναι μία πρωτότυπη υλοποίηση πυρήνα λειτουργικού συστήματος.

Αν και δεν χρησιμοποιεί κώδικα του UNIX, μπορεί να θεωρηθεί

παρεμφερές σύστημα (ο ακριβής όρος είναι *Unix-like*) ή ελεύθερη υλοποίησή του, αφού διαθέτει τις περισσότερες εντολές του και την ίδια σχεδόν δομή αρχείων, ενώ η φιλοσοφία της σχεδιάσής του πλησιάζει περισσότερο το UNIX από οποιοδήποτε άλλο λειτουργικό σύστημα.

Σήμερα το Linux παρέχει όλα όσα θεωρούνται αναγκαία για ένα σύγχρονο πυρήνα λειτουργικού, όπως:

- Υποστήριξη πολυεπεξεργαστικών συστημάτων (SMP)
- πραγματική πολυδιεργασία
- εικονική μνήμη
- διαμοιραζόμενες βιβλιοθήκες
- σωστή διαχείριση μνήμης
- δικτύωση μέσω TCP/IP κ.α.

Ο πυρήνας Linux αρχικά σχεδιάστηκε για επεξεργαστές της οικογένειας x86 (386/486/Pentium), αλλά σήμερα τρέχει σε πολύ μεγάλη ποικιλία επεξεργαστών, όπως οι Alpha (64 bit)<sup>5</sup>, οι Motorola 68000 (Amiga)<sup>6</sup>, PowerPC<sup>7</sup>, MIPS<sup>8</sup> κ.α.

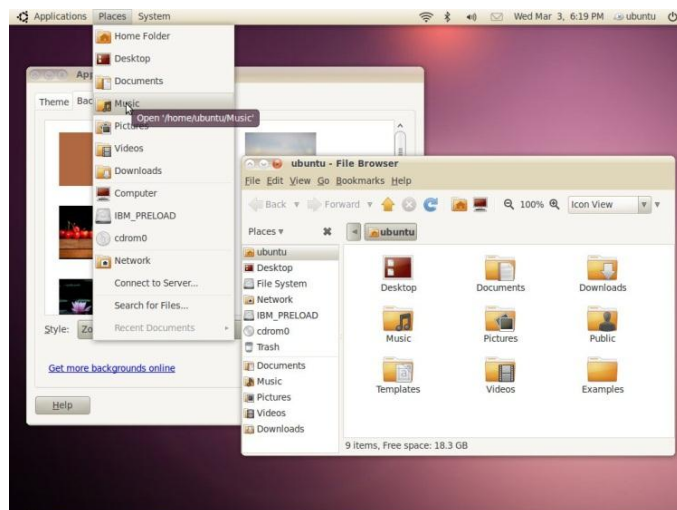
<sup>5</sup> [http://en.wikipedia.org/wiki/DEC\\_Alpha](http://en.wikipedia.org/wiki/DEC_Alpha)

<sup>6</sup> [http://en.wikipedia.org/wiki/Motorola\\_68000\\_family](http://en.wikipedia.org/wiki/Motorola_68000_family)

<sup>7</sup> <http://en.wikipedia.org/wiki/PowerPC>

Κατά την εκπόνηση της διπλωματικής εργασίας, βασιστήκαμε στην τελευταία έκδοση του πυρήνα Linux και συγκεκριμένα στην έκδοση 3.χ.χ. Δεν έγινε κάποια τροποποίηση στον πυρήνα, αλλά λειτουργίες όπως το Linux bridging απενεργοποιήθηκαν για τους σκοπούς της εργασίας.

### 3.2.2 Ubuntu/Linux Distribution (7)



Εικόνα 7: Τυπικό περιβάλλον Ubuntu 10.04 LTS

Το βασικό σημείο ενός λειτουργικού συστήματος είναι ο πυρήνας του. Ωστόσο για την επίτευξη των στόχων που τέθηκαν, είναι απαραίτητη η χρήση ορισμένων εφαρμογών για την παρατήρηση και την αξιολόγηση των επιδόσεων του εικονικού μεταγωγέα. Για το λόγο αυτό χρησιμοποιήθηκε η πλέον διαδεδομένη έκδοση Linux, που ονομάζεται Ubuntu (10.04 Long-Term Support).

Η δομή και η αρχιτεκτονική του Ubuntu έχει τις βάσεις του στο Debian αλλά διαφέρει

στην κοινότητα που το αναπτύσσει καθώς και στη διαδικασία που γίνονται οι ανανεώσεις

του. Το Ubuntu συνεπώς παρέχει συχνές και ανά τακτά χρονικά διαστήματα εκδόσεις και ένα περιβάλλον εργασίας φιλικό προς το χρήστη ενώ διατίθεται στο κοινό τόσο σε έκδοση Desktop όσο και σε έκδοση Server.

Το σημαντικότερο πλεονέκτημα του Ubuntu, και γενικά των Linux-based λειτουργικών συστημάτων, είναι ότι πρόκειται για προϊόν ανοικτού κώδικα με άμεση συνέπεια τη δωρεάν διανομή του σε κάθε ενδιαφερόμενο. Ταυτόχρονα παρέχει μια μεγάλη ποικιλία από εφαρμογές, οι οποίες διατίθενται με το λειτουργικό σύστημα και δίνονται με δωρεάν άδεια χρήσης. Η ασφάλεια, η σταθερότητα και η ευκολία στη χρήση είναι τα τρία κύρια σημεία που εστιάζει η διανομή αυτή. Το εργαλείο “sudo” δίνει τη δυνατότητα για έλεγχο των εφαρμογών και των λειτουργιών που εγκαθίστανται στο λειτουργικό σύστημα, μόνο από το βασικό λογαριασμό, απομονώνοντας με τον τρόπο αυτό περιπτώσεις άπειρων χρηστών που θα μπορούσαν να προκαλέσουν ανεπανόρθωτη ζημιά στο λειτουργικό σύστημα. Το Ubuntu χρησιμοποιεί γραφικό περιβάλλον με επιφάνειες εργασίας βασισμένο στο GNU Panel. Σημαντικό σημείο της ασφάλειας είναι ότι με τη χρήση τοίχους προστασίας (firewall) κλείνει τις δικτυακές πόρτες που δεν είναι απαραίτητο να είναι ανοιχτές. Παράλληλα έχει τη δυνατότητα να «τρέχει» εφαρμογές που είναι σχεδιασμένες για άλλα λειτουργικά συστήματα με τη χρήση κατάλληλου λογισμικού.

Τέλος, το Ubuntu μεταγλωττίζει (compiles) τα πακέτα προς εγκατάσταση χρησιμοποιώντας χαρακτηριστικά από τη συλλογή μεταγλωττιστών που διαθέτει (GNU Compiler Collection – GCC) όπως

<sup>8</sup> [http://en.wikipedia.org/wiki/MIPS\\_architecture](http://en.wikipedia.org/wiki/MIPS_architecture)

το Position-Independent Code (PIC) και το Buffer overflow protection παρέχοντας με τον τρόπο αυτό ακόμα μεγαλύτερη ασφάλεια, με κόστος στην επίδοση της τάξης του 0.01% σε αρχιτεκτονικές 64-bit.

### 3.2.3 Oracle VM VirtualBox (8)

#### 3.2.3.1 Επισκόπηση του VirtualBox



Εικόνα 8: Γραφικό περιβάλλον VirtualBox

Το VirtualBox είναι ένα προϊόν εικονικοποίησης για αρχιτεκτονικές επεξεργαστών x86. Αποτελεί μια αξιόπιστη λύση τόσο για επιχειρήσεις, όσο και για προσωπική χρήση και δοκιμές. Πρόκειται ένα λογισμικό ανοιχτού κώδικα και συνεπώς παρέχεται δωρεάν κάτω από τους όρους του GNU. Το VirtualBox μπορεί να εγκατασταθεί σε οποιοδήποτε λειτουργικό σύστημα υπάρχει διαθέσιμο αυτή τη στιγμή, ενώ τα λειτουργικά συστήματα που δύνανται να φιλοξενήσει διακρίνονται από μεγάλη ποικιλία (Windows NT, 2000, Xp, Server, Vista, Windows7, Solaris, OpenSolaris, Linux κτλ). Όπως και τα περισσότερα προγράμματα ανοιχτού κώδικα, έτσι και το VirtualBox έχει δυναμική ανάπτυξη από την

κοινότητα και συχνές αναβαθμίσεις, τόσο σε χαρακτηριστικά όσο και στα λειτουργικά συστήματα που μπορεί να φιλοξενήσει. Η εταιρία Oracle που συνδέεται με το προϊόν, βεβαιώνει ότι η κάθε έκδοση θα συνάδει με τα επαγγελματικά πρότυπα για την ποιότητα υπηρεσίας που προσφέρει.

Παρακάτω θα παρουσιαστούν ορισμένοι χρήσιμοι όροι οι οποίοι αποτελούν τη βάση για το σχεδιασμό, την ανάλυση και την επίτευξη των στόχων της παρούσας εργασίας και αφορούν το χώρο της εικονικοποίησης.

- **Host Operating System (Host OS):** Πρόκειται για το λειτουργικό σύστημα του υπολογιστή στο οποίο είναι εγκατεστημένο το VirtualBox.
- **Guest Operating System (Guest OS):** Πρόκειται για το λειτουργικό σύστημα το οποίο εγκαταστάθηκε στην εικονική μηχανή (Virtual Machine). Σε θεωρητική βάση το VirtualBox μπορεί να εγκαταστήσει και να λειτουργήσει λειτουργικά συστήματα που υποστηρίζουν x86 αρχιτεκτονικές. Για την εγγενή απόδοση του λειτουργικού συστήματος απαιτήθηκαν συγκεκριμένες βελτιστοποιήσεις, ξεχωριστές για κάθε λειτουργικό σύστημα, ώστε να έχουν τη μέγιστη απόδοση.
- **Virtual Machine (VM):** Πρόκειται για ένα ειδικό περιβάλλον το οποίο δημιουργείται από το λογισμικό του VirtualBox καθώς τρέχει το λειτουργικό σύστημα. Το VM εμφανίζεται είτε σαν παράθυρο στην επιφάνεια εργασίας είτε σε πλήρη οθόνη δίνοντας την αίσθηση απομονωμένης λειτουργίας. Εσωτερικά, το VirtualBox θεωρεί το VM ως ένα σύνολο παραμέτρων (μνήμη, αποθηκευτικός χώρος, επεξεργαστική ισχύς) που καθορίζουν τον τρόπο λειτουργίας του. Αυτές



οι παράμετροι εμφανίζονται και ελέγχονται από τον VirtualBox Manager (σε επίπεδο γραμμής εντολών με τη χρήση της εντολής VboxManage).

### 3.2.3.2 Χαρακτηριστικά του VirtualBox

Η επιλογή και η χρήση του VirtualBox έγινε, γιατί αποτελείται από ορισμένα χαρακτηριστικά τα οποία είναι απαραίτητα για τοπολογίες εικονικών δικτύων. Παρακάτω θα γίνει μια εκτενής παρουσίαση αυτών των βασικών χαρακτηριστικών του VirtualBox.

- **Φορητότητα:** Το VirtualBox, σε αντίθεση με τους hypervisor που λειτουργούν απευθείας σε επίπεδο hardware, απαιτεί την ύπαρξη ενός λειτουργικού συστήματος. Σε μεγάλο βαθμό, λειτουργεί με τον ίδιο ακριβώς τρόπο σε οποιοδήποτε φυσικό μηχάνημα εγκατασταθεί, ενώ εικονικές μηχανές που δημιουργήθηκαν σε ένα συγκεκριμένο Physical Host μπορούν να μεταφερθούν και να λειτουργήσουν σε διαφορετικό Host με διαφορετικό, για παράδειγμα, λειτουργικό σύστημα.
- **Καμία απαίτηση σε εικονικοποίηση υλικού:** Για να λειτουργήσει το VirtualBox δεν απαιτεί την ύπαρξη επεξεργαστή με χαρακτηριστικά σύγχρονης τεχνολογίας (Intel VT-x και AMD-V). Σε αντίθεση λοιπόν με άλλες πλατφόρμες εικονικοποίησης το VirtualBox διαφέρει καθώς μπορεί να χρησιμοποιηθεί με υλικό παλαιότερης τεχνολογίας.
- **Υποστήριξη υλικού:** Υποστηρίζει ένα μεγάλο αριθμό από hardware εξομοιώνοντας πλήρως τις λειτουργίες του. Τέτοιες λειτουργίες είναι η υποστήριξη πολλαπλών επεξεργαστών στο Guest Host, υποστήριξη USB device, πολλαπλή ανάλυση οθόνης, πλήρη ACPI υποστήριξη καθώς και iSCSI υποστήριξη.
- **Πολλαπλά Snapshots:** Παρέχει τη δυνατότητα για αποθήκευση μιας ή περισσότερων δεδομένης κατάστασης του συστήματος. Αυτό επιτρέπει την επιστροφή σε προηγούμενη κατάσταση δημιουργώντας μια διαφορετική συνέχεια για την κατάσταση του VM οδηγώντας στη δημιουργία ενός snapshot tree.
- **Δομημένη αρχιτεκτονική:** Διαθέτει ένα σωστά σχεδιασμένο και επακριβώς καθορισμένο εσωτερικό προγραμματισμό των interfaces που διαχωρίζει με καθαρό τρόπο των κώδικα που χρησιμοποιεί ο server και ο client. Για παράδειγμα, μπορεί ο χρήστης να ξεκινήσει ένα VM από το γραφικό περιβάλλον και να συνεχίσει με τον έλεγχο του από τη γραμμή εντολών.

### 3.2.3.3 Virtual Networking στο VirtualBox

Το λογισμικό VirtualBox παρέχει τη δυνατότητα στο χρήστη για χρησιμοποίηση άνω των 8 εικονικών καρτών PCI Ethernet για κάθε εικονικό μηχάνημα (VM). Επιπλέον, για κάθε μία από αυτές τις κάρτες δίνεται η δυνατότητα επιλογής του υλικού (hardware) που θα προσομοιώσει καθώς και του τρόπου που θα λειτουργήσει η εικονική κάρτα ανάλογα με την τοπολογία δικτύου που είναι επιθυμητή.

Για κάθε εικονική κάρτα δικτύου που ο χρήστη ενεργοποιεί στην εικονική μηχανή παρουσιάζονται οι παρακάτω προσομοιώσεις υλικού:

1. AMD PCNet PCI II (Am79C970A)
2. AMD PCNet FAST III (Am79C973, the default)
3. Intel PRO/1000 MT Desktop (82540EM)
4. Intel PRO/1000 T Server (82543GC)
5. Intel PRO/1000 MT Server (82545EM)
6. Paravirtualized network adapter (virtio-net)

Αντίστοιχα ο χρήστης μπορεί να επιλέξει από τους παρακάτω τρόπους δικτύωσης ώστε το εικονικό μηχάνημα να λειτουργήσει υπό αυτή τη μέθοδο.

1. **Not attached:** Στη συγκεκριμένη λειτουργία το εικονικό μηχάνημα θεωρεί ότι διαθέτει μία κάρτα δικτύου χωρίς όμως να υπάρχει κάποιου είδους συνδεσιμότητα (σα να μην υπάρχει καλώδιο Ethernet σε κάρτα δικτύου φυσικού μηχανήματος)
2. **Network Address Translation (NAT):** Κατά τη λειτουργία αυτή, επιτρέπονται όλες εκείνες οι διαδικασίες που επιτρέπουν στο εικονικό μηχάνημα πρόσβαση στο Internet καθώς και κάποιες απλές λειτουργίες file sharing
3. **Bridged Networking:** Στη λειτουργία αυτή, η οποία είναι και αυτή που χρησιμοποιήθηκε για τη διεξαγωγή όλων των μετρήσεων και πειραμάτων, η εικονική κάρτα δικτύου συνδέεται με μία κάρτα δικτύου του φυσικού μηχανήματος, παρακάμπτοντας το network stack του συστήματος.
4. **Internal Networking:** Χρησιμοποιείται για τη δημιουργία δικτύων μεταξύ των εικονικών μηχανών απομονωμένων από το φυσικό μηχάνημα και τις εφαρμογές που διαχειρίζεται.
5. **Host-only networking:** Η συγκεκριμένη λειτουργία επιλέγεται για τη δημιουργία ενός δικτύου αποτελούμενου από το Physical host και όσα εικονικά μηχανήματα επιλεγούν με αυτή τη λειτουργία.

#### 3.2.3.4 Συνοπτική παρουσίαση του Bridged Networking

Κατά τη λειτουργία του Bridged networking, το VirtualBox χρησιμοποιεί μια συσκευή οδηγό στον Physical Host, η οποία φιλτράρει δεδομένα από την φυσική κάρτα δικτύου. Για το λόγο αυτό, ο οδηγός (driver) αυτός ονομάζεται “net filter”. Αυτό δίνει τη δυνατότητα στο VirtualBox να συλλέγει δεδομένα από τον Physical Host καθώς και να εισάγει νέα, δημιουργώντας με τον τρόπο αυτό ένα νέο δικτυακό interface στο λογισμικό.

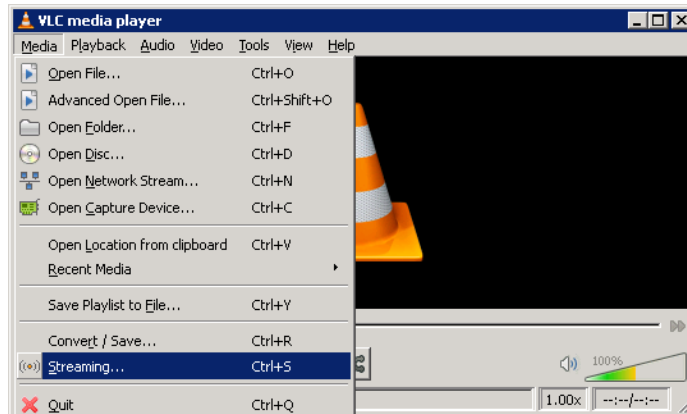
Όταν επιλεγεί η λειτουργία αυτή στο VM, εμφανίζεται στον Host σα να ήταν συνδεδεμένος σε ένα interface του. Ο host μπορεί να στείλει δεδομένα μέσω του συγκεκριμένου interface καθώς και να λάβει δεδομένα από αυτό. Συνεπώς με τη λειτουργία αυτή μπορούν πλέον να υπάρξουν διαδικασίες routing και bridging μεταξύ του VM και του υπόλοιπου δικτύου.

Για να επιτευχθεί ο συγκεκριμένος τρόπος λειτουργίας, απαιτείται όπως προαναφέρθηκε μια συσκευή οδηγός. Στην περίπτωσή μας, επειδή θα δημιουργηθούν αρκετά VM και οι φυσικοί πόροι δεν επαρκούν για να αντιστοιχηθεί μια φυσική κάρτα δικτύου σε κάθε VM, δημιουργήθηκαν TAP interfaces.

Ένα TAP interface προσομοιώνει μία συσκευή στο επίπεδο συνδέσμου και λειτουργεί με πακέτα επιπέδου 2 (π.χ Ethernet frames).

### 3.2.4 VLC Player (9)

#### 3.2.3.1 Επισκόπηση του VLC Player



Εικόνα 9: Γραφικό περιβάλλον VLC Player

Το VLC είναι ένα ελεύθερο λογισμικό ανοιχτού κώδικα και multimedia framework αναπαραγωγής και μετάδοσης πολυμέσων γραμμένο από το εγχείρημα VideoLAN. Πρόκειται ουσιαστικά για ένα πρόγραμμα αναπαραγωγής πολυμέσων, κωδικοποιητή και streamer υποστηρίζοντας πολλά codec ήχου και βίντεο καθώς και αρκετά πρωτόκολλα streaming. Έχει την ικανότητα streaming πάνω από οποιοδήποτε δίκτυο

και της επανακωδικοποίησης και αποθήκευσης αρχείων πολυμέσων σε

διάφορες μορφές. Η βασική του έκδοση περιέχει ένα μεγάλο αριθμό από ελεύθερες βιβλιοθήκες κωδικοποίησης και αποκωδικοποίησης κυρίως από τη βιβλιοθήκη του προγράμματος FFmpeg ενώ παράλληλα χρησιμοποιεί τους δικούς του πολυπλέκτες και αποπολυπλέκτες και τα δικά του πρωτόκολλα.

#### 3.2.3.2 Χαρακτηριστικά του VLC Player

Το VLC Player, ως λογισμικό αναπαραγωγής και μετάδοσης πολυμέσων, είναι βασισμένο σε πακέτα. Μπορεί δηλαδή να αναπαράγει περιεχόμενο ακόμα και σε κατεστραμμένα, ατελή ή ημιτελή αρχεία. Ταυτόχρονα μπορεί να αναπαράγει αρχεία την ώρα που ψηφιοποιούνται από μια συσκευή. Το VLC υποστηρίζει όλες τις μορφές ήχου και βίντεο. Παράλληλα, όπως αναφέρθηκε παραπάνω, μπορεί να διαβάσει αρκετές μορφές ανάλογα με το λειτουργικό σύστημα στο οποίο βρίσκεται. Παρακάτω παρουσιάζονται αναλυτικά τα πρωτόκολλα καθώς και οι τύποι αρχείων ήχου και βίντεο που υποστηρίζει το συγκεκριμένο λογισμικό.

#### Αναγνώσιμες Μορφές:

- **Είσοδος:** UDP/RTP (10) unicast ή multicast, HTTP, FTP, MMS, RTSP, RTMP, DVDs, VCD, SVCD, CD Audio, DVB, Video acquisition (μέσω V4I και DirectShow), RSS/Atom Feeds, και από αρχεία αποθηκευμένα στον υπολογιστή του χρήστη.
- **Μορφές Βίντεο:** Cinepak, Dirac, DV, H.263, H.264/MPEG-4 AVC, HuffYUV, Indeo 3, MJPEG, MPEG-1, MPEG-2, MPEG-4 Part 2, RealVideo 3&4, Sorenson (επιτρέποντας έτσι την απευθείας

αναπαραγωγή του τροποποιημένου Sorenson H.263 για κωδικοποιημένα βίντεο 'κατεβασμένα' από YouTube), Theora, VC-1, VP5, VP6, VP8, και WMV.

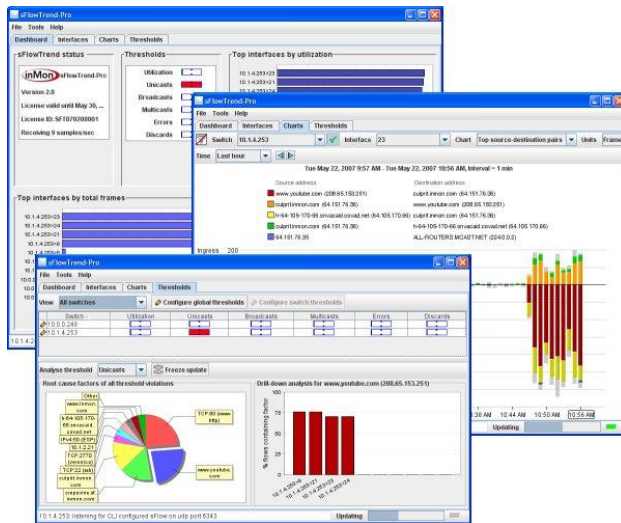
**Μορφές Ήχου:** AAC, AC3, ALAC, AMR, DTS, DV Audio, XM, FLAC, MACE, Mod, MP3, PLS, QDM2/QDMC, RealAudio, Speex, Screamertracker 3/S3M, TTA, Vorbis, WavPack, WMA (WMA 1/2, WMA 3 μερικώς).

### Μορφές Εξόδου για Streaming/Κωδικοποίηση:

- **Πρωτόκολλα streaming:** UDP, HTTP, RTP, RTSP, MMS
- **Μορφές Βίντεο:** H.263, H.264/MPEG-4 AVC, MJPEG, MPEG-1, MPEG-2, MPEG-4 Part 2, VP5, VP6, VP8, Theora, DV
- **Μορφές Ήχου:** AAC, AC3, DV Audio, FLAC, MP3, Speex, Vorbis

## 3.2.4 sFlow Trend (11)

### 3.2.4.1 Επισκόπηση του sFlow Trend



Εικόνα 10: Γραφικό περιβάλλον sFlow Trend

Το sFlow Trend είναι ένα δωρεάν λογισμικό που λειτουργεί ως εργαλείο για την καταγραφή και παρακολούθηση, με γραφική απεικόνιση, της δικτυακής κίνησης. Κάνει χρήση του δημοφιλούς προτύπου sFlow® για τη δημιουργία απεικονίσεων σε πραγματικό χρόνο εκμεταλλευόμενο το bandwidth του δικτύου. Ταυτόχρονα χρησιμοποιεί επεκτάσεις του sFlow προτύπου για την απεικόνιση των επιδόσεων τόσο του Physical Host όσο και ενός server.

Το sFlow Trend αναπτύχθηκε για να αντιμετωπίσει προβλήματα των χρηστών

όπως:

- Ανάλυση και κατανόηση της χρήσης του δικτύου σε επίπεδο χρήστη και εφαρμογής
- Γρήγορη αναγνώριση και επίλυση προβλημάτων ή μη φυσιολογικής δικτυακή κίνησης
- Απεικόνιση σημαντικών παραμέτρων της επίδοσης του Physical Host (π.χ. χρησιμοποίησης κεντρικής μονάδας επεξεργασίας, μνήμης κτλ)
- Δημιουργία αναφορών της παρούσας αλλά και προηγούμενης κατάστασης του συστήματος σε θέμα επιδόσεων

### 3.2.4.2 Αρχιτεκτονική sFlow Trend

Το sFlow Trend όπως αναφέρθηκε, βασίζεται στο διεθνές πρότυπο sFlow<sup>9</sup>, το οποίο δημιουργήθηκε για την παρακολούθηση και απεικόνιση υψηλών ταχυτήτων δικτύων βασιζόμενων σε μεταγωγείς (switches) και δρομολογητές (routers). Ακολούθως θα παρουσιαστεί η βασική δομή και αρχιτεκτονική στου sFlow ενώ παράλληλα αναλύονται οι επιμέρους όροι που το αποτελούν.

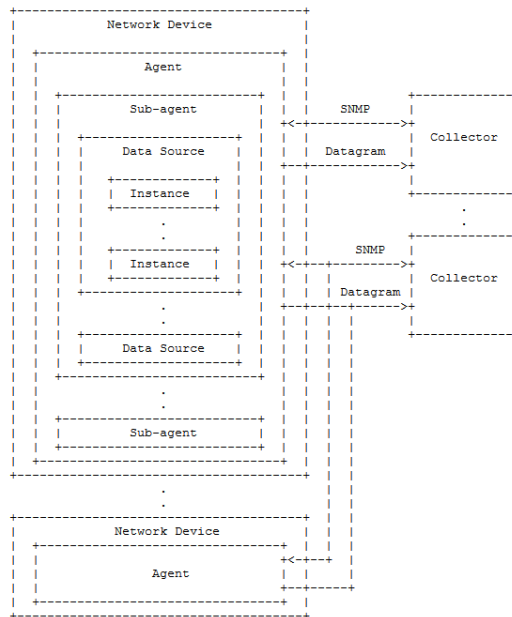
1. **Network Device:** Συσκευή ενός δικτύου η οποία προωθεί πακέτα (switch, router κτλ)
2. **Data Source:** Τοποθεσία μέσα σε μια δικτυακή συσκευή (network device) η οποία έχει τη δυνατότητα μετρήσεων της κίνησης. Τέτοιες θεωρούνται τα interfaces, καθώς και οντότητες όπως τα VLANs.
3. **Packet Flow:** Η διαδρομή ή τροχιά που ακολουθεί ένα πακέτο μέσα στη δικτυακή συσκευή.
4. **Packet Flow Sampling:** Η τυχαία επιλογή ενός μέρους της ροής πακέτων (packet flow) που θεωρείται ως data source
5. **Sampling Rate:** Είναι ο ρυθμός δειγματοληψίας (το κλάσμα) των πακέτων που επιλέχθηκαν στο data source προς τον αριθμό των δειγμάτων που δημιουργήθηκαν.
6. **Packet Flow Record:** Περιγράφει τη συμπεριφορά της διαδρομής του πακέτου (packet flow). Υπάρχουν δύο είδη πληροφορίας:
  - Πληροφορία για το πακέτο αυτό καθ' αυτό, συνήθως η επικεφαλίδα του πακέτου, το μήκος του και η ενθυλάκωση του πακέτου.
  - Πληροφορία για τη διαδρομή που είχε το πακέτο στη συσκευή περιλαμβάνοντας πληροφορίες σχετικές με την επιλογή του συγκεκριμένου μονοπατιού
7. **Counter Sampling:** Περιοδική δειγματοληψία ή τμήματα των μετρητών που συνδέονται με το data source
8. **Sampling Interval:** Το χρονικό διάστημα μεταξύ διαδοχικών δειγμάτων
9. **Counter Record:** Εγγραφή που περιέχει τιμές μετρητών που συνδέονται με το data source στο τέλος ενός διαστήματος δειγματοληψίας.
10. **sFlow Instance:** Διαδικασία μέτρησης η οποία συνδέεται με το data source. Μπορεί να υπάρχουν μία ή περισσότερες διαδικασίες που συνδέονται με ένα data source.
11. **sFlow Agent:** Παρέχει ένα interface για τη ρύθμιση των διαδικασιών του sFlow μέσα σε μία συσκευή. Υποστηρίζει γραμμή εντολών, καθώς και παραμετροποίηση μέσω SNMP. Είναι επίσης υπεύθυνος για τη διατήρηση των συνεδριών κατά την οποία πραγματοποιούνται μετρήσεις μέσω του sflow collector. Ταυτόχρονα, διαχειρίζεται δεδομένα στο sflow datagram με σκοπό να τα στείλει σε sflow collector. Τέλος όταν μία διαδικασία λήξει είναι υπεύθυνος για την απελευθέρωση των δεσμευμένων πόρων του συστήματος.
12. **sFlow Sub-Agent:** Είναι υπεύθυνος για συγκεκριμένο υποσύνολο του data source.

---

<sup>9</sup> <http://en.wikipedia.org/wiki/SFlow>

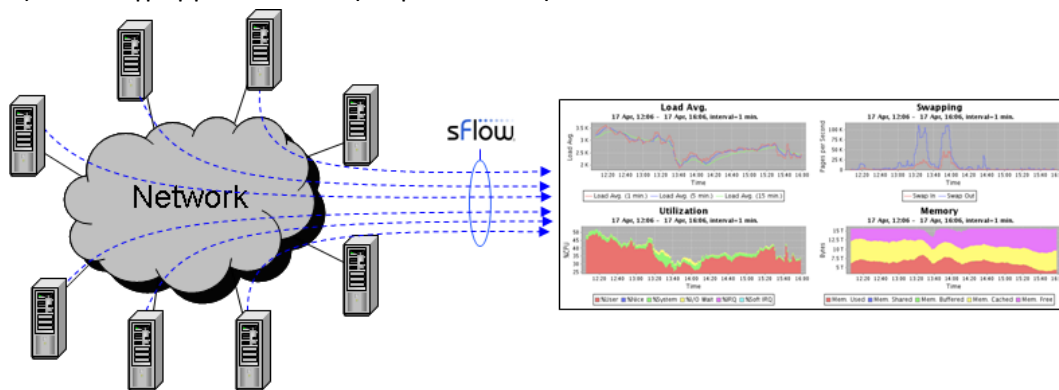
**13. sFlow Datagram:** Είναι ένα σύνολο αυτοδύναμων πακέτων UDP που περιέχει τα δεδομένα των μετρήσεων καθώς και πληροφορίες σχετικά με την πηγή μέτρησης καθώς και με τη διαδικασία.

Στο παρακάτω διάγραμμα παρουσιάζεται η σχέση μεταξύ των διαφορετικών στοιχείων μέσα σε ένα σύστημα sFlow. Αναλύοντας το σχήμα, παρατηρούμε ότι ο sFlow Collector κάνει χρήση του SNMP πρωτοκόλλου για να επικοινωνήσει με τον sFlow Agent ώστε να παραγοντοποιήσει τον τρόπο απεικόνισης της δικτυακής συσκευής. Ο ρυθμός δειγματοληψίας καθώς και ο μετρητής δειγματοληψίας, εκτελείται από διαδικασίες οι οποίες συνδέονται με ξεχωριστά data sources μέσα στον sFlow Agent.



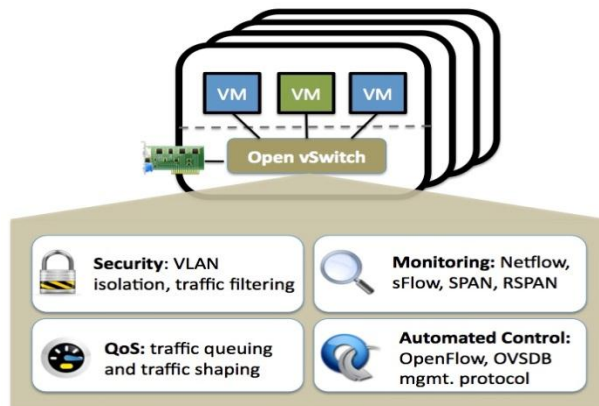
Εικόνα 11: Αρχιτεκτονική sFlow Trend

Κλείνοντας με το sFlow Trend παρατίθεται ένα διάγραμμα το οποίο μας δείχνει την εργασία που πρόκειται να διατελέσει στην πειραματική μας διάταξη κατά την οποία θα συλλέγει δεδομένα και θα τα απεικονίζει σε διαγράμματα κατάλληλα για ανάλυση:



Εικόνα 12: Τρόπος λειτουργίας sFlow Trend

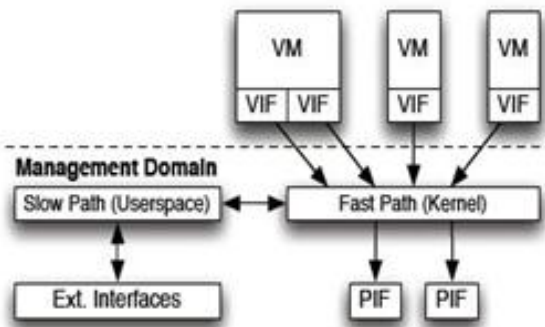
### 3.2.4 Εικονικός Μεταγωγές - Open vSwitch (12)



Εικόνα 13: Open vSwitch

δυνατοτήτων του. Είναι συμβατό με τις περισσότερες linux-based πλατφόρμες συμπεριλαμβανομένων των Xen<sup>10</sup>, XenServer<sup>11</sup>, KVM<sup>12</sup> και QEMU<sup>13</sup>. Ακολούθως θα αναλυθούν και θα παρουσιαστούν η αρχιτεκτονική και η λειτουργία του Open vSwitch.

#### 3.2.4.1 Επισκόπηση του Open vSwitch (13)



Εικόνα 14: Δομή Open vSwitch

Πιο συγκεκριμένα στο fast path βρίσκονται λειτουργίες αναγνώρισης τροποποίησης και προώθησης των πακέτων. Παράλληλα διατηρούνται μετρητές για κάθε νέα εισαγωγή δεδομένων στους πίνακες

Το Open vSwitch είναι ένα ελεύθερο λογισμικό το οποίο δημιουργήθηκε, για να λειτουργεί ως ένας εικονικός μεταγωγέας σε εικονικοποιημένα περιβάλλοντα. Διατέθηκε στο κοινό ως ανοιχτός κώδικας από την εταιρία Nicira networks το Σεπτέμβριο του 2009. Η τεχνολογία αυτή βασίζεται στην αποδοχή ότι η δικτύωση σε virtual networks γίνεται με ένα μοντέλο μεταγωγέα που λειτουργεί σε layer 2. Η διαφορά του από τις υπόλοιπες υλοποιήσεις έγκειται στο γεγονός ότι παρέχει στο χρήστη τη δυνατότητα για πλήρη έλεγχο όλων των λειτουργιών και των

Το Open vSwitch όπως αναφέρθηκε είναι ένα λογισμικό που βρίσκεται μέσα στον επόπτη (hypervisor) και παρέχει τη δυνατότητα επικοινωνίας μεταξύ των εικονικών μηχανών και του physical interface, μέσω των VIF PIF. Υλοποιεί το τυπικό Ethernet switching δηλαδή λειτουργίες VLAN, RSPAN, Basic ACL<sup>14</sup>. Όπως μπορούμε να παρατηρήσουμε από την εικόνα 14 (13) το Open vSwitch βρίσκεται σε δύο μέρη του λειτουργικού συστήματος. Ένα μικρό μέρος του κώδικα τρέχει στο Fast Path (Kernel) ενώ το υπόλοιπο μέρος του κώδικα τρέχει στο Slow Path (Userspace).

<sup>10</sup> <http://xen.org/>

<sup>11</sup> <http://www.citrix.com/English/ps2/products/product.asp?contentID=683148>

<sup>12</sup> [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page)

<sup>13</sup> [http://wiki.qemu.org/Main\\_Page](http://wiki.qemu.org/Main_Page)

<sup>14</sup> [http://en.wikipedia.org/wiki/Access\\_control\\_list](http://en.wikipedia.org/wiki/Access_control_list)

δρομολόγησης. Το μεγαλύτερο μέρος των λειτουργιών του συγκεκριμένου λογισμικού βρίσκονται στο slow path το οποίο έχει υλοποιηθεί στον hypervisor.

Ο λόγος για τον οποίο το fast path έχει ένα μικρό κομμάτι του κώδικα μόνο (περίπου 3000 γραμμές) είναι για να διατηρήσουμε τον πυρήνα του λειτουργικού συστήματος (kernel) απαλλαγμένο από περίπλοκες διαδικασίες με άμεση συνέπεια την αυξημένη απόδοση και ταχύτητα του. Η επιλογή αυτή έγινε επίσης για να είναι εύκολη η ενσωμάτωση του κώδικα σε πολλαπλές πλατφόρμες, τόσο Linux όσο και άλλων λειτουργικών. Αντίθετα ένα μεγάλο μέρος του κώδικα, περί τις 30000 γραμμές -δεκαπλάσιος κώδικας- βρίσκεται στο slow path το οποίο έχει όλες τις δευτερεύουσες λειτουργίες προωθήσεων όπως MAC learning και load-balancing του δικτύου.

Το βασικότερο του πλεονέκτημα του συγκεκριμένου λογισμικού είναι ότι παρέχει στο χρήστη interfaces για τη διαχείριση και τον έλεγχο όλων των λειτουργιών και της κατάστασης του δικτύου κατά τη χρήση του. Τα interfaces αυτά αναλύονται ακολούθως:

1. **Configuration Interface:** Μέσω αυτού του interface ο χρήστης δύναται να ελέγχει τυχόν αλλαγές στην κατάσταση μέσω μιας διαδικασίας (process) η οποία διαβάζει και καταγράφει τις αλλαγές αυτές μέσω ζευγαριών τιμών-κλειδιών. Μέσω τις διαδικασίας αυτής, ο χρήστης μπορεί επίσης να εφαρμόσει πολιτικές ποιότητας της υπηρεσίας (QoS), να ελέγξει τη ροή του δικτύου (NetFlow) και να δεσμεύει ή να αποδεσμεύει interfaces για την βέλτιστη απόδοση και διαθεσιμότητα του δικτύου.
2. **Forwarding Path:** Σε αντίθεση με τους φυσικούς μεταγωγείς, το Open vSwitch παρέχει τη δυνατότητα για παρέμβαση στους πίνακες προώθησης (forwarding tables) και συνεπώς στον προσδιορισμό μιας συγκεκριμένης λειτουργίας των πακέτων (απόρριψη, προώθηση σε συγκεκριμένη πόρτα, ενσωμάτωση) ανάλογα με το είδος της επικεφαλίδας τους (header).
3. **Connectivity management:** Μέσω του συγκεκριμένου interface παρέχεται η δυνατότητα στο χρήστη να διαχειρίζεται την τοπολογία του δικτύου του. Αυτό περιλαμβάνει διαδικασίες δημιουργίας εικονικών μεταγωγέων, διαχείρισης της συνδεσιμότητας των εικονικών interfaces, καθώς και διαχείρισης των φυσικών interfaces του host.

Η γλώσσα προγραμματισμού του συγκεκριμένου λογισμικού είναι C ανεξάρτητη από πλατφόρμα υλοποίησης. Λόγω της φύσης του ανοιχτού λογισμικού, τα χαρακτηριστικά και οι δυνατότητες του προγράμματος είναι δυναμικά και αλλάζουν με τις διαφορετικές εκδόσεις που διατίθενται στο κοινό. Αυτή τη στιγμή βρισκόμαστε στην έκδοση 1.5.0 η οποία υποστηρίζει τις ακόλουθες λειτουργίες:

- Standard 802.1Q<sup>15</sup> VLAN πρωτόκολλο με trunk και access ports
- NIC bonding with or without LACP on upstream switch
- NetFlow, sFlow(R), SPAN, RSPAN, και ERSPAN για αυξημένη δυνατότητα διαχείρισης

---

<sup>15</sup> [http://en.wikipedia.org/wiki/IEEE\\_802.1Q](http://en.wikipedia.org/wiki/IEEE_802.1Q)



- Δυνατότητα προσαρμογής του QoS (Quality of Service) καθώς και των πολιτικών ασφαλείας
- GRE<sup>16</sup>, GRE over IPSEC, και CAPWAP tunneling
- 802.1ag<sup>17</sup> πρωτόκολλο για διαχείριση σε περίπτωση προβληματικών συνδέσεων
- OpenFlow 1.0 (15), λογισμικό για τον έλεγχο των ροών πακέτων του δικτύου
- Βάση δεδομένων με μικρή κατανάλωση πόρων για την καταγραφή των αλλαγών και της υπάρχουσας κατάστασης.
- Επίπεδο συμβατότητας με τον κώδικα του Linux Bridging
- Υψηλή απόδοση στην προώθηση πακέτων με τη χρήση module που τρέχει στον πυρήνα του λειτουργικού (kernel)

Παρακάτω θα αναπτυχθούν μερικοί από τους τρόπους που μπορεί να χρησιμοποιηθεί το Open vSwitch στα εικονικοποιημένα περιβάλλοντα.

1. **Centralized Management:** Το Open vSwitch όπως αναφέρθηκε παρέχει τη δυνατότητα ελέγχου των λειτουργιών του μέσω των interfaces που παρέχει. Ταυτόχρονα είναι εφικτό να δημιουργηθούν εικονικές συνδέσεις μεταξύ των interfaces αυτών σε δύο διαφορετικά φυσικά μηχανήματα. Γίνεται λοιπόν κατανοητό ότι μπορεί να υπάρξει κοινή διαχείριση όλων των μηχανημάτων που διαθέτουν το Open vSwitch από μία κεντρική μονάδα.
2. **Virtual Private Networks:** Όπως ακριβώς στα φυσικά δίκτυα, μία ομάδα από μηχανήματα μπορεί να συνδέονται μεταξύ τους μέσω ενός δεσμευμένου δικτύου, έτσι και στα εικονικοποιημένα περιβάλλοντα οι εικονικές μηχανές μπορούν να δημιουργήσουν ένα εικονικό δεσμευμένο δίκτυο ένα επίπεδο πάνω από το φυσικό επίπεδο. Το Open vSwitch υποστηρίζει λειτουργίες VLAN και GRE tunneling. Πιο συγκεκριμένα το VLAN χρησιμοποιείται για μικρού μεγέθους εγκαταστάσεις δικτύων, το GRE χρησιμοποιείται για την αντιμετώπιση μεγάλου μεγέθους εγκαταστάσεων.
3. **Δυνατότητα μετακίνησης μεταξύ διαφορετικών υποδικτύων:** Ένα από τα προβλήματα στις πλατφόρμες εικονικοποίησης που αντιμετωπίζει το Open vSwitch, είναι η δυνατότητα αντιγραφής και μετακίνησης των ιδιοτήτων των εικονικών δικτύων σε ένα νέο υποδίκτυο χωρίς να χρειάζεται να γίνει στο ίδιο υποδίκτυο όπως γίνεται από τις υπάρχουσες λύσεις.

### 3.2.4.2 Αρχιτεκτονική του Open vSwitch

Το Open vSwitch είναι ένας εικονικός μεταγωγέας που λειτουργεί στο επίπεδο 2 μεταφέροντας πακέτα μεταξύ δύο ή περισσότερων συνδεδεμένων σε αυτό διεπαφών. Το λογισμικό αποτελείται από ένα αντικείμενο που ονομάζεται datapath, το οποίο περιέχει μια λίστα από τις πόρτες που χρησιμοποιούνται από το εικονικό μεταγωγέα, καθώς και έναν πίνακα της ροής (flow table) που συνδέει μια συγκεκριμένη δράση για κάθε ροή. Κάθε πακέτο που διέρχεται από το μεταγωγέα,

<sup>16</sup> [http://en.wikipedia.org/wiki/Generic\\_Routing\\_Encapsulation](http://en.wikipedia.org/wiki/Generic_Routing_Encapsulation)

<sup>17</sup> [http://en.wikipedia.org/wiki/IEEE\\_802.1ag](http://en.wikipedia.org/wiki/IEEE_802.1ag)

συγκρίνεται με τις καταχωρήσεις στον πίνακα της ροής, και στην περίπτωση που αναγνωριστεί εκτελείται η αντίστοιχη δράση. Η αρχιτεκτονική λογισμικού του Open vSwitch διαχωρίζει σαφώς τον ελεγκτή που κάνει τη σύγκριση και τον υπολογισμό στους πίνακες ροής, και του datapath, που κάνει την πραγματική μεταγωγή πακέτων. Τα δύο αυτά κύρια στοιχεία της αρχιτεκτονικής του προγράμματος μπορούν να τρέξουν σε διαφορετικά συστήματα και κατά συνέπεια, το datapath μπορεί να υλοποιηθεί με διάφορες τεχνολογίες, από το λογισμικό στο υλικό.

## ΚΕΦΑΛΑΙΟ 4

### ΑΝΑΛΥΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Στο παρόν κεφάλαιο περιγράφεται αναλυτικά ο τρόπος με τον οποίο σχεδιάστηκε το σύστημα μας, η αρχιτεκτονική του καθώς και τα επιμέρους δομικά συστατικά για κάθε σκέλος της συγκεκριμένης διαδικασίας. Αξίζει εδώ να αναφερθεί, ότι το σύστημα που μελετήθηκε και αναπτύχθηκε χωρίστηκε σε επιμέρους συστήματα για την ευκολότερη παρουσίαση αλλά και τον καλύτερο έλεγχο της απόδοσης σε κάθε στάδιο ξεχωριστά.

#### 4.1 Επισκόπηση του συστήματος

Σκοπός της συγκεκριμένης διπλωματικής εργασίας όπως αναφέρθηκε, είναι ο έλεγχος της απόδοσης της χρήσης του Open vSwitch, ενός λογισμικού το οποίο προσομοιώνει τις λειτουργίες ενός φυσικού μεταγωγέα σε επίπεδο εικονικών δικτύων. Αρχικά εγκαταστάθηκε στον υπολογιστή τα Linux Ubuntu 10.04, καθώς και όλα τα λογισμικά που χρησιμοποιήθηκαν στην εκτέλεση των πειραμάτων, δηλαδή το VirtualBox, το VLC και το sFlow Trend. Εν συνεχεία, δημιουργήθηκαν τέσσερα εικονικά μηχανήματα, με σκοπό τη δημιουργία τοπολογίας δικτύου με δύο VLAN. Παράλληλα, εγκαταστάθηκε στον πυρήνα του Linux το Open vSwitch το οποίο πήρε τη θέση του linux bridge. Μέσω του συγκεκριμένου λογισμικού, δημιουργήθηκαν όλες οι απαραίτητες εικονικές συνδέσεις μεταξύ των εικονικών μηχανών με σκοπό την εγκαθίδρυση δικτυακής σύνδεσης. Έγινε έλεγχος της απόδοσης του συστήματος τόσο με απλή δικτυακή χρήση όσο και με επιπλέον συμφόρηση του δικτύου μέσω του hping3 (13) ενώ κατά τη διάρκεια εκτέλεσης των μετρήσεων λήφθηκε η γραφική απεικόνισή τους μέσω του sFlow Trend. Τέλος έγινε έλεγχος της απόδοσης του Open vSwitch σε περιπτώσεις streaming ήχου και εικόνας σε σχέση με την απλή υλοποίηση δημιουργίας δικτύου που παρέχει το VirtualBox, τόσο σε δίκτυο το οποίο δεν είναι φορτωμένο όσο και σε δίκτυο με μεγάλη συμφόρηση.

Παρακάτω θα αναλυθεί κάθε στάδιο εγκατάστασης ξεχωριστά ώστε να διαχωριστεί η δομή και η αρχιτεκτονική του κάθε στοιχείου του συστήματός μας. Βασικές απαιτήσεις του συστήματός μας ήταν να:

- Είναι επεκτάσιμο, δηλαδή να μπορούν τα αποτελέσματα που θα εξαχθούν να γίνονται διαθέσιμα και σε μεγαλύτερης έκτασης δικτυακές εγκαταστάσεις.
- Είναι γρήγορο και χωρίς απώλειες δεδομένων, καθώς αναζητούμε ένα σύστημα το οποίο να μην έχει μεγάλες απαιτήσεις σε υλικό, ίδιον την εποχής της εικονικοποίησης.
- Διαθέτει όσο το δυνατόν μεγαλύτερη συνδεσιμότητα στο επίπεδο 2, ώστε η υποδομή να είναι πιο απλή αποφεύγοντας με τον τρόπο αυτό τους περιορισμούς στους χρήστες μειώνοντας την πιθανότητα εμφάνισης λάθους.

## 4.2 Στάδιο 1<sup>ο</sup>: Εγκατάσταση λειτουργικού συστήματος και απαραίτητων λογισμικών.

Στο 1<sup>ο</sup> στάδιο δημιουργίας του συστήματος εγκαταστάθηκαν με επιτυχία τόσο το λειτουργικό σύστημα Ubuntu 10.04 όσο και τα απαραίτητα προγράμματα. Η συγκεκριμένη έκδοση του Linux χρησιμοποιήθηκε διότι λειτουργεί απροβλημάτιστα τόσο με το προς αξιολόγηση Open vSwitch όσο και με την πλατφόρμα εικονικοποίησης μας, το VirtualBox. Στην παρούσα διπλωματική εργασία δε θα διερευνήσουμε το θέμα εγκατάστασης του λειτουργικού συστήματος, καθώς αφορά διαφορετικό γνωστικό αντικείμενο και είναι μια διαδικασία η οποία δεν απαιτεί ιδιαίτερες γνώσεις από ένα χρήστη.

Επόμενο βήμα μετά την εγκατάσταση του λειτουργικού συστήματος, ήταν η εγκατάσταση των προγραμμάτων που χρησιμοποιήθηκαν στη συγκεκριμένη διπλωματική εργασία.

Αρχικά εγκαταστάθηκε το VLC player, λογισμικό που χρησιμοποιήθηκε κυρίως για τον έλεγχο της απόδοσης του εικονικού μεταγωγέα σε περιπτώσεις streaming ήχου και εικόνας. Η εγκατάστασή του έγινε από τη γραμμή εντολών του λειτουργικού συστήματος ενώ το γραφικό του περιβάλλον και οι επιλογές που είναι διαθέσιμες για το streaming που μας ενδιαφέρει, παρουσιάζονται ακολούθως. Κατά την εκτέλεση των μετρήσεων χρησιμοποιήθηκε το UDP πρωτόκολλο για τη μετάδοση της πληροφορίας.

Επόμενο πρόγραμμα που εγκαταστάθηκε ήταν το VirtualBox. Σε πρώτη φάση για τη δημιουργία των εικονικών μηχανών και την ανάθεση των πόρων τους, χρησιμοποιήθηκε το γραφικό περιβάλλον της εφαρμογής για μεγαλύτερη ευκολία το οποίο παρουσιάζεται παρακάτω. Δημιουργήθηκαν τέσσερις εικονικές μηχανές, σε κάθε μία από τις οποίες ανατέθηκαν οι ίδιοι υπολογιστικοί πόροι σε επεξεργαστική ισχύ, μνήμη και χωρητικότητα αποθήκευσης. Ταυτόχρονα ξεχωριστά για κάθε εικονική μηχανή αντιστοιχήθηκε μία “πύρτα” (port) για τον έλεγχο της μέσω SSH από τον Host, ώστε να αποφεύγεται η απαιτητική φόρτωση σε γραφικό περιβάλλον του λειτουργικού συστήματος.

Σε κάθε εικονική μηχανή εγκαταστάθηκε λειτουργικό σύστημα Ubuntu 10.04 όμοιο με αυτό του Host για λόγους ευκολίας, ενώ παράλληλα εγκαταστάθηκε εντός των εικονικών μηχανών το VLC player. Οι απαιτήσεις σε υλικό δεν είναι μεγάλες καθώς δε θα χρησιμοποιηθούν για πολύπλοκες εφαρμογές και διαδικασίες και μας ενδιαφέρει μόνο η δικτυακή απόδοση της τοπολογίας.

## 4.3 Στάδιο 2<sup>ο</sup>: Εγκατάσταση του εικονικού μεταγωγέα (Open vSwitch)

Όπως έχει αναφερθεί αρκετές φορές, το Ubuntu Linux 10.04 διαθέτει μία δική του υλοποίηση για μεταγωγέα επιπέδου 2, το οποίο μπορεί να λειτουργήσει για τη σύνδεση εικονικών μηχανών σε εικονικοποιημένα περιβάλλοντα. Αυτό ονομάζεται Linux Bridge και στη συγκεκριμένη διπλωματική εργασία, θα αντικατασταθεί από το προς έρευνα λογισμικό Open vSwitch. Το Open vSwitch, υλοποιήθηκε για εικονικές διατάξεις αρκετών εικονικών μηχανών, διατάξεις στις οποίες το linux bridge δεν αποδίδει αρκετά καλά. Αντίθετα το Open vSwitch έχει ορισμένα σχεδιαστικά πλεονεκτήματα τα οποία το καθιστούν ανώτερο από την υλοποίηση του Linux και καταγράφονται παρακάτω:

1. Ευκολία μετακίνησης της δικτυακής κατάστασης: Το Open vSwitch παρέχει τη δυνατότητα για μεταφορά όλων των ρυθμίσεων καθώς και το στιγμιότυπο του δικτύου μιας δεδομένης χρονικής στιγμής σε διαφορετικούς hosts. Με τη δυνατότητα αυτή ο χρήστης μπορεί να μεταφέρει για παράδειγμα ένα εικονικό μηχάνημα σε διαφορετικό end-host και μαζί του να μεταφέρονται όλες οι ρυθμίσεις και οι τρέχουσες δικτυακές τοπολογίες (QoS, SPAN rules, ACLs, VLANs). Παράλληλα η κατάσταση του Open vSwitch, καταγράφεται και αποθηκεύεται από ένα πραγματικό μοντέλο δεδομένων επιτρέποντας την ανάπτυξη δομημένων συστημάτων αυτοματισμού.
2. Ανταπόκριση σε δυναμικά δίκτυα: Τα εικονικά περιβάλλοντα, χαρακτηρίζονται από μεγάλο βαθμό αλλαγών. Καθώς τα εικονικά μηχανήματα είναι δυναμικά και όχι σταθεροί υπολογιστές, με άμεσο αποτέλεσμα τη διαρκή αλλαγή τους ανάλογα με τις επιθυμίες του χρήστη, το περιβάλλον στο οποίο βρίσκονται αλλάζει διαρκώς. Το Open vSwitch, υποστηρίζει ένα μεγάλο αριθμό χαρακτηριστικών, που επιτρέπουν σε ένα αυτόματο σύστημα ελέγχου να ανταποκρίνεται και να προσαρμόζεται σε αυτές τις αλλαγές. Για το λόγο αυτό το πρόγραμμα διατηρεί μια βάση δεδομένων της κατάστασης του δικτύου, η οποία υποστηρίζει απομακρυσμένη ενεργοποίηση γεγονότων (remote triggering). Ένα πρόγραμμα συνεπώς παρακολουθεί διαρκώς το δίκτυο και δρα ανάλογα με τις αλλαγές, τεχνική η οποία χρησιμοποιείται κυρίως κατά την μετακίνηση των εικονικών μηχανών. Τέλος υποστηρίζει το OpenFlow πρωτόκολλο ως μέθοδο απομακρυσμένου ελέγχου.
3. Τήρηση λογικών ετικετών (logical tags): Αυτή τη στιγμή στην αγορά υπάρχουν προϊόντα τα οποία υποστηρίζουν τη λειτουργία αυτή (Cisco Nexus 1000V, VMware) διατηρώντας ένα λογικό επίπεδο στο δίκτυο ή διαχειρίζονται ετικέτες στα πακέτα του δικτύου. Η τεχνική αυτή μπορεί να χρησιμοποιηθεί για τον μοναδικό προσδιορισμό ενός εικονικού μηχανήματος ή για τη διατήρηση λογικών πλαισίων σχετικών με το λογικό επίπεδο. Στο Openvswitch η λειτουργία αυτή ενσωματώθηκε στον κώδικα του λογισμικού, δηλαδή μέθοδοι για τον προσδιορισμό και τη διατήρηση κανόνων tagging. Αυτοί οι κανόνες αποθηκεύονται σε μια βελτιστοποιημένη μορφή, οπότε οι μεταφορά τους και η τροποποίησή τους γίνεται ευκολότερα σε αντίθεση με την αποθήκευσή τους σε κάποια δικτυακή συσκευή.
4. Ενσωμάτωση του υλικού (Hardware integration): Η διαδρομή προώθησης του Open vSwitch, έχει σχεδιαστεί ώστε να μοιράζει τη διαχείριση των πακέτων στο υλικό, είτε αυτό αφορά κάποιο switch είτε κάποια κάρτα δικτύου. Αυτό έχει σαν αποτέλεσμα, η διαδρομή ελέγχου να μπορεί να ελέγξει τόσο μια υλοποίηση μεταγωγέα σε λογισμικό όσο και μία υλοποίηση μεταγωγέα με υλικό. Η ενσωμάτωση του υλικού δεν αντανακλά μονάχα στην επίδοση στο εικονικοποιημένο περιβάλλον, αλλά επιτρέπει τη δημιουργία ενός αυτοματοποιημένου συστήματος το οποίο ελέγχει τόσο το εικονικό όσο και το φυσικό περιβάλλον.

Παρατηρούμε λοιπόν, ότι το Open vSwitch, ανταποκρίνεται και ταυτόχρονα απευθύνεται σε χρήστες με διαφορετικές ανάγκες από τις λειτουργίες που επιτελούν το Linux Bridge και το Linux networking stack, δίνοντας έμφαση στην ανάγκη για αυτοματοποιημένο και δυναμικό έλεγχο σε μεγάλης έκτασης και όχι μόνο εικονικοποιημένα περιβάλλοντα. Τέλος στόχος του Open vSwitch είναι να κρατήσει το κομμάτι του κώδικα που βρίσκεται στον πυρήνα του λειτουργικού συστήματος όσο μικρότερο γίνεται, και να

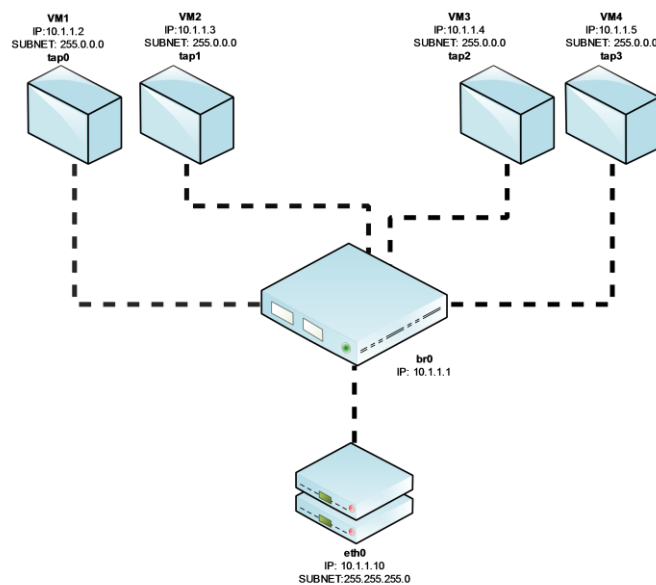
επαναχρησιμοποιεί υπάρχοντα υποσυστήματα όταν αυτά είναι διαθέσιμα (για παράδειγμα το Open vSwitch χρησιμοποιεί το υπάρχον στο Linux, QoS stack).

#### 4.4 Στάδιο 3<sup>ο</sup>: Σχεδιασμός και Λειτουργικότητα Τοπολογίας Δικτύου

Σειρά μετά την εγκατάσταση του Open vSwitch, έχει η δημιουργία της τοπολογίας δικτύου που είναι και το αντικείμενο διερεύνησης της εργασίας. Δημιουργήθηκαν τέσσερις εικονικές μηχανές όπου ανά δύο συνδυάστηκαν για τη δημιουργία δύο VLANs. Σκοπός της συγκεκριμένης τοπολογίας δικτύου είναι να παρέχουμε την απαραίτητη απομόνωση στα δύο υποδίκτυα μας, επιλέγοντας για το κάθε ένα διαφορετικές ρυθμίσεις στον τρόπο με τον οποίο στέλνεται και λαμβάνεται η πληροφορία. Παράλληλα παρέχεται η απαραίτητη ασφάλεια ενώ εμφανίζεται η επιλογή αυτή σε εικονικοποιημένα περιβάλλοντα κάνοντας ένα μεγάλο βήμα για τη σχεδίαση και την ανάπτυξη πολύπλοκων δικτυακών τοπολογιών.

Για την επίτευξη της συγκεκριμένης τοπολογίας απαιτήθηκε η δημιουργία τεσσάρων εικονικών interfaces τα οποία επιλέχθηκαν από κάθε εικονική μηχανή σε bridged networking, λειτουργία που αναπτύξαμε σε προηγούμενο κεφάλαιο, από το VirtualBox. Παράλληλα με χρήση του Open vSwitch δημιουργήσαμε μια γέφυρα (bridge) πάνω στην οποία συνδέθηκαν όλα τα interfaces των εικονικών μηχανών, που αποτελούν μέρος του δικτύου. στόχος αυτής της απλής τοπολογίας ήταν η επικοινωνία μεταξύ των εικονικών μηχανών αλλά και ο έλεγχος της δημιουργίας VLAN, που στην ουσία δημιουργεί δύο ανεξάρτητα δίκτυα, παρ' ότι συνδέονται στην ίδια γέφυρα.

Παρακάτω παρουσιάζεται η δομή της τοπολογίας δικτύου:



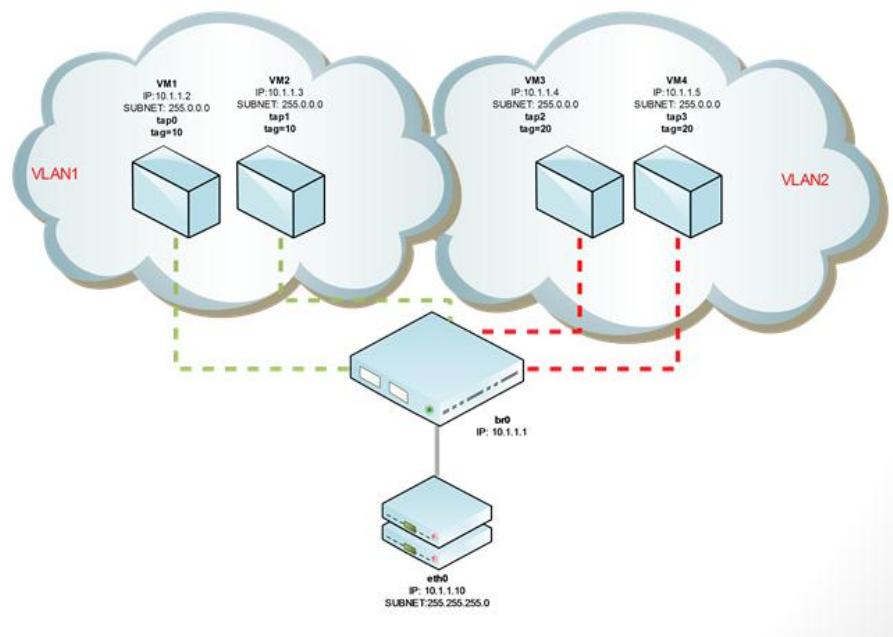
Εικόνα 15: Απλή τοπολογία δίκτυο με τέσσερα εικονικά μηχανήματα και τον host συνδεδεμένα στον εικονικό μεταγωγέα br0

Από το παραπάνω σχεδιάγραμμα και με διαδικασίες που θα αναφερθούν αναλυτικά στο επόμενο κεφάλαιο που ασχολείται με την υλοποίηση, δημιουργήσαμε ένα VLAN ανάμεσα στο VM1-VM2 και ένα δεύτερο VLAN ανάμεσα στο VM3-VM4.

Η επιλογή για χρήση VLAN έγινε διότι η τεχνολογία αυτή αποτελεί σήμερα μία από τις βασικότερες λειτουργίες των μεταγωγέων (switches). Είναι ο κυρίαρχος τρόπος με τον οποίο δημιουργούνται επιπέδου 2 broadcast domains από network segments που συνδέονται σε ένα μεταγωγέα ενώ ταυτόχρονα εξασφαλίζουν ένα σημαντικό επίπεδο ασφαλείας στα υποδίκτυα που δημιουργούνται. Ταυτόχρονα παρέχεται απομόνωση μεταξύ των δύο διαφορετικών VLAN καθώς η χρήση tagging που παρέχεται από το Open vSwitch διαχωρίζει πλήρως τα δύο αυτά υποδίκτυα.

Οι δοκιμές σε αυτό το σημείο θα εστιαστούν στην επιτυχή επικοινωνία των εικονικών μηχανών, καθώς και στην επίδοση του συστήματος σε περιπτώσεις συμφόρησης του δικτύου η οποία δημιουργήθηκε από το λογισμικό hring3. Και στις δύο περιπτώσεις, δηλαδή τόσο στην απλή επικοινωνία μεταξύ των εικονικών μηχανών αλλά και στην περίπτωση όπου υπάρχει συμφόρηση στο δίκτυο, λαμβάνονται μετρήσεις, οι οποίες αναπαρίστανται γραφικά από το λογισμικό sFlow Trend.

Ακολούθως παρουσιάζεται σχηματικά το δίκτυο και οι δικτυακές δοκιμές που αναφέρθηκαν.



Εικόνα 16: Τοπολογία VLAN με συμφόρηση

- Η πράσινη ροή είναι η πληροφορία από το VM1 στο VM2 και αντίστροφα
- Η κόκκινη ροή είναι η πληροφορία από το VM3 στο VM4 και αντίστροφα
- Η γκρι ροή είναι η δημιουργία συμφόρησης από τον host και η ενσωμάτωσή της στο δίκτυο

## 4.5 Στάδιο 4<sup>ο</sup>: Απόδοση συστήματος σε streaming ήχου και εικόνας

Στο στάδιο αυτό του σχεδιασμού του συστήματος, αποφασίστηκε να δοκιμασθεί η επίδοση του Open vSwitch σε περιπτώσεις streaming από τη μία εικονική μηχανή στην άλλη. Οι απαιτήσεις της εποχής μας με τις γρήγορες ταχύτητες, απαιτούν την όσο το δυνατόν καλύτερη απόδοση σε αυτό το σημαντικό κομμάτι της επικοινωνίας, ενώ ο συνδυασμός του με περιβάλλοντα εικονικοποίησης τα οποία θα αποτελούν τη βάση των μελλοντικών δικτύων, καθιστούν τη διερεύνηση τους απαραίτητη.

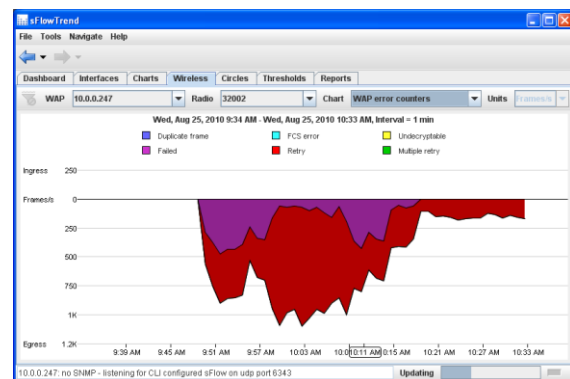
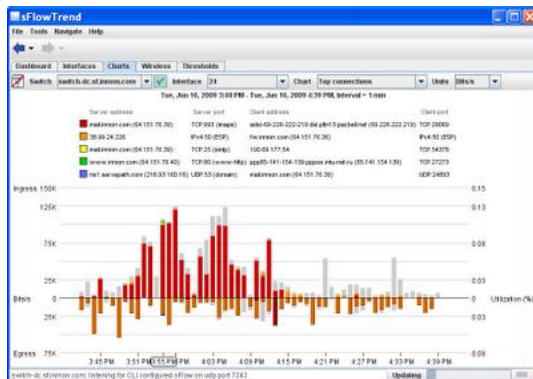
Για την αξιολόγηση των επιδόσεων του εικονικού μεταγωγέα, εφαρμόστηκε και πάλι η ίδια τοπολογία δικτύου με τα 4 εικονικά μηχανήματα και τα 2 VLANs. Ταυτόχρονα εγκαταστάθηκαν όλα τα απαραίτητα εργαλεία σε κάθε εικονική μηχανή, δηλαδή το λογισμικό streaming καθώς και οι απαραίτητες βιβλιοθήκες κωδικοποίησης των αρχείων που μεταδίδονται. Οι δοκιμές που έγιναν αφορούσαν στην επίδοση τόσο με δημιουργία επιπλέον δικτυακής κίνησης όσο και χωρίς συμφόρηση.

Για την κατανόηση της συγκεκριμένης τοπολογίας, δίνονται ακολούθως δύο διαγράμματα που δείχνουν την τοπολογία και τη ροή της πληροφορίας στις δύο περιπτώσεις, μία με συμφόρηση γραμμής και μία υπό φυσιολογικές συνθήκες.

## 4.6 Στάδιο 5ο: Απεικόνιση μετρήσεων με το sFlow Trend

Κατά την δημιουργία και την εκτέλεση των σταδίων του σχεδιασμού του συστήματος, λήφθηκαν στιγμιότυπα που αφορούσαν στη δικτυακή κίνηση. Η απεικόνιση τους έγινε με το λογισμικό sFlow Trend και δείχνει τα ποσοτικά αποτελέσματα που αφορούν το bandwidth του δικτύου τη χρησιμοποίηση του δικτύου καθώς και στατιστικά που αφορούν τα πακέτα, τα πρωτόκολλα και τους προορισμούς αυτών μέσα στο δίκτυο.

Ακολούθως δίνονται μερικά παραδείγματα απεικόνισης του sFlow Trend:



Εικόνα 17: Παραδείγματα απεικόνισης γραφημάτων στο sFlow Trend



## ΚΕΦΑΛΑΙΟ 5

### ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Στο κεφάλαιο αυτό θα παρουσιαστεί αναλυτικά το κάθε στάδιο υλοποίησης του συστήματος που σχεδιάστηκε στο προηγούμενο κεφάλαιο. Ταυτόχρονα θα παρουσιαστούν αναλυτικά όλα τα αποτελέσματα των μετρήσεων τόσο ποσοτικά, όσο και διαγραμματικά.

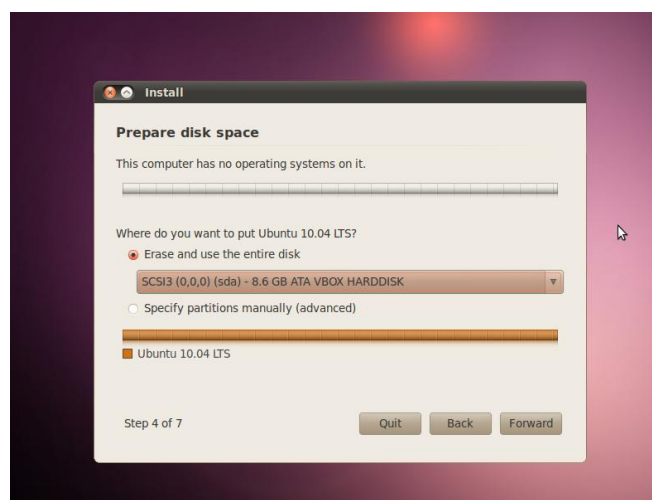
#### 5.1 Υλοποίηση πρώτου σταδίου. Εγκατάσταση λειτουργικού συστήματος και λογισμικών που θα χρησιμοποιηθούν

Καθώς η υλοποίηση του συγκεκριμένου σταδίου είναι μία διαδικασία η οποία περιγράφεται αναλυτικά σε πολλούς οδηγούς του διαδικτύου και δεν απαιτεί μεγάλη εξειδίκευση στην γνώση δε θα δοθεί μεγάλη έκταση στον τρόπο με τον οποίο έγινε.

Αξίζει να τονιστεί ότι έγινε χρήση όλων των πόρων του διαθέσιμου υλικού του υπολογιστή που χρησιμοποιήθηκε. Ο πυρήνας του λειτουργικού συστήματος έχει έκδοση 2.6.32-39, το λογισμικό streaming, δηλαδή ο VLC player είχε έκδοση 1.0.6 και τέλος το sFlow Trend με έκδοση 4.4.03.

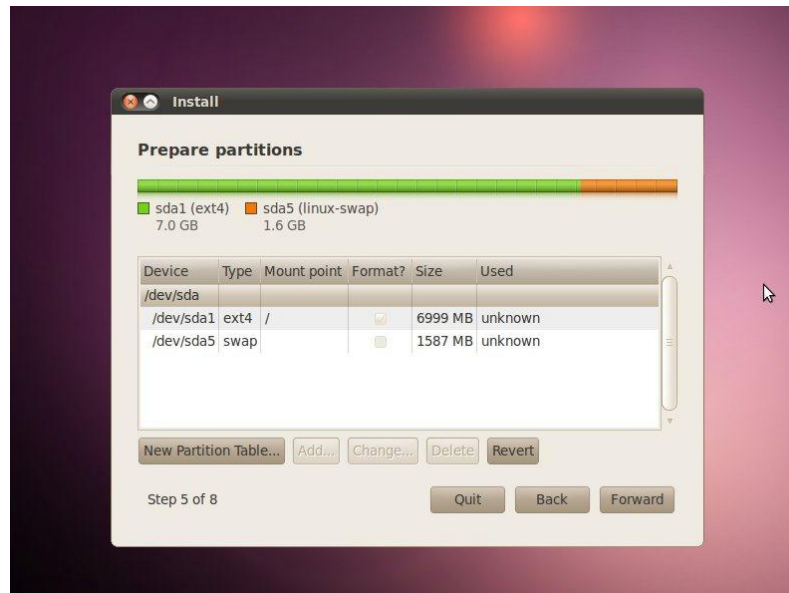
Παρακάτω παρουσιάζεται η διαδικασία σε λίγα βήματα εγκατάστασης του λειτουργικού συστήματος από στιγμιότυπα που λήφθηκαν κατά τη διάρκεια της εγκατάστασης. Παρατίθενται το σημαντικότερο κομμάτι της εγκατάστασης που αφορά την δημιουργία partition και τη μορφή τους.

- Ορισμός του σκληρού δίσκου που θα χρησιμοποιηθεί για την εγκατάσταση. Όπως αναφέρθηκε χρησιμοποιήθηκε ολόκληρος ο σκληρός δίσκος οπότε η πρώτη επιλογή είναι αυτή που χρησιμοποιήθηκε.



Εικόνα 18: Στιγμιότυπο εγκατάστασης Ubuntu

- Δημιουργία δύο partition, ένα που έχει το επου έχει τύπο ext4 και ένα με τύπο swap. Το partition swap αναφέρεται στην εικονική μνήμη του συστήματος, ενώ το ext4 είναι ο κύριος αποθηκευτικός χώρος στον οποίο βρίσκονται και τα αρχεία του συστήματος.



Εικόνα 19: Ορισμός τύπου των διαμορφωμένων τομέων

Με την ολοκλήρωση της εγκατάστασης και αφού γίνει η πρώτη σύνδεση στο νέο λειτουργικό σύστημα, προχωρήσαμε στην επαλήθευση ότι όλα τα επιμέρους υλικά του συστήματος έχουν αναγνωριστεί από το σύστημα και λειτουργούν σωστά (κάρτες δικτύου, σύνολο μνήμης). Εν συνεχεία από τη γραμμή εντολών εγκαταστάθηκε το VLC player.

Παρακάτω παρουσιάζεται η εντολή που δόθηκε καθώς και ένα στιγμιότυπο της εγκατάστασης.

Τόσο το sFlow Trend όσο και το VirtualBox λήφθηκαν από τις αντίστοιχες ιστοσελίδες που διατίθενται και εγκαταστάθηκαν ως πακέτα από το πρόγραμμα διαχείρισης εγκατάστασης προγραμμάτων του λειτουργικού συστήματος. Δύο στιγμιότυπα των συγκεκριμένων εγκαταστάσεων δίνονται ακολούθως.

## 5.2 Υλοποίηση δεύτερου σταδίου. Εγκατάσταση εικονικού μεταγωγέα στη θέση του linux bridge.

Το συγκεκριμένο στάδιο εγκατάστασης είναι ίσως και το πιο σημαντικό σημείο της συγκεκριμένης διπλωματικής εργασίας καθώς πρόκειται να αντικαταστήσει το προκαθορισμένο τρόπο με τον οποίο διαχειρίζεται τα δίκτυα σε εικονικοποιημένα περιβάλλοντα ο πυρήνας του λειτουργικού συστήματος.

Για να κάνουμε compile τα προγράμματα στο userspace είναι απαραίτητο να υπάρχουν ορισμένα προεγκατεστημένα προγράμματα. Αυτά αναφέρονται παρακάτω:

- Ένα make πρόγραμμα. Για παράδειγμα το GNU make.
- Ένα C compiler. Στην περίπτωση μας είναι εγκατεστημένος με το λειτουργικό σύστημα ο 4.4
- Ένα pkg-config πρόγραμμα, το οποίο υπάρχει επίσης προεγκατεστημένο με το λειτουργικό σύστημα.

Παράλληλα για το compile του module στον πυρήνα απαιτείται έκδοση Linux kernel η οποία να υποστηρίζεται (στη δική μας περίπτωση η έκδοση 2.6.32-39 υποστηρίζεται πλήρως), ενώ είναι απαραίτητο να υπάρχει υποστήριξη bridging από τον πυρήνα. Ταυτόχρονα θα πρέπει το bridge module να μην χρησιμοποιείται από τον πυρήνα.

Αφού έγινε έλεγχος ότι όλα τα απαραίτητα προγράμματα υπάρχουν προχωρήσαμε στην εγκατάσταση του Open vSwitch. Η διαδικασία δίνεται αναλυτικά παρακάτω. Μόνο για τη συγκεκριμένη εγκατάσταση δε χρησιμοποιήθηκε κάποιο script καθώς χρειαζόταν έλεγχος ότι όλα εγκαθίστανται σωστά:

1. Αρχικά έγινε λήψη των αρχείων εγκατάστασης από την ιστοσελίδα του προγράμματος και ο φάκελος τοποθετήθηκε στην επιφάνεια εργασίας.
2. Από τη γραμμή εντολών και έχοντας πάντα δικαιώματα διαχειριστή (sudo -i) πλοηγηθήκαμε μέχρι το φάκελο με τα αρχεία και στη συνέχεια έγινε το configure του πακέτου:

Εντολές
<ul style="list-style-type: none"><li>• cd /home/npc/Desktop/Open vSwitch3.2.1</li><li>• ./configure</li></ul>

3. Στον αρχικό φάκελο εκτελούμε την εντολή make για να γίνουν τα απαραίτητα recompilation.

Εντολές
<ul style="list-style-type: none"><li>• cd /home/npc/Desktop/Open vSwitch3.2.1</li><li>• make</li></ul>

4. Στη συνέχεια εκτελούμε την παρακάτω εντολή, για να εγκατασταθούν τα εκτελέσιμα αρχεία καθώς και οι οδηγοί χρήσης του προγράμματος στο λειτουργικό σύστημα. Η προκαθορισμένη διαδρομή που εγκαθίστανται είναι η /usr/local.

Εντολές
<ul style="list-style-type: none"><li>• make install</li></ul>

5. Στο σημείο αυτό έχουν δημιουργηθεί τα module του πυρήνα και σκοπός μας είναι να τα φορτώσουμε στον πυρήνα. Σε αυτό το σημείο παρατηρήθηκε ένα πρόβλημα καθώς κάθε προσπάθεια να φορτωθεί το module οδηγούσε σε αποτυχία. Ύστερα από έρευνα

παρατηρήθηκε ότι υπεύθυνο ήταν το bridge module που προϋπάρχει στον πυρήνα του λειτουργικού μας. Για το λόγο αυτό πρώτα αφαιρέθηκε το συγκεκριμένο module που δημιουργούσε το πρόβλημα και στη συνέχεια φορτώθηκε το module του Open vSwitch.

Εντολές
<ul style="list-style-type: none"><li>• <code>/sbin/rmmod bridge</code> #για την αφαίρεση του bridge module</li><li>• <code>insmod datapath/linux/openvswitch.ko</code> #για την προσθήκη του OVS module</li></ul>

6. Στο βήμα αυτό διαμορφώθηκε η βάση δεδομένων ώστε με τη δημιουργία του φακέλου που θα βρίσκονται τα αρχεία της καθώς και με το αρχείο που θα φυλάει τις ρυθμίσεις που θα γίνονται

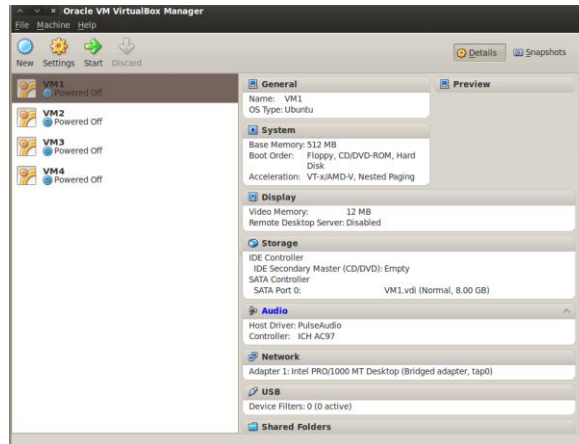
Εντολές
<ul style="list-style-type: none"><li>• <code>mkdir -p /usr/local/etc/Open vSwitch</code></li><li>• <code>ovsdb-tool create /usr/local/etc/Open vSwitch/conf.db vswitchd/vswitch.ovsschema</code></li></ul>

Στο σημείο αυτό ολοκληρώθηκε το στάδιο εγκατάστασης του προγράμματος Open vSwitch. Πριν αρχίσουμε να λειτουργούμε για πρώτη φορά το OVS θα πρέπει να αναφέρουμε ότι σε κάθε μηχανήμα που υπάρχει το Open vSwitch, δηλαδή ο εικονικός μεταγωγέας, θα πρέπει να υπάρχει και η δική του βάση δεδομένων η οποία δημιουργήθηκε κατά την εγκατάσταση.

### 5.3 Υλοποίηση τρίτου σταδίου. Εγκατάσταση τοπολογίας δικτύου και έλεγχος λειτουργικότητας

Αφού η εγκατάσταση όλων των απαραίτητων λογισμικών έγινε με επιτυχία, επόμενο βήμα είναι η εγκαθίδρυση της τοπολογίας δικτύου που αναφέρθηκε στο κεφάλαιο του σχεδιασμού.

Αρχικά θα δημιουργηθούν οι τέσσερις εικονικές μηχανές από το VirtualBox. Όπως έχει αναφερθεί, αρχικά και για ευκολία χρησιμοποιήθηκε το γραφικό περιβάλλον του προγράμματος. Σε κάθε εικονική μηχανή από τις τέσσερις που δημιουργήθηκαν, εγκαταστάθηκε με επιτυχία λειτουργικό σύστημα αντίστοιχο με αυτό του Host. Καθώς οι εικονικές μηχανές δεν έχουν ουσιαστικές διαφορές τόσο στο ρόλο που θα έχουν στην τοπολογία του δικτύου μας, όσο και στα προγράμματα που θα εγκατασταθούν σε αυτές, όλα τα απαραίτητα λογισμικά και βιβλιοθήκες εγκαταστάθηκαν στην πρώτη εικονική μηχανή, στο VM1. Ακολούθως, με τη δυνατότητα που δίνει το VirtualBox για κλωνοποίηση μιας εικονικής μηχανής, δημιουργήθηκαν άλλες τρεις εικονικές μηχανές με τα ίδια χαρακτηριστικά. Παρακάτω δίνεται ένα στιγμιότυπο του γραφικού περιβάλλοντος και των ιδιοτήτων του VM1, πανομοιότυπο για κάθε ένα από τα υπόλοιπα VM.



Εικόνα 20: Στιγμιότυπο του VirtualBox μετά την εγκατάσταση των εικονικών μηχανών

Στο στάδιο αυτό έχουν εγκατασταθεί όλα τα απαραίτητα προγράμματα και συστατικά του δικτύου μας και σειρά έχει η εγκαθίδρυση των μεταξύ τους συνδέσεων.

### 5.3.1 Εγκαθίδρυση συνδέσεων μεταξύ των μερών του συστήματος

Για την αυτοματοποίηση των διαδικασιών δημιουργήθηκαν scripts, τα οποία επιτελούν μια συγκεκριμένη λειτουργία και βοηθούν στην κατανόηση της. Όπως έχει αναφερθεί, σκοπός είναι η δημιουργία μίας τοπολογίας δικτύου στην οποία οι τέσσερις εικονικές μηχανές συνδέονται πάνω στον εικονικό μεταγωγέα μέσω εικονικών διεπαφών (virtual interfaces).

Πρώτο βήμα για τη δημιουργία της τοπολογίας μας είναι η εκκίνηση του daemon Open vSwitch καθώς και η εκκίνηση της βάσης δεδομένων στην οποία αποθηκεύονται όλες οι ρυθμίσεις που γίνονται από τον controller του Open vSwitch. Το script που δημιουργήθηκε παρουσιάζεται ακολούθως.

```
startovs.sh
```

```
#!/bin/bash
```

```
/sbin/rmmod bridge
```

```
cd /home/npc/Desktop/Open vSwitch-1.4.0
```

```
insmod datapath/linux/Open vSwitch_mod.ko
```

```
ovsdb-server --remote=punix:/usr/local/var/run/Open vSwitch/db.sock --pidfile --detach
```

```
ovs-vswitchd --pidfile --detach
```

Η πρώτη εντολή αναφέρεται στο bridge module που τρέχει στον πυρήνα του λειτουργικού καθώς είναι αυτό που θέλουμε να αντικαταστήσουμε με το Open vSwitch.

Η δεύτερη εντολή και τρίτη εντολή, αναφέρεται στην εισαγωγή στον πυρήνα του λειτουργικού του Open vSwitch module υπεύθυνο για τις λειτουργίες του εικονικού μεταγωγέα.

Η τετάρτη εντολή αφορά στην εκκίνηση της βάσης δεδομένων.

Η πέμπτη εντολή αναφέρεται στην εκκίνηση του Open vSwitch daemon.

Μια παρατήρηση που μπορεί να γίνει σε αυτό το σημείο είναι ότι έχουν ενσωματωθεί στις εντολές οι παράμετροι --pidfile και --detach. Το pidfile είναι για τη δημιουργία ενός αρχείου που καταγράφει το PID της διαδικασίας, ενώ το --detach, προκαλεί τη συγκεκριμένη λειτουργία να τρέχει στο παρασκήνιο και όχι στο προσκήνιο. Το daemon τρέχει στο προσκήνιο μόνο όταν συνδεθεί στη βάση δεδομένων.

Δεύτερο βήμα για την τοπολογία, είναι η δημιουργία των εικονικών διεπαφών (virtual interfaces). Έχοντας επιλέξει να διεξάγουμε το πείραμα με το λογισμικό VirtualBox, χρησιμοποιήθηκε η εντολή VBoxTunctl η οποία είναι για τη λειτουργία αυτή. Έχοντας τέσσερις εικονικές μηχανές, θα χρειαστούμε τέσσερις εικονικές διεπαφές. Για το σκοπό αυτό δημιουργήθηκε το ακόλουθο script.

```
vif.sh
#!/bin/bash
VBoxTunctl -b -u hrc
VBoxTunctl -b -u hrc
VBoxTunctl -b -u hrc
VBoxTunctl -b -u hrc
```

Κάθε μία εντολή δημιουργεί ένα virtual interface τα οποία έχουν ονομασία tap0, tap1, tap2, tap3 αντίστοιχα. Για την επιβεβαίωση της επιτυχούς δημιουργίας, είναι απαραίτητος ο έλεγχος με την εντολή ifconfig. Στην εντολή η παράμετρος -b εμφανίζει το όνομα του virtual interface μετά την εκτέλεση της, ενώ η παράμετρος -u ακολουθείται πάντα από το όνομα χρήστη που είναι συνδεδεμένος στο user-space του λειτουργικού συστήματος.

Τρίτο βήμα για το οποίο δημιουργήθηκε ένα ακόμα script, είναι ο ορισμός IP διευθύνσεων στα interfaces που θα χρησιμοποιηθούν, καθώς και η εντολή ifup η οποία θα εμφανίσει τα virtual interfaces, κάνοντάς τα διαθέσιμα για τη συνέχεια του πειράματος. Παρακάτω δίνεται το script που δημιουργήθηκε.

```
ifcommand.sh
#!/bin/bash
ifconfig eth0 10.1.1.10
route del -net 10.0.0.0 netmask 255.0.0.0 dev eth0
ifconfig br0 10.1.1.1
ifconfig tap0 up
ifconfig tap1 up
ifconfig tap2 up
ifconfig tap3 up
```

Στο interface eth0 δίνεται η IP address 10.1.1.10 καθώς δε θα έχουμε σύνδεση με άλλο φυσικό μηχάνημα και συνεπώς η πρόσβαση σε όλα τα εικονικά μηχανήματα δεν είναι απαραίτητα. Αντίθετα θέλουμε να είναι μέρος του δικτύου. Στον εικονικό μεταγωγέα, δίνεται η συγκεκριμένη IP διεύθυνση,

ενώ τέλος με τις τέσσερις τελευταίες εντολές εμφανίζονται τα virtual interfaces στο user-space του linux.

Το τέταρτο και σημαντικότερο βήμα αφορά στη σύνδεση των interfaces στον εικονικό μεταγωγέα και συγκεκριμένα στο Open vSwitch bridge που θα δημιουργηθεί.

#### Addport.sh

```
#!/bin/bash
```

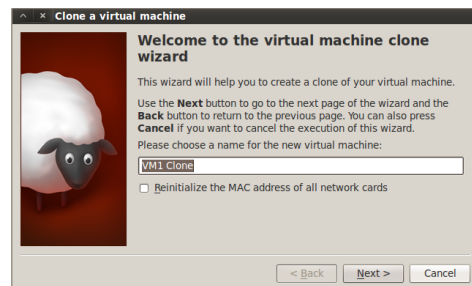
```
ovs-vsctl add-br br0
ovs-vsctl add-port br0 eth0
ovs-vsctl add-port br0 tap0
ovs-vsctl add-port br0 tap1
ovs-vsctl add-port br0 tap2
ovs-vsctl add-port br0 tap3
```

Η εντολή `ovs-vsctl`, η οποία περιγράφεται αναλυτικά μαζί με τις άλλες σημαντικές εντολές του Open vSwitch στο παράρτημα, είναι υπεύθυνη για την παραμετροποίηση του `ovs-vswitchd`, δηλαδή του `daemon` που εκτελεί χρέη εικονικού μεταγωγέα. Ουσιαστικά παρέχει ένα υψηλού επιπέδου interface για την παραμετροποίηση της βάσης δεδομένων του.

Με την πρώτη εντολή δημιουργούμε ένα linux bridge με την ονομασία `br0`. Πάνω σε αυτό το εικονικό bridge θα συνδεθούν όλες οι εικονικές μηχανές. Οι υπόλοιπες τέσσερις εντολές αφορούν στη δημιουργία εικονικών πορτών πάνω στο bridge μέσω των οποίων θα γίνεται η επικοινωνία μεταξύ των συνδεδεμένων μηχανών. Σε αυτές τις πόρτες συνδέονται τόσο εικονικά όσο και φυσικά interfaces. Στη δική μας περίπτωση συνδέθηκαν τα virtual interfaces που δημιουργήσαμε με το προηγούμενο script καθώς και το φυσικό interface `eth0`.

Η δημιουργία των virtual interfaces και η σύνδεσή τους στο linux bridge αφορούν στη δικτυακή σύνδεση των εικονικών μηχανών. Για το λόγο αυτό θα πρέπει σε κάθε εικονική μηχανή να αντιστοιχηθεί το κατάλληλο virtual interface. Από το γραφικό περιβάλλον του VirtualBox μεταβήκαμε στις επιλογές κάθε εικονικής μηχανής και στην κατηγορία Network ενεργοποιήσαμε ένα Network Adapter. Όπως έχει αναφερθεί στην περιγραφή του προγράμματος VirtualBox είχαμε να επιλέξουμε ανάμεσα σε πέντε επιλογές. Αυτή που ταιριάζει στην περίπτωση μας, δηλαδή στην αντιστοίχιση του Network Adapter με ένα εικονικό interface, είναι η επιλογή `bridged adapter`. Για την τοπολογία δικτύου της εργασίας, στο VM1 αντιστοιχήθηκε το `tap0`, στο VM2 το `tap1`, στο VM3 το `tap2` και στο VM4 το `tap3`.

- Στο σημείο αυτό παρουσιάστηκε και το πρώτο πρόβλημα της εγκατάστασης, καθώς κατά την κλωνοποίηση της πρώτης εικονικής μηχανής στην οποία είχε ενεργοποιηθεί ο Network Adapter, δημιουργήθηκαν τρεις εικονικές μηχανές οι οποίες προσομοίωναν το Network Adapter δίνοντας του ακριβώς την ίδια MAC Address. Αυτό είχε ως αποτέλεσμα να μην μπορεί να υπάρχει καμία



Εικόνα 21: Clone machine και αλλαγή MAC address

επικοινωνία μεταξύ των εικονικών μηχανών. Η λύση ήταν η αλλαγή της τυχαίας MAC Address. Λόγω της σύγχυσης που δημιουργήθηκε αποφασίστηκε να σταλεί ως bug του προγράμματος στην ομάδα ανάπτυξής του. Σε επόμενη έκδοση το πρόβλημα διορθώθηκε καθώς κατά την κλωνοποίηση μιας εικονικής μηχανής, δίνεται η επιλογή για αλλαγή της MAC Address όλων των δικτυακών καρτών που είναι ενεργοποιημένες στο αρχικό εικονικό μηχάνημα. (Reinitialize the MAC address of all network cards).

Επόμενο στάδιο είναι ο έλεγχος της επικοινωνίας μεταξύ των εικονικών μηχανών. Για το σκοπό αυτό τέθηκαν σε λειτουργία τα τέσσερα εικονικά μηχανήματα και ανατέθηκαν σε αυτά συγκεκριμένες IP διευθύνσεις. Για λόγους αποφυγής οποιασδήποτε σύγχυσης στην απόδοση IP διευθύνσεων αποφασίστηκε τα virtual machines να έχουν static ip.

Στον ακόλουθο δίνονται οι εντολές για κάθε virtual machine.

#### Απόδοση Static IP Διευθύνσεων στα VM

##### **VM1:**

```
sudo gedit /etc/network/interfaces
auto eth0
iface eth0 inet static
address 10.1.1.2
netmask 255.0.0.0
gateway 10.1.1.1
```

##### **VM2:**

```
sudo gedit /etc/network/interfaces
auto eth0
iface eth0 inet static
address 10.1.1.3
netmask 255.0.0.0
gateway 10.1.1.1
```

##### **VM3:**

```
sudo gedit /etc/network/interfaces
auto eth0
iface eth0 inet static
address 10.1.1.4
netmask 255.0.0.0
gateway 10.1.1.1
```

##### **VM4:**

```
sudo gedit /etc/network/interfaces
auto eth0
iface eth0 inet static
address 10.1.1.5
netmask 255.0.0.0
gateway 10.1.1.1
```



Η διευθύνσεις IP εγγράφηκαν στο αρχείο interfaces ώστε να μην είναι απαραίτητος ο επαναπροσδιορισμός τους κάθε φορά που εκκινούμε το virtual machine αλλά και να έχουμε γνώση για τις κινήσεις των πακέτων (από που προέρχονται και που πηγαίνουν) μέσα στη δικτυακή τοπολογία.

Αρχικά ελέγχθηκε η συνδεσιμότητα των εικονικών μηχανών. Για κάθε ένα virtual machine εκτελέστηκε η εντολή ping με αποδέκτες όλα τα υπόλοιπα. Ως αποτέλεσμα είχαμε μηδενικές απώλειες πακέτων ICMP τόσο μεταξύ των εικονικών μηχανών, όσο και μεταξύ εικονικών μηχανών και Host. Ο μέσος χρόνος δε είναι ελαττωμένος κατά περίπου 10% σε σχέση με τη συνδεσμολογία που παρέχει το VirtualBox για δημιουργία δικτύων μεταξύ των εικονικών μηχανών.

Εν συνεχεία δοκιμάστηκε η απόδοση της ταχύτητας κατά την μεταφορά αρχείων από και προς τον Host καθώς και από και προς τις εικονικές μηχανές. Κατά την εκτέλεση του συγκεκριμένου πειράματος χρησιμοποιήθηκε ένα αρχείο μεγέθους 500 MB το οποίο δοκιμάστηκε για μεταφορά αρχικά από τον Host προς κάθε εικονική μηχανή και στη συνέχεια ταυτόχρονα απ' όλες τις εικονικές μηχανές. Το αποτέλεσμα ήταν ότι κατά την μεταφορά του αρχείου μέσω ssh από κάθε virtual machine ξεχωριστά η ταχύτητα και ο χρόνος ήταν ίδιος για κάθε ένα ξεχωριστά. Αντίθετα κατά την ταυτόχρονη λήψη του αρχείου η ταχύτητα μοιράστηκε στα τέσσερα εικονικά μηχανήματα και ο χρόνος λήψης υποτετραπλασιάστηκε. Συνεπώς μπορεί να συμπεράνει κανείς ότι το Open vSwitch λειτουργεί ιδανικά για περιπτώσεις όπου δεν έχουν οριστεί συγκεκριμένες τιμές για την ποιότητα της υπηρεσίας. Η εντολή που δόθηκε από τα εικονικά μηχανήματα παρατίθεται στην επόμενη εντολή:

#### Εντολή μεταφοράς αρχείου.

- sftp npc@10.1.1.1
- get /media/250Gb/Movies/1.mp4 /home/vm1/Desktop/1.mp4

#### 5.3.1.1 Ενσωμάτωση επιπλέον κίνησης πακέτων στην απλή τοπολογία δικτύου και τροποποίηση της ποιότητας της υπηρεσίας (QoS)

Αφού η συνδεσμολογία των εικονικών μηχανών ήταν επιτυχής, σειρά στο πείραμα είχαν οι μετρήσεις που αφορούσαν την απόδοση του Open vSwitch σε καταστάσεις με υπερφορτωμένο δίκτυο. Για το σκοπό αυτό χρησιμοποιήθηκε το πρόγραμμα hping3 το οποίο έχει τη δυνατότητα να συμπεριφέρεται ως γεννήτρια κίνησης πακέτων στο επιθυμούμενο δίκτυο. Αποστολέας της κίνησης είναι ο Host ενώ προορισμός της κίνησης ήταν ολόκληρη η μάσκα υποδικτύου, δηλαδή τα πακέτα ήταν broadcast.

Σε πρώτη φάση αποφασίστηκε να χρησιμοποιηθεί ο μέγιστος ρυθμός γέννησης πακέτων από το λογισμικό hping3 αρχικά με χρήση πρωτοκόλλου TCP και στη συνέχεια μέσω πρωτοκόλλου UDP. Η εντολή που χρησιμοποιήθηκε και στις δύο περιπτώσεις παρατίθεται ακολούθως.

#### Εντολή γεννήτριας πακέτων

- hping3 --udp -- flood 10.1.1.1
- hping3 -- flood 10.1.1.1

Αποτέλεσμα της χρήσης των δύο παραπάνω εντολών τόσο μεμονωμένα όσο και σε συνδυασμό, ήταν η μη λειτουργικότητα του δικτύου. Τόσο η προσπάθεια επικοινωνίας μεταξύ των εικονικών μηχανών όσο

και αυτή μεταξύ του Host και των εικονικών μηχανών αποτύγχανε μερικώς, καθώς υπήρχαν πακέτα τα οποία δε μεταδίδονταν κατά τη χρήση της εντολής ping ενώ η προσπάθεια για λήψη του αρχείου που αναφέρθηκε από τον server δεν ολοκληρωνόταν με επιτυχία τις περισσότερες φορές των δοκιμών. Αυτή βέβαια είναι μία ακραία περίπτωση που η γεννήτρια πακέτων λειτουργεί σε πολύ μεγάλους ρυθμούς και δεν ανταποκρίνεται σε μία πραγματική χρήση δικτύου.

Μια πιο ρεαλιστική προσέγγιση ενός δικτύου με συμφόρηση είναι με χρήση διαφορετικής εντολής από το πρόγραμμα hping3 με την επιλογή κατάλληλων παραμέτρων. Τα πακέτα δημιουργούνται και αποστέλλονται στη δικτυακή τοπολογία με δύο διαφορετικούς ρυθμούς, ένας με γέννηση 100 πακέτων το δευτερόλεπτο και ένας με δημιουργία 10 πακέτων το δευτερόλεπτο. Οι εντολές που δημιουργούν τα πακέτα UDP και TCP αντίστοιχα δίνονται ακολούθως:

#### Εντολή γεννήτριας πακέτων

- hping3 - -udp --fast 10.1.1.1
- hping3 - -faster 10.1.1.1

Σε αυτή την κατάσταση οι εικονικές μηχανές μπορούν να επικοινωνούν μεταξύ τους με 100% επιτυχία στην αποστολή και στη λήψη πακέτων. Παράλληλα η μεταφορά αρχείων γίνεται πάλι με διαμοιρασμένη ισόποση ταχύτητα μεταξύ των virtual machines. Παρατηρούμε δηλαδή ότι σε μία φυσιολογική κίνηση του δικτύου ο εικονικός μεταγωγέας λειτουργεί με ικανοποιητικό τρόπο καθώς δεν παρατηρούνται απώλειες πακέτων και έλλειψη συνδεσιμότητας.

Όσον αφορά στο κομμάτι της ποιότητας της υπηρεσίας (Quality Of Service) το Open vSwitch δίνει τη δυνατότητα παραμετροποίησης του στις εικονικές μηχανές που θα επιλεγούν. Το κομμάτι αυτό είναι από τα βασικότερα στη σχεδίαση ενός δικτύου καθώς είναι απαραίτητος ο διαχωρισμός των εργασιών που θα πρέπει να έχουν προτεραιότητα καθώς και των αντίστοιχων χρηστών. Για παράδειγμα είναι θεμιτό ορισμένες φορές να μη δίνεται μεγάλο μέρος του εύρους ζώνης μιας διαδικτυακής γραμμής σε χρήστες οι οποίοι δεν είναι αξιόπιστοι. Στην περίπτωση του πειράματός μας, θεωρήσαμε ότι οι δύο από τις τέσσερις εικονικές μηχανές, έχουν μεγαλύτερη προτεραιότητα και μέσω του εργαλείου που παρέχει το Open vSwitch τροποποιήθηκαν κατάλληλα ώστε να συμβαδίζουν με τις ανάγκες. Οι εντολές για την τροποποίηση του QoS στην τοπολογία μας δίνονται ακολούθως

### Εντολές Τροποποίησης Quality Of Service (qosimple.sh script)

#### **VM1:**

```
ovs-vsctl set Interface tap0 ingress_policing_rate=1000  
ovs-vsctl set Interface tap0 ingress_policing_burst=100
```

#### **VM2:**

```
ovs-vsctl set Interface tap1 ingress_policing_rate=10000  
ovs-vsctl set Interface tap1 ingress_policing_burst=1000
```

#### **VM3:**

```
ovs-vsctl set Interface tap2 ingress_policing_rate=1000  
ovs-vsctl set Interface tap2 ingress_policing_burst=100
```

#### **VM4:**

```
ovs-vsctl set Interface tap3 ingress_policing_rate=10000  
ovs-vsctl set Interface tap3 ingress_policing_burst=1000
```

Η πρώτη εντολή αναφέρεται στο μέγιστο ρυθμό με τον οποίο ένα εικονικό μηχάνημα επιτρέπεται να αποστέλλει πακέτα μετρημένο σε kbps. Η δεύτερη εντολή αναφέρεται σε μία παράμετρο στο αλγόριθμο του policing που καθορίζει το μέγιστο μέγεθος δεδομένων (σε Kbit) που μπορεί να λάβει το interface όταν ξεπεραστεί ο μέγιστο ρυθμός που έχει καθοριστεί στο ingress\_policing\_rate.

Συνεπώς για τις εικονικές μηχανές VM1-VM3 ο ρυθμός τέθηκε ίσος με 1Mbps ενώ για τις εικονικές μηχανές VM2-VM4 ο ρυθμός τέθηκε ίσος με 10Mbps. Για την ίδια τοπολογία δικτύου και με την συμφόρηση που δημιουργεί το traffic generator hring3 δοκιμάστηκε η μεταφορά αρχείων μέσα στο δίκτυο από όλες τις εικονικές μηχανές. Οι μετρήσεις που εμφανίζονταν κατά τη διάρκεια της μεταφοράς των αρχείων επιβεβαίωσαν με απόλυτη ακρίβεια ότι στα VM1-VM3 ο ρυθμός μεταφοράς δεν ξεπέρασε το 10Mbps ενώ στα VM2-VM4 δεν ξεπέρασε το 1Mbps (για την ακρίβεια ήταν αρκετά χαμηλότερος καθώς υπήρχε και η συμφόρηση του δικτύου).

Συμπερασματικά μπορούμε να πούμε ότι η τροποποίηση της ποιότητας της υπηρεσίας είναι ένα σημαντικό εργαλείο κατά το σχεδιασμό μιας δικτυακής τοπολογίας και το Open vSwitch υποστηρίζει αυτή τη λειτουργία στα εικονικοποιημένα περιβάλλοντα.

### 5.3.2 Δημιουργία τοπολογίας δικτύου με χρήση VLANs.

Ένα από τα σημαντικότερα ίσως χαρακτηριστικά του Open vSwitch, είναι η δημιουργία VLANs. Στην υλοποίηση της συγκεκριμένης εργασίας, χρησιμοποιήθηκαν και πάλι scripts για τη δημιουργία δύο VLAN μεταξύ των virtual machines. Καθώς η προηγούμενη τοπολογία δικτύου υπάρχει ακόμα στη βάση δεδομένων του Open vSwitch στόχος είναι να διαγραφούν οι παλιές εγγραφές και να δημιουργηθούν νέες οι οποίες θα ανταποκρίνονται στη νέα τοπολογία δικτύου.

Καθώς για την αντιστοίχιση των εικονικών μηχανών θα χρησιμοποιηθεί η μέθοδος απόδοσης μιας συγκεκριμένης ετικέτας (tag), θα διαγραφούν οι παλιές πόρτες και τη θέση τους θα πάρουν νέες πόρτες οι οποίες θα φέρουν συγκεκριμένο tag, ανάλογα με το VLAN στο οποίο ανήκουν. Για τη δημιουργία των δύο VLAN χρησιμοποιήθηκαν τα ακόλουθα scripts.

```
delport.sh
```

```
#!/bin/bash
```

```
ovs-vsctl del-port br0 eth0
```

```
ovs-vsctl del-port br0 tap0
```

```
ovs-vsctl del-port br0 tap1
```

```
ovs-vsctl del-port br0 tap2
```

```
ovs-vsctl del-port br0 tap3
```

Με το δεύτερο script ορίζουμε τις πόρτες επικοινωνίας στο bridge και συγκεκριμένα, τέσσερις πόρτες με ίδιο tagging ανά δύο. Το ακόλουθο script παρουσιάζει τις απαραίτητες εντολές για τη δημιουργία των πορτών των VLANs.

```
addportvlan.sh
```

```
#!/bin/bash
```

```
ovs-vsctl add-port br0 tap0 eth0
```

```
ovs-vsctl add-port br0 tap0 tag=10
```

```
ovs-vsctl add-port br0 tap1 tag=10
```

```
ovs-vsctl add-port br0 tap2 tag=20
```

```
ovs-vsctl add-port br0 tap3 tag=20
```

Τα tap interfaces είναι τα ίδια που χρησιμοποιήθηκαν και στην προηγούμενη τοπολογία δικτύου και αντιστοιχούν στις ίδιες εικονικές μηχανές. Με τον τρόπο αυτό έχουν πλέον δημιουργηθεί οι απαραίτητες συνθήκες για την ομαδοποίηση του VM1 και του VM2 στο VLAN1 και του VM2, VM3 στο VLAN2.

Επόμενο βήμα είναι ο έλεγχος της επικοινωνίας των εικονικών μηχανών αλλά και η επιβεβαίωση ότι παρέχεται η κατάλληλη απομόνωση στα 2 VLANs. Δηλαδή ότι δεν είναι δυνατή η επικοινωνία μεταξύ μηχανών που βρίσκονται σε διαφορετικό VLAN. Οι διευθύνσεις που χρησιμοποιήθηκαν για τις εικονικές μηχανές είναι οι ίδιες με αυτές του υποκεφαλαίου 5.3.1 καθώς για λόγους δοκιμών πρέπει

να επιβεβαιωθεί ότι δεν υπάρχει επικοινωνία ακόμα και στην περίπτωση που ανήκουν στο ίδιο υποδίκτυο.

Αρχικά ελέγχθηκε η συνδεσιμότητα των VLANs. Εκτελέστηκαν εντολές ping από το VM1 στο VM2 και αντίστροφα καθώς και εντολές ping από το VM3 στο VM4. Όλα τα πακέτα έφτασαν στους προορισμούς τους με απόλυτη επιτυχία και χωρίς απώλειες, ενώ ο χρόνος τους κρίνεται ικανοποιητικός και αντίστοιχος με της προηγούμενης συνδεσμολογίας.

Σε επόμενο στάδιο ελέγχθηκε η συνδεσιμότητα μεταξύ των 2 VLANs. Εκτελέστηκαν εντολές ping από το VM1 και το VM2 στο VM3 και VM4, καθώς και αντίστροφα. Ως αποτέλεσμα είχαμε ότι κανένα πακέτο δεν έφτασε στον προορισμό του, προορισμός ο οποίος δεν ήταν καν ανιχνεύσιμος σα διεύθυνση από τα μηχανήματα που δε βρίσκονταν στο ίδιο VLAN. Το γεγονός αυτό μας οδηγεί στο ασφαλές συμπέρασμα ότι υπάρχει η ζητούμενη απομόνωση μεταξύ των εικονικών μηχανών λειτουργία η οποία είναι απαραίτητη όχι μόνο στα εικονικοποιημένα περιβάλλοντα αλλά και σε πιο απαιτητικά δικτυακά περιβάλλοντα.

Όπως και στην προηγούμενη απλή τοπολογία, έτσι και στη συγκεκριμένη τοπολογία με VLAN δοκιμάστηκε η μεταφορά αρχείων μεταξύ των εικονικών μηχανών. Στη συγκεκριμένη περίπτωση ρόλο διαμοιραστή του αρχείου, έπαιξαν τα εικονικά μηχανήματα και όχι ο Host. Τα αποτελέσματα ήταν σχεδόν ταυτόσημα με αυτά της προηγούμενης τοπολογίας, καθώς είχαμε απόλυτη επιτυχία στη μεταφορά του αρχείου είτε γινόταν ταυτόχρονα και στα δύο VLAN είτε γινόταν αποκλειστικά στο ένα μόνο VLAN.

### 5.3.2.1 Ενσωμάτωση επιπλέον κίνησης πακέτων στην τοπολογία δικτύου με VLAN και τροποποίηση της ποιότητας της υπηρεσίας (QoS)

Στο παρόν στάδιο θα δημιουργηθούν και πάλι καταστάσεις συμφόρησης για το δίκτυο εισάγοντας κίνηση στον εικονικό μεταγωγέα από τον Host. Σκοπός είναι να μελετηθεί η απόδοση του Open vSwitch όταν αυτό λειτουργεί με VLANs. Για την επίτευξη του στόχου χρησιμοποιήθηκαν και πάλι οι ίδιες εντολές με το υποκεφάλαιο 5.3.1.1 οι οποίες καταγράφονται παρακάτω:

#### Εντολή γεννήτριας πακέτων

- hping3 --udp -- flood 10.1.1.1
- hping3 -- flood 10.1.1.1
  
- hping3 --udp --fast 10.1.1.1
- hping3 -- faster 10.1.1.1

Για τις δύο πρώτες εντολές τα αποτελέσματα ήταν και πάλι αρνητικά για την επικοινωνία των εικονικών μηχανών, όπως άλλωστε αναμενόταν, καθώς ένα VLAN δεν έχει διαφορές στον τρόπο με τον οποίο λειτουργεί σε περιπτώσεις συμφόρησης από ένα απλό LAN. Αυτό σημαίνει ότι τα virtual machines, δε μπορούσαν να αναγνωριστούν μεταξύ τους και οποιαδήποτε αποστολή πακέτων είχε 100% απώλεια. Οι επόμενες δύο εντολές οι οποίες δημιουργούν δικτυακές κινήσεις φυσιολογικές και ανεκτές, δεν επέδρασαν αρνητικά στην επικοινωνία των εικονικών μηχανών. Υπήρχε μία μικρή αύξηση του μέσου

όρου χρόνου άφιξης των ICMP πακέτων που δημιουργήθηκαν από την εντολή ping, χωρίς ωστόσο να έχουμε καμία απώλεια.

Παράλληλα κατά τη μεταφορά του αρχείου από το ένα εικονικό μηχάνημα στο άλλο είχαμε το μέγιστο δυνατό ρυθμό μεταφοράς ενώ όταν γινόταν ταυτόχρονη αποστολή και λήψη από τα δύο VLAN ο ρυθμός αυτός έπεφτε στο μισό. Ολοκληρώνοντας τις βασικές μετρήσεις σε VLAN περιβάλλοντα, ακολουθήθηκε η ίδια διαδικασία που αναφέρθηκε παραπάνω για την τροποποίηση της ποιότητας της υπηρεσίας στις εικονικές μηχανές. Στη συγκεκριμένη τοπολογία επιλέχθηκε κάθε VLAN να έχει διαφορετικό QoS. Ο τροποποιημένος πίνακας με τις εντολές δίνεται παρακάτω:

#### Εντολές Τροποποίησης Quality Of Service (qosvlan.sh script)

##### **VM1:**

```
ovs-vsctl set Interface tap0 ingress_policing_rate=1000  
ovs-vsctl set Interface tap0 ingress_policing_burst=100
```

##### **VM2:**

```
ovs-vsctl set Interface tap1 ingress_policing_rate=1000  
ovs-vsctl set Interface tap1 ingress_policing_burst=100
```

##### **VM3:**

```
ovs-vsctl set Interface tap2 ingress_policing_rate=10000  
ovs-vsctl set Interface tap2 ingress_policing_burst=1000
```

##### **VM4:**

```
ovs-vsctl set Interface tap3 ingress_policing_rate=10000  
ovs-vsctl set Interface tap3 ingress_policing_burst=1000
```

Από τις παραπάνω εντολές είναι κατανοητό ότι στο VLAN1 επιλέχθηκε μέγιστος ρυθμός μετάδοσης το 1Mbps ενώ για το VLAN2 επιλέχθηκε μέγιστος ρυθμός 10Mbps. Κατά τη μεταφορά αρχείων οι ρυθμοί αυτοί επιβεβαιώθηκαν απόλυτα καθώς στο VLAN1 ο ρυθμός παρ' ότι βρισκόταν πάντα κοντά στο μέγιστο όριο δεν ξεπεράστηκε ποτέ, λειτουργώντας δηλαδή με τον πλέον βέλτιστο τρόπο, ενώ ο ρυθμός για το VLAN2 παρ' ότι δεν έφτασε στη μέγιστη τιμή του κατά τη μεταφορά του αρχείου διατηρούσε πάντα σταθερή τιμή, συνυπολογίζοντας πάντα ότι υπήρχε και η συμφόρηση που δημιουργείτο από το hping.

Συμπερασματικά, το Open vSwitch διατήρησε την καλή απόδοσή του και σε περιβάλλοντα τα οποία έχουν σχεδιασθεί με αρχιτεκτονική VLAN. Αυτό είναι ένα από τα πλεονεκτήματα του εικονικού μεταγωγέα καθώς η δημιουργία VLANs στο άμεσο μέλλον θα είναι αναγκαία σύμφωνα με τις ανάγκες και τα δεδομένα που υπάρχουν σήμερα.

## 5.4 Υλοποίηση 4ου σταδίου: Έλεγχος απόδοσης συστήματος σε streaming ήχου και βίντεο με χρήση VLC

Επόμενο βήμα είναι ο έλεγχος της απόδοσης του Open vSwitch σε περιπτώσεις streaming ήχου και βίντεο. Στην παρούσα φάση χρησιμοποιήθηκε το λογισμικό VLC player. Στο στάδιο αυτό της υλοποίησης θα ελέγξουμε την απόδοση του, τόσο σε περιπτώσεις συμφόρησης, όσο και σε περιπτώσεις που το δίκτυο είναι ελεύθερο από κινήσεις πακέτων που το επιβαρύνουν. Επιπλέον θα ελεγχθεί η απόδοση του τόσο στην απλή τοπολογία με τις τέσσερις εικονικές μηχανές που συνδέονται μεταξύ τους μέσω της γέφυρας του Open vSwitch, όσο και με την τοπολογία με τα VLANs.

### 5.4.1 Έλεγχος απόδοσης συστήματος σε streaming ήχου στην απλή τοπολογία

Αρχικά ελέγχθηκε η απόδοση του συστήματος στον ήχο. Η κωδικοποίηση για τον ήχο είναι η γνωστή σε όλους, συμπίεση MP3. Ρόλο server που θα κάνει το streaming είχε στην απλή τοπολογία μας ο Host. Σκοπός είναι ο ακουστικός, αρχικά, έλεγχος της απόδοσης του συστήματος. Για το λόγο αυτό ο server εκτέμπει το αρχείο ήχου σε όλη τη subnet με σκοπό το αρχείο να είναι διαθέσιμο προς όλες τις εικονικές μηχανές. Η διαχείριση του vlc μπορεί να γίνει είτε από τη γραμμή εντολών, είτε από το γραφικό περιβάλλον που παρέχει η εφαρμογή. Στην περίπτωσή μας, χρησιμοποιήθηκε το γραφικό περιβάλλον, καθώς η χρήση της γραμμής εντολών είχε ορισμένα error τα οποία δεν ήταν αντικείμενο της παρούσας διπλωματικής και αποφασίστηκε το γραφικό περιβάλλον που λειτουργούσε απροβλημάτιστα.

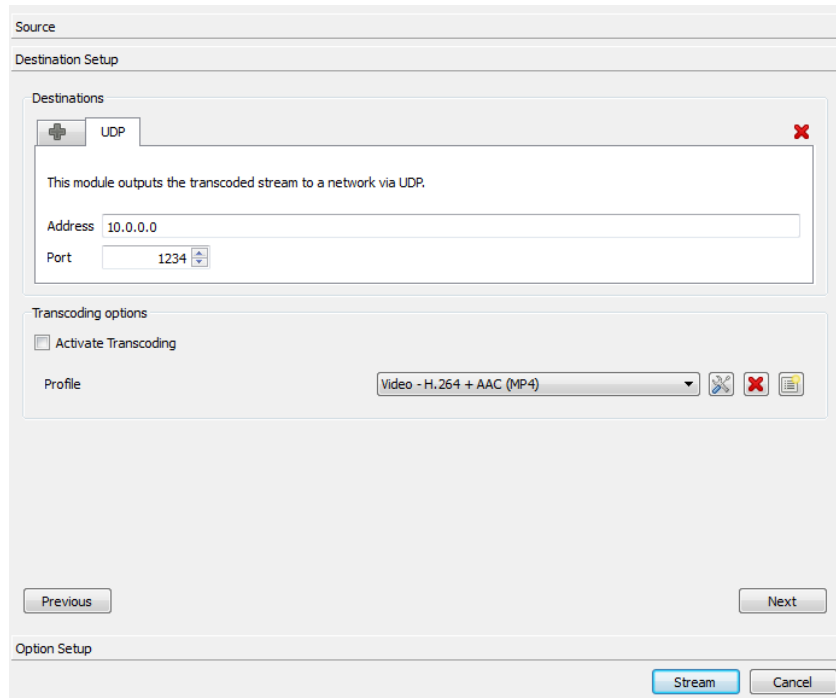
Για την μετάδοση ήχου προτιμήθηκε το πρωτόκολλο UDP καθώς σε περιπτώσεις streaming δεν απαιτούνται μηχανισμοί αξιοπιστίας από τα πρωτόκολλα μετάδοσης διότι παρεμποδίζονται από αυτούς και χειροτερεύει η απόδοσή τους. Στην γενική περίπτωση για την πλειοψηφία των εφαρμογών θα προτιμηθούν αξιόπιστα πρωτόκολλα. Ταυτόχρονα το πρωτόκολλο UDP υποστηρίζει broadcasting, δηλαδή την αποστολή ενός πακέτου σε όλους τους υπολογιστές ενός δικτύου καθώς και multicasting, δηλαδή την αποστολή ενός πακέτου σε συγκεκριμένους υπολογιστές του δικτύου. Η συγκεκριμένη δυνατότητα εφαρμόζεται απόλυτα στις ανάγκες των πειραμάτων μας καθώς μία ροή ήχου μπορεί να μεταδίδεται σε πολλούς συνδρομητές ταυτόχρονα.

Αρχικά επιλέχθηκε το προς streaming αρχείο. Εν συνεχεία επιλέχθηκε το πρωτόκολλο με το οποίο θα γίνει το streaming. Δόθηκε η διεύθυνση που θα γίνει το streaming όπου στην περίπτωσή μας είναι ολόκληρη η subnet καθώς και η πόρτα που θα χρησιμοποιηθεί. Το VLC χρησιμοποιεί τη λειτουργία της επανακωδικοποίησης του αρχείου. Παρ' όλα αυτά στην περίπτωσή μας δε θα χρησιμοποιηθεί αυτή η λειτουργία καθώς θα επιβαρύνει την επεξεργαστική ισχύ. Ως τελευταία επιλογή ορίστηκε το TTL (time to live) που σχετίζεται με τον αριθμό των router από τα οποία θα περάσει η ροή μας. Στη δική μας περίπτωση επιλέχθηκε η τιμή 1. Το γραφικό περιβάλλον του VLC δημιουργεί τέλος την εντολή που θα μπορούσαμε να χρησιμοποιήσουμε στη γραμμή εντολών. Αυτή καταγράφεται για λόγους χρηστικότητας στον επόμενο πίνακα.

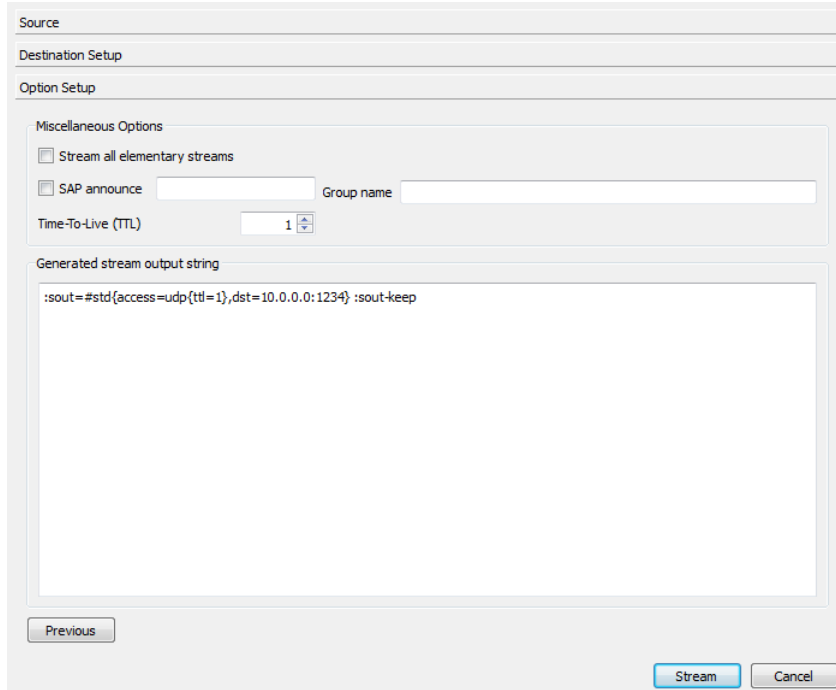
#### Εντολή MP3 streaming – UDP protocol

- `Vlc - -vvv /home/npc/Desktop/music.mp3 :sout=#:std{access=udp{ttl=1},dst=10.0.0.0:1234} :sout-keep`

Παρακάτω δίνονται δύο στιγμιότυπα από τις ρυθμίσεις που έγιναν από το γραφικό περιβάλλον του VLC player.



Εικόνα 22: Streaming ήχου μέσω VLC interface – απλή τοπολογία



Εικόνα 23: Εντολή που εκτελείται από το VLC – απλή τοπολογία



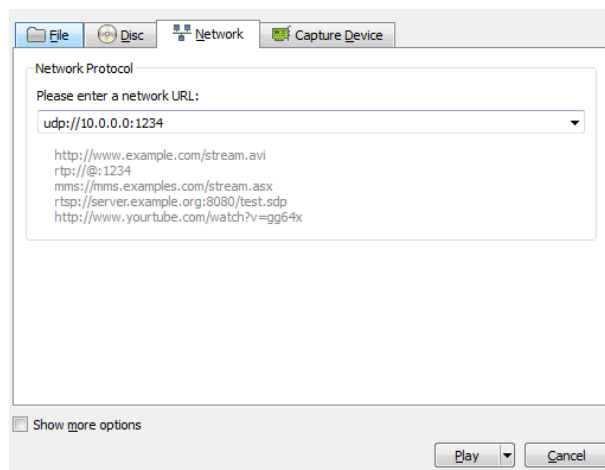
Εκτελώντας την επιλογή stream, ξεκινάει η διαδικασία streaming του ήχου προς όλους τους υπολογιστές του υποδικτύου που θα επιλέξουν να αναπαράγουν τη ροή. Επόμενο βήμα είναι η αναπαραγωγή του αρχείου από τις εικονικές μηχανές και η αξιολόγηση της απόδοσης του.

Κάθε βήμα που θα καταγραφεί στην επόμενη παράγραφο για το VM1, αφορά το σύνολο των εικονικών μηχανών ακολουθώντας ακριβώς την ίδια διαδικασία. Όπως και στην περίπτωση του streaming, έτσι και στην περίπτωση της αναπαραγωγής, ο τρόπος που μπορεί να γίνει είναι είτε μέσα από το γραφικό περιβάλλον είτε από τη γραμμή εντολών. Για την αναπαραγωγή του αρχείου απαιτείται η γνώση του πρωτοκόλλου που χρησιμοποιείται και στην περίπτωση μας είναι το UDP. Από τη γραμμή εντολών συνεπώς, εκτελέστηκε η ακόλουθη εντολή σε κάθε εικονική μηχανή

#### Εντολή Αναπαραγωγής MP3 – UDP protocol

- `vlc udp://10.0.0.0:1234`

Αντίστοιχα ακολουθεί στιγμιότυπο από το γραφικό περιβάλλον της εφαρμογής ως εναλλακτικός τρόπος αναπαραγωγής.



Εικόνα 24: Αναπαραγωγή του streaming ήχου από VLC interface – απλή τοπολογία

Το αποτέλεσμα του streaming ενός απλού αρχείου MP3 ήταν άκρως ικανοποιητικό. Τόσο με ταυτόχρονη αναπαραγωγή από όλες τις εικονικές μηχανές, όσο και μεμονωμένα, ο ήχος δεν εμφάνιζε διακοπές, παρά μόνο στην αρχή και συγκεκριμένα στο στάδιο του buffering.

#### 5.4.2 Έλεγχος απόδοσης συστήματος σε streaming ήχου στην απλή τοπολογία μαζί με δικτυακή συμφόρηση.

Αφού μελετήθηκε η απόδοση του Open vSwitch σε περιπτώσεις streaming σε τοπολογία δικτύου χωρίς μεγάλο φόρτο εργασίας, σειρά είχε η μελέτη της συμπεριφοράς του Open vSwitch σε δίκτυο με μεγάλη συμφόρηση. Για το σκοπό του συγκεκριμένου σταδίου θα γίνει και πάλι χρήση του προγράμματος hring3 χωρίς όμως να δοκιμαστεί η ακραία περίπτωση της πλημμύρας πακέτων. Αντίθετα θα γίνει χρήση εντολών που συμβαδίζουν με ένα δίκτυο που έχει αρκετά μεγάλο δικτυακό φόρτο.

Η δικτυακή τοπολογία παραμένει σταθερή και σε αυτή εισάγουμε κίνηση με ρυθμούς αυξημένους αλλά σε πλαίσια που η δικτυακή λειτουργία έχει αποδειχθεί ότι δεν επηρεάζεται σε μεγάλο βαθμό. Η εντολή δημιουργίας συμφόρησης δίνεται στον ακόλουθο πίνακα.

#### Εντολή γεννήτριας πακέτων

- hping3 --udp --fast 10.1.1.1
- hping3 --faster 10.1.1.1

Στη συγκεκριμένη κατάσταση, η μετάδοση του ήχου γίνεται απρόσκοπτα καθώς κανένα μέρος της επικοινωνίας δεν έχει διακοπεί. Παρ' όλα αυτά η ποιότητα του ήχου έχει ελαττωθεί. Παρατηρήθηκαν διακοπές κατά την αναπαραγωγή του αρχείου κυρίως όταν είχαμε ταυτόχρονη λήψη του από όλες τις εικονικές μηχανές. Όταν το αρχείο αναπαραγόταν από μία μονάχα εικονική μηχανή η ποιότητα απόδοσης ήταν σαφώς ανώτερη και σχεδόν χωρίς διαλείψεις.

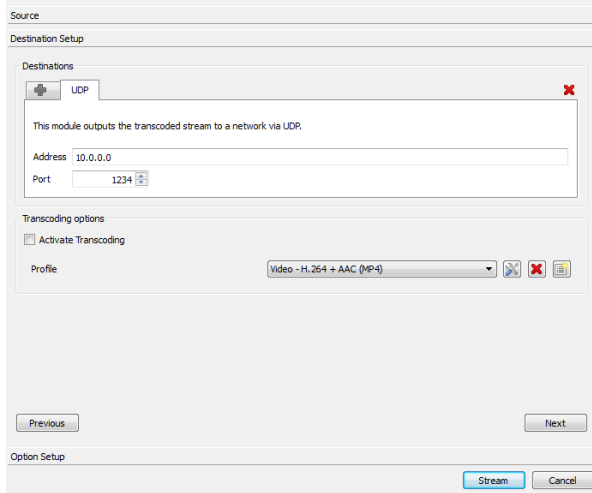
### 5.4.3 Έλεγχος απόδοσης συστήματος σε streaming ήχου στην τοπολογία με χρήση VLAN

Στο συγκεκριμένο στάδιο του πειράματος, και αφού εγκαταστάθηκαν με επιτυχία τα δύο εικονικά LAN με τη διαδικασία που αναφέρθηκε στο υποκεφάλαιο 5.3.2, ελέγχθηκε η ποιότητα που προσφέρει ο εικονικός μεταγωγέας κατά το streaming ενός αρχείου MP3 τόσο σε περιβάλλον με μικρό δικτυακό φόρτο, όσο και σε περιβάλλον με δικτυακή συμφόρηση. Η επιλογή του πρωτοκόλλου μεταφοράς θα είναι ίδια με την προηγούμενη περίπτωση, δηλαδή UDP, για τους λόγους που αναφέρθηκαν.

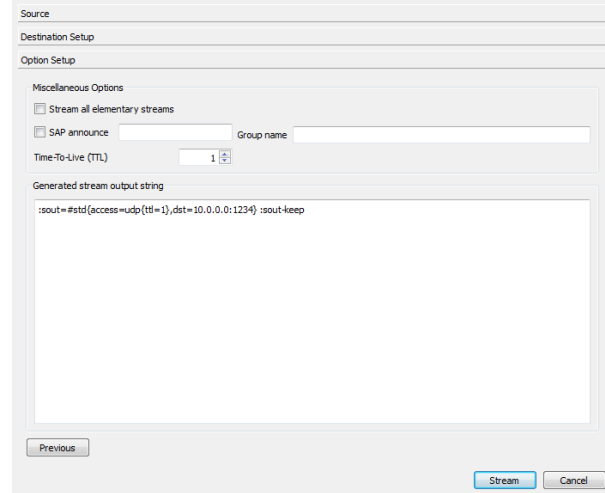
Η διαφορά που παρουσιάζεται στα VLAN είναι ο τρόπος που τελικά επιλέχθηκε για να γίνει το streaming του αρχείου. Ενώ στην περίπτωση της απλής δικτυακή τοπολογίας είχαμε broadcasting του αρχείου, δηλαδή το αρχείο ήταν διαθέσιμο για αναπαραγωγή από οποιοδήποτε μηχάνημα ήταν συνδεδεμένο στο υποδίκτυο, στην περίπτωση των VLAN το streaming θα γίνεται αποκλειστικά από και προς τις εικονικές μηχανές και όχι από τον host. Αυτό γίνεται εύκολα κατανοητό καθώς τα VLAN είναι απομονωμένα από τον Host, του οποίου το interface έχει συνδεθεί ως πόρτα στον εικονικό μεταγωγέα, αλλά δεν του έχουμε ορίσει κάποιο tag. Αυτό έγινε διότι αναζητούμε την απόλυτη απομόνωση των εικονικών LAN. Εδώ αξίζει να τονιστεί ότι σε περίπτωση που θα θέλαμε να συνδέσουμε τον εικονικό μας μεταγωγέα με ένα δεύτερο φυσικό μηχάνημα η πόρτα Ethernet που έχει συνδεθεί στο br0 λειτουργεί ως VLAN trunk και συνεπώς μπορεί να μεταφέρει πακέτα που έχουν tagging. Με τον τρόπο αυτό μπορούν να επεκταθούν τα VLAN σε δεύτερα και τρίτα φυσικά μηχανήματα τα οποία έχουν εικονικές μηχανές και τρέχουν το Open vSwitch με τα κατάλληλα tagging.

Εν συνεχεία ξεκίνησε η διαδικασία streaming από το VM1 στο VM2 και από το VM3 στο VM4, δηλαδή εσωτερικά των VLANs. Και στις δύο περιπτώσεις χρησιμοποιήθηκαν ακριβώς οι ίδιες ρυθμίσεις για την επανακωδικοποίηση και το πρωτόκολλο μεταγωγής. Παρακάτω δίνονται τα στιγμιότυπα από την διαδικασία του streaming τόσο γραφικά όσο και σε γραμμή εντολών.

Αρχικά ορίσαμε τη διεύθυνση από την οποία μπορεί μία εικονική μηχανή που ανήκει στο VLAN1 να αναπαράγει τη δεδομένη ροή. Η διεύθυνση αυτή είναι η IP address του subnet, δηλαδή η διεύθυνση του προορισμού αναπαραγωγής. Το δεύτερο στιγμιότυπο εμφανίζει την εντολή που μπορεί να χρησιμοποιηθεί από τη γραμμή εντολών για μεγαλύτερη ευκολία.



Εικόνα 25: Streaming ήχου μέσω VLC interface – VLAN



Εικόνα 26: Εντολή που εκτελείται από το VLC – VLAN

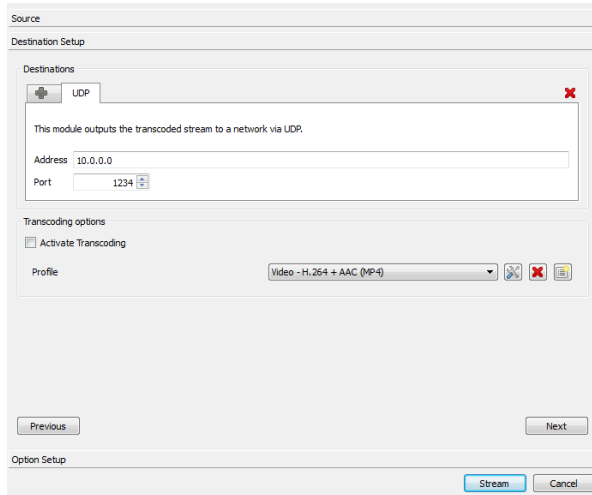
Παρακάτω δίνονται οι εντολές streaming και αναπαραγωγής από τη γραμμή εντολών.

Εντολή MP3 streaming – UDP protocol
<ul style="list-style-type: none"> <li>Vlc - -vvv /home/npc/Desktop/music.mp3 :sout=# std{access=udp{ttl=1},dst=10.0.0.0:1234} :sout-keep</li> </ul>
Εντολή Αναπαραγωγής MP3
<ul style="list-style-type: none"> <li>vlc udp://10.0.0.0:1234</li> </ul>

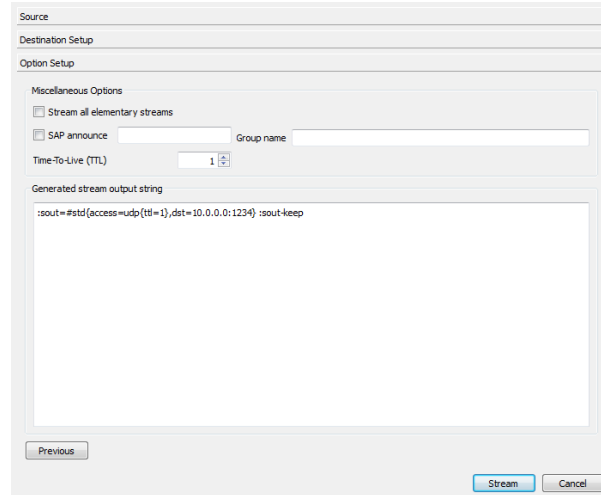
Στο σημείο αυτό και χωρίς να έχει ξεκινήσει το streaming από τη δεύτερη εικονική μηχανή μπορούμε με ασφάλεια να συμπεράνουμε ότι η απόδοση της ποιότητας είναι άριστη. Εκτός από το αρχικό buffering που δημιουργεί κάποιες διαλείψεις δευτερολέπτων, η υπόλοιπη αναπαραγωγή συνεχίζεται χωρίς διακοπές.

Ακολούθησε η δημιουργία streaming στο δεύτερο VLAN και συγκεκριμένα από το VM3 στο VM4. Αντίστοιχα προτιμήθηκε το πρωτόκολλο UDP, transcoding MP3 και για διεύθυνση προορισμού, επιλέχθηκε η IP address του VM4. Παρακάτω δίνονται τόσο τα στιγμιότυπα από το γραφικό

περιβάλλον, όσο και οι εντολές που θα απαιτούνταν σε περίπτωση που επιλεγεί η γραμμή εντολών για την διαδικασία.



Εικόνα 28: Streaming ήχου μέσω VLC interface - VLAN



Εικόνα 27: Εντολή που εκτελείται από VLC - VLAN

Παρακάτω δίνονται οι εντολές streaming και αναπαραγωγής από τη γραμμή εντολών.

Εντολή MP3 streaming – UDP protocol
<ul style="list-style-type: none"> <li>Vlc - -vvv /home/npc/Desktop/music.mp3 :sout=#:std{access=udp{ttl=1},dst=10.0.0.0:1234}:sout-keep</li> </ul>
Εντολή Αναπαραγωγής MP3
<ul style="list-style-type: none"> <li>vlc udp://10.0.0.0:1234</li> </ul>

Όπως και στην περίπτωση του VLAN1 έτσι και στο VLAN2 όταν αυτό αναπαράγει το αρχείο χωρίς οποιαδήποτε άλλη κίνηση αποδίδει με άριστη ποιότητα. Τούτο κρίνεται απόλυτα λογικό καθώς τα δύο VLAN δε θα πρέπει να έχουν ουσιαστικά καμία ποιοτική διαφορά τόσο στον τρόπο λειτουργίας τους όσο και στον τρόπο απόδοσής τους.

Τέλος στην τοπολογία VLAN χωρίς δικτυακό φόρτο δοκιμάστηκε η απόδοση της ποιότητας με ταυτόχρονο streaming και αναπαραγωγή του αρχείου ήχου από τα εικονικά μηχανήματα. Στην περίπτωση αυτή παρατηρήθηκαν ορισμένες διακοπές κατά τη διάρκεια αναπαραγωγής συγκριτικά πάντα με το άριστο αποτέλεσμα που είχαμε κατά την μονωμένη αναπαραγωγή. Ένας βασικός λόγος στον οποίο πιθανότατα οφείλεται αυτή η συμπεριφορά είναι ότι τόσο το transcoding όσο και το streaming είναι διαδικασίες που απαιτούν αρκετή υπολογιστική ισχύ την οποία ένα εικονικό μηχάνημα προσομοιώνει και συνεπώς οι διακοπές δεν οφείλονται στον εικονικό μεταγωγέα αλλά στις διαδικασίες

που αναφέρθηκαν και δεν εκτελούνται πλέον από το φυσικό υλικό του μηχανήματος. Συμπερασματικά το streaming σε VLANs χωρίς μεγάλη δικτυακή κίνηση, κρίνεται άκρως ικανοποιητικό με τη χρήση του Open vSwitch ως εικονικού μεταγωγέα.

#### 5.4.4 Έλεγχος απόδοσης συστήματος σε streaming ήχου στην τοπολογία με VLANs μαζί με δικτυακή συμφόρηση.

Στο παρόν κεφάλαιο θα μελετηθεί η απόδοση του του Open vSwitch σε περιπτώσεις streaming σε τοπολογία δικτύου με μεγάλο φόρτο εργασίας,. Για την επίτευξη της δικτυακής κίνησης έγινε και πάλι χρήση του προγράμματος hping3 με αποδέκτη της κίνησης το Open vSwitch. Σκοπός είναι να δημιουργηθεί κίνηση μέσα στον εικονικό μεταγωγέα και ταυτόχρονα να ελεγχθεί η ποιότητα της πειραματικής διάταξης.

Η δικτυακή τοπολογία παραμένει σταθερή, δηλαδή με δύο VLAN στα οποία υπάρχουν δύο εικονικές μηχανές. Εισάγουμε κίνηση με ρυθμούς αυξημένους σε σχέση με την προηγούμενη δικτυακή κίνηση και επιλέγουμε πάλι πρωτόκολλα UDP και TCP. Η εντολή δημιουργίας συμφόρησης δίνεται στον ακόλουθο πίνακα.

##### Εντολή γεννήτριας πακέτων

- hping3 --udp --fast 10.1.1.1
- hping3 --faster 10.1.1.1

Αναλύοντας τα ακουστικά αποτελέσματα, οδηγούμαστε στο συμπέρασμα ότι με μεμονωμένη αναπαραγωγή από τις εικονικές μηχανές η ποιότητα του ήχου κρίνεται ικανοποιητική όχι όμως σε βαθμό όμοιο με αυτό της τοπολογίας χωρίς δικτυακή συμφόρηση. Παράλληλα, η ταυτόχρονη εκπομπή και αναπαραγωγή του αρχείου μουσικής μέσα στις εικονικές μηχανές με ταυτόχρονη δικτυακή συμφόρηση, κρίνεται μέτρια ως προς την απόδοσή της. Αρκετές διακοπές στη μετάδοση χωρίς ωστόσο χαθεί ολοκληρωτικά η επικοινωνία. Βέβαια πρόκειται για μία πολύ μικρή τοπολογία και μπορούμε να συμπεράνουμε ότι το streaming από εικονική μηχανή σε εικονική μηχανή ενός VLAN χρειάζεται βελτίωση.

#### 5.4.5 Έλεγχος απόδοσης συστήματος σε streaming βίντεο στην απλή τοπολογία

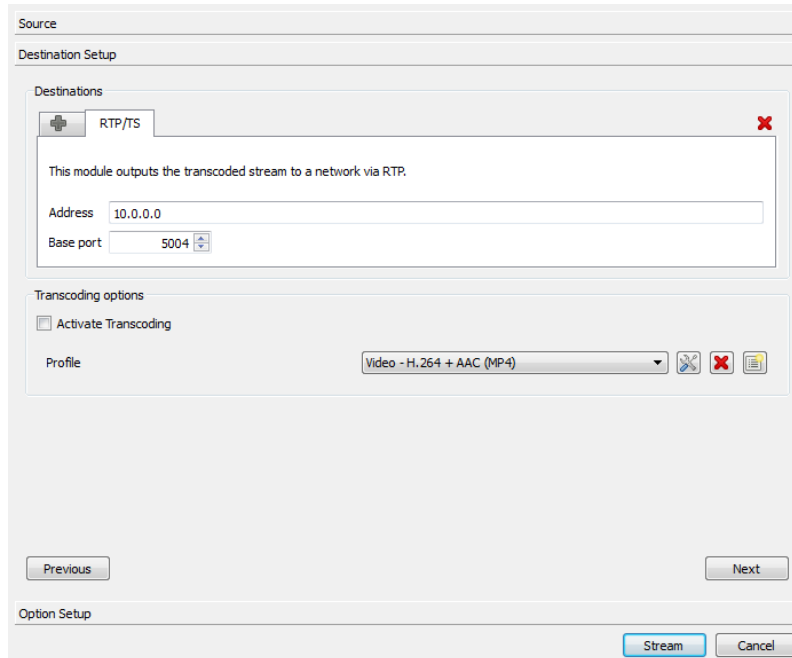
Αφού ολοκληρώθηκε το κομμάτι που αφορούσε τη μετάδοση ήχου στις δικτυακές τοπολογίες μας, σειρά είχε ο έλεγχος της απόδοσης του εικονικού μεταγωγέα σε περιπτώσεις που αφορούν streaming βίντεο. Οι περιπτώσεις που θα μελετηθούν ταυτίζονται με αυτές τις μετάδοσης του ήχου, δηλαδή αφορούν στο streaming βίντεο στην απλή τοπολογία δικτύου όπου τα εικονικά μηχανήματα συνδέονται στον εικονικό μεταγωγέα και βρίσκονται όλα στο ίδιο υποδίκτυο καθώς και στην τοπολογία δικτύου με δύο VLANs, τόσο με δικτυακή συμφόρηση όσο και χωρίς.

Η σημαντικότερη διαφοροποίηση στην περίπτωση video streaming είναι το πρωτόκολλο που θα χρησιμοποιηθεί για το σκοπό αυτό και ονομάζεται RTP (Real-time Transport Protocol). Το συγκεκριμένο πρωτόκολλο προκαθορίζει την μορφή των πακέτων που μεταδίδουν ήχο και βίντεο σε IP δίκτυα. Χρησιμοποιείται ευρέως σε συστήματα επικοινωνιών που αφορούν media streaming, όπως η βιντεοσυνδιαλέξεις, τα τηλεφωνικά συστήματα (Voice IP) καθώς και υπηρεσίες τηλεόρασης (IPTV). Το RTP αποτελείται ουσιαστικά από δύο επιμέρους πρωτόκολλα. Το πρώτο είναι το πρωτόκολλο μεταφοράς δεδομένων το οποίο διαχειρίζεται τη μεταφορά των δεδομένων σε πραγματικό χρόνο καθώς και πληροφορίες που αφορούν timestamps (υπεύθυνες για το συγχρονισμό εικόνας και ήχου), sequence numbers (για απώλειες πακέτων και αναγνώριση αναδιάταξης τους) καθώς και τη μορφή του payload το οποίο προσδιορίζει τη κωδικοποίηση των δεδομένων. Το δεύτερο είναι το πρωτόκολλο ελέγχου, RTCP το οποίο προσδιορίζει την ποιότητα της υπηρεσίας, τον έλεγχο ανατροφοδότησης και τον συγχρονισμό μεταξύ των ροών. Το εύρος ζώνης που χρησιμοποιείται από το πρωτόκολλο ελέγχου είναι μικρό σε σχέση με αυτό του RTP, περίπου 5% του συνολικού εύρους ζώνης.

Αφού έγινε η επιλογή του κατάλληλου πρωτοκόλλου, επόμενο βήμα ήταν η διαδικασία streaming του βίντεο από τον κεντρικό μας Host. Η αναπαραγωγή του αρχείου επιλέχθηκε να γίνεται τόσο ταυτόχρονα όσο και μεμονωμένα από κάθε virtual machine ξεχωριστά. Στο στάδιο αυτό του πειράματος δεν έχουμε επιπλέον δικτυακή κίνηση και συνεπώς η απόδοση του βίντεο θα πρέπει να είναι η βέλτιστη. Το πρόγραμμα streaming είναι και πάλι το VLC με διαφοροποίηση αυτή τη φορά στο πρωτόκολλο το οποίο θα χρησιμοποιηθεί. Καθώς έχουμε μετάδοση και βίντεο και εικόνας θα πρέπει να υπάρξει ένας συνδυασμός της κωδικοποίησης του αρχείου προς streaming. Ωστόσο η επιλογή του αρχείου προς βίντεο έχει γίνει ώστε να αποφευχθεί η διαδικασία της επανακωδικοποίησης. Το codec με το οποίο είναι κωδικοποιημένο το αρχείο είναι MPEG-4 το οποίο χρησιμοποιεί 50% λιγότερο εύρος ζώνης από το MPEG-2 με ανάλυση η οποία είναι εφάμιλλη ενός DVD. Ο ήχος θα παραμείνει σε κωδικοποίηση MP3 και το πρωτόκολλο επιλογής είναι το RTP. Ως διεύθυνση προορισμού του streaming ορίστηκε η 10.0.0.0 δηλαδή όλο το υποδίκτυο ώστε να είναι διαθέσιμο προς κάθε υπολογιστή που είναι συνδεδεμένο σε αυτό.

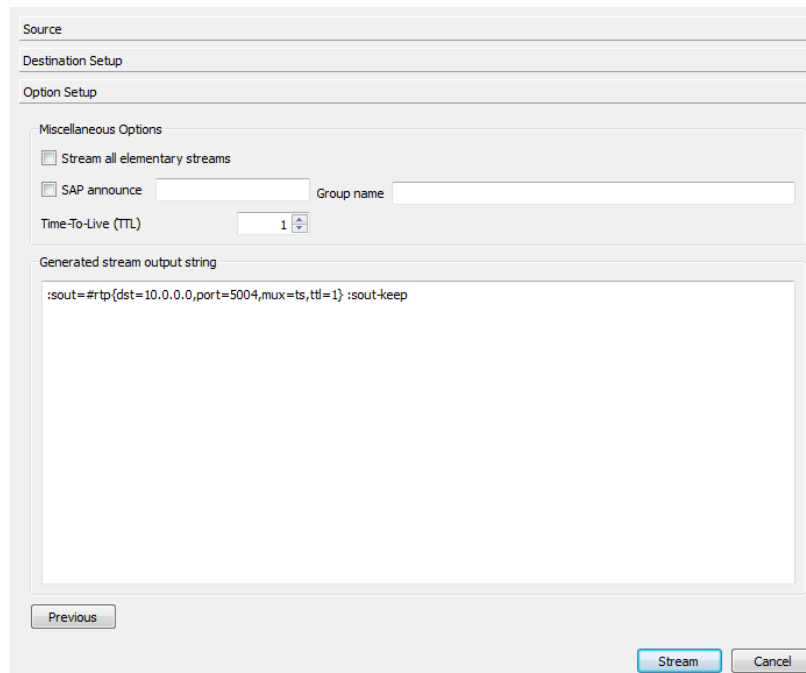
Ακολούθως δίνονται τα γραφικά στιγμιότυπα από το VLC καθώς και οι εντολές που μπορούν να προκαλέσουν streaming μέσα από τη γραμμή εντολών.

Στην πρώτη εικόνα καθορίστηκε η διεύθυνση από την οποία θα είναι διαθέσιμο το βίντεο, η πόρτα καθώς και απενεργοποιήθηκε η επανακωδικοποίηση του αρχείου.



Εικόνα 29: Streaming βίντεο μέσω VLC interface - απλή τοπολογία

Στη δεύτερη εικόνα δίνεται η εντολή η οποία μπορεί να χρησιμοποιηθεί από τη γραμμή εντολών. Παρατηρούμε επίσης το TTL αντίστοιχα ορισμένο στο 1, όσος είναι δηλαδή και ο αριθμός των μεταγωγών από τον οποίο θα περάσει η ροή.



Εικόνα 30: Εντολή που εκτελείται από VLC - απλή τοπολογία

Στον ακόλουθο πίνακα καταγράφεται η εντολή που προκαλεί το ίδιο ακριβώς αποτέλεσμα με το γραφικό περιβάλλον σε γραμμή εντολών.

Εντολή Video streaming – RTP protocol
<ul style="list-style-type: none"><li>Vlc - -vvv /home/vm1/Desktop/1.mp4 :sout=#rtp{dst=10.0.0.0,port=5004,mux=ts,ttl=1} :sout - keep</li></ul>
Εντολή Αναπαραγωγής Video
<ul style="list-style-type: none"><li>vlc rtp10.0.0.0:5004</li></ul>

Έχοντας ανοιχτά και τα τέσσερα εικονικά μηχανήματα ξεκινάμε την αναπαραγωγή του βίντεο αρχικά σε κάθε ένα μηχανήμα ξεχωριστά. Αποτέλεσμα είναι η τέλεια απόδοση εικόνας και ήχου μέσα στην εικονική μηχανή. Το μοναδικό πρόβλημα που εντοπίζεται είναι στο συγχρονισμό εικόνας και ήχου. Αντίθετα δεν παρουσιάζεται κανένα πρόβλημα τόσο στην απεικόνιση του βίντεο, όσο και στην απόδοση του ήχου.

Επόμενο βήμα ήταν η ταυτόχρονη αναπαραγωγή του βίντεο από τις τέσσερις εικονικές μηχανές. Αφού εκτελέστηκε η εντολή της αναπαραγωγής, το αποτέλεσμα που είχαμε ήταν παρ με το προηγούμενο. Παρ' ότι λειτουργούσαν και οι τέσσερις εικονικές μηχανές, τόσο ο ήχος όσο και η εικόνα είχαν άριστη οπτικοακουστική απόδοση. Εδώ αξίζει να αναφερθεί ότι η τοπολογία μας δεν εμφανίζει καμία επιπλέον δικτυακή κίνηση και συνεπώς θα πρέπει να λειτουργεί με τον πλέον βέλτιστο τρόπο.

#### **5.4.6 Έλεγχος απόδοσης συστήματος σε streaming βίντεο στην απλή τοπολογία με δικτυακή συμφόρηση.**

Αφού μελετήθηκε η απόδοση του Open vSwitch σε περιπτώσεις streaming βίντεο σε τοπολογία δικτύου χωρίς μεγάλο δικτυακό φόρτο, σειρά είχε η μελέτη της συμπεριφοράς του Open vSwitch σε δίκτυο με μεγάλη συμφόρηση. Για το σκοπό του συγκεκριμένου σταδίου θα γίνει χρήση του προγράμματος hring. Η πλημμύρα πακέτων και σε αυτή την περίπτωση θα αποφευχθεί καθώς δεν επιθυμείται η διακοπή της επικοινωνίας μεταξύ των εικονικών μηχανών και του μεταγωγέα. Η δικτυακή κίνηση που θα δημιουργηθεί από τη γεννήτρια, συμβαδίζει με αυτή μιας τοπολογίας δίκτυο με υψηλή κίνηση σε φυσιολογικές όμως συνθήκες.

Η δικτυακή τοπολογία παραμένει σταθερή και σε αυτή εισάγουμε κίνηση με ρυθμούς που έχουν περιγραφεί αναλυτικά σε προηγούμενα κεφάλαια της εργασίας. Η εντολή δημιουργίας συμφόρησης δίνεται στον ακόλουθο πίνακα.



#### Εντολή γεννήτριας πακέτων

- hping3 - -udp --fast 10.1.1.1
- hping3 - -faster 10.1.1.1

Στη συγκεκριμένη κατάσταση, η μετάδοση του βίντεο έγινε χωρίς καμία διακοπή τόσο στην εικόνα όσο και στον ήχο ενώ η επικοινωνία δε χάθηκε από κανένα εικονικό μηχάνημα. Υπήρχαν μικρές διαλείψεις και καθυστερήσεις σε μερικά σημεία της αναπαραγωγής, ωστόσο η συνολική απόδοση κρίνεται απόλυτα ικανοποιητική. Ταυτόχρονα δοκιμάστηκε και η μεταφορά αρχείων από και προς τα εικονικά μηχανήματα. Λόγω του περιορισμένου εύρους ζώνης η μετάδοση γινόταν με αργό ρυθμό ωστόσο δεν υπήρξε επιβάρυνση στο σκέλος του streaming. Συμπερασματικά και για το συγκεκριμένο στάδιο του πειράματος μπορούμε να πούμε ότι το Open vSwitch λειτουργεί με πολύ καλή απόδοση κατά σε δεδομένα ενός μικρού δικτύου με μεγάλο φόρτο δικτύου.

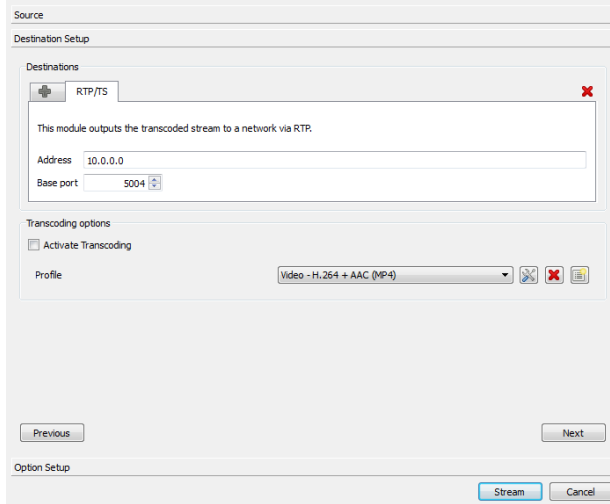
#### 5.4.7 Έλεγχος απόδοσης συστήματος σε streaming βίντεο στην τοπολογία με χρήση VLAN

Στο συγκεκριμένο στάδιο του πειράματος, εφαρμόστηκε και πάλι η δικτυακή εγκατάσταση με τα δύο VLAN τα οποία περιέχουν δύο εικονικές μηχανές το κάθε ένα. Το πρωτόκολλο που εφαρμόζεται για το streaming video είναι το RTP που αναφέρθηκε παραπάνω. Στο παρόν κεφάλαιο θα μελετήσουμε την απόδοση του streaming χωρίς δικτυακό φόρτο γεγονός. Η παρούσα υλοποίηση είναι αρκετά σημαντική καθώς το video streaming σε εικονικές μηχανές είναι ένας από τους τομείς που εμφανίζεται ιδιαίτερα δημοφιλής τα τελευταία χρόνια και όλο και περισσότερες βελτιώσεις γίνονται σε αυτόν.

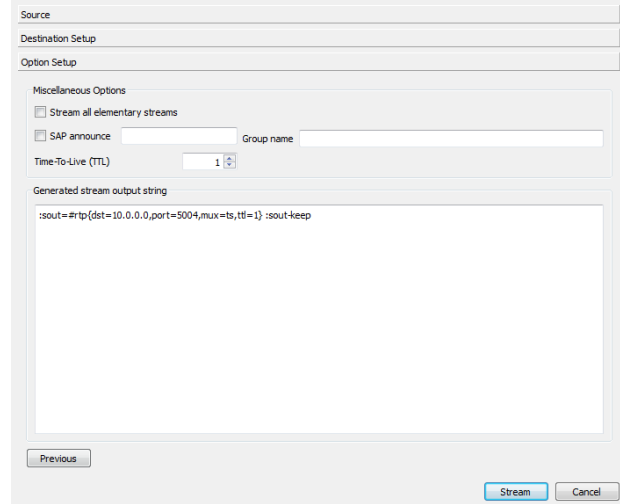
Όπως και στην περίπτωση που είχαμε streaming ήχου, έτσι και στην περίπτωση με το streaming video, η διαφορά των δύο τοπολογιών έγκειται στον τρόπο με τον οποίο γίνεται διαθέσιμο το αρχείο. Στην πρώτη περίπτωση ο Host έχει πρόσβαση σε όλες τις εικονικές μηχανές που είναι συνδεδεμένες στον εικονικό μεταγωγέα. Στην περίπτωση με τα VLAN, μία εικονική μηχανή έχει το ρόλο server, δηλαδή είναι αυτή που θα κάνει διαθέσιμο προς streaming το αρχείο video. Το αρχείο μπορεί να αναπαραχθεί από οποιαδήποτε εικονική μηχανή βρίσκεται στο VLAN1 ή στο VLAN2, καθώς γίνεται streaming σε ολόκληρη τη subnet. Παράλληλα, παρ' ότι όπως θα δούμε στη συνέχεια η διεύθυνση που γίνεται το streaming είναι η ίδια και για τα δυο VLAN, όταν μεταδίδει το βίντεο εικονική μηχανή από το VLAN1 δε μπορεί να αναπαραχθεί από καμία εικονική μηχανή του VLAN2 και αντίστροφα. Αυτό οφείλεται στο γεγονός ότι τα πακέτα προς μεταγωγή ενθυλακώνουν μία ετικέτα καθώς περνάνε από τον εικονικό μεταγωγέα και κατευθύνονται μόνο προς τους προορισμούς που έχουν την αντίστοιχη.

Επόμενο βήμα ήταν η διαδικασία streaming από το VM1 στο VM2 και από το VM3 στο VM4, δηλαδή εσωτερικά των VLANs. Και στις δύο περιπτώσεις χρησιμοποιήθηκαν ακριβώς οι ίδιες ρυθμίσεις μετάδοσης της πληροφορίας. Όπως και στην απλή τοπολογία, δεν επιλέχθηκε η λειτουργία της επανακωδικοποίησης ειδικά τώρα που το streaming γίνεται από εικονική μηχανή και οποιοδήποτε υλικό προσομοιώνεται. Παρακάτω δίνονται τα στιγμιότυπα από την διαδικασία του streaming τόσο γραφικά όσο και σε γραμμή εντολών.

Αρχικά ορίσθηκε η διεύθυνση από την οποία μπορεί μία εικονική μηχανή που ανήκει στο VLAN1 να αναπαράγει τη δεδομένη ροή. Η διεύθυνση αυτή είναι η IP του subnet όπως αναφέρθηκε, δηλαδή η 10.0.0.0. Η πόρτα που χρησιμοποιείται για τη μετάδοση του βίντεο είναι η 5004. Παρακάτω δίνονται τα στιγμιότυπα των ρυθμίσεων καθώς και η εντολή που δημιουργείται για τη γραμμή εντολών.



Εικόνα 32: Αναπαραγωγή μέσω VLC interface - VLAN



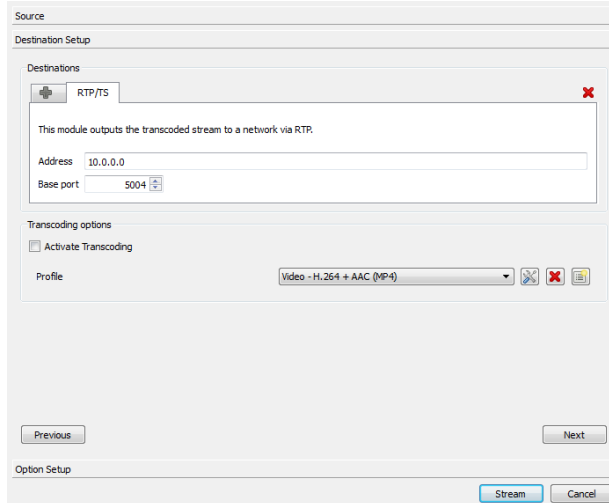
Εικόνα 31: Εντολή δημιουργίας streaming - VLAN

Παρακάτω δίνονται οι εντολές streaming και αναπαραγωγής από τη γραμμή εντολών.

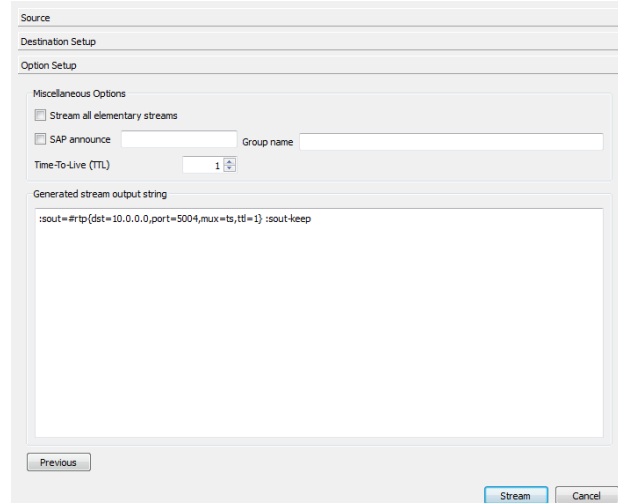
Εντολή Video streaming – RTP protocol
<ul style="list-style-type: none"> <li>Vlc - -vvv /home/npc/Desktop/1.mp4 :sout=#rtp{dst=10.0.0.0,port=5004,mux=ts,tfl=1}:sout-keep</li> </ul>
Εντολή Αναπαραγωγής Video
<ul style="list-style-type: none"> <li>vlc rtp10.0.0.0:5004</li> </ul>

Στο σημείο αυτό και χωρίς να έχει ξεκινήσει το streaming από τη δεύτερη εικονική μηχανή μπορούμε με ασφάλεια να συμπεράνουμε ότι η απόδοση της ποιότητας είναι άριστη. Η απόδοση εικόνας και ήχου είναι όπως η αρχική, ενώ οι διακοπές είναι ελάχιστες έως μηδαμινές.

Ακολούθησε η δημιουργία streaming στο δεύτερο VLAN και συγκεκριμένα από το VM3 στο VM4. Αντίστοιχα επιλέχθηκε το πρωτόκολλο RTP, χωρίς καμία κωδικοποίηση και για διεύθυνση προορισμού, επιλέχθηκε η IP address της subnet. Παρακάτω δίνονται τόσο τα στιγμιότυπα από το γραφικό περιβάλλον, όσο και οι εντολές που θα απαιτούνταν σε περίπτωση που επιλεγθεί η γραμμή εντολών για την διαδικασία.



Εικόνα 34: Αναπαραγωγή μέσω VLC interface - VLAN



Εικόνα 33: Εντολή δημιουργίας streaming - VLAN

Παρακάτω δίνονται οι εντολές streaming και αναπαραγωγής από τη γραμμή εντολών.

Εντολή MP3 streaming – UDP protocol
<ul style="list-style-type: none"> <li>Vlc - -vvv /home/vm1/Desktop/1.mp4 :sout=#rtp{dst=10.0.0.0,port=5004,mux=ts,ttl=1}:sout-keep</li> </ul>
Εντολή Αναπαραγωγής MP3
<ul style="list-style-type: none"> <li>vlc udp://10.0.0.0:1234</li> </ul>

Όπως και στην περίπτωση του VLAN1 έτσι και στο VLAN2 όταν αυτό αναπαράγει το αρχείο χωρίς οποιαδήποτε άλλη κίνηση, το αποδίδει με άριστη ποιότητα. Τούτο κρίνεται απόλυτα λογικό καθώς τα δύο VLAN δε θα πρέπει να έχουν ουσιαστικά καμία ποιοτική διαφορά τόσο στον τρόπο λειτουργίας τους όσο και στον τρόπο απόδοσής τους.

Τέλος στην τοπολογία VLAN χωρίς δικτυακό φόρτο δοκιμάστηκε η απόδοση της ποιότητας με ταυτόχρονο streaming και αναπαραγωγή του αρχείου βίντεο από τα εικονικά μηχανήματα. Στην περίπτωση αυτή παρατηρήθηκαν ορισμένες διακοπές κατά τη διάρκεια αναπαραγωγής κυρίως κατά την εκκίνηση του αρχείου, ενώ αυτές εξαλείφθηκαν στη συνέχεια, συγκριτικά πάντα με το άριστο αποτέλεσμα που είχαμε κατά την μονωμένη αναπαραγωγή. Παράλληλα θα πρέπει να τονίσουμε ότι παρ' ότι χρησιμοποιήθηκε η ίδια διεύθυνση για το streaming, δεν υπήρξε καμία σύγχυση στα πακέτα

και συνεπώς στο αποτέλεσμα του streaming στις εικονικές μηχανές. Συμπερασματικά το streaming σε VLANs χωρίς μεγάλη δικτυακή κίνηση, κρίνεται άκρως ικανοποιητικό με τη χρήση του Open vSwitch ως εικονικού μεταγωγέα.

#### **5.4.8 Έλεγχος απόδοσης συστήματος σε streaming ήχου στην τοπολογία με VLANs μαζί με δικτυακή συμφόρηση.**

Αφού μελετήθηκε η απόδοση του Open vSwitch σε περιπτώσεις streaming βίντεο σε τοπολογία δικτύου VLAN, χωρίς μεγάλο δικτυακό φόρτο, σειρά είχε η μελέτη της συμπεριφοράς του Open vSwitch σε δίκτυο με μεγάλη συμφόρηση. Για το σκοπό του συγκεκριμένου σταδίου θα γίνει χρήση του προγράμματος hping3. Η πλημμύρα πακέτων και σε αυτή την περίπτωση θα αποφευχθεί καθώς δεν επιθυμείται η διακοπή της επικοινωνίας μεταξύ των εικονικών μηχανών και του μεταγωγέα.

Χωρίς να αλλάξουμε την τοπολογία του δικτύου μας, δηλαδή έχοντας δύο VLAN με δύο εικονικές μηχανές το κάθε ένα ξεκινάμε να εισάγουμε δικτυακή κίνηση στον εικονικό μας μεταγωγέα. Και πάλι θα αποφύγουμε τη δημιουργία πλημμύρας πακέτων καθώς χρειαζόμαστε μια σταθερή δομή για τους πειραματισμούς μας. Οι εντολές δημιουργίας και ενσωμάτωσης της δικτυακής ροής στον εικονικό μεταγωγέα δίνονται από τις ακόλουθες δύο εντολές.

Εντολή γεννήτριας πακέτων
<ul style="list-style-type: none"><li>• hping3 --udp --fast 10.1.1.1</li><li>• hping3 --faster 10.1.1.1</li></ul>



Στην περίπτωση αυτή, και όπως αναμενόταν, παρατηρήθηκαν κάποιες διακοπές κατά τη μετάδοση είτε αυτή γινόταν από το ένα VLAN μονάχα, είτε γινόταν και από τα δύο VLAN ταυτόχρονα. Επιπλέον δοκιμάστηκε η αναπαραγωγή του αρχείου από τη μία εικονική μηχανή και η μεταφορά αρχείο με το πρωτόκολλο SFTP από το VLAN2. Σα να εκτελούσαμε δηλαδή διαδικασίες που εμφανίζονται στις καθημερινές χρήσεις μιας συνηθισμένης δικτυακής τοπολογίας. Στην περίπτωση αυτή η απόδοση του εικονικού μεταγωγέα κρίνεται απογοητευτική καθώς ήταν αδύνατη η αναπαραγωγή και κυρίως η παρακολούθηση της εικόνας ενώ αρκετές φορές η σύνδεση διακοπτόταν.

## 5.9 Υλοποίηση 5ου σταδίου: απεικόνιση των λειτουργιών και των επιμέρους μετρήσεων με το λογισμικό sFlow Trend.

Στο κεφάλαιο αυτό θα αναπαρασταθούν γραφικά όλες οι μετρήσεις που έγιναν κατά τη διάρκεια της εκτέλεσης του πειράματος. Για το σκοπό αυτό θα γίνει χρήση του λογισμικού sFlow Trend του οποίου η αρχιτεκτονική έχει αναλυθεί σε προηγούμενο κεφάλαιο. Οι μετρήσεις έγιναν χάρη στη δυνατότητα του Open vSwitch, για δημιουργία ροής δεδομένων από τον μεταγωγέα προς κάποιο πρόγραμμα επεξεργασίας ροής όπως το sFlow Trend. Για την επίτευξη του στόχου μας ήταν απαραίτητη η δημιουργία ροής δεδομένων από τον εικονικό μεταγωγέα προς το πρόγραμμα επεξεργασίας των πακέτων. Η ροή αυτή δημιουργείται από τον μεταγωγέα με χρήση της εντολής `ovs-vsctl`. Η τεχνολογία sFlow αποτελείται από αρκετές συσκευές οι οποίες εκτελούν δύο διαφορετικούς τύπους δειγματοληψίας. Η πρώτη σχετίζεται με την τυχαία δειγματοληψία πακέτων και η δεύτερη με τη περιοδική δειγματοληψία των μετρητών (counter). Τα πακέτα και οι πληροφορίες των μετρητών αποστέλλονται σε flow datagrams σε ένα κεντρικό server που έχει το λογισμικό το οποίο αναλύει και παραθέτει τα αποτελέσματα για την κίνηση του δικτύου. Ο πρώτος τύπος δειγματοληψίας, δηλαδή της δειγματοληψίας πακέτων με ένα συγκεκριμένο ρυθμό δεν παρέχει 100% αξιόπιστα αποτελέσματα, αλλά παρέχει ποσοτική ακρίβεια. Ο δεύτερος τύπος δειγματοληψίας προσδιορίζει πόσο συχνά το δίκτυο στέλνει μετρητές interface.

Τα δειγματοληπτούμενα πακέτα αποστέλλονται με UDP πρωτόκολλο στο interface που έχει προσδιοριστεί και από την πόρτα που έχει ορισθεί. Η επίσημη πόρτα για τη χρησιμοποίηση του sFlow είναι η 6343, πόρτα που θα χρησιμοποιηθεί και στις μετρήσεις μας. Η αναξιοπιστία του UDP πρωτοκόλλου δεν επηρεάζει τα αποτελέσματα, καθώς αν για παράδειγμα χαθούν κάποιοι μετρητές, τότε νέες μετρήσεις θα αποσταλούν όταν ολοκληρωθεί το διάστημα που γίνεται η σταθμοσκόπηση. Η απώλεια πακέτων έχει δίνει πολλή μικρή απόκλιση στον αποτελεσματικό ρυθμό δειγματοληψίας. Το πεδίο payload του πακέτου UDP, περιέχει το sFlow datagram που παρέχει πληροφορίες για την έκδοση, την διεύθυνση πηγής καθώς και πόσα δείγματα περιέχει.

Το λογισμικό sFlow Trend χρησιμοποιήθηκε τόσο για την ευκολία που παρέχει στον τρόπο που παρουσιάζονται τα αποτελέσματα όσο και για το γεγονός ότι είναι ένα ανοιχτό λογισμικό και συνεπώς μπορεί εύκολα να παραμετροποιηθεί. Για τη δημιουργία της ροής η οποία θα μεταφέρει τα πακέτα με τον ρυθμό δειγματοληψίας και το διάστημα σταθμοσκόπησης που θα ορίσουμε, εκτελέσθηκε η ακόλουθη εντολή η οποία εντάχθηκε σε script για την αυτοματοποίηση της όλης διαδικασίας. Η διεύθυνση του collector, δηλαδή του μηχανήματος που θα λάβει τα αρχεία είναι αυτή στην οποία βρίσκεται εγκατεστημένο το πρόγραμμα δηλαδή στον host μας και συγκεκριμένα στο eth0, interface το οποίο είναι συνδεδεμένο στον εικονικό μεταγωγέα μας. Η πόρτα είναι η προεπιλεγμένη ενώ ο ρυθμός δειγματοληψίας ορίσθηκε σε ένα πακέτα ανά 100, ρυθμός που προτείνεται για δίκτυο 100Mbps όπως το δικό μας. Παράλληλα η περίοδος σταθμοσκόπησης ορίσθηκε στα 20 δευτερόλεπτα, δηλαδή κάθε τόσο θα στέλνει τους μετρητές στο πρόγραμμα.

### Δημιουργία sFlow ροής (sFlow.sh)

```
#!/bin/bash
```

```
COLLECTOR_IP=10.1.1.10
```

```
COLLECTOR_PORT=6343
```

```
AGENT_IP=eth0
```

```
HEADER_BYTES=256
```

```
SAMPLING_N=100
```

```
POLLING_SECS=30
```

```
ovs-vsctl --id=@sflow create sflowagent=${AGENT_IP} target=${COLLECTOR_IP}  
${COLLECTOR_PORT} header=${HEADER_BYTES} sampling=${SAMPLING_N}  
polling=${POLLING_SECS} --set bridge br0 sflow=@sflow
```

Με την παραπάνω εντολή δημιουργήθηκε η ροή sFlow η οποία μπορεί να επιβεβαιωθεί μαζί με τα στοιχεία της με την εντολή **ovs-vsctl list sflows**. Στο σημείο αυτό το λογισμικό sFlow Trend έχει τη δυνατότητα να παρέχει γραφικές παραστάσεις και στατιστικά για της τοπολογίες δικτύου που αναλύει. Οι μετρήσεις που λήφθηκαν και θα παρουσιαστούν στη συγκεκριμένη διπλωματική αφορούν στην:

- **Χρησιμοποίηση (Utilization):** Αφορά το ποσοστό της συνολικής χωρητικότητας του εικονικού interface που χρησιμοποιείται (χωρητικότητα γραμμής που χρησιμοποιείται / συνολική χωρητικότητα γραμμής)
- **Μετρητές (Counters):** Αφορά το σύνολο των πλαισίων που διέρχονται από κάθε εικονικό interface της δικτυακής τοπολογίας.
- **Πρωτόκολλα (Protocols):** Αφορά τα πρωτόκολλα με τα οποία μεταφέρονταν τα πακέτα μέσα στο δίκτυο. Καθώς χρησιμοποιήθηκε το hring3 για τη δημιουργία πακέτων αυτά θα πρέπει να εμφανίζονται στις γραφικές παραστάσεις, ενώ θα πρέπει να εμφανισθούν και τα πρωτόκολλα με τα οποία έγινε το streaming και η μεταφορά αρχείων.

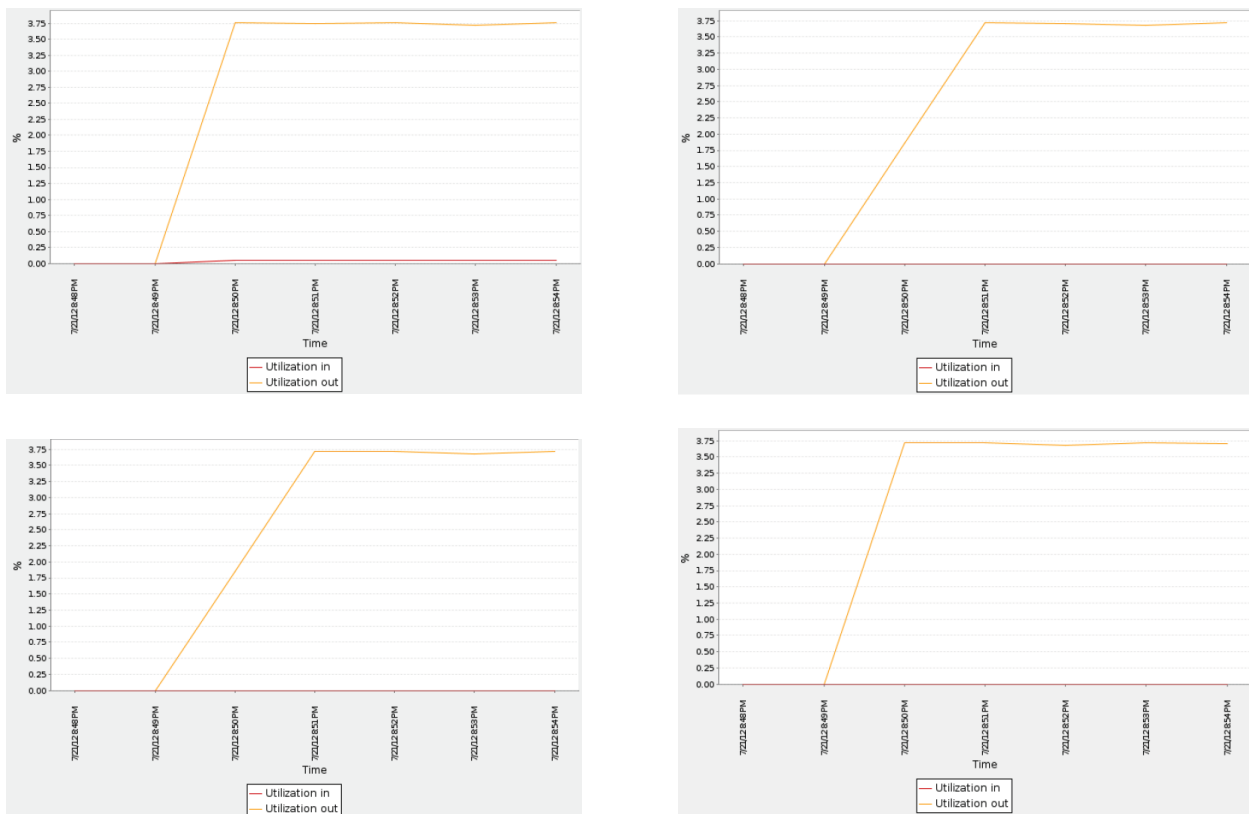
Παρακάτω θα παρουσιασθούν οι γραφικές παραστάσεις για κάθε τοπολογία δικτύου που μελετήθηκε ξεχωριστά και θα αξιολογηθούν τα αποτελέσματα. Αξίζει εδώ να αναφερθεί ότι ορισμένα διαγράμματα δεν ανταποκρίνονται σε 100% ακριβείς μετρήσεις καθώς τόσο η περιορισμένη επεξεργαστική ισχύς, όσο και ο τρόπος με τον οποίο γίνονται οι μετρήσεις (δηλαδή με τη μέθοδο της δειγματοληψίας) δεν οδηγούν σε ασφαλή συμπεράσματα. Ωστόσο μπορεί κανείς εύκολα να επιβεβαιώσει τα οπτικοακουστικά αποτελέσματα, μέσα από τις γραφικές παραστάσεις των εικονικών διεπαφών.

## 5.9.1 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου.

Παρακάτω θα παρουσιαστούν τα αποτελέσματα σε γραφικές παραστάσεις της περίπτωσης streaming ήχου τόσο στην απλή τοπολογία δικτύου με τις τέσσερις εικονικές μηχανές, όσο και στην περίπτωση streaming ήχου σε VLAN. Παράλληλα θα απεικονισθούν τα αποτελέσματα για τις περιπτώσεις streaming χωρίς δικτυακή κίνηση, καθώς και streaming με δικτυακή κίνηση.

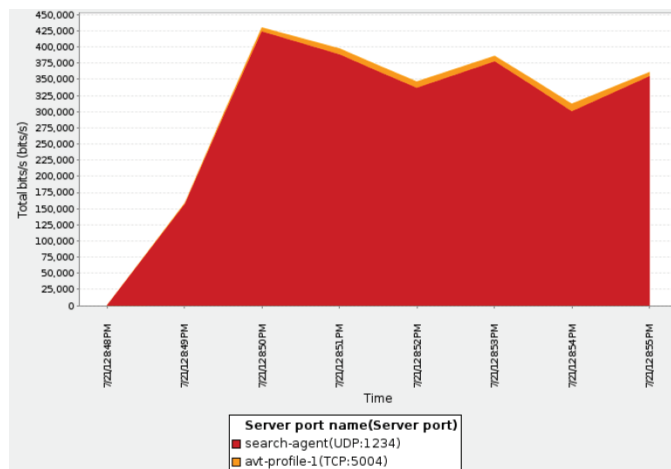
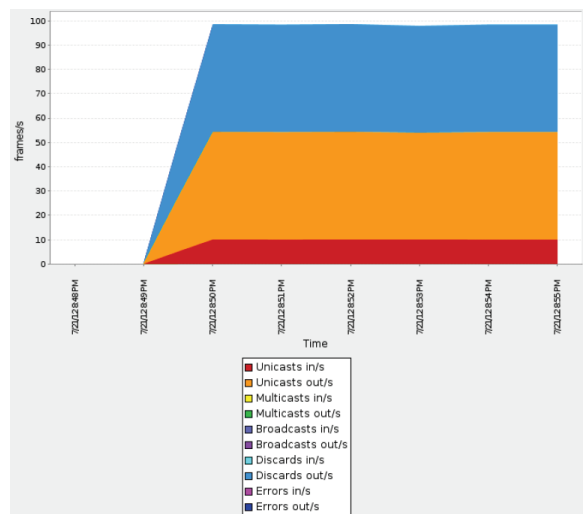
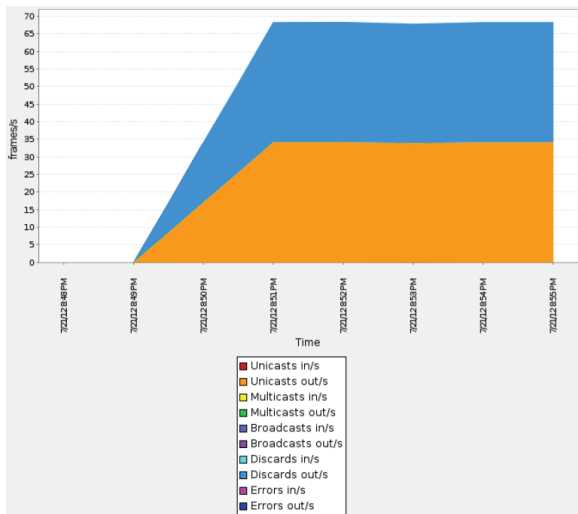
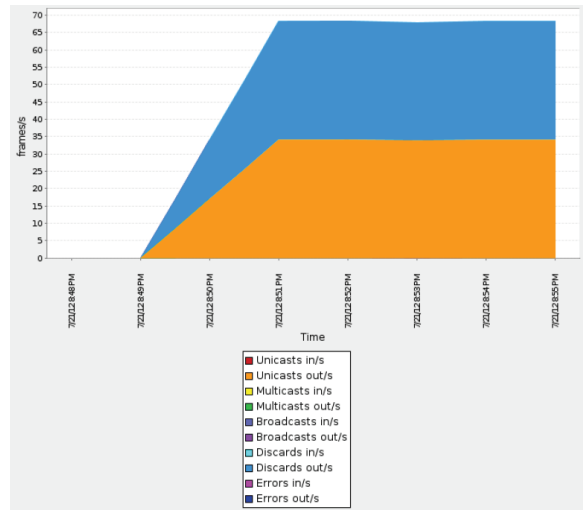
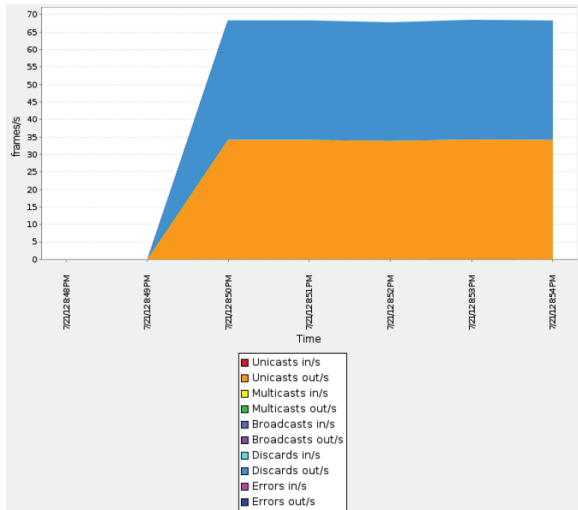
### 5.9.1.1 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε απλή τοπολογία χωρίς δικτυακή συμφόρηση.

Ακολούθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming ήχου χωρίς δικτυακή συμφόρηση.



**Γράφημα 1-4: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Παρατηρούμε ότι κατά την απλή μετάδοση του αρχείου ήχου η χρησιμοποίηση των interfaces δεν είναι μεγάλη καθώς φτάνει σε όλες τις γραμμές κατά την ταυτόχρονη μετάδοση στο 3.75%, γεγονός που αποδεικνύεται από την άριστη απόδοση του ήχου.



**Γράφημα 5-9: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (bits/min)**

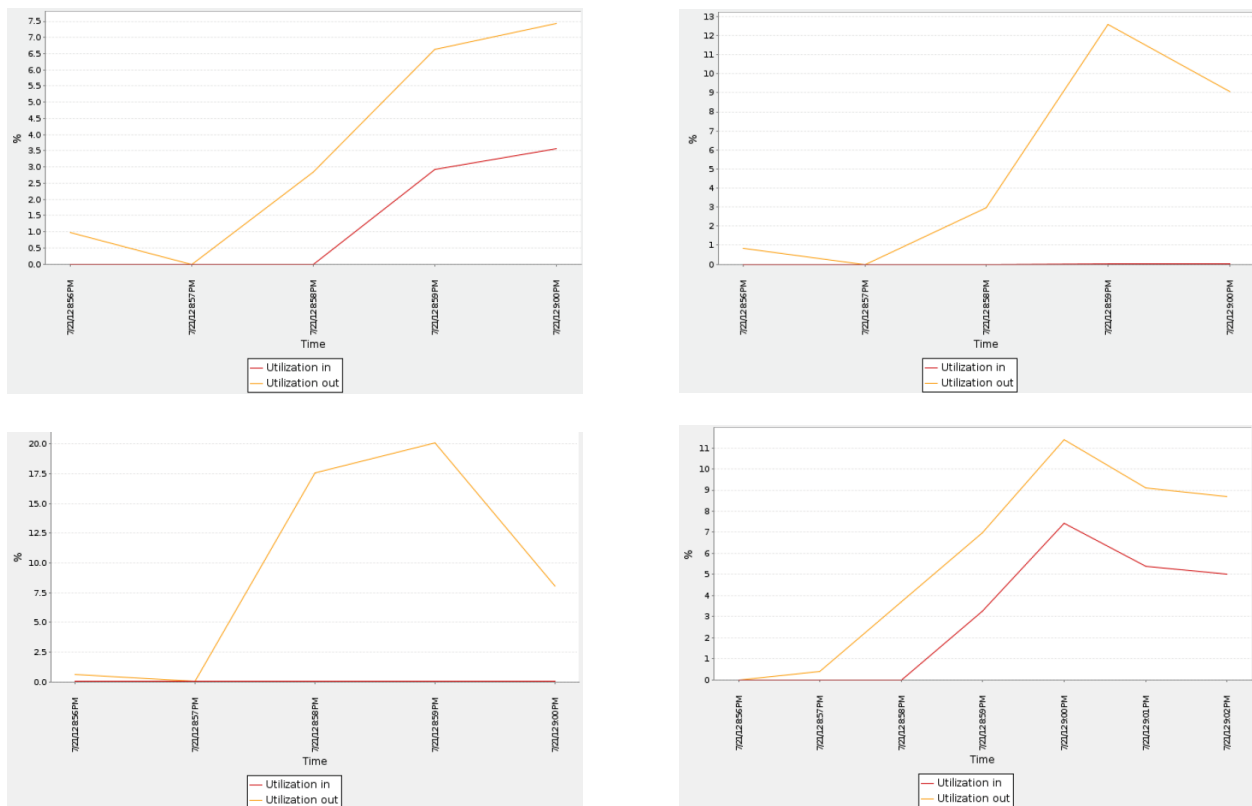
Στις τέσσερις πρώτες γραφικές παραστάσεις, παρατηρούμε τα πλαίσια που μεταδίδονται σε κάθε interface αντίστοιχα (η οριζόντια γραμμή δίνει το χρόνο και η κάθετη τα πλαίσια που μεταδίδονται). Η



πορτοκαλί γραμμή αφορά στα πλαίσια που μεταδίδονται προς το κάθε interface και είναι ουσιαστικά τα πλαίσια ήχου, η μπλε γραμμή αφορά τα πακέτα που απορρίπτονται από τη διαδικασία της δειγματοληψίας ενώ στο τελευταίο διάγραμμα παρουσιάζονται τα πρωτόκολλα που χρησιμοποιήθηκαν κατά τη διάρκεια του πειράματος (η οριζόντια γραμμή δίνει το χρόνο και η κάθετη τα συνολικά bits κάθε πρωτοκόλλου. Διαιρώντας τα συνολικά πακέτα με το χρόνο μπορούμε να έχουμε μια εκτίμηση της ταχύτητας μετάδοσης κάθε χρονική στιγμή.

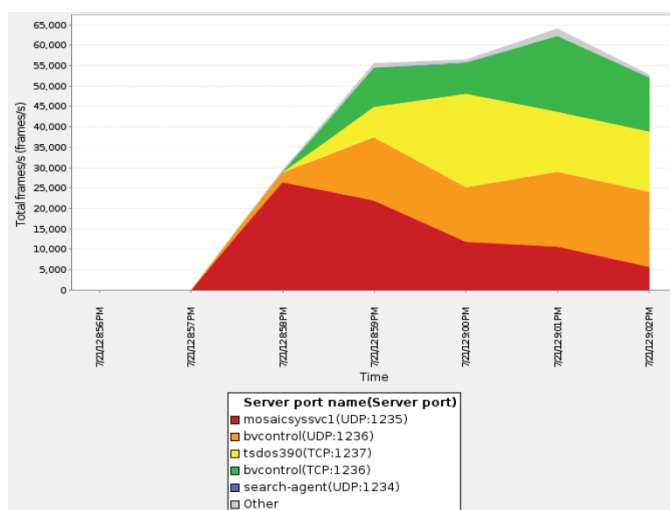
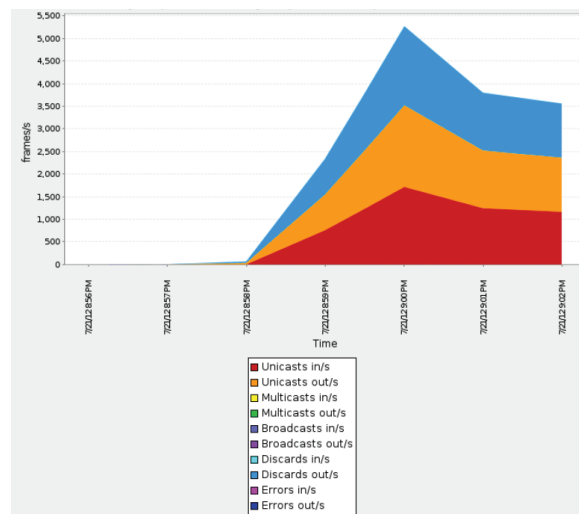
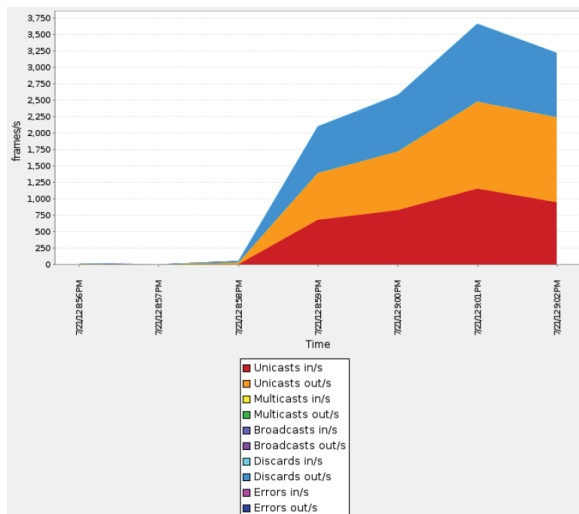
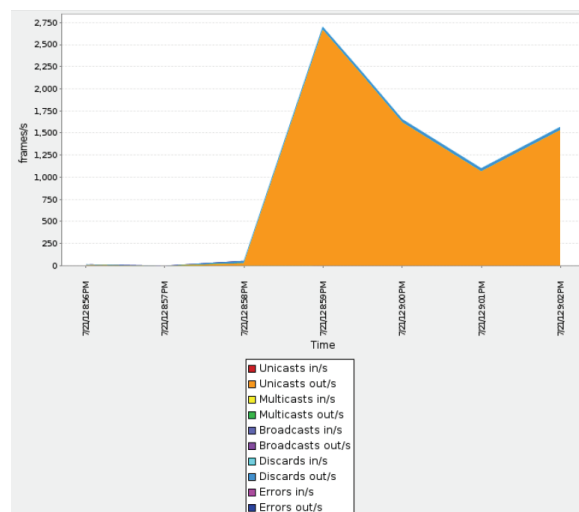
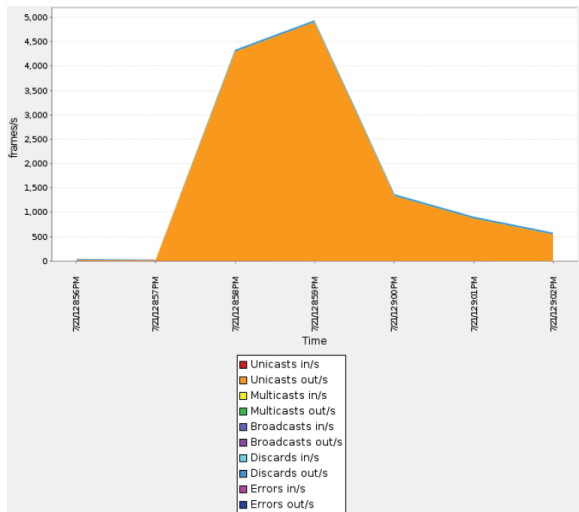
### 5.9.1.2 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε απλή τοπολογία με δικτυακή συμφόρηση.

Ακολουθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming ήχου με δικτυακή συμφόρηση.



**Γράφημα 9-12: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Καθώς εισάγαμε δικτυακή κίνηση φυσικό επακόλουθο ήταν η αύξηση της χρησιμοποίησης των virtual interfaces. Παρατηρούμε ότι αυξάνεται αρκετά η χρησιμοποίηση της γραμμής ενώ στις εικονικές μηχανές 3 και 4 παρατηρούμε ότι έχουμε και χρησιμοποίηση σε εξωτερική κίνηση καθώς η εντολή γέννησης πακέτων απαιτούσε και απάντηση από τον δέκτη των πακέτων και για το λόγο αυτό παρατηρείται διαφοροποίηση σε σχέση με την εντολή γέννησης πακέτων των interfaces των εικονικών μηχανών 1 και 2.

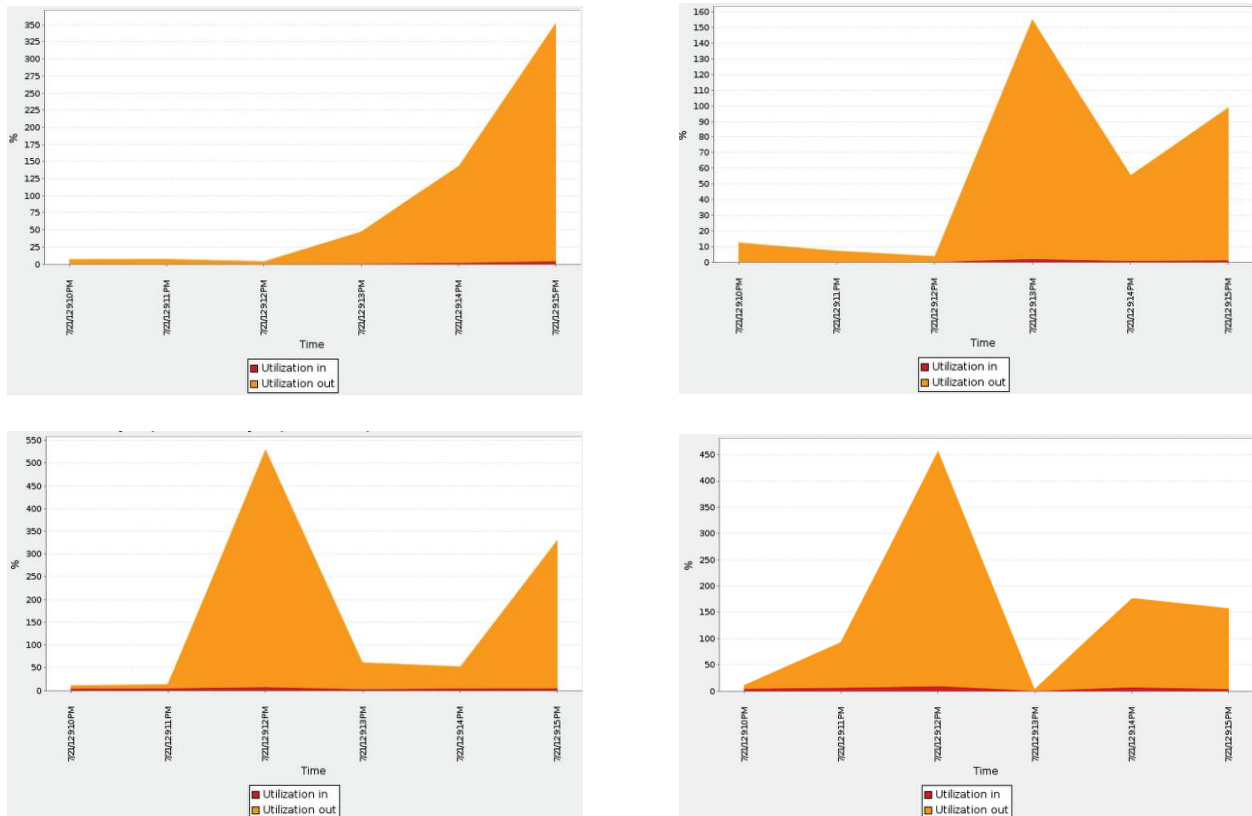


**Γράφημα 12-16: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (frames/min)**

Από τις παραπάνω γραφικές παραστάσεις συμπεραίνουμε ότι παρατηρούμε και πάλι τη δικτυακή κίνηση καθώς και το είδος των πακέτων που δημιουργήθηκαν από το λογισμικό. Η ταχύτητα αποστολής κρίνεται ικανοποιητική ενώ η άριστη απόδοση ήχου επαληθεύεται από το μικρό αριθμό απωλειών πακέτων και τη μικρή σχετικά χρησιμοποίηση της κάθε γραμμής.

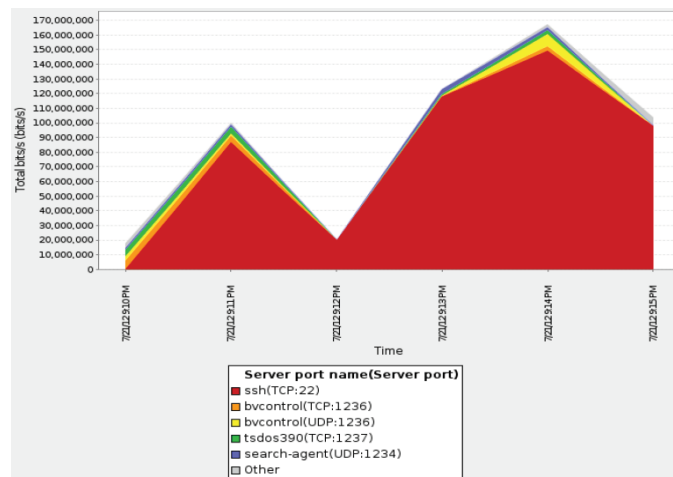
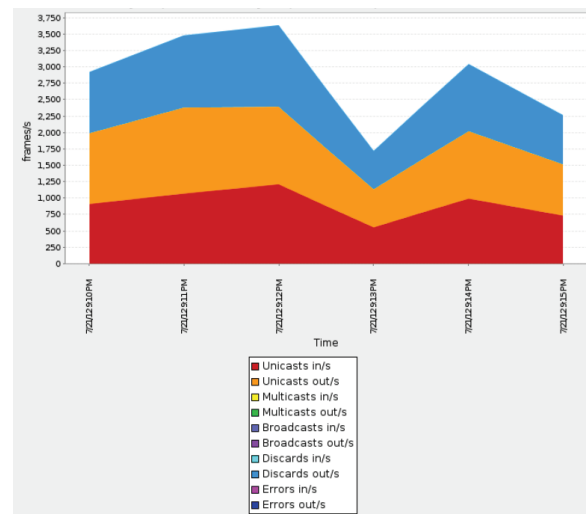
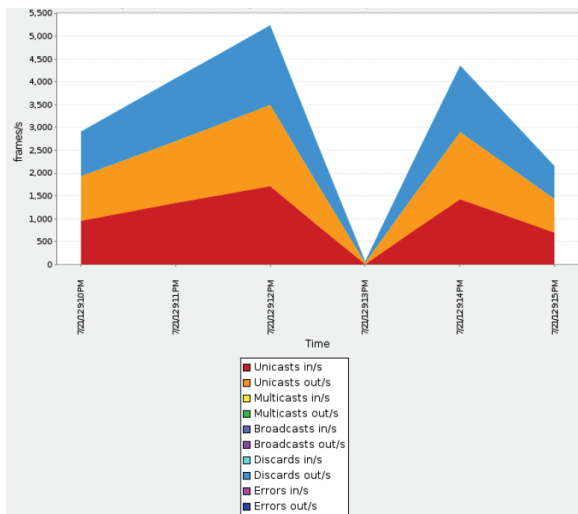
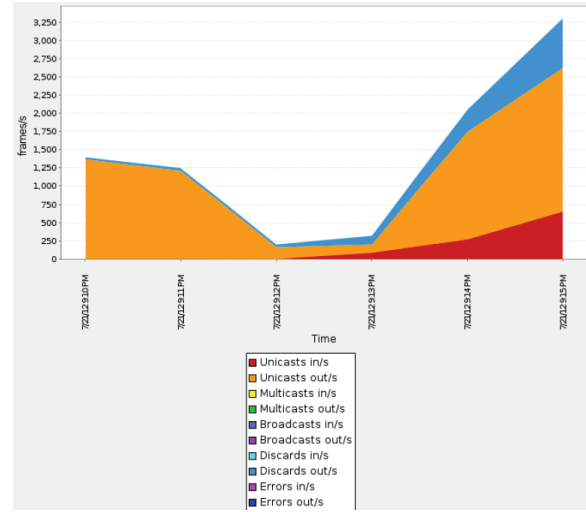
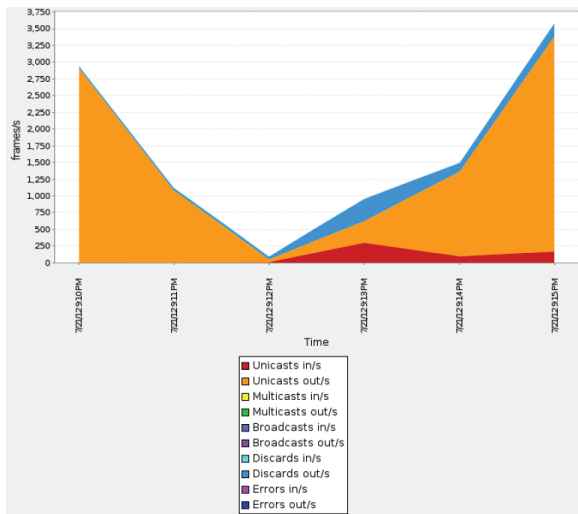
### 5.9.1.3 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε απλή τοπολογία με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείου.

Ακολουθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming ήχου με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείου από τον Host προς τις εικονικές μηχανές.



**Γράφημα 16-19: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Εδώ παρατηρούμε ότι η χρησιμοποίηση της γραμμής ξεπερνάει κατά πολύ το θεωρητικό όριο του 95%. Αυτό συμβαίνει διότι το sFlow Trend θεωρεί τα εικονικά interfaces ως κάρτες δικτύου οι οποίες έχουν δεδομένη ταχύτητα και συγκεκριμένα 10Mbps. Ωστόσο αυτή είναι μάλλον μία θεωρητική τιμή αφού οι ταχύτητες που μπορεί να πιάσει είναι σαφώς μεγαλύτερες. Στην ακραία περίπτωση που έχουμε συμφόρηση και μεταφορά μεγάλου αρχείου που είναι μία επίπονη διαδικασία για το δίκτυο επαληθεύονται και πάλι τα πειραματικά αποτελέσματα αφού παρ' ότι η απόδοση στο streaming ήταν άριστη στο μεγαλύτερο μέρος της, υπήρξαν περιπτώσεις διακοπής της αναπαραγωγής.

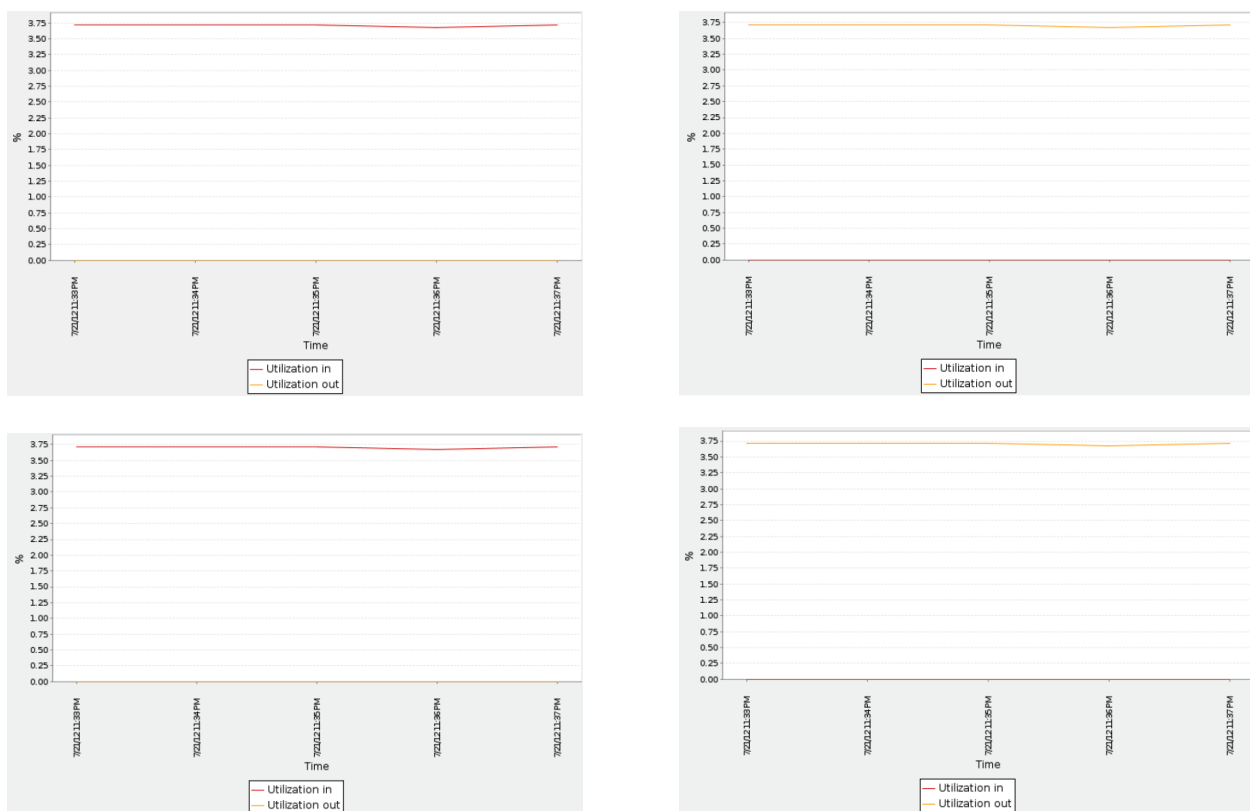


**Γράφημα 20-24: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (bits/min)**

Κατά τα γνωστά στις παραπάνω γραφικές παραστάσεις εμφανίζονται τα πλαίσια που κινήθηκαν στο δίκτυο καθώς και τα πλαίσια που απορρίφθηκαν κατά τη δειγματοληψία. Παράλληλα εμφανίζεται ο συνολικός αριθμός των πλαισίων από κάθε πρωτόκολλο. Παρατηρείται επίσης μία απότομη διακοπή στη ροή των πακέτων. Αυτό συνέβη εξαιτίας της μεγάλης συμφόρησης καθώς στιγμιαία χάθηκε η επικοινωνία μεταξύ των εικονικών μηχανών 1 και 3 και του host.

#### 5.9.1.4 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε τοπολογία VLAN χωρίς δικτυακή συμφόρηση.

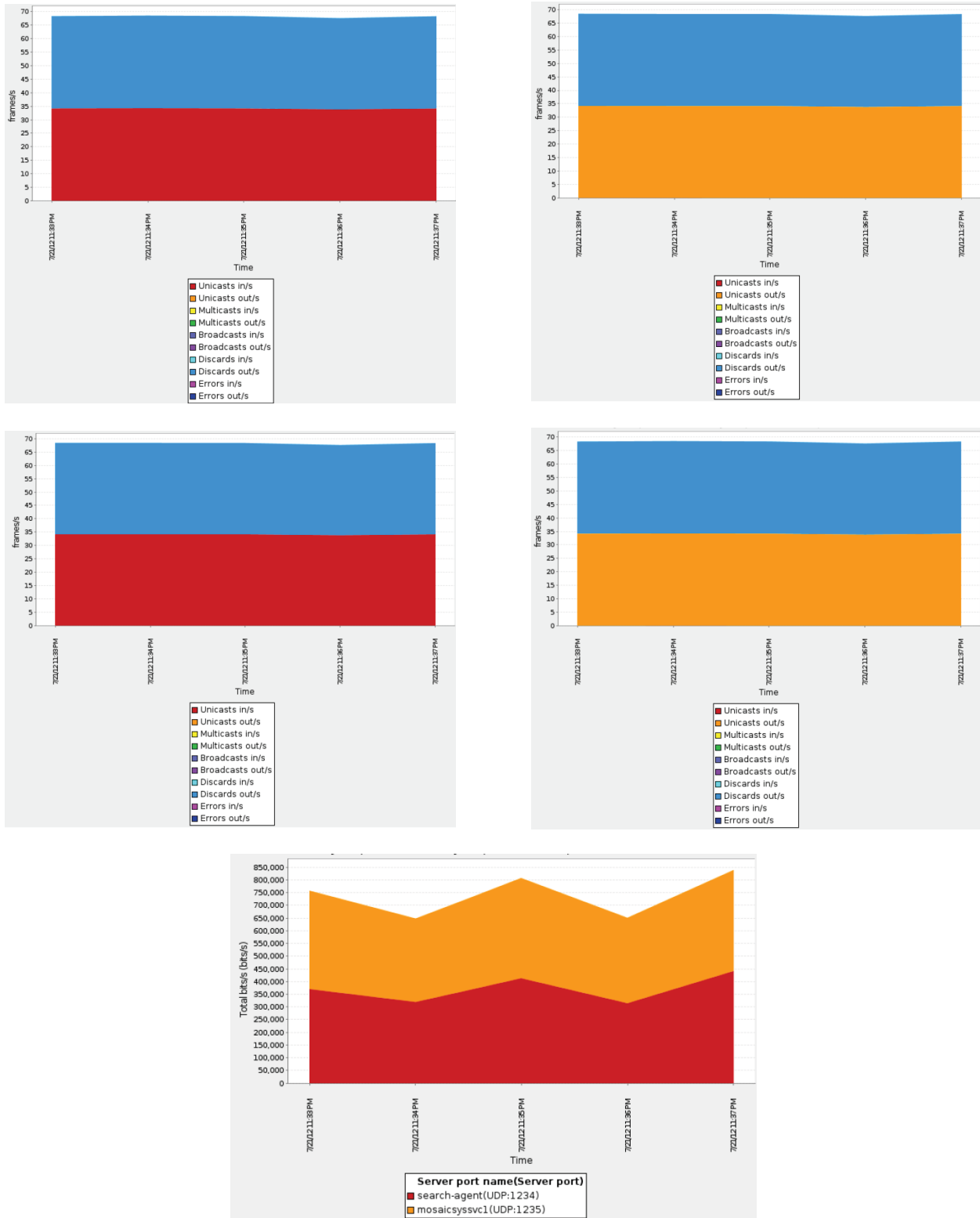
Ακολούθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming ήχου σε τοπολογία VLAN. Στην περίπτωση αυτή έχουμε streaming μεταξύ VM1 – VM4 και VM3 – VM4.



**Γράφημα 24-27: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Στο συγκεκριμένο κομμάτι του πειράματος δεν έχουμε ροή ήχου από μία πηγή σε πολλούς προορισμούς, αλλά συγκεκριμένη ροή από το ένα εικονικό μηχάνημα στο άλλο που βρίσκονται στο ίδιο VLAN. Για το λόγο αυτό στις γραφικές παραστάσεις εμφανίζεται στο interface του VM1 και του VM3 κόκκινη γραμμή χρησιμοποίησης που σημαίνει πληροφορία που φεύγει από την εικονική μηχανή ενώ στα VM2 και VM4 κίτρινη γραμμή που σημαίνει ότι η γραμμή χρησιμοποιείται λόγω εισόδου πακέτων. Παρατηρούμε επίσης ότι και σε τοπολογία VLAN η χρησιμοποίηση παραμένει πολύ μικρή σε όλα τα

interfaces και ίδια περίπου με αυτή της απλής τοπολογίας. Τα αποτελέσματα ανταποκρίνονται απόλυτα με το αποτέλεσμα που είχαμε κατά την εκτέλεση του πειράματος.

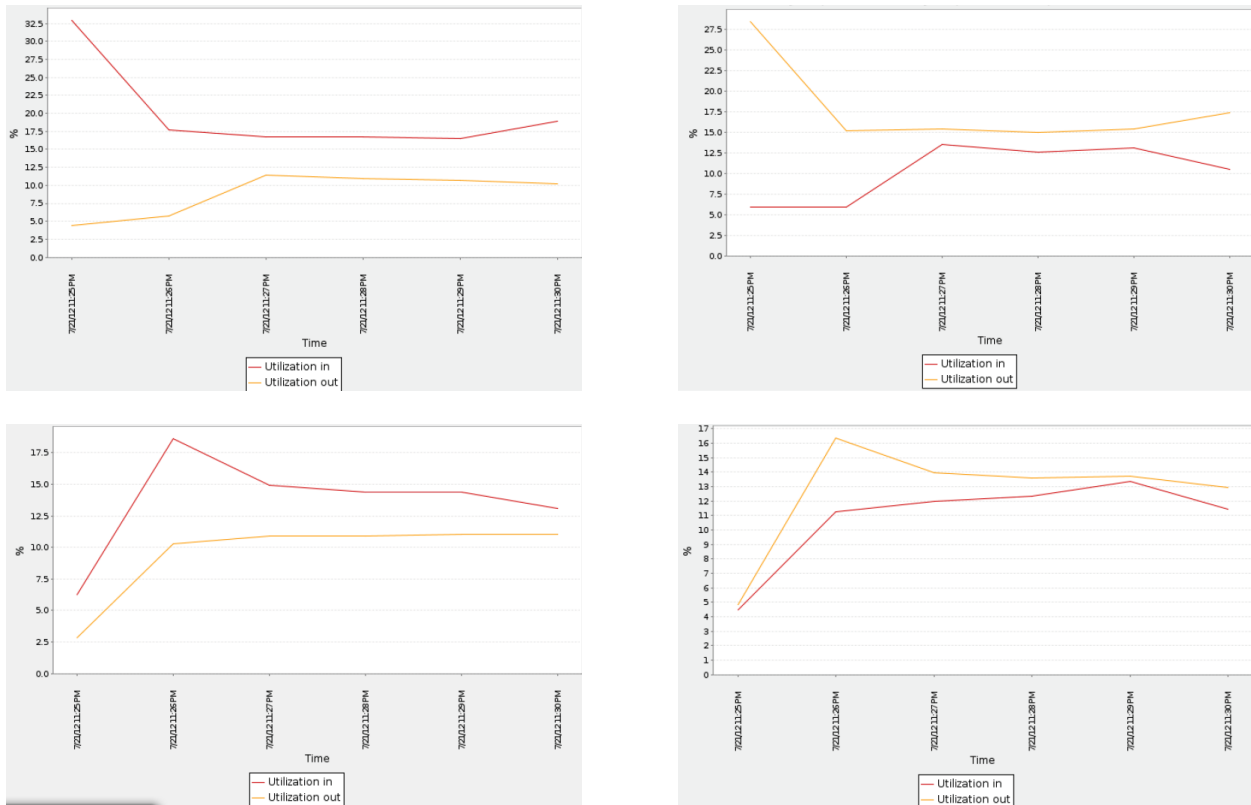


**Γράφημα 28-32: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (bits/min)**

Στις παραπάνω γραφικές παραστάσεις παρουσιάζονται τα πλαίσια που εξέρχονται και εισέρχονται στα εικονικά interfaces καθώς και τα πλαίσια που απορρίπτονται από τη διαδικασία δειγματοληψίας. Στο τελευταίο διάγραμμα παρουσιάζονται τα πρωτόκολλα που χρησιμοποιούνται και τα συνολικά πακέτα που στέλνουν. Παρατηρούμε ότι η ταχύτητα που στέλνουν τα δύο εικονικά μηχανήματα παρ' ότι θεωρούνται απομονωμένα είναι ίδια γεγονός που μας οδηγεί στο συμπέρασμα ότι το Open vSwitch διαχειρίζεται τα VLAN με τον ίδιο ακριβώς τρόπο.

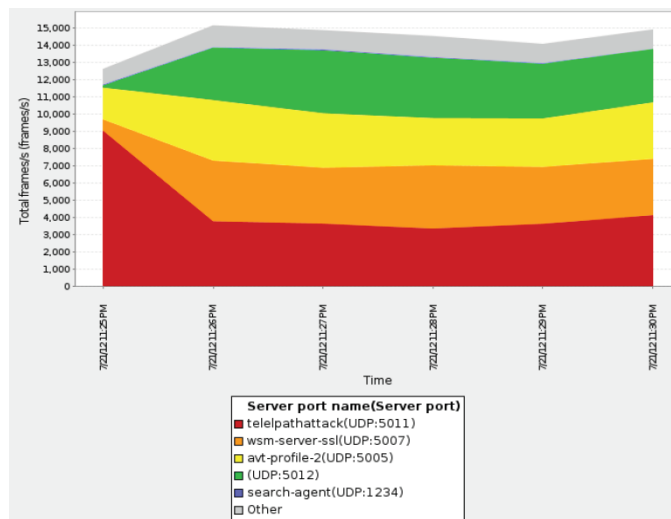
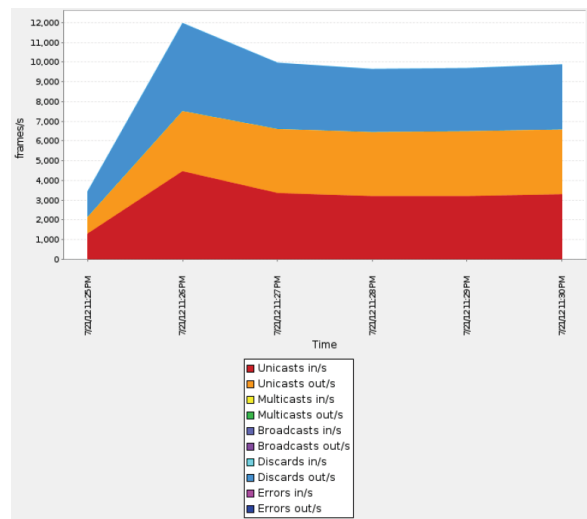
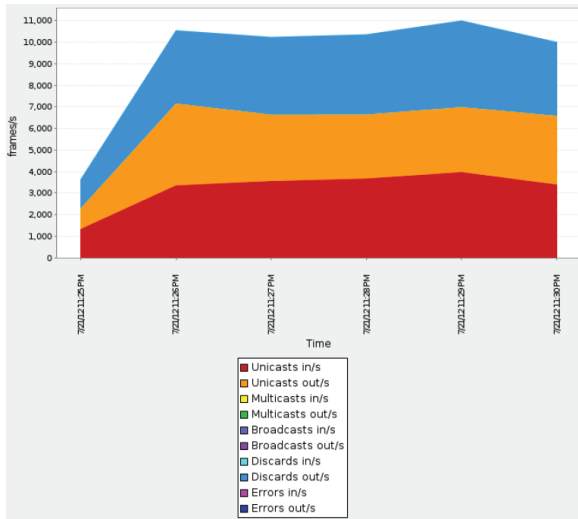
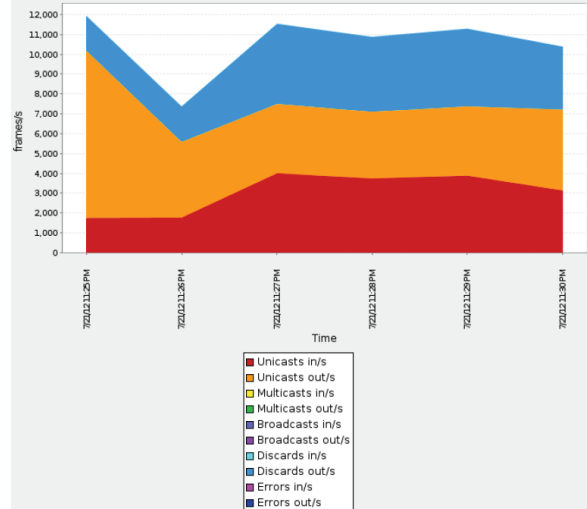
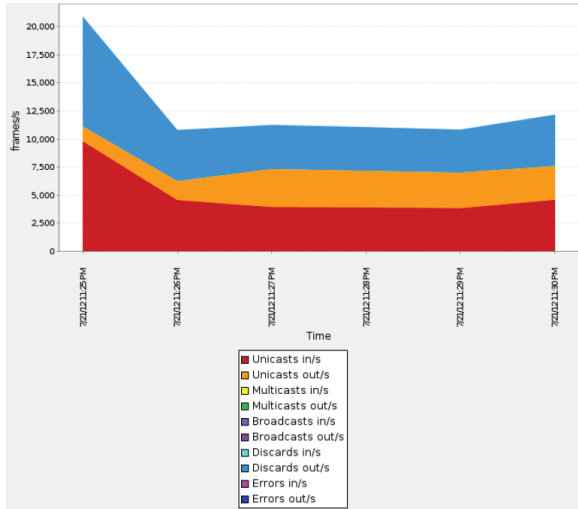
### 5.9.1.5 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε τοπολογία VLAN με δικτυακή συμφόρηση.

Ακολούθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming ήχου σε τοπολογία VLAN με δικτυακή συμφόρηση.



**Γράφημα 33-35: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

ι παραπάνω γραφικές παραστάσεις εμφανίζουν δύο γραμμές κατά τη χρησιμοποίηση της γραμμής. Η μία αφορά τη δημιουργία συμφόρησης στο δίκτυο και η άλλη αφορά τη διαδικασία streaming ήχου από το ένα εικονικό μηχανήματα στο άλλο. Στο σημείο αυτό παρατηρούμε μία αύξηση στη χρησιμοποίηση της γραμμής κατά τη μετάδοση του ήχου γεγονός που επιβεβαιώνεται και από το πείραμα καθώς είχαμε μία με δύο διαλείψεις κατά την αναπαραγωγή. Ωστόσο καθώς ο ήχος είχε άριστη απόδοση συμπεραίνουμε ότι το Open vSwitch σε vlan με συμφόρηση λειτουργεί απροβλημάτιστα.



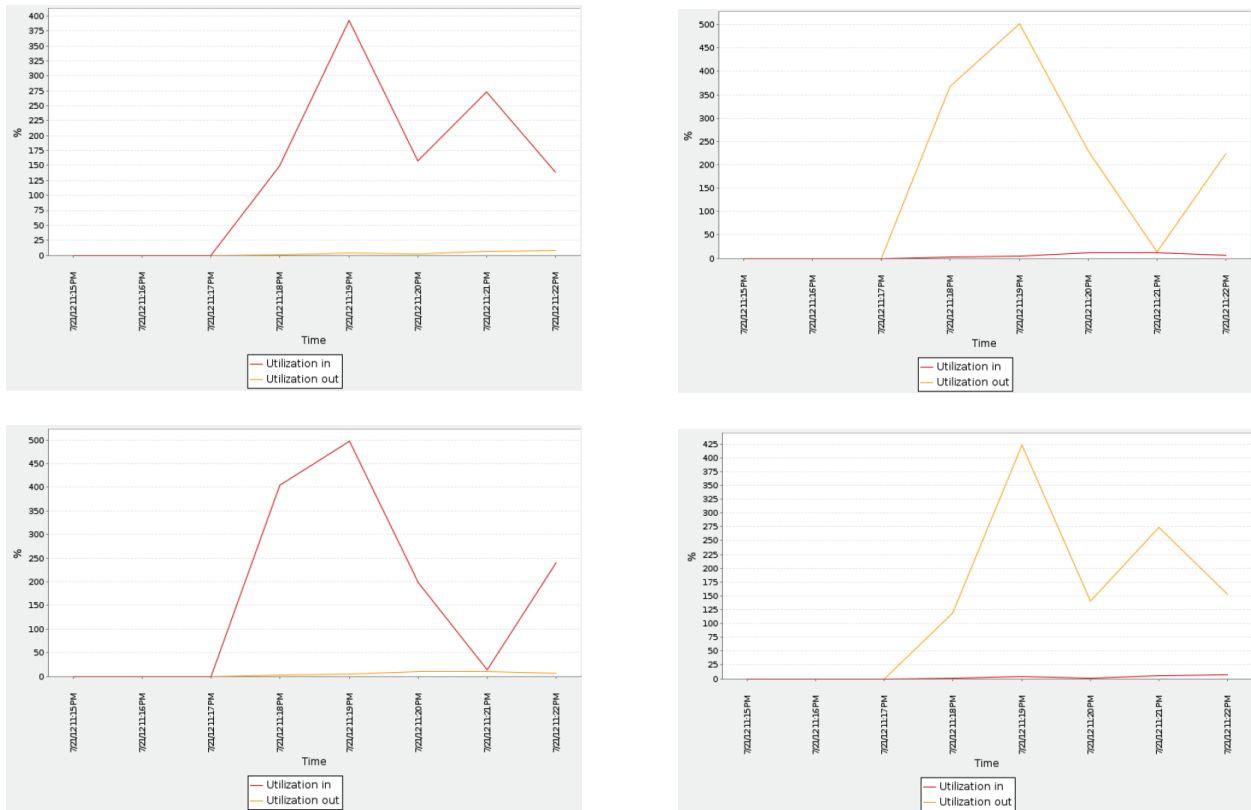
**Γράφημα 36-40: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (frames/min)**



Οι τρεις γραμμές που εμφανίζονται στα τέσσερα πρώτα διαγράμματα σχετίζονται με τις ροές πακέτων από και προς τα εικονικά μηχανήματα. Στο τελευταίο διάγραμμα παρουσιάζονται τα πρωτόκολλα που χρησιμοποιήθηκαν. Εδώ παρατηρήθηκε ένα λάθος κατά την εφαρμογή του πειράματος. Η κίτρινη γραμμή που αφορά τη μετάδοση ήχου στο VLAN2 δε γίνεται με το πρωτόκολλο UDP αλλά με το πρωτόκολλο RTP, το οποίο στην ουσία συμπεριλαμβάνει το UDP. Η μετάδοση του ήχου ωστόσο έγινε χωρίς κανένα πρόβλημα και η απόδοση χαρακτηρίζεται άριστη.

### 5.9.1.6 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε τοπολογία VLAN με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείων.

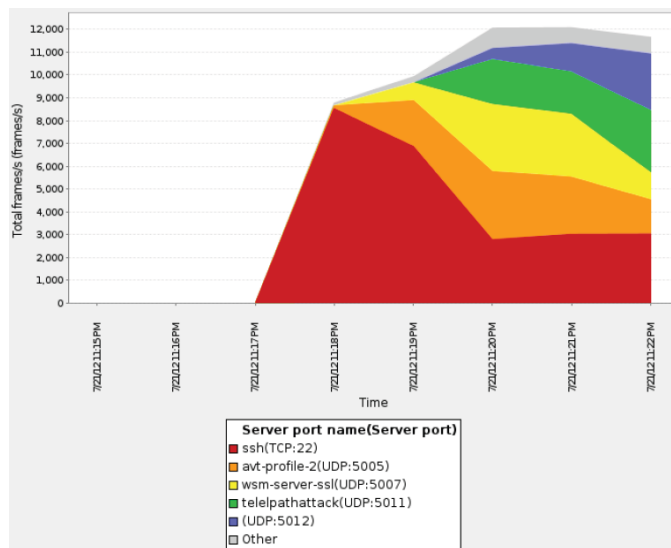
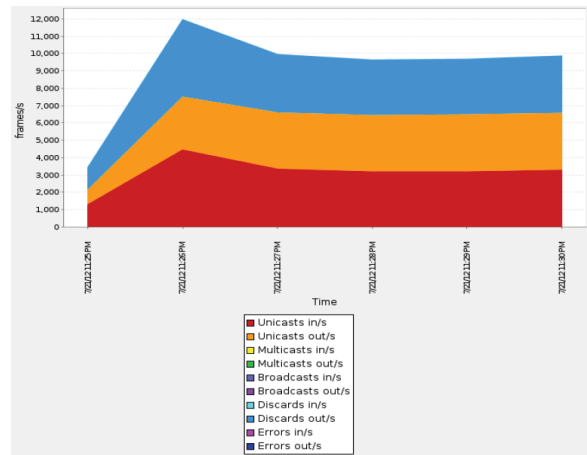
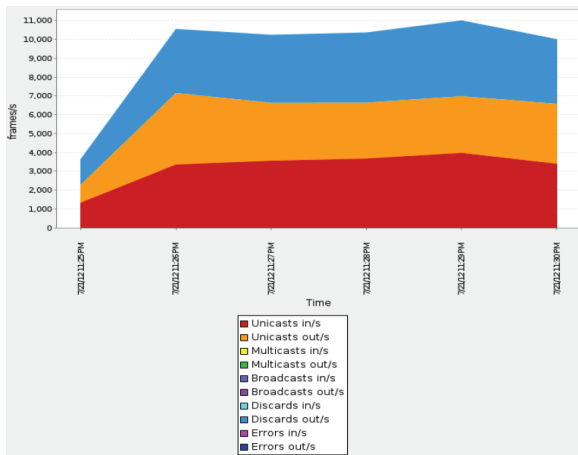
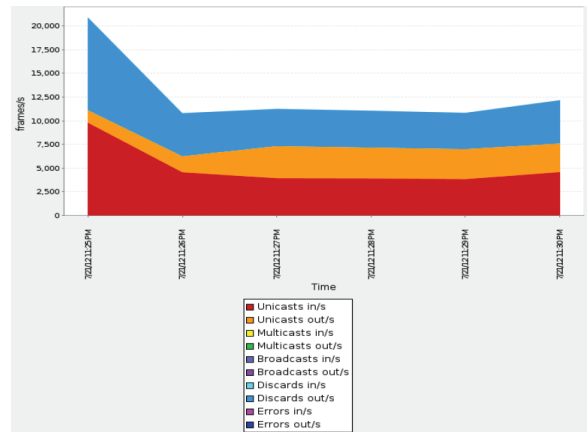
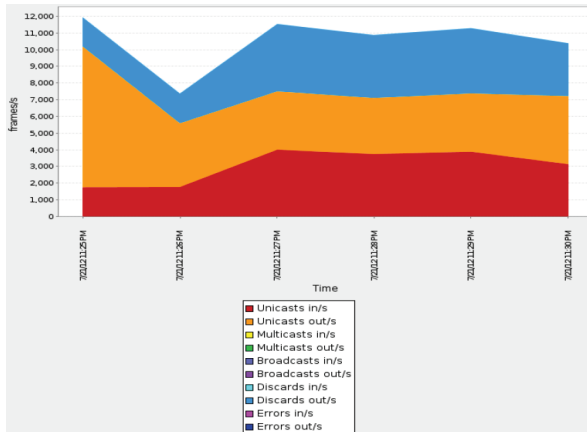
Ακολουθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming ήχου με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείου από το ένα εικονικό μηχανήμα στο άλλο που ανήκουν στο ίδιο VLAN.



**Γράφημα 41-44: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Στο συγκεκριμένο κομμάτι του πειράματος παρατηρείται και πάλι μεγάλη διαφορά στη χρησιμοποίηση του εικονικού interface. Αυτό οφείλεται στο γεγονός ότι ο VM1 και ο VM3 που κάνουν το streaming μοιράζουν ταυτόχρονα και το αρχείο στα δύο εικονικά μηχανήματα. Αυτό δημιουργεί μεγάλη χρησιμοποίηση της γραμμής που στέλνει αλλά και της γραμμής που λαμβάνει τα πακέτα. Παρ' όλα αυτά ο ήχος και πάλι δεν παρουσίασε προβλήματα αν εξαιρέσει κανείς μικρές διακοπές. Ουσιαστικά

καταλήγουμε στο συμπέρασμα ότι σε περιπτώσεις streaming ήχου σε οποιαδήποτε τοπολογία από αυτές που παρουσιάστηκαν, δεν παρουσιάζεται πρόβλημα κατά την αναπαραγωγή.



**Γράφημα 45-49: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (frames/min)**

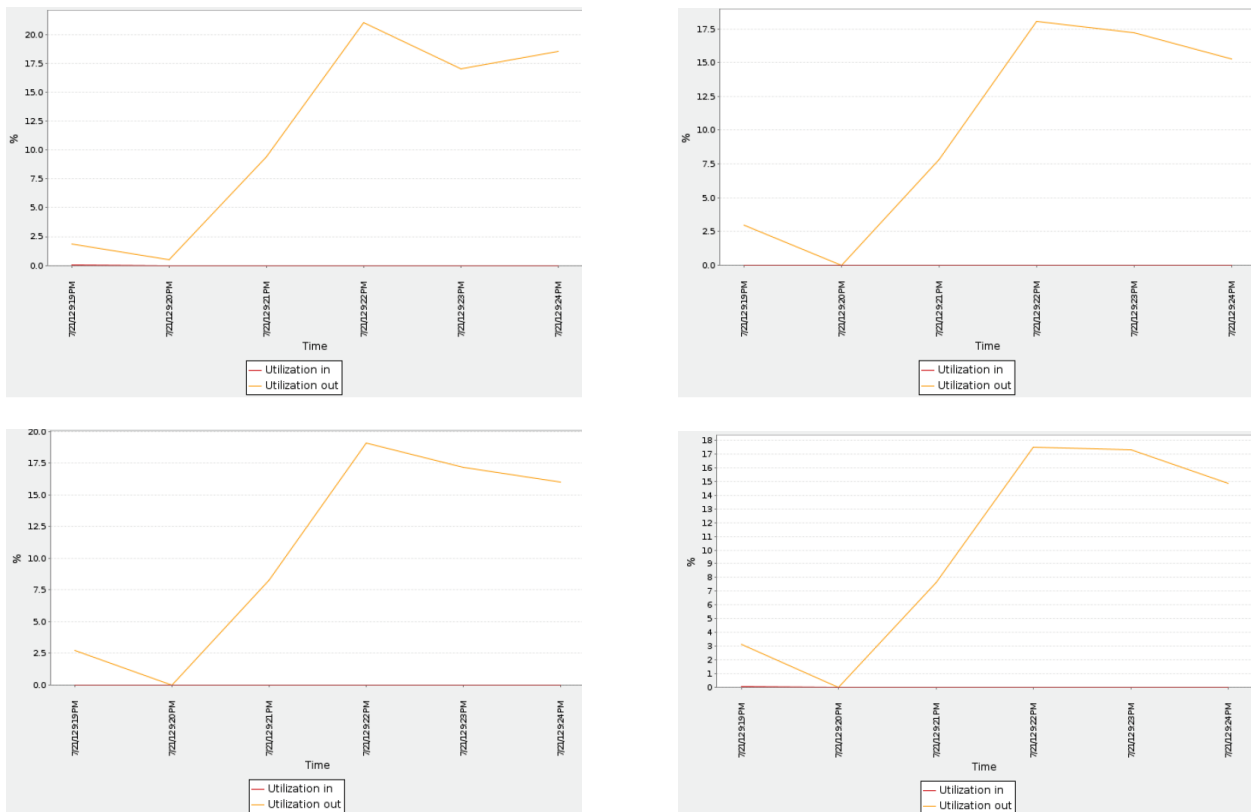
Στο τελευταίο διάγραμμα παρατηρούμε ότι η μεταφορά αρχείου μέσω ssh έχει πάρει το μεγαλύτερο μέρος της κίνησης του συγκεκριμένου πειράματος. Η κίνηση αυτή φαίνεται να έχει και τη μεγαλύτερη προτεραιότητα στην τοπολογία μας και είναι ένα από τα στοιχεία που θα μπορούσαν να προταθούν για αλλαγή ανάλογα με τη χρήση που θέλουμε να γίνεται από τα μηχανήματα.

### 5.5.10 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο.

Παρακάτω θα παρουσιαστούν τα αποτελέσματα σε γραφικές παραστάσεις της περίπτωσης streaming βίντεο, τόσο στην απλή τοπολογία δικτύου με τις τέσσερις εικονικές μηχανές, όσο και στην περίπτωση streaming βίντεο σε VLAN. Παράλληλα θα απεικονισθούν τα αποτελέσματα για τις περιπτώσεις streaming χωρίς δικτυακή κίνηση, καθώς και streaming με δικτυακή κίνηση.

#### 5.5.10.1 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο στην απλή τοπολογία χωρίς δικτυακή συμφόρηση.

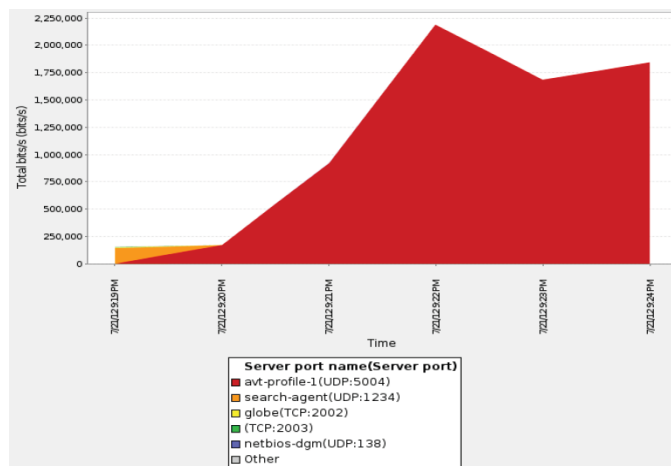
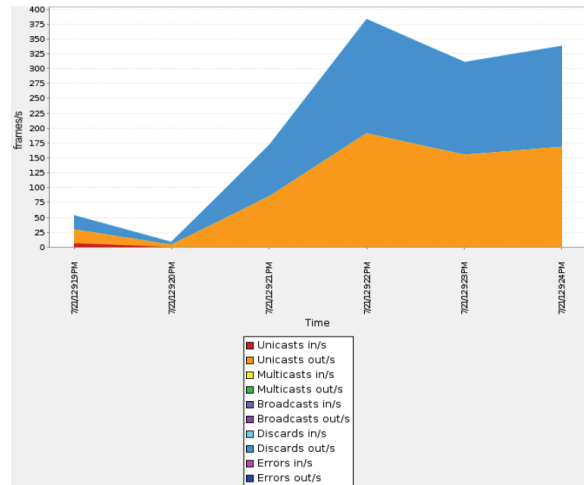
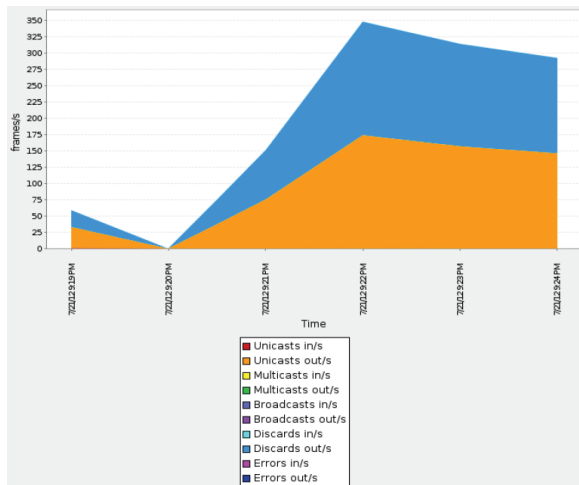
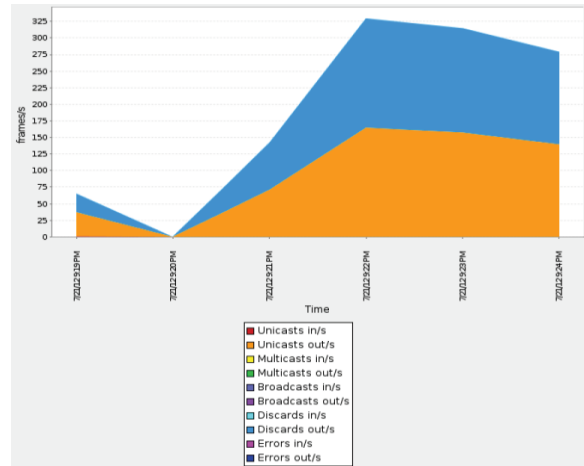
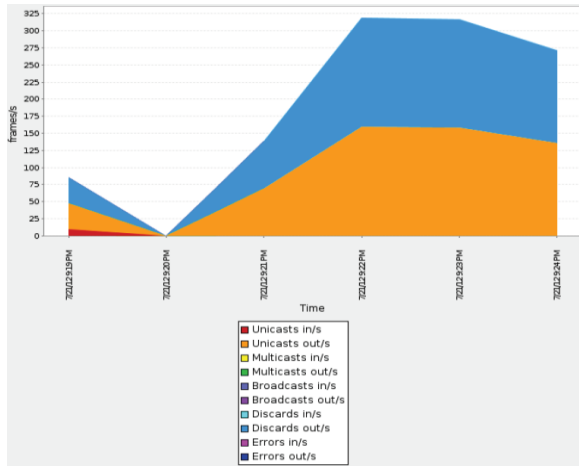
Ακολουθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming βίντεο χωρίς δικτυακή συμφόρηση.



**Γράφημα 50-53: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Παρατηρούμε ότι κατά την απλή μετάδοση του αρχείου βίντεο η χρησιμοποίηση των παρουσιάζει αρκετά μεγάλη διαφορά από την περίπτωση streaming ήχου. Η τιμή της αγγίζει το 18% σε όλα τα interfaces ενώ η απόδοση παρουσιάζει μικρές διαλείψεις στην πιο απλή τοπολογία που μπορεί να

υπάρξει. Το γεγονός αυτό οφείλεται βεβαίως και στις περιορισμένες δυνατότητες του hardware καθώς και στο γεγονός ότι το βίντεο που μεταδίδεται έχει αρκετά μεγάλες απαιτήσεις σε bandwidth.

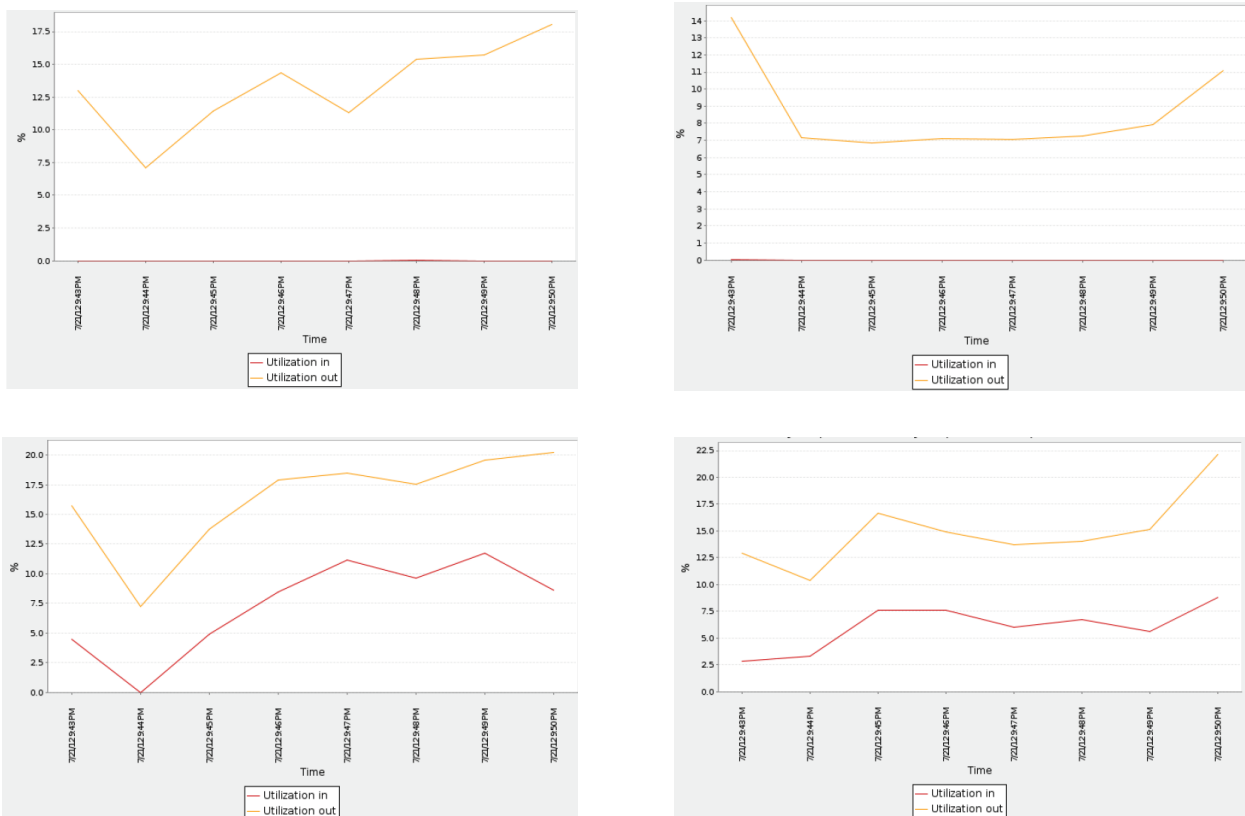


**Γράφημα 54-58: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (bits/min)**

Στις τέσσερις πρώτες γραφικές παραστάσεις, παρατηρούμε τα πλαίσια που μεταδίδονται σε κάθε interface αντίστοιχα. Η πορτοκαλί γραμμή αφορά στα πλαίσια που μεταδίδονται προς το κάθε interface και είναι τα πλαίσια βίντεο, τα οποία προέρχονται από τον host και η μπλε γραμμή αφορά τα πακέτα που απορρίπτονται από τη διαδικασία της δειγματοληψίας. Στο τελευταίο διάγραμμα παρουσιάζονται τα πρωτόκολλα που χρησιμοποιήθηκαν κατά τη διάρκεια του πειράματος, ενώ παρατηρείται ήδη η μεγάλη διαφορά στο αριθμό των bits σε σχέση με την μετάδοση του αρχείου ήχου.

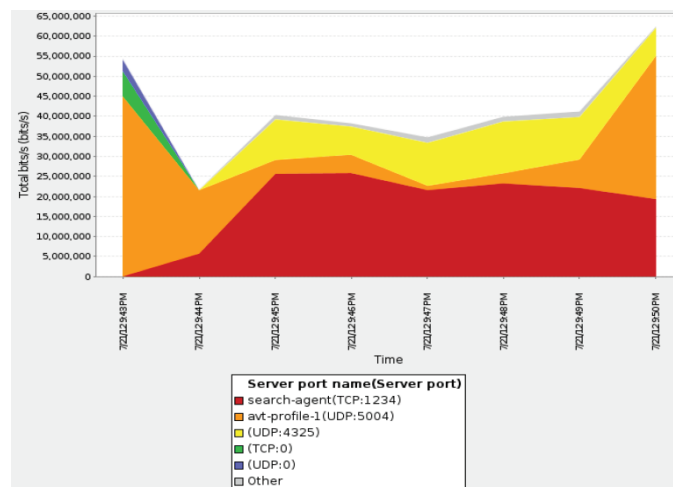
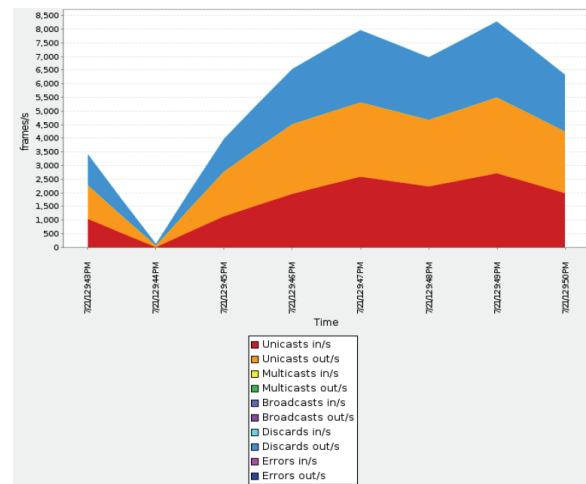
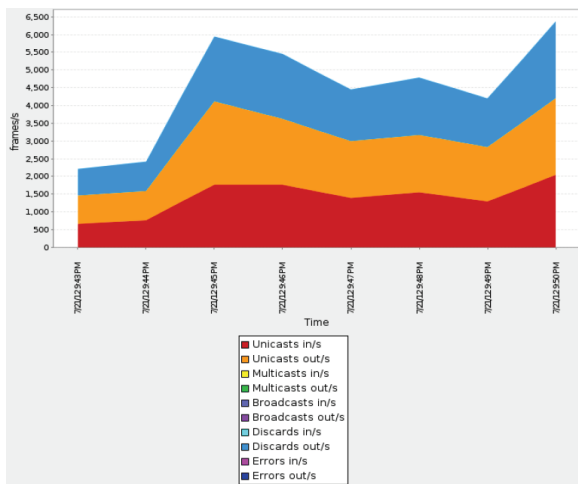
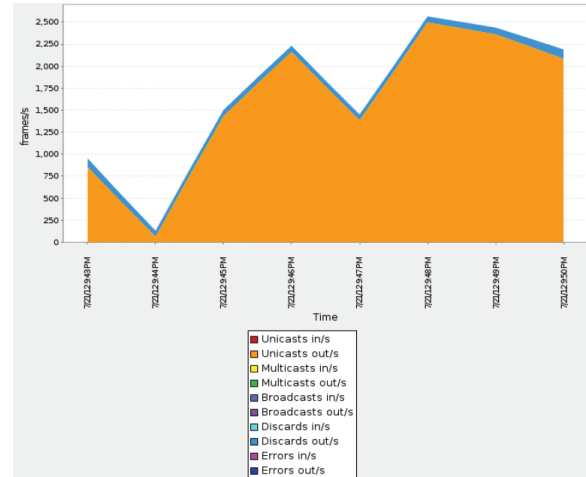
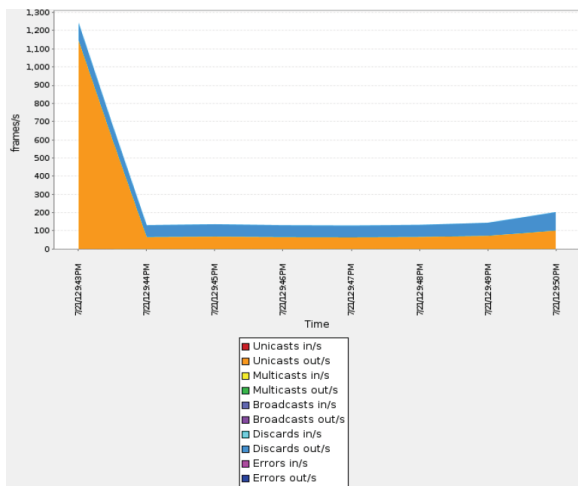
### 5.5.10.2 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming ήχου σε απλή τοπολογία με δικτυακή συμφόρηση.

Ακολούθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming βίντεο με δικτυακή συμφόρηση.



**Γράφημα 58-61: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Καθώς εισάγαμε δικτυακή κίνηση φυσικό επακόλουθο ήταν η αύξηση της χρησιμοποίησης των virtual interfaces. Παρατηρούμε ότι αυξάνεται αρκετά η χρησιμοποίηση της γραμμής ενώ στις εικονικές μηχανές 3 και 4 παρατηρούμε ότι έχουμε και χρησιμοποίηση σε εξωτερική κίνηση καθώς η εντολή γέννησης πακέτων απαιτούσε και απάντηση από τον δέκτη των πακέτων και για το λόγο αυτό παρατηρείται διαφοροποίηση σε σχέση με την εντολή γέννησης πακέτων των interfaces των εικονικών μηχανών 1 και 2. Στο σημείο αυτό η χρησιμοποίηση της γραμμής αγγίζει το 25% και μαζί με την εξωτερική κίνηση το 35%. Ήδη στο σημείο αυτό η απόδοση του βίντεο έχει αρχίσει να μειώνεται τόσο λόγω της συμφόρησης, όσο και λόγω του περιορισμένου hardware.



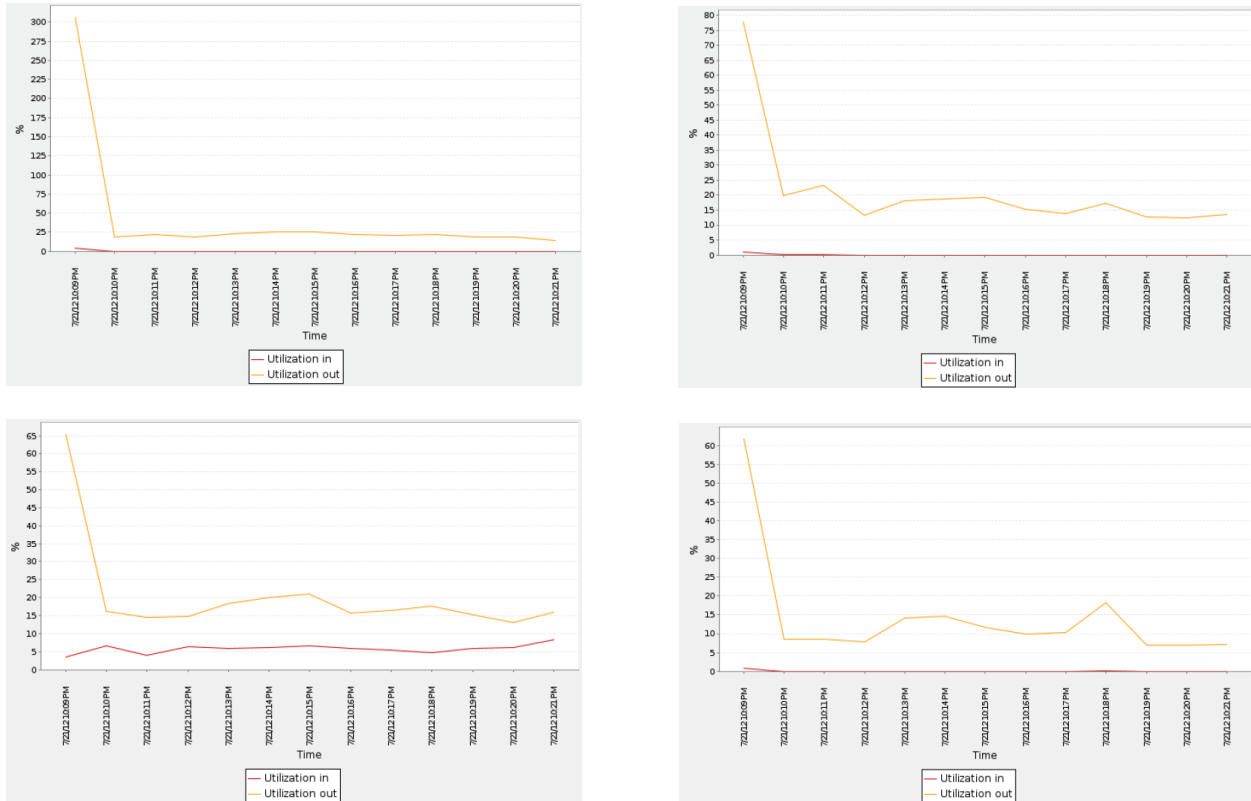
**Γράφημα 62-66: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (bits/min)**

Από τις παραπάνω γραφικές παραστάσεις συμπεραίνουμε ότι παρατηρούμε και πάλι τη δικτυακή κίνηση καθώς και το είδος των πακέτων που δημιουργήθηκαν από το λογισμικό. Η ταχύτητα αποστολής πακέτων κρίνεται αρκετά αυξημένη ενώ όπως παρατηρείται η ταχύτητα που δίνεται για τη μετάδοση

του βίντεο είναι πολύ μικρή σε σχέση με τα υπόλοιπα πρωτόκολλα που μεταφέρονται. Αυτό επαληθεύει την εκτίμηση της μέτριας απόδοσης του βίντεο στην απλή τοπολογία με συμφόρηση.

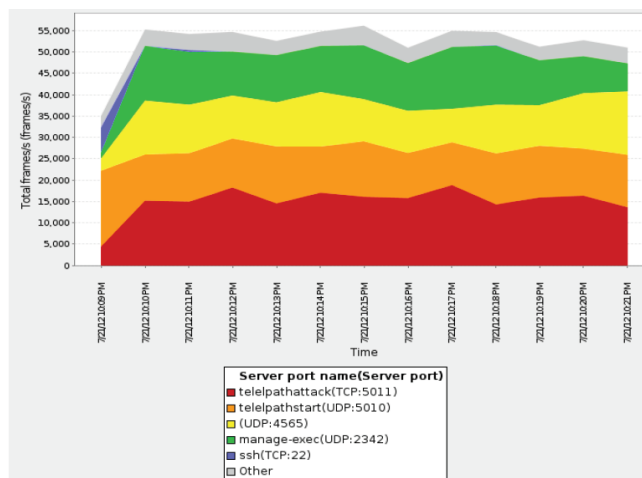
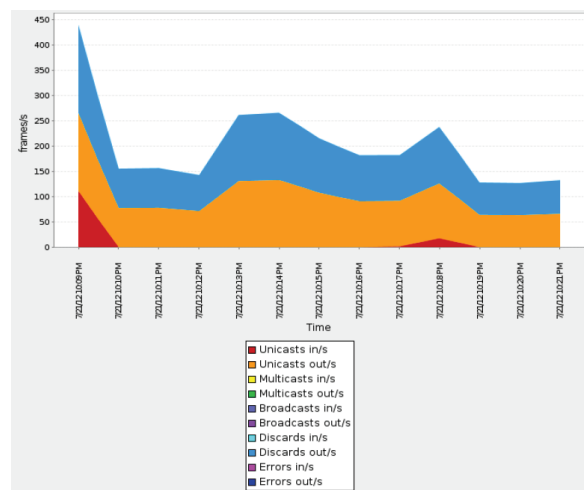
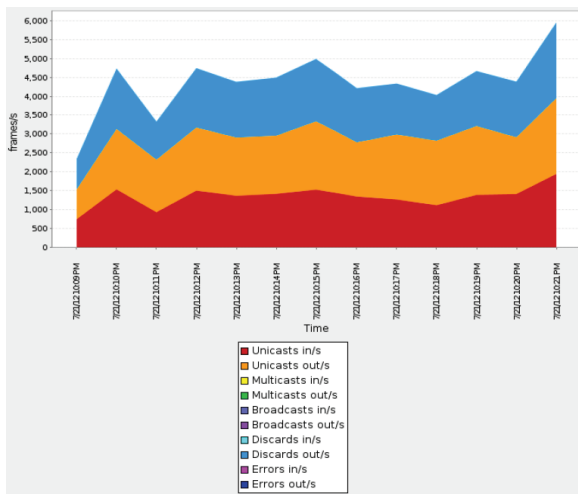
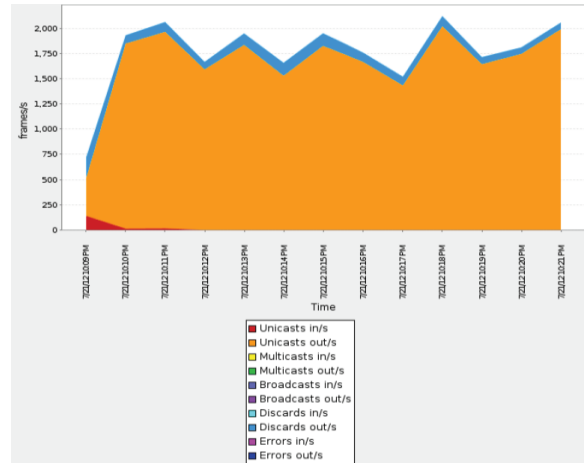
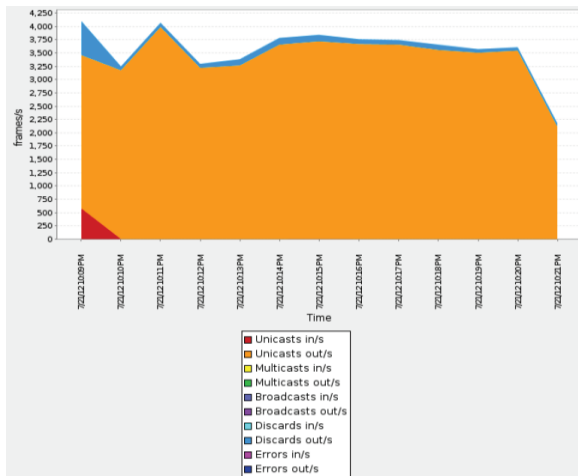
### 5.5.10.3 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο σε απλή τοπολογία με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείου.

Ακολουθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming ήχου με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείου από τον Host προς τις εικονικές μηχανές.



**Γράφημα 67-70: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Στο σημείο αυτό παρατηρούμε ότι η χρησιμοποίηση κυμαίνεται πάλι στα ίδια όρια με πριν. Τούτο δεν κρίνεται φυσιολογικό καθώς σε κάθε περίπτωση streaming βίντεο με συμφόρηση και μεταφορά αρχείου που είδαμε στις προηγούμενες περιπτώσεις, η χρησιμοποίηση της γραμμής αυξανόταν δραματικά. Ωστόσο στις μετρήσεις που ακολούθησαν μπορεί να εξηγηθεί η συγκεκριμένη συμπεριφορά που εμφανίζουν οι συγκεκριμένες γραφικές παραστάσεις.



**Γράφημα 71-75: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (frames/min)**

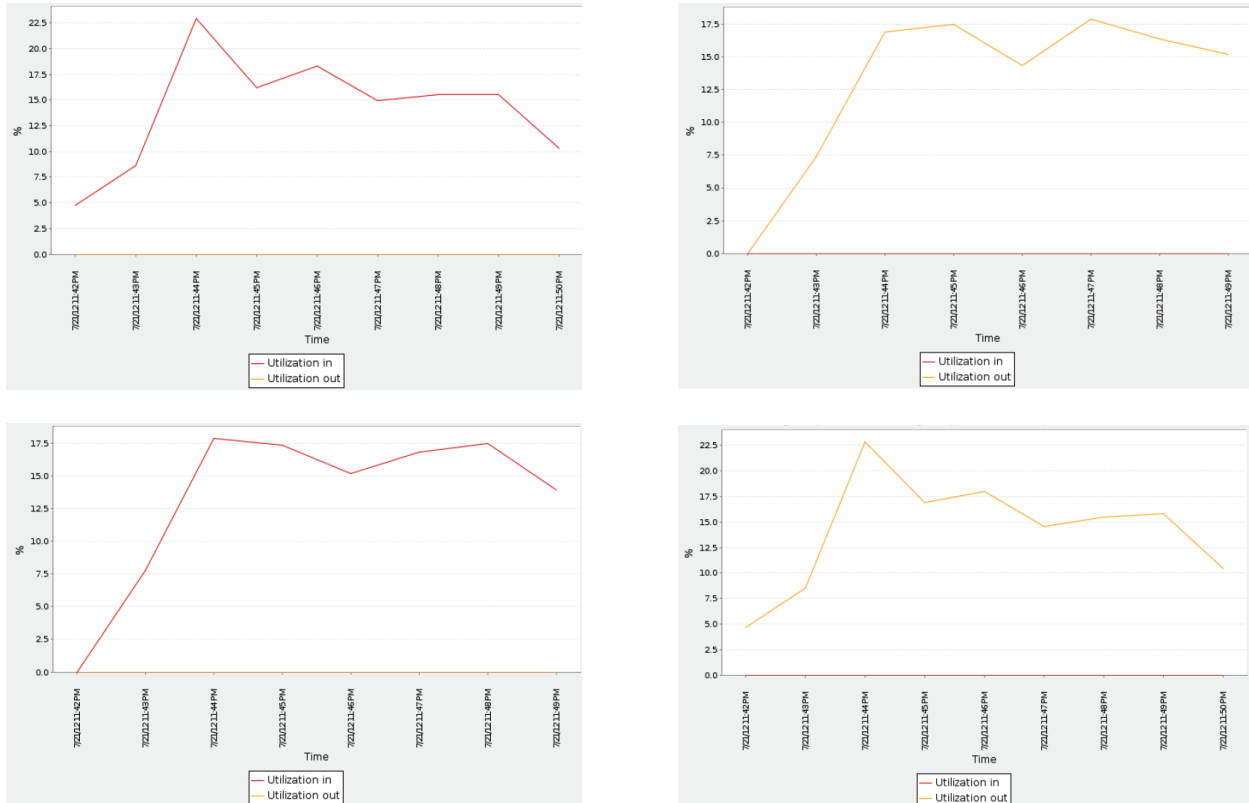
Στο σημείο αυτό μπορούμε να παρατηρήσουμε από το τελευταίο διάγραμμα ότι δεν υπάρχει κίνηση από το πρωτόκολλο ssh, παρά μόνο στα δύο πρώτα λεπτά. Αυτό συνέβη διότι λόγω της μεγάλης συμφόρησης η γραμμή δε μπόρεσε να ανταπεξέλθει και σταμάτησε τις συνδέσεις ssh που είχαν δημιουργηθεί. Το πείραμα επαναλήφθηκε με διαφορετικά αποτελέσματα αλλά η διακοπή στη σύνδεση



είναι σημαντική για την απόδοση του Open vSwitch και έπρεπε να καταγραφεί ως ξεχωριστό αποτέλεσμα.

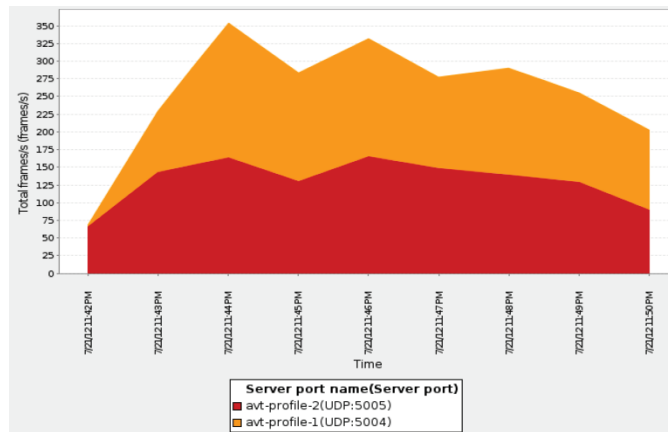
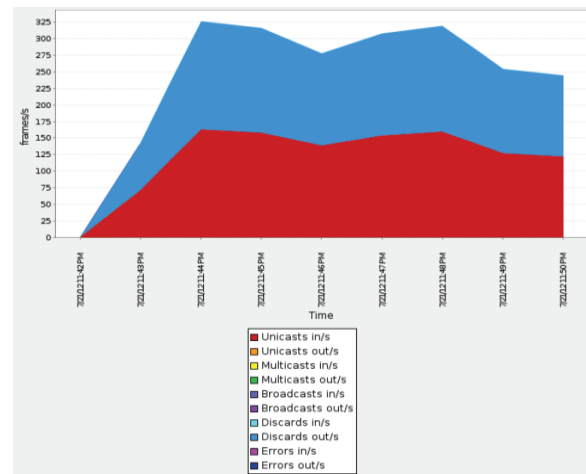
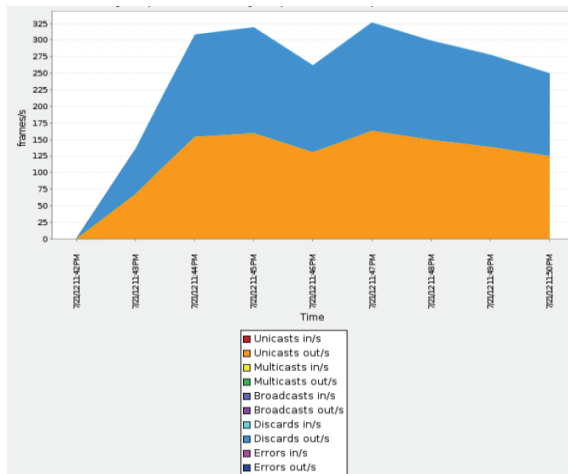
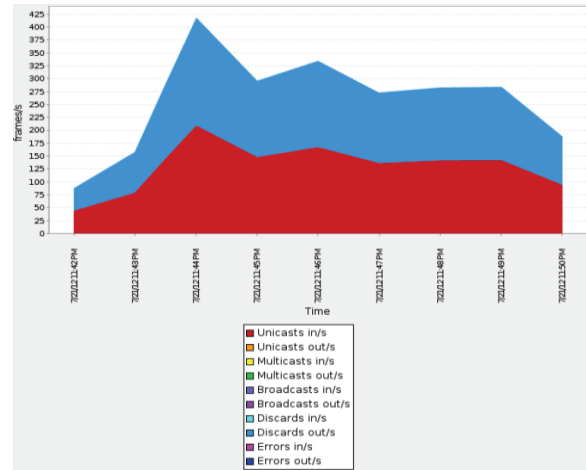
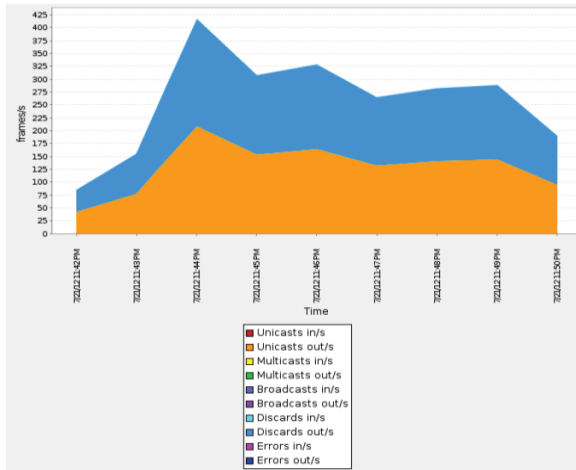
#### 5.5.10.4 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο σε τοπολογία VLAN χωρίς δικτυακή συμφόρηση.

Ακολουθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming βίντεο σε τοπολογία VLAN. Στην περίπτωση αυτή έχουμε streaming μεταξύ VM1 – VM4 και VM3 – VM4.



**Γράφημα 76-79: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Στο συγκεκριμένο κομμάτι του πειράματος δεν έχουμε ροή βίντεο από μία πηγή σε πολλούς προορισμούς, αλλά συγκεκριμένη ροή από το ένα εικονικό μηχάνημα στο άλλο που βρίσκονται στο ίδιο VLAN. Στις παραπάνω γραφικές παραστάσεις εμφανίζεται η χρησιμοποίηση του interface του VM1 και του VM3 κόκκινη γραμμή χρησιμοποίησης που σημαίνει πληροφορία που φεύγει από την εικονική μηχανή ενώ στα VM2 και VM4 κίτρινη γραμμή που σημαίνει ότι η γραμμή χρησιμοποιείται λόγω εισόδου πακέτων. Παρατηρούμε επίσης ότι και σε τοπολογία VLAN η χρησιμοποίηση παραμένει στα ίδια επίπεδα με αυτά της απλής τοπολογίας. Τα αποτελέσματα ανταποκρίνονται απόλυτα με το αποτέλεσμα που είχαμε κατά την εκτέλεση του πειράματος κατά την οποία το βίντεο είχε πολύ καλή απόδοση με ορισμένες διακοπές.



**Γράφημα 80-84: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (frames/min)**

Στις παραπάνω γραφικές παραστάσεις παρουσιάζονται τα πλαίσια που εξέρχονται και εισέρχονται στα εικονικά interfaces καθώς και τα πλαίσια που απορρίπτονται από τη διαδικασία δειγματοληψίας. Στο τελευταίο διάγραμμα παρουσιάζονται τα πρωτόκολλα που χρησιμοποιούνται και τα συνολικά πακέτα που στέλνουν. Στην πιο απλή τοπολογία VLAN έχουμε μόνο δύο ίδια πρωτόκολλα για ροές πακέτων και

παρατηρούμε ότι μεταφέρουν τα πακέτα με την ίδια περίπου ταχύτητα. Δηλαδή το Open vSwitch μοιράζει ισάξια το διαθέσιμο εύρος ζώνης στα δύο VLAN.

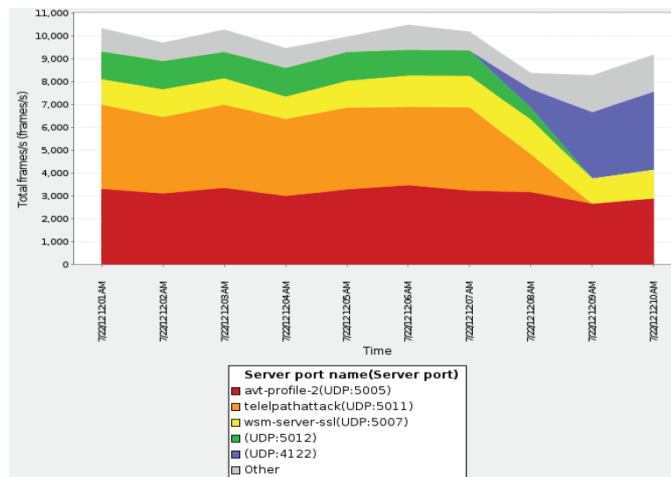
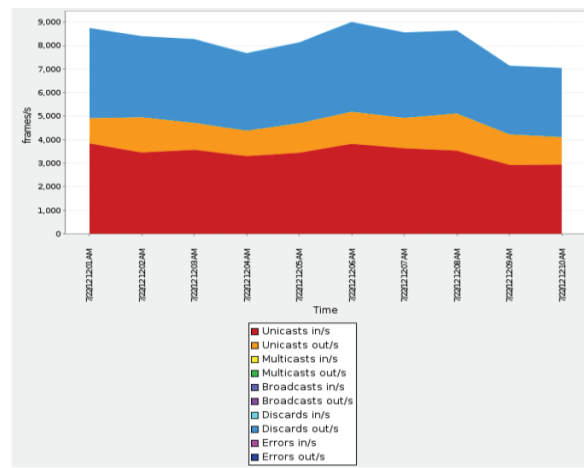
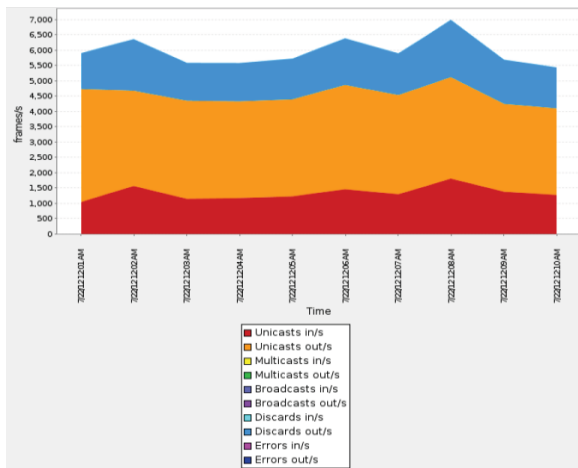
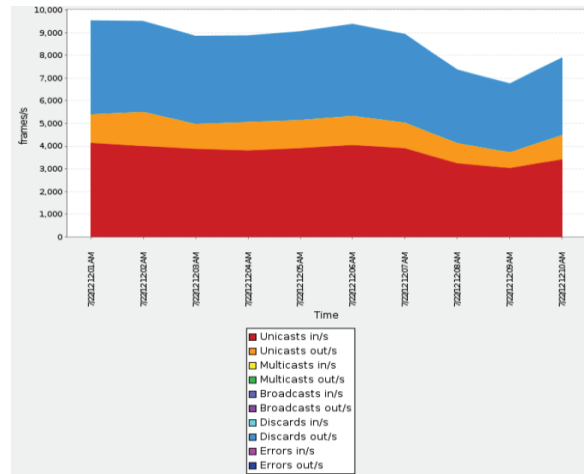
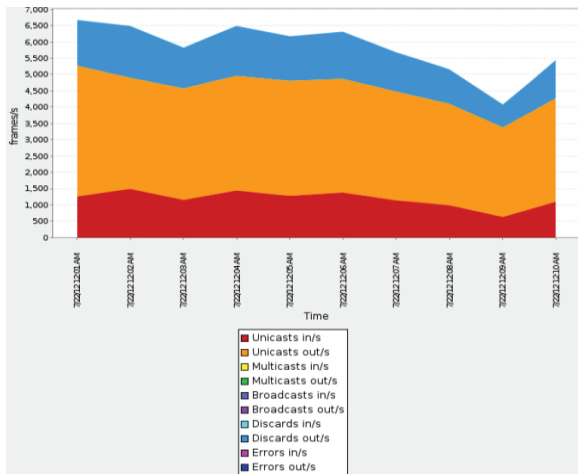
### 5.5.10.5 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο σε τοπολογία VLAN με δικτυακή συμφόρηση.

Ακολούθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming βίντεο σε τοπολογία VLAN με δικτυακή συμφόρηση.



**Γράφημα 85-88: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Οι παραπάνω γραφικές παραστάσεις εμφανίζουν δύο γραμμές κατά τη χρησιμοποίηση της γραμμής. Η μία αφορά τη δημιουργία συμφόρησης στο δίκτυο και η άλλη αφορά τη διαδικασία streaming βίντεο από το ένα εικονικό μηχάνημα στο άλλο. Η γραμμή με τη χαμηλή χρησιμοποίηση είναι αυτή της συμφόρησης και η άλλη αυτή της χρησιμοποίησης λόγω streaming. Παρατηρούμε ότι η χρησιμοποίηση λόγω streaming παρουσιάζει αύξηση σε σχέση με την τοπολογία χωρίς συμφόρηση γεγονός που επιβεβαιώνεται και από τις αρκετές διαλείψεις κατά την αναπαραγωγή του βίντεο. Οδηγούμαστε δηλαδή στο συμπέρασμα ότι η προσθήκη συμφόρησης σε τοπολογία VLAN επηρεάζει άμεσα το αποτέλεσμα της αναπαραγωγής από το εικονικό μηχάνημα.



**Γράφημα 89-93: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (frames/min)**

Οι τρεις γραμμές που εμφανίζονται στα τέσσερα πρώτα διαγράμματα σχετίζονται με τις ροές πακέτων από και προς τα εικονικά μηχανήματα καθώς και με την απόρριψη πακέτων λόγω ρυθμού δειγματοληψίας. Στο τελευταίο διάγραμμα παρουσιάζονται τα πρωτόκολλα που χρησιμοποιήθηκαν.

Εδώ παρατηρήθηκε ένα λάθος κατά την εφαρμογή του πειράματος. Παρατηρούμε ότι απουσιάζει από το διάγραμμα η ροή από το VLAN1. Συμπεραίνουμε λοιπόν ότι η συγκεκριμένη ροή πακέτων έχει διακοπεί είτε ότι βρίσκεται στην περιοχή other του διαγράμματος. Σε κάθε περίπτωση η διακοπή μπορεί να δικαιολογήσει την απόδοση που παρατηρήθηκε κατά την εκτέλεση του πειράματος γεγονός που μας οδηγεί στο συμπέρασμα ότι streaming βίντεο σε VLAN είναι απαγορευτικό με το Open vSwitch και χωρίς μεγάλη επεξεργαστική ισχύ.

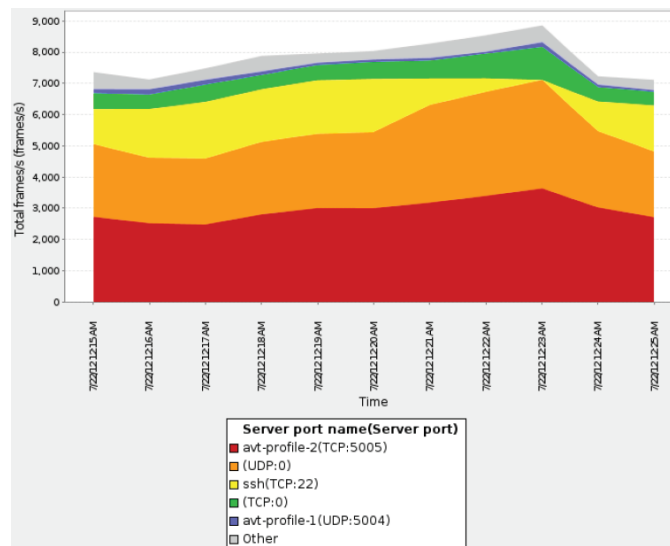
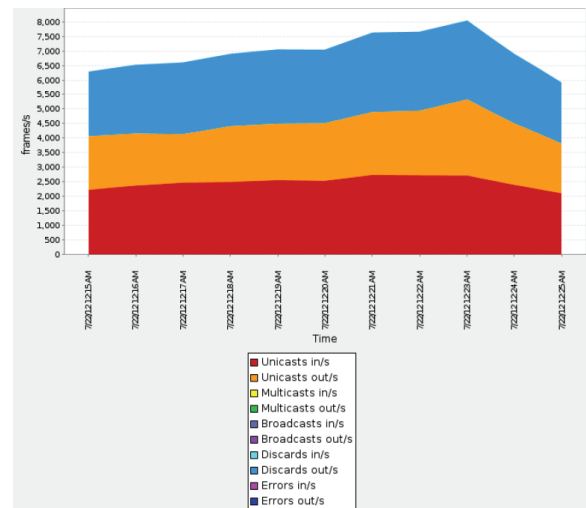
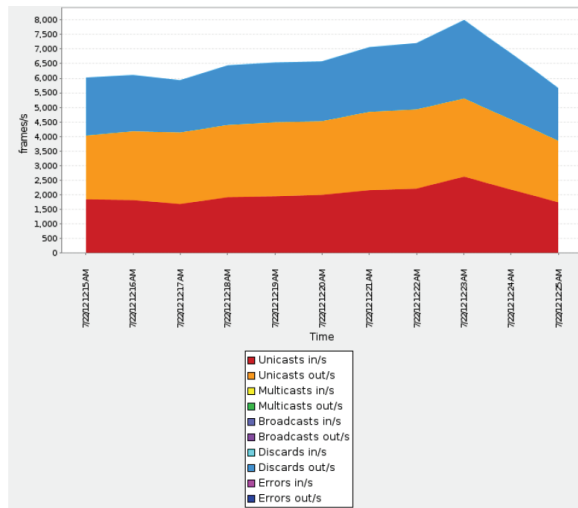
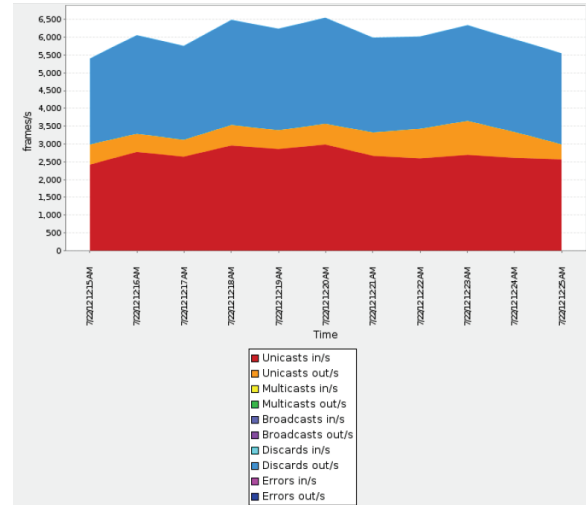
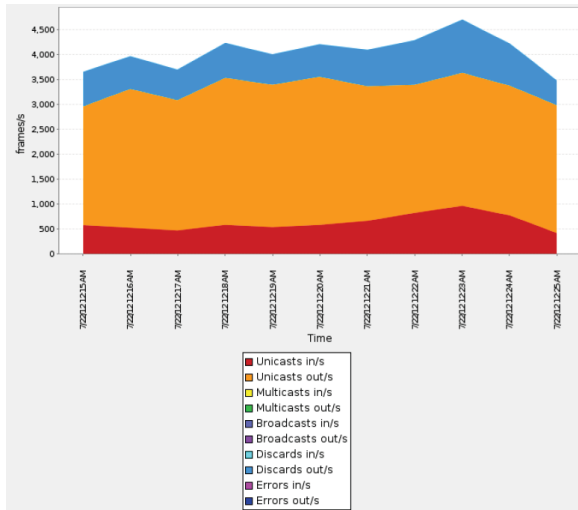
### 5.5.10.6 Απεικόνιση μετρήσεων σε γραφική παράσταση της περίπτωσης streaming βίντεο σε τοπολογία VLAN με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείων.

Ακολούθως δίνονται οι γραφικές παραστάσεις της χρησιμοποίησης, των μετρητών καθώς και των πρωτοκόλλων που χρησιμοποιήθηκαν στην περίπτωση streaming βίντεο με δικτυακή συμφόρηση και ταυτόχρονη μεταφορά αρχείου από το ένα εικονικό μηχάνημα στο άλλο που ανήκουν στο ίδιο VLAN.



**Γράφημα 94-97: Utilization εικονικών interfaces των εικονικών μηχανημάτων VM1-V4**

Στο συγκεκριμένο κομμάτι του πειράματος παρατηρείται και πάλι μεγάλη διαφορά στη χρησιμοποίηση του εικονικού interface. Αυτό οφείλεται στο γεγονός ότι ο VM1 και ο VM3 που κάνουν το streaming μοιράζουν ταυτόχρονα και το αρχείο στα δύο εικονικά μηχανήματα. Αυτό δημιουργεί μεγάλη χρησιμοποίηση της γραμμής που στέλνει αλλά και της γραμμής που λαμβάνει τα πακέτα. Το βίντεο σε αυτό το σημείο παρουσίασε σημαντικότερα προβλήματα κατά την αναπαραγωγή του σε σημείο που ήταν αδύνατη η προβολή του. Στα σημεία μάλιστα που η χρησιμοποίηση παρουσιάζει απότομη πτώση είχαμε απώλεια της σύνδεσης και εν συνεχεία συνεχίστηκε με τα ίδια προβλήματα η αναπαραγωγή.



**Γράφημα 98-102: Μετρητές Ροών (Counters – frames/min) – Πρωτόκολλα (frames/min)**

Στο τελευταίο διάγραμμα παρατηρούμε ότι παρ' ότι η ροή του βίντεο στο VLAN1 έχει το μεγαλύτερο μέρος της δικτυακής κίνησης καθώς και το ssh το δεύτερο μεγαλύτερο μέρος τούτο δε συμβαδίζει με τα αποτελέσματα της πειραματικής μας διάταξης. Παράλληλα παρατηρούμε και τη διακοπή της ssh κίνησης καθώς και της ροής βίντεο στο VLAN2 στο 12:29 του τελευταίου διαγράμματος. Αυτό πιθανότατα οφείλεται στην υπερβολικά μεγάλη κίνηση που δημιουργήσαμε. Τα αποτελέσματα των διαγραμμάτων αυτών έχουν μία απόκλιση με την πραγματικότητα καθώς παρά την καλή απόδοση που φαίνεται να έχει το VLAN1, τούτο δεν ανταποκρίνεται απόλυτα στην πραγματικότητα καθώς η αναπαραγωγή είχε κακή έως απαγορευτική απόδοση για θέαση.

## ΚΕΦΑΛΑΙΟ 6

### ΕΠΙΛΟΓΟΣ

#### 6.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία, παρουσιάσαμε αναλυτικά τη δομή και την αρχιτεκτονική του εικονικού μεταγωγέα, καθώς και ένα μέρος των δυνατοτήτων του. Συμπερασματικά μπορούμε να αναφέρουμε ότι πρόκειται για ένα απαραίτητο εργαλείο για τη δημιουργία εικονικών τοπολογιών είτε σε εικονικοποιημένα περιβάλλοντα, είτε σε φυσικά μηχανήματα. Η απόδοση του στα πειράματα που έγιναν κατά την εκπόνηση της εργασίας, ήταν άριστα για την περίπτωση της απλής διασύνδεσης και της διασύνδεσης VLAN κατά τον έλεγχο της συνδεσιμότητας των εικονικών μηχανών, καθώς και στην περίπτωση της μετάδοσης και αναπαραγωγής ήχου. Αντίθετα παρουσιάστηκαν αρκετά προβλήματα στη μετάδοση και αναπαραγωγή βίντεο από τον host και τα εικονικά μηχανήματα, προβλήματα που δε σχετίζονται αποκλειστικά με τον εικονικό μεταγωγέα αλλά και με την περιορισμένη επεξεργαστική ισχύ που είχαμε διαθέσιμη. Ταυτόχρονα δίνεται στο χρήστη η δυνατότητα, μέσα από την πληθώρα επιλογών που παρέχει ο εικονικός μεταγωγέας, για δημιουργία περίπλοκων δικτυακών τοπολογιών. Ωστόσο μειονέκτημα παραμένει ο τρόπος που ο εικονικός μεταγωγέας αντιμετώπισε τις ροές δεδομένων που δέχτηκε. Σε καταστάσεις πλημμύρα πακέτων παρουσιάστηκαν διακοπές στις επιμέρους συνδέσεις που σε ορισμένες περιπτώσεις οδηγούσαν σε απώλεια πακέτων και άρα και διαδικασιών, όπως η μεταφορά αρχείων από το ένα μηχανήμα στο άλλο. Τελικό συμπέρασμα είναι ότι το Open vSwitch ως ανοιχτό λογισμικό παρέχει αυτή τη στιγμή πολύπλοκες λειτουργίες που επιτελούν προϊόντα της αγοράς επί πληρωμή.

#### 6.2 Προτάσεις για μελλοντικές επεκτάσεις και βελτιώσεις

Τη χρονική περίοδο που γράφτηκε η διπλωματική εργασία δεν υπήρχαν διαθέσιμα κάποια απαραίτητα χαρακτηριστικά που θα έκανα τη χρήση του Open vSwitch ευκολότερη. Για παράδειγμα ένα σημαντικό χαρακτηριστικό θα ήταν η δυνατότητα σύνδεσης του εικονικού μεταγωγέα με έναν εξωτερικό ελεγκτή ο οποίος θα παρείχε το γραφικό περιβάλλον που θα έκανε τις ρυθμίσεις για την παραμετροποίηση του συστήματος ευκολότερη για τον μέσο χρήστη. Ωστόσο η δυνατότητα αυτή έχει πλέον προστεθεί σε επόμενη έκδοση του λογισμικού. Μία δεύτερη επέκταση που χρήζει έρευνας για μελλοντική βελτίωση είναι ο τρόπος με τον οποίο ο εικονικός μεταγωγέας δίνει προτεραιότητα στα πρωτόκολλα καθώς και στις πόρτες από τις οποίες τα δέχεται. Η λειτουργία αυτή θα ήταν πολύ χρήσιμη καθώς ο διαχειριστής του δικτύου θα είχε το πλεονέκτημα να μπορεί να δίνει προτεραιότητα στα πρωτόκολλα και τις ροές που αυτός θεωρεί σημαντικές. Παράλληλα μια τέτοια λειτουργία θα προέβλεπε την κακόβουλη χρήση προγραμμάτων γέννησης πακέτων με ρυθμούς που θα δημιουργούσε προβλήματα στη συνδεσιμότητα του δικτύου. Τέλος αξίζει να σημειωθεί ότι στην τελευταία έκδοση πυρήνα Linux, το Open vSwitch έχει



ενσωματωθεί ως βασικό εργαλείο διαχείρισης τοπολογιών στο λειτουργικό σύστημα μαζί με όλα τα πακέτα λειτουργιών του και δεν απαιτείται η εγκατάστασή του στο userspace. Η έκδοση αυτή ενσωματώθηκε τον Μάιο του 2012 και ενώ οι μετρήσεις και τα αποτελέσματα είχαν ήδη παρθεί. Ωστόσο ενδιαφέρον θα είχε η συγκριτική απόδοση του εργαλείου που βρίσκεται ενσωματωμένο στον πυρήνα Linux και της τοπολογίας που εκτελέστηκαν σε αυτή τη διπλωματική εργασία.

## ΠΑΡΑΡΤΗΜΑ Α

### ΟΔΗΓΟΣ ΕΓΚΑΤΑΣΤΑΣΗΣ OPEN VSWITCH ΜΕΣΩ SCRIPTS

#### A.1 Βασικά χαρακτηριστικά Open vSwitch

Παρακάτω θα γίνει παρουσίαση του βασικού API του Open vSwitch.

- **Ovsdb-server:** Είναι ο daemon που ξεκινάει τη βάση δεδομένων στην οποία καταγράφονται όλα τα γεγονότα της χρήσης του ovs-vsctl.
- **Ovs-vswitchd:** Είναι ο daemon που διαχειρίζεται και ελέγχει οποιονδήποτε αριθμό από Open vSwitch switches υπάρχουν στο φυσικό μηχάνημα. Συνδέεται με τη βάση δεδομένων, δημιουργεί τα απαραίτητα datapath και στη συνέχεια εκτελεί τις διαδικασίες μεταγωγής.
- **Ovs-vsctl:** Είναι το πρόγραμμα εκείνο το οποίο παραμετροποιεί τον ovs-vswitchd παρέχοντας ένα high-level interface στη βάση δεδομένων. Μέσω της σύνδεσης αυτής, εφαρμόζει αλλαγές και καθορίζει τις εντολές που δέχεται. Η λειτουργία του τερματίζει μόλις εφαρμοστούν οι εντολές που του δόθηκαν
- **Ovs-appctl:** Είναι το πρόγραμμα εκείνο, το οποίο παραμετροποιεί τους Open vSwitch daemons που βρίσκονται σε λειτουργία.
- **Ovs-brcompatd:** Είναι ο daemon που παρέχει συμβατότητα με το front-end των Linux Bridge εργαλείων. Ωστόσο αφού το openvswitch.ko module θα πάρει τη θέση του linux bridge δεν υπάρχει λόγος να αναλυθεί περαιτέρω.
- **Ovs-dpctl:** Είναι το πρόγραμμα εκείνο που διαχειρίζεται τα datapaths. Έχει τη δυνατότητα δημιουργίας, τροποποίησης και διαγραφής οποιουδήποτε datapath. Κάθε μηχάνημα έχει μέγιστο δημιουργίας datapaths και συνεπώς και αριθμό εικονικών μεταγωγέων.

#### A.2 Script για την αυτοματοποιημένη δημιουργία απλής τοπολογίας και VLAN τοπολογία.

1) Αρχικά ξεκινάμε τον ovs-vswitchd καθώς και τη βάση δεδομένων

##### Startovs.sh

```
#!/bin/bash
```

```
/sbin/rmmod bridge
```

```
cd /home/npc/Desktop/Open vSwitch-1.4.0
```

```
insmod datapath/linux/Open vSwitch_mod.ko
```

```
ovsdb-server --remote=punix:/usr/local/var/run/Open vSwitch/db.sock --pidfile --detach
```

```
ovs-vswitchd --pidfile --detach
```

2) Στη συνέχεια δημιουργούμε τα εικονικά interfaces

#### **Vif.sh**

```
#!/bin/bash
```

```
VBoxTunctl -b -u npc
```

```
VBoxTunctl -b -u npc
```

```
VBoxTunctl -b -u npc
```

```
VBoxTunctl -b -u npc
```

3) Ενεργοποιούμε τα εικονικά interfaces και ορίζουμε τη διεύθυνση του εικονικού μεταγωγέα.

#### **Ifcommand.sh**

```
#!/bin/bash
```

```
ifconfig eth0 10.1.1.10
```

```
route del -net 10.0.0.0 netmask 255.0.0.0 dev eth0ifconfig br0 10.1.1.1
```

```
ifconfig tap0 up
```

```
ifconfig tap1 up
```

```
ifconfig tap2 up
```

```
ifconfig tap3 up
```

4) Συνδέουμε τις πόρτες που θα δημιουργήσουν το δίκτυο μας χωρίς VLAN tagging.

#### **addport.sh**

```
#!/bin/bash
```

```
ovs-vsctl add-br br0
```

```
ovs-vsctl add-port br0 eth0
```

```
ovs-vsctl add-port br0 tap0
```

```
ovs-vsctl add-port br0 tap1
```

```
ovs-vsctl add-port br0 tap2
```

```
ovs-vsctl add-port br0 tap3
```

5) Για να δημιουργήσουμε VLAN θα πρέπει να διαγράψουμε τις πόρτες που δεν είχαν tagging

#### **delport.sh**

```
#!/bin/bash
```

```
ovs-vsctl del-port br0 eth0
```

```
ovs-vsctl del-port br0 tap0
```

```
ovs-vsctl del-port br0 tap1
```

```
ovs-vsctl del-port br0 tap2
```

```
ovs-vsctl del-port br0 tap3
```

6) Συνδέουμε τις πόρτες που θα δημιουργήσουν το VLAN δίκτυο.

#### addportvlan.sh

```
#!/bin/bash

ovs-vsctl add-port br0 eth0
ovs-vsctl add-port br0 tap0 tag=10
ovs-vsctl add-port br0 tap1 tag=10
ovs-vsctl add-port br0 tap2 tag=20
ovs-vsctl add-port br0 tap3 tag=20
```

- **Για την περίπτωση που θέλουμε να τροποποιήσουμε την ποιότητα της υπηρεσίας χρησιμοποιούμε το qosimple.sh στην περίπτωση της απλής τοπολογίας**

#### qosimple.sh

```
#!/bin/bash
# VM1
ovs-vsctl set Interface tap0 ingress_policing_rate=10000
ovs-vsctl set Interface tap0 ingress_policing_burst=1000
# VM2
ovs-vsctl set Interface tap1 ingress_policing_rate=1000
ovs-vsctl set Interface tap1 ingress_policing_burst=100
# VM3
ovs-vsctl set Interface tap2 ingress_policing_rate=10000
ovs-vsctl set Interface tap2 ingress_policing_burst=1000
# VM4
ovs-vsctl set Interface tap3 ingress_policing_rate=1000
ovs-vsctl set Interface tap3 ingress_policing_burst=100
```

- **Για την περίπτωση που θέλουμε να τροποποιήσουμε την ποιότητα της υπηρεσίας χρησιμοποιούμε το qosvlan.sh στην περίπτωση της απλής τοπολογίας**

#### qosvlan.sh

```
#!/bin/bash
# VM1
ovs-vsctl set Interface tap0 ingress_policing_rate=1000
ovs-vsctl set Interface tap0 ingress_policing_burst=100
# VM2
ovs-vsctl set Interface tap1 ingress_policing_rate=1000
ovs-vsctl set Interface tap1 ingress_policing_burst=100
# VM3
ovs-vsctl set Interface tap2 ingress_policing_rate=10000
ovs-vsctl set Interface tap2 ingress_policing_burst=1000
# VM4
ovs-vsctl set Interface tap3 ingress_policing_rate=10000
ovs-vsctl set Interface tap3 ingress_policing_burst=1000
```

## ΠΑΡΑΡΤΗΜΑ Β΄

### ΟΔΗΓΟΣ ΔΗΜΙΟΥΡΓΙΑΣ ΣΥΜΦΟΡΗΣΗ ΚΑΙ OPENFLOW ΡΟΗΣ

Στο παράρτημα αυτό δίνονται script τα οποία μπορούν να δημιουργήσουν κίνηση μέσω του προγράμματος hping3. Ο χρήστης μπορεί να τροποποιήσει τα script αυτά ανάλογα με τις διευθύνσεις που θέλει να χρησιμοποιήσει καθώς και τον ρυθμό αποστολής.

#### congestionfast.sh

```
#!/bin/bash
Hping3 -udp -fast x.x.x.x -p xxxx
Hping3 -fast x.x.x.x -p xxxx
Hping3 -fast x.x.x.x -p xxxx -3
```

#### congestionfaster.sh

```
#!/bin/bash
Hping3 -udp -faster x.x.x.x -p xxxx
Hping3 -faster x.x.x.x -p xxxx
Hping3 -faster x.x.x.x -p xxxx -3
```

#### congestionflood.sh

```
#!/bin/bash
Hping3 -udp -flood x.x.x.x -p xxxx
Hping3 -flood x.x.x.x -p xxxx
Hping3 -flood x.x.x.x -p xxxx -3
```

Το ακόλουθο script δημιουργεί ροή πακέτων από τον εικονικό μεταγωγέα προς το πρόγραμμα που επεξεργάζεται τα δεδομένα.

#### sFlow.sh

```
#!/bin/bash

COLLECTOR_IP=10.1.1.10
COLLECTOR_PORT=6343
AGENT_IP=eth0
HEADER_BYTES=256
SAMPLING_N=100
POLLING_SECS=30

ovs-vsctl --id=@sflow create sflowagent=${AGENT_IP} target="${COLLECTOR_IP} ${COLLECTOR_PORT}"
header=${HEADER_BYTES} sampling=${SAMPLING_N} polling=${POLLING_SECS} --set bridge br0
sflow=@sflow
```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **Tanenbaum, Andrew S.** *Δίκτυα Υπολογιστών*. σ.λ. : Εκδόσεις Κλειδάριθμος, 2007.
2. **Institute, Information Sciences.** INTERNET PROTOCOL. *PROTOCOL SPECIFICATION*. [Ηλεκτρονικό] <http://www.ietf.org/rfc/rfc791.txt>.
3. **Postel, J.** IETF. *User Datagram Protocol*. [Ηλεκτρονικό] <http://www.ietf.org/rfc/rfc768.txt>.
4. **Wikipedia.** Virtual private network. *Wikipedia The Free Encyclopedia*. [Ηλεκτρονικό] [http://en.wikipedia.org/wiki/Virtual\\_Private\\_Network](http://en.wikipedia.org/wiki/Virtual_Private_Network).
5. **Virtualization.** Virtualization in Computer Networks. *Wikipedia*. [Ηλεκτρονικό] <http://en.wikipedia.org/wiki/Virtualization>.
6. **Wikipedia.** Wikipedia. *Wikipedia The Free Encyclopedia*. [Ηλεκτρονικό] [http://en.wikipedia.org/wiki/Network\\_virtualization](http://en.wikipedia.org/wiki/Network_virtualization).
7. **Community.** Ubuntu. *Home*. [Ηλεκτρονικό] [www.ubuntu.com](http://www.ubuntu.com).
8. **Oracle.** Oracle VM VirtualBox. *VirtualBox*. [Ηλεκτρονικό] <https://www.virtualbox.org/>.
9. **organization, Non-profit.** VideoLAN. *VideoLAN - Official page for VLC media player*. [Ηλεκτρονικό] <http://www.videolan.org/vlc/>.
10. **Schulzrinne, H.** IETF. *A Transport Protocol for Real-Time Applications*. [Ηλεκτρονικό] <http://www.ietf.org/rfc/rfc3550.txt>.
11. **InMon.** sFlow Trend product. *InMon*. [Ηλεκτρονικό] <http://www.inmon.com/products/sFlowTrend.php>.
12. **Networks, Nicira.** Open vSwitch: An Open Virtual Switch. *Open vSwitch*. [Ηλεκτρονικό] <http://openvswitch.org/>.
13. **Ben Blaff, Justin Pettit, Martin Casado, Jesse Gross.** *Virtual Switching in an Era of Advanced Edges*. Florida : s.n., 2010.
14. **Ben Blaff, Justin Pettit, Martin Casado, Scott Shenker, Teemu Koponen, Keith Amidon.** *Extending Networking into the Virtualization Layer*. 2009.
15. **Community.** OpenFlow - Enabling Innovation in Your Network. *Openflow*. [Ηλεκτρονικό] <http://www.openflow.org/>.
16. **Sanfilippo, Salvatore.** Hping. *Home*. [Ηλεκτρονικό] <http://linux.die.net/man/8/hping3>.