

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή Εργασία

“Σχεδίαση, ανάπτυξη και αξιολόγηση μηχανισμού προσαρμογής υπηρεσιών για την εγγυημένη παροχή ποιότητας (QoS), σε διαδραστικό περιβάλλον εκπομπής ευρείας κάλυψης.”

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΛΟΥΚΑΣ ΣΑΚΙΖΛΟΓΛΟΥ

Ηράκλειο, Μάιος 2010

ΕΙΣΗΓΗΤΗΣ: ΕΥΑΓΓΕΛΟΣ ΠΑΛΛΗΣ

*Σε όλους όσους, εν γνώσει ή εν αγνοία τους,
με ενέπνευσαν και με υποστήριξαν.*

Περίληψη

Η πτυχιακή αυτή εργασία επικεντρώνεται στην σχεδίαση και στην ανάπτυξη ενός μηχανισμού προσαρμογής υπηρεσιών, για την εγγυημένη παροχή QoS, σε διαδραστικά περιβάλλοντα εκπομπής ευρείας κάλυψης. Προτείνεται επίσης, η εφαρμογή του μηχανισμού σε ένα δίκτυο DVB-T καθώς θεωρείται πως σε ένα τέτοιο σύνθετο περιβάλλον, όπου η χρονικά μεταβαλλόμενη κίνηση δημιουργείται/προωθείται από τους επίγειους κόμβους διανομής και η ποιότητα των παρεχόμενων υπηρεσιών εξαρτάται άμεσα από τους διαθέσιμους δικτυακούς πόρους και από τις απαιτήσεις των υπηρεσιών σε εύρος ζώνης, η σε-πραγματικό-χρόνο προσαρμογή των υπηρεσιών είναι νευραλγικής σημασίας σε περιπτώσεις που οι συνθήκες μετάδοσης δεν είναι οι απαιτούμενες. Ο μηχανισμός χρησιμοποιεί αντικειμενικά μεγέθη που χαρακτηρίζουν τη δικτυακή απόδοση (Loss Rate, Bit rate, Jitter) για να αξιολογήσει την ποιότητα της παρεχόμενων υπηρεσιών και αν χρειαστεί να τις προσαρμόσει ανάλογα με το εύρος των διαθέσιμων πόρων. Τέλος, γίνεται αξιολόγηση του μηχανισμού μέσα από μια σειρά πειραματικών μετρήσεων, όπου μπορεί να παρατηρηθεί ο χρόνος απόκρισης και η αποτελεσματικότητά του.

Ευχαριστίες

Η παρούσα πτυχιακή εργασία είναι αποτέλεσμα έρευνας και εργασίας που έλαβε χώρα στο Εργαστήριο Έρευνας και Ανάπτυξης Τηλεπικοινωνιακών Συστημάτων ΠΑΣΙΦΑΗ υπό την επίβλεψη του καθηγητή κ. Ευάγγελου Πάλλη.

Με την ολοκλήρωση της εργασίας θα ήθελα καταρχάς να ευχαριστήσω τον επιβλέποντα καθηγητή μου Κο Ευάγγελο Πάλλη για την καθοδήγηση και τις δυνατότητες που μου προσέφερε για τη διεξαγωγή και διεκπεραίωση της εργασίας μου.

Επίσης χρωστώ μεγάλη ευγνωμοσύνη στον Κο Ανάργυρο Σιδέρη για όλο το χρόνο που επένδυσε σε μένα προσφέροντας την ανεξάντλητη βοήθεια του, για την εμπιστοσύνη του, αλλά και για την κατανόηση που επέδειξε.

Έπειτα θα ήθελα να ευχαριστήσω τον Κο Ευάγγελο Μαρκάκη, για την αμέριστη συμπαράσταση και βοήθεια που μου παρείχε. Η υποστήριξη του ήταν μεγάλης σημασίας για μένα.

Ευχαριστώ επίσης τον Κο Γεώργιο Μαστοράκη για τις καίριες υποδείξεις του και την γενικότερη συμβολή του.

Στο πρόσωπο του Γιώργου Αλεξίου δεν βρήκα μόνο ένα πολύτιμο συνεργάτη αλλά και έναν καλό φίλο. Τον ευχαριστώ και αυτόν θερμά.

Επιπλέον θα ήθελα να ευχαριστήσω τα μέλη του Εργαστηρίου ΠΑΣΙΦΑΗ για τη συνεργασία και τη βοήθειά τους.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την ανεκτίμητη βοήθεια και υποστήριξη που μου προσφέρει όλα αυτά τα χρόνια.

Λουκάς Σακιζλόγλου

Ηράκλειο, Μάιος 2010

Περιεχόμενα

1. Εισαγωγή.....	12
1.1. Γενικά	12
1.1. Στόχοι	12
1.2. Δομή της Πτυχιακής Εργασίας	13
2. Θεωρητικό Υπόβαθρο.....	14
2.1. Πρότυπα/Τεχνολογίες.....	14
2.1.1. Ψηφιακή Τηλεόραση και Δίκτυα Ευρείας Κάλυψης (Broadcast Networks)...	14
2.1.2. DVB-T	14
2.1.2.1. Αρχιτεκτονική ενός δικτύου DVB-T	15
2.1.2. MPEG	17
2.1.3. QoS.....	17
2.1.4. WLAN	18
2.1.5. XML	20
2.2. Πρωτόκολλα.....	22
2.2.1. UDP	22
2.3. Λογισμικό	23
2.3.1. VLC	23
2.3.2. TC	24
2.3.2.1. HTB.....	25
2.3.2.2. TBF.....	26
2.3.2.3. NETEM.....	26
2.3.3. IPTABLES	26
2.4. Βιβλιοθήκες.....	26
2.4.1. Βιβλιοθήκες της JAVA.....	27
2.4.1.1. Jchart2D	27
2.4.1.2. Dom4J	27
2.4.1.3. Jrcap	27

2.4.1.4. Jaxen	28
2.4.2. Βιβλιοθήκη της C	28
2.4.2.1. Libnetfilter_queue	28
3. Προτεινόμενος Μηχανισμός Προσαρμογής Υπηρεσιών σε Δίκτυο Ευρείας Κάλυψης	29
3.1. Προτεινόμενη Αρχιτεκτονική του Δικτύου	29
3.2. Περιγραφή του Μηχανισμού	29
4. Υλοποίηση.....	35
4.1. Τελικός Χρήστης (End User)	35
4.2. Ενδιάμεσος Κόμβος Διανομής (Cell Main Node)	35
4.3. Κεντρικό Σημείο Εκπομπής (Πλατφόρμα DVB-T) – Core Router (CR)	36
4.4. Λειτουργία του Μηχανισμού	37
4.5. Πειραματικά Σενάρια	43
5. Ανάλυση και Παρουσίαση Αποτελεσμάτων.....	46
5.1. 1 ^ο Σενάριο.....	46
5.2. 2 ^ο Σενάριο.....	52
5.3. 3 ^ο Σενάριο.....	58
6. Συμπεράσματα – Μελλοντικές Προτάσεις.....	66
A. Παράρτημα.....	68
A.1 Υπολογισμός Δικτυακών Χαρακτηριστικών (Γλώσσα Java).....	68
A.1.1 Υπολογισμός Ποσοστού Απωλειών (Loss Rate)	68
A.1.2 Υπολογισμός Ρυθμού Μετάδοσης (Bitrate)	70
A.1.3 Υπολογισμός Διακύμανσης Απωλειών (Jitter)	72
A.1.4 Υπολογισμός Μονόδρομης Καθυστέρησης (One-Way Delay)	74
A.2 Script Περιορισμού Εύρους Ζώνης.....	75
A.3 Δρομολόγηση και Διαμόρφωση Πακέτων (Γλώσσα C)	76
A.4 Κωδικοποίηση ροής Video (εκτέλεση απο BASH)	79
A.5 Γραφήματα Μετρήσεων (ανα χρήστη)	80
A.5.1. Ροή στη θύρα 5000	80
A.5.2. Ροή στη θύρα 5001	86

A.5.3. Ροή στη θύρα 5002	92
A.5.4. Ροή στη θύρα 5003	98
A.5.5. Ροή στη θύρα 5004	101
Βιβλιογραφία	104

Πίνακας Σχημάτων

Σχήμα 1 Διάγραμμα ενός DVB πομπού	15
Σχήμα 2 Γενική Αρχιτεκτονική ενός DVB-T δικτύου	16
Σχήμα 3 Ανασκόπηση των πρωτοκόλλων 802.11.....	19
Σχήμα 4 Αρχιτεκτονική ενός WLAN δικτύου	20
Σχήμα 5 Παράδειγμα κώδικα ενός XML αρχείου	21
Σχήμα 6 Παράδειγμα δομής ενός XML αρχείου.....	22
Σχήμα 7 Επικεφαλίδα UDP	23
Σχήμα 8 Το VLC σε λειτουργία	24
Σχήμα 9 Λειτουργία του HTB αλγόριθμου	25
Σχήμα 10 Η αρχιτεκτονική της εφαρμογής CMNM	31
Σχήμα 11 Η αρχιτεκτονική της εφαρμογής CM	32
Σχήμα 12 Προτεινόμενο Δίκτυο DVB-T.....	34
Σχήμα 13 Ο Διαμορφωτής COFDM που Παρουσιάζει τα Στοιχεία της Διαμόρφωσης	36
Σχήμα 14 Υλοποίηση του DVB-T δικτύου	37
Σχήμα 15 Η εφαρμογή CMNM.....	39
Σχήμα 16 Η εφαρμογή CM.....	40
Σχήμα 17 Το αρχείο XML	41
Σχήμα 18 Αξιολόγηση Κίνησης - Έξοδος του CM.....	42
Σχήμα 19 Απόφαση του CM για προσαρμογή.....	42
Σχήμα 20 Το δίκτυο μαζί με το μηχανισμό.....	43
Σχήμα 21 1 ^ο Σενάριο – 1 ^η Μέτρηση.....	46
Σχήμα 22 1 ^ο Σενάριο – 1 ^η Μέτρηση - Screenshot.....	47
Σχήμα 23 1ο Σενάριο – 2η Μέτρηση	48
Σχήμα 24 1ο Σενάριο – 2η Μέτρηση - Screenshot	49
Σχήμα 25 1 ^ο Σενάριο – 3 ^η Μέτρηση.....	50
Σχήμα 26 1 ^ο Σενάριο – 3 ^η Μέτρηση - Screenshot πριν την προσαρμογή	51
Σχήμα 27 1 ^ο Σενάριο – 3 ^η Μέτρηση - Screenshot μετά την προσαρμογή.....	51
Σχήμα 28 2ο Σενάριο – 1η Μέτρηση	52
Σχήμα 29 2ο Σενάριο – 1η Μέτρηση - Screenshot	53
Σχήμα 30 2 ^ο Σενάριο - 2 ^η Μέτρηση.....	54
Σχήμα 31 2 ^ο Σενάριο - 2 ^η Μέτρηση - Screenshot	55
Σχήμα 32 2 ^ο Σενάριο – 3 ^η Μέτρηση.....	56
Σχήμα 33 2 ^ο Σενάριο – 3 ^η Μέτρηση – Screenshot πριν την προσαρμογή	57
Σχήμα 34 2 ^ο Σενάριο – 3 ^η Μέτρηση – Screenshot μετά την προσαρμογή.....	58
Σχήμα 35 3 ^ο Σενάριο – 1 ^η Μέτρηση.....	59

Σχήμα 36 3 ^ο Σενάριο – 1 ^η Μέτρηση - Screenshot.....	60
Σχήμα 37 3 ^ο Σενάριο – 2 ^η Μέτρηση.....	61
Σχήμα 38 3 ^ο Σενάριο – 2 ^η Μέτρηση - Screenshot.....	62
Σχήμα 39 3 ^ο Σενάριο – 3 ^η Μέτρηση.....	63
Σχήμα 40 3 ^ο Σενάριο – 3 ^η Μέτρηση – Screenshot πριν την προσαρμογή	64
Σχήμα 41 3 ^ο Σενάριο – 3 ^η Μέτρηση – Screenshot μετά την προσαρμογή.....	65

Ακρωνύμια

AP **Access Point**

CMN Cell Main Node

DSL Digital Subscriber Loop

DVB Digital Video Broadcasting

DVB-H Digital Video Broadcasting - Handheld

DVB-RCS Digital Video Broadcasting - Return Channel via Satellite

DVB-S Digital Video Broadcasting - Satellite

DVB-T Digital Video Broadcasting - Terrestrial

FTP File Transfer Protocol

HTTP Hypertext Transfer Protocol

IP Internet Protocol

LAN Local Area Network

PC Personal Computer

QoS Quality of Service

RTT Round Trip Time

SLA Service Level Agreement

SMTP Simple Mail Transfer Protocol

TBF Token Bucket Filter

ToS Type of Service Field

UDP User Datagram Protocol

VoD Video On Demand

WAN Wide Area Network

WLAN wireless local area network

WWW World Wide Web

XML Extensible Markup Language

1. Εισαγωγή

1.1. Γενικά

Μια πλατφόρμα επίγειας ψηφιακής τηλεόρασης (DVB-T) χρησιμοποιείται κυρίως ως ένα σύστημα εκπομπής “μπουκέτων” ψηφιακών τηλεοπτικών προγραμμάτων προς ένα μεγάλο αριθμό χρηστών/τηλεθεατών, διασκορπισμένων στην περιοχή κάλυψης. Το ενδογενές χαρακτηριστικό της να συνδυάζει/ενσωματώνει ψηφιακά τηλεοπτικά προγράμματα (MPEG-2) και υπηρεσίες Διαδικτύου (IP) στο ίδιο ρεύμα μεταφοράς, επιτρέπουν την αξιοποίησή της ως δικτυακή υποδομή για την διασύνδεση ενδιάμεσων κόμβων επικοινωνίας, μέσω των οποίων είναι εφικτή η επικοινωνία πάροχων ετερογενών υπηρεσιών και τελικών χρηστών.

Σε ένα τέτοιο σύνθετο περιβάλλον, όπου η χρονικά μεταβαλλόμενη κίνηση δημιουργείται/προωθείται από τους επίγειους κόμβους διανομής, η ποιότητα των παρεχόμενων υπηρεσιών εξαρτάται άμεσα από τους διαθέσιμους δικτυακούς πόρους καθώς και από τις απαιτήσεις των υπηρεσιών σε εύρος ζώνης. Είναι λοιπόν προφανές, ότι μία στατική και προκαθορισμένη κατανομή των διαθέσιμων δικτυακών πόρων μεταξύ των κόμβων πρόσβασης, δεν εξασφαλίζει την βέλτιστη εκμετάλλευση του φάσματος και κατά συνέπεια, την εγγυημένη ποιότητα υπηρεσίας. Προς αυτή την κατεύθυνση, απαιτείται η δυναμική διαχείριση των διαθέσιμων δικτυακών πόρων καθώς και η σε-πραγματικό-χρόνο προσαρμογή των υπηρεσιών σε αυτούς, για τη βελτιστοποίηση των επιδόσεων των διαδραστικών υποδομών DVB-T που χρησιμοποιούνται ως δικτυακές πλατφόρμες.

1.1. Στόχοι

Σε αυτό το πλαίσιο, ο κύριος στόχος αυτής της πτυχιακής εργασίας είναι να σχεδιάσει, να αναπτύξει και να αξιολογήσει ένα μηχανισμό προσαρμογής υπηρεσιών σε πραγματικό χρόνο, που θα επιτρέψει την εγγυημένη παροχή ποιότητας υπηρεσιών σε ένα διαδραστικό περιβάλλον εκπομπής ευρείας κάλυψης.

1.2. Δομή της Πτυχιακής Εργασίας

Σε πρώτη φάση, θα αναφερθούν και θα περιγραφούν συνοπτικά οι τεχνολογίες, τα πρωτόκολλα και το λογισμικό που χρησιμοποιήθηκε κατά την υλοποίηση της εργασίας. Στόχος αυτού του κεφαλαίου είναι να δομήσει ένα κατάλληλο θεωρητικό υπόβαθρο ώστε να καταστήσει το σύγγραμμα κατανοητό για τον εκάστοτε αναγνώστη. Στη συνέχεια, στο τρίτο κεφάλαιο, θα γίνει μια περιγραφή του μηχανισμού προσαρμογής υπηρεσιών εστιάζοντας στην διαδικασία κατασκευής του, στον τρόπο λειτουργίας του αλλά και στους στόχους και στους περιορισμούς της εφαρμογής του. Το τέταρτο κεφάλαιο θα περιγράψει το πειραματικό δίκτυο DVB-T στο οποίο θα δοκιμαστεί ο μηχανισμός και στο πέμπτο κεφάλαιο θα πραγματοποιηθεί η αξιολόγηση του μηχανισμού μέσω πειραματικών μετρήσεων, οι οποίες μετέπειτα θα σχολιαστούν. Τέλος, θα αναλυθούν τα συμπεράσματα που προέκυψαν από τις πειραματικές μετρήσεις και θα κατατεθούν κάποιες προτάσεις για μελλοντική εργασία, που θα μπορούσε να διευρύνει την προοπτική της πτυχιακής εργασίας.

2. Θεωρητικό Υπόβαθρο

Σε αυτό το κεφάλαιο θα αναφερθούν και θα περιγραφούν συνοπτικά οι τεχνολογίες, τα πρωτόκολλα και το λογισμικό που χρησιμοποιήθηκε κατά την υλοποίηση της εργασίας.

2.1. Πρότυπα/Τεχνολογίες

2.1.1. Ψηφιακή Τηλεόραση και Δίκτυα Ευρείας Κάλυψης (Broadcast Networks)

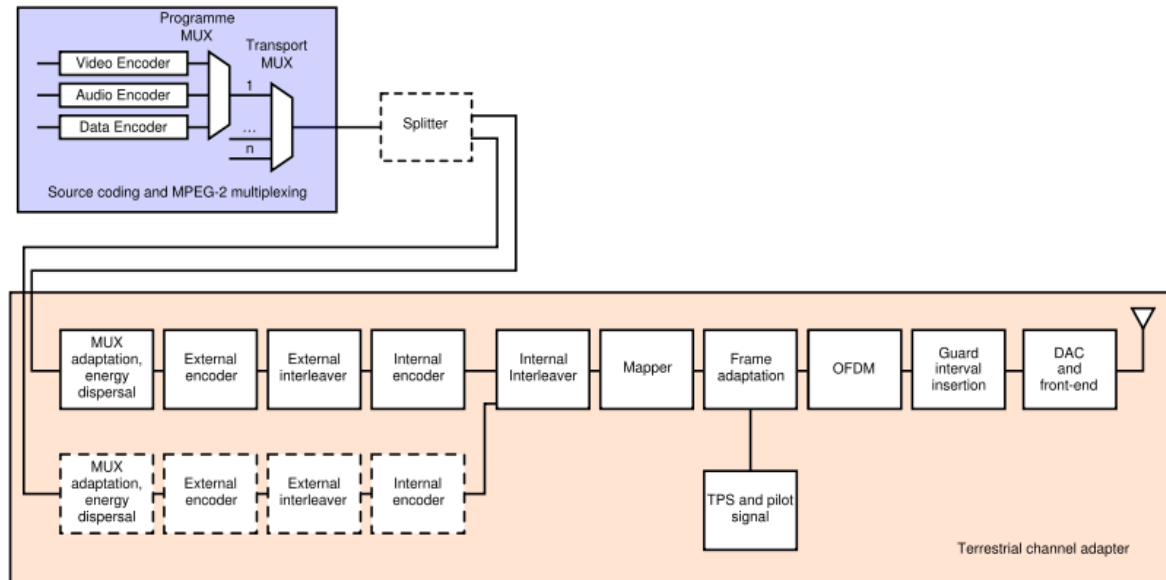
Η ψηφιακή τηλεόραση, με αργό αλλά αυξανόμενο ρυθμό, έχει μπει τα τελευταία χρόνια στη ζωή των πολιτών της Ευρωπαϊκής Ένωσης αλλά και της Ελλάδας ειδικότερα. Ο βασικός λόγος που έκανε αναγκαία την εισαγωγή της νέας αυτής τεχνολογίας δεν είναι άλλος από τα πλεονεκτήματα της σε σχέση με το παλαιότερα πρότυπα. Η τεχνολογία της ψηφιακής τηλεόρασης υλοποιείται παγκοσμίως από τρεις οικογένειες προτύπων: DVB (Digital Video Broadcasting), που χρησιμοποιείται στην Ευρώπη, ATSC (Advanced Television Systems Committee) που χρησιμοποιείται στην Αμερική και ISDB (Integrated Services Digital Broadcasting) που χρησιμοποιείται στην Ιαπωνία.

Η οικογένεια προτύπων DVB αποτελείται από τα παρακάτω πρότυπα DVB-T (Digital Video Broadcasting-Terrestrial), DVB-S (Digital Video Broadcasting-Satellite), DVB-RCS (Digital Video Broadcasting - Return Channel via Satellite) και DVB-H (Digital Video Broadcasting - Handheld).

2.1.2. DVB-T

Το πρότυπο DVB-T (Digital Video Broadcasting-Terrestrial), με το οποίο ασχολείται η εργασία αυτή, χρησιμοποιείται για τη μετάδοση ψηφιακής επίγειας τηλεόρασης. Σε ένα σύστημα DVB-T το οπτικοακουστικό σήμα μεταδίδεται συμπιεσμένο, χρησιμοποιώντας διαμόρφωση πολλαπλών φερόντων στο σχήμα της πολυπλεξίας με ορθογώνια διαίρεση συχνότητας και κωδικοποίηση καναλιού (COFDM – Coded Orthogonal Frequency Division Multiplexing). Η μέθοδος κωδικοποίησης πηγής που χρησιμοποιείται είναι το πρότυπο MPEG-2, ενώ πρόσφατα υιοθετήθηκε και το H.264. Στα DVB-T συστήματα η μετάδοση επιτυγχάνεται εκπέμποντας σε ένα από τα κανάλια

21-69 της μπάντας των UHF, όπως και τα “παραδοσιακά” συστήματα αναλογικής μετάδοσης, έχοντας διαθέσιμο εύρος ζώνης 8 MHz.



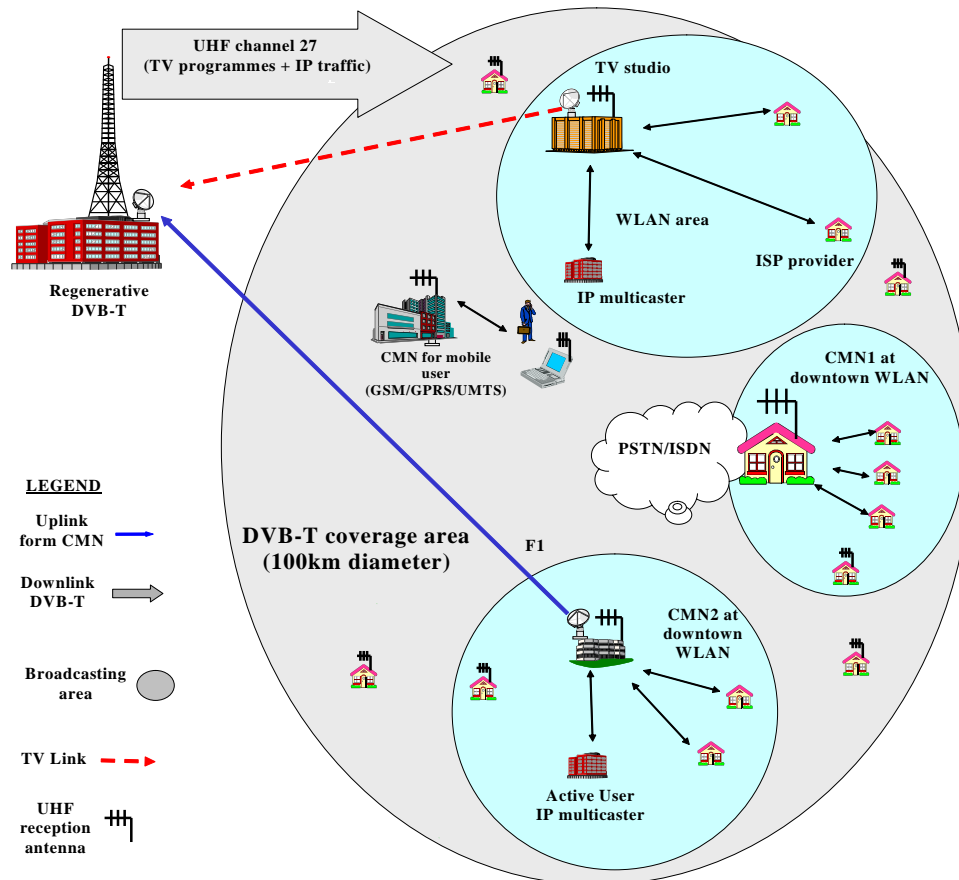
Σχήμα 1 Διάγραμμα ενός DVB πομπού

Το πρότυπο αυτό υποστηρίζει μόνο μονόδρομη κίνηση και έχει τη δυνατότητα εκπομπής επιπλέον υπηρεσιών, όπως IPTV και IPRadio, όπου δεν χρειάζονται αλληλεπίδραση με τον τελικό χρήστη. Για να επιτευχθεί η αμφίδρομη επικοινωνία και άρα να υπάρξει δυνατότητα για παροχή IP υπηρεσιών όπως HTTP, FTP, SMTP, Video On Demand ή Audio On Demand θα πρέπει να προβλεφθεί η ύπαρξη ενός καναλιού επιστροφής ώστε να γίνονται οι αιτήσεις για την υπηρεσία που ζητά ο τελικός χρήστης. Αυτό το κανάλι επιστροφής θα μπορούσε να είναι ένα δίκτυο PSTN, ISDN, GSM, GPRS, DVBS, xDSL ή οποιαδήποτε άλλη τεχνολογία εξυπηρετεί την εκάστοτε εφαρμογή.

2.1.2.1. Αρχιτεκτονική ενός δικτύου DVB-T

Όπως αναφέρθηκε και προηγουμένως ένα σύστημα ψηφιακής τηλεόρασης με ένα κανάλι επιστροφής (xDSL, WLAN, DVB-RCS, PSTN, ISDN, GSM, GPRS) μπορεί να γίνει διαδραστικό, όμως πέρα από μια απλή αλληλεπίδραση του τελικού χρήστη με το περιεχόμενο (VoD, AoD) μπορεί να μεταφέρει και IP δεδομένα. Αυτό ακριβώς το πλεονέκτημα είναι που η εργασία εκμεταλλεύεται ώστε να πετύχει την παροχή

δικτυακών υπηρεσιών μέσα από ένα υβριδικό σύστημα όπου συνεργάζονται η ψηφιακή τηλεόρασης αλλά και άλλες τεχνολογίες.



Σχήμα 2 Γενική Αρχιτεκτονική ενός DVB-T δικτύου

Παραδοσιακά, ένα τέτοιο σύστημα αποτελείται από το κεντρικό σημείο εκπομπής (πλατφόρμα DVB-T), από πολλούς ενδιάμεσους κόμβους διανομής (Cell Main Node - CMN) και από τους τελικούς χρήστες (End Users). Οι CMN είναι υπεύθυνοι για τη δρομολόγηση της κίνησης από και προς τους τελικούς χρήστες. Λαμβάνουν την κίνηση από μία κάρτα τηλεόρασης (DVB-T) και την προωθούν στους τελικούς χρήστες αλλά παράλληλα δρομολογούν και την κίνηση από τους τελικούς χρήστες προς την πλατφόρμα ψηφιακής τηλεόρασης μέσω του καναλιού επιστροφής που διαθέτει ο εκάστοτε CMN (WLAN, Ethernet). Στο Σχήμα 2 φαίνεται η γενική αρχιτεκτονική ενός τέτοιου συστήματος.

2.1.2. MPEG

Η Ομάδα Ειδικών Κινούμενης Εικόνας ή MPEG (Moving Picture Experts Group) είναι μια ομάδα εργασίας ISO/ IEC, υπεύθυνη για την ανάπτυξη των τηλεοπτικών και ακουστικών προτύπων κωδικοποίησης. Ένα από τα πρότυπα της οικογένειας MPEG το οποίο αξιοποιείται για τη δημιουργία τηλεοπτικών σημάτων είναι το MPEG-2. Το πρότυπο αυτό υποστηρίζει εφαρμογές με μεγάλες απαιτήσεις στην ποιότητα του video και προσφέρει μεταβλητούς ρυθμούς μετάδοσης από τη στιγμή που ο βαθμός συμπίεσης μεταβάλλεται αντιστρόφως ανάλογα με την πολυπλοκότητα των πλαισίων που κωδικοποιούνται. Έχει καθιερωθεί σε παγκόσμιο επίπεδο ως το πρότυπο για συμπίεση της ψηφιακής τηλεόρασης, αφού παρέχει υψηλό βαθμό συμπίεσης διατηρώντας την εικόνα σε υψηλό επίπεδο ποιότητας. Το πρότυπο MPEG-2 σε συνδυασμό με το πρότυπο Multi Protocol Encapsulation (MPE) ενθυλακώνει τα IP δεδομένα σε ροές μεταφοράς οι οποίες αποστέλλονται μέσω δικτυακών πλατφόρμων DVB. Εκτός από την επίγεια ψηφιακή τηλεόραση το πρότυπο MPEG-2 αξιοποιείται ακόμα στην δορυφορική τηλεόραση (Satellite Digital TV), στην καλωδιακή τηλεόραση (Cable TV) και στους δίσκους DVD (Digital Video Disk).

2.1.3. QoS

Ως πρότυπο Ποιότητας Υπηρεσίας (QoS – Quality of Service) ορίζεται η διαδικασία που διασφαλίζει την ποιότητα μετάδοσης και λήψης μιας παρεχόμενης υπηρεσίας. Στην περίπτωση που αυτή η διαδικασία είναι επικεντρωμένη στην ποιότητα της μετάδοσης, που ορίζεται από τις απαιτήσεις της υπηρεσίας, το QoS συχνά ονομάζεται και Network QoS και κριτήρια μέτρησης και αξιολόγησης του αποτελούν αντικειμενικά κριτήρια δικτυακών επιδόσεων όπως: η μονόδρομη καθυστέρηση (One-way Delay), η διακύμανση καθυστέρησης (Jitter), το ποσοστό απωλειών (Loss Rate), το RTT (Round-Trip Time), κ.α.

Σε διαφορετική περίπτωση, που η παροχή ποιότητας υπηρεσίας είναι επικεντρωμένη στην ποιότητα της λήψης της υπηρεσίας, που ορίζεται από τις απαιτήσεις του χρήστη, το QoS συχνά αναφέρεται και ως Perceived QoS (PQoS) και υπολογίζεται από κριτήρια

περισσότερο υποκειμενικά, όπως: στατιστικές μελέτες, ερωτηματολόγια, MOS (Mean Opinion Score), κ.α.

Η πιο συνηθισμένη λύση για την παροχή QoS είναι η ανακατανομή των διαθέσιμων πόρων, μια μέθοδο που εφαρμόζουν και ήδη υπάρχουσες τεχνολογίες όπως η DiffServ, IntServ και MPLS. Στις περιπτώσεις όμως που κάτι τέτοιο δεν είναι δυνατό, απαιτείται η προσαρμογή των εκάστοτε υπηρεσιών στους διαθέσιμους πόρους. Μια τέτοια περίπτωση εξετάζεται και σε αυτή την εργασία.

2.1.4. WLAN

Ως ασύρματο τοπικό δίκτυο (Wireless LAN - WLAN) ορίζεται η σύνδεση δύο ή περισσότερων υπολογιστών χωρίς την χρήση καλωδίων. Το WLAN χρησιμοποιεί την διαμόρφωση σήματος Spread Spectrum ή/και OFDM και τα ραδιοκύματα για να πετύχει επικοινωνία μεταξύ συσκευών σε μια μικρή περιοχή. Αυτό δίνει την δυνατότητα στους χρήστες να μπορούν να μετακινούνται μέσα σε μία ορισμένη περιοχή και να συνεχίζουν να είναι συνδεδεμένοι στο δίκτυο.

Για τους απλούς χρήστες Η/Υ, το ασύρματο δίκτυο έχει γίνει δημοφιλές λόγω της ευκολίας που έχει στην εγκατάσταση και την ελευθερία κινήσεων που δίνει στο χρήστη. Το τελευταίο προτέρημα, σε συνδυασμό με τη μεγάλη αύξηση πωλήσεων φορητών συσκευών με τη δυνατότητα σύνδεσης στο Διαδίκτυο, υπογράμμισαν την ευκολία αλλά και την ανάγκη για περισσότερα ασύρματα δίκτυα.

Το πρώτο WLAN αναπτύχθηκε από το Πανεπιστήμιο της Χαβάη και ονομαζόταν ALOHA net. Το συγκεκριμένο δίκτυο περιλάμβανε πολλές καινοτομίες και υπήρξε η βάση για το τοπικό δίκτυο Ethernet και για τα ασύρματα δίκτυα. Χρησιμοποιούσε τοπολογία αμφίδρομου αστέρα για την επικοινωνία επτά υπολογιστών, οι οποίοι ήταν εγκατεστημένοι σε τέσσερα νησιά, με ένα κεντρικό υπολογιστή, χωρίς να χρησιμοποιούνται ενσύρματες γραμμές.

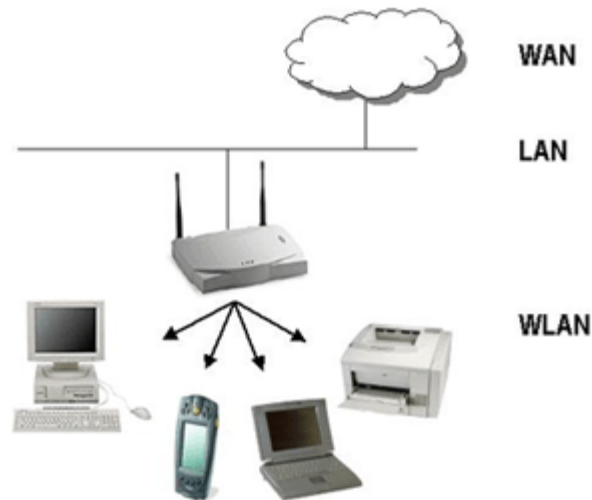
Στην αρχή της πορείας των ασύρματων δικτύων, το υλικό τους ήταν τόσο ακριβό που χρησιμοποιούνταν μόνο όπου η εγκατάσταση δομημένης καλωδίωσης ήταν δύσκολη ή αδύνατη. Κατά την ανάπτυξη των WLAN κυριάρχησαν τα πρότυπα IEEE 802.11x.

Protocol	Release Date	Op. Frequency	Throughput (Typ)	Data Rate (Max)	Modulation Technique	Range (Radius Indoor)	Range (Radius Outdoor)
Legacy	1997	2.4 GHz	0.9 Mbit/s	2 Mbit/s		~20 Meters	~100 Meters
802.11a	1999	5 GHz	23 Mbit/s	54 Mbit/s	OFDM	~35 Meters	~120 Meters
802.11b	1999	2.4 GHz	4.3 Mbit/s	11 Mbit/s	DSSS	~38 Meters	~140 Meters
802.11g	2003	2.4 GHz	19 Mbit/s	54 Mbit/s	OFDM	~38 Meters	~140 Meters
802.11n	June 2009	2.4 GHz 5 GHz	74 Mbit/s	248 Mbit/s		~70 Meters	~250 Meters
802.11y	June 2008	3.7 GHz	23 Mbit/s	54 Mbit/s		~50 Meters	~5000 Meters

Σχήμα 3 Ανασκόπηση των πρωτοκόλλων 802.11

Όπως αναφέρθηκε και παραπάνω, τα WLAN έγιναν δημοφιλή κυρίως για την ευκολία χρήσης τους, το μικρό κόστος και την ευκολία ενσωμάτωσής τους με υπάρχοντα δίκτυα και εξοπλισμό. Πολλοί υπολογιστές που πωλούνται σήμερα και σχεδόν το σύνολο των φορητών υπολογιστών έρχονται με εξοπλισμό πρόσβασης σε ασύρματο δίκτυο.

Η τεχνολογία των ασύρματων τοπικών δικτύων, αν και παρέχει πολλές ευκολίες και παρέχει τα πλεονεκτήματα που αναφέρονται παραπάνω, έχει και τις αδυναμίες της. Για ορισμένες εγκαταστάσεις, η χρήση τοπικών δικτύων μπορεί να μην είναι η επιθυμητή λύση. Αυτό κυρίως οφείλεται σε περιορισμούς που εισαγάγει η ίδια η τεχνολογία όπως η ασφάλεια, ακτίνα χρήσης, αξιοπιστία, ταχύτητα και πολλούς άλλους περιορισμούς.



Σχήμα 4 Αρχιτεκτονική ενός WLAN δικτύου

Όλα τα συστήματα που μπορούν να συνδεθούν στο ασύρματο μέσο σε ένα δίκτυο αναφέρονται ως σταθμοί. Όλοι οι σταθμοί είναι εξοπλισμένοι με ασύρματες κάρτες δικτύου. Οι σταθμοί χωρίζονται σε δύο κατηγορίες σημεία πρόσβασης και πελάτες.

Τα σημεία πρόσβασης (Access Point - AP) είναι σταθμοί βάσης για το ασύρματο τοπικό δίκτυο. Εκπέμπουν και λαμβάνουν ραδιοσυχνότητες έτσι ώστε οι ασύρματες συσκευές να επικοινωνούν μεταξύ τους (δηλ. όλες οι ασύρματες συσκευές επικοινωνούν μέσω των AP). Το AP είναι το κέντρο και ο ελεγκτής του ασύρματου δικτύου, ορίζει την ακτίνα δράσης και τις παραμέτρους των πρωτοκόλλων που θα χρησιμοποιηθούν. Το AP συνδέει το ασύρματο δίκτυο με υπάρχοντα ενσύρματα δίκτυα. Πολλά AP συνδεδεμένα με ενσύρματο μέσο και λειτουργώντας σε διαφορετικά κανάλια συχνότητας συνδυάζονται για να επεκτείνουν την ακτίνα ενός ασύρματου τοπικού δικτύου.

Οι πελάτες σε ένα ασύρματο τοπικό δίκτυο μπορεί να είναι μεταφερόμενες συσκευές όπως φορητοί υπολογιστές, PDA (Personal digital Assistants), τηλέφωνα IP ή σταθερές συσκευές όπως προσωπικοί υπολογιστές, κ.α., εξοπλισμένα με μια ασύρματη κάρτα δικτύου.

2.1.5. XML

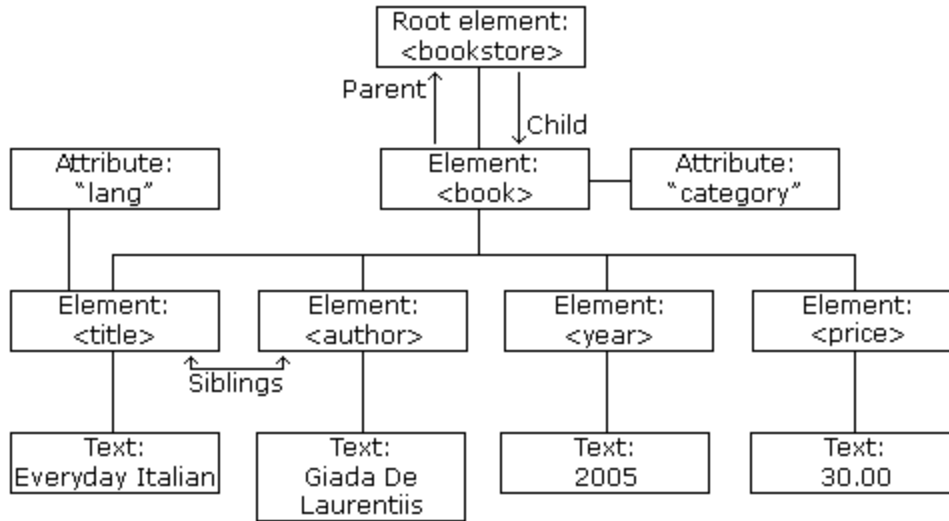
Η XML (Extensible Markup Language) είναι ένα σύνολο κανόνων κωδικοποίησης ηλεκτρονικών εγγράφων που δημιουργήθηκε από τη W3C και ορίζεται στο πρότυπο XML1.0.

Οι στόχοι της XML επικεντρώνονται στην απλότητα, στη γενικότητα άλλα και στην ευκολία της χρήσης της μέσω του Διαδικτύου. Είναι ίσως ο πιο διαδεδομένος τρόπος αναπαράστασης δομών δεδομένων και χρησιμοποιείται ευρέως στα web services αλλά και γενικώς στην ανταλλαγή πληροφοριών στο Διαδίκτυο, όπως προαναφέρθηκε. Έως τη στιγμή που γράφονται αυτές οι γραμμές, εκατοντάδες γλώσσες που βασίζονται στην XML έχουν αναπτυχθεί με πιο γνωστές τις [RSS](#), [Atom](#), [SOAP](#), and [XHTML](#).

Η σύνταξη της XML αναπαριστά ένα δέντρο: όλα τα έγγραφα πρέπει να περιέχουν τη ρίζα (root) και στη συνέχεια απλώνονται χρησιμοποιώντας σαν κλαδιά τα στοιχεία (elements). Κάθε στοιχείο φυσικά μπορεί να έχει και υπό-στοιχεία (sub-elements). Τα στοιχεία επίσης προσδιορίζονται από τις ιδιότητες (attributes). Κάθε στοιχείο πρέπει να ξεκινά με το όνομα του στοιχείου περιστοιχισμένο από <> και να τελειώνει με το όνομα του στοιχείου περιστοιχισμένο από </>. Παραδείγματος χάριν, το στοιχείο element δηλώνεται: <element> και κλείνει: </element> . Στο Σχήμα 5 φαίνεται ένα παράδειγμα της σύνταξης και στο Σχήμα 6 φαίνεται η δομή ενός XML αρχείου.

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J. K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Σχήμα 5 Παράδειγμα κώδικα ενός XML αρχείου



Σχήμα 6 Παράδειγμα δομής ενός XML αρχείου

2.2. Πρωτόκολλα

Για την παροχή της υπηρεσίας επιλέχθηκε να χρησιμοποιηθεί το πρωτόκολλο μεταφοράς δεδομένων UDP. Οι λόγοι για αυτή την επιλογή αφορούν στην απλότητα που χαρακτηρίζει το συγκεκριμένο πρωτόκολλο αλλά και στην ευρεία χρήση του από τις σύγχρονες Διαδικτυακές εφαρμογές (ειδικά για την μετάδοση Video όπως εξηγείται και παρακάτω, κάτι που επιχειρείται και στη συγκεκριμένη εργασία).

2.2.1. UDP

Ένα από τα πρωτόκολλα μεταφοράς πληροφοριών που υποστηρίζει η στοίβα πρωτοκόλλων του Internet είναι το Πρωτόκολλο Δεδομενογραφημάτων Χρήστη ή αλλιώς UDP (User Datagram Protocol). Το UDP είναι ασυνδεδειστροφές, προσφέρει δηλ. τη δυνατότητα να στέλνουν οι εφαρμογές ενθυλακωμένα ακατέργαστα δεδομενογραφήματα IP χωρίς να πρέπει να εγκαταστήσουν μια σύνδεση. Πολλές εφαρμογές πελάτη-εξυπηρετητή, που απαιτούν μία αίτηση και μία απόκριση για τη λειτουργία τους, χρησιμοποιούν το UDP για προφανείς λόγους.

Το Πρωτόκολλο UDP (User Datagram Protocol), χρησιμοποιείται όταν σε μια επικοινωνία δεν απαιτείται η επιβεβαίωση των δεδομένων η απλά ο χρόνος που απαιτείται είναι απαγορευτικός. Ένα καλό παράδειγμα χρήσης του πρωτοκόλλου αυτού είναι στη μετάδοση πολυμεσικού περιεχομένου όπως βίντεο ή ήχος που είναι σε πραγματικό χρόνο και οι καθυστερήσεις ή οι επιβεβαιώσεις είναι περιττές.

+	Bits 0 - 15	16 - 31
0	Source Port Θύρα Προέλευσης	Destination Port Θύρα Προορισμού
32	Length μέγεθος	Checksum
64	Data Δεδομένα	

Σχήμα 7 Επικεφαλίδα UDP

Στο Σχήμα 7 φαίνεται η δομή ενός UDP πακέτου. Παρατηρούμε ότι η επικεφαλίδα αποτελείται από τέσσερα πεδία (source port, destination port, length, checksum) αν και τα δύο από αυτά δεν είναι υποχρεωτικά να συμπληρωθούν (source port, checksum). Το πεδίο source port περιέχει τη θύρα του αποστολέα, το πεδίο destination port την θύρα του παραλήπτη, το πεδίο length το μέγεθος του πακέτου σε bytes και το checksum πεδίο 16-bit το οποίο χρησιμοποιείται για επαλήθευση της ορθότητας του πακέτου στο σύνολό του, δηλαδή τόσο της κεφαλίδας όσο και των δεδομένων.

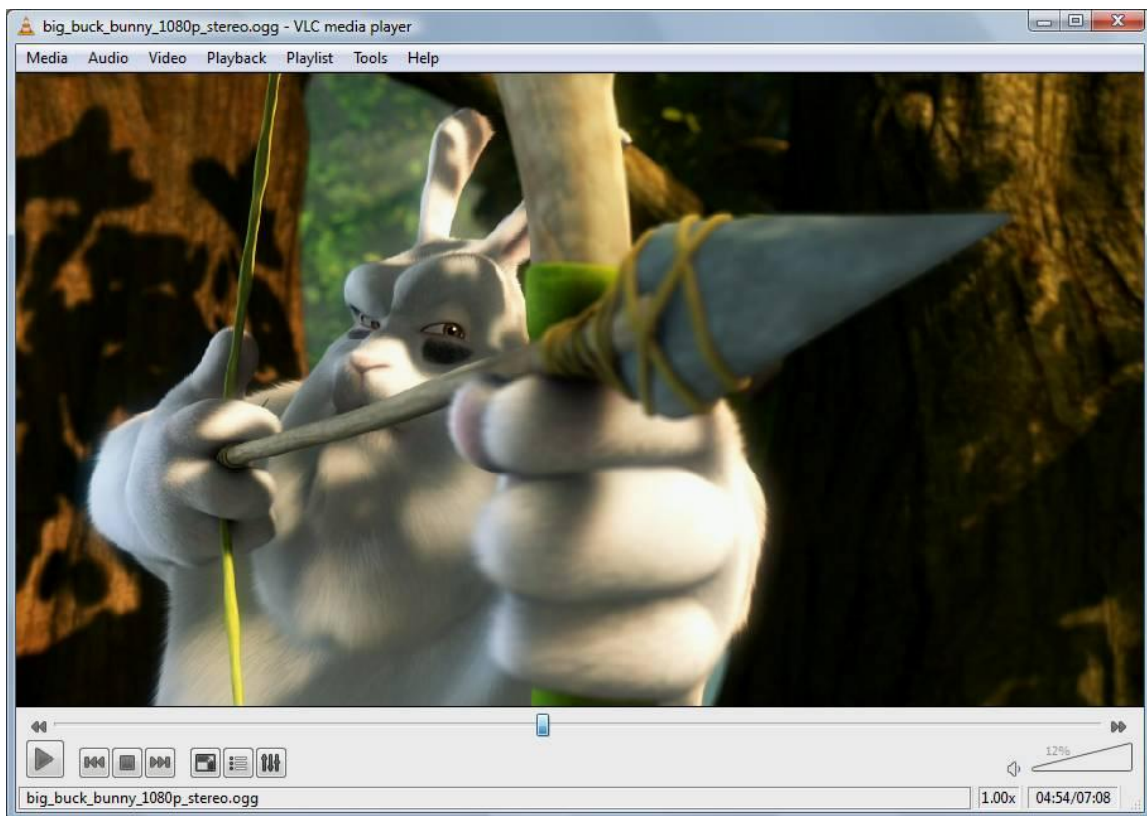
2.3. Λογισμικό

2.3.1. VLC

Η εφαρμογή ανοιχτού λογισμικού VLC είναι μια εφαρμογή αναπαραγωγής πολυμέσων, γραμμένη από το VideoLAN project. Η εφαρμογή υποστηρίζει την αναπαραγωγή

σχεδόν όλων των κωδικοποιήσεων ήχου και εικόνας που χρησιμοποιούνται στα πολυμέσα, το streaming αλλά και την επιλογή αποθήκευσης αρχείων σε πολλές τυποποιήσεις (format). Παράλληλα η εφαρμογή καταφέρνει να κρατά στο ελάχιστο τους απαιτούμενους πόρους συστήματος για να λειτουργήσει και μπορεί να εκτελεστεί σχεδόν σε όλα τα λειτουργικά συστήματα. Τα πλεονεκτήματα αυτά καθιστούν την εφαρμογή πολύ δημοφιλή ανάμεσα στους χρήστες του Internet και όχι μόνο.

Οι περισσότεροι κωδικοποιητές που χρησιμοποιούνται στο VLC προέρχονται από τη βιβλιοθήκη libavcodec από το FFMPEG Project (μια άλλη πολύ σημαντική εφαρμογή αναπαραγωγής πολυμέσων) αλλά η εφαρμογή χρησιμοποιεί τους δικούς της πολυπλέκτες και αποπολυπλέκτες για τη λειτουργία της.



Σχήμα 8 Το VLC σε λειτουργία

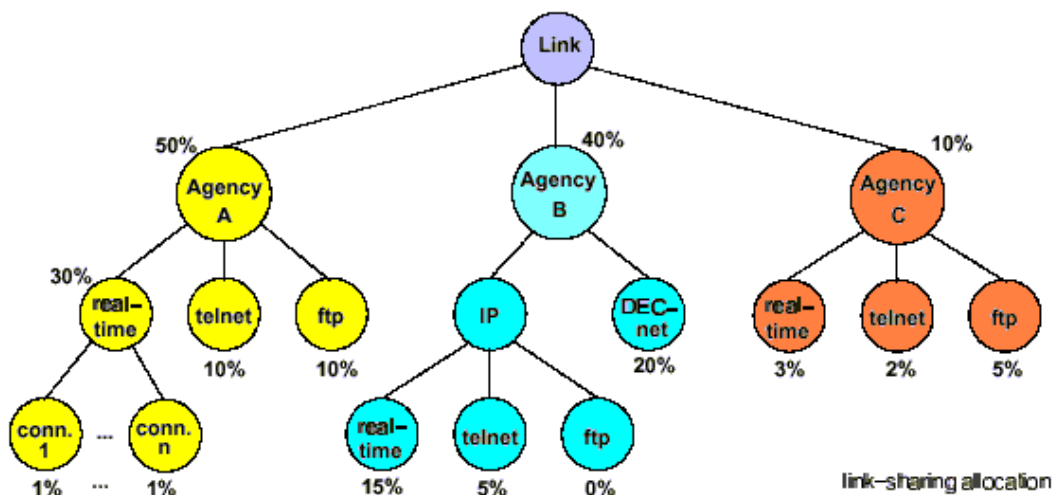
2.3.2. TC

Το Traffic Control είναι μια εφαρμογή για Unix-Like λειτουργικά συστήματα που περιλαμβάνει μηχανισμούς και λειτουργίες για την επεξεργασία πακέτων και για τη λήψη μιας απόφασης αν θα επεξεργασθεί ή απορρίψει κάποιο πακέτο η κάρτα δικτύου. Οι λειτουργίες συμπεριλαμβάνουν την τοποθέτηση πακέτων σε ουρά, την εφαρμογή πολιτικών στην κίνηση μιας κάρτας δικτύου, τη διαμόρφωση της κίνησης, την ταξινόμηση συγκεκριμένων ροών, κ.α.

Παρακάτω ακολουθούν οι λειτουργίες που χρησιμοποιήθηκαν κατά την υλοποίηση της εργασίας.

2.3.2.1. HTB

Ο αλγόριθμος HTB (Hierarchical Token Bucket) χρησιμοποιείται για την μορφοποίηση της δικτυακής κίνησης σε ένα ιεραρχικό μοντέλο (Σχήμα 9) όπου κάθε κλάση παίρνει το δικαίωμα να στείλει τα πακέτα που έχει στις ουρές με ένα μέσο ρυθμό μετάδοσης και με ελεγχόμενους καταιγισμούς (bursts) της ανάλογα με το ποσοστό εύρους που διαθέτει. Ο αλγόριθμος αυτός χρησιμοποιείται για τον καταμερισμό του bandwidth και μπορεί να χρησιμοποιηθεί και για τη διαφοροποίηση υπηρεσιών.



Σχήμα 9 Λειτουργία του HTB αλγόριθμου

2.3.2.2. TBF

Ο αλγόριθμος Token Bucket Filter χρησιμοποιείται για τον έλεγχο του ποσού των δεδομένων που εισέρχονται σε ένα δίκτυο, επιτρέποντας έτσι να υπάρχουν συσσωρευμένα ποσά δεδομένων προς αποστολή. Χρησιμοποιείται συνήθως στο πλαίσιο διαμόρφωσης της κίνησης σε ένα δίκτυο ή στον περιορισμό του ρυθμού μετάδοσης δεδομένων.

2.3.2.3. NETEM

Το Netem είναι ένα εργαλείο που έχει τη δυνατότητα να προσθέτει καθυστέρηση σε μία συγκεκριμένη δικτυακή ροή. Υποστηρίζει επίσης την πρόσθεση διακύμανσης καθυστέρησης καθώς και την πρόσθεση απωλειών.

2.3.3. IPTABLES

Το εργαλείο iptables χρησιμοποιείται και αυτό για την διαχείριση της δικτυακής κίνησης. Μέσα στις λειτουργίες που παρέχει είναι το φιλτράρισμα πακέτων με την εισαγωγή κανόνων όπως επίσης και η τοποθέτηση των πακέτων σε ουρές για την μετέπειτα επεξεργασία τους.

2.4. Βιβλιοθήκες

Κάποιες από τις πιο σημαντικές λειτουργίες της εφαρμογής, όπως η σύλληψη των πακέτων και η επεξεργασία τους ή το φιλτράρισμα τους, δεν θα μπορούσαν να κατασκευασθούν χωρίς τη χρήση κάποιων βιβλιοθηκών. Ακολουθεί η αναφορά και μια μικρή περιγραφή των δυνατοτήτων τους.

2.4.1. Βιβλιοθήκες της JAVA

Παρακάτω αναφέρονται οι βιβλιοθήκες που χρησιμοποιήθηκαν στο κομμάτι της εφαρμογής που κατασκευάστηκε με τη γλώσσα JAVA.

2.4.1.1. Jchart2D

Η JChart2D είναι μια βιβλιοθήκη που υλοποιείται στη JAVA για την κατασκευή γραφημάτων. Έχει σχεδιαστεί για να εμφανίζει ίχνη ροών που αποτελούνται από πολλά σημεία. Είναι απλή στη χρήση της μα ταυτόχρονα πολύ αποτελεσματική.

Στην μηχανισμό που κατασκευάστηκε κατά τη διάρκεια αυτής της εργασίας, η συγκεκριμένη βιβλιοθήκη χρησιμοποιήθηκε στην κατασκευή γραφημάτων των μετρήσεων που πραγματοποιήθηκαν.

2.4.1.2. Dom4J

Η βιβλιοθήκη Dom4J είναι μια απλή στη χρήση της βιβλιοθήκη, ανοιχτού λογισμικού, και επιτρέπει την δημιουργία και επεξεργασία XML αρχείων όπως επίσης και την εφαρμογή σε αυτά τα αρχεία των XPath's. Υλοποιείται σε οποιαδήποτε πλατφόρμα JAVA και υποστηρίζει πλήρως τα πρότυπα DOM, SAX και JAXP.

Στον μηχανισμό, η Dom4J χρησιμοποιήθηκε για την κατασκευή των XML αρχείων που χρησιμοποιούνται για την επικοινωνία των κεντρικών κόμβων διανομής με τον κεντρικό εξυπηρετητή.

2.4.1.3. Jrcap

Η Jrcap είναι μια βιβλιοθήκη για τη σύλληψη και την αποστολή πακέτων, γραμμένη στη JAVA. Αποτελεί ένα βασικό μέρος του μηχανισμού που κατασκευάστηκε καθώς η σύλληψη και το φιλτράρισμα των πακέτων γίνεται με χρήση αυτής της βιβλιοθήκης. Η

Jrcap έχει τη δυνατότητα να συλλάβει Ethernet, IPv4, IPv6, ARP/RARP, TCP, UDP, και ICMPv4 πακέτα.

2.4.1.4. Jaxen

Η Jaxen είναι μια βιβλιοθήκη ανοιχτού λογισμικού για τη χρήση XPath, γραμμένη σε JAVA. Μπορεί να προσαρμοστεί σε πολλά διαφορετικά μοντέλα αντικειμένων όπως τα DOM, XOM, Dom4J και JDOM.

2.4.2. Βιβλιοθήκη της C

Παρακάτω αναφέρονται η βιβλιοθήκη που χρησιμοποιήθηκε στο κομμάτι της εφαρμογής που κατασκευάστηκε με τη γλώσσα C.

2.4.2.1. Libnetfilter_queue

Η βιβλιοθήκη libnetfilter_queue προσφέρει τη δυνατότητα σύλληψης πακέτων που φτάνουν στην κάρτα δικτύου, το φιλτράρισμα τους ανάλογα με κανόνες που θέτει ο χρήστης και την τοποθέτηση τους σε συγκεκριμένες ουρές. Έπειτα μπορεί κάποιος να καλέσει συναρτήσεις όπου θα επεξεργάζονται τα πακέτα που βρίσκονται στις ουρές.

Όλες οι βιβλιοθήκες που χρησιμοποιήθηκαν έχουν αδειοδοτηθεί από τη GNU/LGPL.

3. Προτεινόμενος Μηχανισμός Προσαρμογής Υπηρεσιών σε Δίκτυο Ευρείας Κάλυψης

3.1. Προτεινόμενη Αρχιτεκτονική του Δικτύου

Το δίκτυο που θα χρησιμοποιηθεί για τις μετρήσεις και την αξιολόγηση του μηχανισμού, βασίζεται στο γενική αρχιτεκτονική ενός DVB-T δικτύου που περιγράφηκε παραπάνω (κεφ. 2.1.2.1. Αρχιτεκτονική ενός δικτύου DVB-T). Συμπεριλαμβάνει δε, δύο ενδιάμεσους κόμβους διανομής καθώς θεωρείται ότι έτσι προσομοιώνεται σωστά ένα δίκτυο DVB-T στην πιο απλή του εκδοχή.

Συνολικά, λοιπόν, το δίκτυο αποτελείται από τους τελικούς χρήστες που στο τμήμα του δικτύου που ελέγχει ο CMN1 χρησιμοποιούν ως δίκτυο πρόσβασης ADSL τεχνολογία και στο τμήμα που ελέγχει ο CMN2 χρησιμοποιείται ένα ασύρματο δίκτυο WLAN(802.11g). Οι τελικοί χρήστες συνδέονται μέσω του δικτύου πρόσβασης στους CMN οι οποίοι συνδέονται με τη σειρά τους στο κεντρικό σημείο εκπομπής (πλατφόρμα DVB-T). Για να επιτευχθεί η αμφίδρομη επικοινωνία μεταξύ της πλατφόρμας DVB-T και των CMN χρησιμοποιείται σαν κανάλι καθόδου η DVB-T πλατφόρμα, όπου εκτός από τα IP δεδομένα θα μπορούσαν να αποστέλλονται και τα ψηφιακά τηλεοπτικά προγράμματα, ενώ σαν κανάλι ανόδου-επιστροφής χρησιμοποιείται στο πρώτο κόμβο διανομής (CMN1) μια Ethernet σύνδεση και στον δεύτερο (CMN2) ένα ασύρματο δίκτυο WLAN(802.11g).

Η κίνηση δημιουργείται από τον service provider (τελικός χρήστης που υπόκειται στο CMN2) δρομολογείται μέσω του CMN2 προς την πλατφόρμα DVB-T και από εκεί στον CMN1 για να προωθηθεί στους τελικούς χρήστες. Γίνεται προφανές ότι όλη η κίνηση περνά από τους CMN's όπου είναι και τα σημεία που παρακολουθείται και αποστέλλεται στο κεντρικό σημείο εκπομπής προς αξιολόγηση. Το σημείο που είναι υπεύθυνο για την αξιολόγηση της κίνησης και για τη λήψη απόφασης προσαρμογής της υπηρεσίας είναι ο Core Router (CR).

3.2. Περιγραφή του Μηχανισμού

Με γνώμονα το παραπάνω πλαίσιο εφαρμογής, αλλά και τους στόχους της εργασίας που αναφέρθηκαν σε προηγούμενο κεφάλαιο (σελ. 12) κατασκευάστηκε ένας κεντροποιημένος μηχανισμός όπου θα έχει τις εξής δυνατότητες

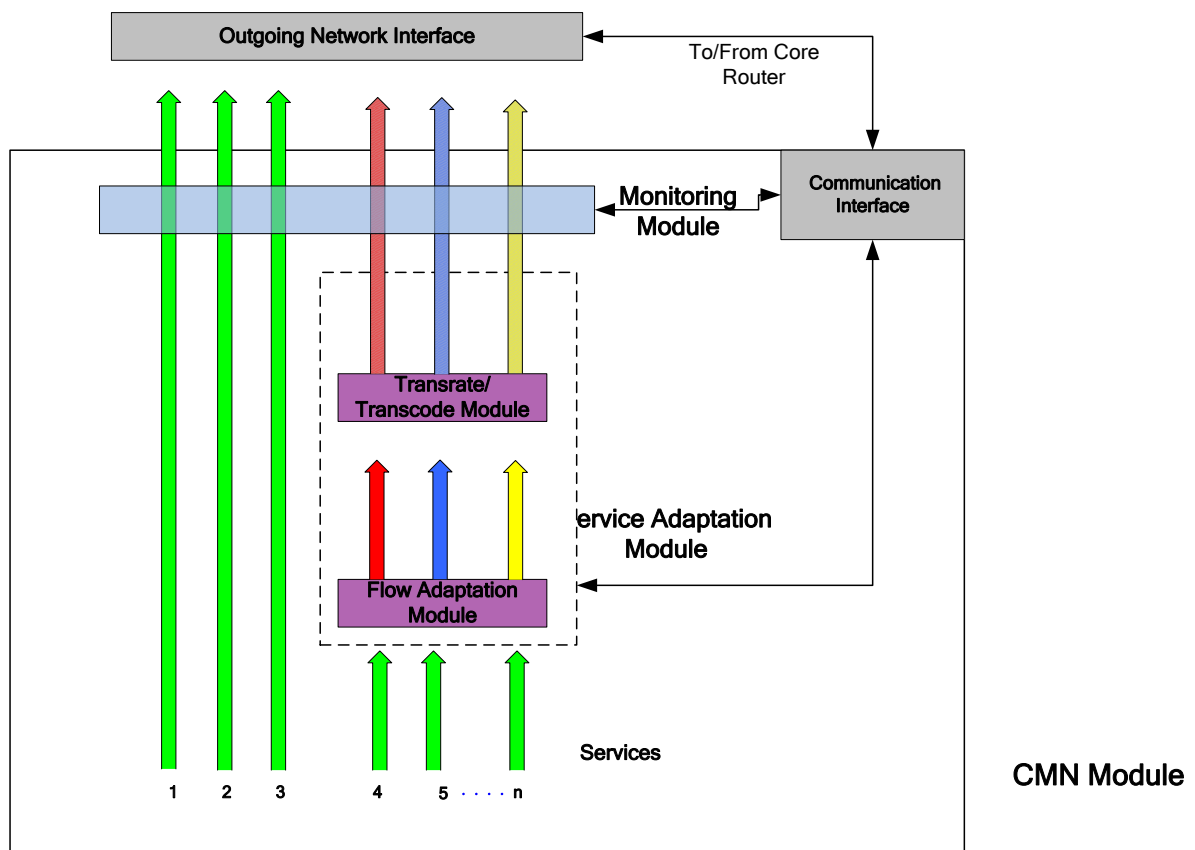
- Θα μπορεί να παρακολουθεί την κίνηση σε ένα δίκτυο και να την αξιολογεί
- Αν κρίνει πως οι συνθήκες μετάδοσης δεν είναι οι απαιτούμενες, θα μπορεί να λάβει την απόφαση να γίνει προσαρμογή της υπηρεσίας με στόχο πάντα την παροχή PQoS.
- Η προαναφερθείσα απόφαση θα μπορεί να ληφθεί για κάθε ροή στο δίκτυο ξεχωριστά.

Ο μηχανισμός προσαρμογής υπηρεσιών που κατασκευάστηκε χρησιμοποιεί αντικειμενικά μεγέθη δικτυακών επιδόσεων και αξιολογεί τα αποτελέσματα με στόχο πάντα τη παροχή PQoS. Αποτελείται από δύο κύρια μέρη, το CMN Module και το Core Module:

CMN Module (CMNM): Η εφαρμογή αυτή (Σχήμα 10) είναι υπεύθυνη για τη συλλογή/αποστολή πληροφοριών σχετικών με το είδος της κίνησης που διέρχεται από τους επίγειους κόμβους διανομής και για την προσαρμογή της υπηρεσίας - εάν λάβει μια τέτοια εντολή από τον CR (Core Router). Κατασκευάστηκε με χρήση της γλώσσας JAVA και το βασικό δομικό της στοιχείο είναι η βιβλιοθήκη Jrcar. Η εφαρμογή αποτελείται από μερικές ακόμη μονάδες λογισμικού:

- SAM (Service Adaptation Module). Το συγκεκριμένο μέρος του μηχανισμού δρα μόνο αν έχει ληφθεί απόφαση για την προσαρμογή της υπηρεσίας. Αν όντως έχει ληφθεί η απόφαση, τότε η παρούσα εφαρμογή, μέσω των FAM (Flow Adaptation Module) και TM (Transrating Module) αναλαμβάνει την προσαρμογή της υπηρεσίας
- FAM (Flow Adaptation Module). Αυτό το τμήμα του μηχανισμού, αναλαμβάνει τη σύλληψη των πακέτων για να μπορέσει να γίνει πρώτον η τροποποίηση τους και δεύτερον η εκ νέου δρομολόγησή τους. Η εφαρμογή κατασκευάστηκε με χρήση της γλώσσας C και έγινε επίσης εκτενής χρήση της βιβλιοθήκης Libnetfilter_queue
- TM (Transrating Module). Το τμήμα αυτό αναλαμβάνει την προσαρμογή της υπηρεσίας.

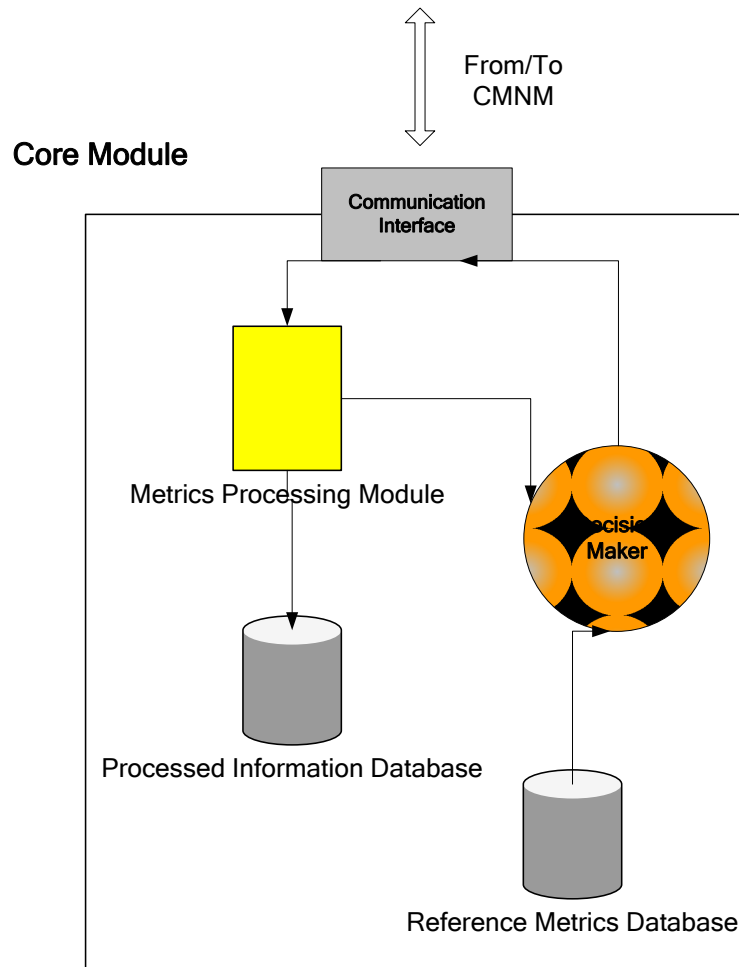
- Monitoring Module. Η εφαρμογή αυτή είναι υπεύθυνη για την παρακολούθηση των εξερχόμενων υπηρεσιών και τη δημιουργία αναφορών για την απόδοση του δικτύου.



Σχήμα 10 Η αρχιτεκτονική της εφαρμογής CMNM

Η εφαρμογή CMNM και πιο συγκεκριμένα τα τμήματα SAM και Monitoring πρέπει να βρίσκονται σε άμεση επικοινωνία με τον Core Router καθώς το SAM πρέπει να είναι έτοιμο να λάβει την εντολή προσαρμογής της υπηρεσίας αν ο Core αποφασίσει ότι η ποιότητα της υπηρεσίας δεν είναι ικανοποιητική και το Monitoring πρέπει να παρακολουθεί συνεχώς τις εξερχόμενες ροές και να δημιουργεί αναφορές μέσω των οποίων ο CR αξιολογεί το δίκτυο.

CORE Module (CM): Αυτό το τμήμα του μηχανισμού (Σχήμα 11) είναι υπεύθυνο για την επικοινωνία με τις εφαρμογές που συλλαμβάνουν την κίνηση στους επίγειους κόμβους διανομής και την αξιολόγηση της απόδοσης του δικτύου. Έπειτα, εφόσον κριθεί σκόπιμο, η εφαρμογή αυτή αναλαμβάνει την λήψη απόφασης για προσαρμογή της προσφερόμενης υπηρεσίας. Επίσης έχει τη δυνατότητα δημιουργίας γραφημάτων απο την συλληφθείσα κίνηση στους επίγειους κόμβους διανομής. Όπως και στην προηγούμενη εφαρμογή, για την κατασκευή της εφαρμογής CM χρησιμοποιήθηκε η γλώσσα JAVA



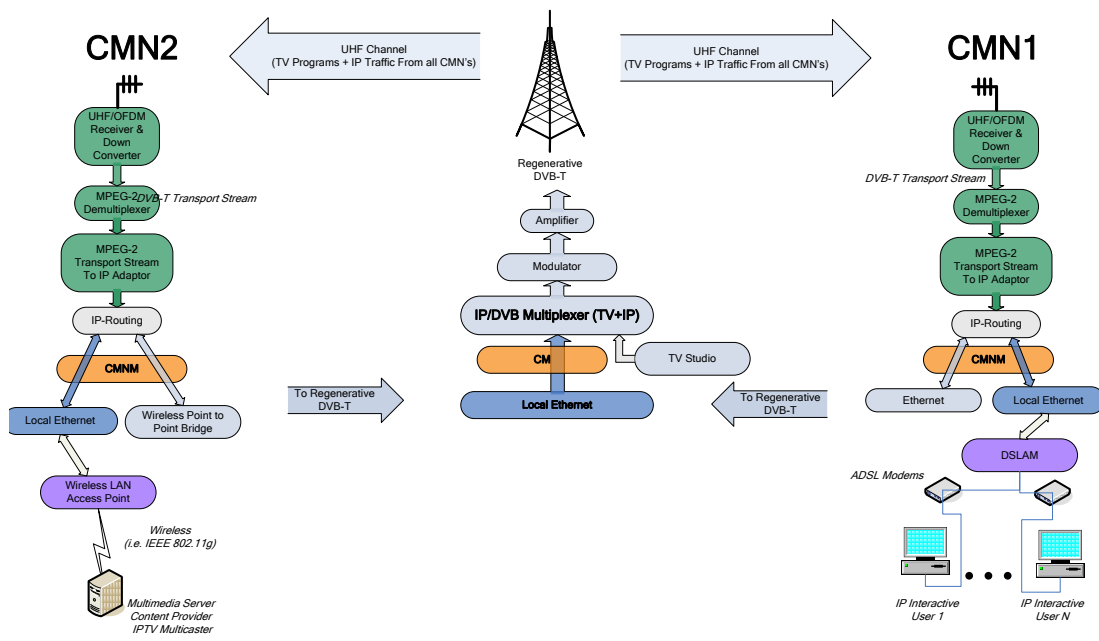
Σχήμα 11 Η αρχιτεκτονική της εφαρμογής CM

Τα βασικά στοιχεία της εφαρμογής είναι:

- RMD (Reference Metrics Database): Η βάση δεδομένων που περιέχει τις τιμές των μεγεθών που θεωρούνται το κριτήριο για την παροχή ποιότητας υπηρεσίας. Οι τιμές αυτές δίνονται απο το χρήστη με την εκτέλεση της εφαρμογής.
- MPM (Metrics Processing Module): Αυτή η μονάδα λογισμικού παραλαμβάνει τις αναφορές που δημιουργούνται απο το CMNM αναφορικά με την κίνηση και στην συνέχεια επεξεργάζεται τα δεδομένα ώστε να μετατραπούν σε χρήσιμες πληροφορίες για την αξιολόγηση της κίνησης. Έχει επίσης τη δυνατότητα να αποθηκεύει τις πληροφορίες που δημιουργεί, με στόχο τη δημιουργία μιας βάσης δεδομένων με προηγούμενες μετρήσεις, απο όπου θα μπορούν να αναχθούν γενικότερα συμπεράσματα για τις ροές.
- DM (Decision Maker): Το στοιχείο DM είναι υπεύθυνο για τη λήψη της απόφασης που αφορά στην ποιότητα της παρεχόμενης υπηρεσίας και στην ανάγκη προσαρμογής της. Επεξεργάζεται τα δεδομένα που παίρνει απο το RMD και το MPM και προωθεί την απόφαση του προς το CMNM.
- PID (Processed Information Database): Αποτελεί τη βάση δεδομένων που αποθηκεύονται οι προηγούμενες μετρήσεις του MPM.

Με βάση τον τρόπο λειτουργίας του μηχανισμού και την αρχιτεκτονική του δικτύου που περιγράφεται παραπάνω (3.1. Προτεινόμενη Αρχιτεκτονική του) γίνεται προφανές πως το CMNM, εφόσον είναι υπεύθυνο για την παρακολούθηση και σύλληψη της κίνησης, πρέπει να εκτελεστεί στους CMN, απο που διέρχεται όλη η κίνηση στο προτεινόμενο δίκτυο. Όσον αφορά στο CM, που είναι η εφαρμογή που αξιολογεί την κίνηση απο όλους τους εν δυνάμει υπάρχοντες CMN και αποφασίζει αν θα προσαρμοστεί, είναι προφανές πως θα πρέπει να εκτελεστεί στο κεντρικό σημείο του δικτύου, δηλαδή τον Core Router.

Η προτεινόμενη αρχιτεκτονική του δικτύου σε συνδυασμό με τα τμήματα του μηχανισμού, μπορεί να παρατηρηθεί συνολικά στο Σχήμα 12.



Σχήμα 12 Προτεινόμενο Δίκτυο DVB-T

4. Υλοποίηση

Η εργασία αυτή υλοποιήθηκε με εξοπλισμό του ερευνητικού εργαστηρίου Τηλεπικοινωνιών και Δικτύων του Α.Τ.Ε.Ι Κρήτης [ΠΑΣΙΦΑΗ](#). Σε όλους τους υπολογιστές που χρησιμοποιήθηκαν στο πειραματικό δίκτυο ήταν εγκατεστημένο δωρεάν λειτουργικό σύστημα Linux διανομής Gentoo.

4.1. Τελικός Χρήστης (End User)

Ο τελικός χρήστης του δικτύου που ελέγχει ο CMN2 (service Provider) χρησιμοποίησε μια ασύρματη κάρτα PCI για τη σύνδεση του στο AP (Access Point) όπου συνδέεται καλωδιακά με τον αντίστοιχο CMN. Οι τελικοί χρήστες που συνδέονται με το CMN1 χρησιμοποιούν για την επικοινωνία τους ένα απλό ADSL modem. Το modem με τη σειρά του συνδέεται με ένα DSLAM και αυτό επικοινωνεί το CMN1.

4.2. Ενδιάμεσος Κόμβος Διανομής (Cell Main Node)

Στους CMN's είναι εγκατεστημένο λειτουργικό σύστημα Linux (Gentoo). Στον CMN2 όπου είναι συνδεδεμένος ο Service Provider υπάρχει εγκατεστημένη μια ενσύρματη κάρτα δικτύου όπου συνδέεται στο AP, μια ασύρματη κάρτα δικτύου για σύνδεση με το κεντρικό σημείο διανομής όπου χρησιμοποιείτε σαν κανάλι επιστροφής και μία κάρτα ψηφιακής τηλεόρασης για επικοινωνία από την πλατφόρμα DVB-T προς τον CMN1 όπου χρησιμοποιείται σαν κανάλι καθόδου.

Ο CMN1 χρησιμοποιήθηκε για να συνδεθούν οι τελικοί χρήστες-αποδέκτες της υπηρεσίας. Είναι και εκεί εγκατεστημένη μια κάρτα δικτύου και μια κάρτα ψηφιακής τηλεόρασης όπως στο CMN2, για τους ίδιους λόγους. Η μόνη διαφορά εντοπίζεται στο κανάλι επιστροφής όπου αντί για WLAN έχουμε μια απλή σύνδεση Ethernet για την επικοινωνία με την πλατφόρμα.

Οι CMN's αναλαμβάνουν την παρακολούθηση της απόδοσης του δικτύου και την αναφορά αυτής της απόδοσης στο κεντρικό κόμβο διανομής.

4.3. Κεντρικό Σημείο Εκπομπής (Πλατφόρμα DVB-T) – Core Router (CR)

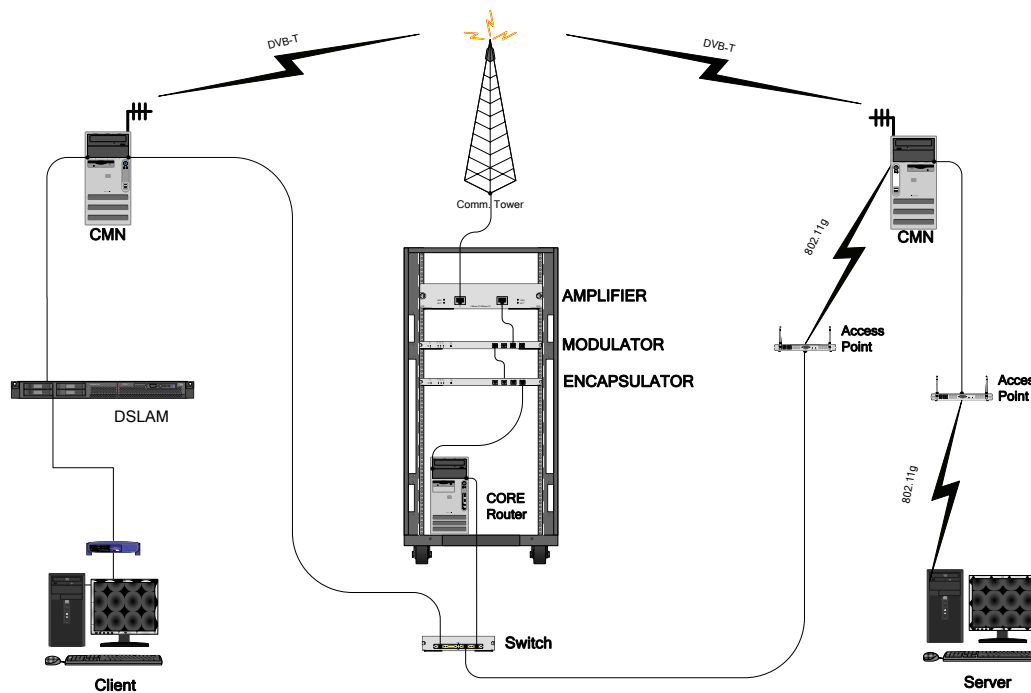
Το κεντρικό σημείο εκπομπής αποτελείται από τον Core Router, τον ενθυλακωτή-πολυπλέκτη (Encapsulator-Multiplexer), τον διαμορφωτή (Modulator) από τον ενισχυτή (Amplifier). Όπως αναφέραμε ο CR λαμβάνει την κίνηση από τους CMN (μέσω του WLAN από τον CMN2 και μέσω του Ethernet από τον CMN1). Έχει την ευθύνη επίσης να αξιολογεί τις αναφορές που του αποστέλλουν οι δύο CMN και αποφασίζει με βάση τα κριτήρια που έχουν δοθεί, αν η κίνηση έχει την απαραίτητη ποιότητα. Αν αποφασίσει πως μια προσαρμογή της υπηρεσίας είναι επιβεβλημένη, τότε εκτελεί το κομμάτι του λογισμικού που την πραγματοποιεί. Όποια και αν είναι αυτή η απόφαση, ο CR στη συνέχεια προωθεί την κίνηση στον Encapsulator-Multiplexer μέσω μιας ενσύρματης κάρτας δικτύου για να ενθυλακωθούν τα πακέτα στο MPEG2-TS μαζί με τα τηλεοπτικά προγράμματα που θα μπορούσαν να προσφέρονται σαν μέρος του μπουκέτου υπηρεσιών από έναν MPEG 2 TS Server. Ο Encapsulator-Multiplexer ρυθμίστηκε να δεσμεύει για τις IP υπηρεσίες 7 Mbps, ενώ για τα ψηφιακά τηλεοπτικά προγράμματα 14 Mbps.



Σχήμα 13 Ο Διαμορφωτής COFDM που Παρουσιάζει τα Στοιχεία της Διαμόρφωσης

Ο διαμορφωτής DVB-T (COFDM) ρυθμίστηκε σε διαμόρφωση 64QAM, ρυθμό κώδικα 7/8 και διάστημα φύλαξης (guard interval) ίσο με το 1/32 του μήκους συμβόλου (Σχήμα Σχήμα 13). Οι παράμετροι αυτές αντιστοιχούν σε ωφέλιμο ρυθμό δεδομένων ίσο με 21.11 Mbps. Επίσης ως φέροντα σήματα δηλώνονται 8K. Η συχνότητα του σήματος εκπομπής είναι στα 626 MHz (κανάλι 40) με επίπεδο δύναμης (power level) 1Watt και το συγκεκριμένο κανάλι έχει εύρος 8MHz.

Στο Σχήμα 14 φαίνεται το σύστημα όπως υλοποιήθηκε στο εργαστήριο ΠΑΣΙΦΑΗ.

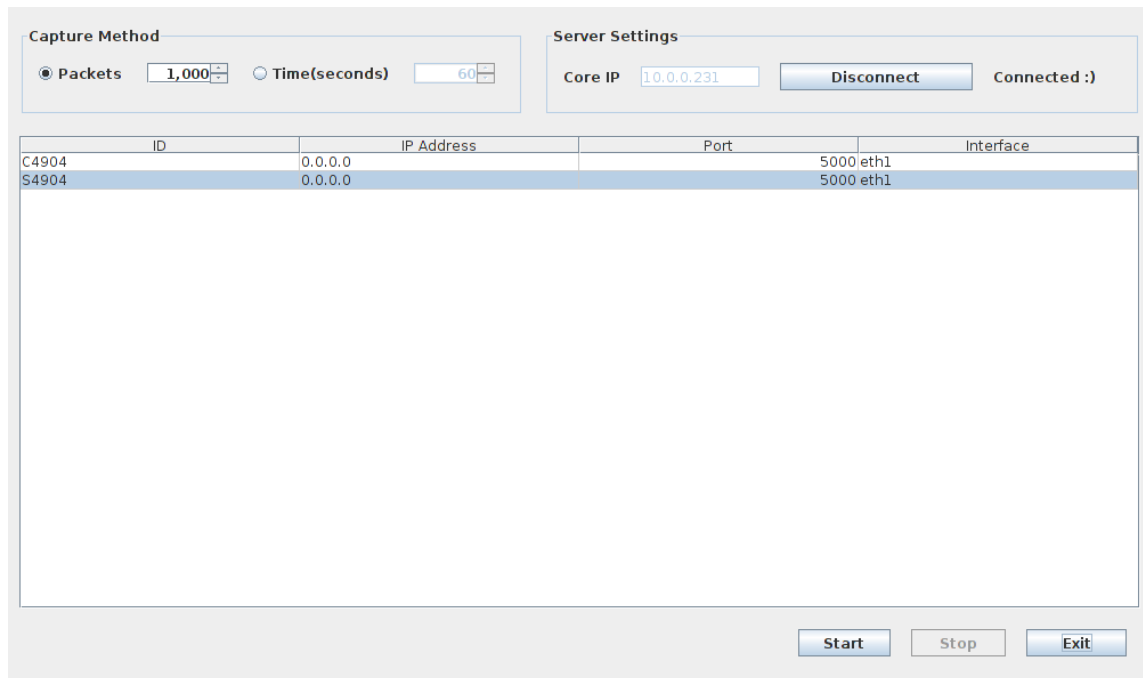


Σχήμα 14 Υλοποίηση του DVB-T δικτύου

4.4. Λειτουργία του Μηχανισμού

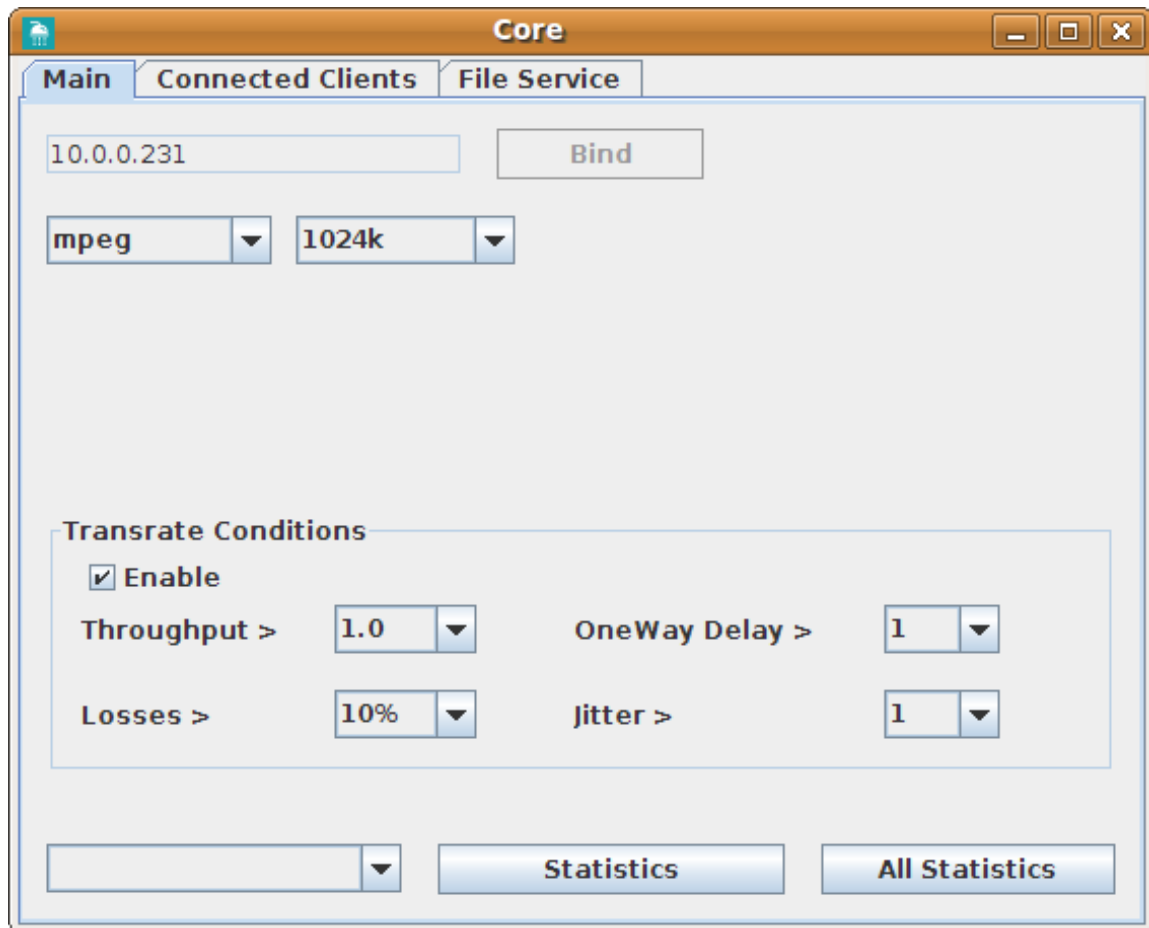
Σε αυτό το σημείο έχουν ήδη περιγραφεί οι στόχοι με βάση τους οποίους κατασκευάστηκε ο μηχανισμός προσαρμογής υπηρεσιών όπως επίσης και το δίκτυο ευρείας κάλυψης στο οποίο θα δοκιμαστεί και θα αξιολογηθεί. Ακολουθεί μια αναλυτικότερη αναφορά στον τρόπο λειτουργίας του μηχανισμού και μια τοποθέτηση του μηχανισμού στο δίκτυο που περιγράφηκε παραπάνω.

Η εφαρμογή CMNM (CMN Module – Σχήμα 15) εκτελείται στους δύο CMN. Ο χρήστης πριν αρχίσει να συλλαμβάνει την κίνηση πρέπει να δώσει κάποιες απαραίτητες παραμέτρους όπως την διεύθυνση του Core Module (CM) για να επικοινωνήσουν όταν θα έχει ετοιμαστεί η αναφορά απόδοσης του δικτύου, τον αριθμό των πακέτων που επιθυμεί να συλλαμβάνει η εφαρμογή πριν στείλει τη αναφορά και τον αριθμό των ροών που θα παρακολουθούνται. Σε κάθε ροή, με την είσοδο της στο πρόγραμμα, ανατίθεται ένας τυχαίος αναγνωριστικός αριθμός (max. 4 ψηφία) που χρησιμοποιείται γενικώς για την επεξεργασία της ροής απο το CMNM και απο το CM. Ο χρήστης μπορεί να παρακολουθεί όλες ροές επιθυμεί, σε κάθε περίπτωση όμως χρειάζεται να δηλώσει την κάρτα δικτύου που θα εισέρχεται/εξέρχεται η ροή, τη διεύθυνση της (υπενθυμίζεται σε αυτό το σημείο ότι ως διεύθυνση ορίζεται η IP διεύθυνση ενός υπολογιστή σε συνδυασμό με τη θύρα του υπολογιστή) καθώς και το αν αυτή η ροή υπόκειται στο δίκτυο του Service Provider ή αν υπόκειται στο δίκτυο του CMN1, δηλαδή στο δίκτυο των τελικών χρηστών/αποδεκτών της υπηρεσίας. Αυτή η πληροφορία δίνεται με την είσοδο της ροής στην εφαρμογή. Αν η ροή εξέρχεται απο το δίκτυο του CMN2 μπαίνει, όπως φαίνεται και στο Σχήμα 14, το γράμμα «S» (απο τη λέξη Server) μπροστά απο τον αναγνωριστικό αριθμό, ενώ αν η ροή εξέρχεται απο το δίκτυο του CMN1 μπαίνει μπροστά απο τον αναγνωριστικό αριθμό το γράμμα «C» (απο τη λέξη Client) Με τον τρόπο αυτό, γίνεται ευκολότερη η ανάλυση των αποτελεσμάτων στο CM.



Σχήμα 15 Η εφαρμογή CMNM

Η εφαρμογή CM (Σχήμα 16) εκτελείται στον Core Router και είναι υπεύθυνη για την αξιολόγηση της απόδοσης του δικτύου. Όπως και στην εφαρμογή CMNM πριν εκτελεστεί η εφαρμογή, ο χρήστης πρέπει να εισάγει κάποια δεδομένα, όπως τη διεύθυνση που πρέπει να δέχεται συνδέσεις ο CM για να επικοινωνήσει με τους CMNM, όπως επίσης και τα κριτήρια με βάση τα οποία θα γίνει η αξιολόγηση της απόδοσης. Επίσης το CM έχει τη δυνατότητα παρουσίασης των αποτελεσμάτων που συλλέγει σε ξεχωριστά γραφήματα.



Σχήμα 16 Η εφαρμογή CM

Ας υποθεθεί πως ξεκινά η παροχή της υπηρεσίας από το Service Provider προς τους αποδέκτες της υπηρεσίας. Το CM αρχίζει να συλλαμβάνει την κίνηση. Καθώς συλλαμβάνει τα πακέτα ταυτόχρονα δομεί ένα XML αρχείο δημιουργώντας ένα element (βλ. 2.1.5. XML σελ. 20) για κάθε ροή που παρακολουθείται (Σχήμα 17 Το αρχείο XML). Κάτω από αυτό το element δημιουργείται ακόμα ένα element για κάθε πακέτο. Τα βασικά χαρακτηριστικά του πακέτου αποθηκεύονται με τη μορφή attributes σε κάθε element του πακέτου. Τα χαρακτηριστικά του πακέτου που αποθηκεύονται είναι τα εξής: αναγνωριστικό του πακέτου (packet id), ο χρόνος που το πακέτο έφθασε (timestamp), το ωφέλιμο φορτίο του πακέτου (payload) και ο έλεγχος εγκυρότητας του πακέτου (checksum). Αυτά τα χαρακτηριστικά είναι αρκετά για τον υπολογισμό των δικτυακών επιδόσεων που θα ληφθούν υπόψη για την αξιολόγηση της απόδοσης.


```
in.gr - Google Search
file:///home/ni...as/output.xml
file:///home/ni...as/output.xml
<packet id="36050" timestamp="1272573290.154826" payload="1472" checksum="4139194976"/>
<packet id="36059" timestamp="1272573290.154826" payload="1472" checksum="538334975"/>
<packet id="36060" timestamp="1272573290.166041" payload="1472" checksum="3918679732"/>
</flow>
<!-- flow flow_id="C3030" -->
<packet id="35861" timestamp="1272573287.934187" payload="1472" checksum="2835749080"/>
<packet id="35862" timestamp="1272573287.947532" payload="1472" checksum="710938700"/>
<packet id="35863" timestamp="1272573287.956756" payload="1472" checksum="762339093"/>
<packet id="35864" timestamp="1272573287.967906" payload="1472" checksum="2340322247"/>
<packet id="35865" timestamp="1272573287.979060" payload="1472" checksum="1342780332"/>
<packet id="35866" timestamp="1272573287.990222" payload="1472" checksum="1679559837"/>
<packet id="35867" timestamp="1272573288.1523" payload="1472" checksum="2057342368"/>
<packet id="35868" timestamp="1272573288.12728" payload="1472" checksum="4074804140"/>
<packet id="35869" timestamp="1272573288.24005" payload="1472" checksum="2328288375"/>
<packet id="35870" timestamp="1272573288.35159" payload="1472" checksum="4108775021"/>
<packet id="35871" timestamp="1272573288.46374" payload="1472" checksum="409946364"/>
<packet id="35872" timestamp="1272573288.57589" payload="1472" checksum="2298566498"/>
<packet id="35873" timestamp="1272573288.68804" payload="1472" checksum="2708842287"/>
<packet id="35874" timestamp="1272573288.80015" payload="1472" checksum="2363286967"/>
<packet id="35875" timestamp="1272573288.93266" payload="1472" checksum="1035562903"/>
<packet id="35876" timestamp="1272573288.102936" payload="1472" checksum="1195977549"/>
<packet id="35877" timestamp="1272573288.113932" payload="1472" checksum="3921132996"/>
<packet id="35878" timestamp="1272573288.124929" payload="1472" checksum="4425860328"/>
<packet id="35879" timestamp="1272573288.13695" payload="1472" checksum="1149602272"/>
<packet id="35880" timestamp="1272573288.147426" payload="1472" checksum="2956724307"/>
<packet id="35881" timestamp="1272573288.158566" payload="1472" checksum="521430608"/>
<packet id="35882" timestamp="1272573288.169784" payload="1472" checksum="332018774"/>
<packet id="35883" timestamp="1272573288.180953" payload="1472" checksum="902790331"/>
<packet id="35884" timestamp="1272573288.192169" payload="1472" checksum="1360417394"/>
<packet id="35885" timestamp="1272573288.203389" payload="1472" checksum="817790642"/>
<packet id="35886" timestamp="1272573288.214595" payload="1472" checksum="1427060180"/>
<packet id="35887" timestamp="1272573288.225813" payload="1472" checksum="1030347078"/>
<packet id="35888" timestamp="1272573288.237026" payload="1472" checksum="2287698372"/>
<packet id="35889" timestamp="1272573288.251369" payload="1472" checksum="1807098220"/>
<packet id="35890" timestamp="1272573288.259460" payload="1472" checksum="2418251474"/>
<packet id="35891" timestamp="1272573288.270671" payload="1472" checksum="3039954029"/>
<packet id="35892" timestamp="1272573288.281885" payload="1472" checksum="765577451"/>
</flow>
<!-- flow flow_id="C3030" -->
<packet id="35861" timestamp="1272573287.934187" payload="1472" checksum="2835749080"/>
<packet id="35862" timestamp="1272573287.947532" payload="1472" checksum="710938700"/>
<packet id="36060" timestamp="1272573290.166041" payload="1472" checksum="3918679732"/>
Done
```

Σχήμα 17 Το αρχείο XML

Θα πρέπει να σημειωθεί εδώ, πως η επιλογή να αποθηκεύονται τα χαρακτηριστικά με τη μορφή XML αρχείου, έγινε με γνώμονα την διάδοση που τυχαίνει να έχει αυτή η τεχνολογία καθώς και την περαιτέρω ανάπτυξη του μηχανισμού. Αυτά τα θέματα όμως θα συζητηθούν εκτενέστερα στο 6. Συμπεράσματα – Μελλοντικές Προτάσεις όπου θα αναλυθούν οι μελλοντικές προτάσεις.

Αφού δημιουργήσουν το αρχείο XML οι CMNM το στέλνουν στο CM που είναι υπεύθυνος για τον υπολογισμό των δικτυακών χαρακτηριστικών και την αξιολόγηση τους (Σχήμα 18). Τα χαρακτηριστικά που υπολογίζονται είναι ο ρυθμός μετάδοσης δεδομένων (Bitrate), η μονόδρομη καθυστέρηση (One-Way Delay), η διακύμανση καθυστέρησης (Jitter) και το ποσοστό απωλειών επί τοις εκατό (Loss Rate). Απο εκεί και πέρα είτε η απόδοση του δικτύου χαρακτηρίζεται επαρκής για την παροχή QoS οπότε η παροχή της υπηρεσίας συνεχίζεται κανονικά, είτε η απόδοση χαρακτηρίζεται ανεπαρκής (Σχήμα 19) και μπαίνει σε εφαρμογή το SAM (Service Adaptation Module) (σελ. 29).

```

File Edit View Terminal Help
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 13.3797503565 Bytes/sec, Server Bitrate= 13.3584427236 Bytes/sec, Jitter= 5.25016 msec, Losses= 2/1000:0.2%, One Way Delay= 0.03885msec
Analysis elapsed time:0.311 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 13.7726126802 Bytes/sec, Server Bitrate= 13.7457187772 Bytes/sec, Jitter= 8.23242 msec, Losses= 2/1000:0.2%, One Way Delay= 0.01327msec
Analysis elapsed time:0.266 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 13.7726926407 Bytes/sec, Server Bitrate= 13.7649407956 Bytes/sec, Jitter= 5.82483 msec, Losses= 1/1000:0.1%, One Way Delay= 0.01174msec
Analysis elapsed time:0.289 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 13.3616838938 Bytes/sec, Server Bitrate= 13.3467788546 Bytes/sec, Jitter= 6.9033 msec, Losses= 3/1000:0.3%, One Way Delay= 0.02033msec
Analysis elapsed time:0.332 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Transrate Request @10.0.0.231, target flow: C4904
Client Bitrate= 12.7482784266 Bytes/sec, Server Bitrate= 10.7224701543 Bytes/sec, Jitter= 12.83338 msec, Losses= 102/1000:10.2%, One Way Delay= 0.32443msec
Analysis elapsed time:0.477 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 8.0086475263 Bytes/sec, Server Bitrate= 10.1751991828 Bytes/sec, Jitter= 11.70168 msec, Losses= 143/1000:14.3%, One Way Delay= 0.01405msec
Analysis elapsed time:0.667 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 10.8912023778 Bytes/sec, Server Bitrate= 10.9026110419 Bytes/sec, Jitter= 11.53085 msec, Losses= 151/1000:15.1%, One Way Delay= 0.31152msec
Analysis elapsed time:0.343 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 16.9913690346 Bytes/sec, Server Bitrate= 12.26585832 Bytes/sec, Jitter= 9.93494 msec, Losses= 236/1000:23.6%, One Way Delay= 0.03106msec
Analysis elapsed time:0.448 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 10.724433536 Bytes/sec, Server Bitrate= 12.4041813151 Bytes/sec, Jitter= 10.17131 msec, Losses= 66/1000:6.6%, One Way Delay= 0.0082msec
Analysis elapsed time:0.543 secs

```

Σχήμα 18 Αξιολόγηση Κίνησης - Έξοδος του CM

```

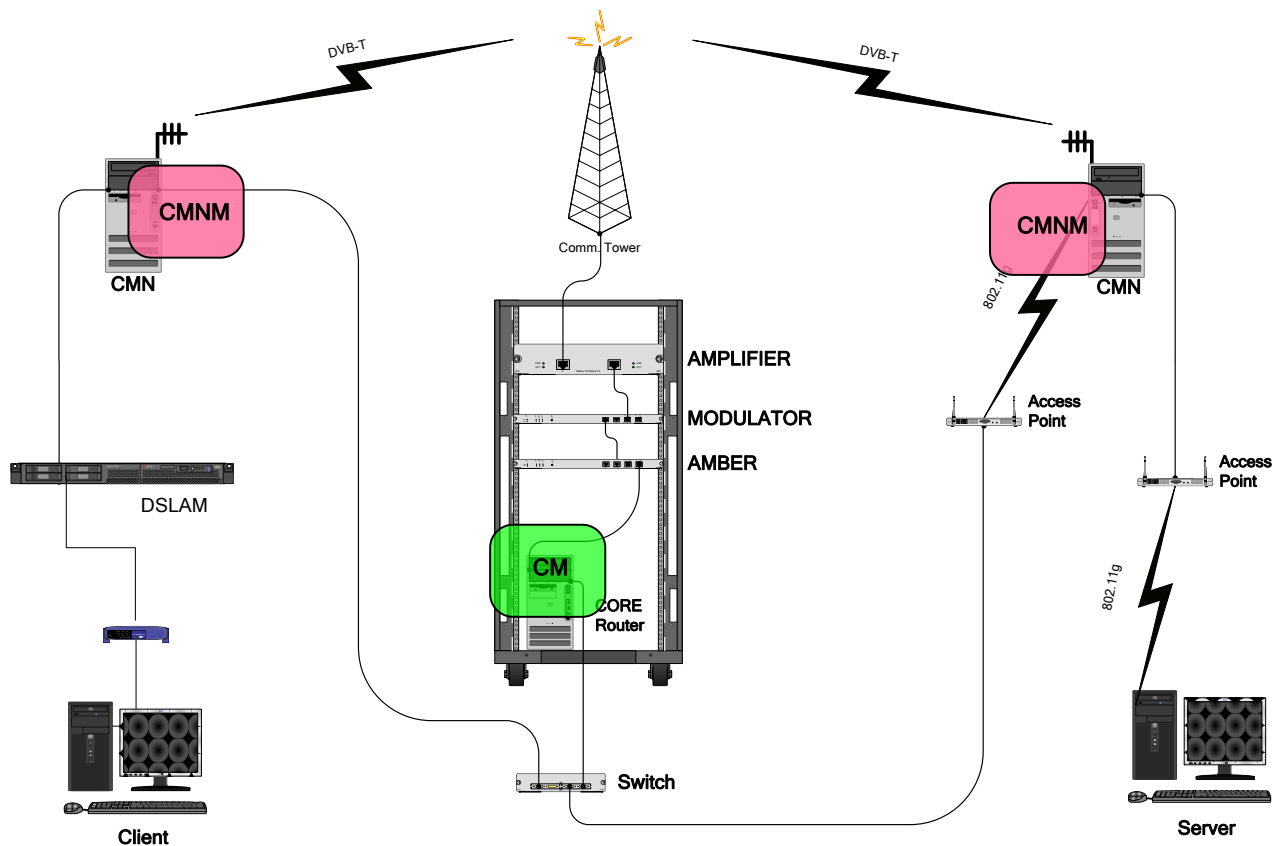
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 13.3616838938 Bytes/sec, Server Bitrate= 13.3467788546 Bytes/sec, Jitter= 6.9033 msec, Losses= 3/1000:0.3%, One Way Delay= 0.02033msec
Analysis elapsed time:0.312 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Transrate Request @10.0.0.231, target flow: C4904
Client Bitrate= 12.7482784266 Bytes/sec, Server Bitrate= 10.7224701543 Bytes/sec, Jitter= 12.83338 msec, Losses= 102/1000:10.2%, One Way Delay= 0.32443msec
Analysis elapsed time:0.477 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 8.0086475263 Bytes/sec, Server Bitrate= 10.1751991828 Bytes/sec, Jitter= 11.70168 msec, Losses= 143/1000:14.3%, One Way Delay= 0.01405msec
Analysis elapsed time:0.667 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 16.9913690346 Bytes/sec, Server Bitrate= 12.26585832 Bytes/sec, Jitter= 9.93494 msec, Losses= 236/1000:23.6%, One Way Delay= 0.03106msec
Analysis elapsed time:0.448 secs
Reading File From 10.0.0.231
Flow S4904 Searching for C4904
Flow Found
Analyzing... C4904,S4904
Client Bitrate= 10.724433536 Bytes/sec, Server Bitrate= 12.4041813151 Bytes/sec, Jitter= 10.17131 msec, Losses= 66/1000:6.6%, One Way Delay= 0.0082msec
Analysis elapsed time:0.543 secs

```

Σχήμα 19 Απόφαση του CM για προσαρμογή

Η βασική λειτουργία του SAM είναι χρησιμοποιώντας το IPTables και τη libnetfilter_queue να οδηγεί την κίνηση σε συγκεκριμένες ουρές και προς στιγμήν να δρομολογεί την κίνηση στον ίδιο το CMN που παρέχει την υπηρεσία. Έτσι ο CMN έχει τη δυνατότητα λαμβάνοντας την υπηρεσία να την προσαρμόσει αναλόγως και να την δρομολογήσει εκ νέου προς τους τελικούς αποδέκτες (τους χρήστες του δικτύου CMN1).

Μια εικόνα του δικτύου στο σύνολο του σε συνδυασμό με το μηχανισμό δίνεται απο το Σχήμα 20



Σχήμα 20 Το δίκτυο μαζί με το μηχανισμό

4.5. Πειραματικά Σενάρια

Για την αξιολόγηση του μηχανισμού πραγματοποιήθηκαν 3 σενάρια όπου ο Service Provider αναπαριστά την παροχή μιας Video υπηρεσίας με τη μέθοδο Streaming, χρησιμοποιώντας την εφαρμογή VLC σε έναν χρήστη (1^ο σενάριο), σε τρεις χρήστες (2^ο σενάριο) και σε πέντε χρήστες (3^ο σενάριο). Οι χρήστες προσομοιώνονται με την αποστολή με τη μέθοδο μονοεκπομπής (unicast) των ροών σε διαφορετικές θύρες του ίδιου υπολογιστή. Ο τελικός χρήστης αναπαράγει τις ροές επίσης με το πρόγραμμα VLC.

Τα Video είναι συμπιεσμένο με τον κωδικοποιητή MPEG2 στο 1Mbit και ο ήχος είναι συμπιεσμένος με τον κωδικοποιητή MPEG-Layer 3 στα 128Kbit. Η διάρκεια του είναι 5 λεπτά.

Για κάθε σενάριο πραγματοποιήθηκαν 3 πειραματικές μετρήσεις:

-Αρχικά, πραγματοποιείται μια μέτρηση αναφοράς όπου παρατηρείται η απόδοση του δικτύου υπό κανονικές συνθήκες.

-Έπειτα πραγματοποιείται μια μέτρηση όπου περίπου στη μέση της μετάδοσης με το πρόγραμμα tc θα περιοριστεί το εύρος ζώνης κάποιον ροών. Η μέτρηση αυτή θέτει τα όρια της ποιότητας υπηρεσίας.

-Στην τελευταία μέτρηση, τίθεται σε λειτουργία ο μηχανισμός προσαρμογής υπηρεσιών. Στη μέση της μετάδοσης περιορίζεται εκ νέου το εύρος ζώνης κάποιον ροών και στη συνέχεια αξιολογείται η λειτουργία του μηχανισμού και θα παρατηρείται ο χρόνος απόκρισης του.

Σημειώνεται εδώ πως η απόδοση του δικτύου αξιολογείται κάθε 1000 πακέτα. Εμπειρικά παρατηρήθηκε πως αυτό συνέπιπτε με μια χρονική περίοδο των 10 δευτερολέπτων περίπου. Η επιλογή αυτή έγινε για να αποφευχθεί ο κίνδυνος να παίρνουν τα CMNM δείγματα πολύ συχνά και να οδηγούνται σε λάθος συμπεράσματα. Επίσης, με τον τρόπο αυτό εξομαλύνονται τα γραφήματα των αποτελεσμάτων.

Απο τις μετρήσεις δημιουργούνται **4 γραφήματα** : το γράφημα του ρυθμού μετάδοσης δεδομένων (**Bitrate**), της μονόδρομης καθυστέρησης (**One-Way Delay**), της διακύμανσης καθυστέρησης (**Jitter**) και του ποσοστού απωλειών επί τοις εκατό (**Loss Rate**). Επίσης απο κάθε μέτρηση έχουν ληφθεί και **screenshots** τα οποία χρησιμοποιούνται στην αξιολόγηση/επιβεβαίωση των γραφημάτων.

Απο μετρήσεις που επαναλήφθηκαν πριν απο την ολοκλήρωση της εργασίας μπόρεσε να εξαχθεί ένα βασικό συμπέρασμα, ότι το «κατώφλι» ποιότητας το οποίο αναφέρθηκε στο κεφάλαιο που εξηγείται η έννοια QoS και αποτελεί βασικό στόχο του κατασκευασθέντος μηχανισμού, βρίσκεται περίπου στο 10% στο ποσοστό απωλειών. Όσον αφορά τις αυξήσεις του Jitter και του One-Way Delay, αυτές καταπολεμούνται αποτελεσματικά απο τους buffers και τους αλγόριθμους διόρθωσης λαθών της εφαρμογής VLC.

Έτσι λοιπόν, τέθηκε το βασικό κριτήριο για την προσαρμογή της υπηρεσίας να είναι το ποσοστό απωλειών και πιο συγκεκριμένα αν αυτό το ποσοστό ξεπερνά το επίπεδο του 10%.

1^ο Σενάριο

Αποστέλλεται μια ροή στο 1^ο σενάριο στη θύρα 5000 του τελικού αποδέκτη. Στη δεύτερη μέτρηση το πρόγραμμα περιορίζει το εύρος ζώνης της ροής στο 1Mbit. Στην τρίτη μέτρηση η υπηρεσία κωδικοποιείται από το μηχανισμό, όταν χρειάζεται προσαρμογή, στα 512Kbit.

2^ο Σενάριο

Στο 2^ο σενάριο αποστέλλονται τρεις ροές στον τελικό αποδέκτη η πρώτη στη θύρα 5000, η δεύτερη στη 5001, η τρίτη στη 5002. Στη δεύτερη μέτρηση οι περιορισμοί που γίνονται με το tc είναι οι εξής: το εύρος ζώνης της πρώτης ροής μειώνεται στο 1Mbit, της δεύτερης στα 900Kbit και της τρίτης στα 700Kbit. Η προσαρμογή της υπηρεσίας, στην τρίτη μέτρηση, κωδικοποιεί την υπηρεσία για όλες τις ροές στα 512Kbit.

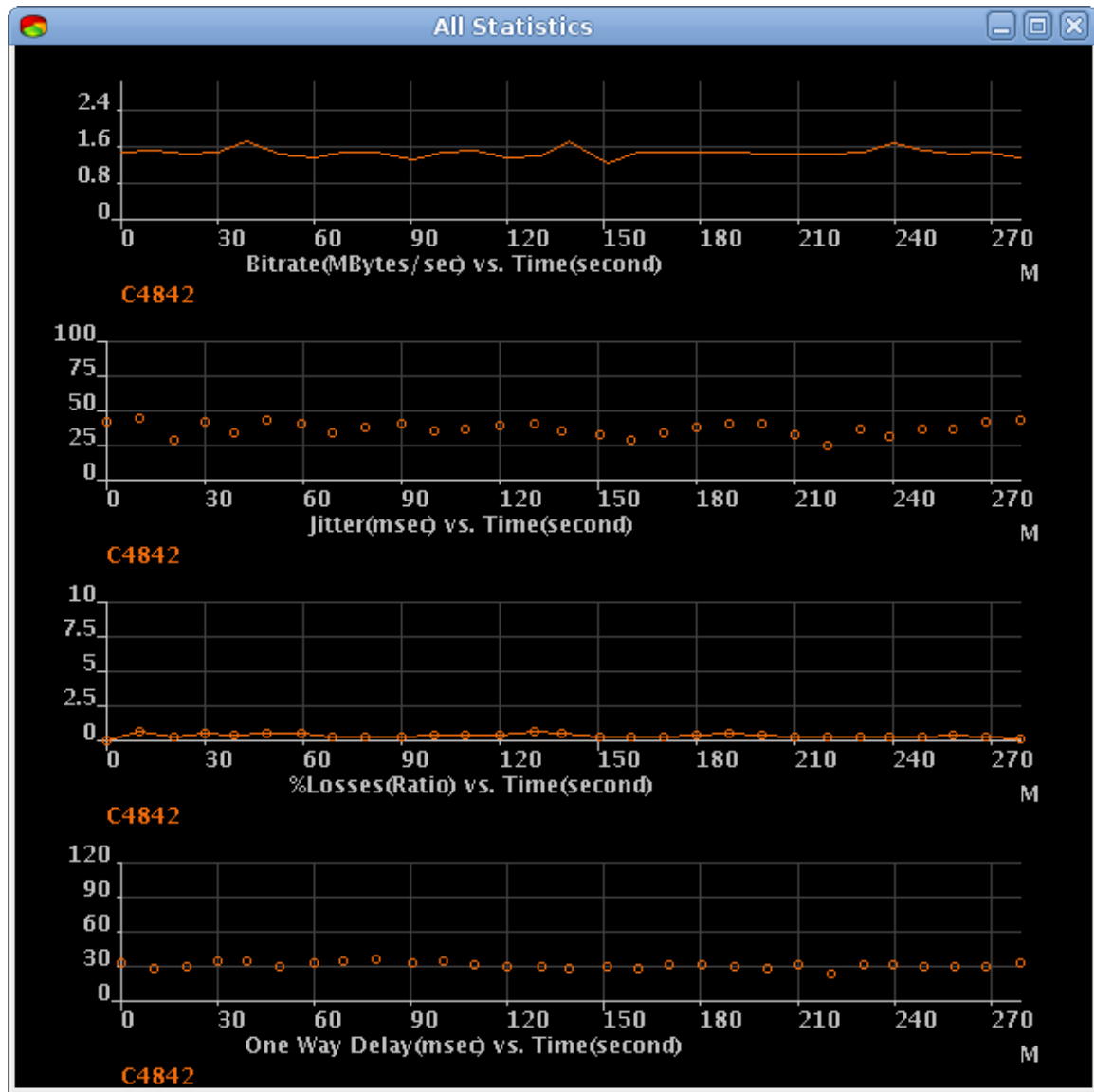
3^ο Σενάριο

Στο 3^ο σενάριο αποστέλλονται πέντε ροές στον τελικό αποδέκτη η πρώτη στη θύρα 5000, η δεύτερη στη 5001, η τρίτη στη 5002, η τέταρτη στη 5003 και η πέμπτη στη 5004. Στη δεύτερη μέτρηση οι περιορισμοί που γίνονται με το tc είναι οι εξής: το εύρος ζώνης της πρώτης ροής μειώνεται στο 1Mbit, της δεύτερης στα 900Kbit και της τρίτης στα 700Kbit, της τέταρτης στα 500Kbit. Η πέμπτη ροή μένει ανεπηρέαστη. Η προσαρμογή της υπηρεσίας, στην τρίτη μέτρηση, κωδικοποιεί την υπηρεσία για τις τρεις πρώτες ροές, όταν κρίνει ότι κάτι τέτοιο χρειάζεται, στα 512Kbit και για την τέταρτη ροή, στα 256Kbit, για να παρέχει QoS. Η πέμπτη ροή μένει ξανά ανεπηρέαστη.

5. Ανάλυση και Παρουσίαση Αποτελεσμάτων

5.1. 1^ο Σενάριο

1^η Μέτρηση



Σχήμα 21 1^ο Σενάριο – 1^η Μέτρηση

Απο τη μέτρηση στο Σχήμα 21 μπορεί κάποιος να παρατηρήσει στο γράφημα του bitrate ότι το bitrate της ροής είναι λίγο παραπάνω (1.5 Mbit) από αυτό που θα έπρεπε

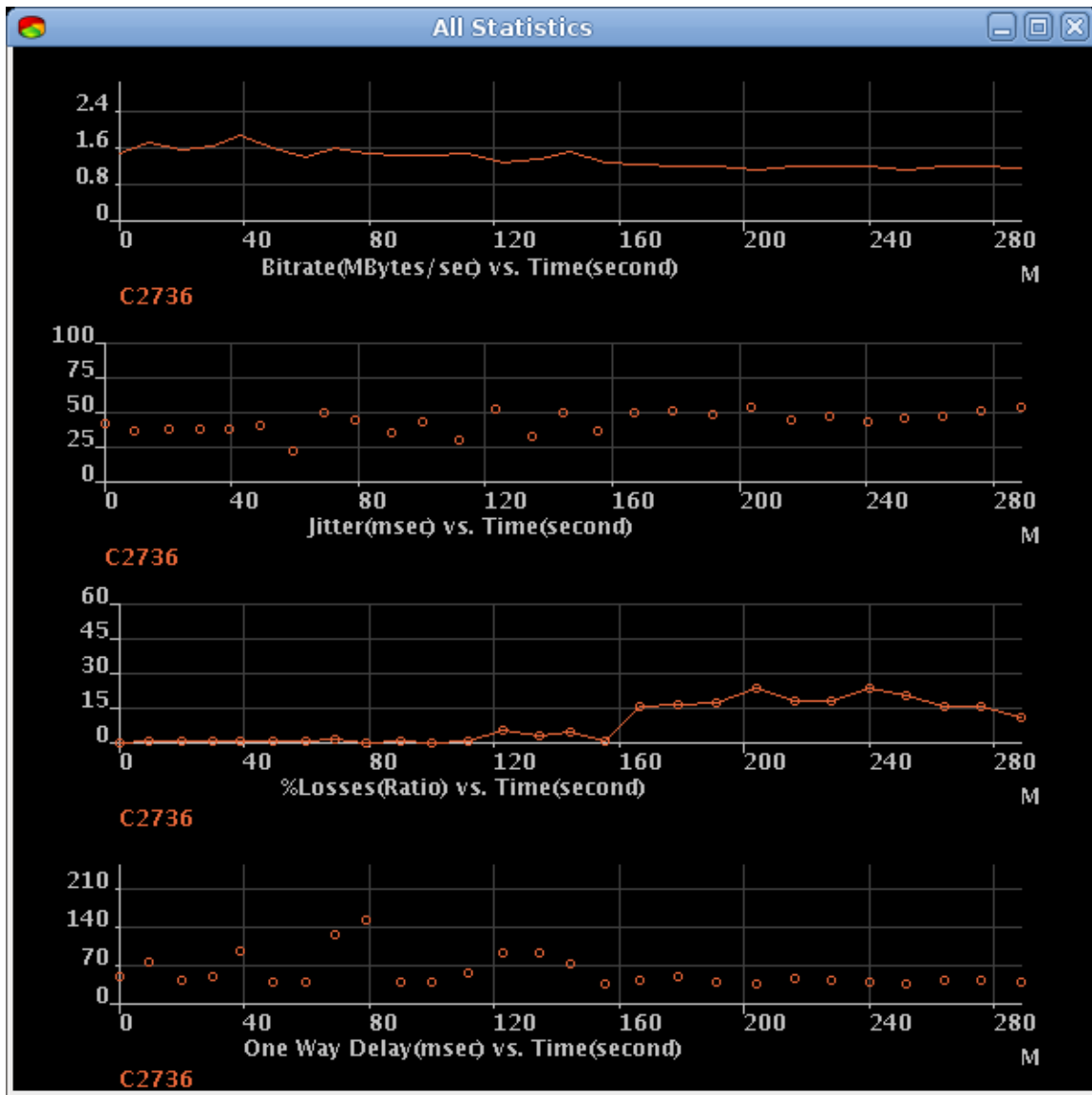
να είναι ($1\text{Mbit} + 128\text{Kbit} = 1.128\text{Kbit}$) κάτι που οφείλεται στις επικεφαλίδες του πακέτου (overhead). Στο γράφημα του Loss rate το ποσοστό απωλειών κυμαίνεται γύρω στο 1% κάτι που όπως θα φανεί και απο τα screenshots (Σχήμα 22) κρίνεται πολύ ικανοποιητικό. Επίσης κανονικές και αναμενόμενες είναι και οι τιμές του Jitter και του One-way Delay.



Σχήμα 22 1^ο Σενάριο – 1^η Μέτρηση - Screenshot

2^η Μέτρηση

Παρατηρείται πως στα 150'' που εκτελούνται οι εντολές που περιορίζουν το εύρος ζώνης, μειώνεται αναμενόμενα στο 1Mbit το bitrate και το Loss Rate αυξάνεται πάνω απο 15%. Παρατηρείται επίσης μια μικρή άνοδος στο Jitter καθώς και στο One-Way Delay (Σχήμα 23).



Σχήμα 23 1ο Σενάριο – 2η Μέτρηση

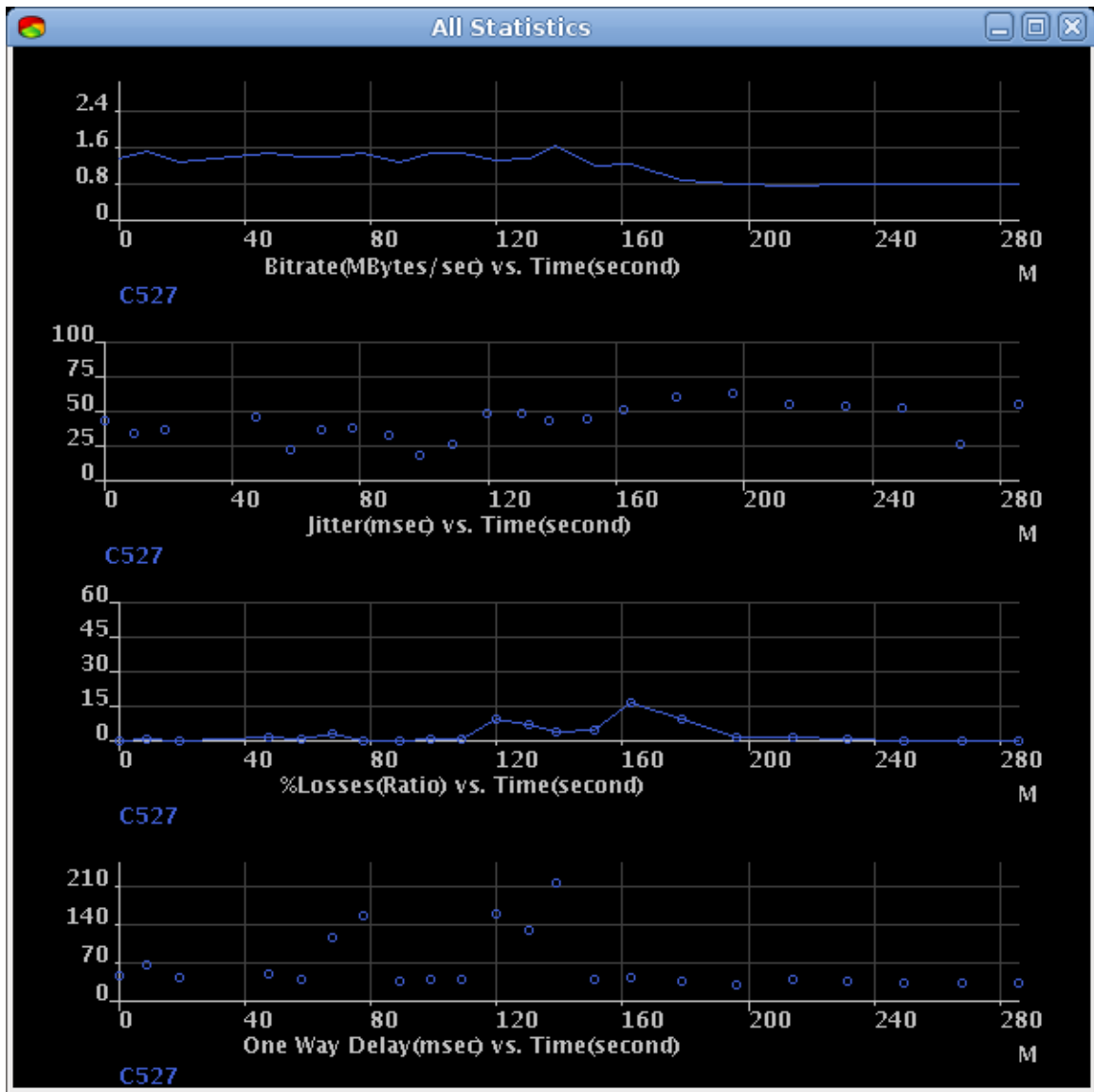
Αν και η μείωση του εύρους ζώνης είναι στα όρια του συνολικού bitrate του video, παρατηρείται μεγάλη μείωση στην ποιότητα της εικόνας (Σχήμα 24)



Σχήμα 24 1ο Σενάριο – 2η Μέτρηση - Screenshot

3^η Μέτρηση

Παρατηρείται πως η ροή έχει κανονικά αποτελέσματα μέχρι και τη στιγμή που περιορίζεται το εύρος ζώνης. Τη στιγμή εκείνη το ποσοστό απωλειών ξεπερνά το 10% και το ο μηχανισμός προσαρμογής τίθεται σε λειτουργία. Μετά απο περίπου **7-8 sec (χρόνος απόκρισης)** video έχει υποστεί μια προσαρμογή, έχει κωδικοποιηθεί στα 512Kbit και ο χρήστης απολαμβάνει ξανά την υπηρεσία (όπως δείχνουν και τα screenshots στο Σχήμα 27). Μετά την καινούργια κωδικοποίηση παρατηρείται μείωση του Loss Rate σε μηδενικά επίπεδα καθώς και μείωση στο One-Way Delay (Σχήμα 25).



Σχήμα 25 1^ο Σενάριο – 3^η Μέτρηση



Σχήμα 26 1^ο Σενάριο – 3^η Μέτρηση - Screenshot πριν την προσαρμογή

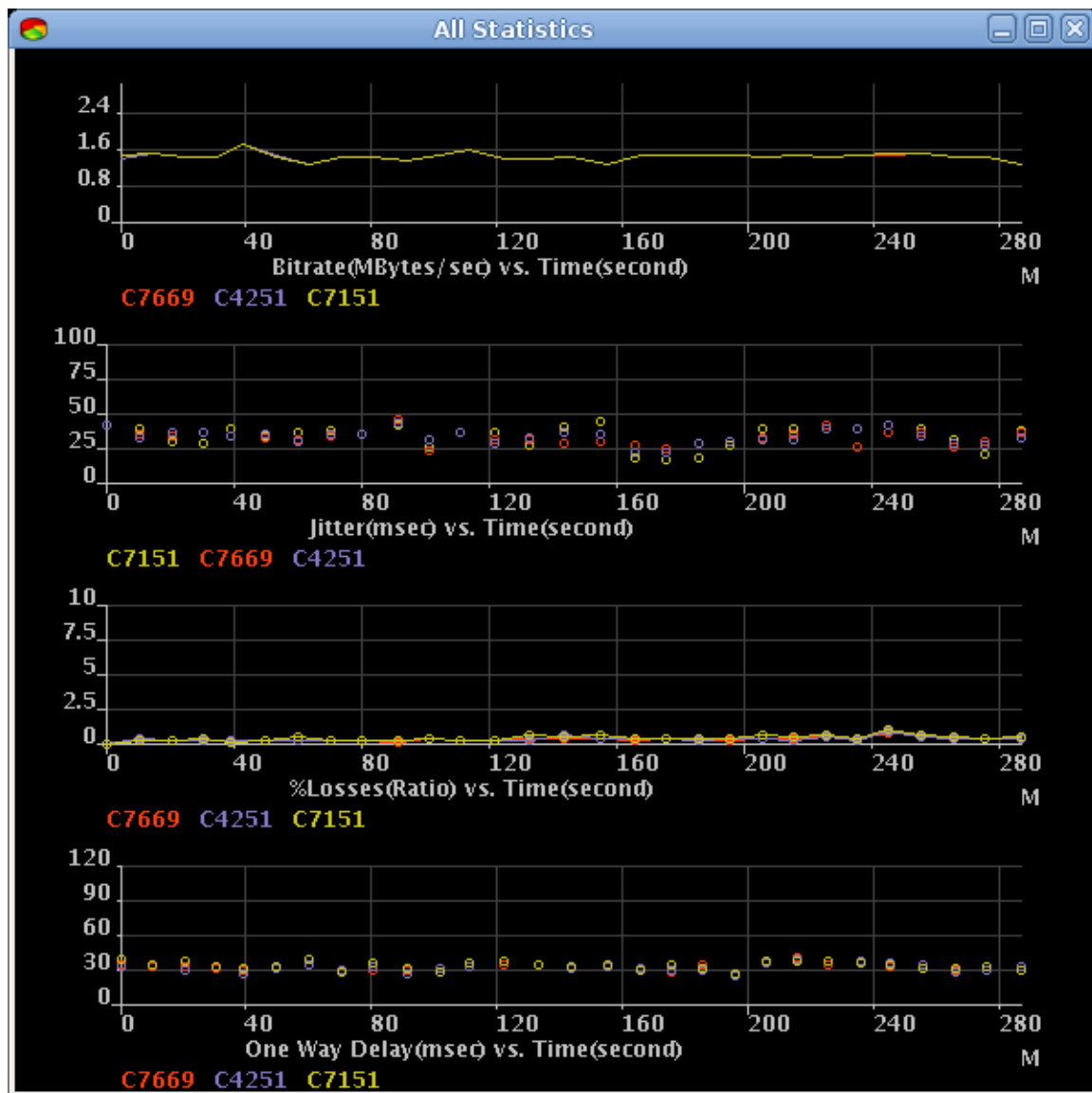


Σχήμα 27 1^ο Σενάριο – 3^η Μέτρηση - Screenshot μετά την προσαρμογή

5.2. 2^ο Σενάριο

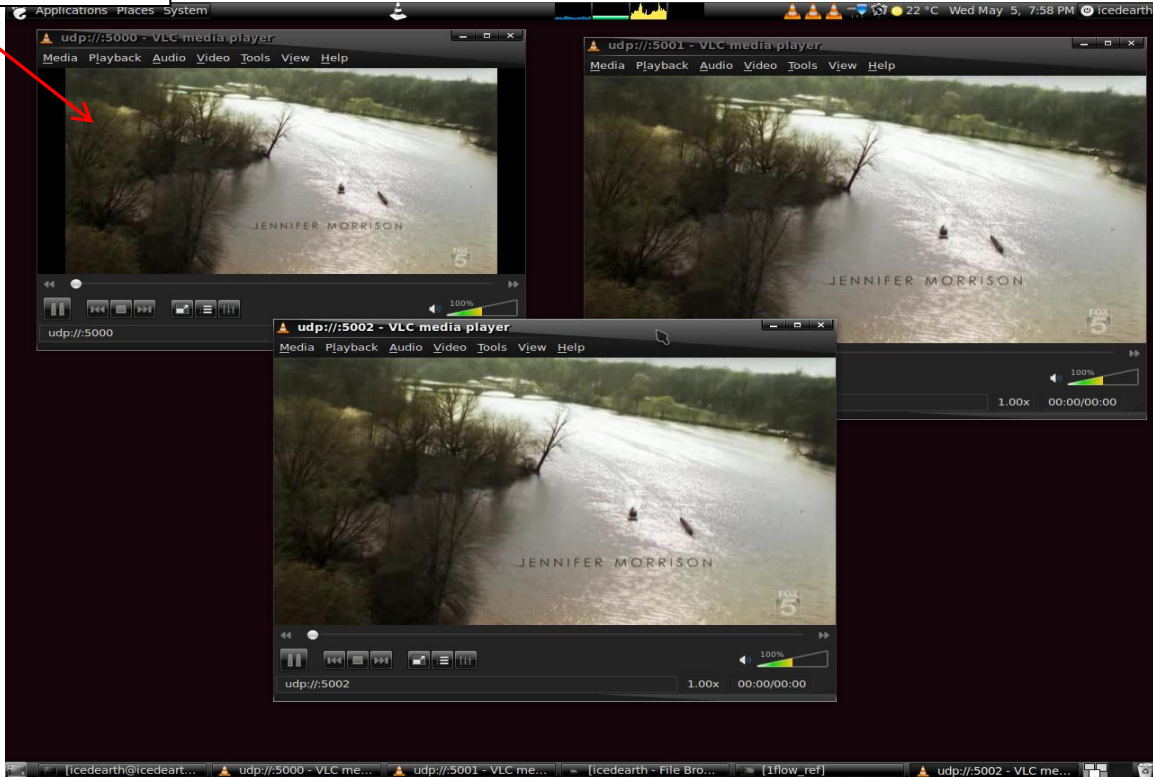
1^η Μέτρηση

Στη μέτρηση με τρεις τελικούς αποδέκτες παρατηρούνται οι ίδιες μετρήσεις με προηγούμενως που κρίθηκαν άκρως ικανοποιητικές. Όπως φαίνεται και απο τα screenshots στο Σχήμα 29 η ποιότητα της εικόνας κρίνεται και αυτή άκρως ικανοποιητική.



Σχήμα 28 2ο Σενάριο – 1η Μέτρηση

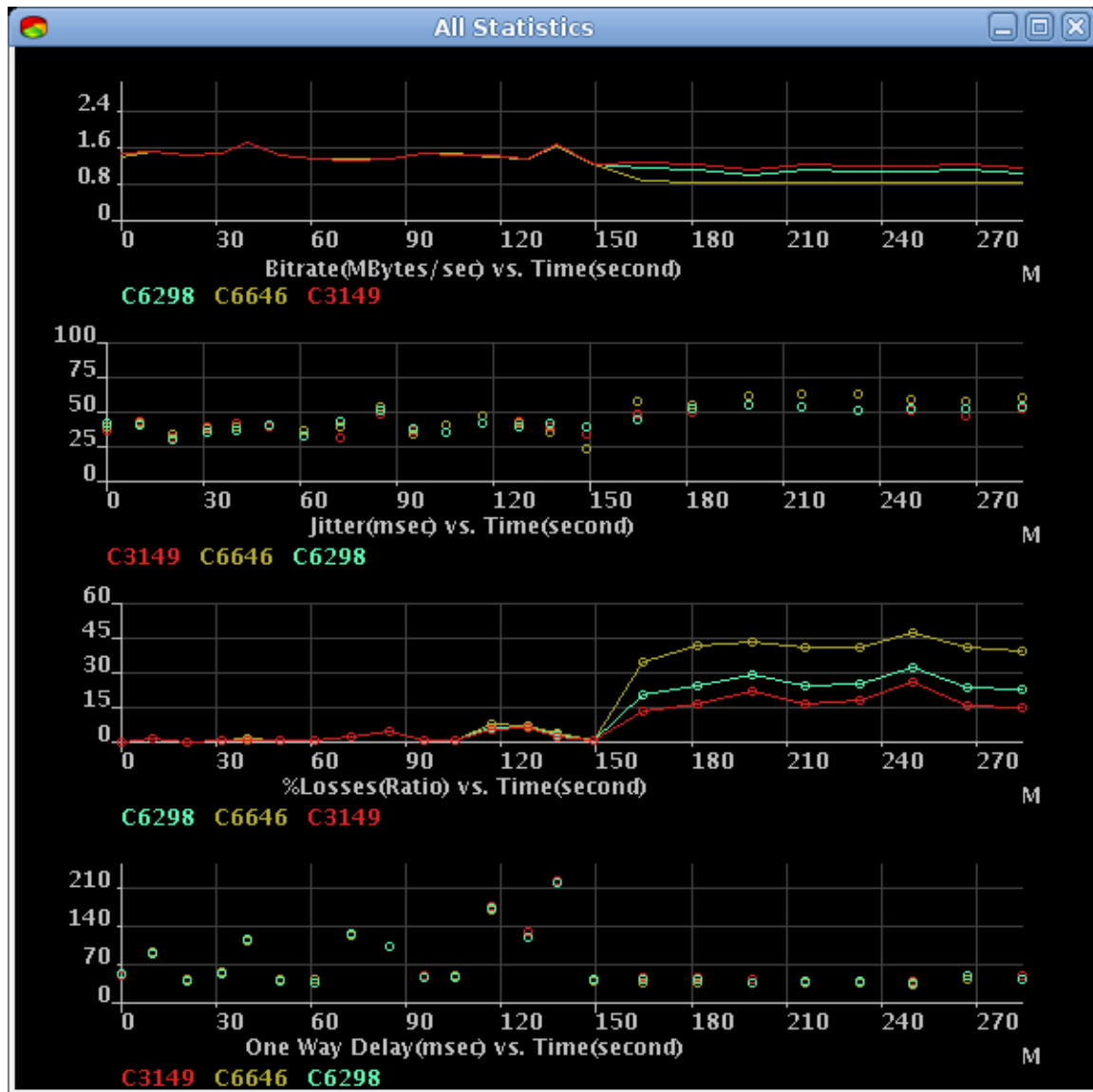
AllStreams@1MBps



Σχήμα 29 2ο Σενάριο – 1η Μέτρηση - Screenshot

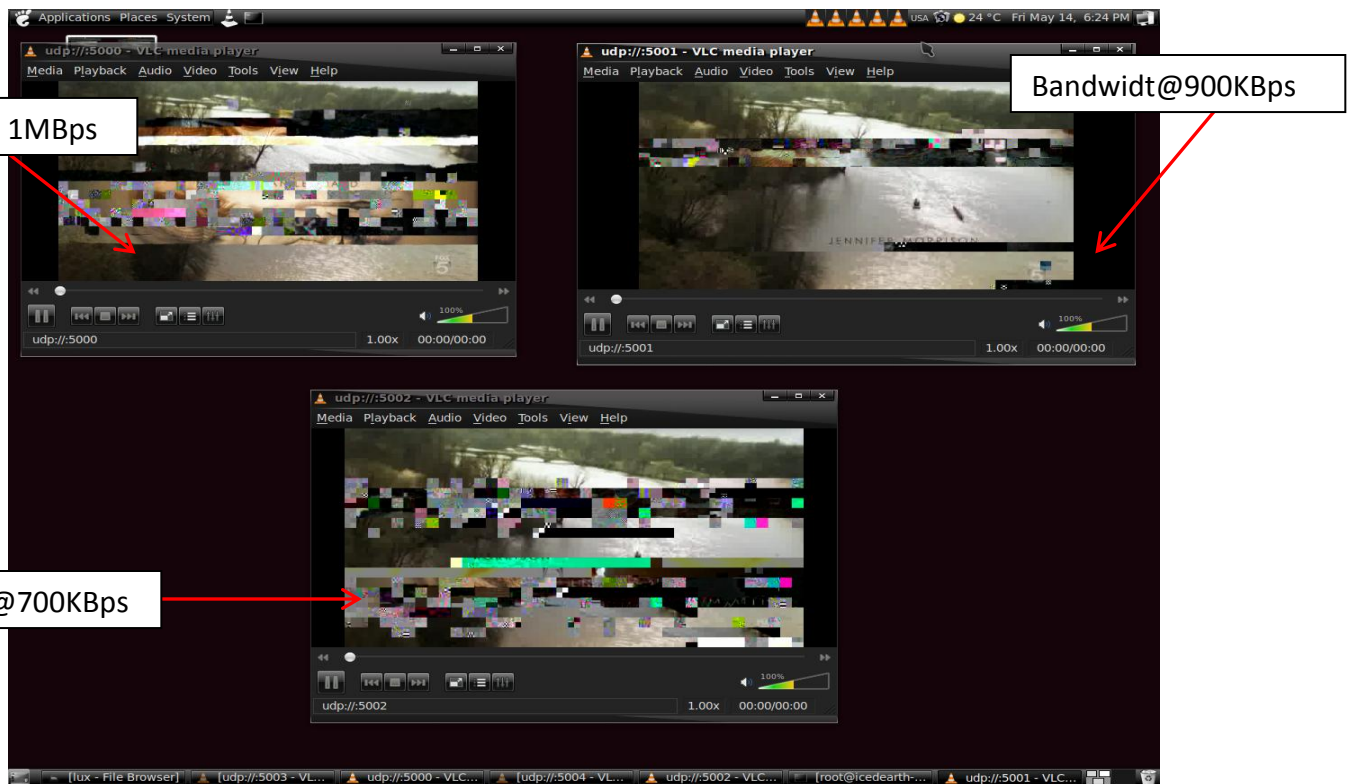
2^η Μέτρηση

Στη συγκεκριμένη μέτρηση μπορεί να παρατηρηθεί η ακρίβεια των γραφημάτων. Κάποιος μπορεί ξεκάθαρα να παρατηρήσει τις διαφορές μεταξύ της θύρας 5000 και 5002 (1Mbit και 700Kbit αντίστοιχα) και την αντιστρόφως ανάλογη αύξηση στο Loss Rate. Φαίνεται επίσης και αύξηση στο Jitter όπως και μεγάλη αύξηση στο One-Way Delay τη στιγμή που περιορίστηκε το εύρος ζώνης (Σχήμα 30).



Σχήμα 30 2^ο Σενάριο - 2^η Μέτρηση

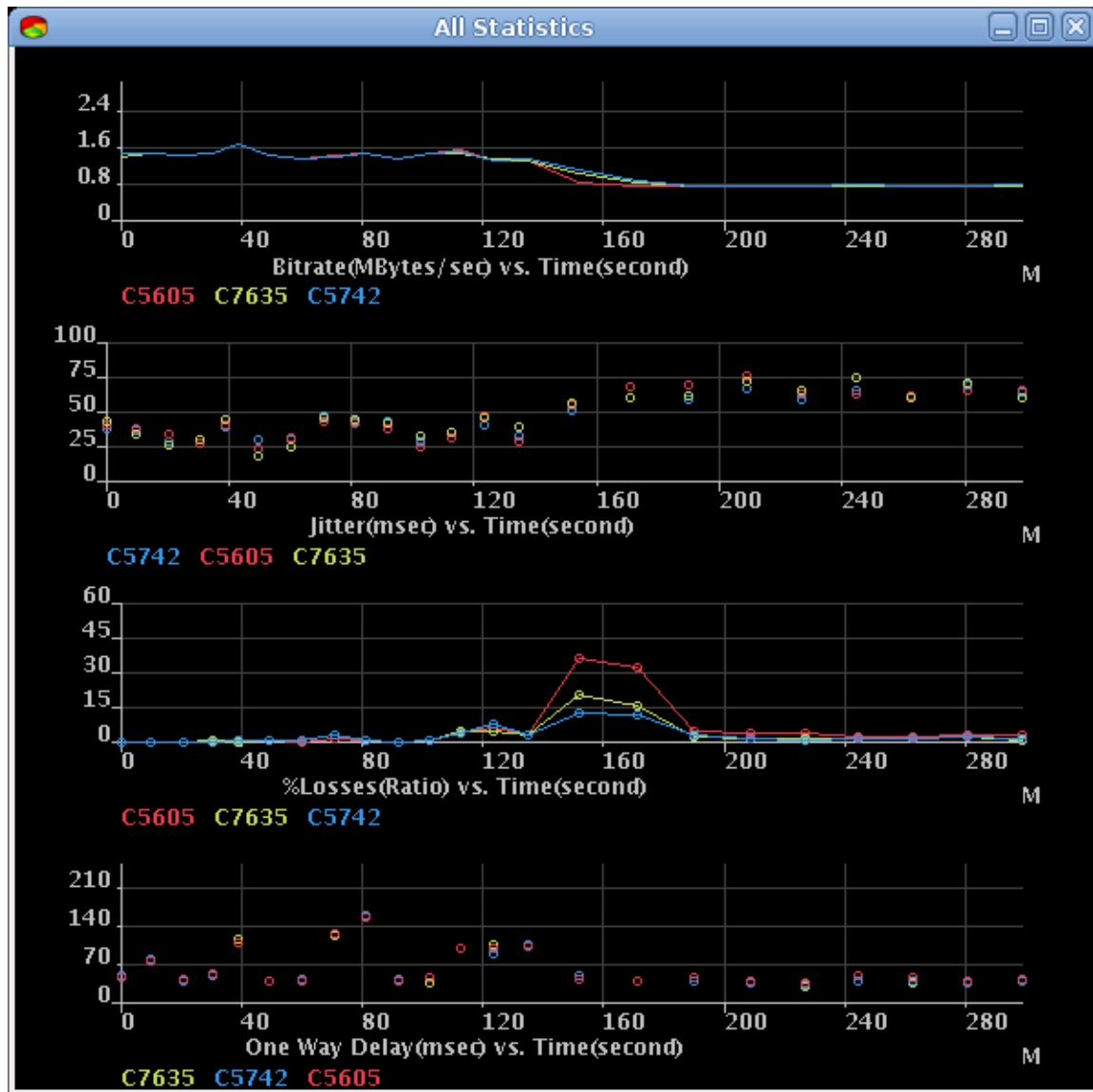
Η ποιότητα της εικόνας παρουσιάζει μια ελλάσωνα κλιμάκωση, εξαρτώμενη και απο τον περιορισμό του εύρους ζώνης (Σχήμα 31).



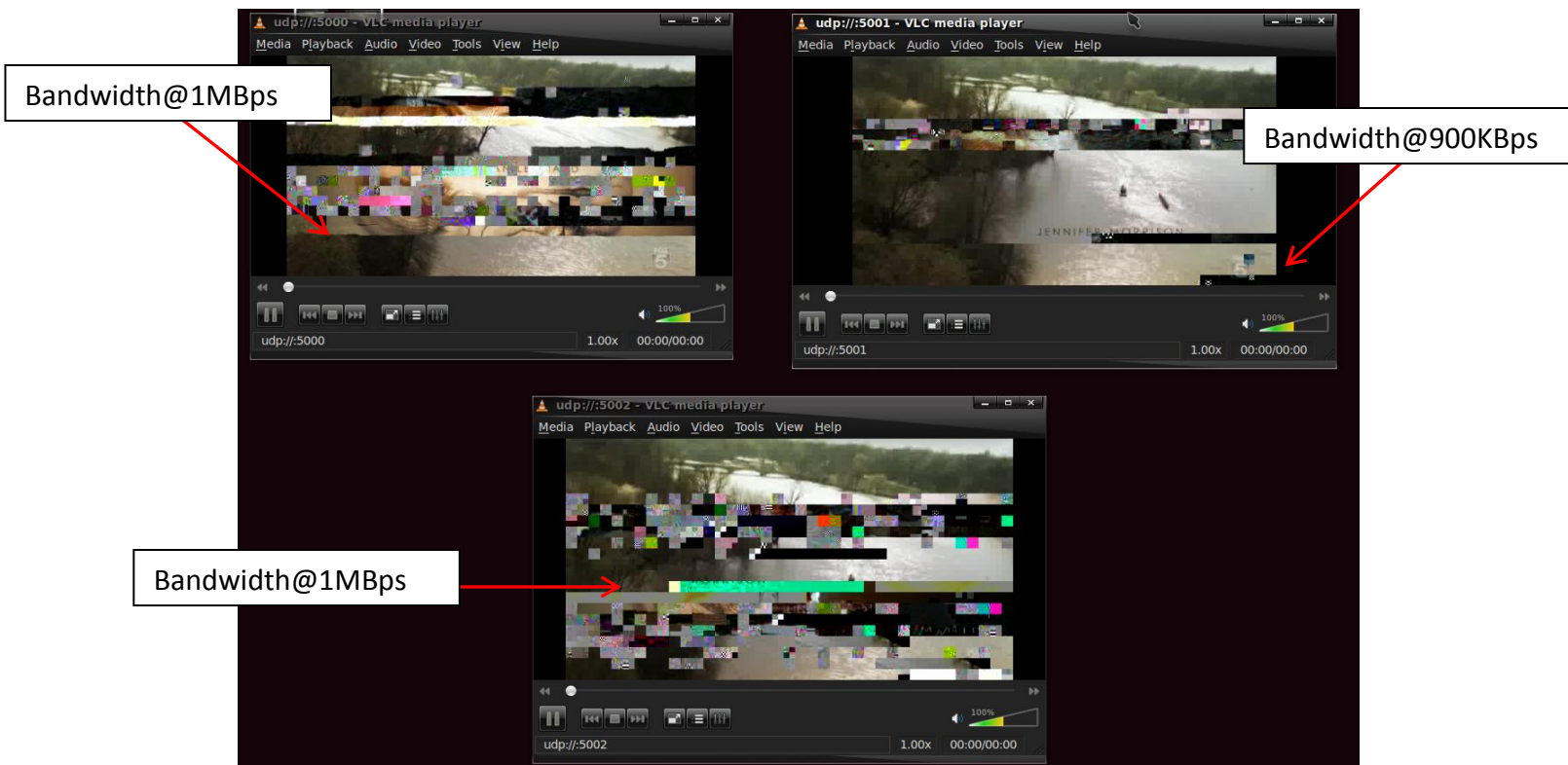
Σχήμα 31 2^ο Σενάριο - 2^η Μέτρηση - Screenshot

3^η Μέτρηση

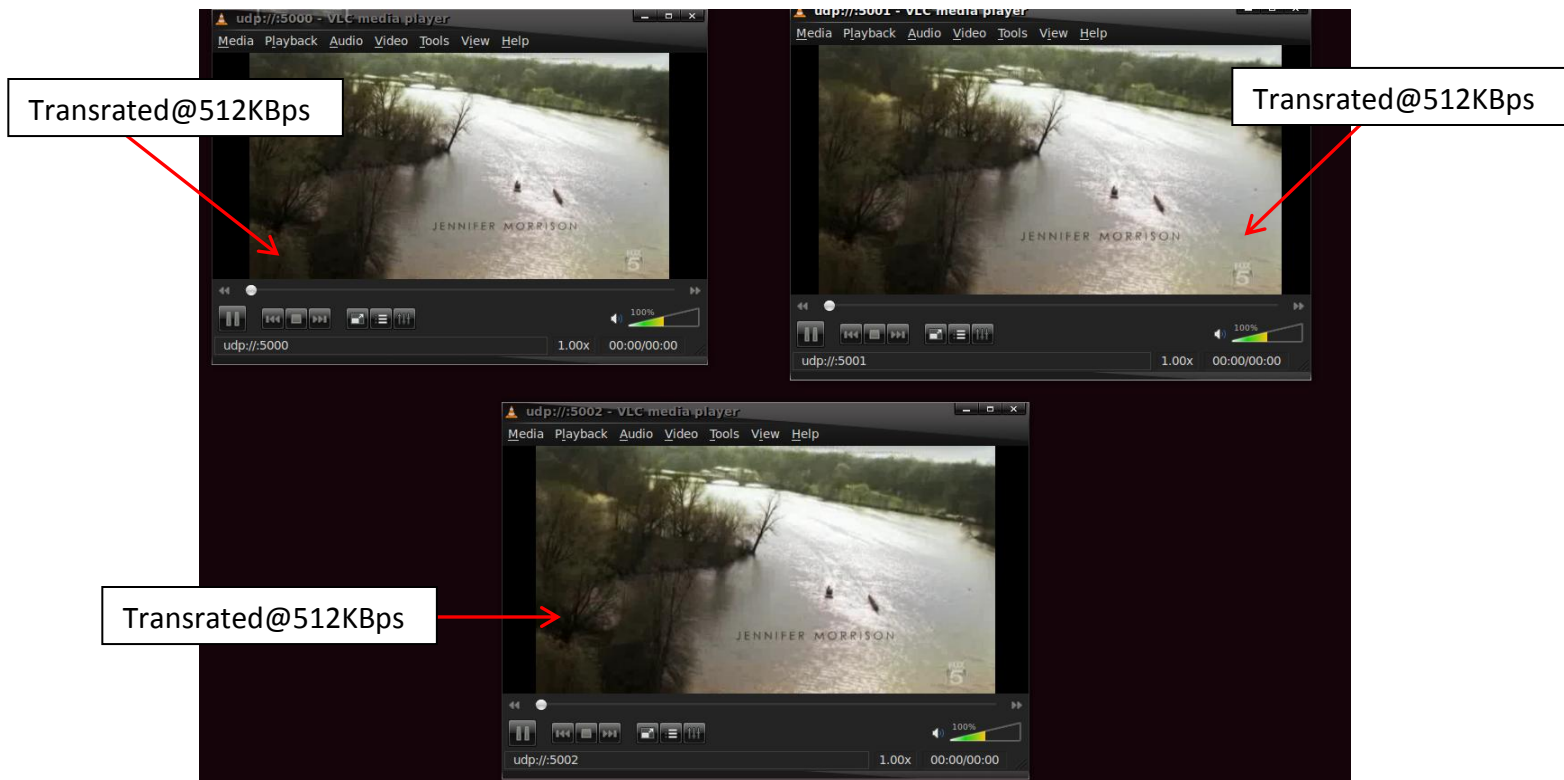
Στη μέτρηση με τρεις ροές, όσον αφορά στο μηχανισμό προσαρμογής υπηρεσιών, παρατηρείται η ίδια αποτελεσματικότητα. Μετά τον περιορισμό του εύρους ζώνης τα επίπεδα του ποσοστού απωλειών ανεβαίνουν πάνω από το 10% και ο μηχανισμός αποφασίζει προσαρμογή υπηρεσιών για όλες τις ροές. Μετά από περίπου **10 sec** το video αναπαράγεται κανονικά, με λίγη χειρότερη ποιότητα εικόνας (η κωδικοποίηση έγινε στα 512Kbit), σε όλες τις ροές (όπως φαίνεται και από τα screenshots στο Σχήμα 34)



Σχήμα 32 2^ο Σενάριο – 3^η Μέτρηση



Σχήμα 33 2^ο Σενάριο – 3^η Μέτρηση – Screenshot πριν την προσαρμογή

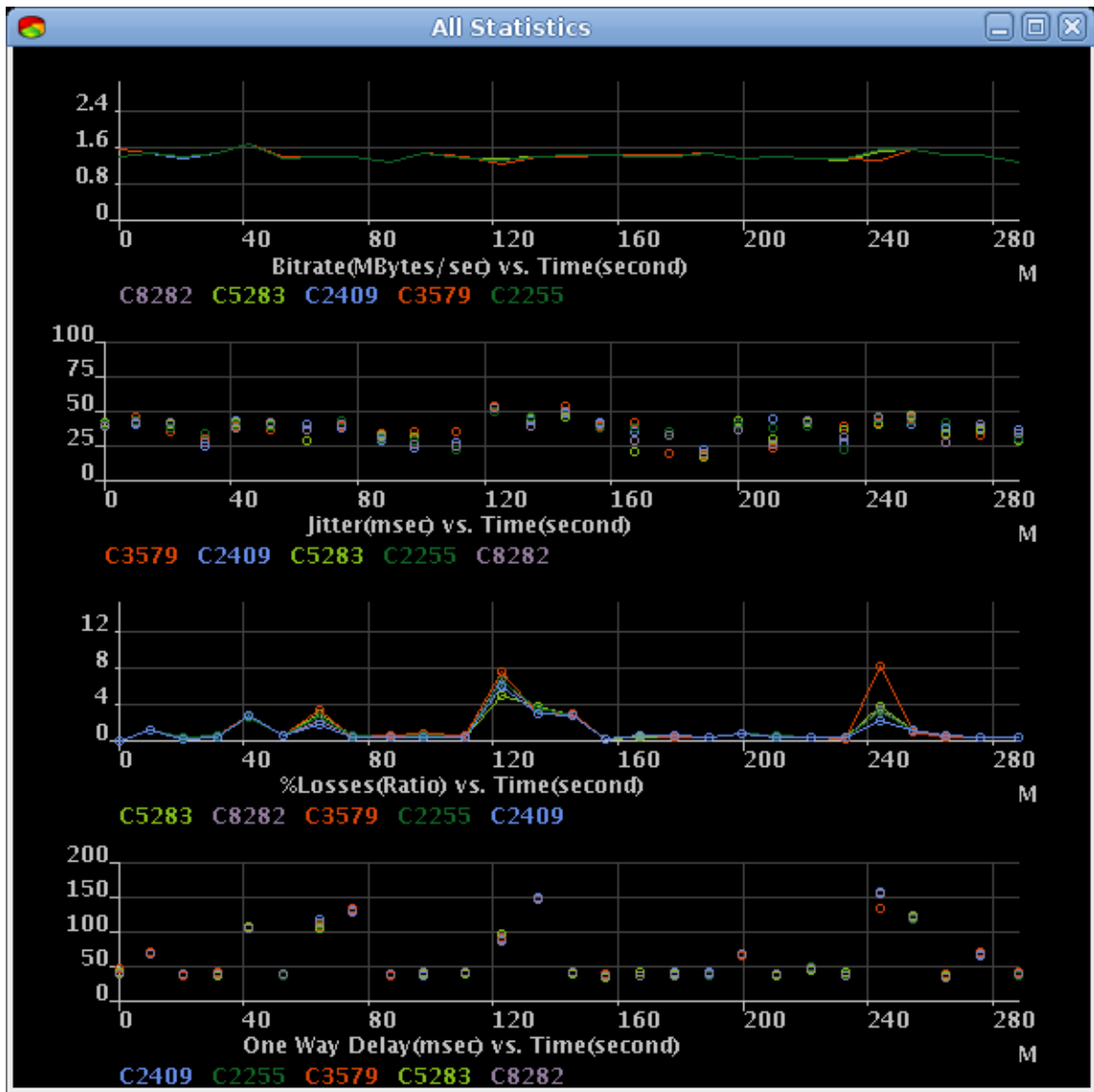


Σχήμα 34 2^ο Σενάριο – 3^η Μέτρηση – Screenshot μετά την προσαρμογή

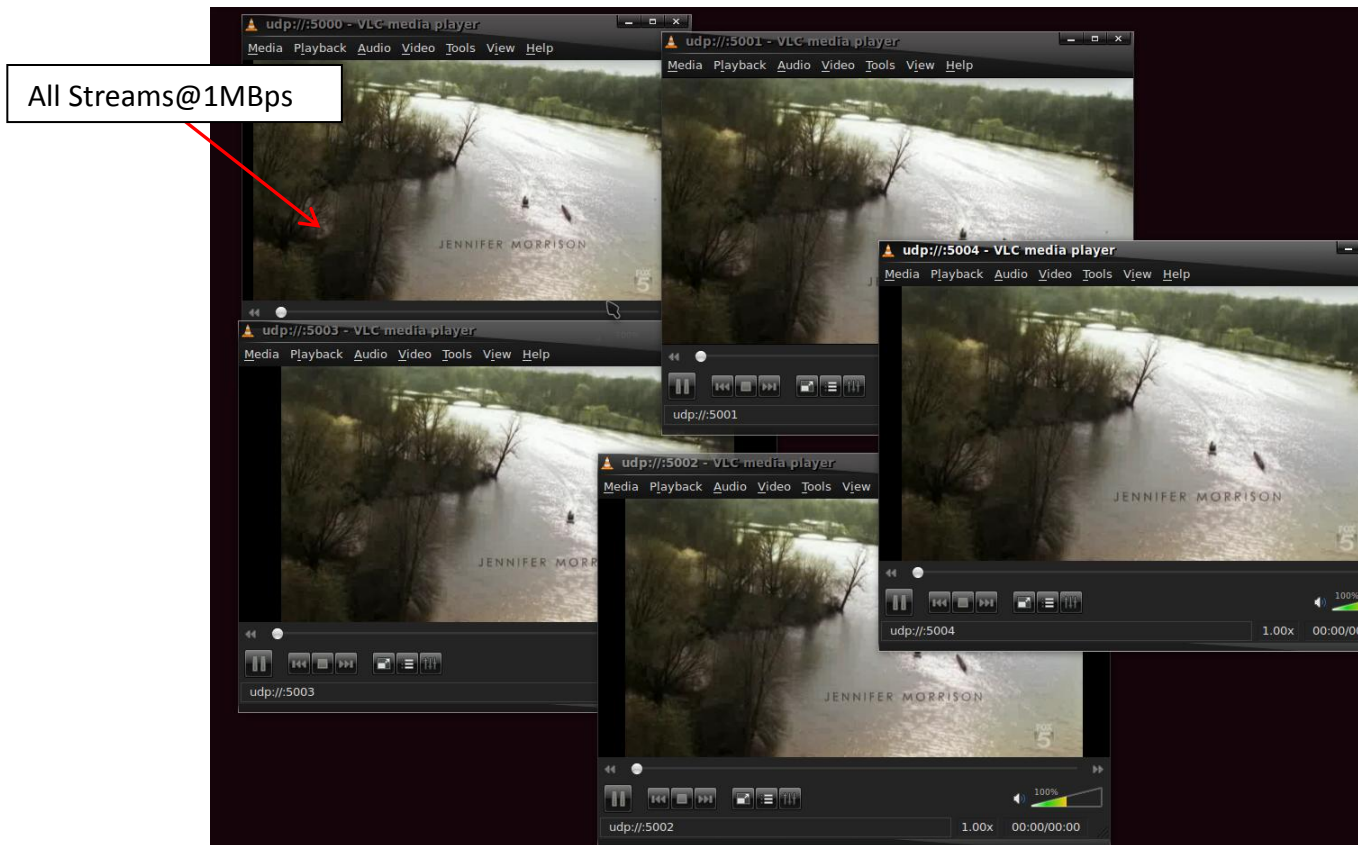
5.3. 3^ο Σενάριο

1^η Μέτρηση

Παρατηρούνται κάποιες εξάρσεις στο Loss Rate και στο One-way Delay άλλα τίποτα ικανό να αυξήσει κατά πολύ το μέσο όρο τους. Επίσης στα screenshots δεν παρατηρήθηκε καμία παραμόρφωση της εικόνας.



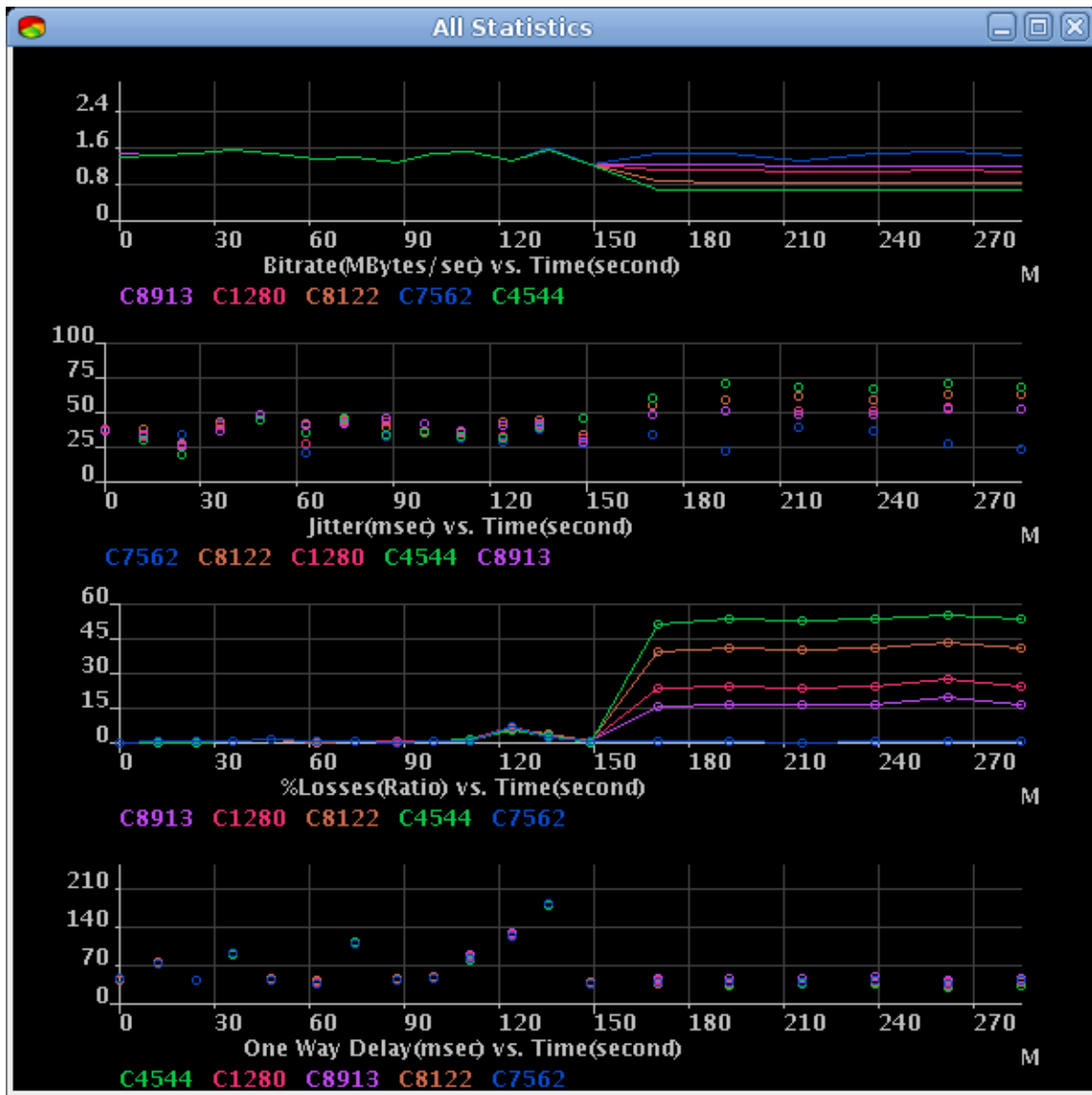
Σχήμα 35 3^ο Σενάριο – 1^η Μέτρηση



Σχήμα 36 3^ο Σενάριο – 1^η Μέτρηση - Screenshot

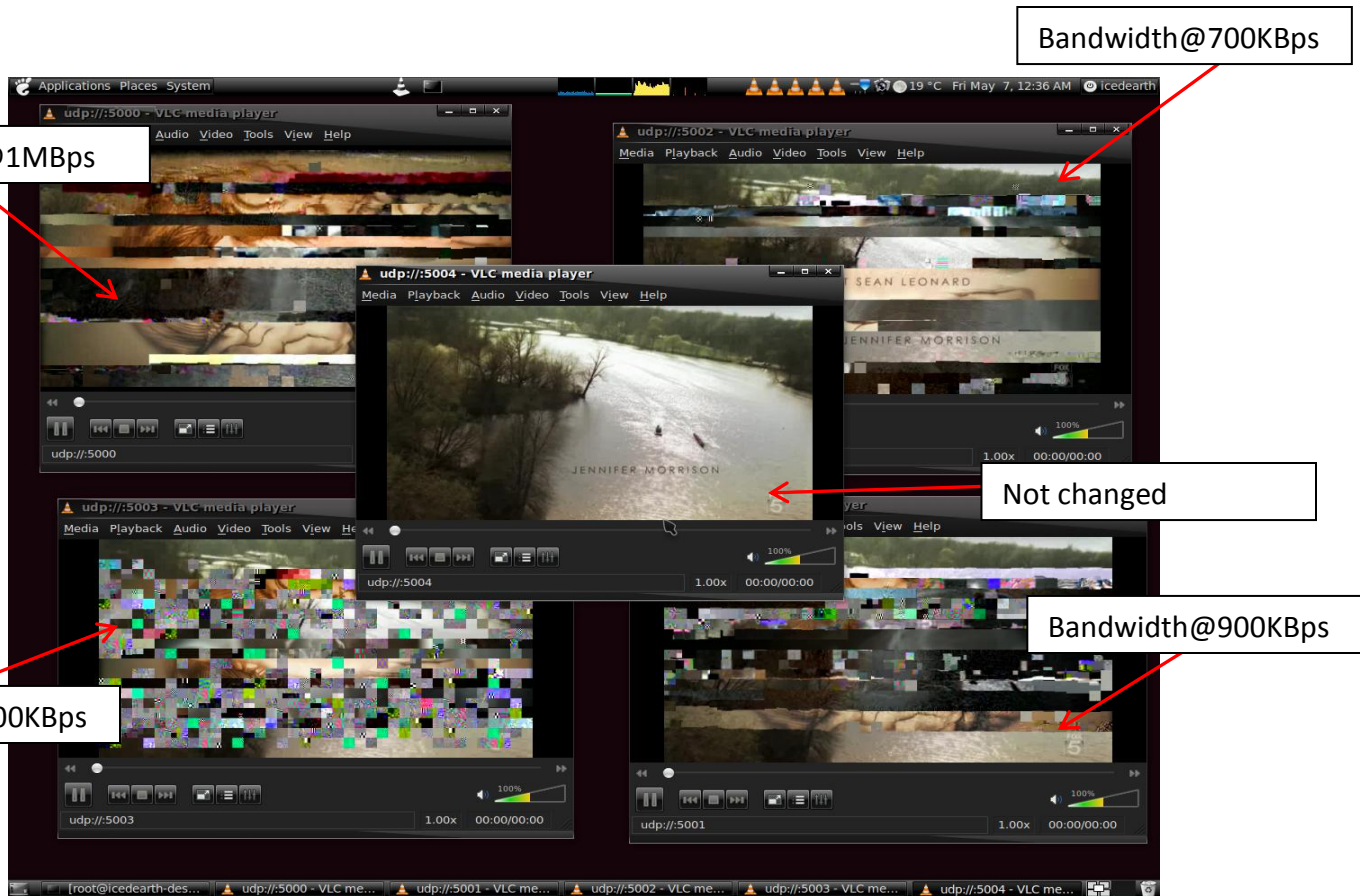
2^η Μέτρηση

Παρατηρούνται τα ίδια αποτελέσματα με την προηγούμενη μέτρηση. Ξεχωρίζει η ροή που έχει μείνει ανέπαφη (μπλε ροή - στη θύρα 5004) διότι το Loss Rate μένει σχεδόν 0 και το Jitter πολύ χαμηλό. Οι υπόλοιπες ροές έχουν αναμενόμενη συμπεριφορά.



Σχήμα 37 3^ο Σενάριο – 2^η Μέτρηση

Αρχίζει να σχηματίζεται από τις μετρήσεις μια διαχωριστική γραμμή στο γράφημα των απωλειών, καθώς παρατηρείται πως για κάθε ροή που έχει ποσοστό απωλειών πάνω από 10% προκαλείται μεγάλη μείωση στην ποιότητα της εικόνας.



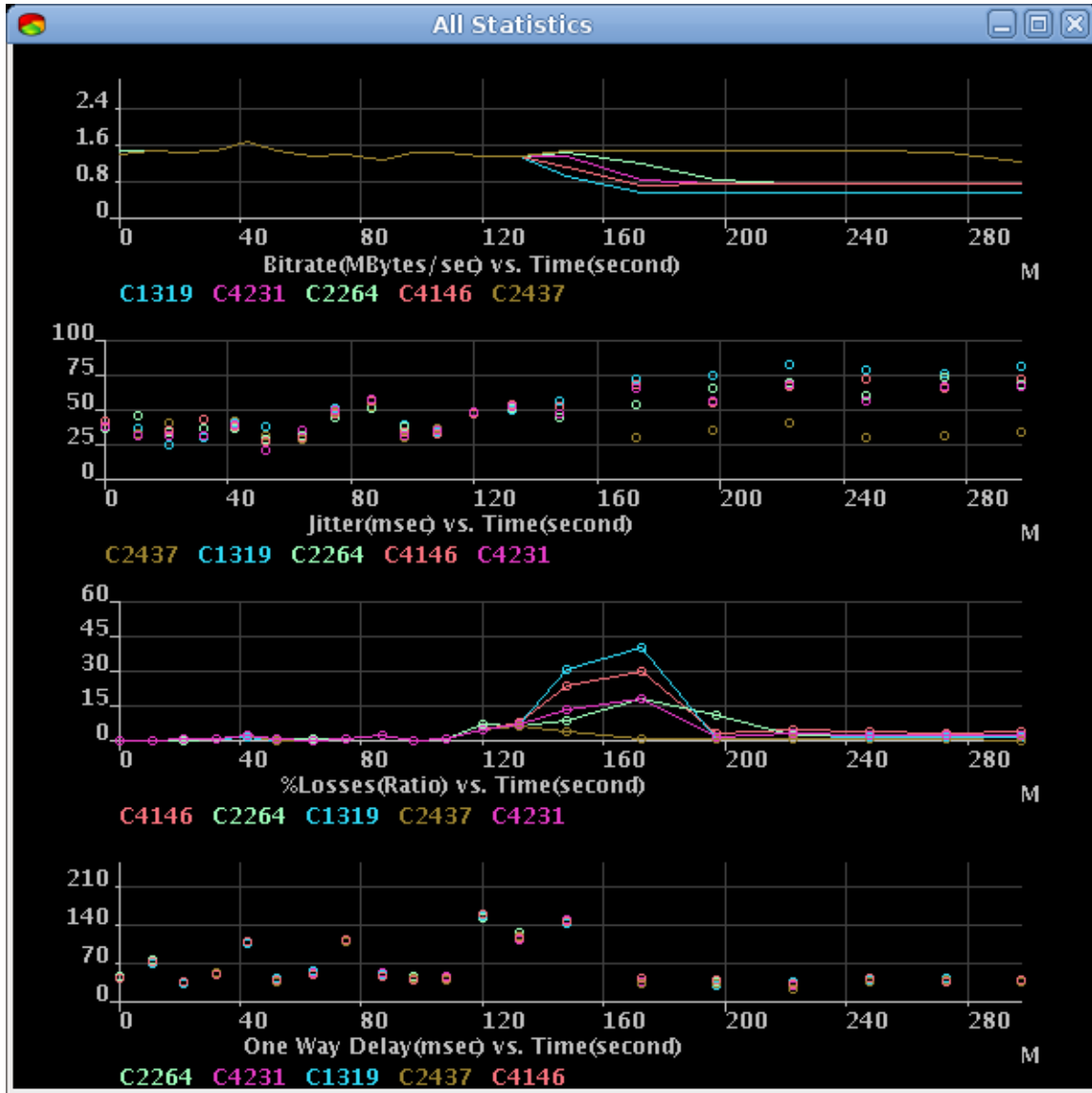
Σχήμα 38 3^ο Σενάριο – 2^η Μέτρηση - Screenshot

Στα screenshots (Σχήμα 38) είναι άξια λόγου η εμπειρική διαφορά που παρατηρείται ανάλογα με το ποσοστό απωλειών. Στη ροή στη θύρα 5004 (500Kbit εύρος ζώνης ενώ το video είναι 1Mbit) δεν μπορούν καν να διακριθούν βασικά στοιχεία της εικόνας, αντίθετα με τις άλλες ροές που οι απώλειες είναι λιγότερες.

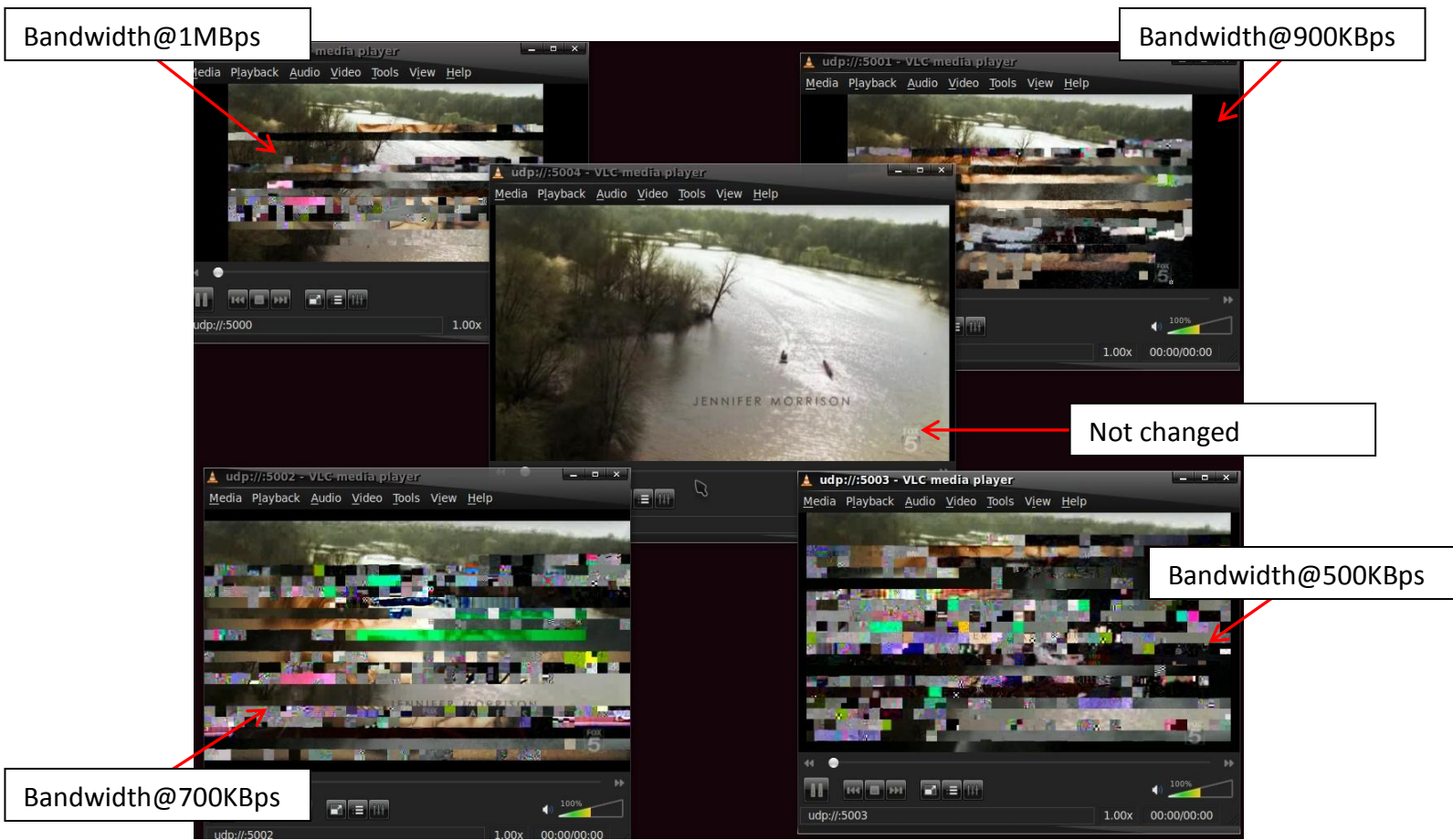
3^η Μέτρηση

Οι προηγούμενες μετρήσεις απέδειξαν την αποτελεσματικότητα του μηχανισμού αλλά αυτή η τελευταία μέτρηση αποδεικνύει τη λειτουργικότητα του και τον ανεξάρτητο τρόπο με τον οποίο αξιολογεί κάθε ροή. Σε αυτή τη μέτρηση η ροή στη 5004 μένει ανέπαφη και απο τον περιορισμό του εύρους ζώνης αλλά και απο το μηχανισμό όπως φαίνεται στο Σχήμα 39 , καθώς το ποσοστό απώλειας κυμαίνεται στο 1%. Ενδιαφέρον παρουσιάζει και η ροή στη θύρα 5003 που περιορίζεται στα 500Kbit και άρα λογικά η προσαρμογή που εφαρμοζόταν μέχρι τώρα (κωδικοποίηση του video στα 512Kbit) δεν

θα ήταν αρκετά αποτελεσματική, έτσι η ροή αυτή προσαρμόστηκε με διαφορετικό τρόπο απο τις άλλες (κωδικοποίηση στα 256Kbit). Στα screenshots (Σχήμα 41) μπορεί ο χρήστης να αναπαράγει κανονικά τη ροή με λίγο χειρότερη ποιότητα απο τις άλλες και λίγο περισσότερη καθυστέρηση (latency).



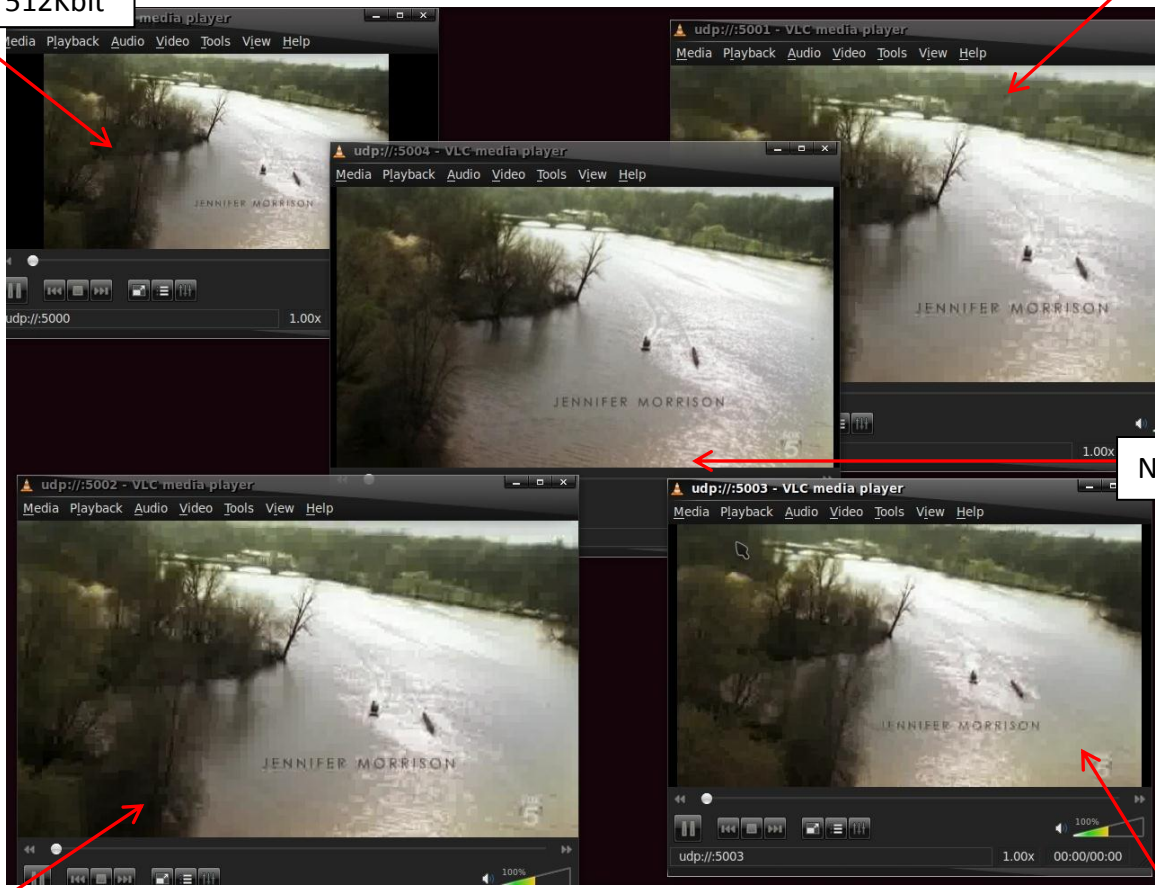
Σχήμα 39 3^ο Σενάριο – 3^η Μέτρηση



Σχήμα 40 3^ο Σενάριο – 3^η Μέτρηση – Screenshot πριν την προσαρμογή

Transrated @ 512Kbit

Transrated @ 512Kbit



No transrating

Transrated @ 512Kbit

Σχήμα 41 3^ο Σενάριο – 3^η Μέτρηση – Screenshot μετά την προσαρμογή

Transrated @ 256Kbit

6. Συμπεράσματα – Μελλοντικές Προτάσεις

Κατά την υλοποίηση της εργασίας, κατασκευάστηκε ένας μηχανισμός που έχει τη δυνατότητα να προσαρμόζει μια υπηρεσία βάσει κάποιων δοθέντων κριτηρίων στην αρχή της εκτέλεσης του.

Παρατηρήθηκε πως ο χρόνος απόκρισης του μηχανισμού ήταν περίπου στα 7-8 δευτερόλεπτα, ένα μέγεθος που κρίνεται ικανοποιητικό αν σκεφτεί κανείς το σενάριο στο σύνολο του: ένας χρήστης απολαμβάνει μια προσφερόμενη υπηρεσία και ξαφνικά για άγνωστους λόγους οι συνθήκες της μετάδοσης δυσχεραίνουν αυξάνοντας κατά πολύ το ποσοστό απωλειών (στη συγκεκριμένη περίπτωση που μελέτησε η εργασία, πάνω από 10%). Υπό κανονικές συνθήκες, ο χρήστης δεν θα είχε (και δεν είχε) τη δυνατότητα να λάβει την υπηρεσία. Με το μηχανισμό προσαρμογής υπηρεσιών που κατασκευάστηκε, ο χρήστης λαμβάνει ξανά την προαναφερθείσα υπηρεσία και την αναπαράγει στον υπολογιστή, με ικανοποιητική ποιότητα, μέσα σε ένα χρονικό περιθώριο κάτω των 10 δευτερολέπτων.

Ένας ακόμα στόχος που τέθηκε πριν την κατασκευή του μηχανισμού, ήταν η ικανότητα να επεξεργάζεται και να αξιολογεί ξεχωριστά κάθε ροή, κάτι που επίσης επιτεύχθη, όπως φαίνεται από τις μετρήσεις στο κεφάλαιο 5 (σελ 43). Ο μηχανισμός έχει τη δυνατότητα να προσαρμόζει τις ροές που κρίνει πως δεν παρέχουν μια ποιότητα υπηρεσίας και να αφήνει ανεπηρέαστες τις υπόλοιπες.

Συνολικά ο μηχανισμός κατάφερε μείωση του ποσοστού απωλειών της μετάδοσης έως και 35% (Σχήμα 39) σε κάποιες ροές και επίσης κατάφερε να παρέχει μια ποιότητα εικόνας που κρίθηκε ικανοποιητική (από τις δικτυακές μετρήσεις όπως επίσης και από αισθητικούς ελέγχους) σε κάθε περίπτωση.

Φυσικά, λόγω και του εκπαιδευτικού χαρακτήρα που διέπει αυτή την εργασία, η κατασκευή του μηχανισμού έγινε με κάποιους περιορισμούς που αν στο μέλλον αρθούν, θα διευρυνθούν οι προοπτικές ανάπτυξης του μηχανισμού, θα αυξηθεί η αποτελεσματικότητα του αλλά και η επεκτασιμότητά του, δηλαδή η δυνατότητα λειτουργίας του σε διευρυμένα διαδραστικά περιβάλλοντα. Πιο συγκεκριμένα, ο μηχανισμός αυτός αποτελεί μια συμπληρωματική λύση για την βέλτιστη κατανομή του εύρους ζώνης και εξαρτάται άμεσα από την υπολογιστική ισχύ των διαθέσιμων υπολογιστικών συστημάτων και από το πλήθος των ροών προς παρακολούθηση. Ο μηχανισμός θα ήταν εν γένει αποτελεσματικότερος αν υπήρχε η δυνατότητα να γίνεται επιλογή των χαρακτηριστικών της προσαρμογής κάτι που στην υλοποίηση που περιγράφηκε στην εργασία δεν υπάρχει. Επίσης, ένας τομέας που έχει δυνατότητες

βελτίωσης είναι τα κριτήρια για την προσαρμογή της υπηρεσίας (θα μπορούσε να προστεθεί στη λίστα των κριτηρίων το jitter) καθώς τώρα η προσαρμογή γίνεται μόνο με βάση το ποσοστό απωλειών. Τέλος, ο μηχανισμός κατασκευάστηκε με βάση ένα κεντρικοποιημένο δίκτυο, που ένας υπολογιστής «μαθαίνει» (μέσω των CMN's) τις συνθήκες του δικτύου και αποφασίζει για την προσαρμογή. Είναι προφανές πως θα γινόταν πολύ πιο ευέλικτος εάν υπήρχε η δυνατότητα να αποφασίζει ξεχωριστά ο κάθε CMN.

A. Παράρτημα

A.1 Υπολογισμός Δικτυακών Χαρακτηριστικών (Γλώσσα Java)

A.1.1 Υπολογισμός Ποσοστού Απωλειών (Loss Rate)

```
//-----losses-----  
  
    System.out.println("calculating losses");  
  
    int loss = 0;  
  
    double timeab = 0, timeSum = 0, owdelay = 0, percentage = 0;  
  
    Vector concatVector = null, concatVector2 = null;  
  
    concatVector = new Vector();  
    concatVector2 = new Vector();  
    Object obj = null;  
  
    String concat2 = null;  
    for (int i = 0; i <= vector2bounds; i++) {  
  
        tmp2 = (table) packet2outer.get(i);  
        int tempId2 = tmp2.getId();  
        long tempChksum2 = tmp2.getChksum();  
        concat2 = Integer.toString(tempId2) + "." + Long.toString(tempChksum2);  
        concatVector2.addElement(concat2);  
    }  
}
```

```

String concat1 = null;
for (int i = 0; i <= vector1bounds; i++) {

    tmp = (table) packet1outer.get(i);
    int tempId = tmp.getId();
    long tempChksum = tmp.getChksum();
    concat1 = Integer.toString(tempId) + "." + Long.toString(tempChksum);
    concatVector.addElement(concat1);
}

for (int i = 0; i <= vector2bounds; i++) {
    obj = concatVector2.get(i);
    if (!concatVector.contains(obj)) {
        //System.out.println("just found a loss!");
        loss += 1;
    }
}

percentage = ((double) loss * 100) / vector2bounds;

```

A.1.2 Υπολογισμός Ρυθμού Μετάδοσης (Bitrate)

```
//-----bitrate-----  
  
System.out.println("calculating throughput");  
  
table tmp;  
  
table tmp2;  
  
int vector1bounds = packet1outer.size() - 1;  
int vector2bounds = packet2outer.size() - 1;  
  
double payload1 = 0, payload2 = 0;  
  
double throughput1, throughput2;  
  
tmp = (table) packet1outer.get(0);  
double timestampFirst1 = tmp.getTimestamp();  
tmp = (table) packet1outer.get(vector1bounds);  
double timestampLast1 = tmp.getTimestamp();  
  
tmp2 = (table) packet2outer.get(0);  
double timestampFirst2 = tmp2.getTimestamp();  
tmp2 = (table) packet2outer.get(vector2bounds);  
double timestampLast2 = tmp2.getTimestamp();  
  
for (int i = 0; i < vector1bounds; i++) {  
    tmp = (table) packet1outer.get(i);  
    payload1 = payload1 + tmp.getPayload();  
}
```

```
}
```

```
for (int i = 0; i < vector2bounds; i++) {  
    tmp2 = (table) packet2outer.get(i);  
    payload2 = payload2 + tmp2.getPayload();  
}
```

```
throughput1 = (payload1 / ((timestampLast1 - timestampFirst1) * 100000));  
throughput2 = (payload2 / ((timestampLast2 - timestampFirst2) * 100000));
```

A.1.3 Υπολογισμός Διακύμανσης Απωλειών (Jitter)

```
//-----jitter-----  
  
    // System.out.println("calculating jitter");  
  
    table tmp2f, tmp2s, tmp1f, tmp1s;  
  
    double timeJ2second, timeJ2first, timeJ1second, timeJ1first, jitterab2 = 0, jitterab1 = 0,  
jitter, jitter2 = 0, jitter1 = 0, averageJitter = 0;  
  
    double[] jitterV2 = null, jitterV1 = null, jitterVector = null;  
  
    jitterV1 = new double[vector1bounds];  
  
    jitterV2 = new double[vector2bounds];  
  
    jitterVector = new double[vector2bounds];  
  
  
  
    for (int j = 1; j <= vector2bounds; j++) {  
  
        tmp2s = (table) packet2outer.get(j);  
  
        tmp2f = (table) packet2outer.get(j - 1);  
  
        timeJ2second = tmp2s.getTimestamp();  
  
        timeJ2first = tmp2f.getTimestamp();  
  
        jitterab2 = timeJ2second - timeJ2first;  
  
        jitterV2[j - 1] = (jitterab2);  
  
    }  
  
  
  
    for (int i = 1; i <= vector1bounds; i++) {  
  
        tmp1s = (table) packet1outer.get(i);  
  
        tmp1f = (table) packet1outer.get(i - 1);  
  
        timeJ1second = tmp1s.getTimestamp();
```



```

timeJ1first = tmp1f.getTimestamp();
jitterab1 = timeJ1second - timeJ1first;
jitterV1[i - 1] = jitterab1;
}

for (int k = 0; k < jitterV2.length; k++) {
    jitter2 = jitterV2[k];
    jitter1 = jitterV1[k];
    jitter = Math.abs(jitter2 - jitter1);
    jitterVector[k] = jitter;
}

for (int l = 0; l < jitterVector.length; l++) {
    averageJitter = averageJitter + jitterVector[l];
}
averageJitter = averageJitter / jitterVector.length;
averageJitter = averageJitter * 1000;
System.out.println("average Jitter = " + averageJitter);

```

A.1.4 Υπολογισμός Μονόδρομης Καθυστέρησης (One-Way Delay)

```
//-----one-way delay-----  
  
System.out.println("calculating average one-way delay ");  
  
for (int j = 0; j <= vector2bounds; j++) {  
    tmp2 = (table) packet2outer.get(j);  
    for (int k = 0; k <= vector1bounds; k++) {  
        tmp = (table) packet1outer.get(k);  
        if ((tmp.getId() == tmp2.getId()) && (tmp.getChksum() == tmp2.getChksum()))  
        {  
            timeab = tmp2.getTimestamp() - tmp.getTimestamp();  
            timeSum = timeSum + timeab;  
            timeab = 0;  
        }  
    }  
}  
  
owdelay = (double) Math.abs(timeSum / vector2bounds);  
  
owdelay = owdelay * 1000;
```

A.2 Script Περιορισμού Εύρους Ζώνης

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

```
IFACE=$1
```

```
tc qdisc del dev $IFACE root
```

```
tc qdisc add dev $IFACE handle 1: root htb
```

```
tc class add dev $IFACE parent 1: classid 1:1 htb rate 100Mbps
```

```
tc class add dev $IFACE parent 1:1 classid 1:11 htb rate 100Mbps
```

```
tc class add dev $IFACE parent 1:1 classid 1:12 htb rate 100Mbps
```

```
tc class add dev $IFACE parent 1:1 classid 1:13 htb rate 100Mbps
```

```
tc class add dev $IFACE parent 1:1 classid 1:14 htb rate 100Mbps
```

```
tc qdisc add dev $IFACE parent 1:11 handle 10: tbf rate 1000kbit buffer 1600 limit 3000
```

```
tc qdisc add dev $IFACE parent 1:12 handle 20: tbf rate 900kbit buffer 1600 limit 3000
```

```
tc qdisc add dev $IFACE parent 1:13 handle 30: tbf rate 700kbit buffer 1600 limit 3000
```

```
tc qdisc add dev $IFACE parent 1:14 handle 40: tbf rate 550kbit buffer 1600 limit 3000
```

```
tc filter add dev $IFACE protocol ip prio 1 u32 match ip dport 5000 0xffff flowid 1:11
```

```
tc filter add dev $IFACE protocol ip prio 1 u32 match ip dport 5001 0xffff flowid 1:12
```

```
tc filter add dev $IFACE protocol ip prio 1 u32 match ip dport 5002 0xffff flowid 1:13
```

```
tc filter add dev $IFACE protocol ip prio 1 u32 match ip dport 5003 0xffff flowid 1:14
```

A.3 Δρομολόγηση και Διαμόρφωση Πακέτων (Γλώσσα C)

```
h = nfq_open();
if (!h) {
    fprintf(stderr, "error during nfq_open()\n");
    exit(1);
}

printf("unbinding existing nf_queue handler for AF_INET (if any)\n");
if (nfq_unbind_pf(h, AF_INET) < 0) {
    fprintf(stderr, "error during nfq_unbind_pf()\n");
    exit(1);
}

printf("binding nfnetlink_queue as nf_queue handler for AF_INET\n");
if (nfq_bind_pf(h, AF_INET) < 0) {
    fprintf(stderr, "error during nfq_bind_pf()\n");
    exit(1);
}

printf("binding this socket to queue '0'\n");
qh = nfq_create_queue(h, 0, &cb, NULL);
if (!qh) {
    fprintf(stderr, "error during nfq_create_queue()\n");
```

```

    exit(1);
}

printf("setting copy_packet mode\n");
if (nfq_set_mode(qh, NFQNL_COPY_PACKET, 0xffff) < 0) {
    fprintf(stderr, "can't set packet_copy mode\n");
    exit(1);
}

fd = nfq_fd(h);

printf("hi\n");

system("iptables -D OUTPUT -o ra0 -p udp --dport 5000 -j NFQUEUE --queue-num
0");

system("iptables -t mangle -D PREROUTING -i eth0 -p udp --dport 5000 -j
NFQUEUE --queue-num 0");

system("iptables -A OUTPUT -o ra0 -p udp --dport 5000 -j NFQUEUE --queue-num
0");

system("iptables -t mangle -A PREROUTING -i eth0 -p udp --dport 5000 -j
NFQUEUE --queue-num 0");

while ((rv = recv(fd, buf, sizeof (buf), 0)) && rv >= 0) {
    // printf("pkt received\n");
    nfq_handle_packet(h, buf, rv);
}

printf("unbinding from queue 0\n");

```

```

nfq_destroy_queue(qh);
nfq_close(h);
return (EXIT_SUCCESS);
}

int cb(struct nfq_q_handle *qh, struct nfgenmsg *nfmsg, struct nfq_data *nfa, void *data) {

    int payload_length = 0, id = 0;
    char *payload;
    struct nfqnl_msg_packet_hdr *ph;

    // printf("entering callback\n");

    ph = ph = nfq_get_msg_packet_hdr(nfa);
    id = ntohs(ph->packet_id);

    // printf ("%d %d %d\n",NF_INET_FORWARD,NF_INET_LOCAL_OUT, ph-
    >hook);

    payload_length = nfq_get_payload(nfa, &payload);
    modify_packets(payload, payload_length, ph->hook);
    return nfq_set_verdict(qh, id, NF_ACCEPT, payload_length, payload);
}

```

A.4 Κωδικοποίηση ροής Video (εκτέλεση απο BASH)

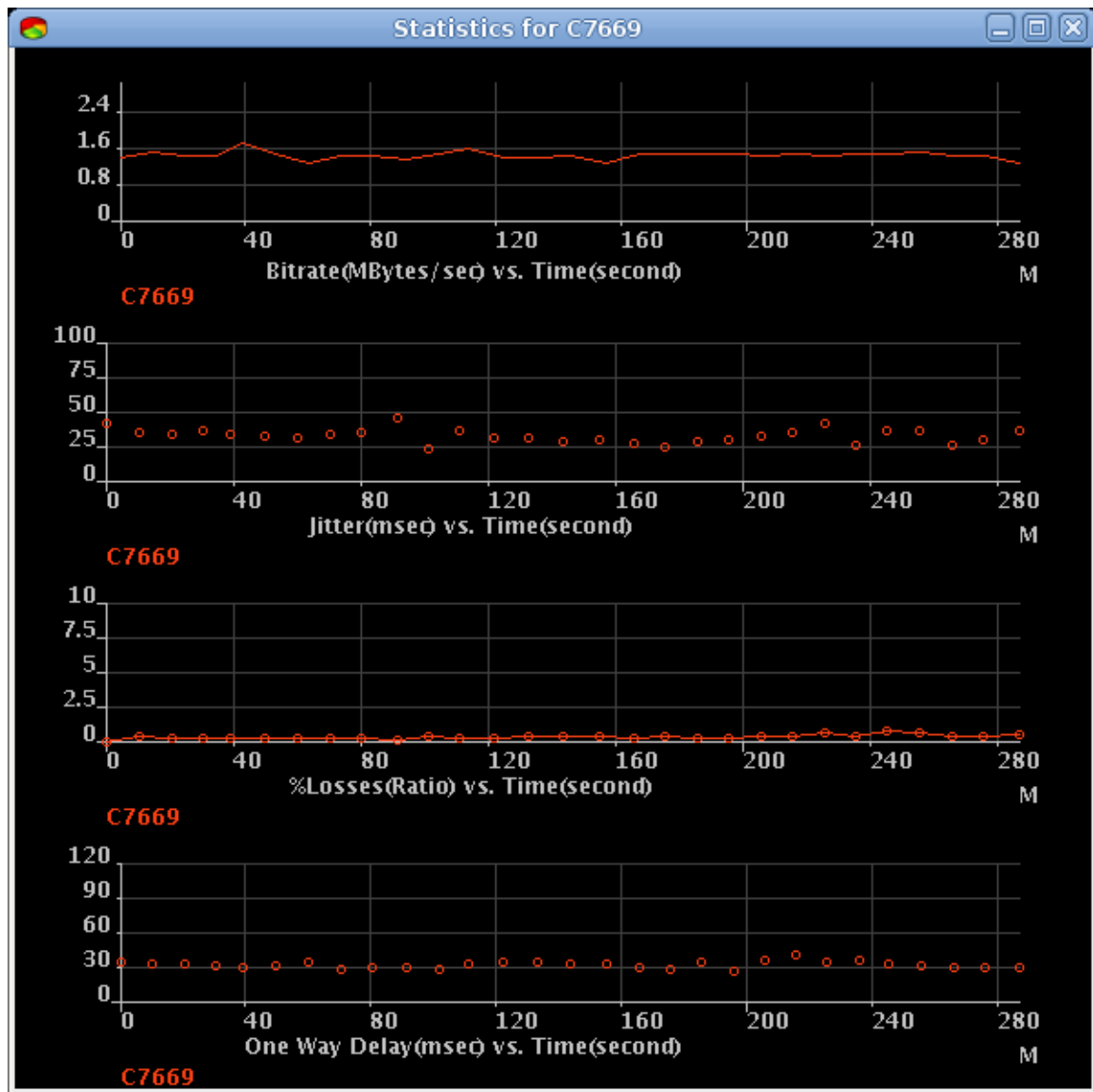
```
#transcoding udp input at port 5000

vlc -vvv -I dummy udp://@:5000
:sout='#transcode{vcodec=mp2v,vb=512,scale=1,acodec=mpga,ab=32,channels=1,deinterl
ace}:duplicate{dst=std{access=udp,mux=ts,dst=192.168.0.5:5000},
dst=std{access=udp,mux=ts,dst=192.168.0.5:5001},
dst=std{access=udp,mux=ts,dst=192.168.0.5:5002},
dst=std{access=udp,mux=ts,dst=192.168.0.5:5003},
dst=std{access=udp,mux=ts,dst=192.168.0.5:5004}}'
```

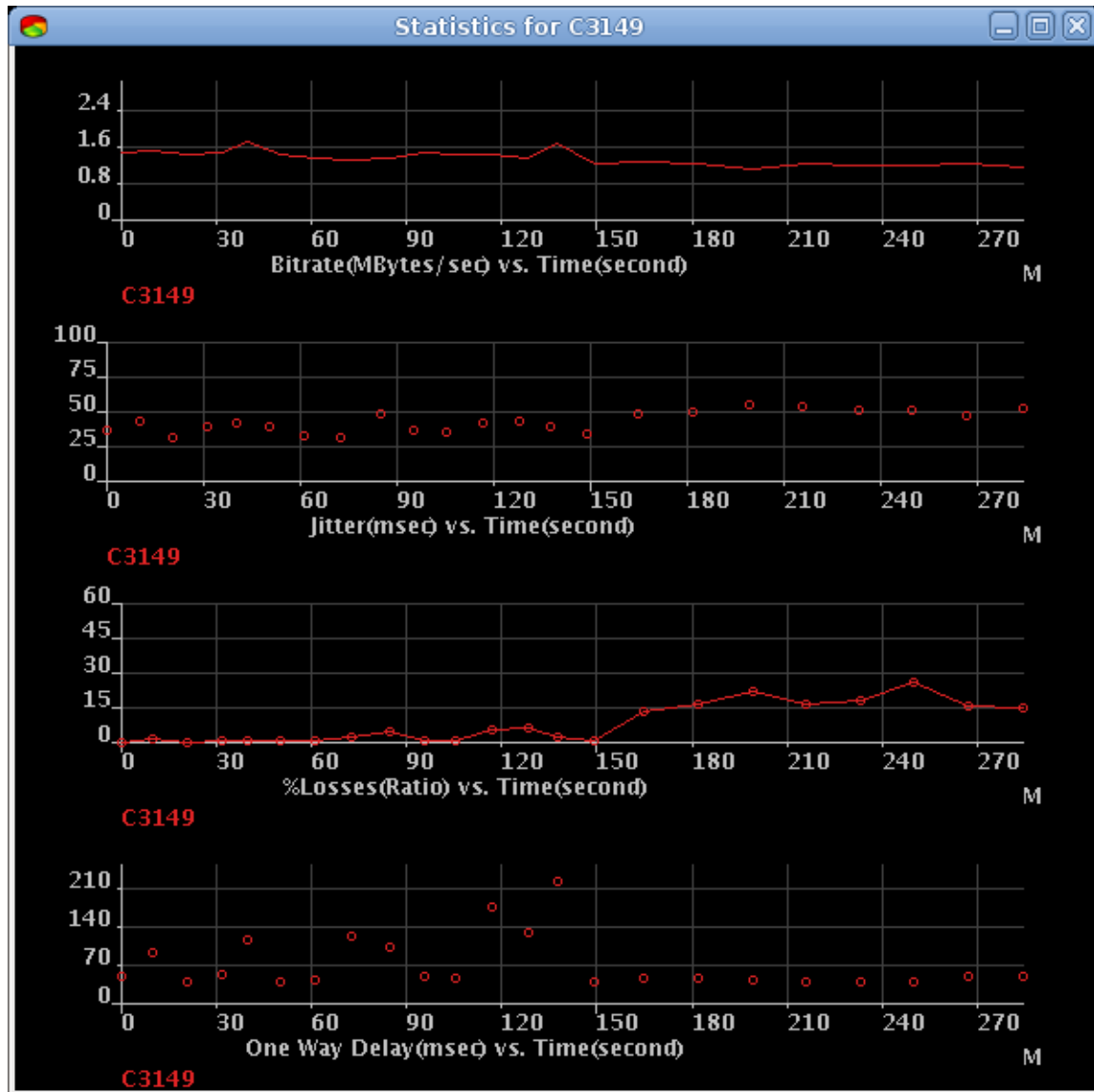
A.5 Γραφήματα Μετρήσεων (ανα χρήστη)

A.5.1. Ροή στη θύρα 5000

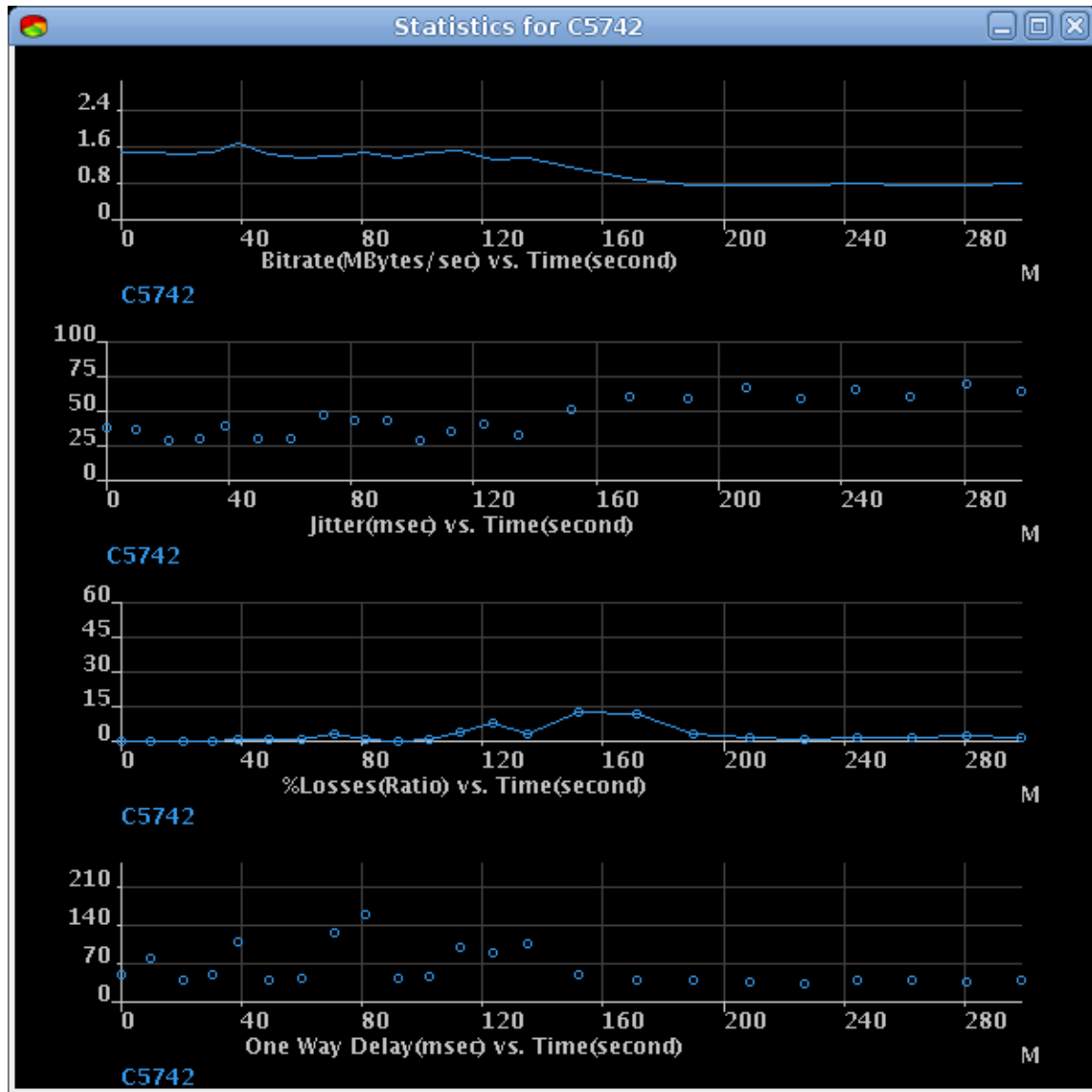
3 τελικοί αποδέκτες – Μέτρηση υπο κανονικές συνθήκες



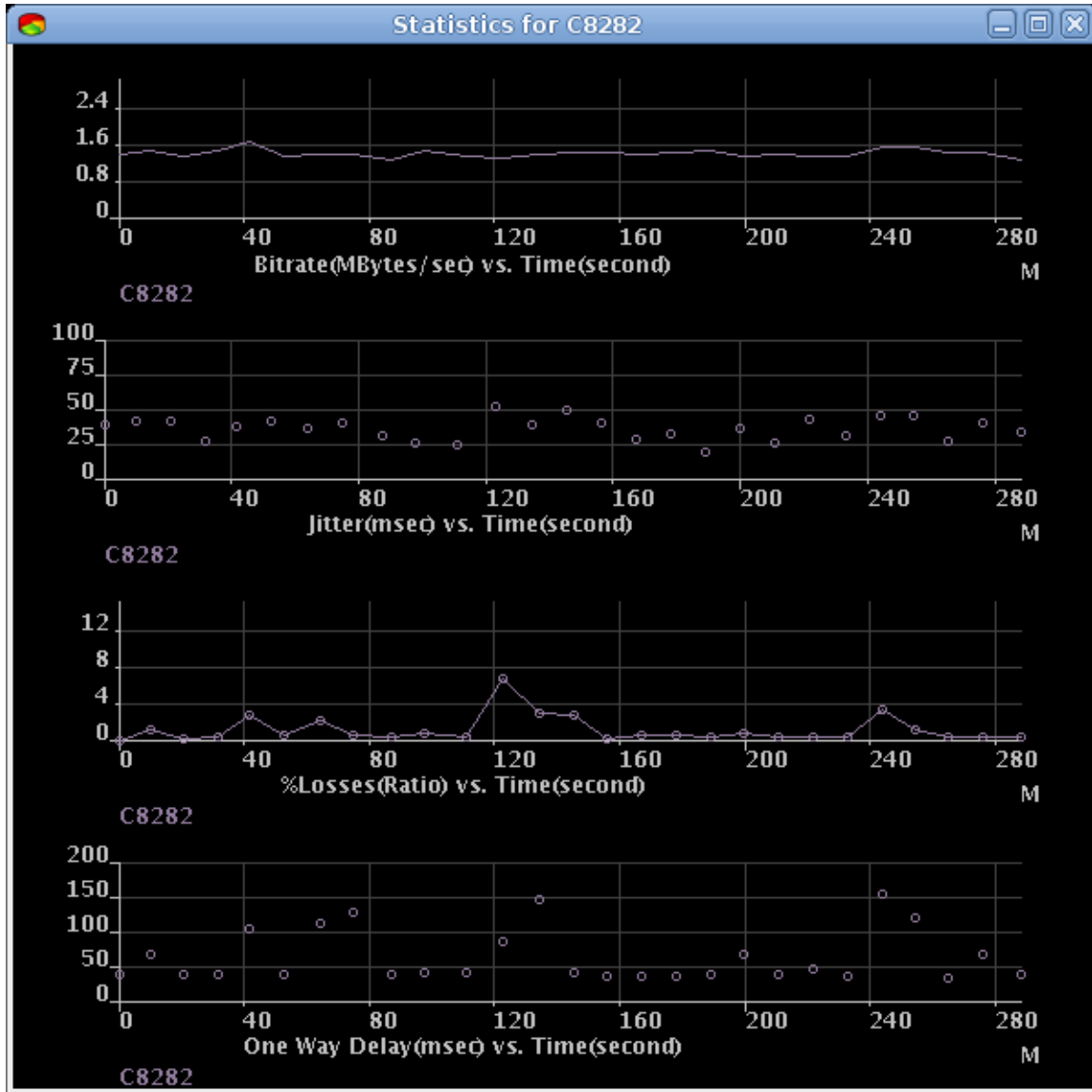
3 τελικοί αποδέκτες – Μέτρηση υπο δυσμενείς συνθήκες



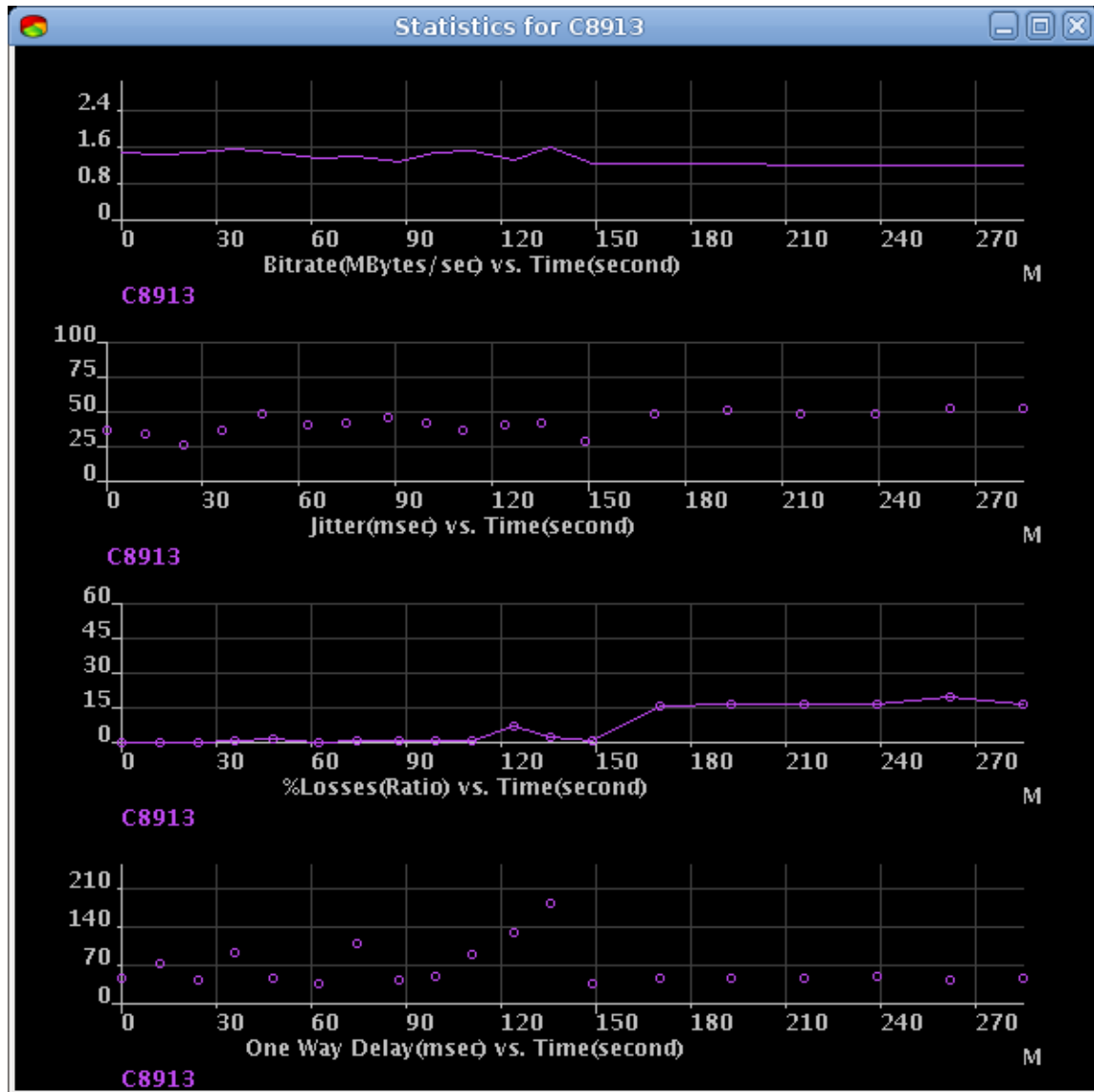
3 τελικοί αποδέκτες – Μέτρηση με το μηχανισμό σε λειτουργία



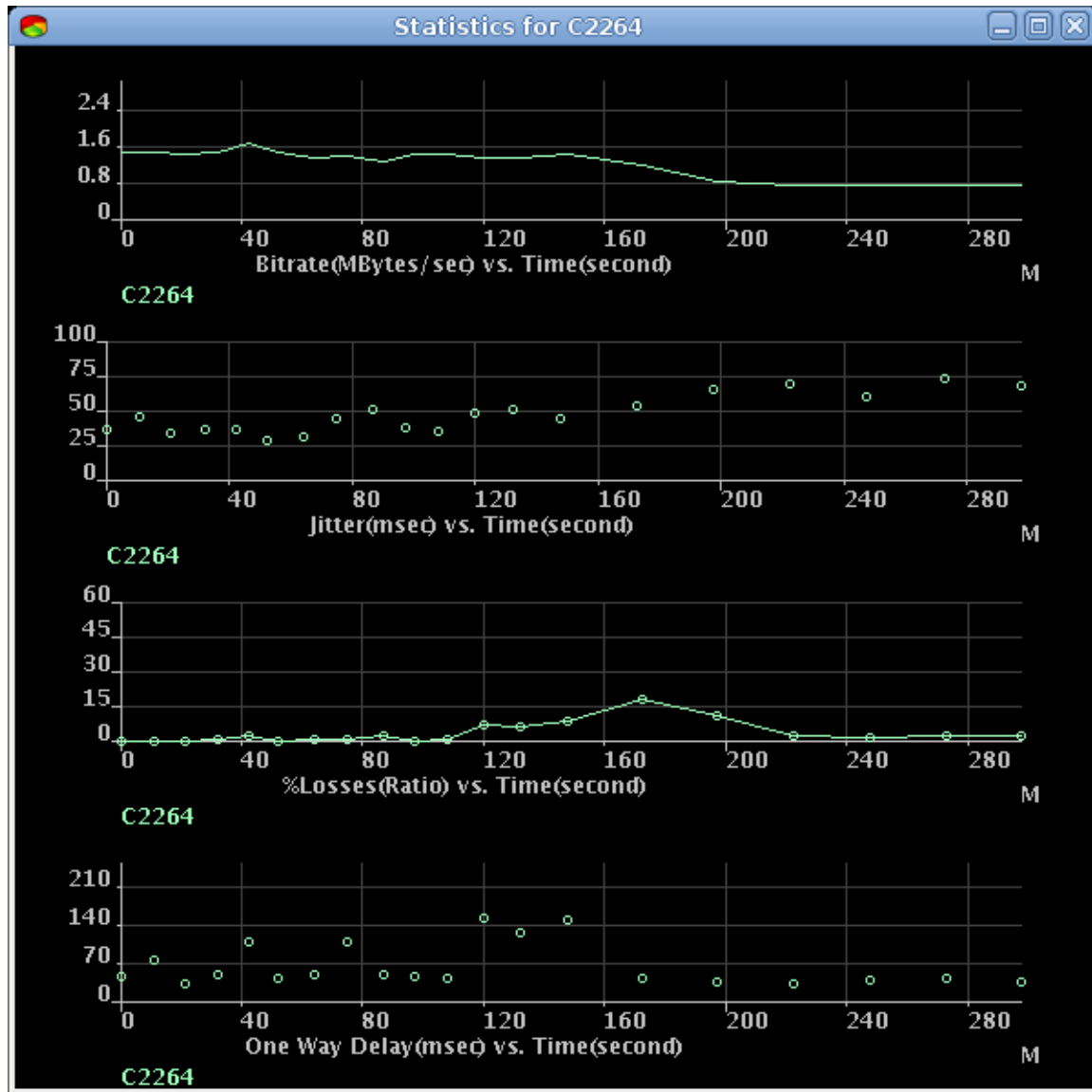
5 τελικοί αποδέκτες – Μέτρηση υπο κανονικές συνθήκες



5 τελικοί αποδέκτες – Μέτρηση υπο δυσμενείς συνθήκες

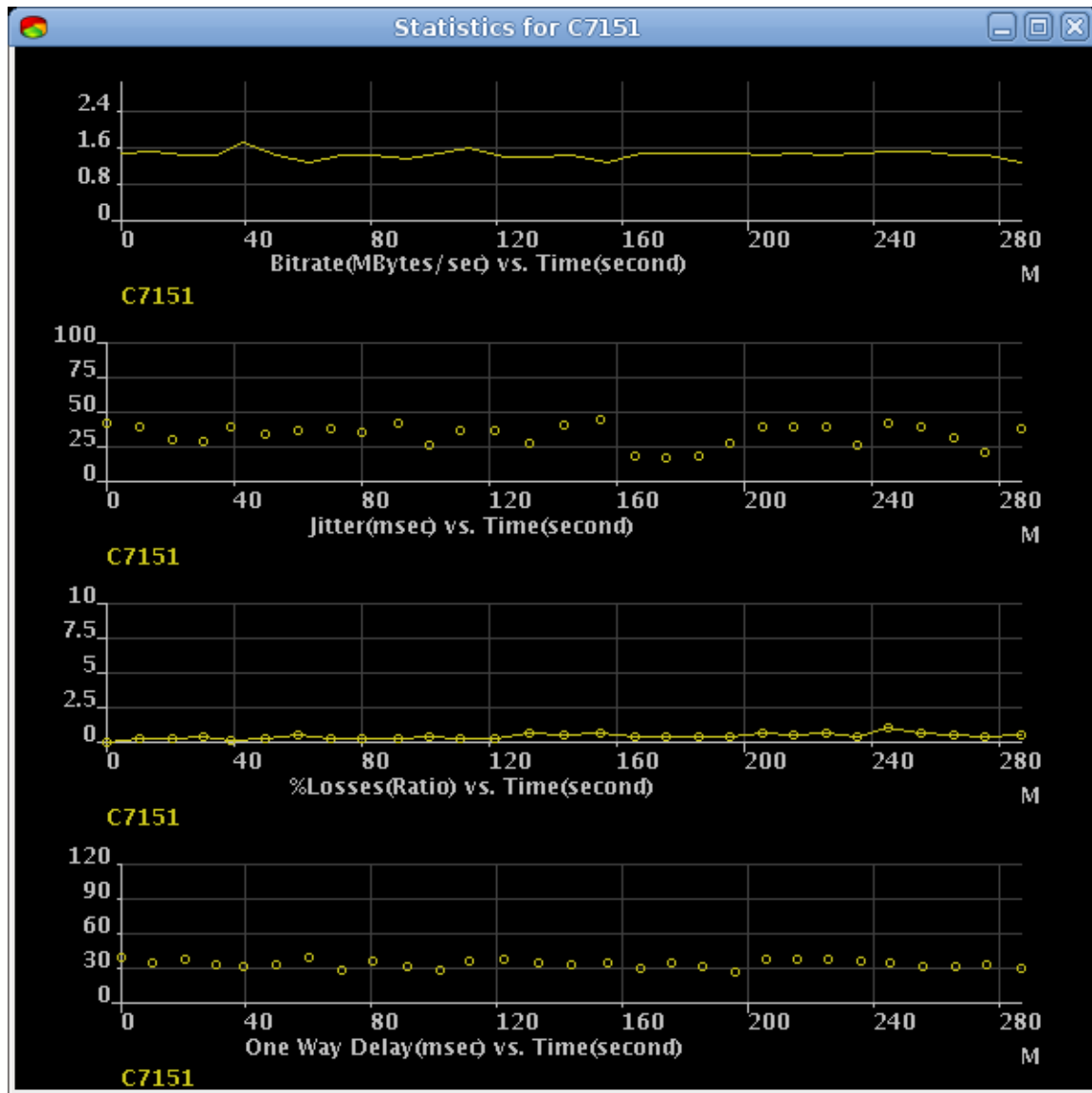


5 τελικοί αποδέκτες – Μέτρηση με το μηχανισμό σε λειτουργία

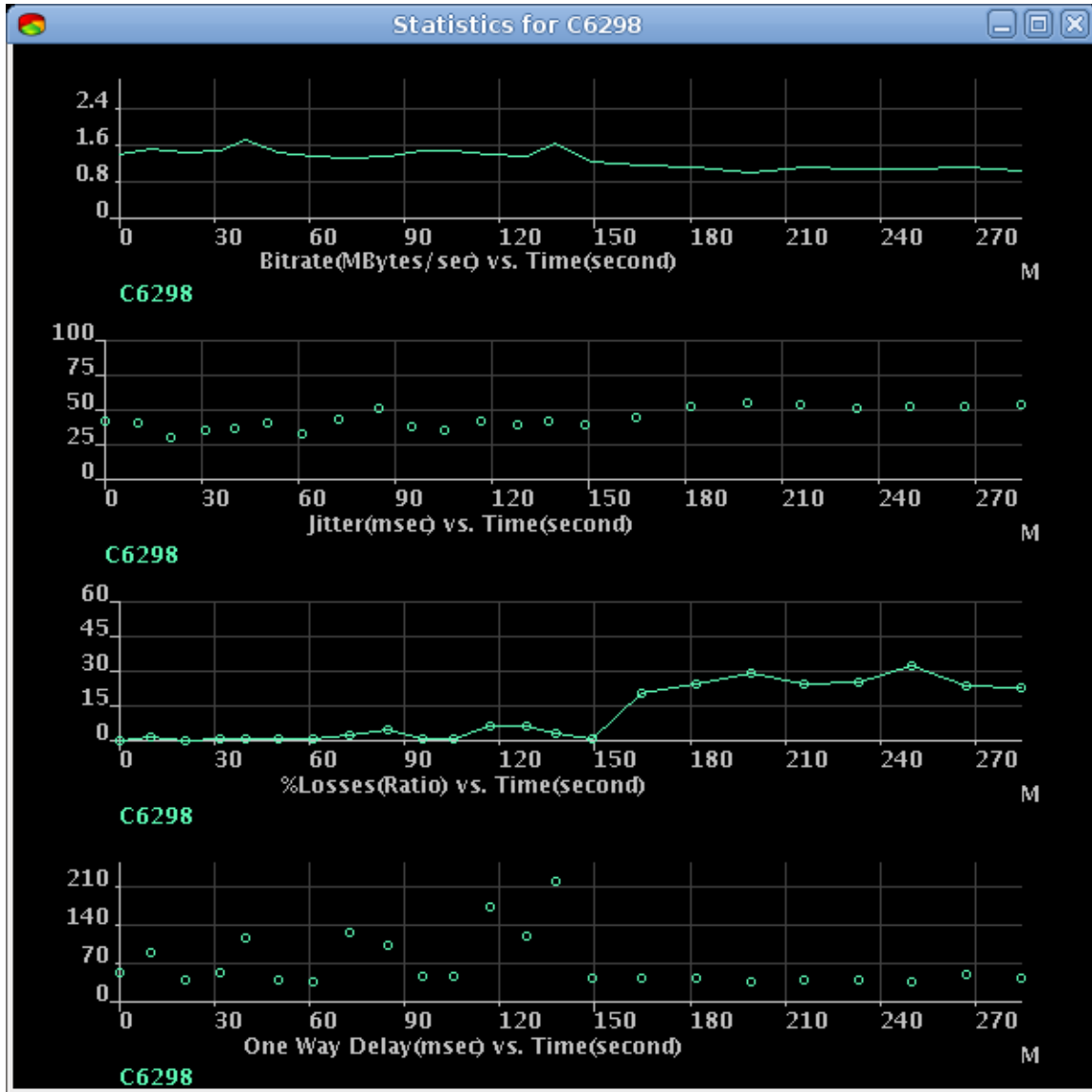


A.5.2. Ροή στη θύρα 5001

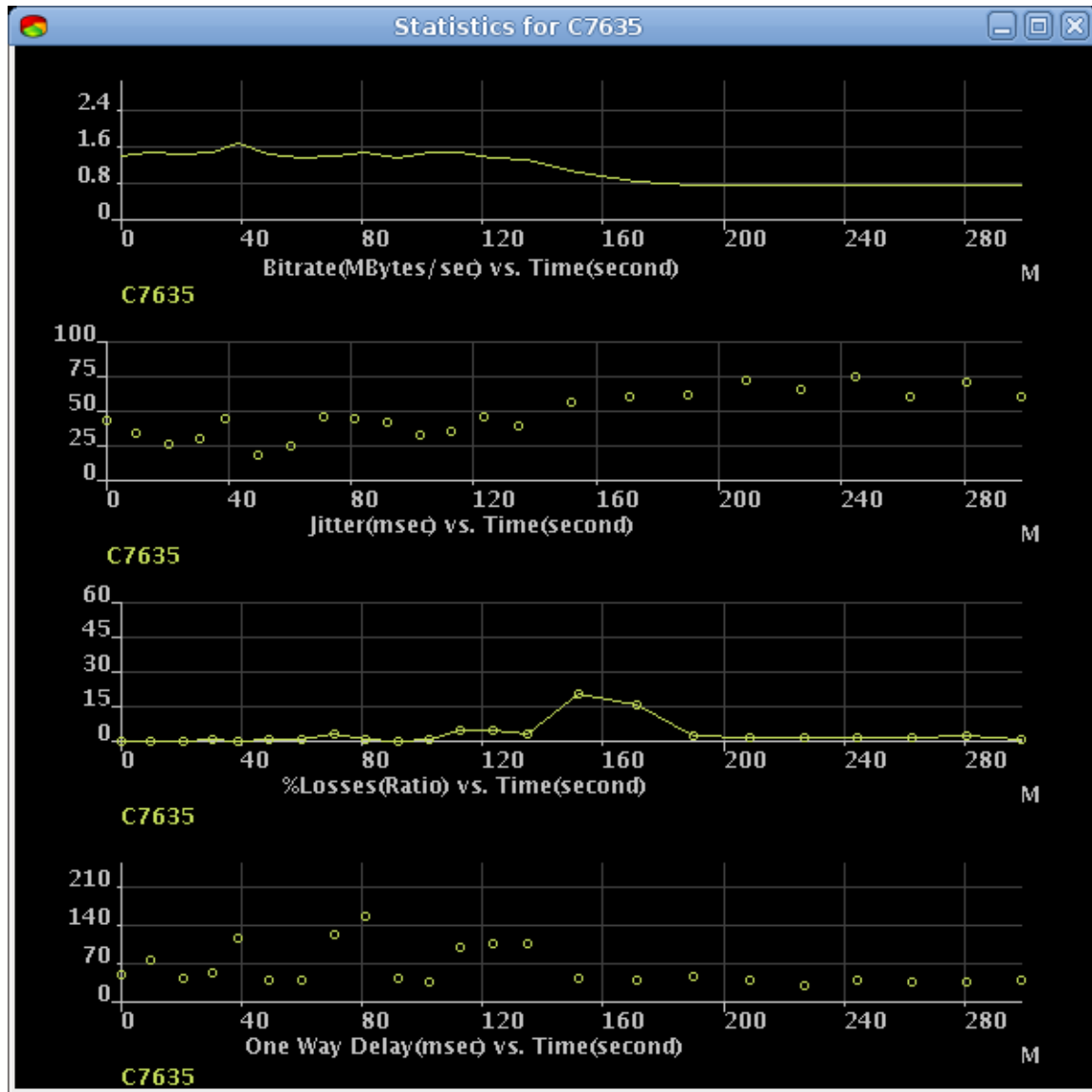
3 τελικοί αποδέκτες – Μέτρηση υπο κανονικές συνθήκες



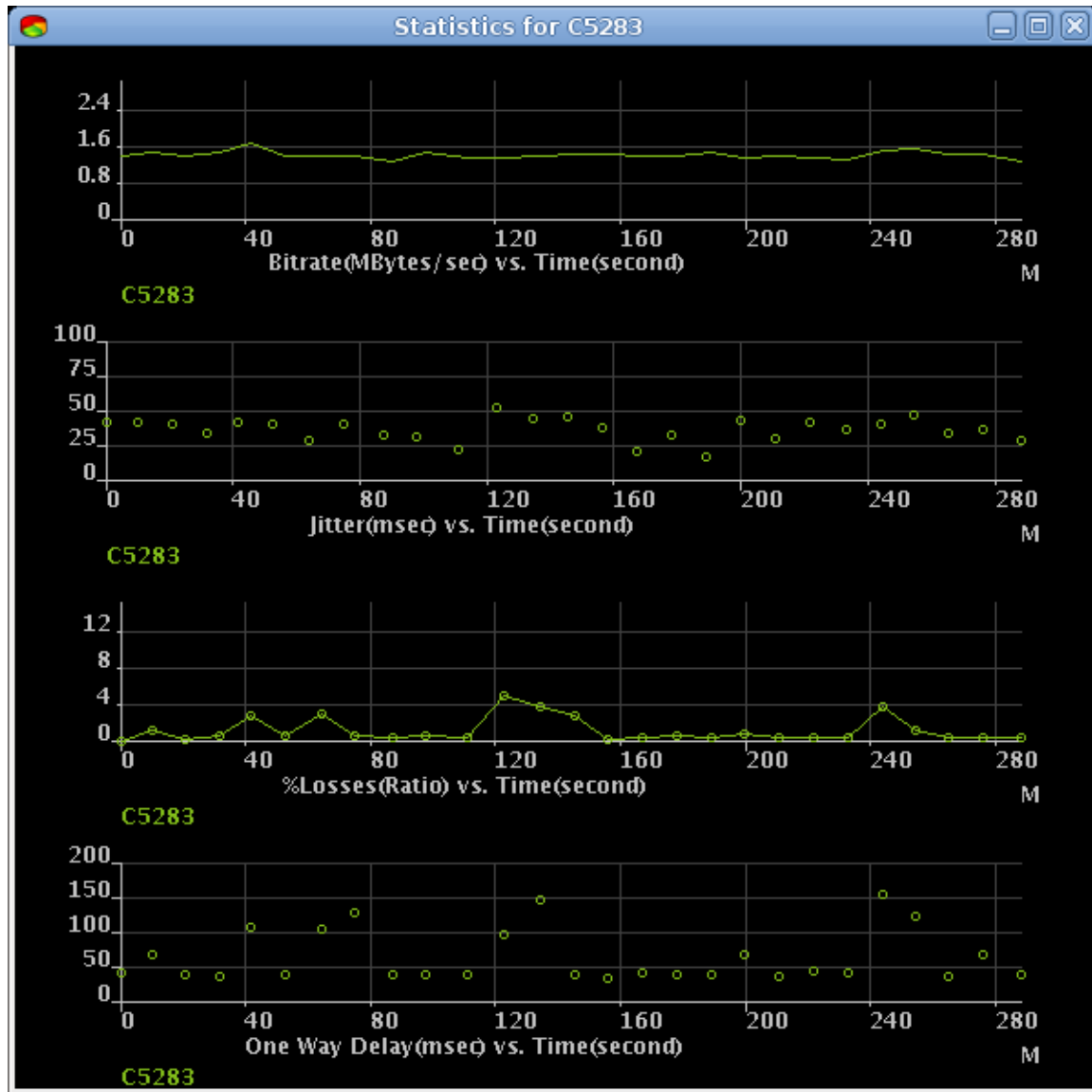
3 τελικοί αποδέκτες – Μέτρηση υπο δυσμενείς συνθήκες



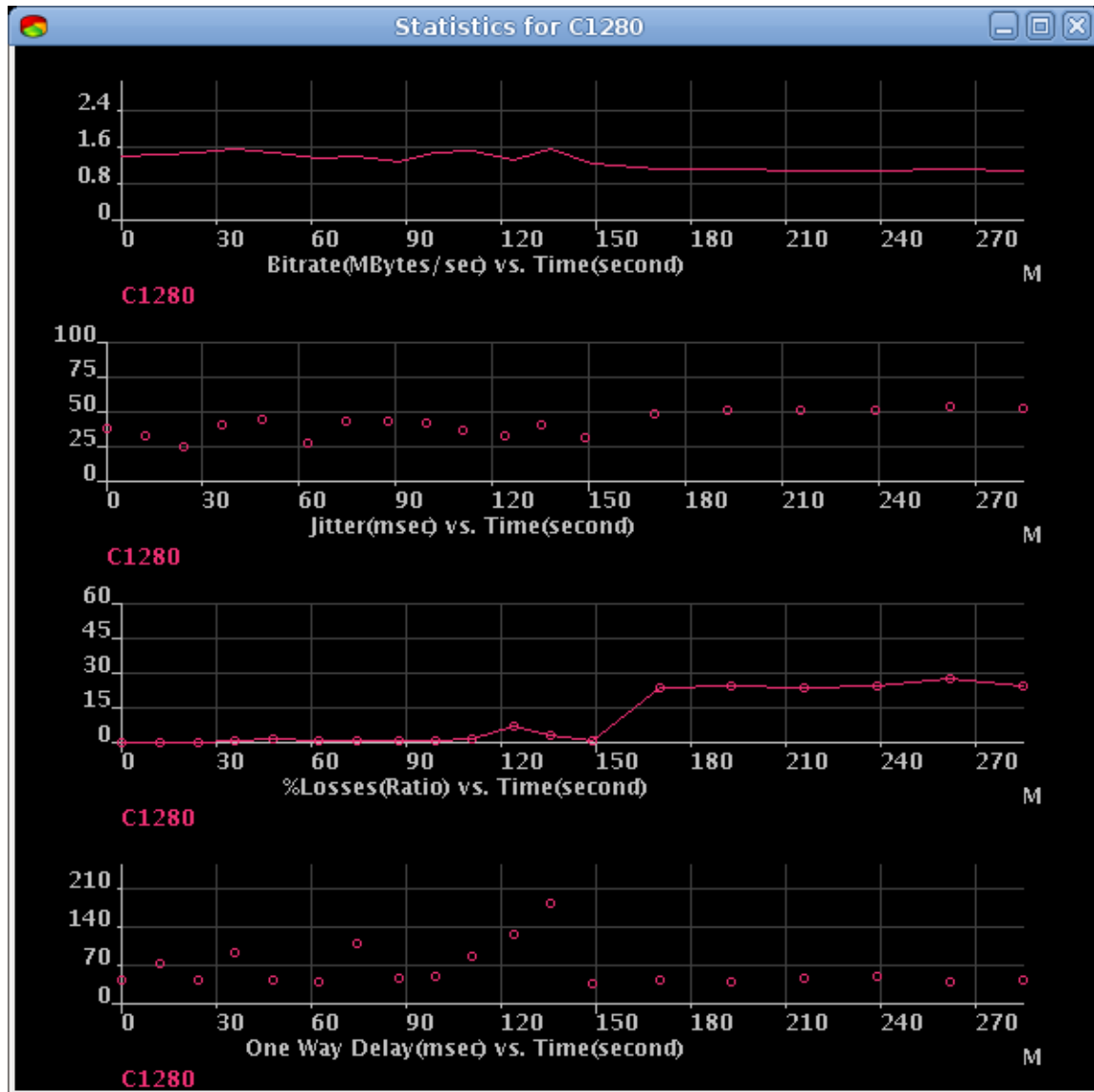
3 τελικοί αποδέκτες – Μέτρηση με το μηχανισμό σε λειτουργία



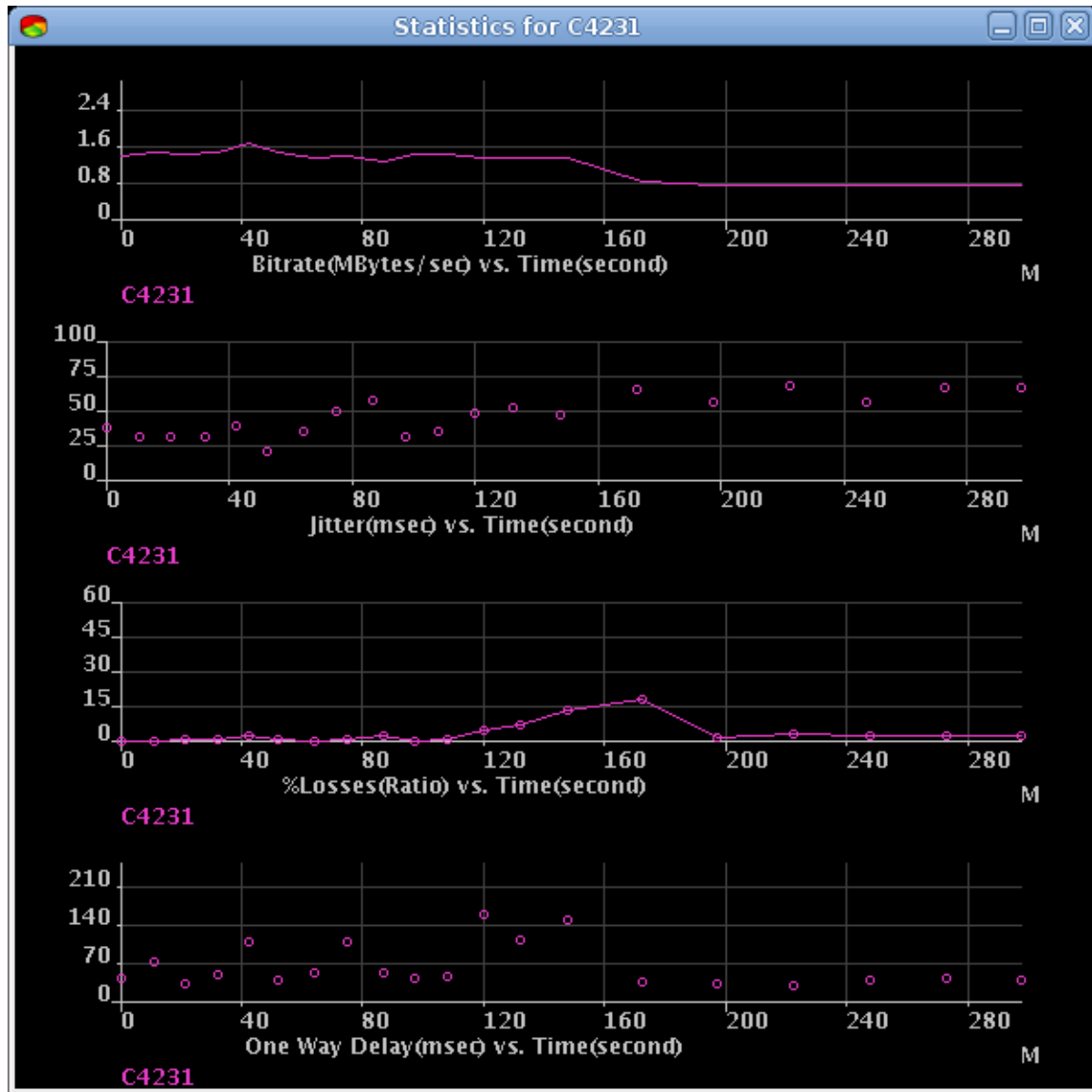
5 τελικοί αποδέκτες – Μέτρηση υπο κανονικές συνθήκες



5 τελικοί αποδέκτες – Μέτρηση υπο δυσμενείς συνθήκες

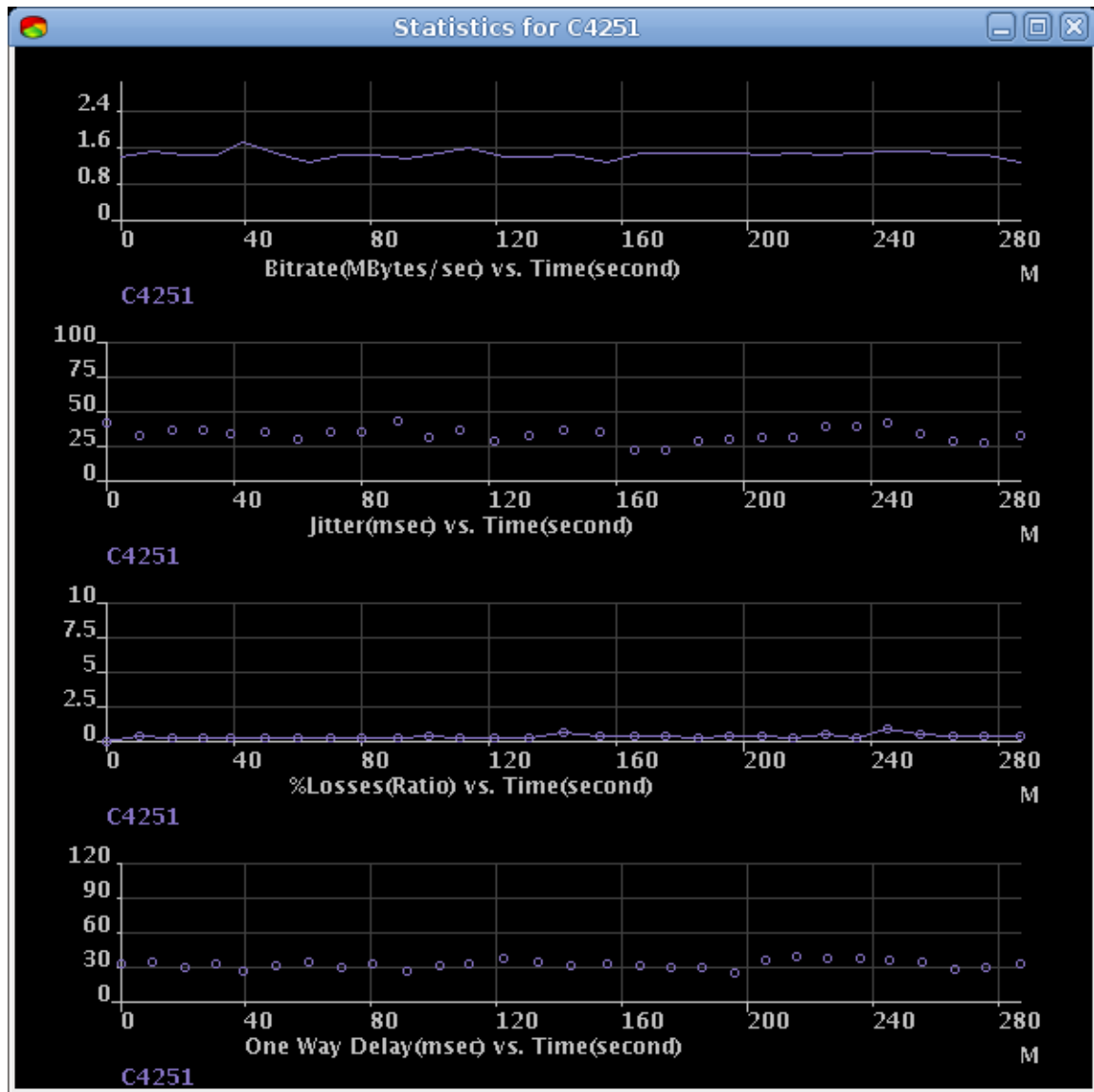


5 τελικοί αποδέκτες – Μέτρηση με το μηχανισμό σε λειτουργία

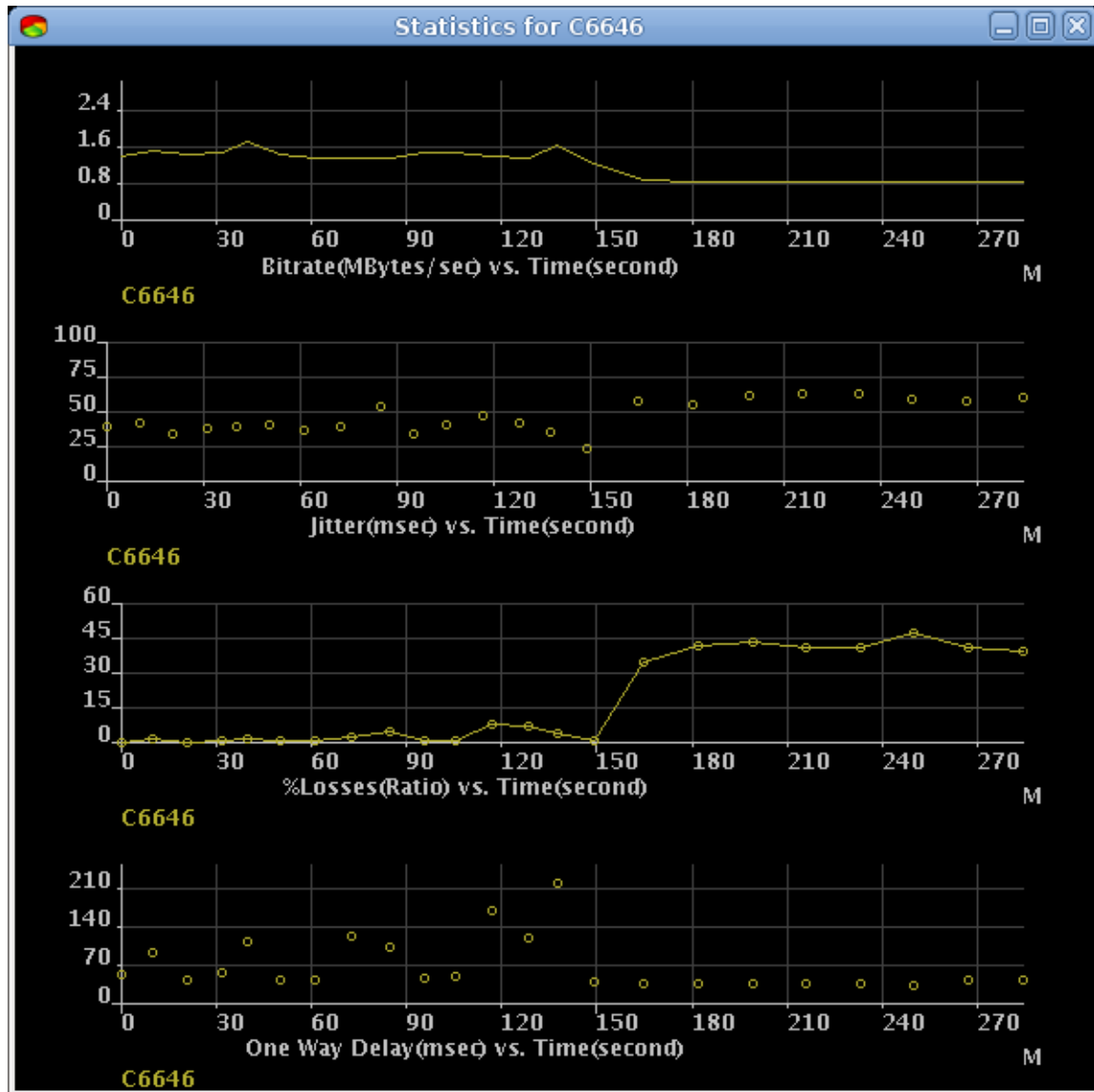


A.5.3. Ροή στη θύρα 5002

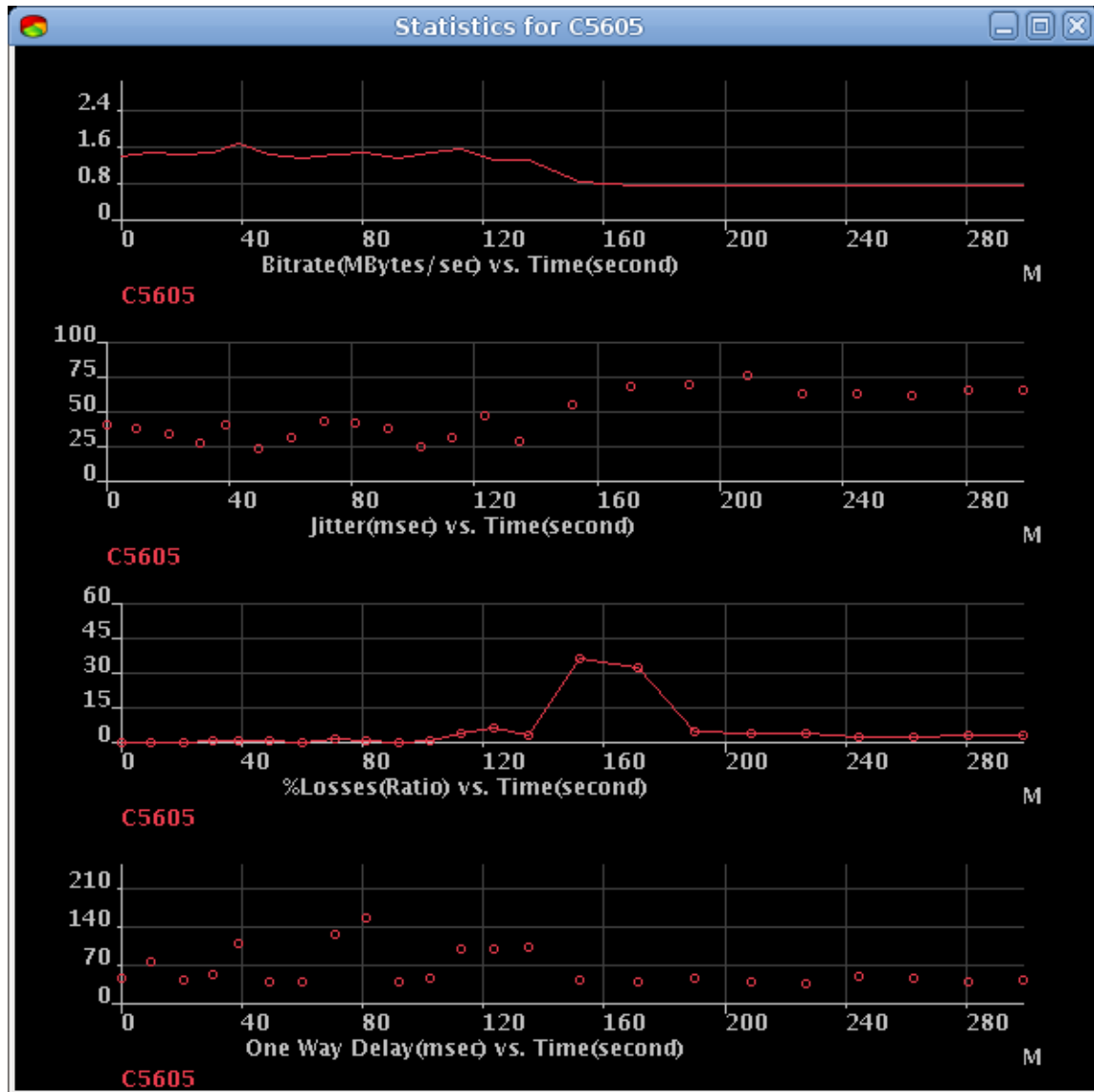
3 τελικοί αποδέκτες – Μέτρηση υπο κανονικές συνθήκες



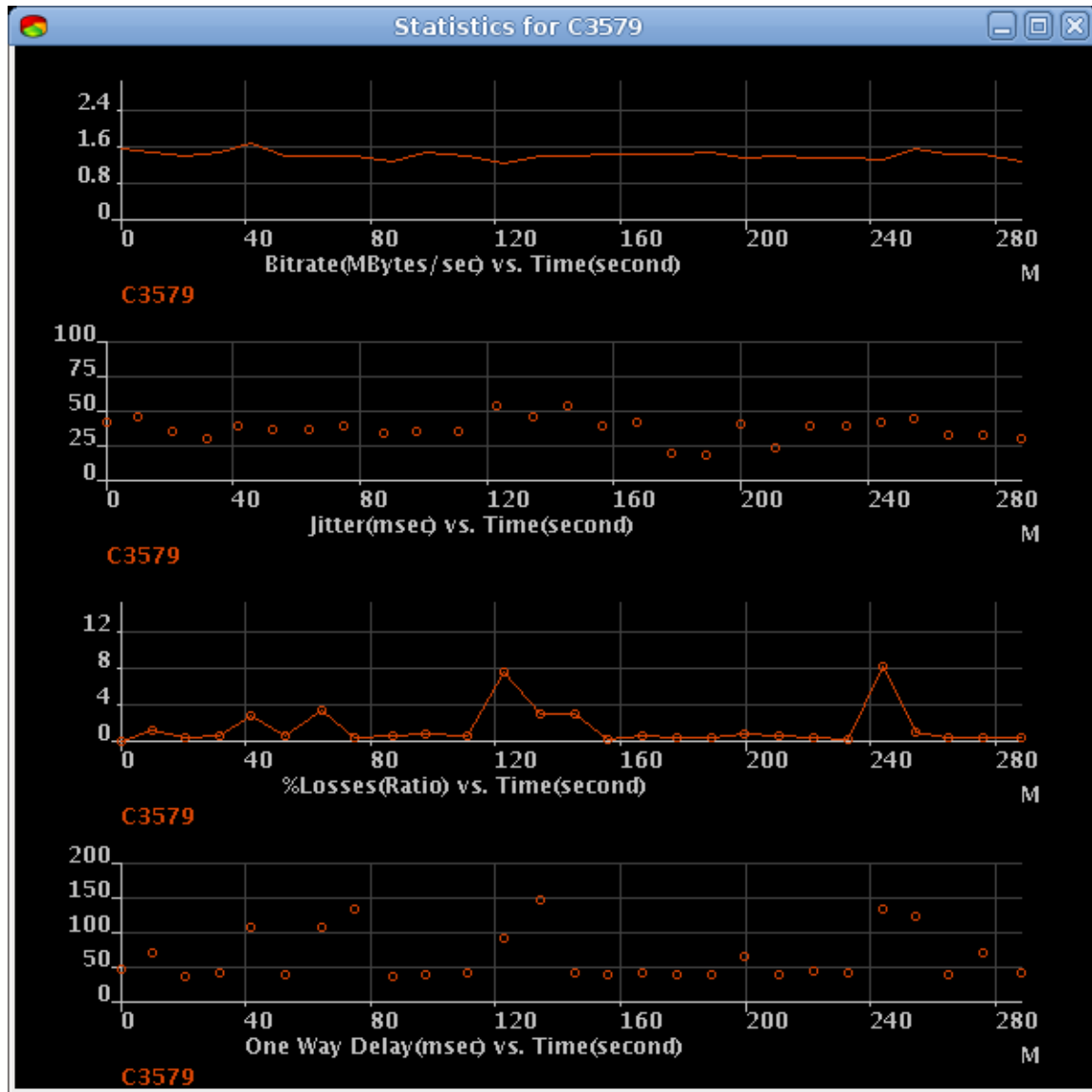
3 τελικοί αποδέκτες – Μέτρηση υπο δυσμενείς συνθήκες



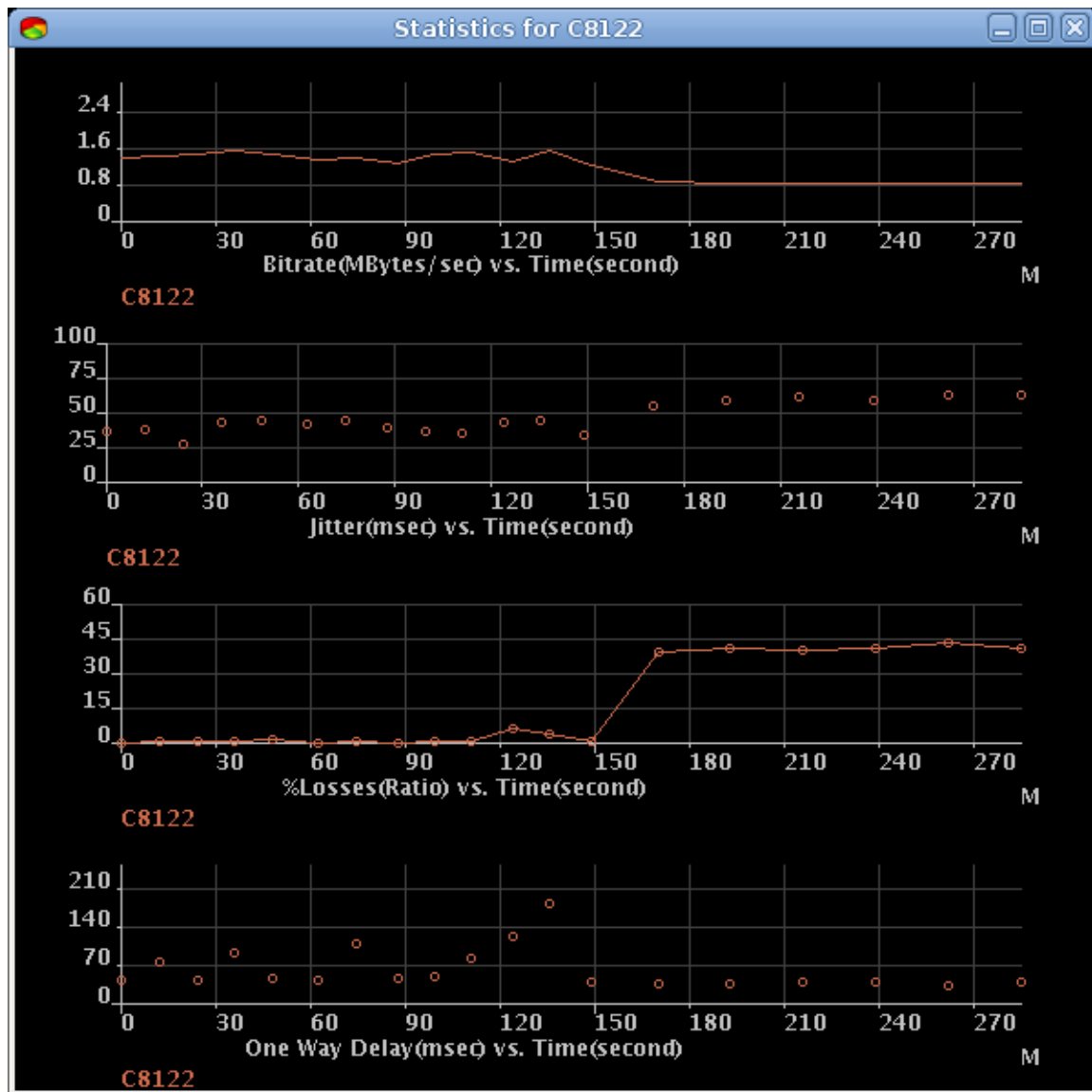
3 τελικοί αποδέκτες – Μέτρηση με το μηχανισμό σε λειτουργία



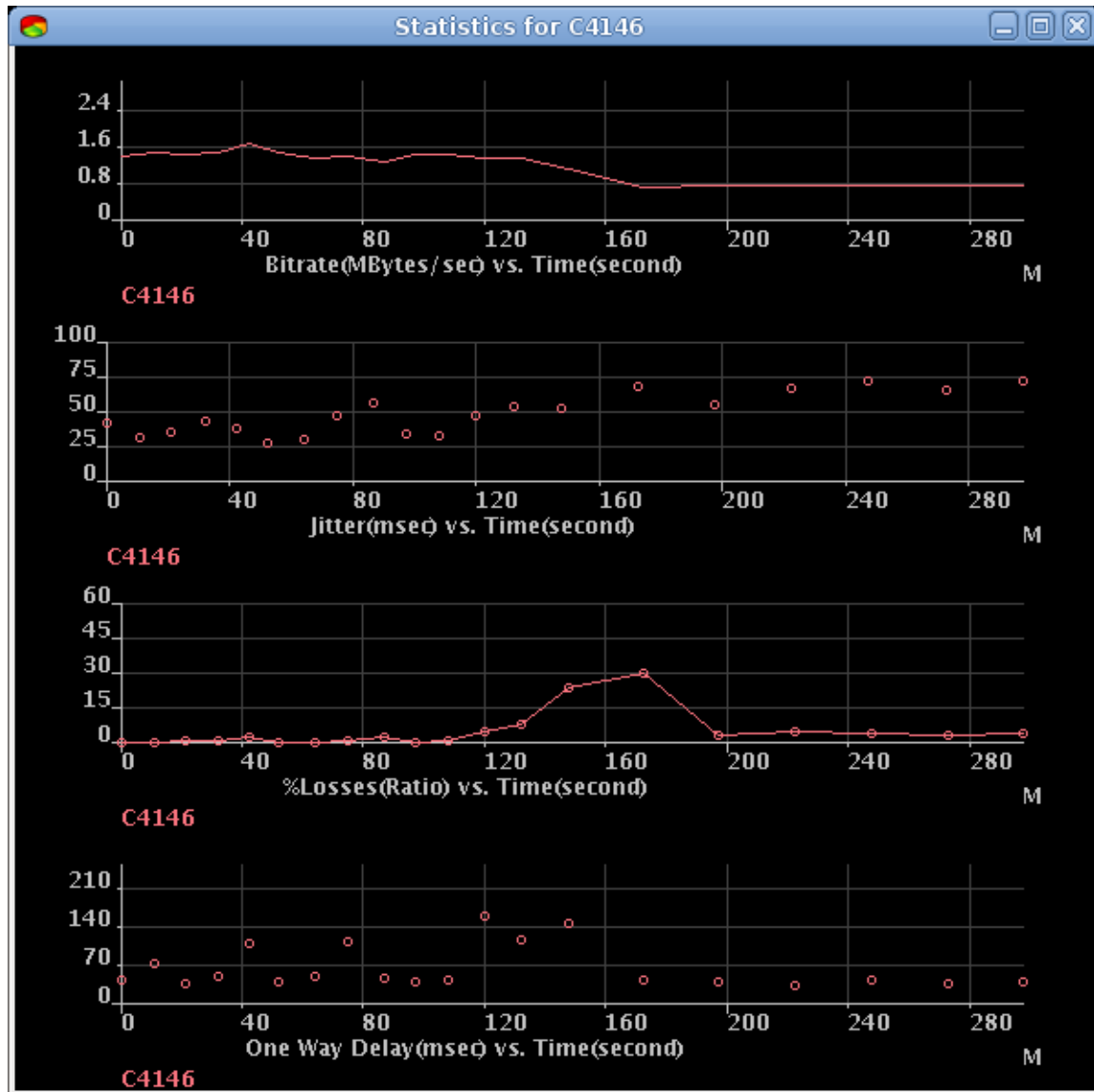
5 τελικοί αποδέκτες – Μέτρηση υπο κανονικές συνθήκες



5 τελικοί αποδέκτες – Μέτρηση υπο δυσμενείς συνθήκες

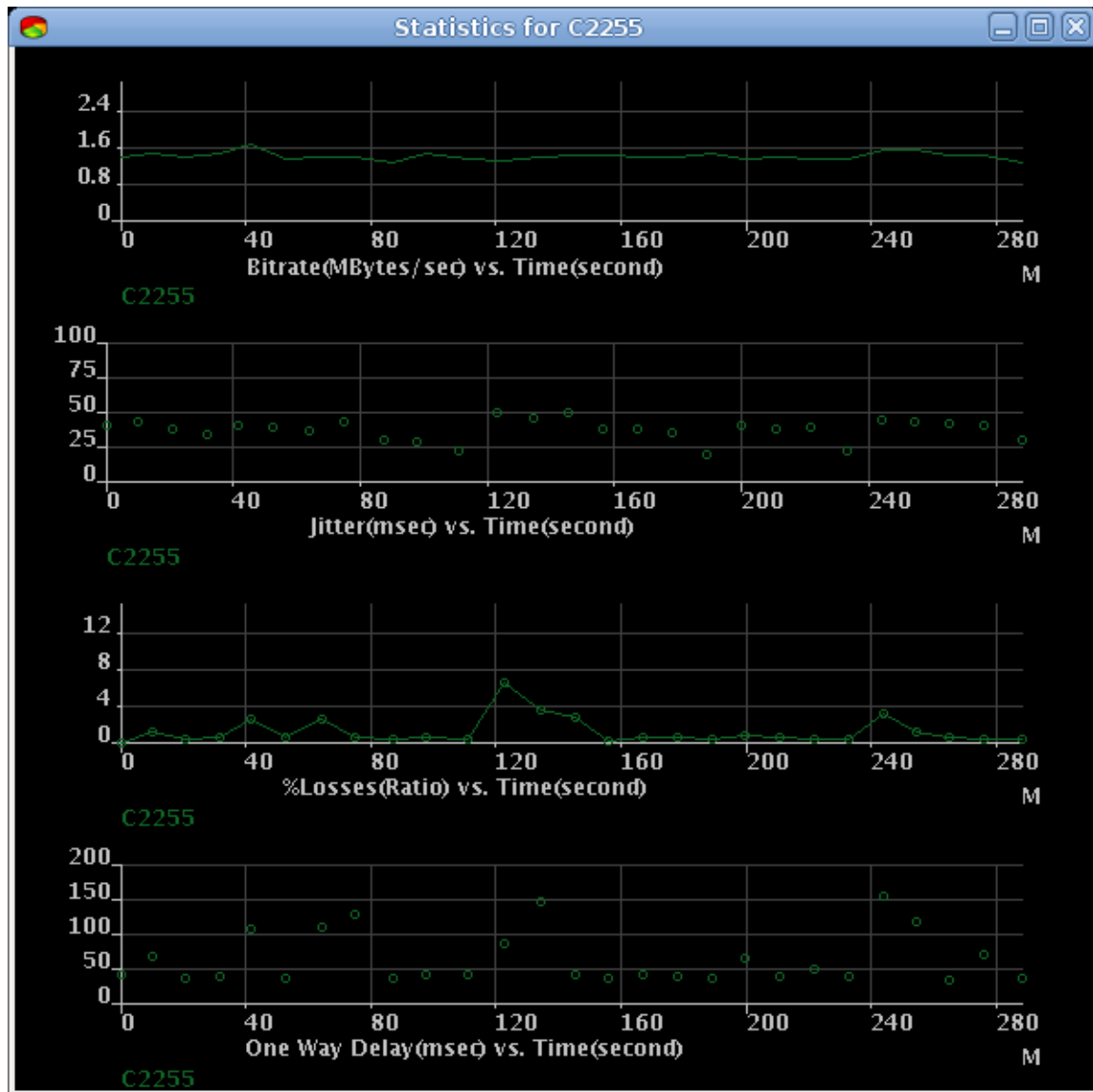


5 τελικοί αποδέκτες – Μέτρηση με το μηχανισμό σε λειτουργία

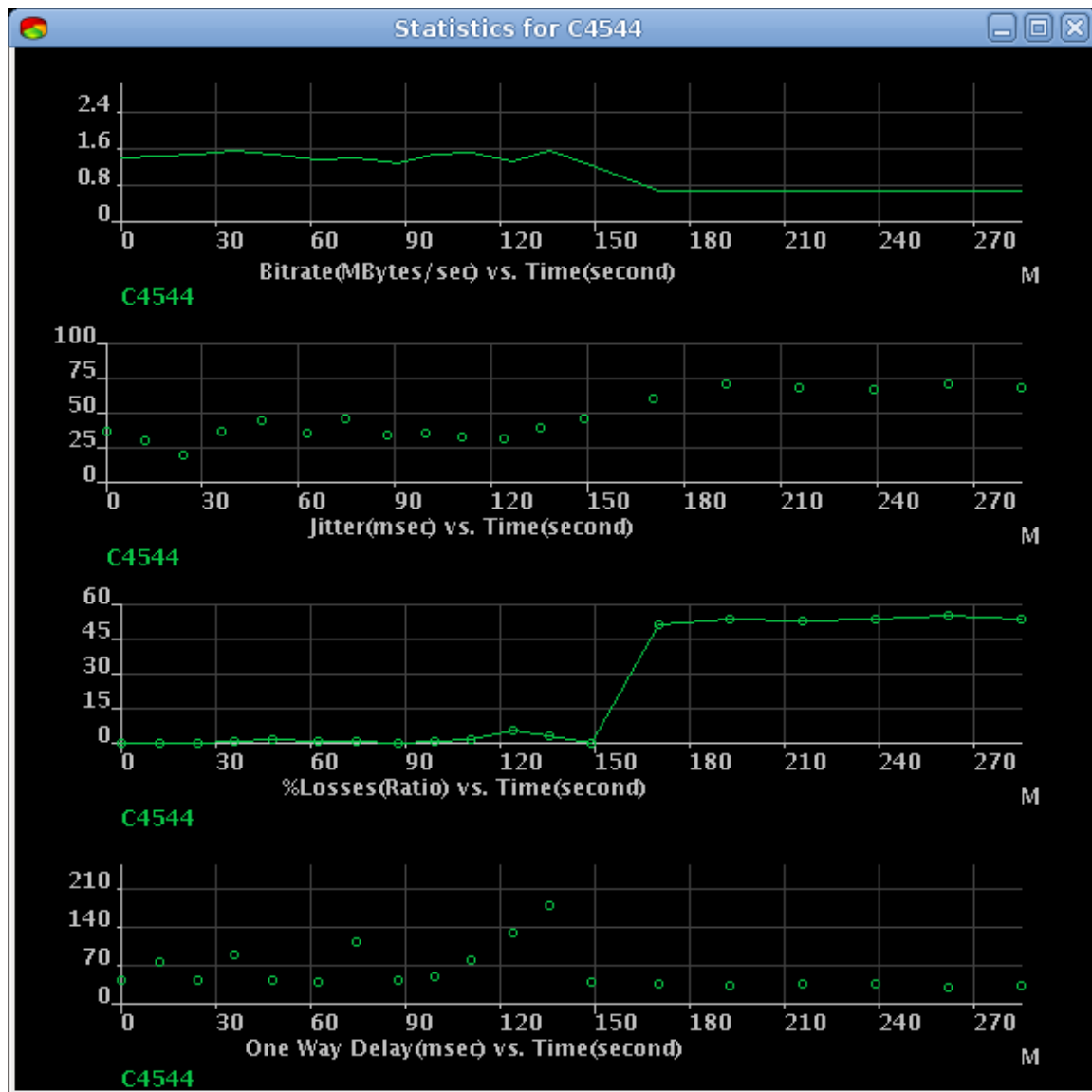


A.5.4. Ροή στη θύρα 5003

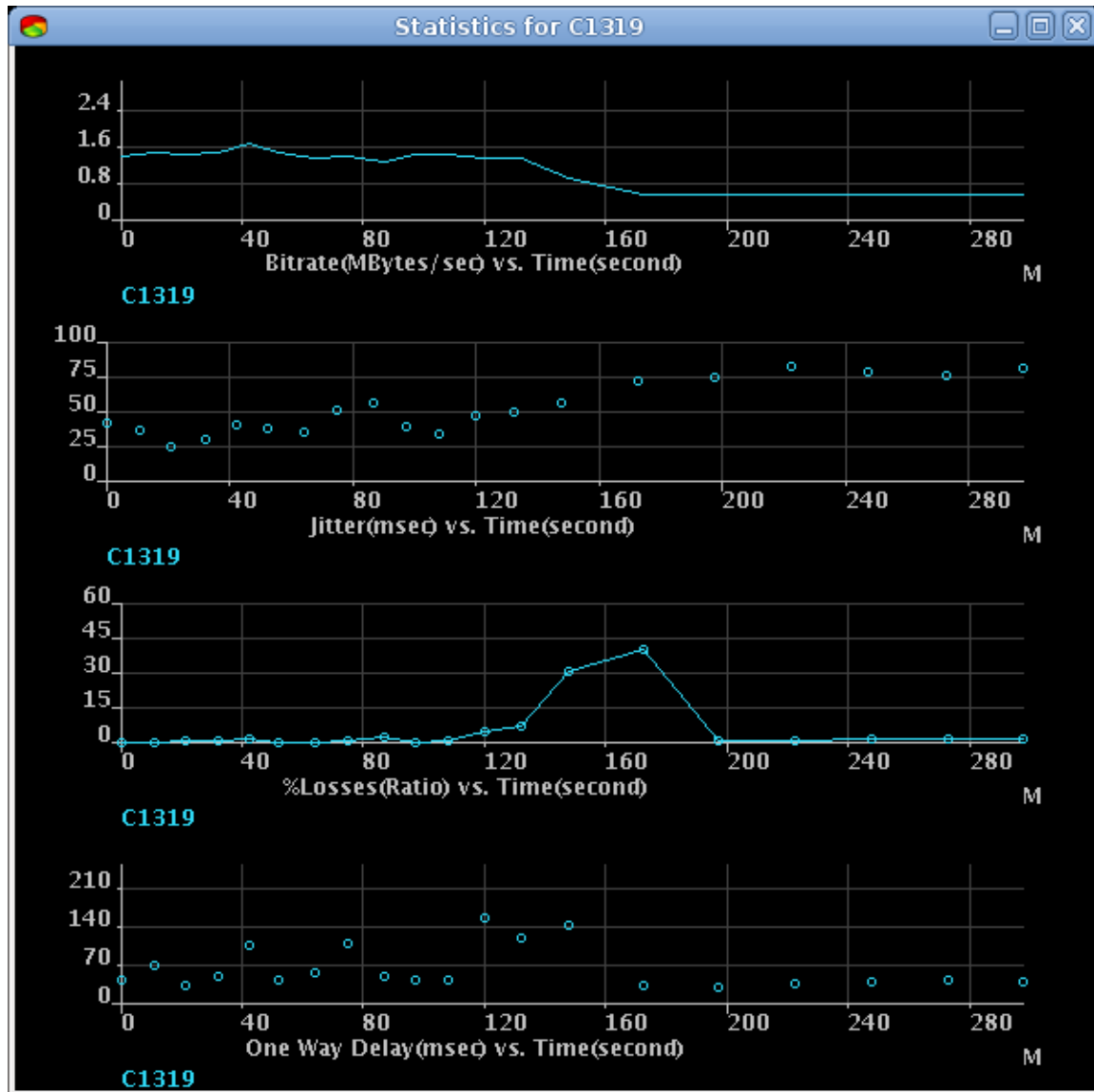
5 τελικοί αποδέκτες – Μέτρηση υπο κανονικές συνθήκες



5 τελικοί αποδέκτες – Μέτρηση υπο δυσμενείς συνθήκες

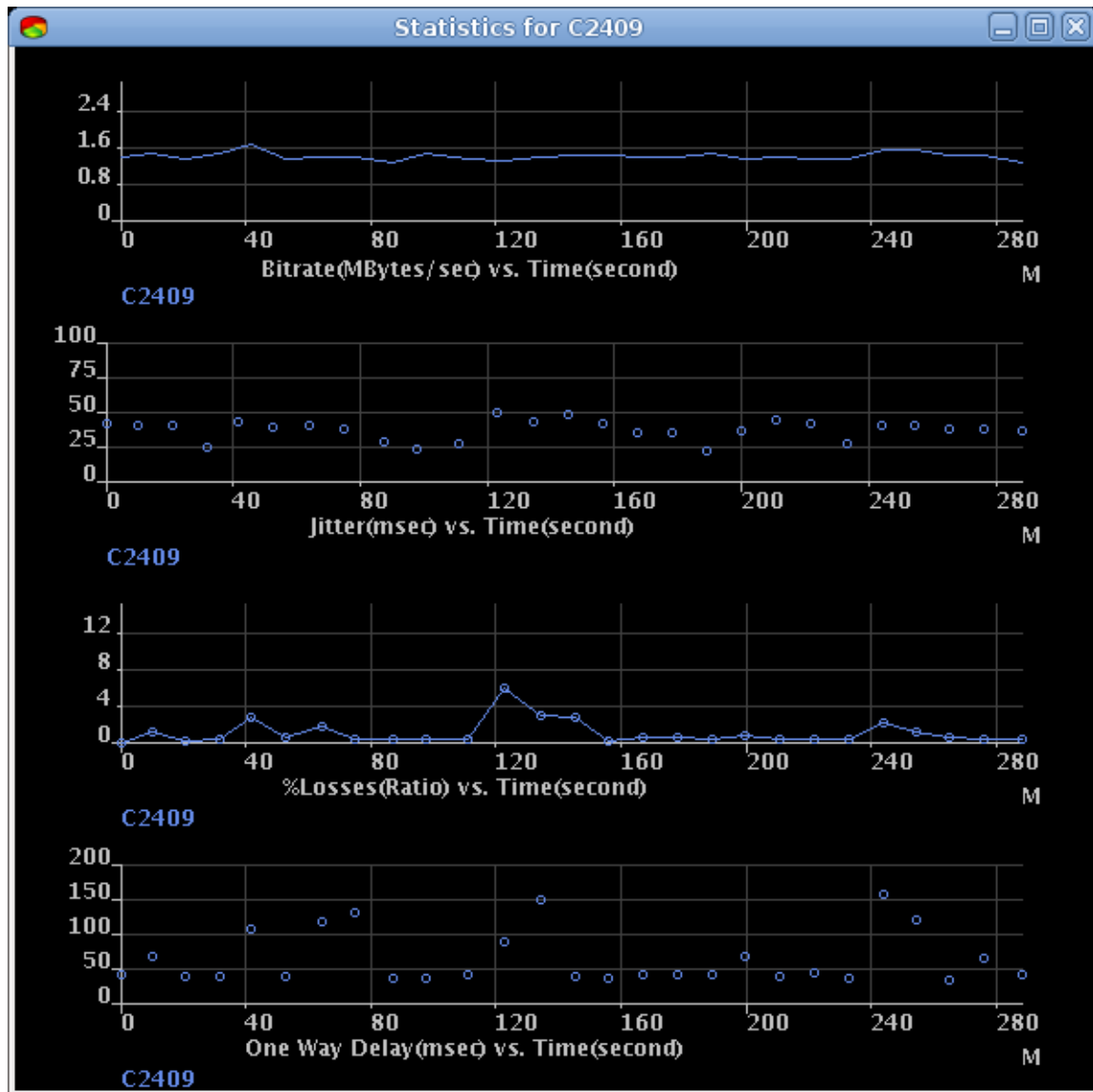


5 τελικοί αποδέκτες – Μέτρηση με το μηχανισμό σε λειτουργία

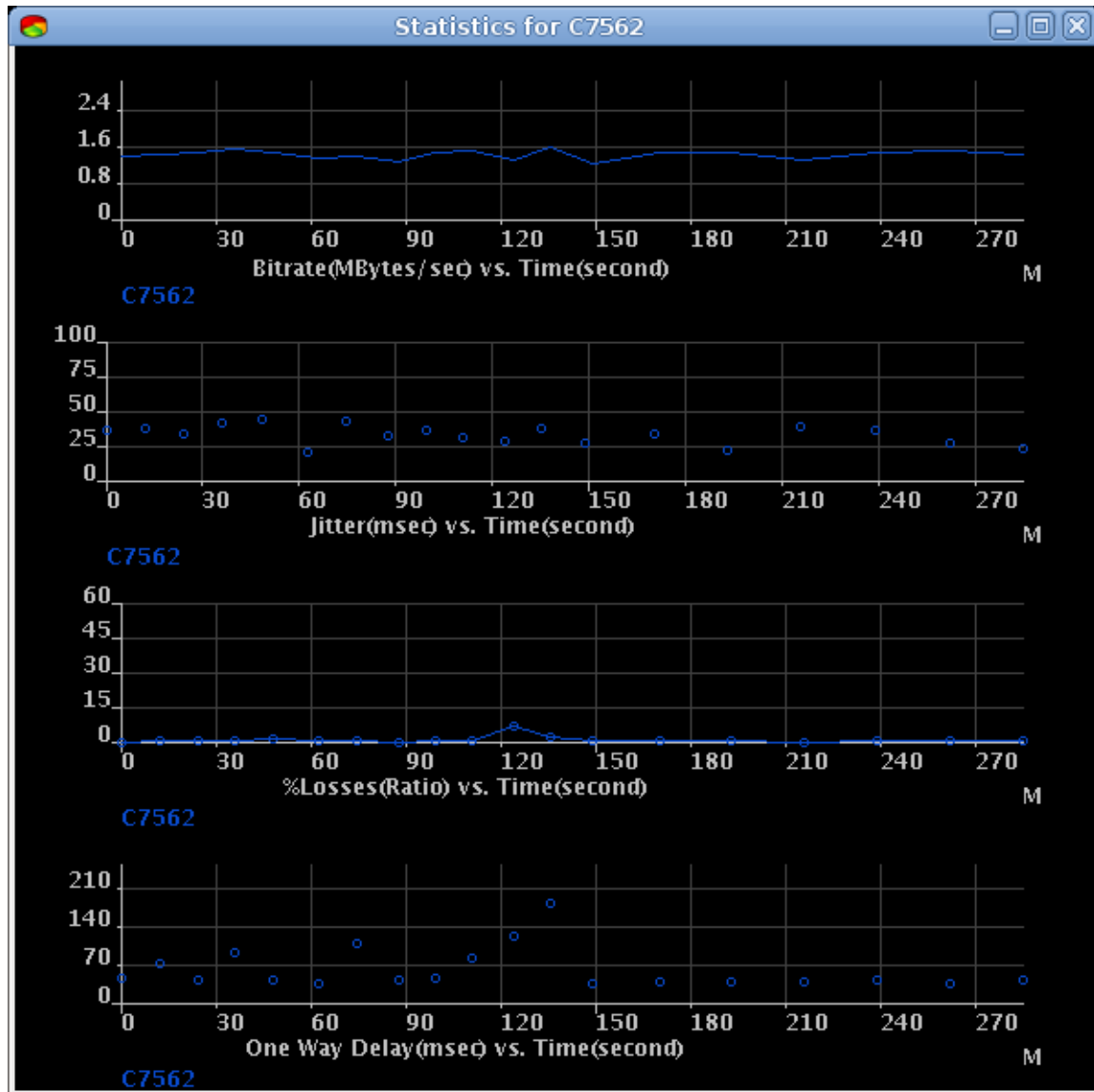


A.5.5. Ροή στη θύρα 5004

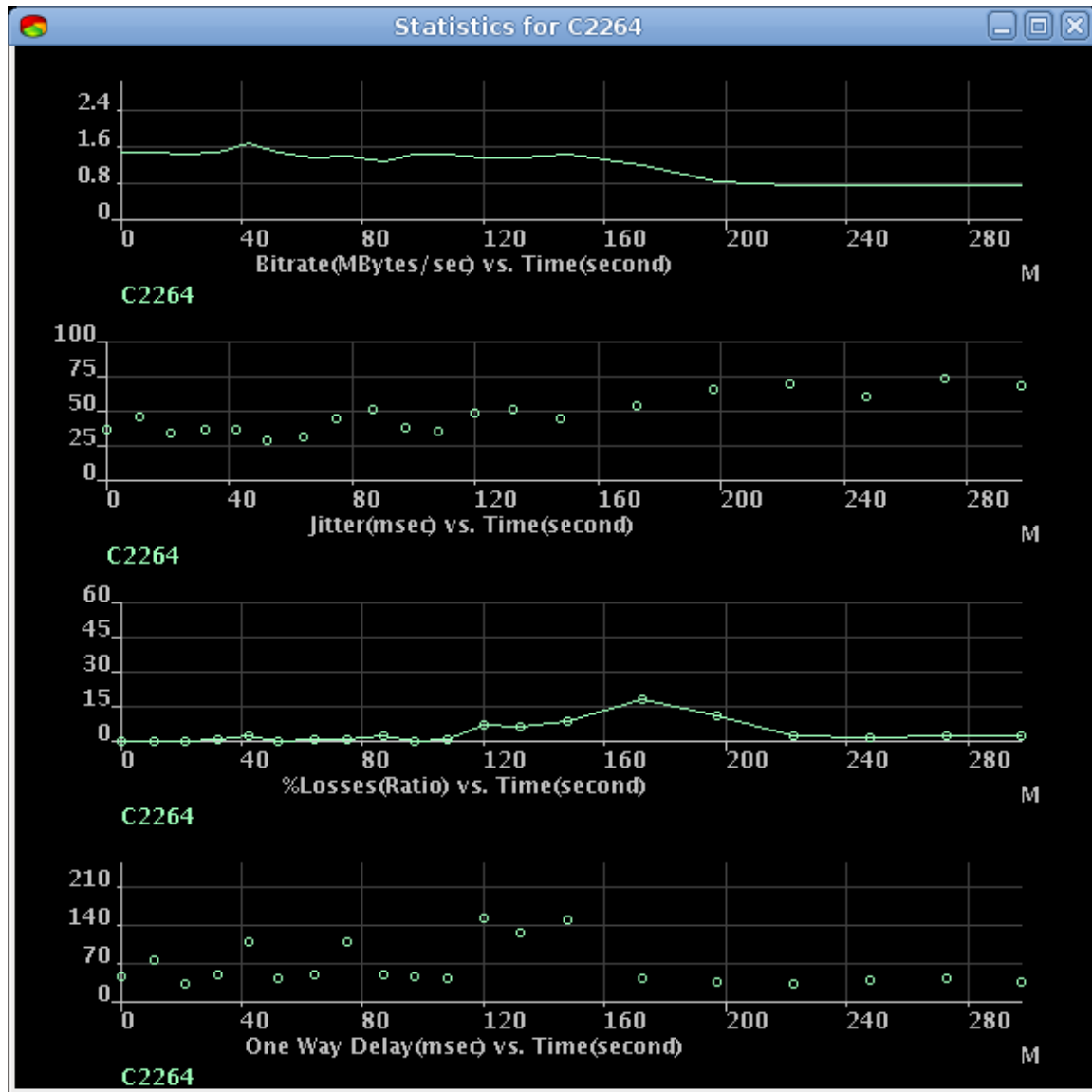
5 τελικοί αποδέκτες – Μέτρηση υπο κανονικές συνθήκες



5 τελικοί αποδέκτες – Μέτρηση υπο δυσμενείς συνθήκες



5 τελικοί αποδέκτες – Μέτρηση με το μηχανισμό σε λειτουργία



Βιβλιογραφία

- [1] iproute2+tc notes: <http://snafu.freedom.org/linux2.2/iproute-notes.html>
- [2] *“Digital Switchover: Developing Infrastructures for Broadband Access”, 6th Information Society Technologies, 6th Framework Programme (ATHENA FP6-507312).*
- [3] *Digital Television Broadcasting (DVB) forum.* <http://www.dvb.org>
- [4] ETS 300 744: Digital Video Broadcasting (DVB): Framing structure, channel coding and modulation for Digital Terrestrial Television (DVB-T). (1997). ETSI.
- [5] *IP Performance Metrics (IPPM) Working group.* (n.d.). Retrieved from <http://www.ietf.org/html.charters/ippm-charter.html>
- [6] *Linux Advanced Routing & Traffic Control.* <http://lartc.org/>
- [7] Ni., X. X. (March/April 1999). *Internet QoS: A Big Picture.* IEEE Network.
- [8] Tenenbaum, A. *“Δίκτυα Υπολογιστών”, Τρίτη έκδοση.*
- [9] *The Internet Engineering Task Force (IETF).* <http://www.ietf.org/>
- [10] *VLC Media Player.* <http://www.videolan.org/vlc/>
- [11] *Wikipedia Encyclopedia.* <http://en.wikipedia.org>
- [12] *IBM Developers.* <http://www.ibm.com/developerworks/xml/library/x-dom4j.html>
- [13] *Sun's Java Tutorial.* <http://java.sun.com/docs/books/tutorial>
- [14] *JPCap webpage* <http://netresearch.ics.uci.edu/kfujii/jpcap/doc/>
- [15] Sideris, A. *End-to-end quality of service in an interactive digital TV environment*
- [16] Ivor Horton *Beginning C: From Novice to Professional, Fourth Edition*