

**ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΚΡΗΤΗΣ**

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



Πτυχιακή Εργασία

**“Διατερματική ποιότητα υπηρεσίας σε ένα δίκτυο
DVB-T”**

**ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΤΣΟΓΚΟΣ ΣΩΤΗΡΙΟΣ
ΗΜΕΡΟΜΗΝΙΑ: 06/07/2007**

ΕΙΣΗΓΗΤΗΣ: ΣΙΔΕΡΗΣ ΑΝΑΡΓΥΡΟΣ

Στην οικογένεια μου,
με ιδιαίτερη εκτίμηση και αγάπη

Ευχαριστίες

Με την ολοκλήρωση της πτυχιακής μου εργασίας, η οποία υλοποιήθηκε στο Εργαστήριο Τηλεπικοινωνιών και Δικτύων (ΠΑΣΙΦΑΗ) του Α.Τ.Ε.Ι Κρήτης, θα ήθελα να ευχαριστήσω τους ανθρώπους οι οποίοι βοήθησαν στην περάτωση αυτής της εργασίας.

Κατά κύριο λόγο, οφείλω να ευχαριστήσω τον επιβλέποντά μου από το Α.Τ.Ε.Ι Κρήτης κ. Σιδέρη Ανάργυρο ο οποίος με υποστήριξε καθ' όλη τη διάρκεια της πτυχιακής εργασίας. Αισθάνομαι όσο λίγοι, ευνοημένος που στο διάστημα αυτό ένοιωθα πάντα τη σιγουριά ότι σε κάθε βήμα είχα την υλική αλλά και ηθική βοήθεια που χρειαζόμουν για να προχωρήσω στην ολοκλήρωση αυτής της εργασίας. Ευγνωμοσύνη οφείλω και στους Ερευνητές καθηγητές του Α.Τ.Ε.Ι Κρήτης Δρ. Ζαχαρόπουλο Βασίλειο, Δρ. Πάλλη Ευάγγελο, Μαστοράκη Γεώργιο, Μαρκάκη Ευάγγελο και Ξυλούρη Γεώργιο.

Τέλος, ευχαριστώ όλους εκείνους που ήταν δίπλα μου σε όλη αυτή την προσπάθεια παρέχοντας απεριόριστη ψυχολογική υποστήριξη και κατανόηση.

Ηράκλειο, Ιούλιος 2007

Τσόγκος Σωτήριος

Περιεχόμενα

1. Εισαγωγή	10
1.1 Γενικά.....	10
1.2 Αντικείμενο Εργασίας.....	11
1.3 Διάρθρωση Εργασίας.....	11
2. Θεωρητικό Υπόβαθρο.....	13
2.1 DVB	13
2.1.1 DVB-T	13
2.2 xDSL.....	15
2.3 WLAN.....	16
2.4 VOIP	18
2.5 Αρχιτεκτονική Δικτύου Διαδραστικής Ψηφιακής Τηλεόρασης.....	19
2.5.1 Εισαγωγή	19
2.5.2 Το δίκτυο ATHENA	19
2.6 Ποιότητα Υπηρεσίας.....	21
2.6.1 Τεχνολογίες παροχής Ποιότητας Υπηρεσίας.....	22
2.7 Διαφοροποιημένες Υπηρεσίες (Differentiated Services – DiffServ).....	23
2.7.1 Εισαγωγή	23
2.7.2 Αρχιτεκτονική Διαφοροποιημένων Υπηρεσιών	24
2.7.3 Per Hop Behavior.....	27
2.8 Αλγόριθμοι Μορφοποίησης Κίνησης.....	28
2.8.1 FIFO.....	28
2.8.2 PRIO	28
2.8.3 SFQ	29
2.8.4 RED.....	30
2.8.5 HTB.....	31
2.8.6 DSMARK	32
2.8.7 Ταξινόμητες	32
2.9 Πρωτόκολλα Μεταφοράς Δεδομένων	33
2.9.1 Το Πρωτόκολλο TCP.....	33
2.9.2 Το Πρωτόκολλο UDP	35
3. Παρουσίαση και Υλοποίηση Αρχιτεκτονικής Δικτύου Πειραματικών Μετρήσεων....	37
3.1 Εισαγωγή	37
3.2 Σχεδιασμός Συστήματος	37
3.3 Υλοποίηση Δικτύου Πειραματικών Μετρήσεων.....	40
3.3.1 Ρύθμιση Εξοπλισμού	42
3.3.1.1 Ρύθμιση Παροχέα Υπηρεσιών και του αντίστοιχου CMN-DER.....	42
3.3.1.2 Ρύθμιση Τελικού Χρήστη και του αντίστοιχου CMN-DER.....	43
3.3.1.3 Ρύθμιση Γεννήτριας Κίνησης και του αντίστοιχου CMN-DER.....	44
3.3.1.4 Ρύθμιση DVB-T πλατφόρμας και DCR	45
3.4 Περιγραφή Πειραματικής Διαδικασίας.....	46
3.4.1 Σενάριο 1 ^ο	47
3.4.2 Σενάριο 2 ^ο	47
3.4.3 Σενάριο 3 ^ο	48
3.4.4 Σενάριο 4 ^ο	49

3.5 Μετρική Επιδόσεων.....	50
3.6 Περιγραφή Προγραμμάτων	51
4. Παρουσίαση και Ανάλυση Αποτελεσμάτων.....	53
4.1 Σενάριο 1 ^ο	53
4.1.1 Μέρος 1 ^ο	53
4.1.2 Μέρος 2 ^ο	55
4.1.3 Μέρος 3 ^ο	57
4.2 Σενάριο 2 ^ο	58
4.2.1 Μέρος 1 ^ο	58
4.2.2 Μέρος 2 ^ο	60
4.2.3 Μέρος 3 ^ο	62
4.3 Σενάριο 3 ^ο	63
4.3.1 Μέρος 1 ^ο	63
4.3.2 Μέρος 2 ^ο	66
4.3.3 Μέρος 3 ^ο	70
4.4 Σενάριο 4 ^ο	72
4.4.1 Μέρος 1 ^ο	72
4.4.1.1 TCP κίνηση.....	72
4.4.1.2 UDP κίνηση	74
4.4.1.3 VoIP επικοινωνία.....	75
4.4.2 Μέρος 2 ^ο	78
4.4.2.1 TCP κίνηση.....	78
4.4.2.2 UDP κίνηση	80
4.4.2.3 VoIP επικοινωνία.....	82
4.4.3 Μέρος 3 ^ο	85
4.4.3.1 TCP κίνηση.....	85
4.4.3.2 UDP κίνηση	87
4.4.3.3 VoIP επικοινωνία.....	88
5. Συμπεράσματα - Επίλογος.....	91
5.1 Αποτελέσματα.....	91
5.2 Μελλοντικές Προτάσεις.....	92
6. Βιβλιογραφία - Αναφορές.....	94
7. Παραρτήματα.....	96
7.1 Γλωσσάριο	96
7.2 Εντολές Εφαρμογής DiffServ	98
7.2.1 Σενάριο 1 ^ο	98
7.2.1.1 DiffServ Edge Router	98
7.2.1.2 DiffServ Core Router.....	99
7.2.2 Σενάριο 2 ^ο	100
7.2.2.1 DiffServ Edge Router	100
7.2.2.2 DiffServ Core Router.....	102
7.2.3 Σενάριο 3 ^ο	103
7.2.3.1 DiffServ Edge Router	103
7.2.3.2 DiffServ Core Router.....	104
7.2.4 Σενάριο 4 ^ο	105
7.2.4.1 DiffServ Edge Router	105

7.2.4.2 DiffServ Core Router.....	107
7.3 Εντολές Δημιουργίας, “Σύλληψης” και Ανάλυσης Κίνησης.....	109
7.3.1 TCP κίνηση.....	109
7.3.1.1 Δημιουργία.....	109
7.3.1.2 Σύλληψη.....	109
7.3.1.3 Ανάλυση.....	110
7.3.1.4 Γραφική Απεικόνιση.....	110
7.3.2 UDP κίνηση.....	110
7.3.2.1 Συγχρονισμός ρολογιών.....	110
7.3.2.2 Δημιουργία.....	111
7.3.2.3 Σύλληψη.....	111
7.3.2.4 Ανάλυση.....	112
7.3.2.5 Γραφική Απεικόνιση.....	113
7.3.3 VoIP κίνηση.....	113
7.4 Προγράμματα Ανάλυσης UDP Κίνησης.....	114
7.4.1 Πρόγραμμα ipn4_dvbstream_createendfiles.pl.....	114
7.4.2 Πρόγραμμα ipn4_losses.pl.....	118
7.4.3 Πρόγραμμα ipn4_sender_receiver_rate.pl.....	125
7.4.4 Πρόγραμμα ipn4_align_for_delay_jitt.pl.....	128
7.4.5 Πρόγραμμα ipn4_timestamp.pl.....	129
7.4.6 Πρόγραμμα ipn4_inter_arrival_for_jitter.pl.....	131
7.4.7 Πρόγραμμα ipn4_jitter.pl.....	138
7.4.8 Πρόγραμμα ipn4_one_way_delay.pl.....	140
7.4.9 Πρόγραμμα losses.gpl.....	143
7.4.10 Πρόγραμμα onewaydelay.gpl.....	143
7.4.11 Πρόγραμμα jitter.gpl.....	144

Λίστα Σχημάτων

Σχήμα 1 : Διάγραμμα ενός DVB-T πομπού.....	14
Σχήμα 2 : Αρχιτεκτονική xDSL δικτύου.	15
Σχήμα 3 : Αρχιτεκτονική WLAN δικτύου.....	17
Σχήμα 4 : Γενική αρχιτεκτονική ενός VoIP δικτύου.....	18
Σχήμα 5 : Γενική αρχιτεκτονική ATHENA IST FP6-507312.....	20
Σχήμα 6 : Σχηματική αναπαράσταση μιας περιοχής Διαφοροποιημένων Υπηρεσιών.....	25
Σχήμα 7 : Ενέργειες για την εφαρμογή DiffServ σε : α)DER, β)DCR.....	25
Σχήμα 8 : α)Η επικεφαλίδα των IP πακέτων, β) το πεδίο ToS, γ) το πεδίο DSCP.....	26
Σχήμα 9 : Σχηματική περιγραφή του αλγόριθμου FIFO.....	28
Σχήμα 10 : Σχηματική περιγραφή του αλγόριθμου PRIO.....	29
Σχήμα 11 : Σχηματική περιγραφή του αλγόριθμου SFQ.....	29
Σχήμα 12 : Σχηματική περιγραφή του αλγόριθμου RED.....	30
Σχήμα 13 : α) Περιγραφή της λειτουργίας του αλγόριθμου TBF, β) αναπαράσταση της διαμοίρασης του εύρους ζώνης.....	31
Σχήμα 14 : Σχηματική περιγραφή του αλγόριθμου DSMARK.....	32
Σχήμα 15 : Η επικεφαλίδα TCP.....	34
Σχήμα 16 : Η επικεφαλίδα UDP.....	35
Σχήμα 17 : Δίκτυο Πυρήνα του συστήματος.....	38
Σχήμα 18 : Δίκτυα Πρόσβασης του συστήματος.....	39
Σχήμα 19 : Γενική αρχιτεκτονική συστήματος IDTV με DiffServ.....	40
Σχήμα 20 : Αρχιτεκτονική συστήματος IDTV με DiffServ.....	41
Σχήμα 21 : Ρύθμιση του Service Provider και του αντίστοιχου CMN-DER.....	42
Σχήμα 22 : Ρύθμιση του τελικού χρήστη και του αντίστοιχου CMN-DER.....	43
Σχήμα 23 : Ρύθμιση του background traffic generator και του αντίστοιχου CMN-DER.....	44
Σχήμα 24 : Ο διαμορφωτής COFDM που παρουσιάζει τα στοιχεία της διαμόρφωσης.....	45
Σχήμα 25 : Γραφική παράσταση του Throughput.....	54
Σχήμα 26 : Γραφική παράσταση του RTT.....	54
Σχήμα 27 : Γραφική παράσταση του Throughput.....	55
Σχήμα 28 : Γραφική παράσταση του RTT.....	56
Σχήμα 29 : Γραφική παράσταση των retransmissions.....	56
Σχήμα 30 : Γραφική παράσταση του Throughput.....	57
Σχήμα 31 : Γραφική παράσταση του RTT.....	58
Σχήμα 32 : Γραφική παράσταση της One Way Delay.....	59
Σχήμα 33 : Γραφική παράσταση του Jitter.....	59
Σχήμα 34 : Γραφική παράσταση της One Way Delay.....	60
Σχήμα 35 : Γραφική παράσταση του Jitter.....	61
Σχήμα 36 : Γραφική παράσταση των απωλειών.....	61
Σχήμα 37 : Γραφική παράσταση της One Way Delay.....	62
Σχήμα 38 : Γραφική παράσταση του Jitter.....	63
Σχήμα 39 : Γραφική παράσταση της One Way Delay.....	64
Σχήμα 40 : Γραφική παράσταση του Jitter.....	64
Σχήμα 41 : Γραφική παράσταση της One Way Delay.....	65
Σχήμα 42 : Γραφική παράσταση του Jitter.....	65
Σχήμα 43 : Γραφική παράσταση της One Way Delay.....	66

Σχήμα 44 : Γραφική παράσταση του Jitter.	67
Σχήμα 45 : Γραφική παράσταση των απωλειών.	67
Σχήμα 46 : Γραφική παράσταση της One Way Delay.	68
Σχήμα 47 : Γραφική παράσταση του Jitter.	69
Σχήμα 48 : Γραφική παράσταση των απωλειών.	69
Σχήμα 49 : Γραφική παράσταση της One Way Delay.	70
Σχήμα 50 : Γραφική παράσταση του Jitter.	71
Σχήμα 51 : Γραφική παράσταση της One Way Delay.	71
Σχήμα 52 : Γραφική παράσταση του Jitter.	72
Σχήμα 53 : Γραφική παράσταση του Throughput.	73
Σχήμα 54 : Γραφική παράσταση του RTT.	73
Σχήμα 55 : Γραφική παράσταση της One Way Delay.	74
Σχήμα 56 : Γραφική παράσταση του Jitter.	75
Σχήμα 57 : Γραφική παράσταση της One Way Delay.	76
Σχήμα 58 : Γραφική παράσταση του Jitter.	76
Σχήμα 59 : Γραφική παράσταση της One Way Delay.	77
Σχήμα 60 : Γραφική παράσταση του Jitter.	77
Σχήμα 61 : Γραφική παράσταση του Throughput.	78
Σχήμα 62 : Γραφική παράσταση του RTT.	79
Σχήμα 63 : Γραφική παράσταση των retransmissions.	79
Σχήμα 64 : Γραφική παράσταση της One Way Delay.	80
Σχήμα 65 : Γραφική παράσταση του Jitter.	81
Σχήμα 66 : Γραφική παράσταση των απωλειών.	81
Σχήμα 67 : Γραφική παράσταση της One Way Delay.	82
Σχήμα 68 : Γραφική παράσταση του Jitter.	83
Σχήμα 69 : Γραφική παράσταση των απωλειών.	83
Σχήμα 70 : Γραφική παράσταση της One Way Delay.	84
Σχήμα 71 : Γραφική παράσταση του Jitter.	84
Σχήμα 72 : Γραφική παράσταση των απωλειών.	85
Σχήμα 73 : Γραφική παράσταση του Throughput.	86
Σχήμα 74 : Γραφική παράσταση του RTT.	86
Σχήμα 75 : Γραφική παράσταση της One Way Delay.	87
Σχήμα 77 : Γραφική παράσταση της One Way Delay.	89
Σχήμα 78 : Γραφική παράσταση του Jitter.	89
Σχήμα 79 : Γραφική παράσταση της One Way Delay.	90
Σχήμα 80 : Γραφική παράσταση του Jitter.	90

Λίστα Πινάκων

Πίνακας 1 : Συνοπτική περιγραφή των 802.11 πρωτοκόλλων.	16
Πίνακας 2 : Παρουσίαση των τιμών του DSCP πεδίου των AF κλάσεων.	27
Πίνακας 3 : Τιμές Αξιολόγησης των χαρακτηριστικών QoS.	53

1. Εισαγωγή

1.1 Γενικά

Η εξέλιξη των τεχνολογιών ψηφιακής τηλεοπτικής μετάδοσης και η σύγκλιση αυτών με τις υπάρχουσες τηλεπικοινωνιακές τεχνολογίες οδήγησε στην ανάπτυξη καινοτόμων συστημάτων, που χρησιμοποιούν ένα κοινό κανάλι μεταφοράς για την παροχή ετερογενών υπηρεσιών. Τα παραπάνω συστήματα απαρτίζουν ένα περιβάλλον Διαδραστικής Ψηφιακής Τηλεόρασης (Interactive Digital Television – IDTV), μέσα από το οποίο οι τελικοί χρήστες έχουν την δυνατότητα πρόσβασης όχι μόνο σε ψηφιακά τηλεοπτικά προγράμματα, αλλά και σε διαδραστικές δικτυακές εφαρμογές, όπως Internet, IP-TV και VoIP. Η διαδραστικότητα αυτή επιτυγχάνεται με την χρήση “καναλιών επιστροφής” που μεταφέρουν τις αιτήσεις των χρηστών στους φορείς παροχής των υπηρεσιών, ενώ οι ζητούμενες υπηρεσίες μεταδίδονται προς τους χρήστες μέσω ενός UHF καναλιού. Στην Ευρώπη, για την επίγεια εκπομπή τόσο ψηφιακών τηλεοπτικών υπηρεσιών όσο και υπηρεσιών IP χρησιμοποιείται το πρότυπο DVB-T^[1], που προσφέρει υψηλούς ρυθμούς μετάδοσης και ευρεία γεωγραφική κάλυψη, ενώ για την υλοποίηση του καναλιού επιστροφής μπορεί να χρησιμοποιηθεί μια από τις σημερινές τεχνολογίες, όπως xDSL, ISDN, PSTN, WLAN, UMTS και GSM^[2]. Με τον τρόπο αυτό επιτυγχάνεται η ενοποίηση των τεχνολογιών μετάδοσης ψηφιακού τηλεοπτικού σήματος με τις τεχνολογίες των τηλεπικοινωνιών.

Ωστόσο, σε ένα τόσο περίπλοκο περιβάλλον, όπως αυτό της διαδραστικής ψηφιακής τηλεόρασης, όπου η πλατφόρμα DVB-T λειτουργεί σαν υποδομή για την παροχή δικτυακών υπηρεσιών, η χρονικά μεταβαλλόμενη κίνηση που δημιουργείται από κάθε χρήστη επηρεάζει την λειτουργία του δικτύου. Επίσης, ένας από τους σημαντικότερους παράγοντες που πρέπει να ληφθεί υπόψη είναι η “ευαισθησία” των παρεχόμενων υπηρεσιών, τόσο σε ποσοτικά όσο και ποιοτικά χαρακτηριστικά, όπως είναι η καθυστέρηση και η διακύμανση της. Το πρωτόκολλο IP^[3], που χρησιμοποιείται σήμερα διαβιβάζει τα πακέτα βασιζόμενο στο “παραδοσιακό” μοντέλο Βέλτιστης Προσπάθειας (Best Effort), που χρησιμοποιείται και στο Διαδίκτυο. Σύμφωνα με αυτό, η εξυπηρέτηση της κίνησης γίνεται το συντομότερο δυνατό, χωρίς όμως να παρέχεται

καμιά εγγύηση τόσο για το χρόνο παράδοσής της όσο και για την ίδια την παράδοση. Ορισμένες, όμως, εφαρμογές απαιτούν να λάβουν εξυπηρέτηση στην διαβίβαση των πακέτων τους που να ικανοποιεί κάποιες προϋποθέσεις, όπως χαμηλή καθυστέρηση. Συνεπώς, κρίνεται αναγκαία η ενσωμάτωση στο δίκτυο κατάλληλων μηχανισμών παροχής Ποιότητας Υπηρεσιών (Quality of Service - QoS) με σκοπό την διασφάλιση της σε ικανοποιητικά επίπεδα. Μία από τις πιο διαδεδομένες τεχνικές παροχής QoS, η οποία προσφέρει κατηγοριοποίηση της δικτυακής κίνησης και διαφοροποίηση στον τρόπο εξυπηρέτησης αυτής είναι ο μηχανισμός των Διαφοροποιημένων Υπηρεσιών (Differentiated Services – DiffServ) ^[4].

1.2 Αντικείμενο Εργασίας

Σκοπός της πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός δικτύου επίγειας ψηφιακής τηλεόρασης για την παροχή αμφίδρομων διαδραστικών υπηρεσιών. Στην συνέχεια, ακολουθεί η μελέτη του μηχανισμού λειτουργίας της τεχνολογίας DiffServ και ο τρόπος με τον οποίο μπορεί αυτή να ενσωματωθεί στην αρχιτεκτονική ενός διαδραστικού DVB-T δικτύου για την βελτιστοποίηση της απόδοσης και της ποιότητας των παρεχόμενων υπηρεσιών. Τέλος, μέσω μιας σειράς πειραμάτων που θα γίνουν καλύπτοντας διαφορετικά πιθανά σενάρια, θα πραγματοποιηθεί ανάλυση των μετρήσεων που θα προκύψουν και θα εκτιμηθούν τα αποτελέσματα της παρουσίας της τεχνολογίας DiffServ στα χαρακτηριστικά του δικτύου.

1.3 Διάρθρωση Εργασίας

Ακολουθώντας την εισαγωγή, στο δεύτερο κεφάλαιο παρατίθενται οι απαραίτητες θεωρητικές πληροφορίες που απαιτούνται ώστε ο αναγνώστης να εξοικειωθεί και να κατανοήσει την λειτουργία των τεχνολογιών που χρησιμοποιούνται για την επίτευξη αυτής της πτυχιακής εργασίας. Στο τρίτο κεφάλαιο παρουσιάζεται η αρχιτεκτονική του δικτύου παροχής διαδραστικής ψηφιακής τηλεόρασης και υπηρεσιών IP που θα χρησιμοποιηθεί και περιγράφεται η υλοποίηση του. Στην συνέχεια, στο

τέταρτο κεφάλαιο ακολουθεί η ανάλυση, η μελέτη και η παρουσίαση των αποτελεσμάτων που προκύπτουν από τις πειραματικές μετρήσεις που πραγματοποιήθηκαν. Τέλος, στο πέμπτο κεφάλαιο διατυπώνονται τα συμπεράσματα της εργασίας και παρατίθενται κάποιες προτάσεις για μελλοντικές μελέτες.

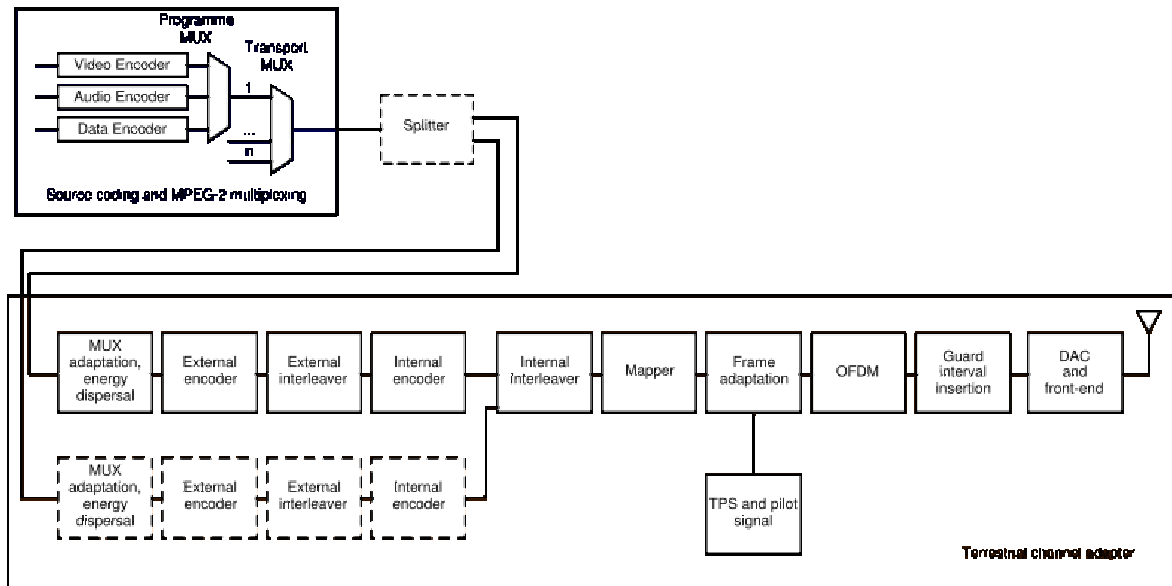
2. Θεωρητικό Υπόβαθρο

2.1 DVB

Το DVB, ακρωνύμιο για το Digital Video Broadcasting, είναι μια οικογένεια προτύπων που ορίζει τις προδιαγραφές για την ψηφιακή μετάδοση οπτικοακουστικών σημάτων και δεδομένων, χρησιμοποιώντας δορυφορικά, καλωδιακά και επίγεια μέσα μετάδοσης. Αναπτύχθηκε από το DVB Project^[5] και εγκρίθηκε από το Ευρωπαϊκό Ίδρυμα Προτύπων Τηλεπικοινωνιών (ETSI)^[6]. Τα πιο γνωστά μέλη της οικογένειας προτύπων DVB είναι τα : DVB-S και DVB-S2 (δορυφορικά), DVB-C (καλωδιακό), DVB-T (επίγειο) και DVB-H (επίγειο για κινητό χρήστη).

2.1.1 DVB-T

Ένα από τα πιο διαδεδομένα πρότυπα επίγειας ψηφιακής μετάδοσης τηλεοπτικού σήματος είναι το πρότυπο DVB-T. 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^[7], 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.

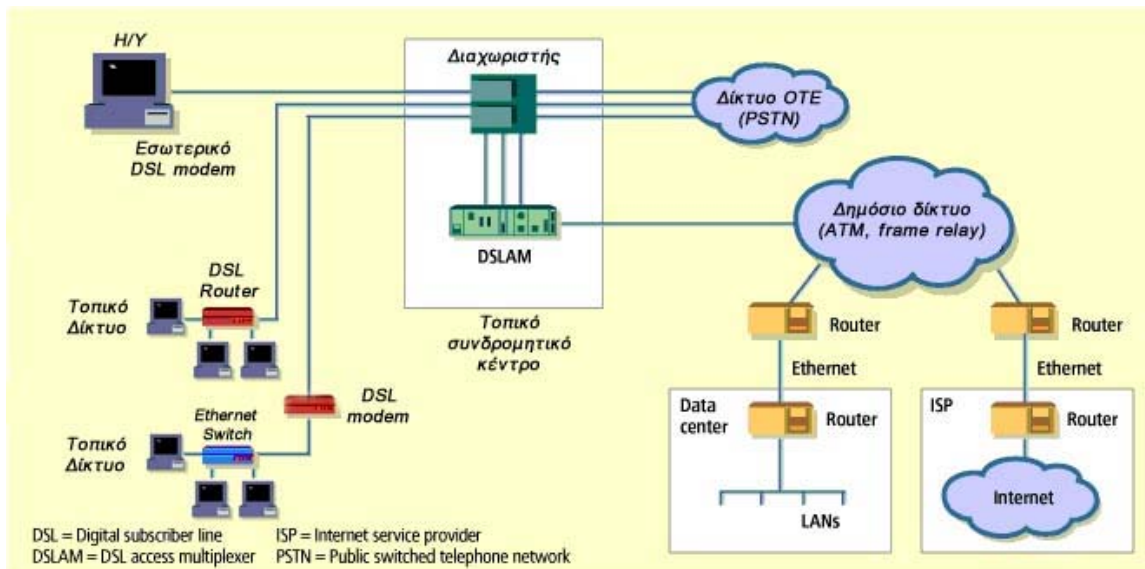


Σχήμα 1 : Διάγραμμα ενός DVB-T πομπού.

Το πρότυπο DVB-T προσδιορίζει ένα μονόδρομο σύστημα εκπομπής κατάλληλο για την μετάδοση υπηρεσιών, όπως IP-TV και IP-Radio, χωρίς να παρέχει διαδραστικότητα. Για την επίτευξη διαδραστικότητας και αμφίδρομης επικοινωνίας απαιτείται η επέκταση του συστήματος προσθέτοντας ένα κανάλι επιστροφής DVB-RCT (Return Channel Terrestrial), ώστε να είναι δυνατή η επικοινωνία και η μεταφορά δεδομένων από τον χρήστη προς τον παροχέα υπηρεσιών. Για την υλοποίηση του καναλιού επιστροφής μπορούν να χρησιμοποιηθούν διάφορες από τις σημερινές διαθέσιμες τεχνολογίες, όπως xDSL, ISDN, PSTN, WLAN, GPRS, UMTS. Η ύπαρξη του καναλιού επιστροφής ανοίγει τον δρόμο στην παροχή περισσότερων υπηρεσιών, όπως VoIP, Internet, Email και κατ' απαίτηση οπτικοακουστικού υλικού, κάνοντας χρήση του DVB-T καναλιού επικοινωνίας.

2.2 xDSL

Ο όρος xDSL (Ψηφιακή Συνδρομητική Γραμμή - Digital Subscriber Line) περιγράφει μια οικογένεια τεχνολογιών που παρέχουν ψηφιακή μετάδοση δεδομένων πάνω από το παραδοσιακό τηλεφωνικό δίκτυο, πετυχαίνοντας υψηλούς ρυθμούς μετάδοσης. Ο λόγος που συμβαίνει αυτό είναι ότι χρησιμοποιούνται οι υψηλές συχνότητες του χάλκινου καλωδίου, που μένουν ανεκμετάλλευτες από την κλασσική τηλεφωνία. Δυστυχώς, οι συχνότητες αυτές εξασθενούν σύντομα, με αποτέλεσμα η συγκεκριμένη τεχνολογία να μπορεί να λειτουργήσει σε αποστάσεις έως 5 χμ. από το τηλεφωνικό κέντρο. Η πιο δημοφιλής τεχνολογία DSL είναι η ADSL (Asymmetric DSL). Χαρακτηριστικό του ADSL είναι το ότι οι ταχύτητες λήψης και αποστολής δεδομένων διαφέρουν - σε αυτό οφείλει και τη λέξη “ασύμμετρη” στο όνομά του. Για παράδειγμα, μια τυπική ταχύτητα για ADSL στην Ελλάδα είναι η 1024/256 Kbps. Εξελιγμένες εκδόσεις του ADSL είναι το ADSL2 και το ADSL2+, οι οποίες παρέχουν μεγαλύτερες ταχύτητες αξιοποιώντας ακόμα περισσότερες συχνότητες του καλωδίου. Η μέγιστη ταχύτητα που μπορεί να επιτύχει το ADSL2+ είναι τα 24/1 Mbps (ή τα 24/3,5 Mbps σε περίπτωση που υλοποιεί το πρότυπο ITU G.992.5 Annex M).



Σχήμα 2 : Αρχιτεκτονική xDSL δικτύου.

2.3 WLAN

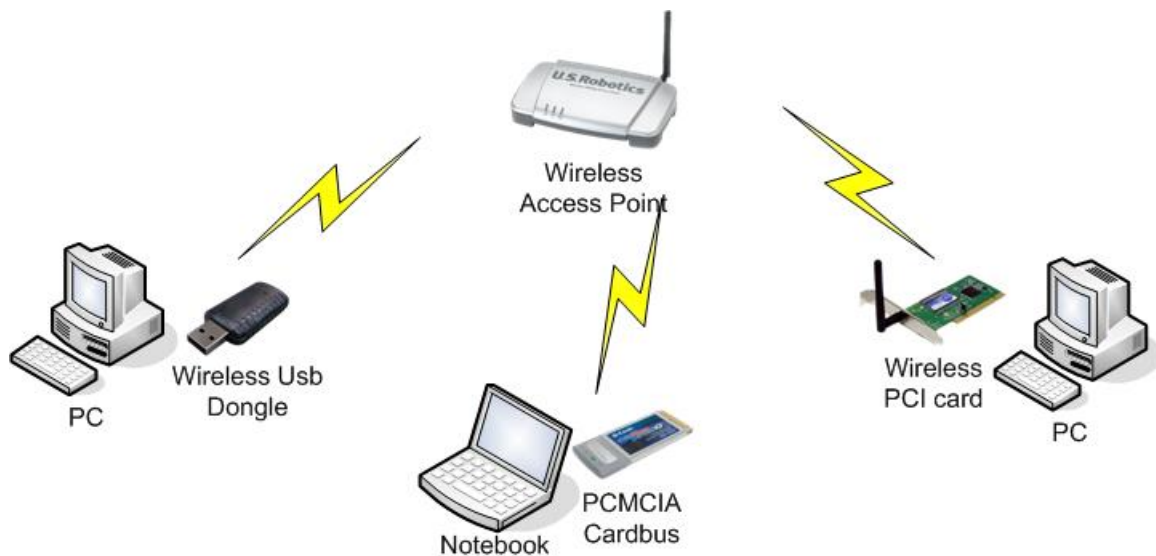
Η τεχνολογία των ασύρματων τοπικών δικτύων (Wireless Local Area Network - WLAN) είναι ένα είδος τοπικού δικτύου που χρησιμοποιεί ραδιοκύματα υψηλής συχνότητας αντί καλωδίων για την σύνδεση και επικοινωνία δύο ή περισσότερων υπολογιστών. Κάνει χρήση των τεχνολογιών διαμόρφωσης της φασματικής εξάπλωσης (spread spectrum) ή της ορθογωνικής πολύπλεξης διαίρεσης συχνότητας (Orthogonal Frequency Division Multiplex – OFDM). Η ομάδα εργασίας του IEEE^[8], που έχει αναλάβει την ανάπτυξη των πρωτοκόλλων που σχετίζονται με τα ασύρματα δίκτυα έχει ορίσει ένα σύνολο πρωτοκόλλων με την ονομασία IEEE 802.11x.

Πρωτόκολλο	Συχνότητα Λειτουργίας	Throughput	Ρυθμός Μετάδοσης	Εμβέλεια (Εσωτερικοί χώροι)	Εμβέλεια (Εξωτερικοί χώροι)
Legacy 802.11	2.4-2.5 GHz	0.7 Mbps	2 Mbps	~25 μέτρα	~75 μέτρα
802.11a	5.15-5.25 GHz, 5.25-5.35 GHz, 5.725-5.875 GHz	23 Mbps	54 Mbps	~30 μέτρα	~100 μέτρα
802.11b	2.4-2.5 GHz	4 Mbps	11 Mbps	~35 μέτρα	~110 μέτρα
802.11g	2.4-2.5 GHz	19 Mbps	54 Mbps	~35 μέτρα	~110 μέτρα
802.11n	2.4 GHz and/or 5 GHz	74 Mbps	248 Mbps = 2x2 ant	~70 μέτρα	~160 μέτρα

Πίνακας 1 : Συνοπτική περιγραφή των 802.11 πρωτοκόλλων.

Το IEEE 802.11 υποστηρίζει ρυθμούς μετάδοσης έως 2Mbps, ενώ η έκδοση IEEE 802.11b (γνωστή ως 802.11 High Rate ή Wi-Fi) υποστηρίζει ρυθμούς μέχρι και 11Mbps. Στην οικογένεια των προτύπων IEEE 802.11x ανήκουν δύο ακόμα πρότυπα τα IEEE 802.11a και IEEE 802.11g (54Mbps). Τμήμα του 802.11b αποτελεί το WEP (Wired Equivalent Privacy, μυστικότητα αντίστοιχη με τα καλωδιωμένα δίκτυα), το οποίο κάνει χρήση του αλγορίθμου RC4 προσφέροντας δυνατότητα εξουσιοδότησης

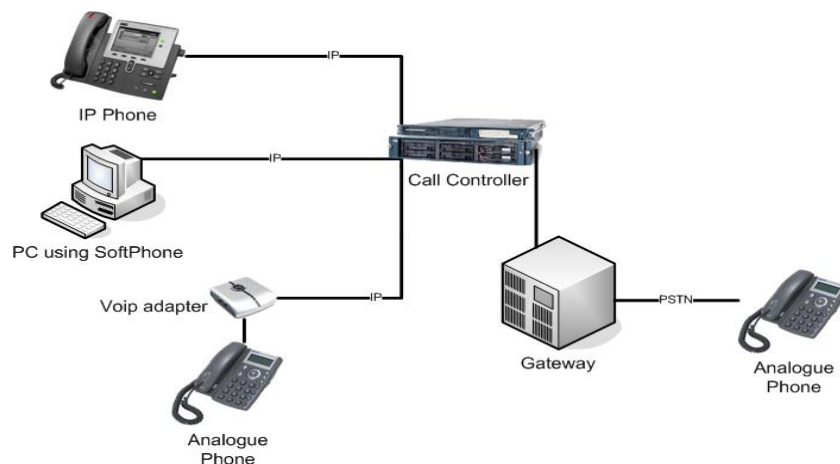
κάθε κόμβου και κρυπτογράφησης των δεδομένων. Οι προδιαγραφές για τη ραδιομετάδοση καθορίζουν λειτουργία μέσα στη ζώνη συχνοτήτων 2.4 GHz ISM (Industrial, Scientific, Medical). Τόσο το IEEE 802.11 όσο και το IEEE 802.11b χρησιμοποιούν ραδιοκυματικές τεχνικές ευρέως φάσματος (spread spectrum), οι οποίες αυξάνουν την αξιοπιστία του συστήματος, προωθούν το ρυθμό απόδοσης του δικτύου και περιορίζουν τις παρεμβολές. Το τρίτο πρότυπο που αναπτύχθηκε είναι το 802.11g, το οποίο χρησιμοποιεί την OFDM διαμόρφωση και έχει συμβατότητα με το 802.11b λειτουργώντας όμως καλύτερα λόγω των ομοιοτήτων του με το 802.11a. Τέλος, αναμένεται μέσα στο 2008 η επίσημη κυκλοφορία του 802.11n, το οποίο θα προσφέρει υψηλότερους ρυθμούς μετάδοσης από τα παλαιότερα πρωτόκολλα φτάνοντας στα 248 Mbps. Η χρήση του προτύπου 802.11b ή του 802.11g για την υλοποίηση μιας αμφίδρομης επικοινωνίας καθιστά το χρήστη ικανό να έχει ευρυζωνική πρόσβαση στις διαδραστικές υπηρεσίες που επιθυμεί, ακόμα και όταν βρίσκεται σε κίνηση. Επιπροσθέτως, τα χαρακτηριστικά (υψηλοί ρυθμοί μετάδοσης και χρήση τεχνικών spread spectrum και OFDM) των προτύπων 802.11b και 802.11g δημιουργούν ένα δίκτυο στο οποίο το κανάλι επιστροφής έχει συγκρίσιμη απόδοση με το ευρυζωνικό κανάλι προώθησης.



Σχήμα 3 : Αρχιτεκτονική WLAN δικτύου.

2.4 VOIP

Η τεχνολογία VoIP (Voice over IP) είναι η μετάδοση της φωνής κάνοντας χρήση των IP δικτύων και του Διαδικτύου, αντί των “παραδοσιακών” τηλεφωνικών δικτύων. Βασίζεται στη μετατροπή της αναλογικής φωνής σε ψηφιακή μορφή (Analog to Digital Conversion - ADC) και στο κατακερματισμό του ψηφιακού σήματος της φωνής σε πακέτα κατάλληλου μεγέθους. Τα πακέτα αυτά στη συνέχεια μεταδίδονται μέσω του Διαδικτύου με τη χρήση ειδικού πρωτοκόλλου για τη μετάδοση δεδομένων πραγματικού χρόνου (Real Time Protocol - RTP) μαζί με την απαραίτητη σηματοδότηση. Στον αποδέκτη με μια αντίστροφη διαδικασία επαναδημιουργείται το αναλογικό σήμα (Digital to Analog Conversion - DAC) καταληπτό από το ανθρώπινο αυτί. Το μεγαλύτερο πλεονέκτημα του VoIP είναι το μειωμένο κόστος. Οι υπηρεσίες VoIP είναι πολύ φθηνότερες από τις παραδοσιακές επίγειες υπηρεσίες και, σε ορισμένες περιπτώσεις, ακόμα και δωρεάν. Άλλο μεγάλο πλεονέκτημα του VoIP είναι η φορητότητα του, καθώς χρησιμοποιεί το παγκόσμιο δίκτυο του Διαδικτύου και έτσι οι χρήστες δεν δεσμεύονται με κάποια συγκεκριμένη τοποθεσία. Αρκεί κάποιος να έχει υπολογιστή, ευρυζωνική σύνδεση και, σε ορισμένες περιπτώσεις, έναν προσαρμογέα τηλεφώνου, για να μπορεί να κάνει κλήσεις χρησιμοποιώντας το λογαριασμό VoIP. Η σύνδεση σε ένα VoIP δίκτυο επιτυγχάνεται με την χρήση ενός ηλεκτρονικού υπολογιστή που διαθέτει μικρόφωνο και ηχεία, με την χρήση ειδικής τηλεφωνικής συσκευής (VoIP phones) ή χρησιμοποιώντας ένα κοινό τηλέφωνο που συνδέεται με ένα VoIP προσαρμογέα.



Σχήμα 4 : Γενική αρχιτεκτονική ενός VoIP δικτύου.

2.5 Αρχιτεκτονική Δικτύου Διαδραστικής Ψηφιακής Τηλεόρασης

2.5.1 Εισαγωγή

Μια πλατφόρμα επίγειας ψηφιακής τηλεόρασης είναι δυνατόν να παρέχει εκτός από ψηφιακά τηλεοπτικά προγράμματα MPEG-2 και υπηρεσίες IP. Το ενδογενές αυτό χαρακτηριστικό σε συνδυασμό με την ύπαρξη καναλιών επιστροφής (Return Channels), επιτρέπει την υλοποίηση ευρυζωνικών υποδομών για την παροχή αμφίδρομων διαδραστικών υπηρεσιών. Μερικές από τις υπηρεσίες που παρέχονται από ένα σύστημα DVB-T είναι :

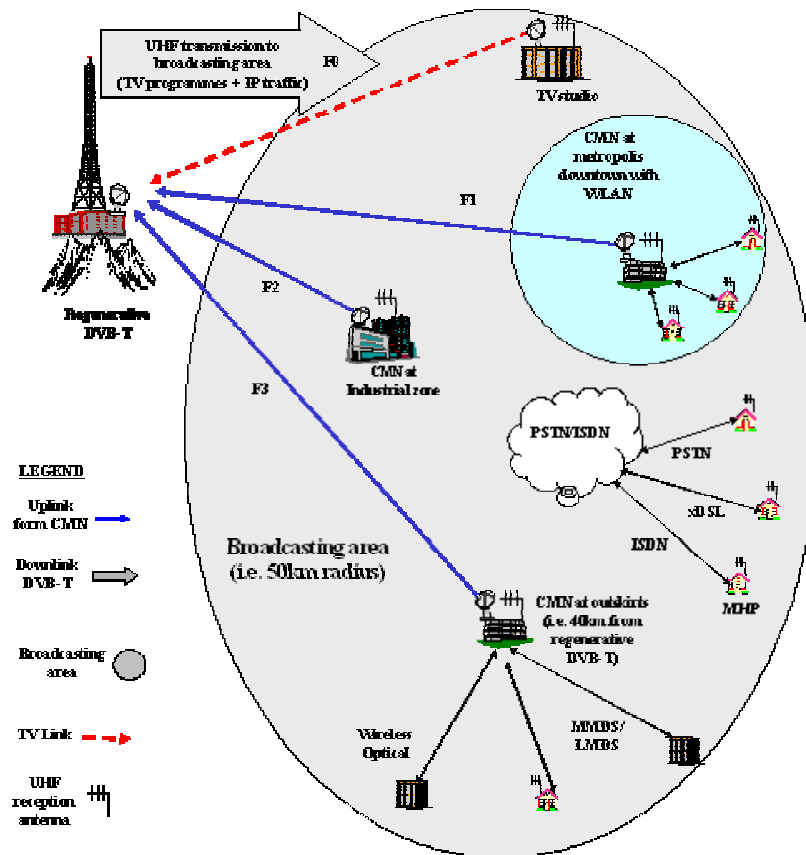
- Ψηφιακά τηλεοπτικά προγράμματα MPEG-2.
- Υπηρεσίες IP-TV και IP-Radio.
- Πρόσβαση στο Διαδίκτυο.
- Υπηρεσίες Video On Demand και Audio On Demand.
- VoIP (Voice over IP) τηλεφωνία.

2.5.2 Το δίκτυο ATHENA

Λαμβάνοντας υπ' όψιν την τοπική και δικτυακή διάσταση της νέας επίγειας ψηφιακής τηλεόρασης (DVB-T), το Ευρωπαϊκό ερευνητικό έργο ATHENA IST FP6-507312^[9] (www.ist-athena.org) προτείνει την υιοθέτηση της ψηφιακής μετάβασης στα UHF αξιοποιώντας τη ροή του DVB-T σε αναγεννητικούς σχηματισμούς για την υλοποίηση μίας ευρυζωνικής υποδομής πρόσβασης και τη δημιουργία ενός κοινού καναλιού μεταφοράς κίνησης IP διαθέσιμο σε όλη την περιοχή ευρυεκπομπής. Η συνολική αρχιτεκτονική αποτελείται από δύο υποσυστήματα :

1. Από ένα Κεντρικό Σημείο Ευρυεκπομπής (πλατφόρμα DVB-T).
2. Από ένα σύνολο Ενδιάμεσων Κόμβων Διανομής (Cell Main Nodes – CMN's).

Η DVB-T περιοχή κάλυψης διαιρείται σε κυψέλες, κάθε μία εκ' των οποίων εξυπηρετείται από ένα CMN, ο οποίος με την σειρά του εξυπηρετεί κάποιον αριθμό χρηστών, παρέχοντας συνδεσιμότητα με χρήση διάφορων τεχνολογιών. Η αμφίδρομη επικοινωνία μεταξύ κάθε CMN και του κεντρικού σημείου ευρυεκπομπής επιτυγχάνεται με τη χρήση δύο ξεχωριστών μονόδρομων καναλιών επικοινωνίας.



Σχήμα 5 : Γενική αρχιτεκτονική ATHENA IST FP6-507312.

Η IP κίνηση ενθυλακώνεται στο ρεύμα μεταφοράς DVB-T MPEG-2 χρησιμοποιώντας την τεχνική Ενθυλάκωσης Πολλαπλών Πρωτοκόλλων (Multi Protocol Encapsulation – MPE) παρέχοντας έτσι ένα ιδεατό δίκτυο κορμού ETHERNET για τις IP υπηρεσίες. Ακολουθεί η πολυπλεξία της IP κίνησης και των τηλεοπτικών προγραμμάτων σε μια ροή DVB-T, η οποία με την σειρά της διαμορφώνεται χρησιμοποιώντας Κωδικοποιημένη Ορθογώνικη Πολύπλεξη Συχνότητας (COFDM) και εκπέμπεται μέσω ενός καναλιού UHF .

2.6 Ποιότητα Υπηρεσίας

Ο όρος Ποιότητα Υπηρεσίας (Quality of Service - QoS)^[10] αναφέρεται στο τρόπο με τον οποίο ένα δίκτυο παρέχει εγγυήσεις στην εξυπηρέτηση μιας δικτυακής κίνησης, κάνοντας αποδοτική χρήση των διαθέσιμων δικτυακών πόρων. Η Ποιότητα Υπηρεσίας επιτυγχάνεται με την χρήση διάφορων μηχανισμών που ρυθμίζουν την κίνηση ενός καναλιού μετάδοσης, δεσμεύουν το κατάλληλο εύρος ζώνης και διαχειρίζονται τους πόρους του δικτύου ανάλογα με τις απαιτήσεις των διαφόρων χρηστών ή ροών δεδομένων, έτσι ώστε τα χαρακτηριστικά που επηρεάζουν την ποιότητα να διατηρούνται σε αποδεκτές τιμές.

Η Ποιότητα Υπηρεσίας έχει ιδιαίτερο ενδιαφέρον για την αδιάλειπτη μετάδοση βίντεο, φωνής και πολυμεσικής πληροφορίας. Είναι δεδομένα που απαιτούν μεγάλο εύρος ζώνης και είναι δύσκολη η μετάδοσή τους με τη χρήση “best effort” πρωτοκόλλων, όπως το κλασσικό IP. Για παράδειγμα μια ροή οπτικοακουστικής πληροφορίας θα πρέπει να μεταδοθεί δεσμεύοντας μεγαλύτερο εύρος ζώνης και έχοντας υψηλή προτεραιότητα σε αντίθεση με μια ροή δεδομένων που δεν έχει ιδιαίτερες απαιτήσεις στην μετάδοση της.

Τα χαρακτηριστικά ενός δικτύου που επηρεάζουν την ποιότητα των υπηρεσιών είναι:

- Η καθυστέρηση (delay – latency), ο χρόνος μετάβασης ενός πακέτου από τον αποστολέα στον παραλήπτη.
- Η διακύμανση (jitter) στην καθυστέρηση.
- Οι απώλειες (losses).
- Το διαθέσιμο εύρος ζώνης (bandwidth).

2.6.1 Τεχνολογίες παροχής Ποιότητας Υπηρεσίας

Τα σημερινά IP δίκτυα αναβαθμίζονται από το “best effort” μοντέλο μετάδοσης σε κάποιο μοντέλο που θα μπορεί να παρέχει διαφορετικά επίπεδα υπηρεσίας για συγκεκριμένες απαιτήσεις Ποιότητας Υπηρεσίας. Στο “best effort” μοντέλο η κίνηση μεταδίδεται όσο πιο γρήγορα επιτρέπει το δίκτυο και όλες οι ροές αντιμετωπίζονται κατά τον ίδιο τρόπο από αυτό, αλλά δεν παρέχεται καμία εγγύηση για την Ποιότητα Υπηρεσίας. Για να καταστεί εφαρμόσιμη η Ποιότητας Υπηρεσίας, ο οργανισμός Internet Engineering Task Force (IETF)^[11] ανέπτυξε τρεις μηχανισμούς και αρχιτεκτονικές που παρέχουν διαφοροποίηση υπηρεσιών και αναλαμβάνουν να επιληφθούν των ζητημάτων της δέσμευσης πόρων και της βελτιστοποίησης της απόδοσης ενός δικτύου. Οι μηχανισμοί αυτοί είναι : οι Διαφοροποιημένες Υπηρεσίες^[4] (Differentiated Services – DiffServ - *RFC 2475, 2474*), οι Ενοποιημένες Υπηρεσίες^[12] (Integrated Services – IntServ - *RFC 1633*) σε συνδυασμό με το Πρωτόκολλο Δέσμευσης Πόρων^[13] (Resource ReserVation Protocol – RSVP – *RFC 2205*) και η Μεταγωγή Επισημάτων Πολλών Πρωτοκόλλων^[14] (Multi-Protocol Label Switching – MPLS – *RFC 3031*).

Η τεχνολογία των Ενοποιημένων Υπηρεσιών είναι αρκετά πολύπλοκη και σύνθετη σε σύγκριση με τις άλλες δύο τεχνολογίες . Παρέχει απόλυτες εγγυήσεις όσον αφορά το εύρος ζώνης και τις καθυστερήσεις για μεμονωμένες ροές, καθώς το Πρωτόκολλο Δέσμευσης Πόρων αναλαμβάνει την αξιόπιστη μετάδοση μέσα από το κανάλι επικοινωνίας.

Αντίθετα με τις Ενοποιημένες Υπηρεσίες, ο μηχανισμός των Διαφοροποιημένων Υπηρεσιών είναι πιο απλός, προσφέρει επεκτασιμότητα και δεν απαιτεί εκ’ των προτέρων δέσμευση πόρων ή κάποιο ειδικό εξοπλισμό. Εγγυείται για την παρεχόμενη υπηρεσία στην κρίσιμη δικτυακή κίνηση, όπως την ροή οπτικοακουστικής πληροφορίας και συγχρόνως παρέχει “best effort” μετάδοση σε υπηρεσίες χωρίς ιδιαίτερες απαιτήσεις.

Τέλος η τεχνολογία της Μεταγωγής Επισημάτων Πολλών Πρωτοκόλλων προσβλέπει στον αποδοτικό προσδιορισμό, δρομολόγηση, προώθηση και μεταγωγή της ροής της κίνησης μέσα στο δίκτυο, συνδυάζοντας την μεταγωγή με ετικέτα (label) και την παραδοσιακή δρομολόγηση του IP. Η λειτουργία της στηρίζεται στην δημιουργία κυκλωμάτων μεταγωγής με προκαθορισμένα χαρακτηριστικά, όπως εύρος ζώνης. Το

μειονέκτημα της συγκεκριμένης τεχνολογίας έγκειται στο γεγονός ότι για την υλοποίηση της απαιτεί μεγάλες δαπάνες στην απόκτηση ειδικού εξοπλισμού.

2.7 Διαφοροποιημένες Υπηρεσίες (Differentiated Services – DiffServ)

2.7.1 Εισαγωγή

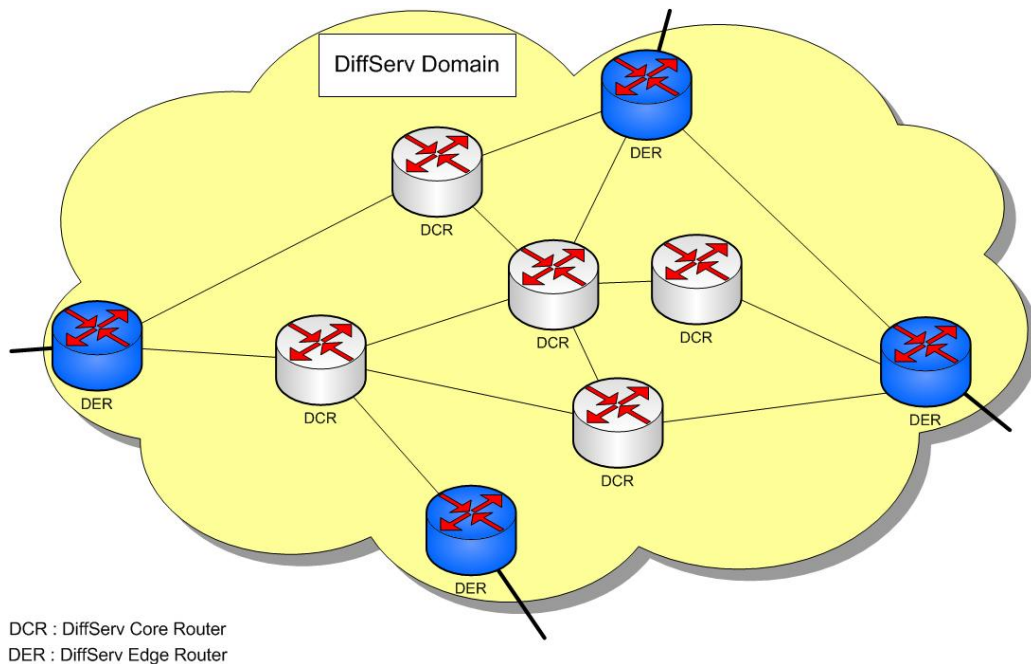
Τα σύγχρονα δίκτυα δεδομένων μεταφέρουν πολλούς και διαφορετικούς τύπους υπηρεσιών, συμπεριλαμβανομένης της φωνής, του βίντεο, την σε πραγματικό χρόνο μετάδοση ροών πολυμεσικής πληροφορίας, του ηλεκτρονικού ταχυδρομείου και την μεταφορά αρχείων. Πολλοί από τους προτεινόμενους μηχανισμούς Ποιότητας Υπηρεσίας, που επέτρεψαν σε αυτές τις υπηρεσίες να συνυπάρξουν, ήταν πολύπλοκοι και απέτυχαν να αντεπεξέλθουν στις απαιτήσεις του Διαδικτύου. Το 1998, ο IETF δημοσίευσε το RFC 2475 με τίτλο “Αρχιτεκτονική για τις Διαφοροποιημένες Υπηρεσίες”, όπου ορίζει ένα νέο μηχανισμό παροχής Ποιότητας Υπηρεσίας. Σήμερα, οι Διαφοροποιημένες Υπηρεσίες έχουν αντικαταστήσει κατά ένα μεγάλο μέρος άλλους μηχανισμούς παροχής Ποιότητας Υπηρεσίας, όπως τις Ενοποιημένες Υπηρεσίες, ως το κυριότερο μέσο για την διασφάλιση της Ποιότητας Υπηρεσίας στα δίκτυα υπολογιστών. Οι Διαφοροποιημένες Υπηρεσίες είναι μια τεχνολογία δικτύωσης υπολογιστικών συστημάτων που στοχεύει στην διαφοροποίηση, ρύθμιση και διαχείριση της δικτυακής κίνησης και στην παροχή και διασφάλιση της Ποιότητας Υπηρεσιών στα μοντέρνα IP δίκτυα. Λειτουργεί βασισόμενη στην αρχή της Ταξινόμησης Κίνησης, κατά την οποία η δικτυακή κίνηση κατηγοριοποιείται σε ένα συγκεκριμένο αριθμό συνόλων, που ονομάζονται σύνολα συμπεριφοράς. Κάθε σύνολο συμπεριφοράς μπορεί να αποτελείται από διάφορες και μεταξύ τους ανεξάρτητες ροές που απαιτούν όμως την ίδια ποιότητα στην παροχή υπηρεσιών από το δίκτυο. Οι προϋποθέσεις που πρέπει να πληροί μια ροή για να ταξινομηθεί προσδιορίζονται σε μια συμφωνία μεταξύ του παροχέα της υπηρεσίας και του πελάτη – χρήστη, την Συμφωνία Στάθμης (παρεχόμενης) Υπηρεσίας (Service Level Agreement - SLA). Η συμφωνία αυτή μπορεί να περιέχει και λεπτομερείς κανόνες

ρύθμισης της κίνησης, οι οποίοι με την σειρά τους συντάσσουν ένα Συμφωνητικό Ρύθμισης Κίνησης (Traffic Conditioning Agreement – TCA). Το Συμφωνητικό Ρύθμισης Κίνησης προσδιορίζει πότε η κίνηση είναι εντός των συνθηκών (in profile), πότε εκτός συνθηκών (out-of-profile) και τι ενέργειες πρέπει να ληφθούν ώστε να ταξινομηθεί και να ρυθμιστεί αναλόγως.

2.7.2 Αρχιτεκτονική Διαφοροποιημένων Υπηρεσιών

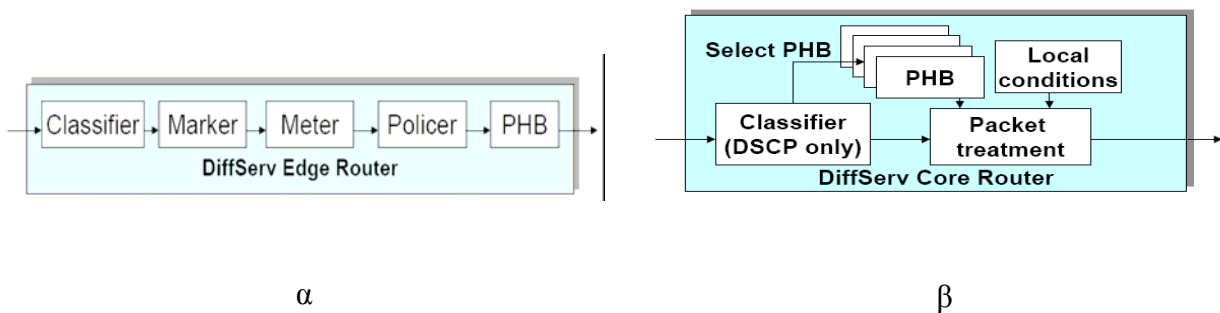
Η αρχιτεκτονική των Διαφοροποιημένων Υπηρεσιών είναι βασισμένη σε ένα απλό μοντέλο όπου η κίνηση που εισέρχεται σε ένα δίκτυο που υλοποιεί διαφοροποίηση υπηρεσιών (DiffServ Domain), κατηγοριοποιείται, ρυθμίζεται και ταξινομείται στα διάφορα σύνολα συμπεριφοράς (Behavior Aggregates – BA). Οι ενέργειες αυτές πραγματοποιούνται στα όρια του δικτύου από τους Δρομολογητές Παρυφής (DiffServ Edge Routers - DER). Πιο αναλυτικά, οι λειτουργίες που εκτελεί ένας Δρομολογητής Παρυφής είναι :

1. Ταξινόμηση της δικτυακής κίνησης : Η ταξινόμηση επιτυγχάνεται μαρκάροντας κατάλληλα το Κωδικοσημείο Διαφοροποιημένων Υπηρεσιών (Differentiated Services CodePoint – DSCP) κάθε IP πακέτου. Τα IP πακέτα που φέρουν την ίδια τιμή στο Κωδικοσημείο ανήκουν στο ίδιο σύνολο συμπεριφοράς.
2. Μέτρηση, αστυνόμευση και διαμόρφωση της εισερχόμενης κίνησης με τέτοιο τρόπο ώστε να παρέχεται εγγυημένη Ποιότητα Υπηρεσίας στα σύνολα ροών.



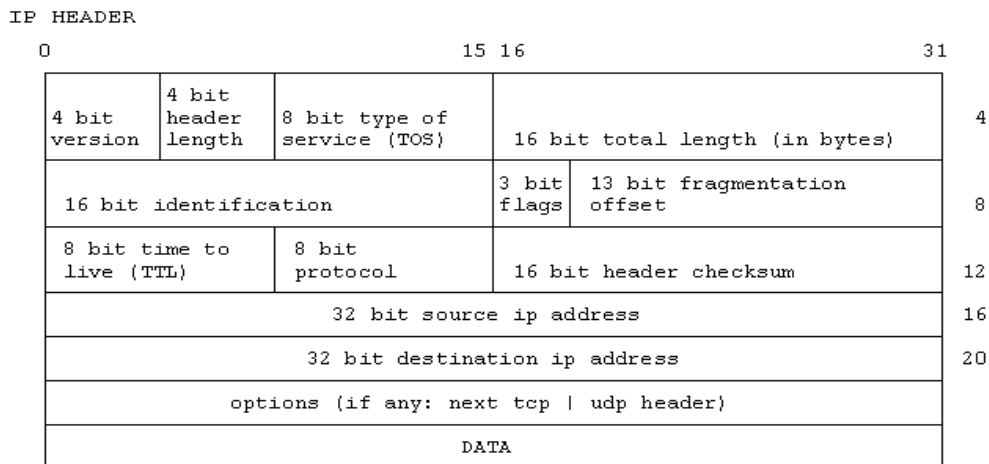
Σχήμα 6 : Σχηματική αναπαράσταση μιας περιοχής Διαφοροποιημένων Υπηρεσιών.

Μέσα στο δίκτυο τα πακέτα διαβιβάζονται στον προορισμό τους από τους Δρομολογητές Πυρήνα (DiffServ Core Routers - DCR), οι οποίοι ελέγχουν την τιμή του DSCP κάθε πακέτου και το προωθούν σύμφωνα με τους κανόνες που διέπουν το σύνολο συμπεριφοράς στο οποίο ανήκουν (Per-Hop Behavior - PHB)^[15]. Για να επιτευχθεί το επιθυμητό επίπεδο ποιότητας εφαρμόζονται διάφορες μέθοδοι διαχείρισης ουρών (Queuing Disciplines), πιθανό επαναπροσδιορισμό (remarking) της τιμής του DSCP και συντονισμό (scheduling) της κίνησης.

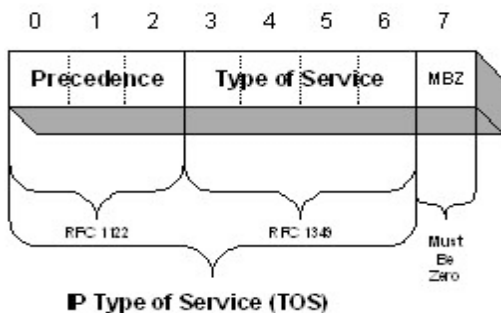


Σχήμα 7 : Ενέργειες για την εφαρμογή DiffServ σε : α)DER, β)DCR.

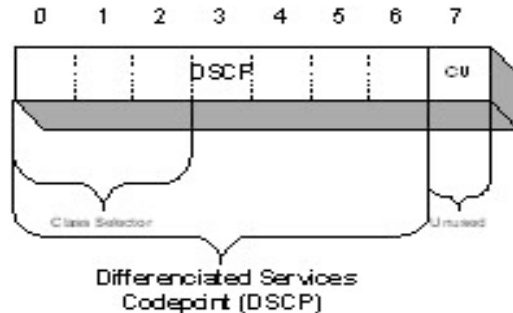
Κάθε σύνολο συμπεριφοράς προσδιορίζεται από την τιμή ενός πεδίου των 8 bits, που υπάρχει μέσα στην επικεφαλίδα των IP πακέτων (IP header). Οι σχεδιαστές των Διαφοροποιημένων Υπηρεσιών αποφάσισαν να χρησιμοποιήσουν την δεύτερη οκτάδα bits της IP επικεφαλίδας μετονομάζοντας την από πεδίο ToS (Type of Service Field) σε DS πεδίο (Differentiated Services Field) [16]. Για την διαφοροποίηση των υπηρεσιών γίνεται χρήση μόνο των 6 πρώτων bits (από αριστερά) από τα διαθέσιμα 8 του πεδίου, που ονομάζονται Κωδικοσημείο Διαφοροποιημένων Υπηρεσιών (Differentiated Services CodePoint – DSCP), αφήνοντας τα δύο τελευταία ανεκμετάλλευτα. Θεωρητικά, ένα δίκτυο θα μπορούσε να έχει μέχρι 64 (2^6) διαφορετικά σύνολα συμπεριφοράς χρησιμοποιώντας όλες τις δυνατές τιμές που μπορεί να πάρει το DSCP. Η δομή της IP επικεφαλίδας και των πεδίων DS και ToS φαίνεται στο παρακάτω σχήμα :



α



β



γ

Σχήμα 8 : α) Η επικεφαλίδα των IP πακέτων, β) το πεδίο ToS, γ) το πεδίο DSCP.

2.7.3 Per Hop Behavior

Όπως αναφέρθηκε παραπάνω, η Per Hop Behavior (PHB) καθορίζεται από την τιμή που φέρει το DSCP του κάθε πακέτου και υποδεικνύει τον τρόπο με τον οποίο αυτό θα προωθηθεί από τους δρομολογητές πυρήνα του δικτύου. Ο οργανισμός IETF έχει τυποποιήσει και προτείνει τρεις κατηγορίες PHB :

1. Την Εσπευσμένη Προώθηση (Expedited Forwarding – EF – *RFC 3246*)^[17] που παρέχει υψηλή ποιότητα μετάδοσης, προωθώντας τα πακέτα με μικρή καθυστέρηση, χαμηλό jitter, ελάχιστες απώλειες και εξασφαλισμένο εύρος ζώνης. Η τιμή του DSCP της EF κατηγορίας είναι : 101110
2. Την Εξασφαλισμένη Προώθηση (Assured Forwarding – AF – *RFC 2597*)^[18], η οποία προσφέρει την δυνατότητα ταξινόμησης των υπηρεσιών σε τέσσερις υποκατηγορίες (κλάσεις) με διαφορετική προτεραιότητα διαβίβασης. Στις τρεις πρώτες κλάσεις εφαρμόζεται το “Ολυμπιακό” πρότυπο για τον διαχωρισμό των υπηρεσιών σε : χρυσή, ασημένια και χάλκινη, παραχωρώντας την ανάλογη προτεραιότητα. Επιπρόσθετα κάθε κλάση ορίζεται από τρία ιεραρχικά επίπεδα απόρριψης πακέτων (drop precedence). Οι προτεινόμενες τιμές του DSCP των AF κλάσεων φαίνονται στον πίνακα :

	Class 1	Class 2	Class 3	Class 4
Low Drop Precedence	<i>AF11</i> 001010	<i>AF21</i> 010010	<i>AF31</i> 011010	<i>AF41</i> 100010
Medium Drop Precedence	<i>AF12</i> 001100	<i>AF22</i> 010100	<i>AF32</i> 011100	<i>AF42</i> 100100
High Drop Precedence	<i>AF13</i> 001110	<i>AF23</i> 010110	<i>AF33</i> 011110	<i>AF43</i> 100110

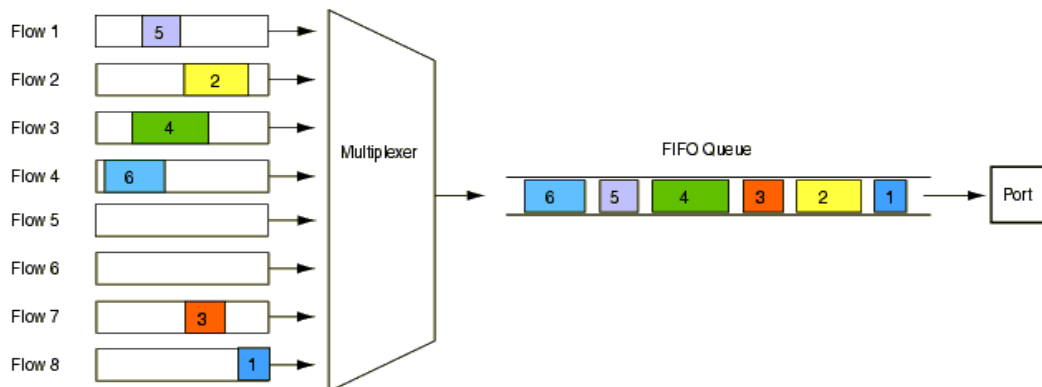
Πίνακας 2 : Παρουσίαση των τιμών του DSCP πεδίου των AF κλάσεων.

3. Την Προκαθορισμένη PHB (Default), που χρησιμοποιείται για την κίνηση βέλτιστης προσπάθειας (Best Effort – BE), η οποία δεν παρέχει εγγυήσεις για την ποιότητα της υπηρεσίας. Η τιμή του DSCP είναι : 000000.

2.8 Αλγόριθμοι Μορφοποίησης Κίνησης

2.8.1 FIFO

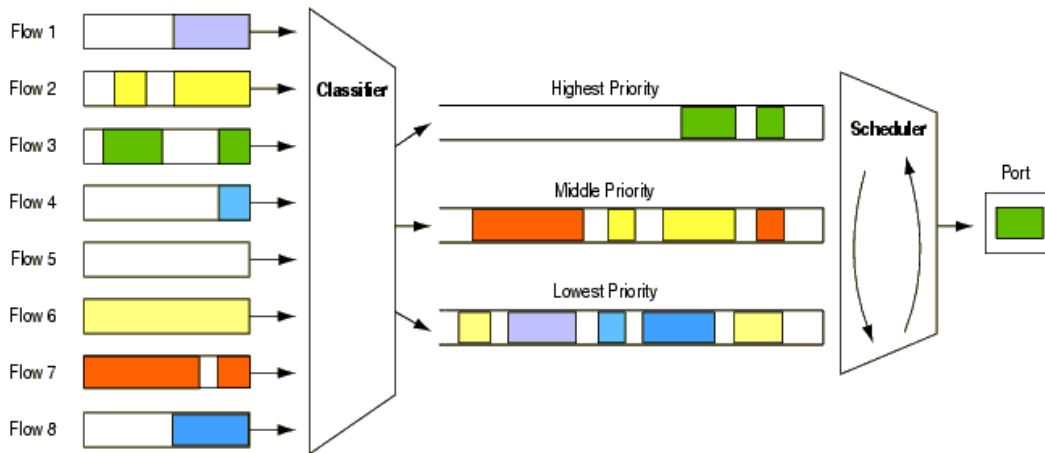
Ο αλγόριθμος FIFO (First In – First Out) είναι ο πιο απλός απ’ όσους μπορούν να εφαρμοστούν για την διαχείριση της κίνησης σε μια διεπαφή δικτύου. Η λειτουργία του είναι ιδιαίτερα απλή. Τα εισερχόμενα πακέτα τοποθετούνται κατά την άφιξη τους σε μια ουρά αναμονής συγκεκριμένου μεγέθους. Η εξυπηρέτηση των πακέτων γίνεται ακριβώς με την ίδια σειρά που τα πακέτα τοποθετήθηκαν στην ουρά, δηλαδή το πακέτο που έφτασε πρώτο θα προωθηθεί και πρώτο. Όταν η ουρά γεμίσει, τα πακέτα που καταφθάνουν θα απορρίπτονται (tail – drop).



Σχήμα 9 : Σχηματική περιγραφή του αλγόριθμου FIFO.

2.8.2 PRIO

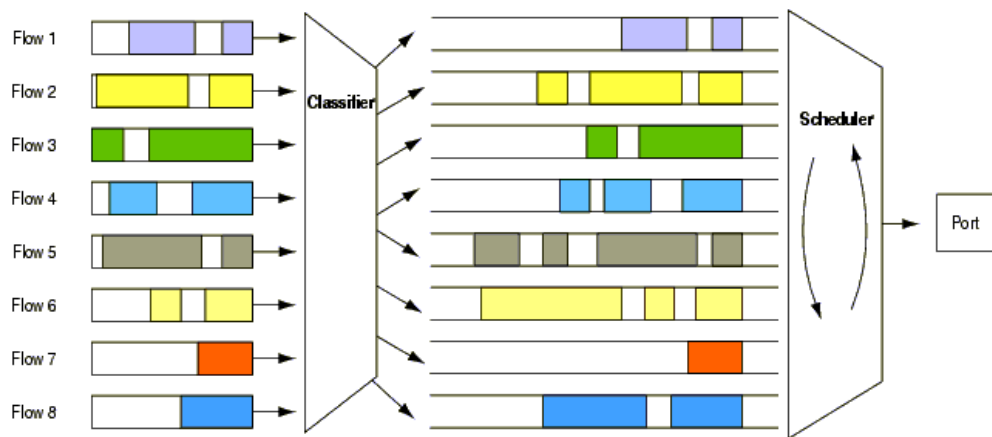
Μια απλή μέθοδος που υποστηρίζει διαφοροποίηση υπηρεσιών είναι ο αλγόριθμος PRIO (Priority Queuing). Σύμφωνα με τον αλγόριθμο τα πακέτα που καταφθάνουν ταξινομούνται από το σύστημα και τοποθετούνται σε ουρές διαφορετικής προτεραιότητας, σε κάθε μια από τις οποίες εφαρμόζεται ο αλγόριθμος FIFO. Για να εξυπηρετηθεί ένα πακέτο μια ουράς πρέπει όλες οι ουρές υψηλότερης προτεραιότητας να είναι κενές.



Σχήμα 10 : Σχηματική περιγραφή του αλγόριθμου PRIO.

2.8.3 SFQ

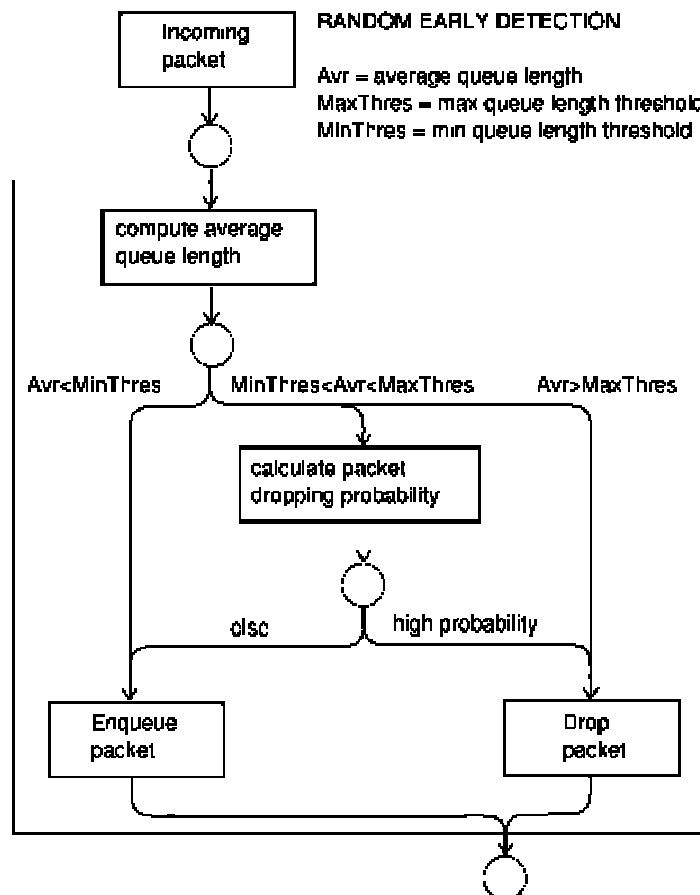
Ο αλγόριθμος SFQ (Stochastic Fair Queuing) είναι σχεδιασμένος με τέτοιο τρόπο ώστε να εξασφαλίζει κάθε ροή να έχει επαρκή πρόσβαση στους πόρους του δικτύου και παράλληλα να αποτρέπει μια καταγιστική ροή από την κατανάλωση περισσότερου εύρους ζώνης από αυτό που της αντιστοιχεί. Τα πακέτα αρχικά ταξινομούνται από το σύστημα σε ροές και στην συνέχεια εξυπηρετείται ένα πακέτο την φορά από κάθε ουρά με κυκλική σειρά, παραλείποντας τις κενές ουρές.



Σχήμα 11 : Σχηματική περιγραφή του αλγόριθμου SFQ.

2.8.4 RED

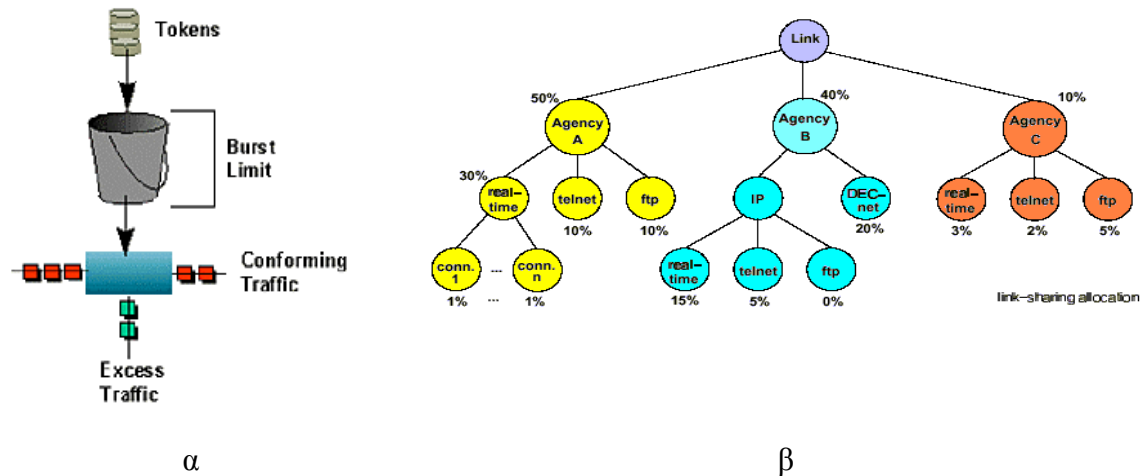
Μια εναλλακτική αντιμετώπιση της τεχνικής drop tail που εφαρμόζεται στην FIFO ουρά είναι ο αλγόριθμος RED (Random Early Detection). Σκοπός του είναι να μην αφήνει την FIFO ουρά να γεμίσει, απορρίπτοντας επιλεκτικά πακέτα όταν χρειάζεται. Με αυτό τον τρόπο αντιμετωπίζεται η συμφόρηση του δικτύου και το TCP πρωτόκολλο καταφέρνει να αποκτήσει γρηγορότερα τον κατάλληλο ρυθμό αποστολής δεδομένων. Μια πιο εξελιγμένη μορφή του RED είναι ο αλγόριθμος GRED (Generalized RED) που υποστηρίζει πολλαπλές προτεραιότητες απόρριψης, καθιστώντας τον κατάλληλο για εφαρμογή στις Διαφοροποιημένες Υπηρεσίες.



Σχήμα 12 : Σχηματική περιγραφή του αλγόριθμου RED.

2.8.5 HTB

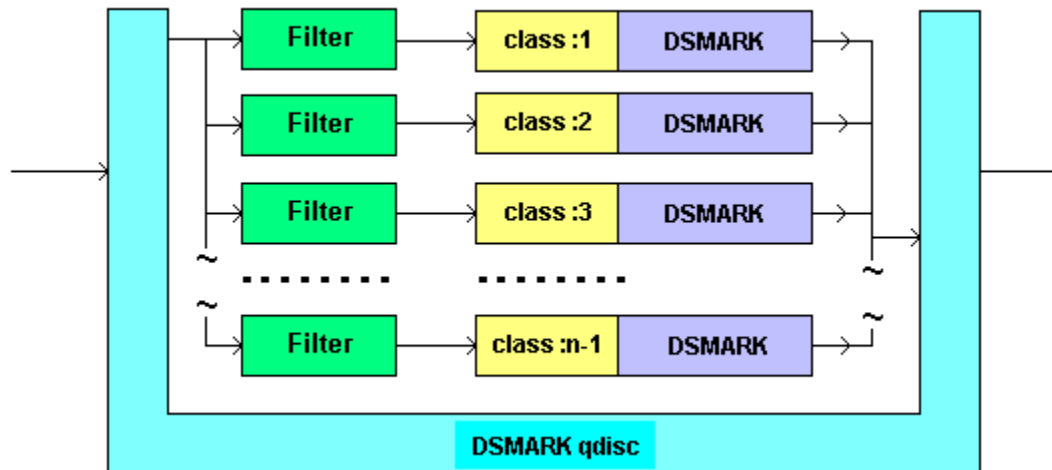
Ο αλγόριθμος HTB είναι μια εξελιγμένη μορφή του αλγόριθμου κάδου με σκυτάλη (Token Bucket algorithm). Χρησιμοποιείται, κυρίως, για την μορφοποίηση της δικτυακής κίνησης και τον περιορισμό του ρυθμού μετάδοσης δεδομένων επιβάλλοντας ένα μέσο ρυθμό αποστολής, επιτρέποντας όμως ελεγχόμενους καταγισμούς (bursts). Ο τρόπος λειτουργίας του αλγόριθμου κάδου με σκυτάλη μπορεί να περιγραφεί με την ύπαρξη ενός κάδου συγκεκριμένου μεγέθους, μέσα στον οποίο δημιουργούνται ανά τακτά χρονικά διαστήματα κάποιες “άδειες” (tokens), που αντιπροσωπεύουν το δικαίωμα αποστολής του πακέτου. Ένα πακέτο, για να μεταδοθεί, πρέπει να πάρει μια άδεια. Υπάρχουν, όμως, παραλλαγές του αλγόριθμου που επιτρέπουν περισσότερα από ένα πακέτα να αποσταλούν με μία άδεια. Τα πακέτα μεταδίδονται μέχρι να τελειώσουν οι άδειες που υπάρχουν μέσα στον κάδο. Η έννοια της ιεραρχίας του HTB εισέρχεται με την δυνατότητα του να λειτουργήσει ως εργαλείο καταμέρισης του εύρους ζώνης μιας ζεύξης, καθιστώντας δυνατή την διαφοροποίηση υπηρεσιών.



Σχήμα 13 : α) Περιγραφή της λειτουργίας του αλγόριθμου TBF, β) αναπαράσταση της διαμοίρασης του εύρους ζώνης.

2.8.6 DSMARK

Το DSMARK είναι ένας μηχανισμός που αναπτύχθηκε για να ολοκληρώσει τις απαιτήσεις της τεχνολογίας των Διαφοροποιημένων Υπηρεσιών. Αντίθετα με τους παραπάνω αλγόριθμους, το DSMARK δεν ελέγχει, αστυνομεύει ή μορφοποιεί την δικτυακή κίνηση. Δεν καθορίζει προτεραιότητες, δεν εισάγει καθυστερήσεις και δεν απορρίπτει πακέτα. Η μοναδική του λειτουργία είναι να μαρκάρει τα πακέτα, κάνοντας χρήση του πεδίου DS της επικεφαλίδας IP που αναφέρθηκε παραπάνω. Στο ακόλουθο σχεδιάγραμμα φαίνεται η θέση του DSMARK στην διαδικασία της υλοποίηση των Διαφοροποιημένων Υπηρεσιών.



Σχήμα 14 : Σχηματική περιγραφή του αλγόριθμου DSMARK.

2.8.7 Ταξινομητές

Τέλος, ένας σημαντικός παράγοντας για την αστυνόμευση και την κατηγοριοποίηση των πακέτων είναι οι ταξινομητές (classifiers). Η αστυνόμευση προϋποθέτει μέτρηση της δικτυακής κίνησης, ώστε να μπορεί να γίνει έλεγχος αν συμφωνεί με τους όρους που αναφέρονται στο συμβόλαιο για την παρεχόμενη ποιότητα υπηρεσίας. Μια από τις θεμελιώδεις αρχές της αρχιτεκτονικής των Διαφοροποιημένων Υπηρεσιών είναι να μην επιτρέπεται περισσότερη κίνηση από αυτή για την οποία

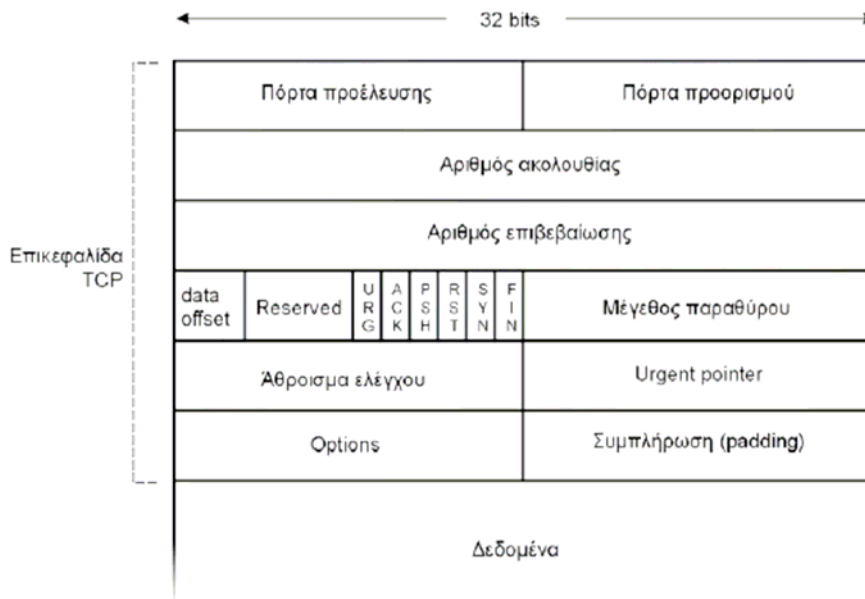
σχεδιάστηκε το δίκτυο, για να μην υπερφορτώνονται οι ουρές αναμονής. Οι ταξινομητές επιλέγουν, ελέγχουν και κάνουν κατηγοριοποίηση των ροών κίνησης με βάση τα κριτήρια που καθορίζονται από τις παραμέτρους τους. Μέσα στις αρμοδιότητες τους έγκειται και να αποφασίζουν για τις ενέργειες που θα ληφθούν όταν μια ροή δεν πληροί ή αντίθετα, πληροί τους κανόνες που προ-συμφωνήθηκαν. Κάποιοι από τους ταξινομητές που χρησιμοποιούνται είναι : ο fw, ο u32, ο route και ο tcindex.

2.9 Πρωτόκολλα Μεταφοράς Δεδομένων

2.9.1 Το Πρωτόκολλο TCP

Το πρωτόκολλο μεταφοράς TCP (Transmission Control Protocol)^[19] είναι σήμερα το πιο δημοφιλές ίσως πρότυπο στις επικοινωνίες δεδομένων μέσω του Internet και όχι μόνο. Σχεδιάστηκε για πρώτη φορά το 1981 για χρήση στο αμερικανικό DARPA για να εξασφαλίσει αξιόπιστη επικοινωνία μεταξύ δύο τερματικών. Από τότε, έχει γίνει αντικείμενο πάμπολλων ερευνητικών προσπαθειών και έχει υποστεί πολλές προσθήκες και βελτιώσεις. Πολλές από τις βελτιώσεις αυτές θα υιοθετηθούν στα πλαίσια αυτής της πτυχιακής για να μεγιστοποιηθεί η επίδοση του TCP σε ασύμμετρα δίκτυα DVB-T.

Το TCP παρέχει υπηρεσίες μεταφοράς δεδομένων με σύνδεση (connection-oriented) και έλεγχο ροής (flow control) χρησιμοποιώντας το IP ως επίπεδο δικτύου. Οργανώνει τα δεδομένα σε τμήματα (segments) με επικεφαλίδα, της οποίας τα πεδία διευκολύνουν τις λειτουργίες ελέγχου ροής και αποφυγής λαθών, όπως φαίνεται και από το σχήμα που ακολουθεί.



Σχήμα 15 : Η επικεφαλίδα TCP.

Πολλές νοητές παράλληλες συνδέσεις μπορούν να εγκατασταθούν στην ίδια φυσική διαδρομή IP χάρη στις νοητές «πόρτες» (ports) των οποίων οι αριθμοί δηλώνονται στην αρχή της κεφαλίδας. Για ανίχνευση χαμένων πακέτων, το κάθε τμήμα αριθμείται με έναν συγκεκριμένο αριθμό ακολουθίας (sequence number) τον οποίο κάθε φορά ο αποστολέας αυξάνει κατά τον αριθμό των bytes που έχουν ως τώρα σταλεί επιτυχώς. Ο παραλήπτης απαντά δηλώνοντας στο πεδίο του αριθμού επιβεβαίωσης (acknowledgment number) τον αριθμό ακολουθίας του επόμενου τμήματος που αναμένει. Τμήματα για τα οποία η επιβεβαίωση καθυστερεί περισσότερο από ένα χρονικό διάστημα RTO (Retransmission TimeOut) επανεκπέμπονται.

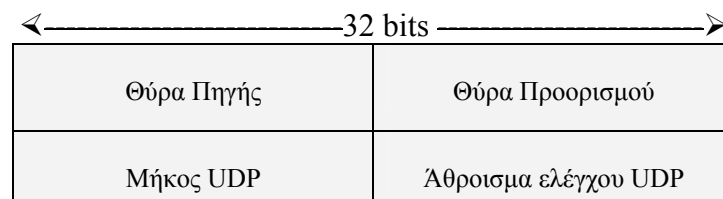
Προκειμένου να μην καθυστερείται η ανταλλαγή δεδομένων από τη φυσική καθυστέρηση του καναλιού, το TCP επιχειρεί να στείλει έναν συγκεκριμένο όγκο δεδομένων χωρίς να περιμένει την επιβεβαίωση για το πρώτο τμήμα. Ο όγκος αυτός των δεδομένων που ανά πάσα στιγμή βρίσκονται ανεπιβεβαίωτα στο δίκτυο λέγεται παράθυρο (TCP window) ή παράθυρο συμφόρησης (congestion window). Το παράθυρο συμφόρησης αρχίζει με την τιμή του ενός τμήματος και αυξάνεται με κάθε επιτυχή επιβεβαίωση, αυξάνοντας έτσι και την ταχύτητα μετάδοσης. Η αύξηση είναι αρχικά εκθετική (διαδικασία αργής εκκίνησης - «slow start») και αργότερα γραμμική

(διαδικασία αποφυγής συμφόρησης - «congestion avoidance»). Το παράθυρο που χρησιμοποιεί ο αποστολέας δεν μπορεί να υπερβεί την ονομαστική τιμή (receiver advertised window) που δηλώνει ο παραλήπτης σε κάθε επιβεβαίωση χρησιμοποιώντας το αντίστοιχο πεδίο της επικεφαλίδας. Σε περίπτωση απώλειας πακέτου, ο αποστολέας μειώνει το παράθυρο συμφόρησης στο ήμισυ της προηγούμενης τιμής του, με αποτέλεσμα να μειώνεται αντίστοιχα και η ταχύτητα αποστολής.

Είναι γεγονός ότι το TCP είναι το πιο διαδεδομένο πρωτόκολλο στις διαδικτυακές συνδέσεις. Είναι χαρακτηριστικό ότι το 95% των bytes, το 90% των πακέτων και το 75% των συνδέσεων στο Internet σήμερα χρησιμοποιούν το TCP.

2.9.2 Το Πρωτόκολλο UDP

Η στοίβα πρωτοκόλλων του Internet υποστηρίζει επίσης ένα πρωτόκολλο μεταφοράς πληροφοριών χωρίς σύνδεση, το Πρωτόκολλο Δεδομενογραφημάτων Χρήστη UDP (User Datagram Protocol)^[19]. Το UDP προσφέρει έναν τρόπο για να στέλνουν οι εφαρμογές ενθυλακωμένα ακατέργαστα δεδομενογραφήματα IP χωρίς να πρέπει να εγκαταστήσουν μια σύνδεση. Πολλές εφαρμογές πελάτη-εξυπηρετητή, που έχουν μία αίτηση και μία απόκριση, προτιμούν να χρησιμοποιήσουν το UDP παρά να μπου στον κόπο να εγκαταστήσουν και κατόπιν να απολύσουν μια σύνδεση.



Σχήμα 16 : Η επικεφαλίδα UDP.

Ένα τεμάχιο UDP αποτελείται από μια επικεφαλίδα των 8 byte (64 bit), ακολουθούμενη από δεδομένα. Η επικεφαλίδα φαίνεται στο παραπάνω σχήμα. Οι δύο θύρες εξυπηρετούν την ίδια λειτουργία όπως και στο TCP: την αναγνώριση των ακραίων σημείων στα μηχανήματα πηγής και προορισμού. Το πεδίο Μήκος UDP (UDP length)

αφορά στην επικεφαλίδα 8 byte και στα δεδομένα. Το πεδίο Άθροισμα ελέγχου UDP (UDP checksum) περιλαμβάνει την ίδια ψευδό-επικεφαλίδα με το TCP, την επικεφαλίδα UDP και τα δεδομένα UDP, συμπληρωμένα ώστε να σχηματίζουν έναν άρτιο αριθμό, αν χρειασθεί. Το Άθροισμα ελέγχου UDP είναι προαιρετικό και καταχωρείται ως 0 όταν δεν υπολογίζεται. (Το πραγματικά υπολογισμένο 0 καταχωρείται με όλα τα bit ίσα με 1, που είναι το ίδιο σε συμπλήρωμα ως προς 1). Το να μην χρησιμοποιηθεί είναι ανόητο, εκτός εάν η ποιότητα των δεδομένων δεν έχει μεγάλη σημασία (π.χ. η ψηφιοποιημένη φωνή).

Το UDP είναι ένα μη αξιόπιστο πρωτόκολλο, για εφαρμογές που δεν θέλουν τον έλεγχο της ακολουθίας ή της ροής του TCP και επιθυμούν να χρησιμοποιήσουν δικό τους. Επίσης χρησιμοποιείται ευρέως σε γρήγορες εφαρμογές και ερωταποκρίσεις, τύπου πελάτη-εξυπηρετητή, όπου η άμεση παράδοση είναι σπουδαιότερη από τη σωστή παράδοση, όπως π.χ. είναι η μετάδοση φωνής ή βίντεο.

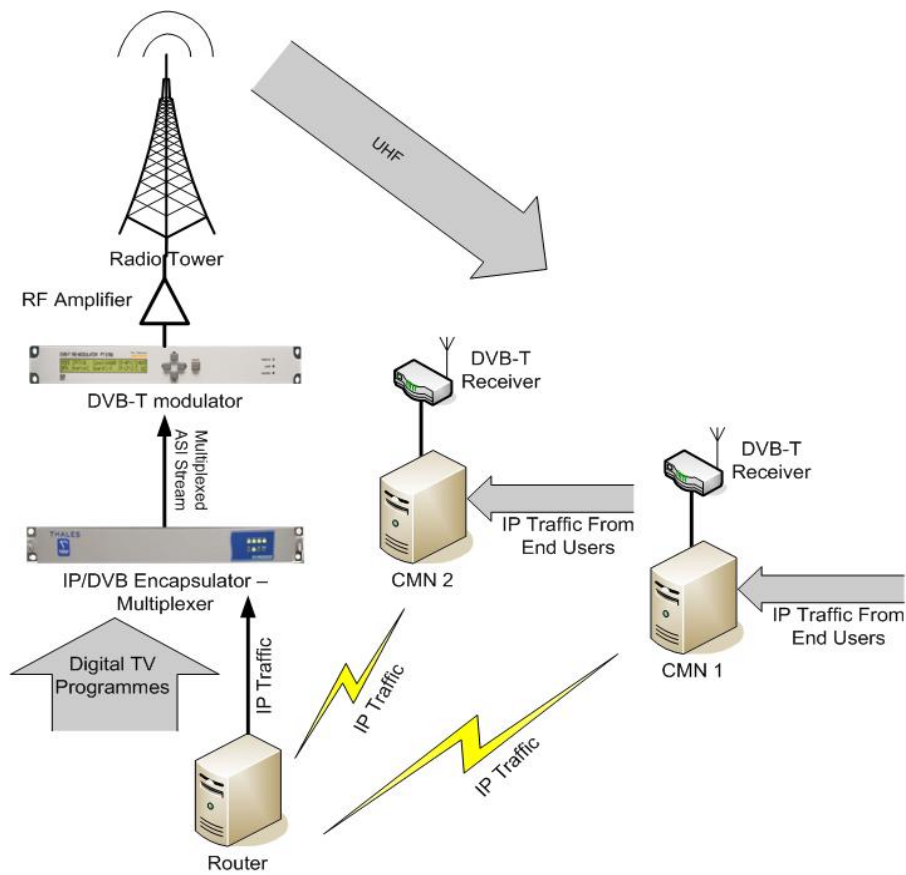
3. Παρουσίαση και Υλοποίηση Αρχιτεκτονικής Δικτύου Πειραματικών Μετρήσεων

3.1 Εισαγωγή

Στην παράγραφο 2.4 έγινε μια γενική περιγραφή ενός συστήματος παροχής υπηρεσιών διαδραστικής ψηφιακής τηλεόρασης καθώς επίσης και υπηρεσιών IP. Δόθηκε μια επιγραμματική αναφορά σε αυτές τις υπηρεσίες και παράλληλα παρουσιάστηκε η γενική αρχιτεκτονική ενός τέτοιου συστήματος. Στο κεφάλαιο αυτό θα περιγραφεί λεπτομερειακά το περιβάλλον IDTV αναλύοντας εις βάθος τα στοιχεία από τα οποία αποτελείται και τον τρόπο υλοποίησης του. Η προτεινόμενη αρχιτεκτονική που θα περιγραφεί σχεδιάστηκε με σκοπό να επιτευχθεί η εφαρμογή της Ποιότητας Υπηρεσίας ακολουθώντας το μοντέλο των Διαφοροποιημένων Υπηρεσιών.

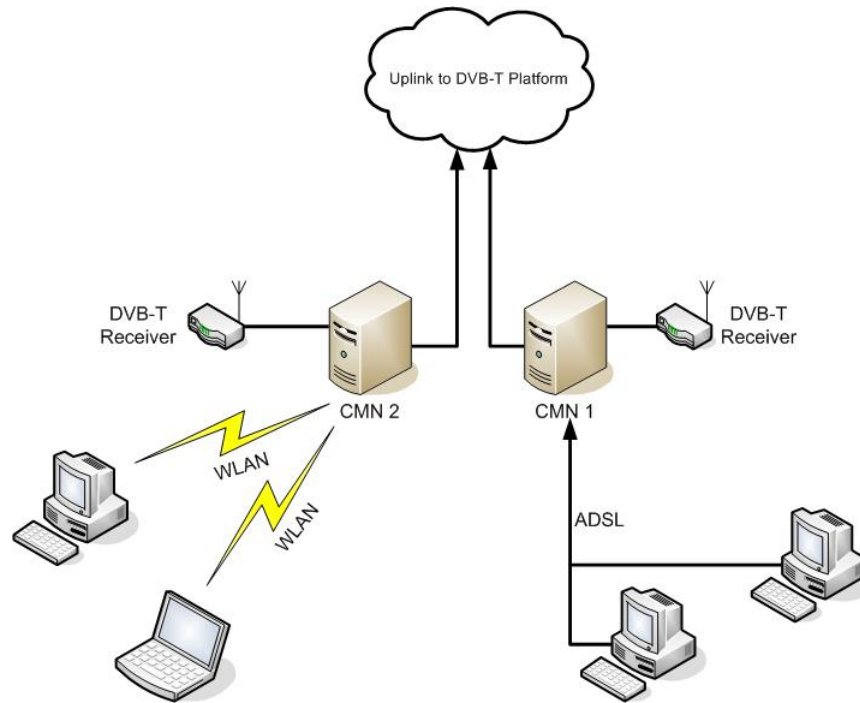
3.2 Σχεδιασμός Συστήματος

Όπως έχει ήδη αναφερθεί, ένα σύστημα IDTV αποτελείται από το κεντρικό σημείο ευρυεκπομπής, ένα αριθμό CMN's και τους τελικούς χρήστες. Το μοντέλο που θα ακολουθηθεί είναι το DVB-RCT (DVB Terrestrial Return Channel). Σύμφωνα με το μοντέλο αυτό η αμφίδρομη επικοινωνία μεταξύ ενός CMN και του σημείου ευρυεκπομπής απαιτεί την παρουσία δύο ζεύξεων, της κατερχόμενης ζεύξης (downlink) για την μετάδοση από το σημείο εκπομπής προς τους CMN's και της ανερχόμενης (uplink) για την επικοινωνία των CMN's με την πλατφόρμα DVB-T. Ως κατερχόμενη ζεύξη χρησιμοποιείται το κανάλι DVB-T, ενώ η τεχνολογία WLAN θα χρησιμοποιηθεί για την δημιουργία της ανερχόμενης ζεύξης ή αλλιώς για το κανάλι επιστροφής. Το δίκτυο που ορίζουν αυτά τα δύο κανάλια επικοινωνίας αποτελεί το Δίκτυο Πυρήνα (Core Network) του συστήματος.



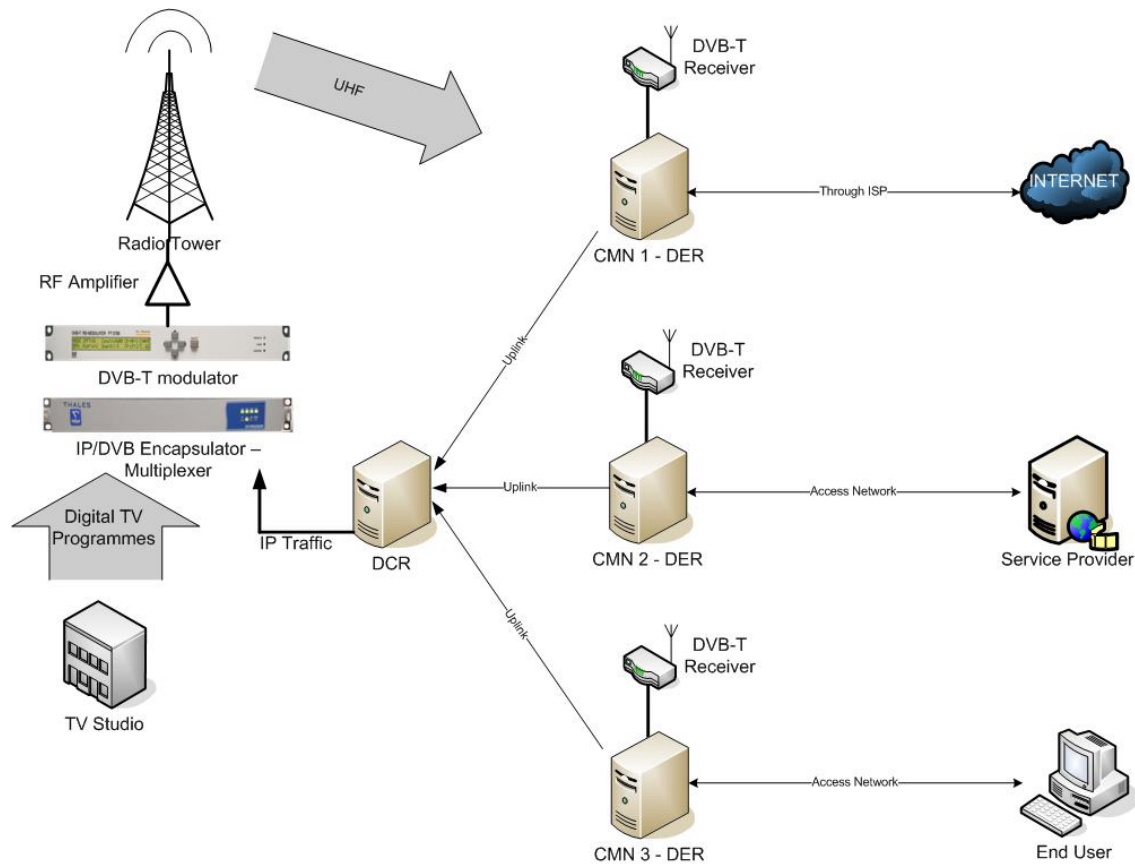
Σχήμα 17 : Δίκτυο Πυρήνα του συστήματος.

Η πρόσβαση στο δίκτυο πυρήνα επιτυγχάνεται με χρήση κάποιας από τις διαθέσιμες τεχνολογίες πρόσβασης. Οι τελικοί χρήστες μπορούν να επιλέξουν μεταξύ των τεχνολογιών WLAN, xDSL, PSTN και άλλων, για να συνδεθούν σε έναν CMN και αποκτήσουν επικοινωνία με το υπόλοιπο δίκτυο. Στη συγκεκριμένη αρχιτεκτονική θα χρησιμοποιηθούν οι τεχνολογίες WLAN και ADSL. Αυτή η διασύνδεση των χρηστών με κάποιον CMN ορίζει το Δίκτυο Πρόσβασης (Access Network).



Σχήμα 18 : Δίκτυα Πρόσβασης του συστήματος.

Η αρχιτεκτονική ενός δικτύου που υποστηρίζει τον μηχανισμό των Διαφοροποιημένων Υπηρεσιών παρουσιάζει κάποια αναλογία με την αρχιτεκτονική ενός IDTV σύστημα, βοηθώντας έτσι την ενσωμάτωση της πρώτης στην δεύτερη. Προσθέτοντας στις λειτουργίες κάθε CMN την λειτουργικότητα ενός DER και στους δρομολογητές, που προωθούν την IP κίνηση των CMN's στην DVB-T πλατφόρμα, τις εργασίες που εκτελεί ένας DCR επιτυγχάνεται η συνύπαρξη των δύο αρχιτεκτονικών. Στο ολοκληρωμένο δίκτυο που προκύπτει κάθε CMN, εκτός από την επικοινωνία των τελικών χρηστών, αναλαμβάνει και την ταξινόμηση, αστυνόμευση και διαμόρφωση της εισερχόμενης δικτυακής κίνησης. Οι δρομολογητές του δικτύου πυρήνα, πλέον, θα διαχειρίζονται την κίνηση εξασφαλίζοντας την τήρηση των PHB's και την διασφάλιση της ποιότητας στις προσφερόμενες υπηρεσίες.

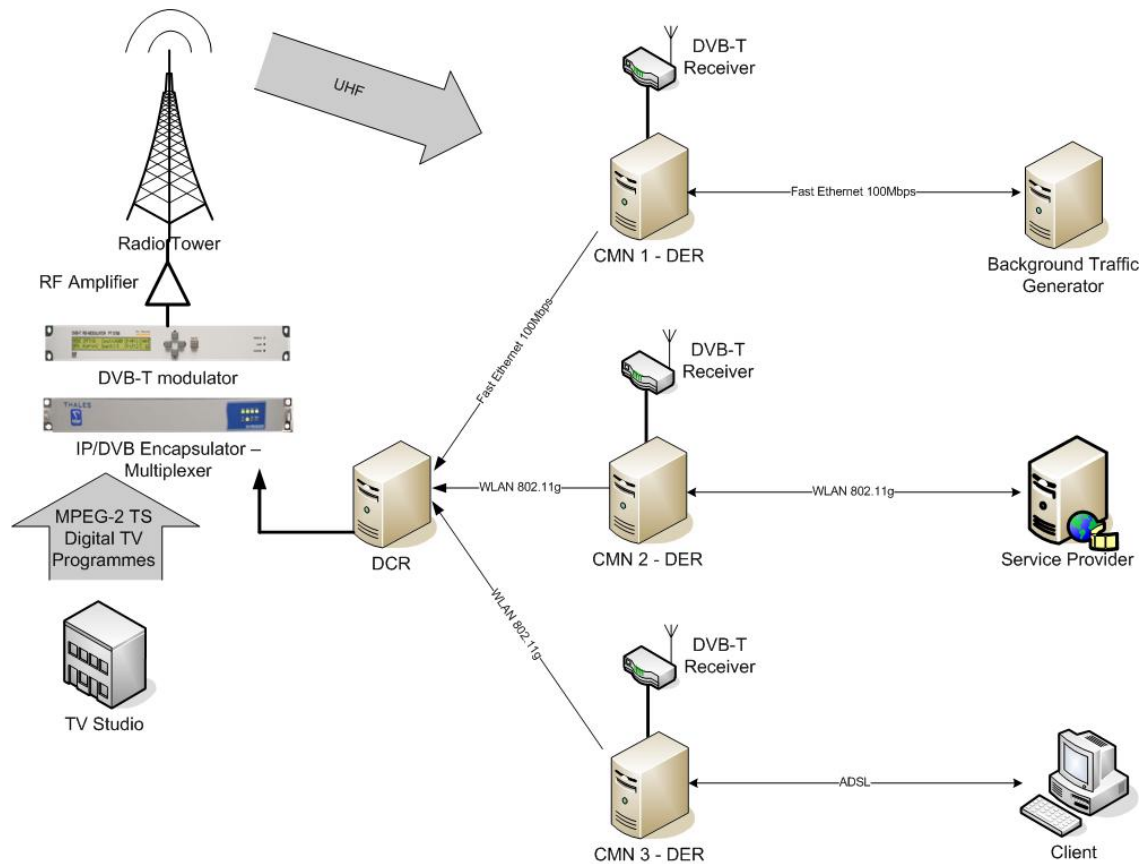


Σχήμα 19 : Γενική αρχιτεκτονική συστήματος IDTV με DiffServ.

3.3 Υλοποίηση Δικτύου Πειραματικών Μετρήσεων

Για να μελετηθεί η συμπεριφορά του δικτύου και να εξεταστούν τα οφέλη της ενσωμάτωσης του μηχανισμού παροχής QoS σε ένα περιβάλλον IDTV, θα πραγματοποιηθεί μια σειρά πειραματικών μετρήσεων κάτω από πραγματικές συνθήκες, κάνοντας χρήση του εργαστηριακού εξοπλισμού του εργαστηρίου Έρευνας και Ανάπτυξης Τηλεπικοινωνιακών Συστημάτων ([Pasiphae](#)) του Ανώτατου Τεχνολογικού Ιδρύματος Κρήτης. Η πειραματική προσέγγιση θα καλύψει έναν αριθμό πιθανών σεναρίων, ώστε να μπορεί να πραγματοποιηθεί αξιολόγηση του δικτύου.

Η ολοκληρωμένη αρχιτεκτονική στην οποία θα βασιστεί το πειραματικό σκέλος της πτυχιακής εργασίας φαίνεται στο σχεδιάγραμμα που ακολουθεί.



Σχήμα 20 : Αρχιτεκτονική συστήματος IDTV με DiffServ.

Οι οντότητες από τις οποίες θα αποτελείται το παραπάνω σύστημα είναι οι εξής :

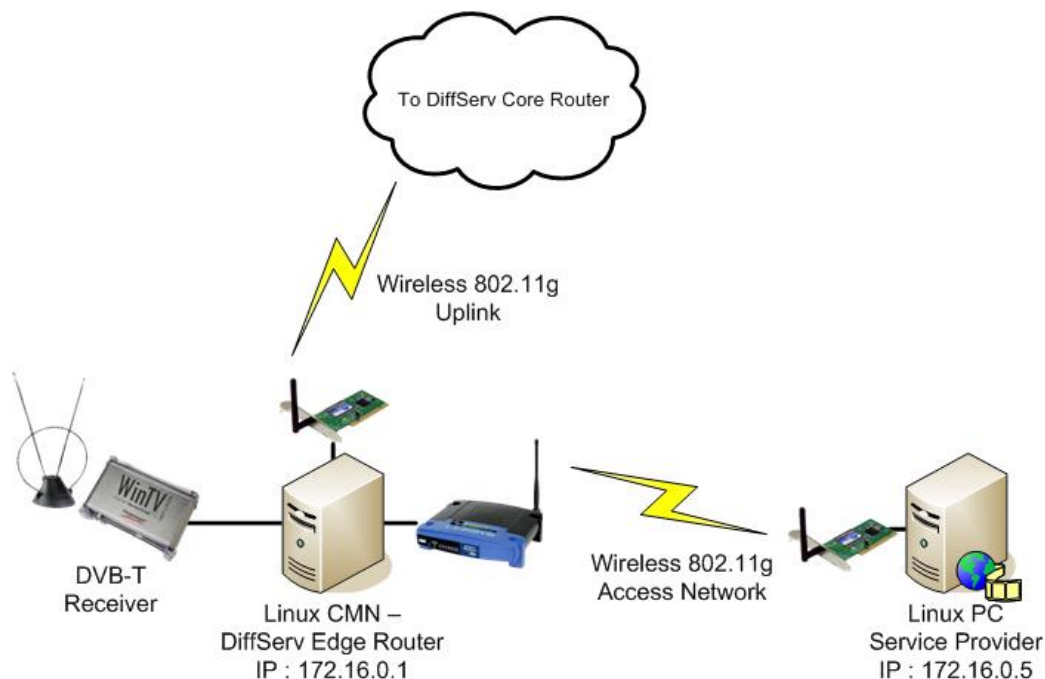
1. Ένας Παροχέας Υπηρεσιών (Service Provider - SP) που θα προσφέρει τις διαδραστικές υπηρεσίες στον τελικό χρήστη.
2. Ένας υπολογιστής που θα δημιουργεί κίνηση Best Effort (Background Traffic Generator) στο δίκτυο κορμού.
3. Τον τελικό χρήστη.
4. Τρεις DiffServ Edge Routers, οι οποίοι θα αναλάβουν την σήμανση και ρύθμιση της δικτυακής κίνησης που παράγεται από τον SP, τον Background Traffic Generator και τον τελικό χρήστη.
5. Τον DiffServ Core Router, ο οποίος θα προωθεί την κίνηση που λαμβάνει από τους DER's στην DVB-T πλατφόρμα σύμφωνα με τις PHB's.
6. Ένα παροχέα MPEG-2 ψηφιακών τηλεοπτικών προγραμμάτων.

3.3.1 Ρύθμιση Εξοπλισμού

Για την υλοποίηση της αρχιτεκτονικής που φαίνεται στο σχήμα θα χρησιμοποιηθεί ένας αριθμός υπολογιστών, οι οποίοι πρέπει να ρυθμιστούν αναλόγως με τις λειτουργίες που θα εκτελεί έκαστος.

3.3.1.1 Ρύθμιση Παροχέα Υπηρεσιών και του αντίστοιχου CMN-DER

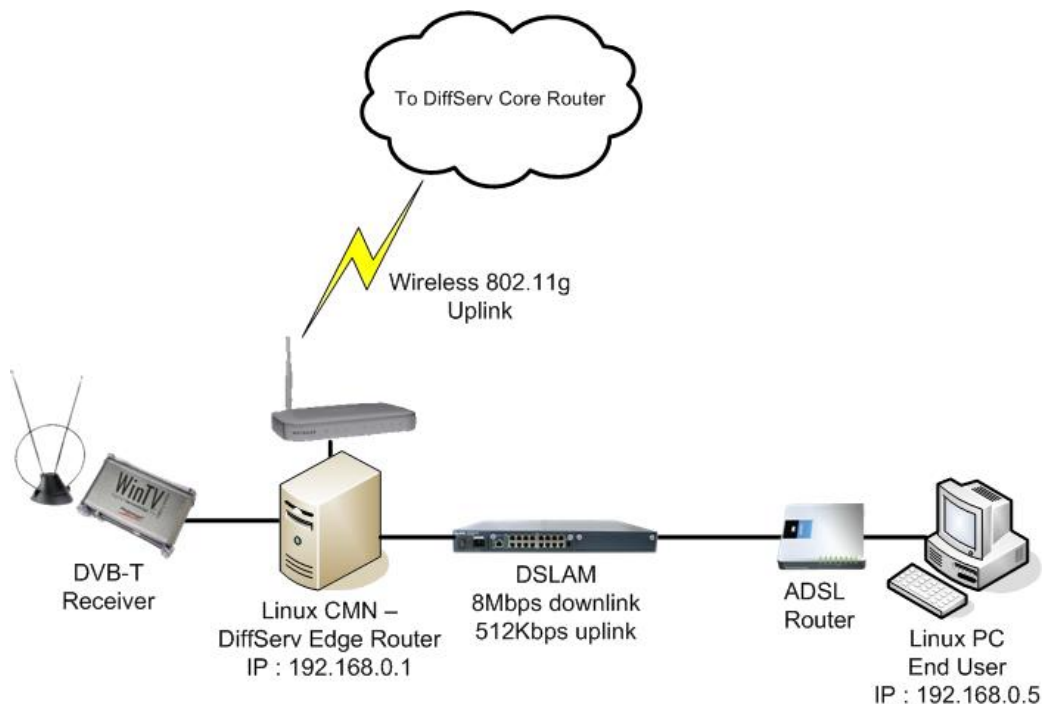
Για την υλοποίηση του παροχέα διαδραστικών υπηρεσιών θα γίνει χρήση ενός υπολογιστή στον οποίο έχει εγκατασταθεί το ελεύθερο λειτουργικό σύστημα Linux (Ubuntu Distribution) και επικοινωνεί με τον CMN – DER μέσω μιας ασύρματης δικτυακής κάρτας PCI. Από την πλευρά του CMN – DER χρησιμοποιείται ένας υπολογιστή με εγκατεστημένο το λειτουργικό σύστημα Linux (Debian Distribution), δύο ασύρματες δικτυακές διεπαφές για την επικοινωνία με τον Service Provider και την DVB-T πλατφόρμα και ένα δέκτη DVB-T σήματος.



Σχήμα 21 : Ρύθμιση του Service Provider και του αντίστοιχου CMN-DER.

3.3.1.2 Ρύθμιση Τελικού Χρήστη και του αντίστοιχου CMN-DER

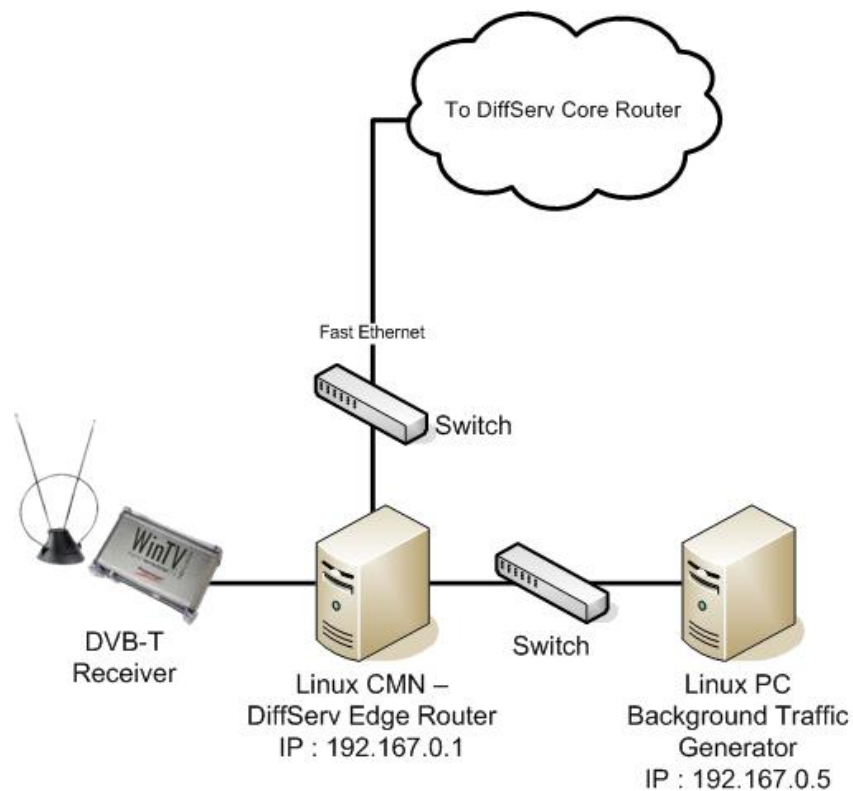
Για την υλοποίηση του τελικού χρήστη θα γίνει χρήση ενός υπολογιστή στον οποίο έχει εγκατασταθεί το ελεύθερο λειτουργικό σύστημα Linux (Debian Distribution) και επικοινωνεί με τον CMN – DER μέσω ενός ADSL router, ο οποίος συνδέεται με μια συσκευή DSLAM για την δημιουργία μιας γραμμής ADSL (8Mbps/512Kbps). Από την πλευρά του CMN – DER χρησιμοποιείται ένας υπολογιστή με εγκατεστημένο το λειτουργικό σύστημα Linux (Ubuntu Distribution), μια κάρτα δικτύου Fast Ethernet για την επικοινωνία με το DSLAM, μια ασύρματη κάρτα δικτύου για την επικοινωνία με την DVB-T πλατφόρμα και ένα δέκτη DVB-T σήματος.



Σχήμα 22 : Ρύθμιση του τελικού χρήστη και του αντίστοιχου CMN-DER.

3.3.1.3 Ρύθμιση Γεννήτριας Κίνησης και του αντίστοιχου CMN-DER

Για την υλοποίηση του Background Traffic Generator θα γίνει χρήση ενός υπολογιστή στον οποίο έχει εγκατασταθεί το ελεύθερο λειτουργικό σύστημα Linux (Debian Distribution) και επικοινωνεί με τον CMN – DER μέσω μιας κάρτας δικτύου Fast Ethernet. Από την πλευρά του CMN – DER χρησιμοποιείται ένας υπολογιστή με εγκατεστημένο το λειτουργικό σύστημα Linux (Debian Distribution), μια κάρτα δικτύου Fast Ethernet για την επικοινωνία με τον Background Traffic Generator, μια κάρτα δικτύου Fast Ethernet για την επικοινωνία με την DVB-T πλατφόρμα και ένα δέκτη DVB-T σήματος.



Σχήμα 23 : Ρύθμιση του background traffic generator και του αντίστοιχου CMN-DER.

3.3.1.4 Ρύθμιση DVB-T πλατφόρμας και DCR

Για την υλοποίηση του DiffServ Core Router θα γίνει χρήση ενός υπολογιστή στον οποίο έχει εγκατασταθεί το ελεύθερο λειτουργικό σύστημα Linux (Debian Distribution) και προωθεί την κίνηση που λαμβάνει από τους CMN's/DER's μέσω μιας κάρτας δικτύου Fast Ethernet στην μονάδα του IP ενθυλακωτή/πολυπλέκτη, σύμφωνα με τα εκάστοτε PHB's.

Ένας MPEG 2 TS Server παρέχει 3 ψηφιακά τηλεοπτικά προγράμματα χρησιμοποιώντας ένα ASI interface για την επικοινωνία με τον IP ενθυλακωτή.

Στην DVB-T πλατφόρμα, η μονάδα του IP ενθυλακωτή/πολυπλέκτη, που πολυπλέκει τα λαμβανόμενα ψηφιακά τηλεοπτικά προγράμματα και την δικτυακή κίνηση σε ένα ρεύμα μεταφοράς, ρυθμίστηκε να δεσμεύει για τις IP υπηρεσίες 8 Mbps, ενώ για τα ψηφιακά τηλεοπτικά προγράμματα 13 Mbps.

Ο πομπός DVB-T (COFDM) ρυθμίστηκε σε διαμόρφωση 16QAM, ρυθμό κώδικα 7/8 και διάστημα φύλαξης (guard interval) ίσο με το 1/32 του μήκους συμβόλου. 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 522 MHz (κανάλι 27) με επίπεδο δύναμης (power level) 1Watt και το συγκεκριμένο κανάλι έχει εύρος 8MHz.



Σχήμα 24 : Ο διαμορφωτής COFDM που παρουσιάζει τα στοιχεία της διαμόρφωσης.

3.4 Περιγραφή Πειραματικής Διαδικασίας

Στην ενότητα αυτή θα γίνει μια περιγραφή της πειραματικής διαδικασίας που θα ακολουθηθεί μέσω κάποιων σεναρίων, κάθε ένα από τα οποία θα καλύψει συγκεκριμένες περιπτώσεις δικτυακής κίνησης. Σκοπός είναι να παρατηρηθεί η συμπεριφορά του δικτύου και να αξιολογηθεί η επίδοση του από την μέτρηση κάποιων χαρακτηριστικών που επηρεάζουν την ποιότητα των υπηρεσιών.

Αρχικά, στο πρώτο σενάριο θα μελετηθεί η συμπεριφορά του δικτύου στην παρουσία δικτυακής κίνησης πρωτοκόλλου μεταφοράς TCP, στις εξής περιπτώσεις : α) η κίνηση να είναι η μόνη που χρησιμοποιεί το δίκτυο, β) να προστεθεί UDP background κίνηση και γ) να εφαρμοστεί ο μηχανισμός DiffServ και να επαναληφθεί η β περίπτωση. Στο δεύτερο σενάριο θα πραγματοποιηθούν οι ίδιες, με το πρώτο σενάριο, μετρήσεις με την διαφορά ότι πλέον θα υπάρχει UDP κίνηση για την προσομοίωση της μετάδοσης video, αντί της TCP. Στο τρίτο σενάριο θα ακολουθηθεί το ίδιο μοτίβο, με την διαφορά ότι στο δίκτυο την θέση της μετάδοσης video θα πάρει η τηλεφωνική επικοινωνία VoIP. Τέλος, στο τέταρτο σενάριο και στη πρώτη περίπτωση θα μεταδίδονται οι τρεις παραπάνω κινήσεις ταυτόχρονα. Στο δεύτερο μέρος αυτού του σεναρίου θα προστεθεί στις υπάρχουσες κινήσεις και UDP background κίνηση και στο τελευταίο μέρος θα εφαρμοστεί ο μηχανισμός DiffServ επαναλαμβάνοντας την διαδικασία του δεύτερου μέρους. Στα σενάρια κάθε μέτρηση θα διαρκεί 120 δευτερόλεπτα. Για την δημιουργία της TCP κίνησης θα χρησιμοποιηθεί το εργαλείο iperf, για την UDP και την background κίνηση το εργαλείο Mgen και για την VoIP το εργαλείο D-ITG.

3.4.1 Σενάριο 1^ο

- Μέρος 1^ο : Θα δημιουργηθεί κίνηση από τον Service Provider προς τον τελικό χρήστη, κάνοντας χρήση του πρωτοκόλλου μεταφοράς TCP και αξιοποιώντας όλο το διαθέσιμο εύρος ζώνης.
- Μέρος 2^ο : Θα δημιουργηθεί κίνηση από τον Service Provider προς τον τελικό χρήστη, κάνοντας χρήση του πρωτοκόλλου μεταφοράς TCP και αξιοποιώντας όλο το διαθέσιμο εύρος ζώνης, ενώ παράλληλα η γεννήτρια παραγωγής background κίνησης (θόρυβος) μεταδίδει με ρυθμό 7 Mbps.
- Μέρος 3^ο : Με ενεργοποιημένο τον μηχανισμό DiffServ θα δημιουργηθεί κίνηση από τον Service Provider προς τον τελικό χρήστη, κάνοντας χρήση του πρωτοκόλλου μεταφοράς TCP και αξιοποιώντας όλο το διαθέσιμο εύρος ζώνης, ενώ παράλληλα η γεννήτρια παραγωγής background κίνησης (θόρυβος) μεταδίδει με ρυθμό 7 Mbps. Η TCP κίνηση θα οριστεί σαν EF και θα λάβει 4.7 Mbps από το εύρος ζώνης του καναλιού, ενώ η background κίνηση θα οριστεί σαν Best Effort και θα λάβει το υπόλοιπο διαθέσιμο εύρος ζώνης.

3.4.2 Σενάριο 2^ο

- Μέρος 1^ο : Θα δημιουργηθεί κίνηση από τον Service Provider προς τον τελικό χρήστη, κάνοντας χρήση του πρωτοκόλλου μεταφοράς UDP και μεταδίδοντας με ρυθμό 3 Mbps, για την προσομοίωση video streaming.
- Μέρος 2^ο : Θα δημιουργηθεί κίνηση από τον Service Provider προς τον τελικό χρήστη, κάνοντας χρήση του πρωτοκόλλου μεταφοράς UDP και μεταδίδοντας με ρυθμό 3 Mbps, για την προσομοίωση video streaming, ενώ παράλληλα η γεννήτρια παραγωγής background κίνησης (θόρυβος) μεταδίδει με ρυθμό 7 Mbps.
- Μέρος 3^ο : Με ενεργοποιημένο τον μηχανισμό DiffServ θα δημιουργηθεί κίνηση από τον Service Provider προς τον τελικό χρήστη, κάνοντας χρήση του

πρωτοκόλλου μεταφοράς UDP και μεταδίδοντας με ρυθμό 3 Mbps για την προσομοίωση video streaming, ενώ παράλληλα η γεννήτρια παραγωγής background κίνησης (θόρυβος) μεταδίδει με ρυθμό 7 Mbps. Η UDP κίνηση θα οριστεί σαν EF και θα λάβει 3.3 Mbps από το εύρος ζώνης του καναλιού, ενώ η background κίνηση θα οριστεί σαν Best Effort και θα λάβει το υπόλοιπο διαθέσιμο εύρος ζώνης.

3.4.3 Σενάριο 3^ο

- Μέρος 1^ο: Θα γίνει προσομοίωση αμφίδρομης τηλεφωνικής επικοινωνίας πάνω από IP δίκτυο(VoIP) μεταξύ του Service Provider και του τελικού χρήστη κάνοντας χρήση του πρωτοκόλλου G.729.2.
- Μέρος 2^ο: Θα γίνει προσομοίωση αμφίδρομης VoIP επικοινωνίας μεταξύ του Service Provider και του τελικού χρήστη κάνοντας χρήση του πρωτοκόλλου G.729.2, ενώ παράλληλα η γεννήτρια παραγωγής background κίνησης (θόρυβος) μεταδίδει με ρυθμό 7 Mbps.
- Μέρος 3^ο: Με ενεργοποιημένο τον μηχανισμό DiffServ θα γίνει προσομοίωση αμφίδρομης VoIP επικοινωνίας μεταξύ του Service Provider και του τελικού χρήστη κάνοντας χρήση του πρωτοκόλλου G.729.2, ενώ παράλληλα η γεννήτρια παραγωγής background κίνησης (θόρυβος) μεταδίδει με ρυθμό 7 Mbps. Η VoIP κίνηση θα οριστεί σαν EF και θα λάβει 60 Kbps από το εύρος ζώνης του καναλιού, ενώ η background κίνηση θα οριστεί σαν Best Effort και θα λάβει το υπόλοιπο διαθέσιμο εύρος ζώνης.

3.4.4 Σενάριο 4^ο

- Μέρος 1^ο : Θα μεταδίδονται ταυτόχρονα από τον Service Provider προς τον τελικό χρήστη κινήσεις TCP αξιοποιώντας το διαθέσιμο εύρος ζώνης και UDP με ρυθμό μετάδοσης 3 Mbps και παράλληλα θα γίνεται προσομοίωση αμφίδρομης επικοινωνίας VoIP μεταξύ του Service Provider και του τελικού χρήστη κάνοντας χρήση του πρωτοκόλλου G.729.2.
- Μέρος 2^ο: Θα μεταδίδονται ταυτόχρονα από τον Service Provider προς τον τελικό χρήστη κινήσεις TCP αξιοποιώντας το διαθέσιμο εύρος ζώνης και UDP με ρυθμό μετάδοσης 3 Mbps και παράλληλα θα γίνεται προσομοίωση αμφίδρομης επικοινωνίας VoIP μεταξύ του Service Provider και του τελικού χρήστη κάνοντας χρήση του πρωτοκόλλου G.729.2 και η γεννήτρια παραγωγής background κίνησης (θόρυβος) θα μεταδίδει με ρυθμό 7 Mbps.
- Μέρος 3^ο: Με ενεργοποιημένο τον μηχανισμό DiffServ θα μεταδίδονται ταυτόχρονα από τον Service Provider προς τον τελικό χρήστη κινήσεις TCP αξιοποιώντας το διαθέσιμο εύρος ζώνης και UDP με ρυθμό μετάδοσης 3 Mbps και παράλληλα θα γίνεται προσομοίωση αμφίδρομης επικοινωνίας VoIP μεταξύ του Service Provider και του τελικού χρήστη κάνοντας χρήση του πρωτοκόλλου G.729.2 και η γεννήτρια παραγωγής background κίνησης (θόρυβος) θα μεταδίδει με ρυθμό 7 Mbps. Η VoIP κίνηση θα οριστεί σαν AF11 και θα λάβει 60 Kbps από το εύρος ζώνης του καναλιού, η UDP κίνηση θα οριστεί σαν AF21 και θα λάβει 3.2 Mbps, η TCP κίνηση θα οριστεί σαν AF31 και θα λάβει 3.2 Mbps και τέλος η background κίνηση θα οριστεί σαν Best Effort και θα λάβει το υπόλοιπο διαθέσιμο εύρος ζώνης.

3.5 Μετρική Επιδόσεων

Η Ποιότητα Υπηρεσίας, όπως έχει προαναφερθεί είναι μια γενική έννοια, και εξαρτάται από τις τιμές ενός συνόλου χαρακτηριστικών μεγεθών, βάση των οποίων μπορεί να γίνει αξιολόγηση και μελέτη ενός δικτύου.

Τα χαρακτηριστικά αυτά είναι :

- Η Ρυθμοαπόδοση (Throughput), το οποίο ορίζεται σαν το πόσο των δεδομένων που μεταφέρονται από ένα δικτυακό κόμβο σε έναν άλλο ανά μονάδα χρόνου.
- Η Καθυστέρηση (Delay – Latency) : α) Στην TCP επικοινωνία είναι γνωστή σαν Καθυστέρηση Κυκλικής Διαδρομής (Round-Trip delay Time – RTT) και αναφέρεται στον χρόνο που διανύθηκε από τη στιγμή που στάλθηκε ένα πακέτο έως την στιγμή που λήφθηκε η επιβεβαίωση του. β) Στην UDP επικοινωνία είναι γνωστή με τον όρο Μονόδρομη Καθυστέρηση (One Way Delay) και αναφέρεται στον χρόνο που χρειάζεται ένα πακέτο για να μεταδοθεί από ένα δικτυακό κόμβο σε έναν άλλο.
- Το Τρέμουλο (Jitter), που είναι η διακύμανση της καθυστέρησης, δηλαδή του χρόνου μεταφοράς από άκρο σε άκρο.
- Οι Απώλειες (Losses) : α) Στην TCP επικοινωνία σαν απώλειες θεωρείται ο αριθμός των επαναμεταδόσεων πακέτων που χάθηκαν στην διαδρομή. β) Στην UDP κίνηση ορίζεται ως ο λόγος του αριθμού των πακέτων που ελήφθησαν προς τον αριθμό των πακέτων που στάλθηκαν.

Από μελέτες που έχουν γίνει οι τιμές που θεωρούνται καλές για τα παραπάνω χαρακτηριστικά είναι : για την καθυστέρηση 0 - 150 ms, για το jitter 0 – 20 ms και για τις απώλειες 0 – 0.5%. Σαν αποδεκτά όρια τιμών θα είναι : για την καθυστέρηση 150 - 300 ms, για το jitter 20 – 50 ms και για τις απώλειες 0.5 – 1.5%. Κακές τιμές θα θεωρηθούν : για την καθυστέρηση μεγαλύτερες από 300 ms, για το jitter μεγαλύτερες από 50 ms και για τις απώλειες μεγαλύτερες από 1.5%^[23].

3.6 Περιγραφή Προγραμμάτων

Για την επίτευξη της πτυχιακής εργασίας έγινε χρήση διάφορων λογισμικών εφαρμογών και προγραμμάτων. Τα προγράμματα αυτά χρησιμοποιήθηκαν κυρίως για την παραγωγή δικτυακής κίνησης, την “σύλληψη”, παρουσίαση και ανάλυση των χαρακτηριστικών αυτής και για την υλοποίηση του μηχανισμού των Διαφοροποιημένων Υπηρεσιών. Παρακάτω γίνεται μια αναφορά στα προγράμματα που χρησιμοποιήθηκαν.

- [Iperf](#)^[14] : Το Iperf είναι ένα εργαλείο για δημιουργία TCP και UDP κινήσεων, το οποίο λειτουργεί σε όλα τα συστήματα (Unix, Windows, MacOS κλπ.). Μερικά από τα χαρακτηριστικά του γνωρίσματα είναι ότι παρέχει χρήσιμες πληροφορίες και αποτελέσματα για το εύρος ζώνης, τις απώλειες, την διακύμανση της καθυστέρησης και γενικότερα για την απόδοση του δικτύου.
- [MGEN](#)^[16] : Το MGEN (Multi-Generator) είναι ένα λογισμικό ανοιχτού κώδικα και παρέχει τη δυνατότητα να εκτελεστούν μετρήσεις για την απόδοση δικτύων που παρέχουν IP υπηρεσίες, δημιουργώντας UDP κίνηση.
- [D-ITG](#)^[15] : Το D-ITG είναι ένα πρόγραμμα ικανό να παράγει δικτυακές κινήσεις στα επίπεδα δικτύου, μεταφοράς και εφαρμογής. Στις πειραματικές μετρήσεις της συγκεκριμένης πτυχιακής το εργαλείο D-ITG θα χρησιμοποιηθεί για την δημιουργία της VoIP επικοινωνίας.
- [Tcpdump](#)^[21] : Το Tcpdump είναι ένα εργαλείο “σύλληψης” και παρακολούθησης της δικτυακής κυκλοφορίας, το οποίο σε συνεργασία με άλλα προγράμματα βοηθάει στην ανάλυση των διαφόρων χαρακτηριστικών των δικτυακών κινήσεων.
- [Tcptrace](#)^[25] : Το Tcptrace είναι ένα εργαλείο που χρησιμοποιείται για την ανάλυση αρχείων που έχουν δημιουργηθεί από διάφορα προγράμματα “σύλληψης” δικτυακής κίνησης, όπως είναι το tcpdump και το WinDump. Το tcptrace μπορεί να παράγει αρχεία τα οποία περιέχουν διαφόρους τύπους πληροφοριών για κάθε υπαρκτή κίνηση, όπως επαναμεταδόσεις, καθυστέρηση, ρυθμοαπόδοση και άλλα. Μπορεί επίσης να παράγει γραφικές παραστάσεις για κάθε μια από τις παραπάνω πληροφορίες, για περαιτέρω ανάλυση.

- [Iproute2 και tc](#)^[17] : Το iproute2 είναι μια συλλογή εφαρμογών για την διαχείριση IP δικτυακής κίνησης σε περιβάλλον Linux. Από τα εργαλεία που προσφέρει πιο σημαντικά θεωρούνται το ip και το tc. Το εργαλείο tc δίνει την δυνατότητα εισαγωγής ουρών, φίλτρων και την διαχείριση του εύρους ζώνης μιας ζεύξης.
- [Iptables](#)^[24] : 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 και η σήμανση των πεδίων ToS και DSCP της επικεφαλίδας των IP πακέτων.
- [Gnuplot](#)^[22] και [Xplot](#)^[26] : τα δύο αυτά προγράμματα χρησιμοποιούνται για την γραφική απεικόνιση συνόλων δεδομένων.

4. Παρουσίαση και Ανάλυση Αποτελεσμάτων

Στο κεφάλαιο που ακολουθεί θα γίνει μια αναφορά στα αποτελέσματα των πειραματικών μετρήσεων που πραγματοποιήθηκαν, τα οποία θα παρουσιαστούν γραφικά, δίνοντας επεξηγήσεις κατά την ανάλυση τους. Η αξιολόγηση των αποτελεσμάτων θα γίνει με βάση τον παρακάτω πίνακα^[23].

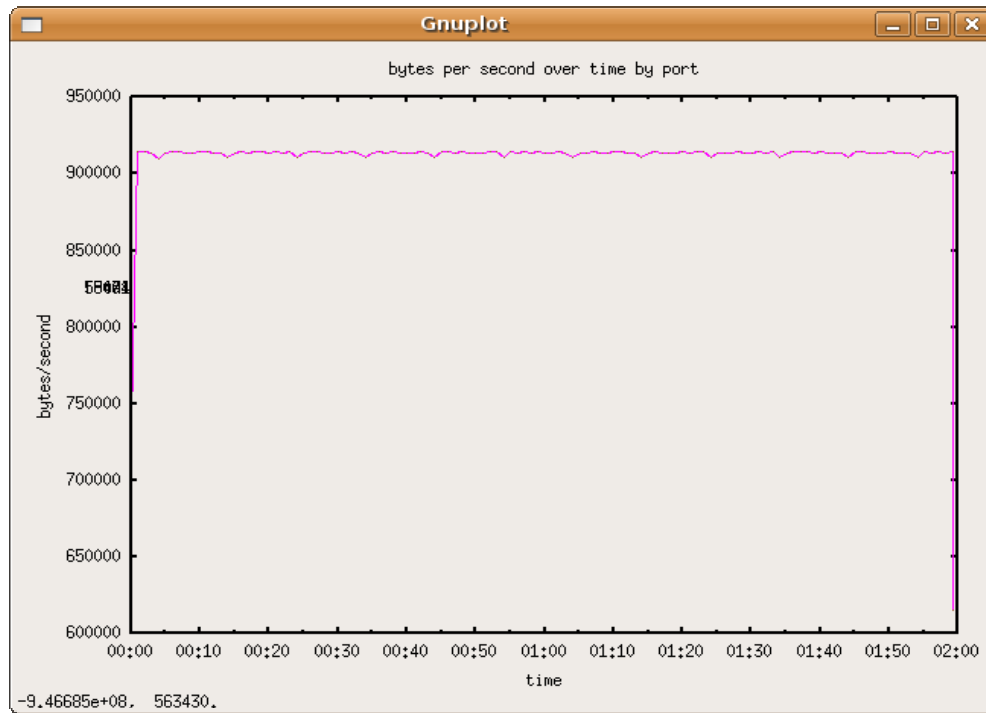
	Ικανοποιητικές Τιμές	Αποδεκτές–Μέτριες Τιμές	Κακές Τιμές
One Way Delay	0-150 ms	150-300 ms	>300 ms
Jitter	0-20 ms	20-50 ms	> 50 ms
Losses	0-0.5%	0.5-1.5%	> 1.5%

Πίνακας 3 : Τιμές Αξιολόγησης των χαρακτηριστικών QoS.

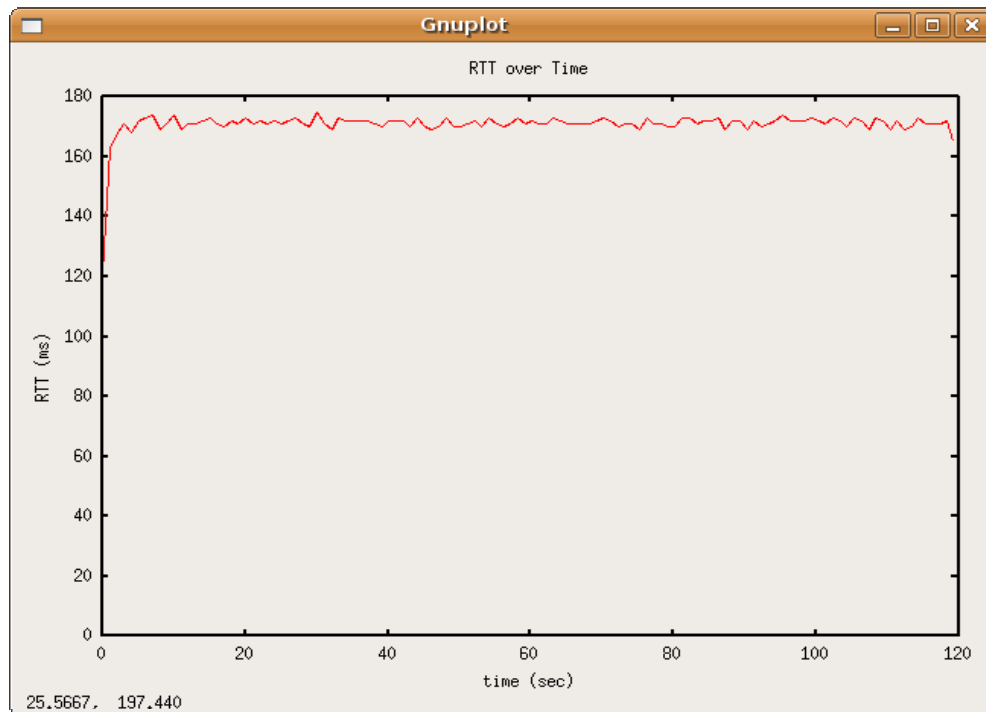
4.1 Σενάριο 1^ο

4.1.1 Μέρος 1^ο

Κατά την πειραματική μέτρηση στο δίκτυο μεταδιδόταν μόνο TCP κίνηση κάνοντας χρήση όλου του διαθέσιμου εύρους ζώνης. Αναλύοντας τα αποτελέσματα προκύπτει ότι η κίνηση έλαβε throughput 6.92 Mbps, με καμία επαναμετάδοση, που σημαίνει μηδενικές απώλειες και RTT καθυστέρηση στα 171.8 ms. Η μετάδοση αυτή με τις τιμές που αναφέρθηκαν θεωρείται καλή, κάνοντας αποδοτική και ποιοτική χρήση του δικτύου. Παρακάτω παρουσιάζονται γραφικές παραστάσεις του throughput και της RTT καθυστέρησης.



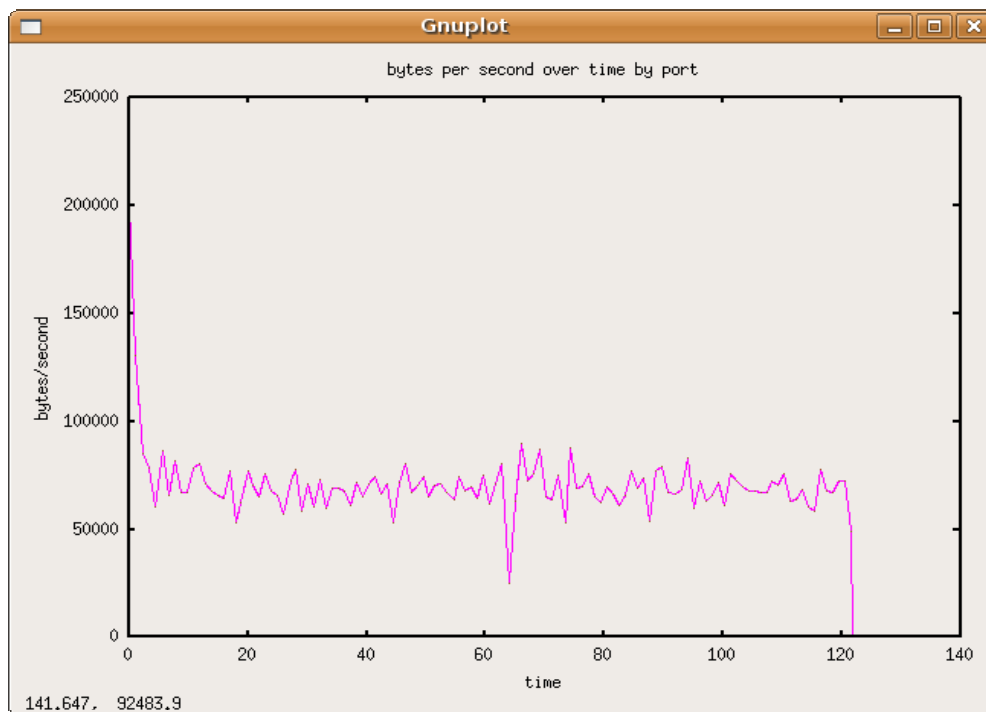
Σχήμα 25 : Γραφική παράσταση του Throughput.



Σχήμα 26 : Γραφική παράσταση του RTT.

4.1.2 Μέρος 2°

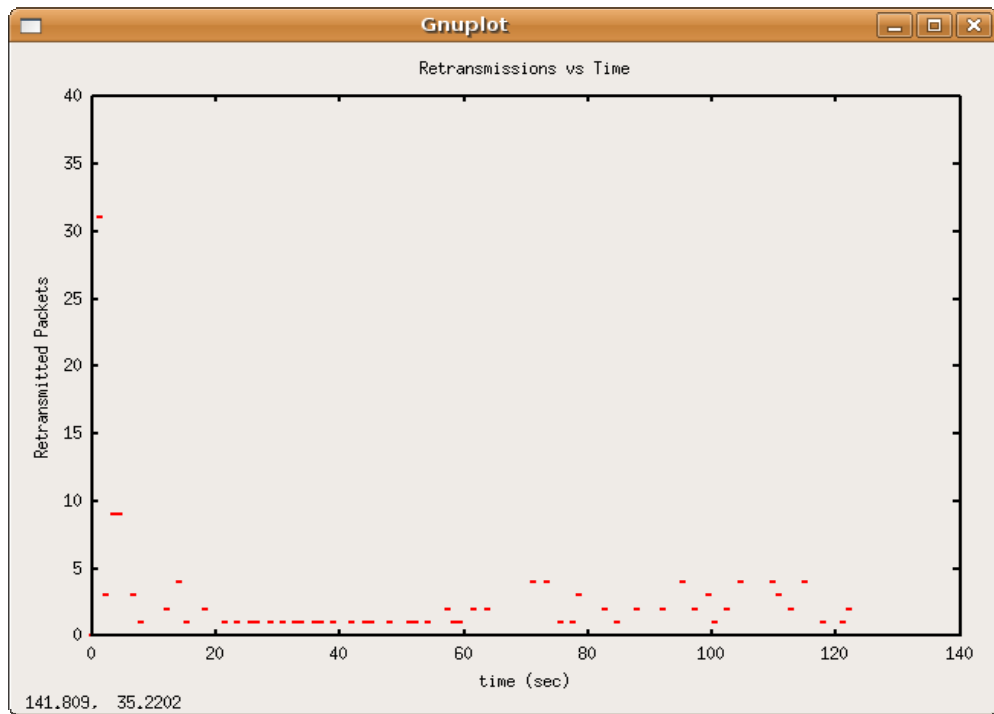
Κατά την πειραματική μέτρηση στο δίκτυο μεταδιδόταν TCP κίνηση κάνοντας χρήση του διαθέσιμου εύρους ζώνης, παράλληλα με background κίνηση των 7 Mbps. Στην περίπτωση αυτή η TCP κίνηση άγγιξε τα 530.7 Kbps throughput, με 147 επαναμετάδοσεις πακέτων και RTT καθυστέρηση στα 371.2 ms. Παρατηρείται, ότι η παρουσία της επιπλέον κίνησης στο κανάλι δημιούργησε συμφόρηση στο δίκτυο και πολλά προβλήματα στην μετάδοση της TCP κίνησης με αποτέλεσμα μεγάλο αριθμό απωλειών και υψηλή καθυστέρηση. Συνεπώς, επιβεβαιώνεται η άποψη για την μη ικανοποιητική παροχή ποιότητας υπηρεσιών στα IP δίκτυα που ακολουθούν το μοντέλο “Best Effort” μετάδοσης. Παρακάτω παρουσιάζονται γραφικές παραστάσεις του throughput, της RTT καθυστέρησης και των επαναμεταδόσεων.



Σχήμα 27 : Γραφική παράσταση του Throughput.



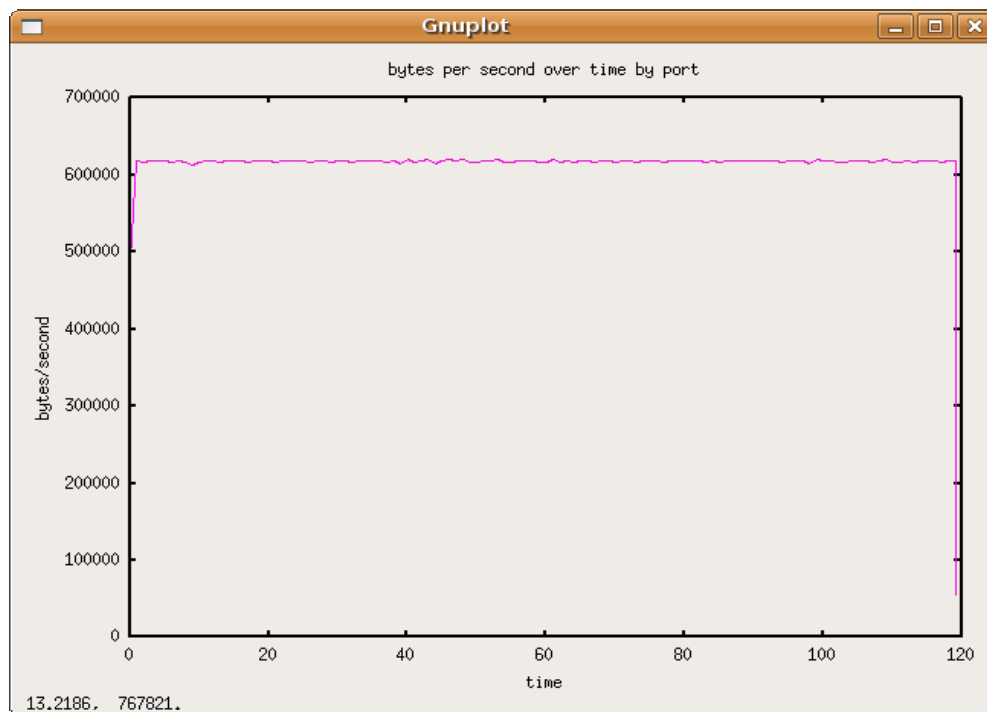
Σχήμα 28 : Γραφική παράσταση του RTT.



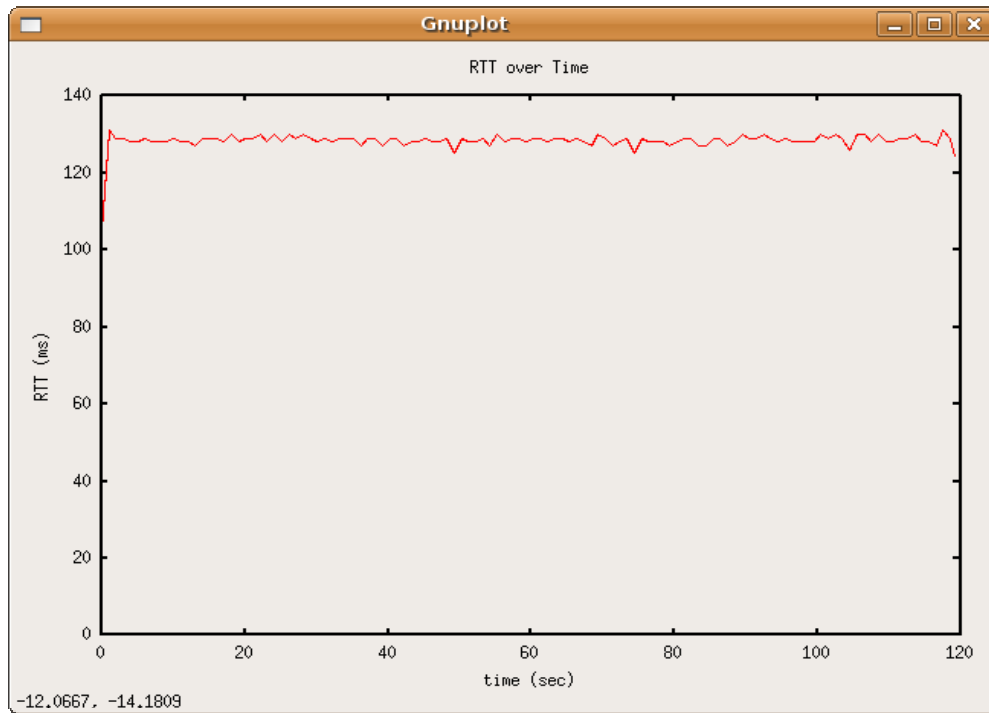
Σχήμα 29 : Γραφική παράσταση των retransmissions.

4.1.3 Μέρος 3°

Κατά την πειραματική μέτρηση στο δίκτυο ενεργοποιείται ο μηχανισμός DiffServ, ενώ μεταδιδόταν TCP κίνηση κάνοντας χρήση του διαθέσιμου εύρους ζώνης, παράλληλα με background κίνηση των 7 Mbps. Στην περίπτωση αυτή η TCP κίνηση άγγιξε τα 4.67 Mbps throughput, με καμία επαναμετάδοση πακέτων και RTT καθυστέρηση στα 129.3 ms. Παρατηρείται, ότι η παρουσία του μηχανισμού διασφάλισης της Ποιότητας Υπηρεσιών παρείχε εγγυημένη ποιότητα και έκανε αποδοτική χρήση των δικτυακών πόρων. Η επιπλέον κίνηση στο κανάλι δεν δημιούργησε συμφόρηση στο δίκτυο αλλά περιορίστηκε στο εύρος ζώνης που της αναλογούσε χωρίς να παρεμποδίσει την μετάδοση της TCP κίνησης. Οι παραπάνω τιμές είναι στα επίπεδα μια καλής μετάδοσης. Παρακάτω παρουσιάζονται γραφικές παραστάσεις του throughput και της RTT καθυστέρησης.



Σχήμα 30 : Γραφική παράσταση του Throughput.

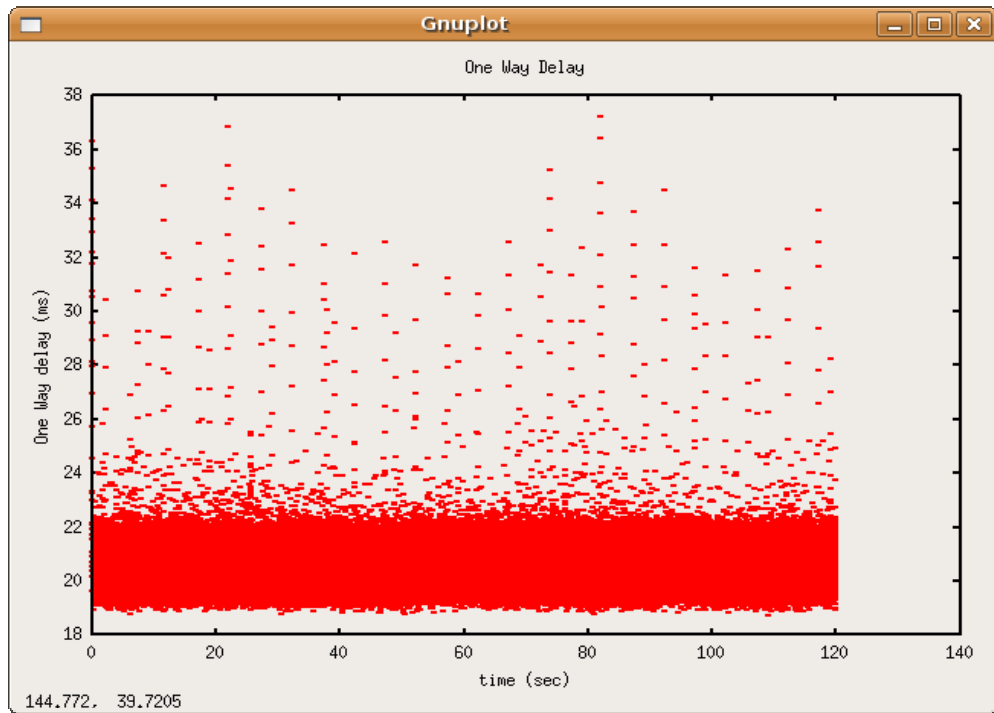


Σχήμα 31 : Γραφική παράσταση του RTT.

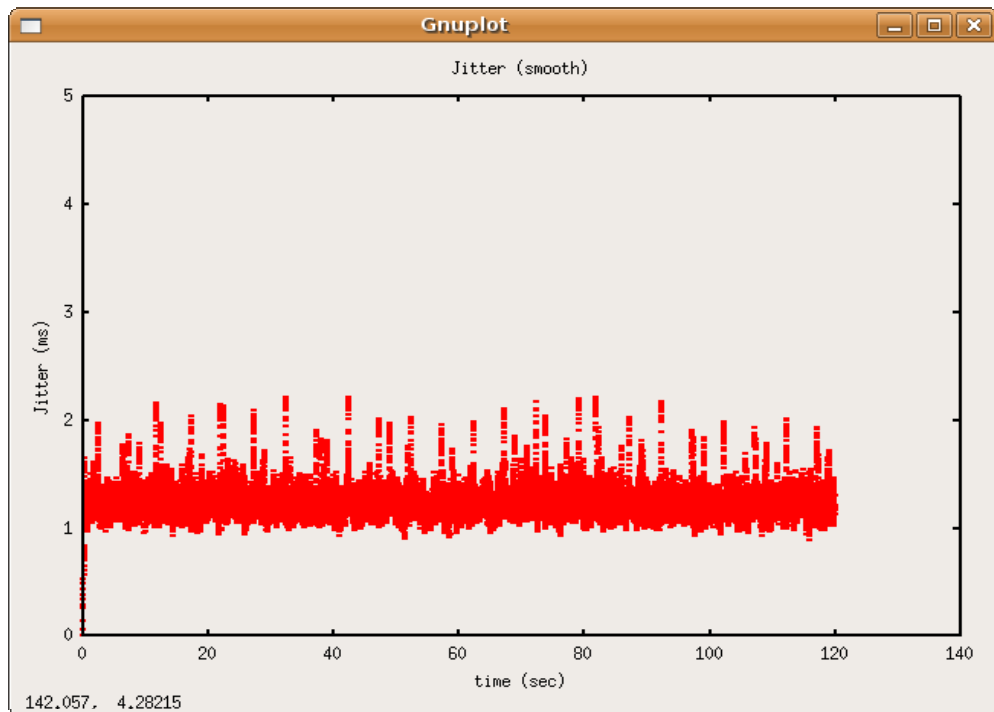
4.2 Σενάριο 2^ο

4.2.1 Μέρος 1^ο

Κατά την πειραματική μέτρηση στο δίκτυο μεταδιδόταν μόνο UDP κίνηση με ρυθμό μετάδοσης 3 Mbps. Από την ανάλυση των μετρήσεων προκύπτει ότι η προσομοίωση μετάδοσης video έλαβε 3.08 Mbps throughput, χωρίς απώλειες, με one way delay στα 20.62 ms και με την τιμή του jitter στα 1.23 ms. Ακολουθούν οι γραφικές παραστάσεις της one way delay και του jitter.



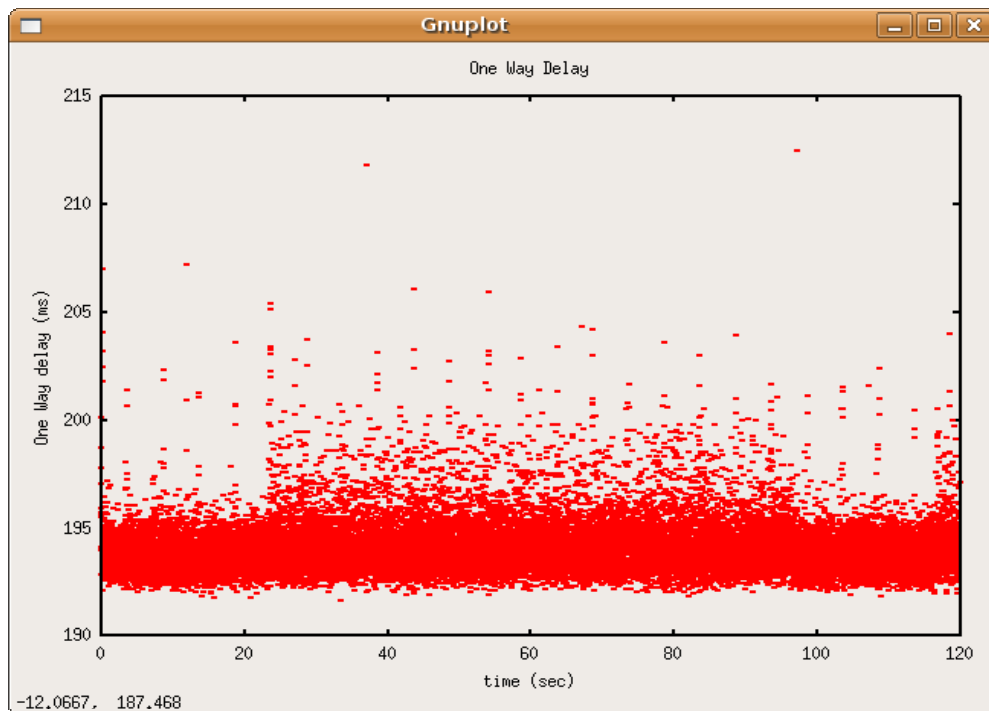
Σχήμα 32 : Γραφική παράσταση της One Way Delay.



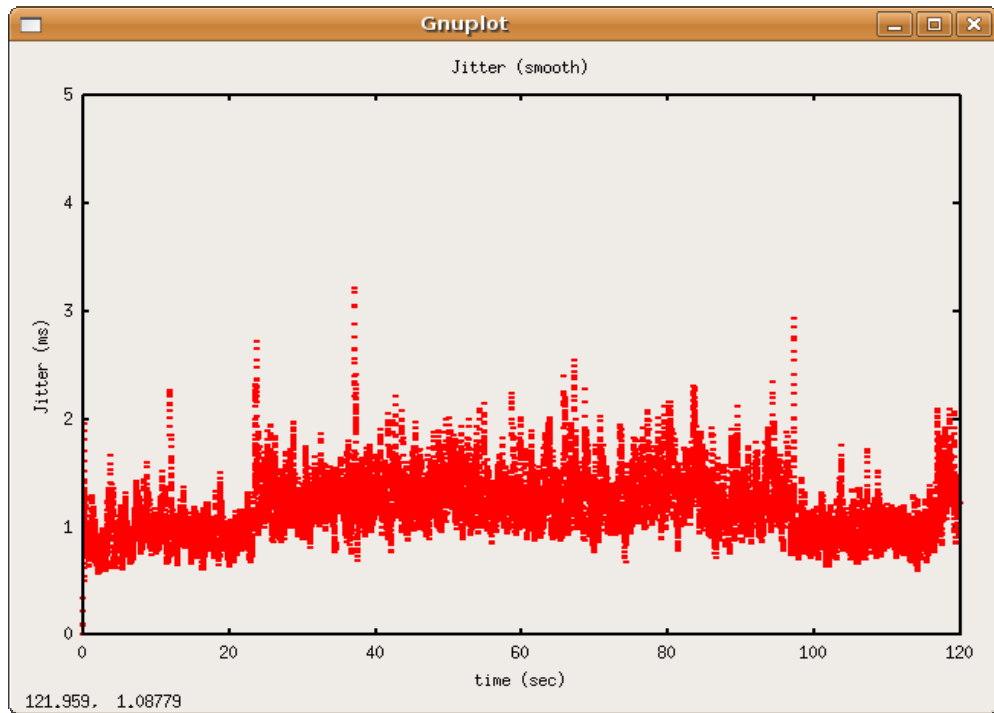
Σχήμα 33 : Γραφική παράσταση του Jitter.

4.2.2 Μέρος 2°

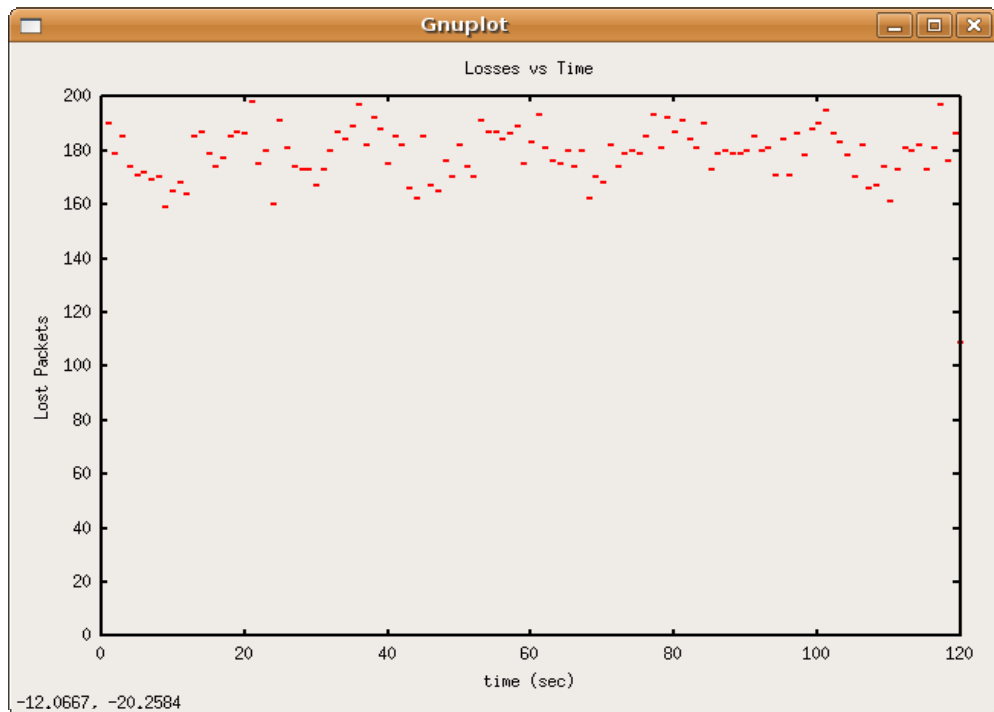
Κατά την πειραματική μέτρηση στο δίκτυο μεταδιδόταν UDP κίνηση με ρυθμό μετάδοσης 3 Mbps, παράλληλα με background κίνηση των 7 Mbps. Από την ανάλυση των μετρήσεων προκύπτει ότι η προσομοίωση μετάδοσης video έλαβε μόλις 1.58 Mbps throughput, με 48.8% απώλειες, με one way delay στα 194.14 ms και με την τιμή του jitter στα 1.18 ms. Παρατηρώντας τα αποτελέσματα γίνεται αντιληπτή η κακή ποιότητα στην μετάδοση, η οποία λόγω της συμφόρησης του δικτύου είχε ως επακόλουθο το μεγάλο ποσοστό απωλειών. Συνεπώς, η σωστή μετάδοση video μέσα από ένα τέτοιο δίκτυο δεν είναι δυνατή. Ακολουθούν οι γραφικές παραστάσεις της one way delay, του jitter και των απωλειών.



Σχήμα 34 : Γραφική παράσταση της One Way Delay.



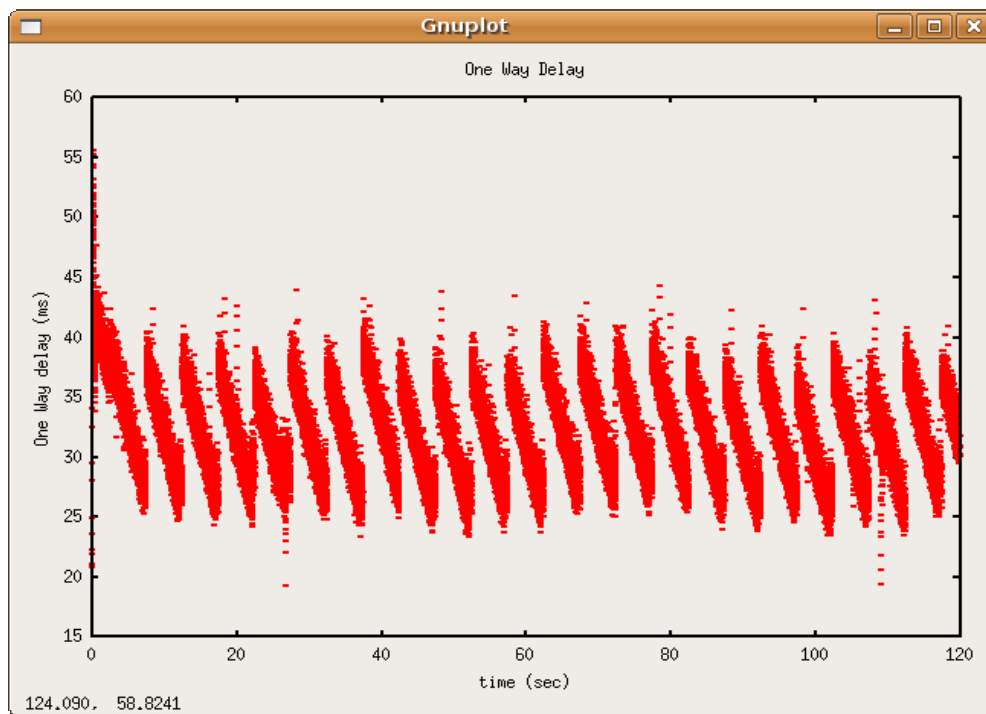
Σχήμα 35 : Γραφική παράσταση του Jitter.



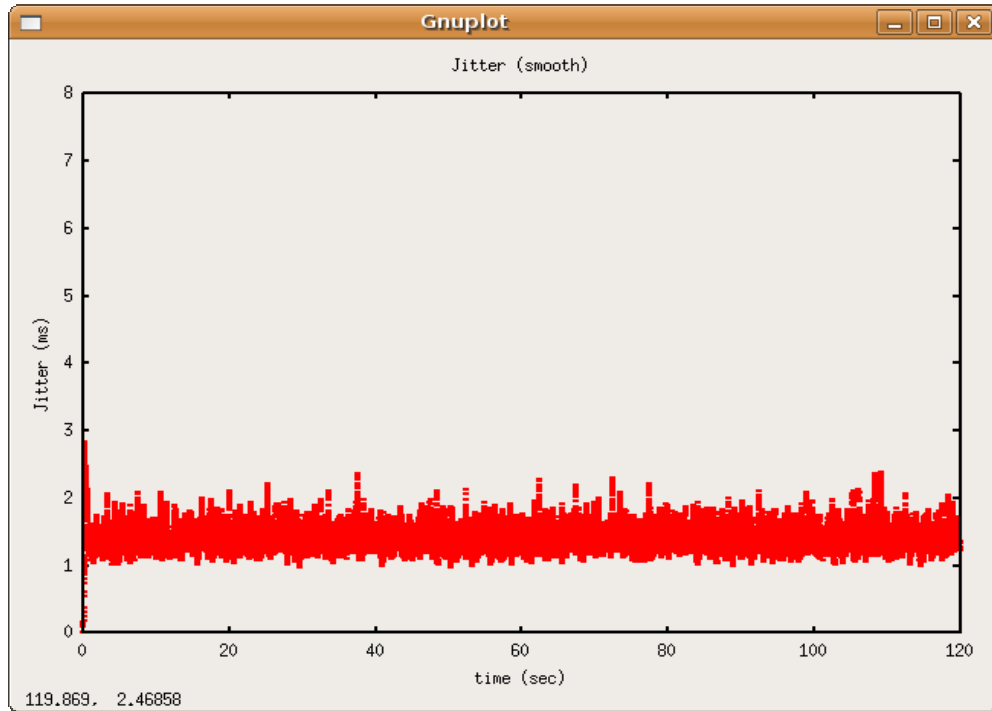
Σχήμα 36 : Γραφική παράσταση των απωλειών.

4.2.3 Μέρος 3^ο

Κατά την πειραματική μέτρηση ενεργοποιείται ο μηχανισμός DiffServ και στο δίκτυο μεταδίδεται UDP κίνηση με ρυθμό μετάδοσης 3 Mbps, παράλληλα με background κίνηση των 7 Mbps. Από την ανάλυση των μετρήσεων προκύπτει ότι η μετάδοσης έλαβε το επιθυμητό throughput στα 3.08 Mbps, με 0% απώλειες, με one way delay στα 31.96 ms και με την τιμή του jitter στα 1.42 ms. Παρατηρώντας τα αποτελέσματα φαίνεται η θετική συνεισφορά του μηχανισμού Qos σε σχέση με το προηγούμενο πείραμα, διασφαλίζοντας την ποιότητα στην μετάδοση και επιτυγχάνοντας απόλυτα ικανοποιητικές τιμές στα χαρακτηριστικά που καθορίζουν την ποιότητα. Ακολουθούν οι γραφικές παραστάσεις της one way delay και του jitter.



Σχήμα 37 : Γραφική παράσταση της One Way Delay.



Σχήμα 38 : Γραφική παράσταση του Jitter.

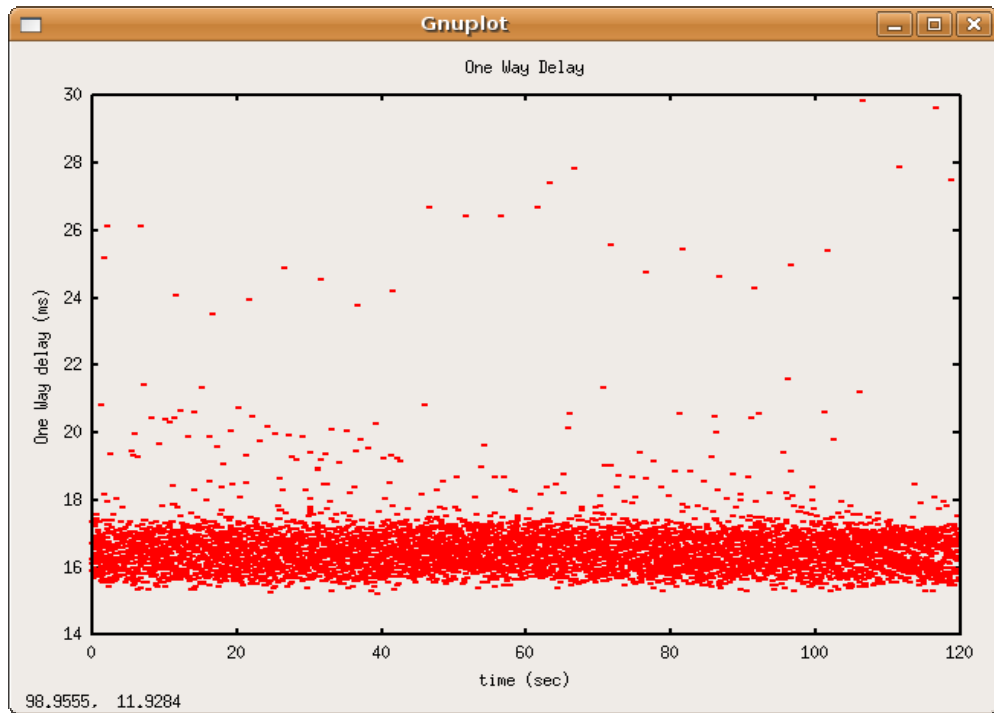
4.3 Σενάριο 3°

4.3.1 Μέρος 1°

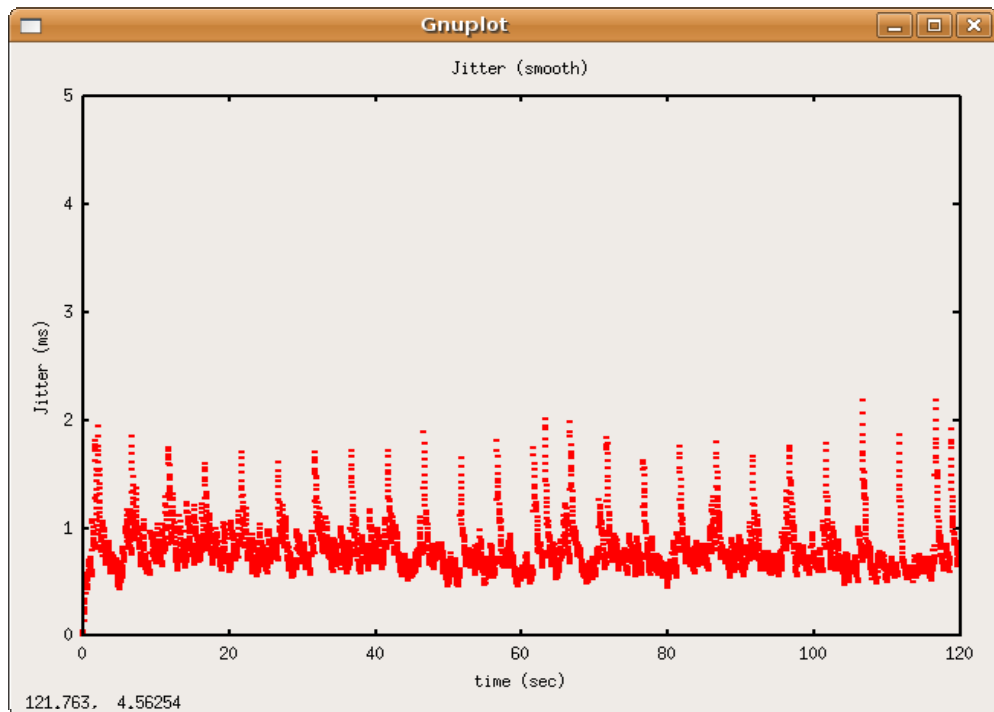
Κατά την πειραματική μέτρηση στο δίκτυο εγκαθιδρυόταν μια αμφίδρομη επικοινωνία VoIP μεταξύ του τελικού χρήστη και του Service Provider. Αναλύοντας τις δύο κινήσεις ξεχωριστά προκύπτει :

- Από 172.16.0.5 (Provider) -> 192.168.0.5 (χρήστης) :

Μετρήθηκε ότι το throughput είναι 24 Kbps, όσο δηλαδή απαιτεί η μία μόνο κίνηση της αμφίδρομης επικοινωνίας, ενώ δεν παρατηρήθηκαν απώλειες. Η τιμή της one way delay είναι 16.55 ms και του jitter 0.8 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.

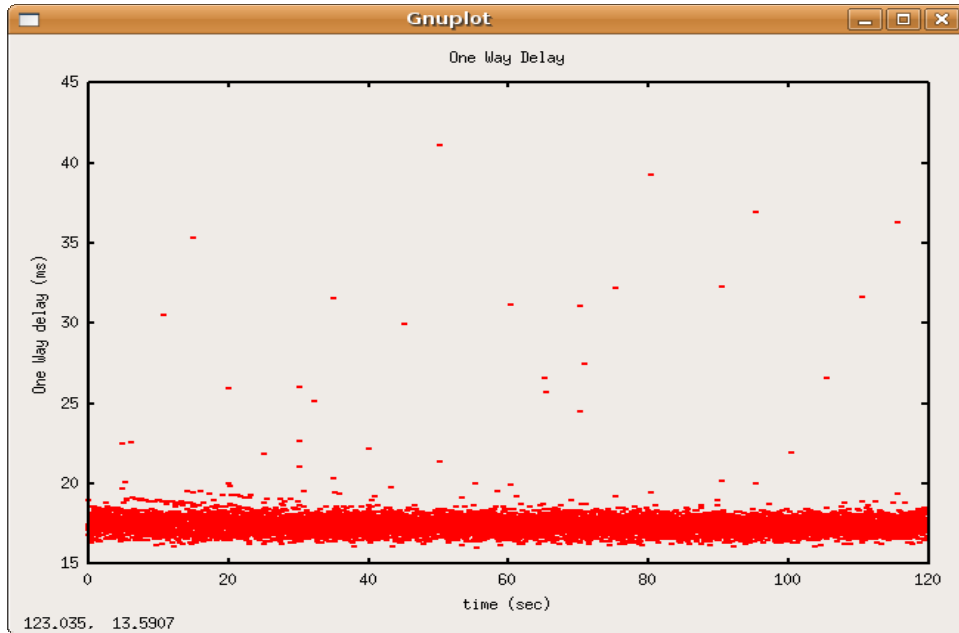


Σχήμα 39 : Γραφική παράσταση της One Way Delay.

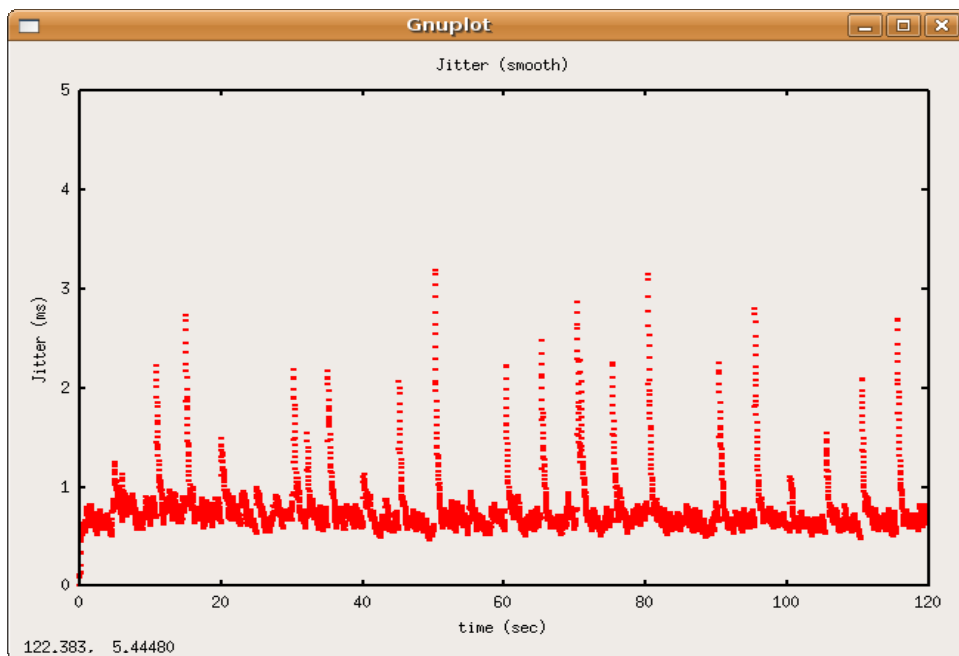


Σχήμα 40 : Γραφική παράσταση του Jitter.

- Από 192.168.0.5 (χρήστης) -> 172.16.0.5 (Provider) :
Μετρήθηκε ότι το throughput είναι 24 Kbps, δεν παρατηρήθηκαν απώλειες, η τιμή της one way delay είναι 17.45 ms και του jitter 0.77 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.



Σχήμα 41 : Γραφική παράσταση της One Way Delay.



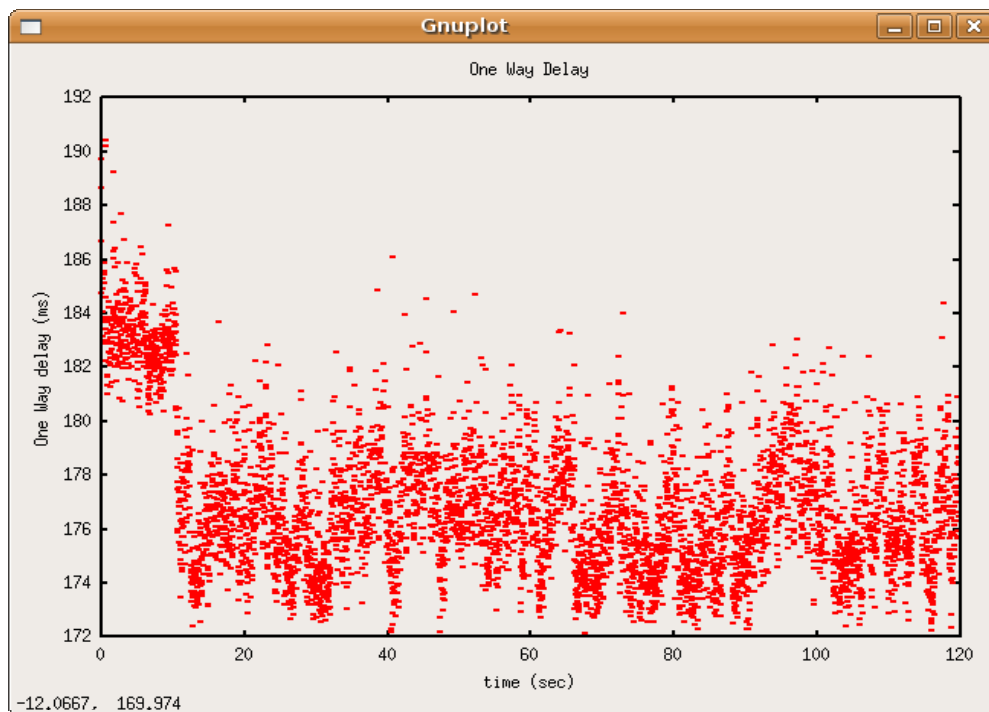
Σχήμα 42 : Γραφική παράσταση του Jitter.

4.3.2 Μέρος 2^ο

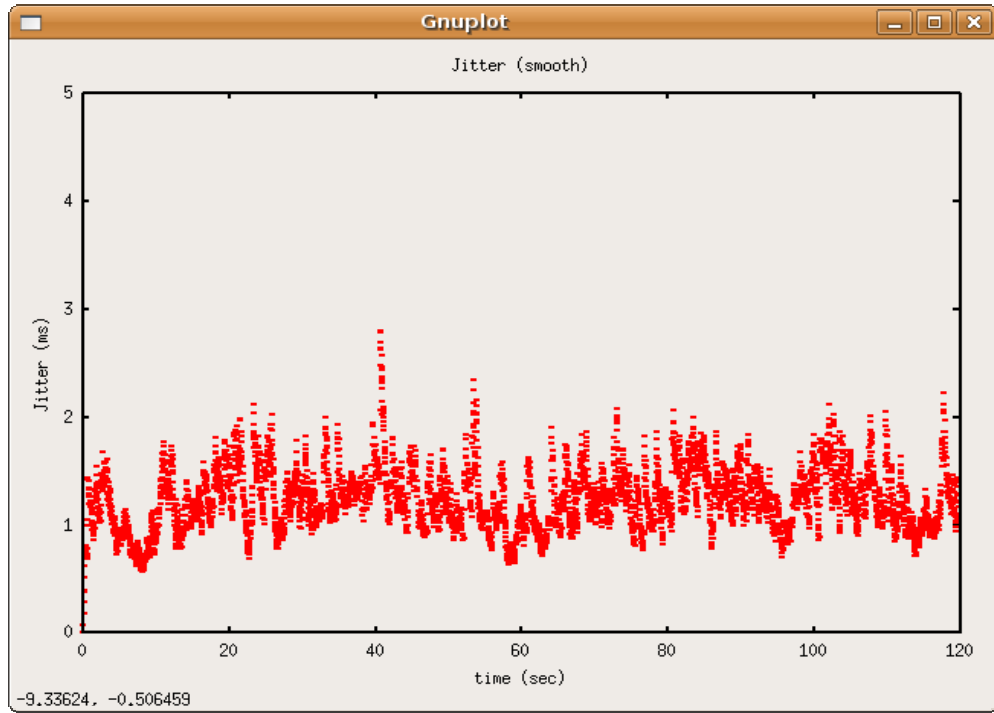
Κατά την πειραματική μέτρηση στο δίκτυο εγκαθιδρύοταν μια αμφίδρομη επικοινωνία VoIP μεταξύ του τελικού χρήστη και του Service Provider, παράλληλα με background κίνηση των 7 Mbps. Αναλύοντας τις δύο κινήσεις, από τις οποίες αποτελείται η VoIP επικοινωνία, ξεχωριστά προκύπτει :

- Από 172.16.0.5 (Provider) -> 192.168.0.5 (χρήστης) :

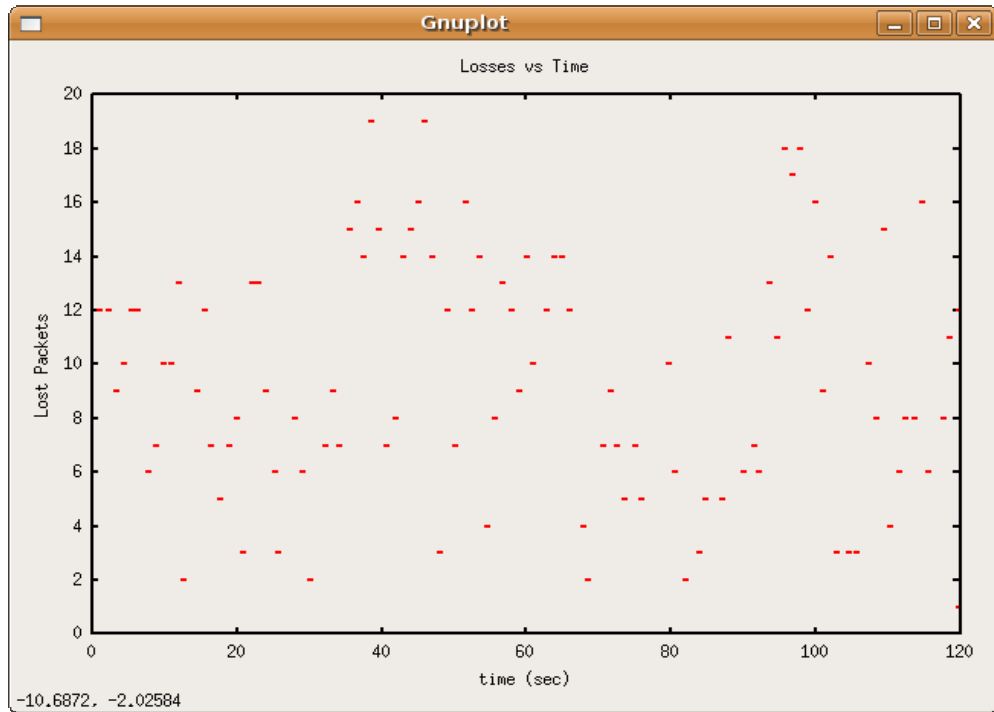
Μετρήθηκε ότι το throughput είναι 20.15 Kbps, παρατηρήθηκαν απώλειες της τάξης του 16.06%, η τιμή της one way delay είναι 176.9 ms και του jitter 1.25 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.



Σχήμα 43 : Γραφική παράσταση της One Way Delay.

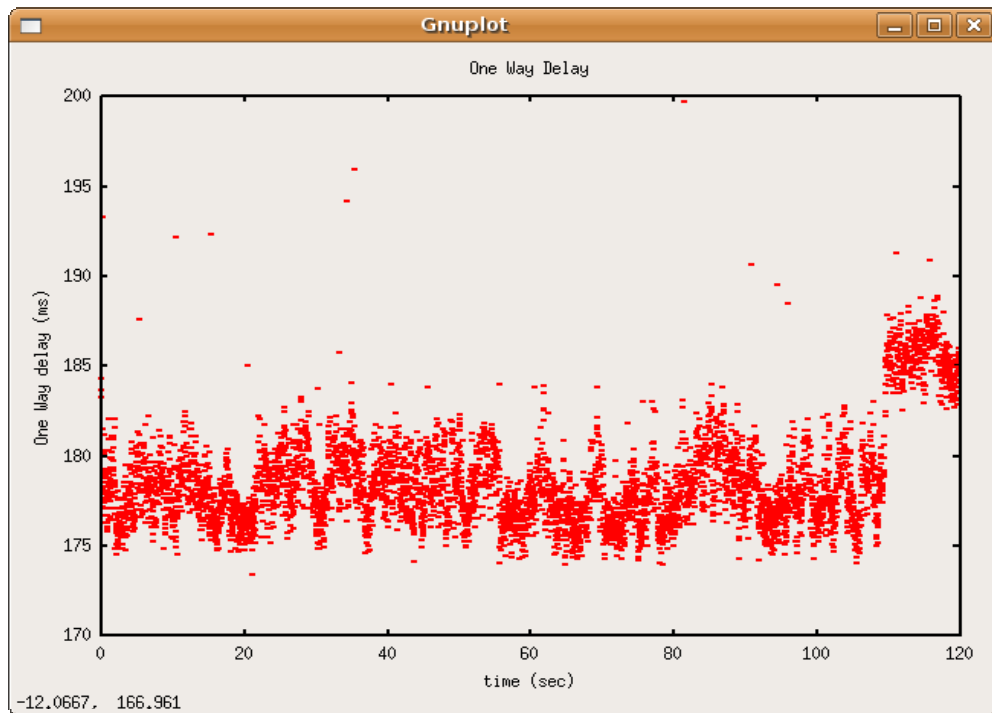


Σχήμα 44 : Γραφική παράσταση του Jitter.

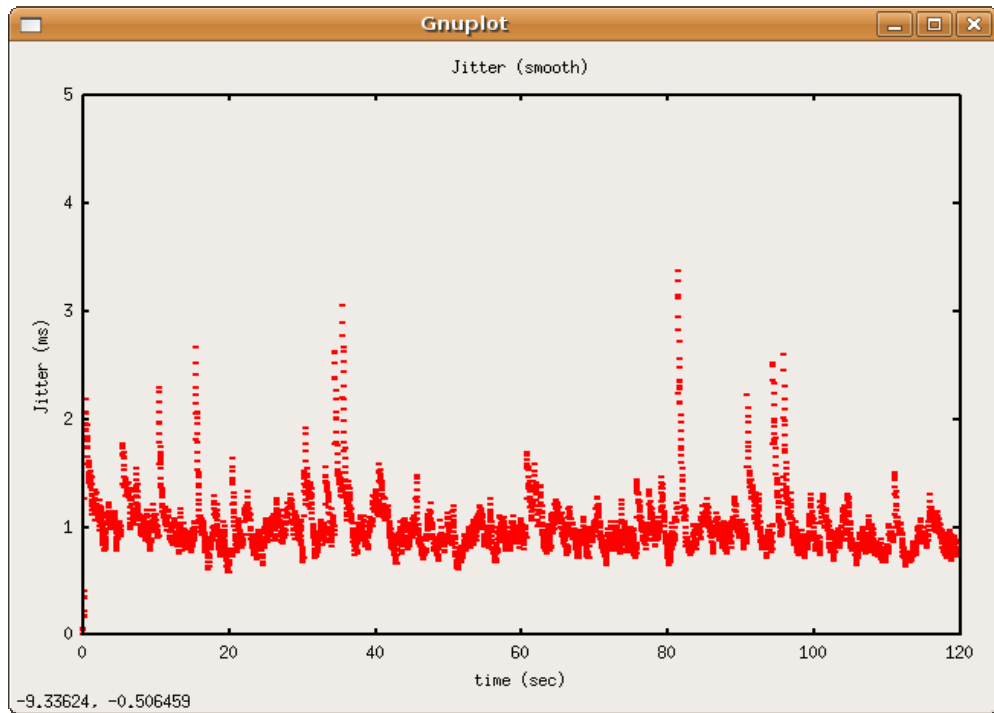


Σχήμα 45 : Γραφική παράσταση των απωλειών.

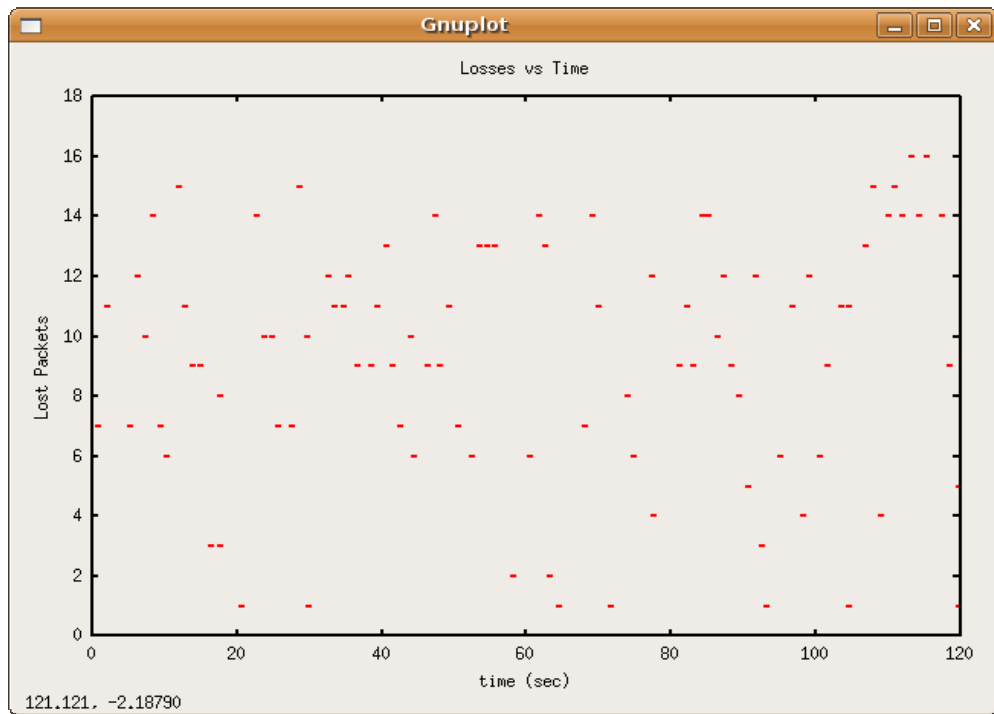
- Από 172.16.0.5 (Provider) -> 192.168.0.5 (χρήστης) :
Μετρήθηκε ότι το throughput είναι 20.5 Kbps, παρατηρήθηκαν απώλειες της τάξης του 14.65%, η τιμή της one way delay είναι 178.55 ms και του jitter 1.01 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.



Σχήμα 46 : Γραφική παράσταση της One Way Delay.



Σχήμα 47 : Γραφική παράσταση του Jitter.



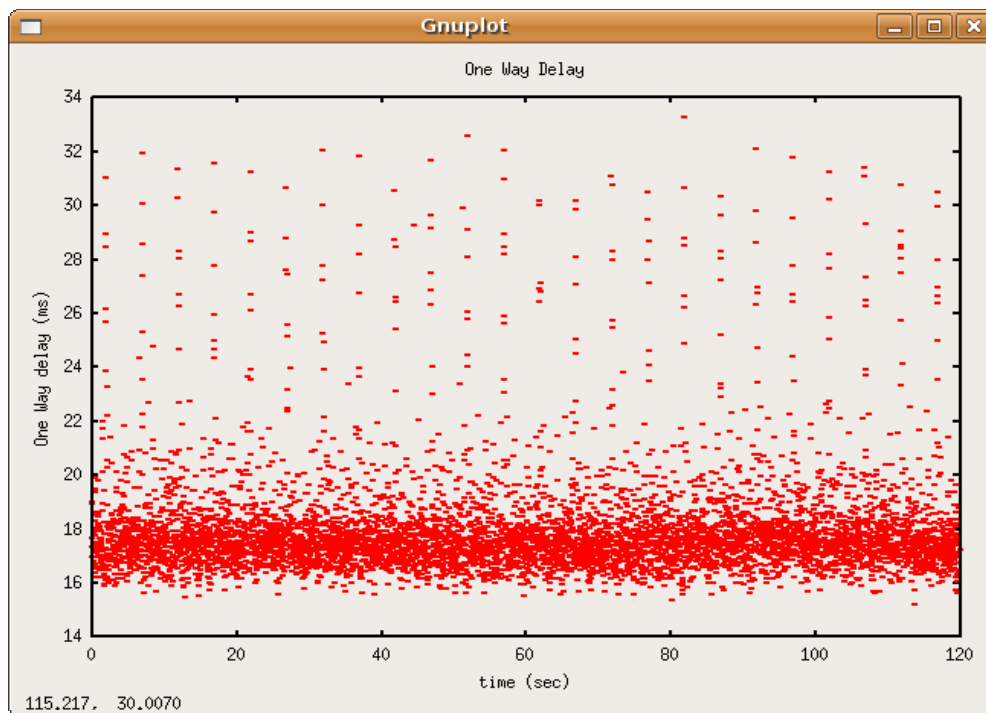
Σχήμα 48 : Γραφική παράσταση των απωλειών.

4.3.3 Μέρος 3^ο

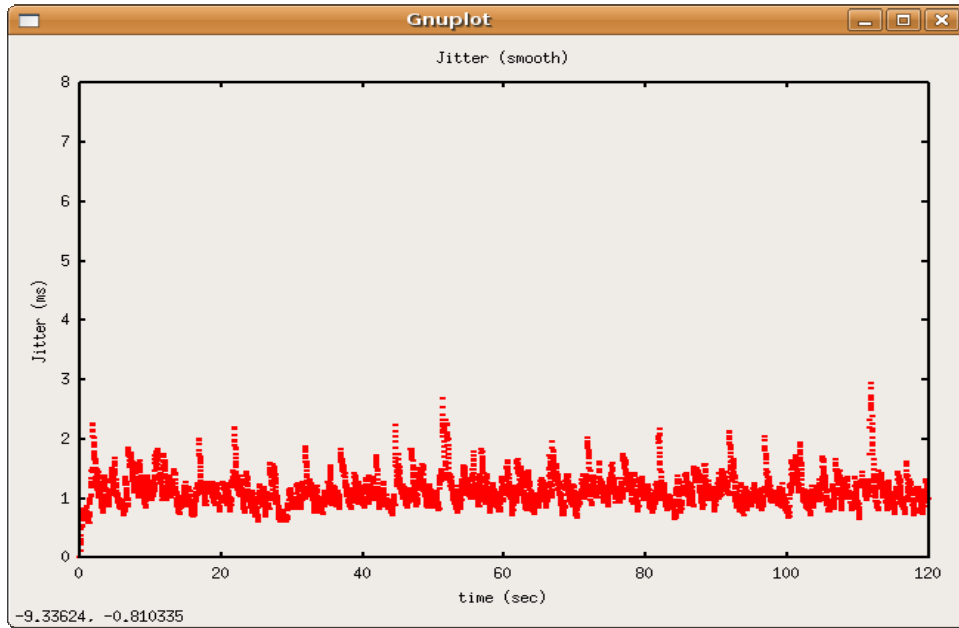
Κατά την πειραματική μέτρηση ενεργοποιείται ο μηχανισμός DiffServ και στο δίκτυο εγκαθιδρύεται μια αμφίδρομη επικοινωνία VoIP μεταξύ του τελικού χρήστη και του Service Provider, παράλληλα με background κίνηση των 7 Mbps. Αναλύοντας τις δύο κινήσεις, από τις οποίες αποτελείται η VoIP επικοινωνία, ξεχωριστά προκύπτει :

- Από 172.16.0.5 (Provider) -> 192.168.0.5 (χρήστης) :

Μετρήθηκε ότι το throughput είναι 24Kbps, δεν παρατηρήθηκαν απώλειες, η τιμή της one way delay είναι 17.97 ms και του jitter 1.13 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.

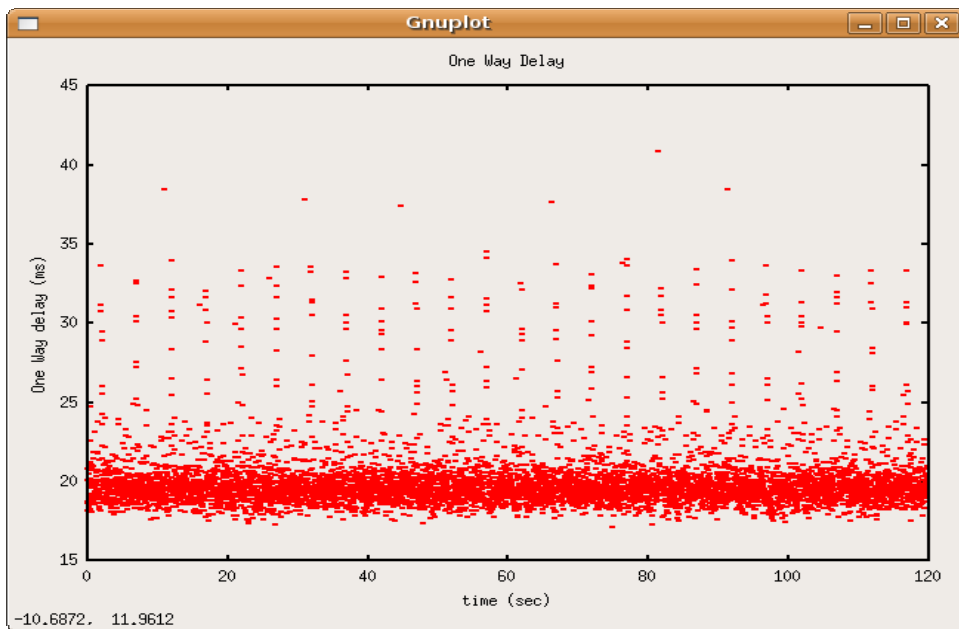


Σχήμα 49 : Γραφική παράσταση της One Way Delay.

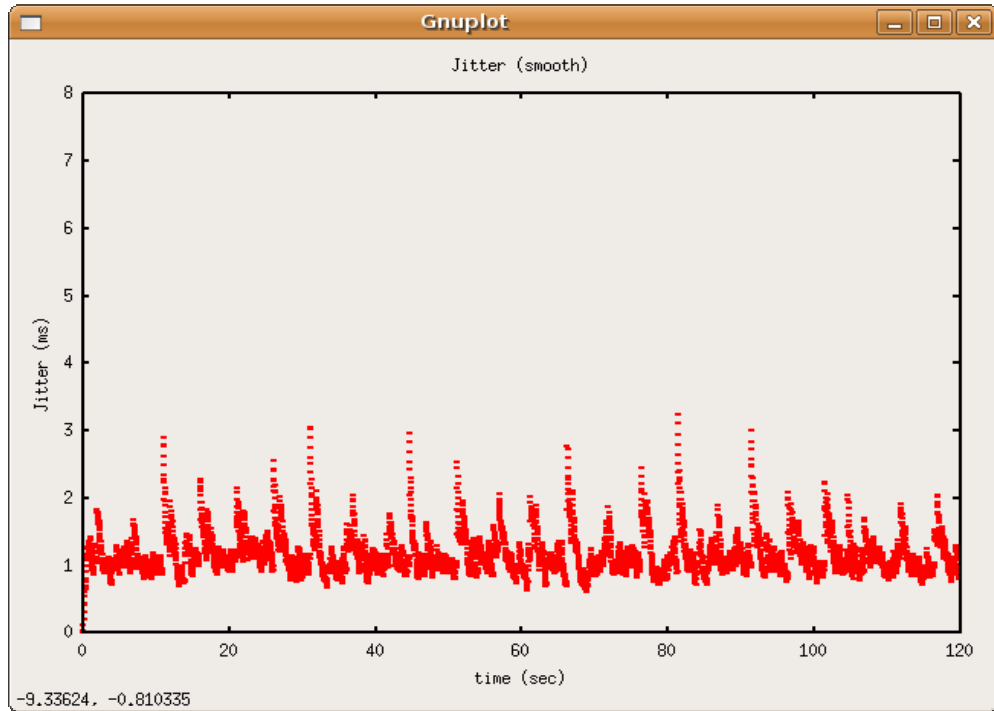


Σχήμα 50 : Γραφική παράσταση του Jitter.

- Από 172.16.0.5 (Provider) -> 192.168.0.5 (χρήστης) :
Μετρήθηκε ότι το throughput είναι 24 Kbps, δεν παρατηρήθηκαν απώλειες, η τιμή της one way delay είναι 20.07 ms και του jitter 1.17 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.



Σχήμα 51 : Γραφική παράσταση της One Way Delay.



Σχήμα 52 : Γραφική παράσταση του Jitter.

4.4 Σενάριο 4°

4.4.1 Μέρος 1°

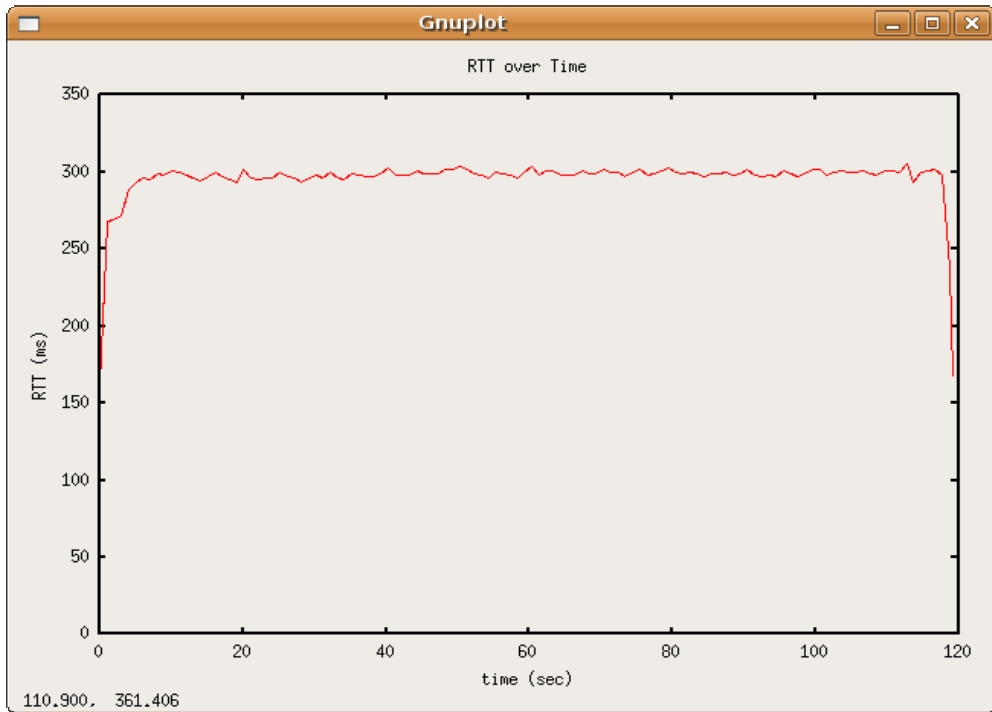
Κατά την πειραματική μέτρηση στο δίκτυο μεταδιδόταν ταυτόχρονα μια TCP κίνηση, μια UDP κίνηση με ρυθμό μετάδοσης 3 Mbps και εγκαθιδρυόταν μια αμφίδρομη επικοινωνία VoIP. Θα γίνει ανάλυση και παρουσίαση των αποτελεσμάτων για κάθε κίνηση ξεχωριστά.

4.4.1.1 TCP κίνηση

Η κίνηση έλαβε throughput 3.88 Mbps, με καμία επαναμετάδοση, που σημαίνει μηδενικές απώλειες και RTT καθυστέρηση στα 296 ms. Παρακάτω παρουσιάζονται γραφικές παραστάσεις του throughput και της RTT καθυστέρησης.



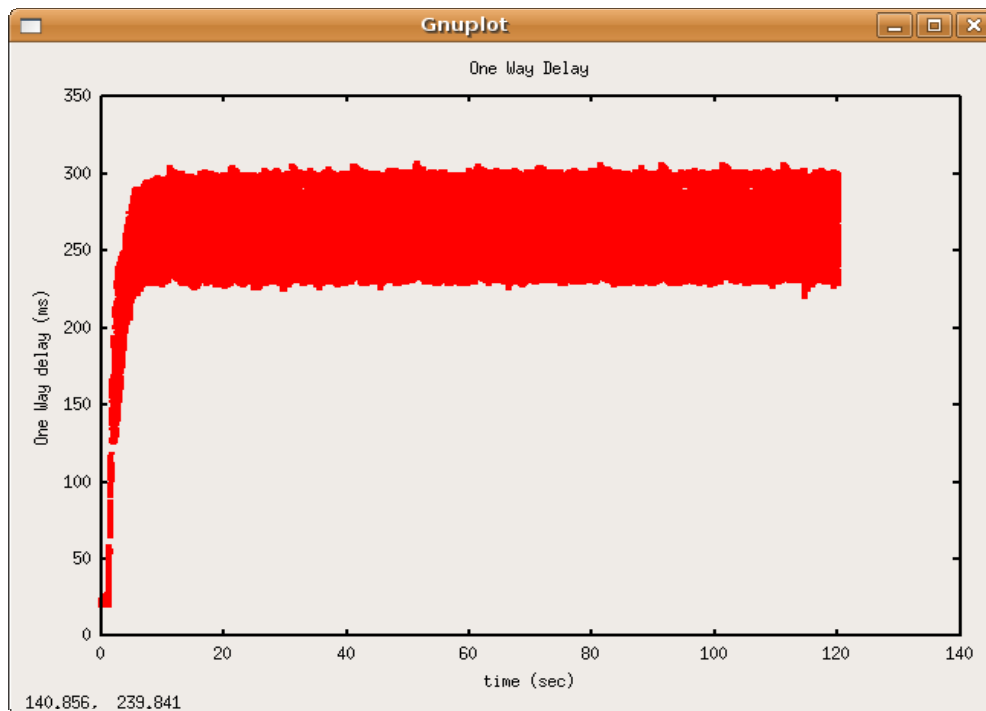
Σχήμα 53 : Γραφική παράσταση του Throughput.



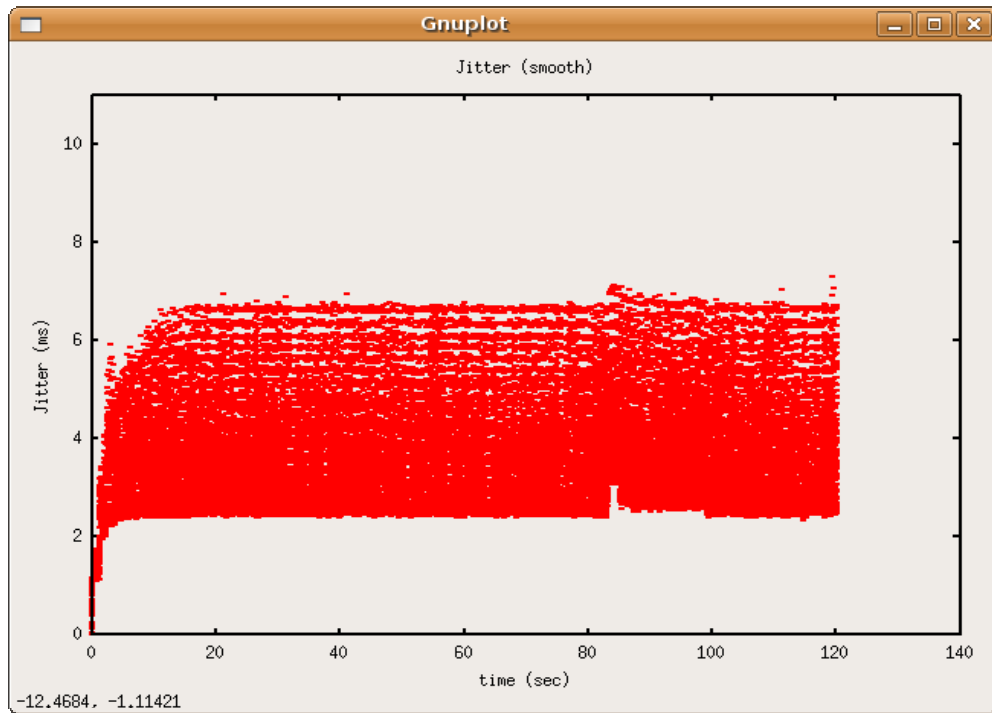
Σχήμα 54 : Γραφική παράσταση του RTT.

4.4.1.2 UDP κίνηση

Η προσομοίωση μετάδοσης video έλαβε 3.07 Mbps throughput, χωρίς απώλειες, με one way delay στα 260.31 ms και με την τιμή του jitter στα 3.64 ms. Από τις τιμές, αν εξαιρεθεί η σχετικά μεγάλη τιμή της one way καθυστέρησης, βγαίνει το συμπέρασμα ότι η μετάδοση δεν είχε ιδιαίτερα προβλήματα από την παρουσία των άλλων κινήσεων στο δίκτυο. Ακολουθούν οι γραφικές παραστάσεις της one way delay και του jitter.



Σχήμα 55 : Γραφική παράσταση της One Way Delay.



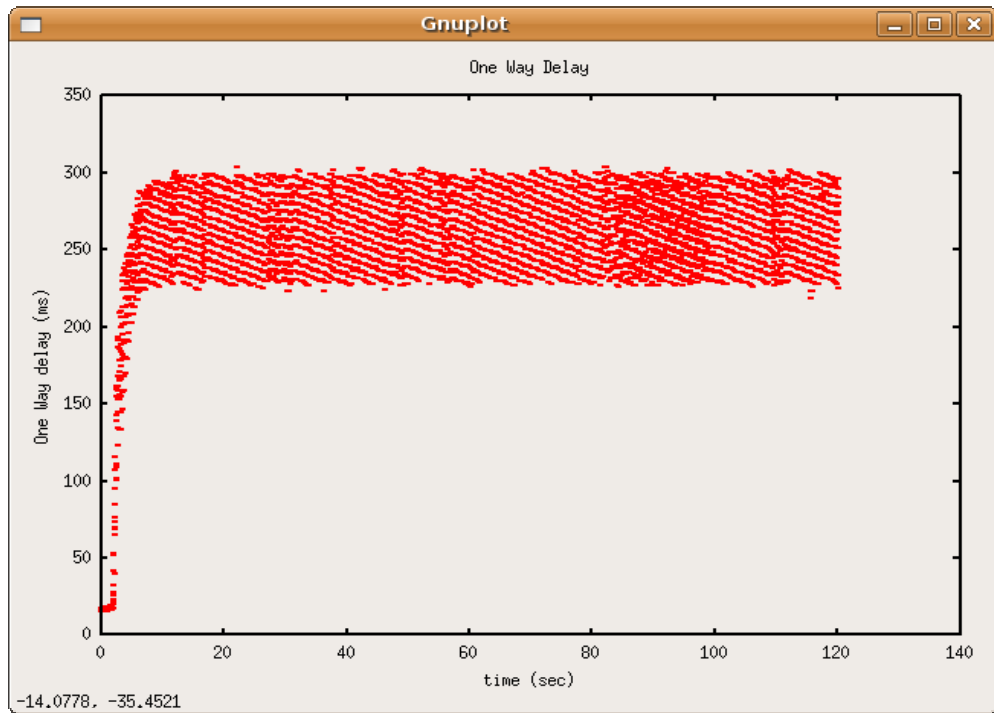
Σχήμα 56 : Γραφική παράσταση του Jitter.

4.4.1.3 VoIP επικοινωνία

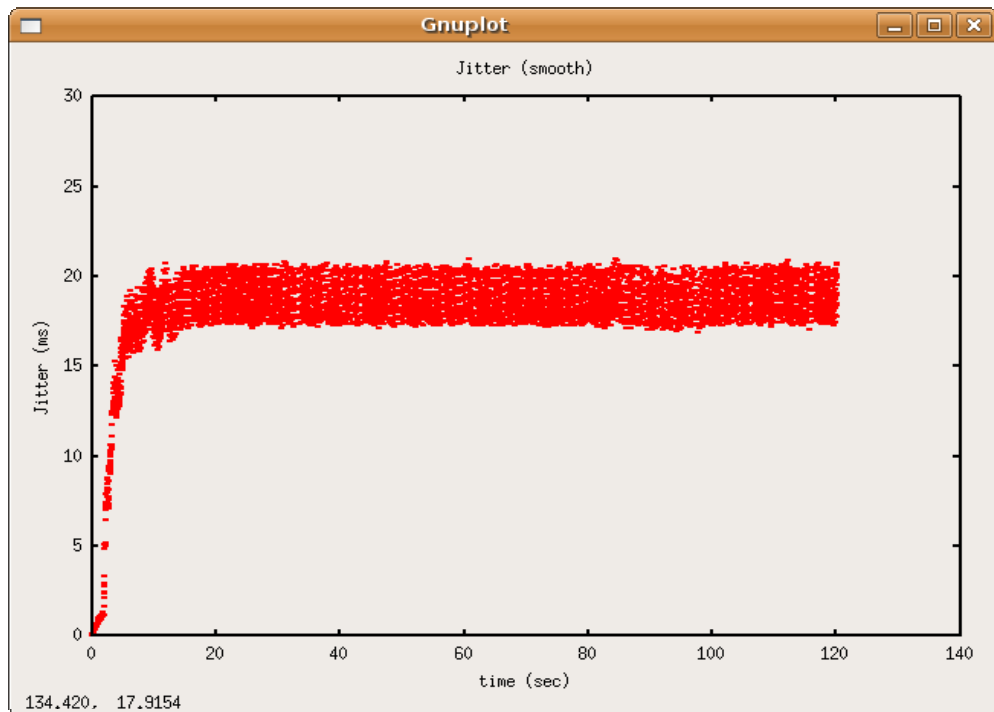
Αναλύοντας τις δύο κινήσεις, από τις οποίες αποτελείται η VoIP επικοινωνία, ξεχωριστά προκύπτει :

- Από 172.16.0.5 (Provider) -> 192.168.0.5 (χρήστης) :

Μετρήθηκε ότι το throughput είναι 23.95 Kbps, δεν παρατηρήθηκαν απώλειες, η τιμή της one way delay είναι 257.32 ms και του jitter 18.34 ms. Από τις παραπάνω τιμές η μετάδοση αυτή θα χαρακτηριστεί μέτρια. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.

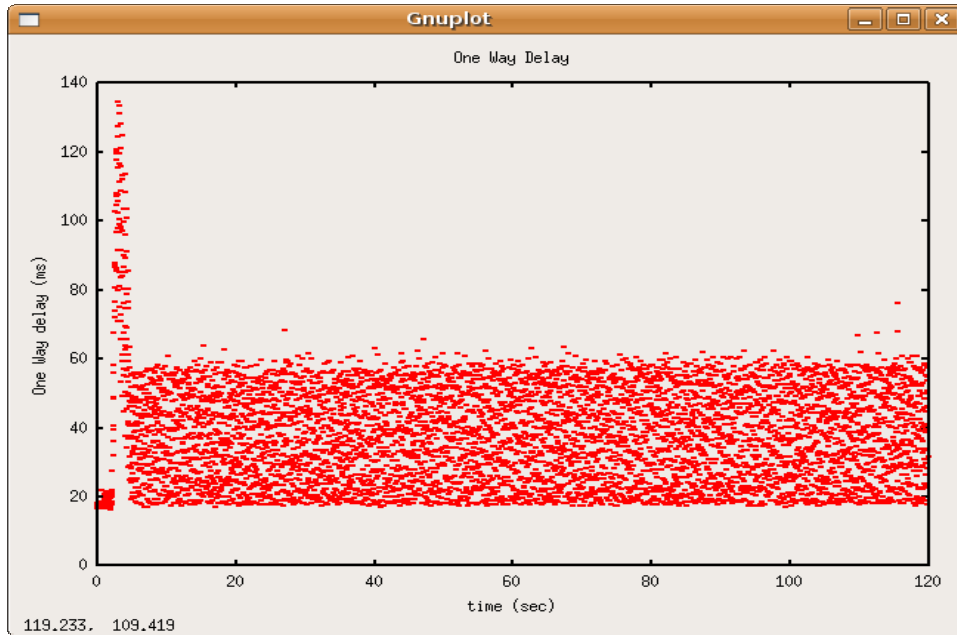


Σχήμα 57 : Γραφική παράσταση της One Way Delay.

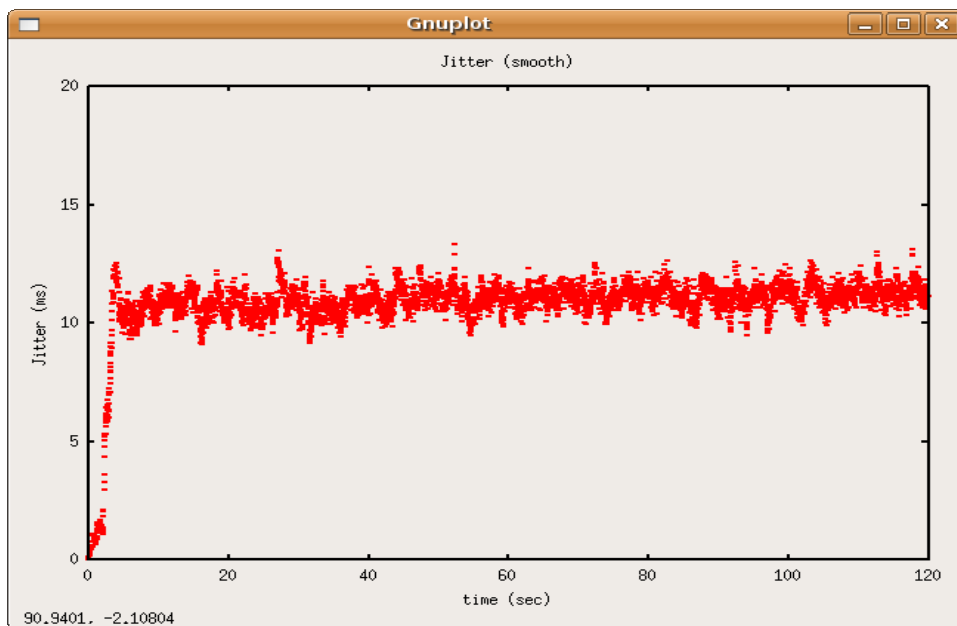


Σχήμα 58 : Γραφική παράσταση του Jitter.

- Από 192.168.0.5 (χρήστης) -> 172.16.0.5 (Provider) :
Μετρήθηκε ότι το throughput είναι 24 Kbps, δεν παρατηρήθηκαν απώλειες, η τιμή της one way delay είναι 38.14 ms και του jitter 10.83 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.



Σχήμα 59 : Γραφική παράσταση της One Way Delay.



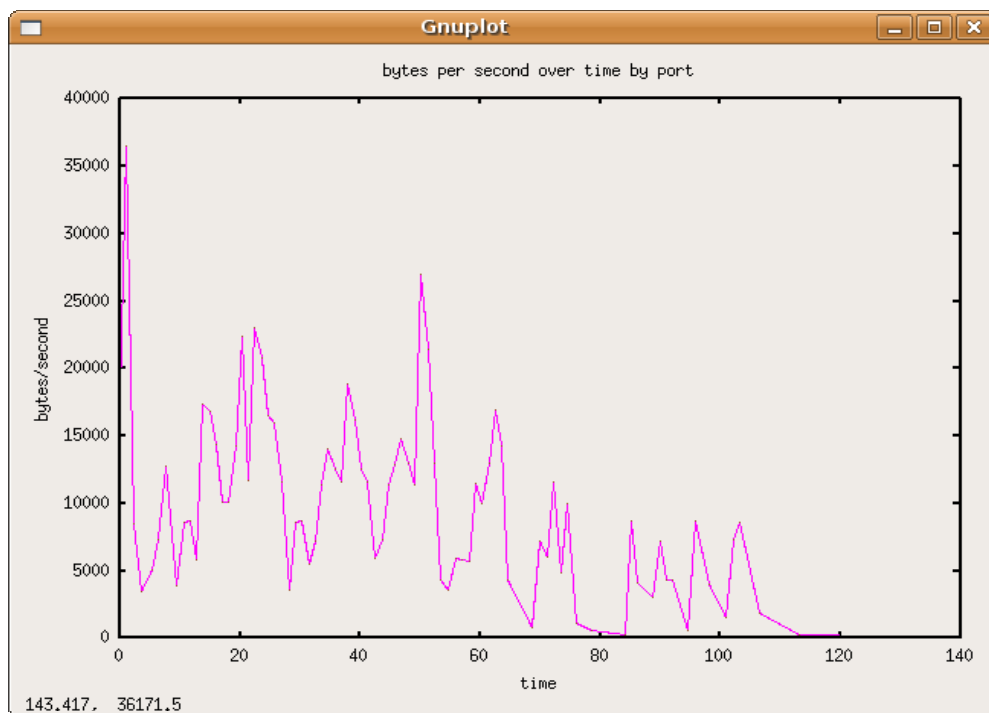
Σχήμα 60 : Γραφική παράσταση του Jitter.

4.4.2 Μέρος 2^ο

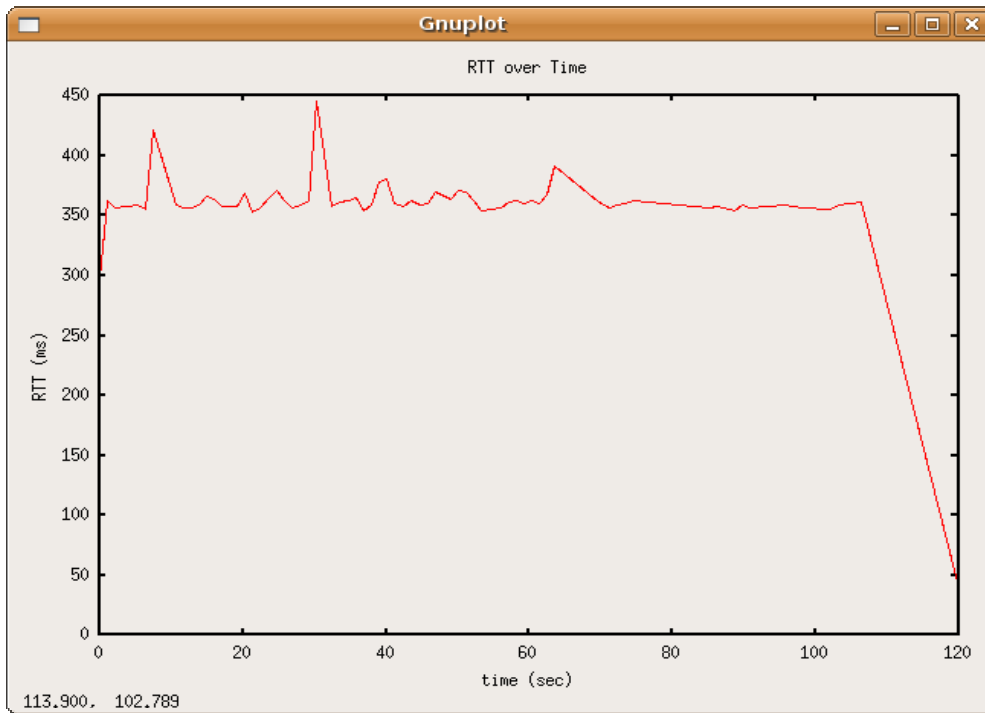
Κατά την πειραματική μέτρηση στο δίκτυο μεταδιδόταν ταυτόχρονα μια TCP κίνηση, μια UDP κίνηση με ρυθμό μετάδοσης 3 Mbps και εγκαθιδρυόταν μια αμφίδρομη επικοινωνία VoIP, παράλληλα με background κίνηση των 5 Mbps. Θα γίνει ανάλυση και παρουσίαση των αποτελεσμάτων για κάθε κίνηση ξεχωριστά.

4.4.2.1 TCP κίνηση

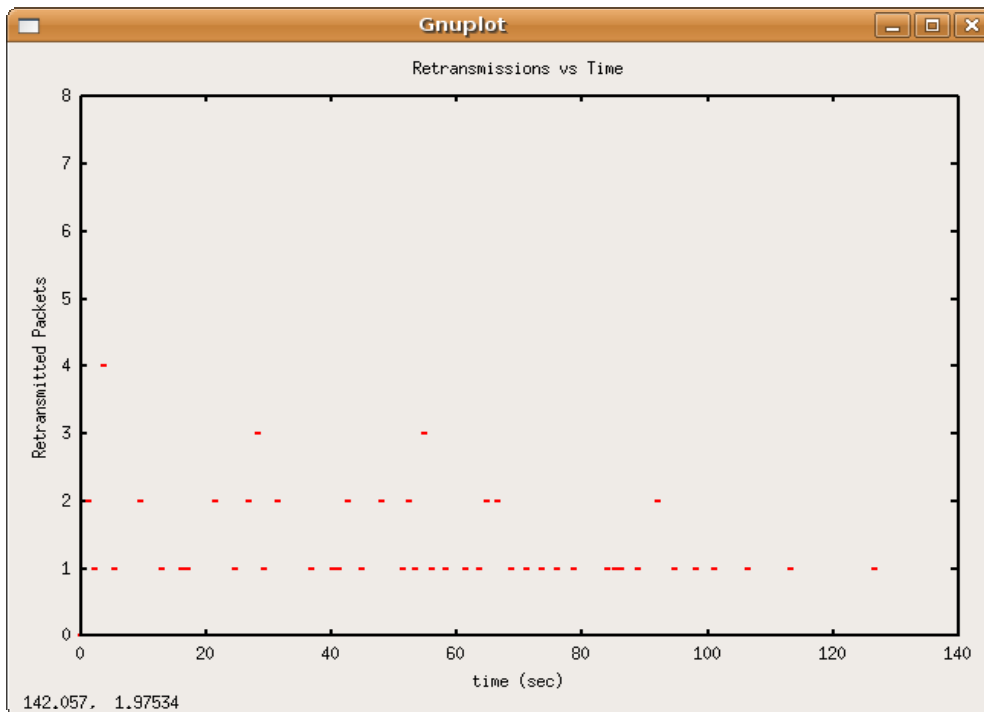
Η κίνηση έλαβε throughput 53.75 Kbps, με 64 επαναμεταδόσεις πακέτων και RTT καθυστέρηση στα 339.5 ms. Παρακάτω παρουσιάζονται γραφικές παραστάσεις του throughput, της RTT καθυστέρησης και των επαναμεταδόσεων.



Σχήμα 61 : Γραφική παράσταση του Throughput.



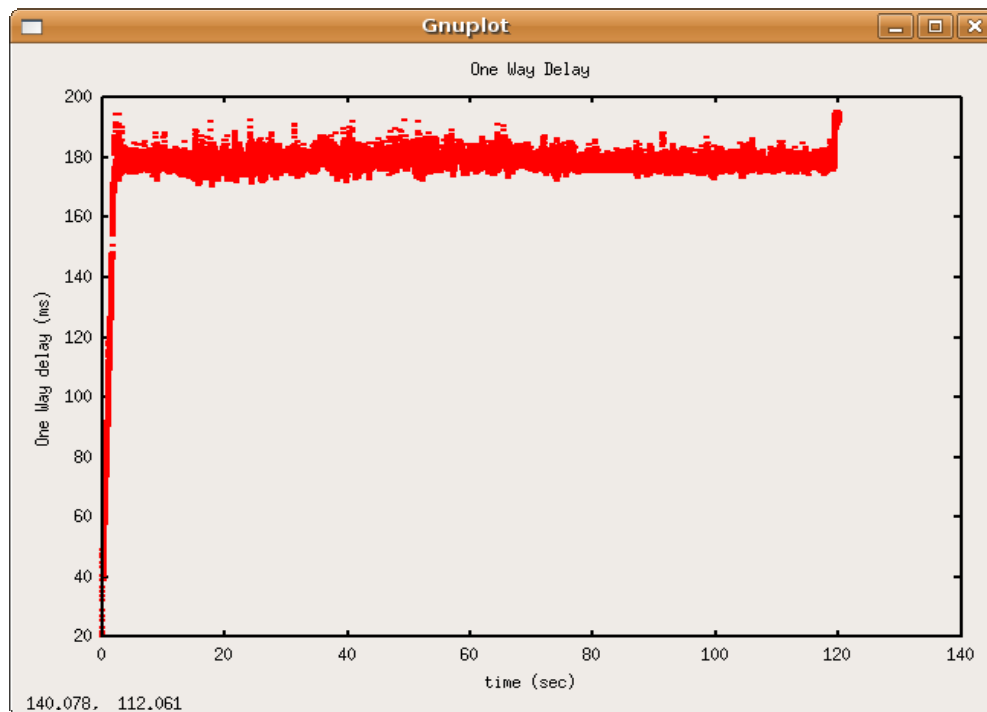
Σχήμα 62 : Γραφική παράσταση του RTT.



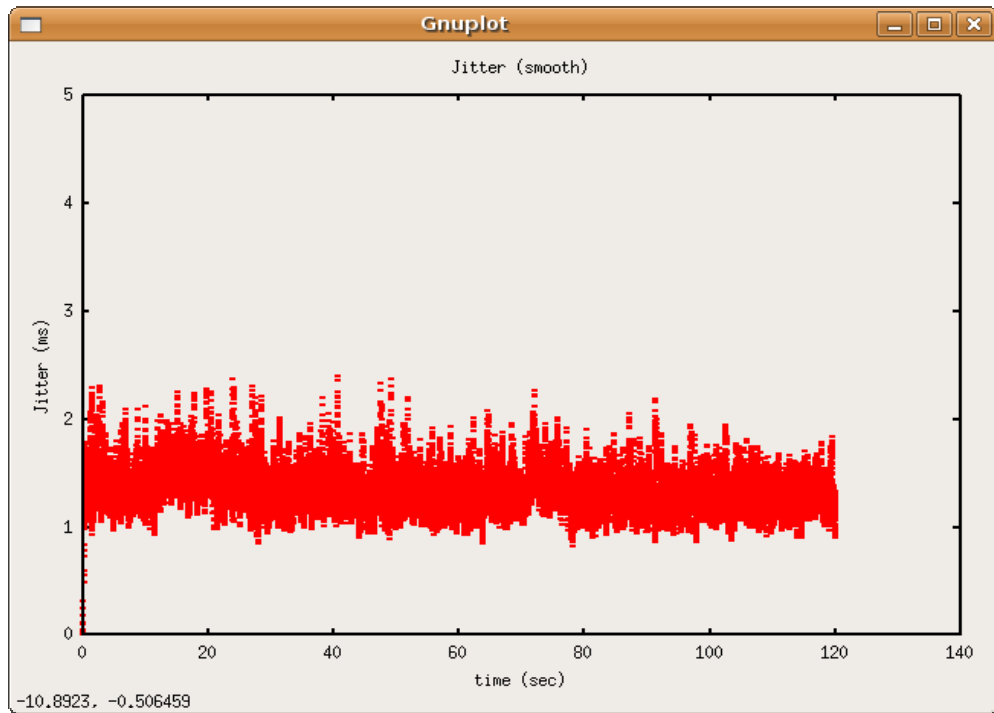
Σχήμα 63 : Γραφική παράσταση των retransmissions.

4.4.2.2 UDP κίνηση

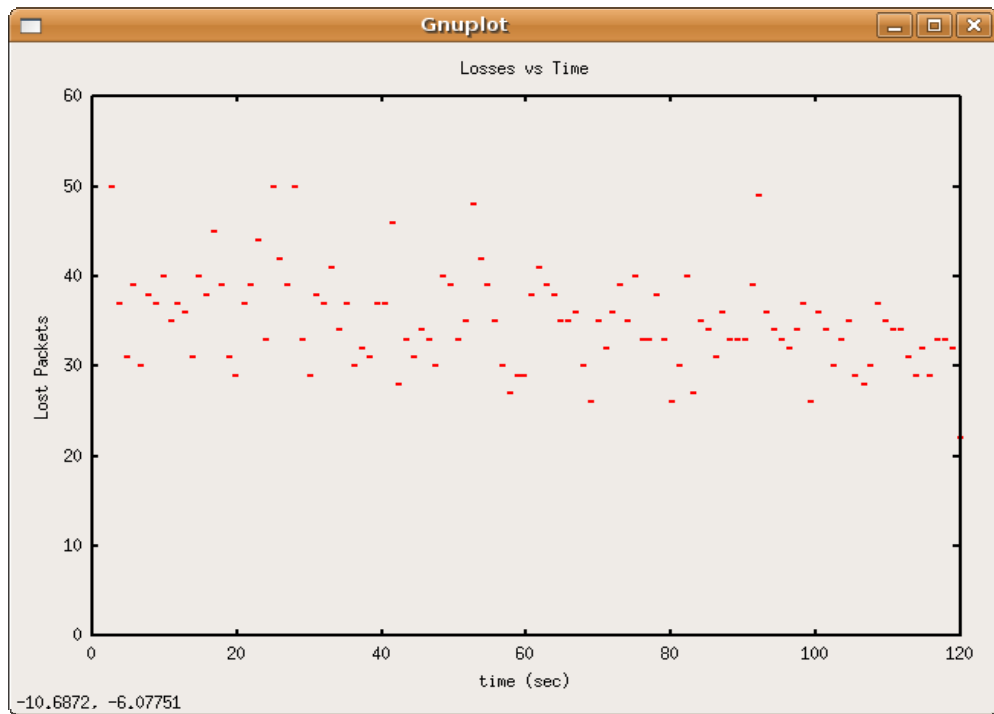
Η προσομοίωση μετάδοσης video έλαβε 2.79 Mbps throughput, με 9.24% απώλειες, με one way delay στα 177.54 ms και με την τιμή του jitter στα 1.35 ms. Παρατηρείται ότι η μετάδοση είχε προβλήματα από την παρουσία των άλλων κινήσεων στο δίκτυο. Ακολουθούν οι γραφικές παραστάσεις της one way delay, του jitter και των απωλειών.



Σχήμα 64 : Γραφική παράσταση της One Way Delay.



Σχήμα 65 : Γραφική παράσταση του Jitter.



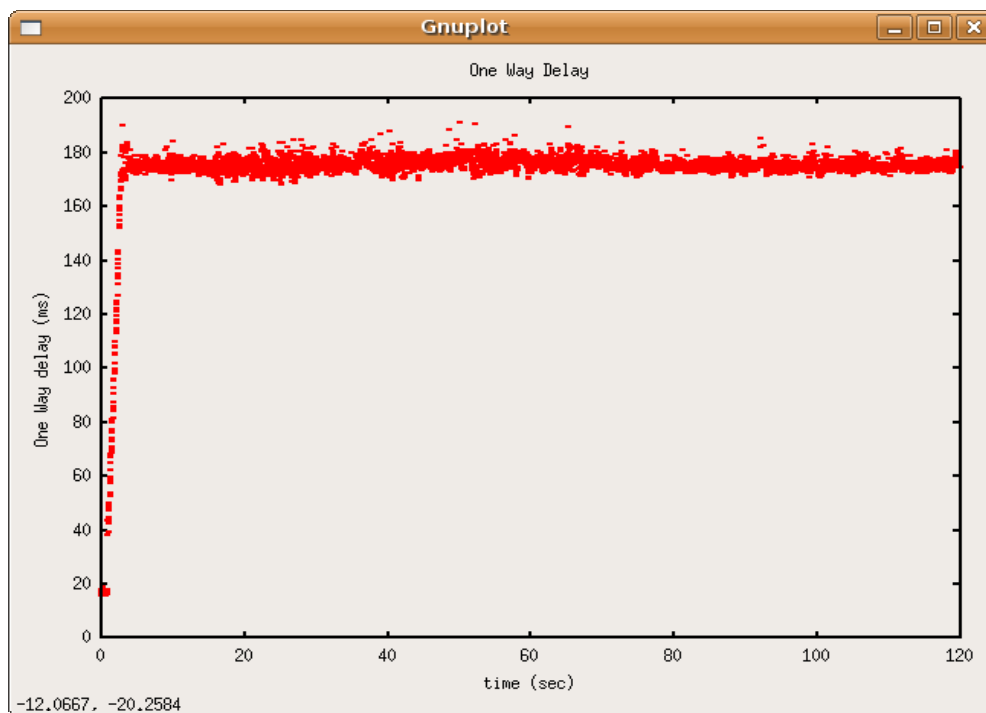
Σχήμα 66 : Γραφική παράσταση των απωλειών.

4.4.2.3 VoIP επικοινωνία

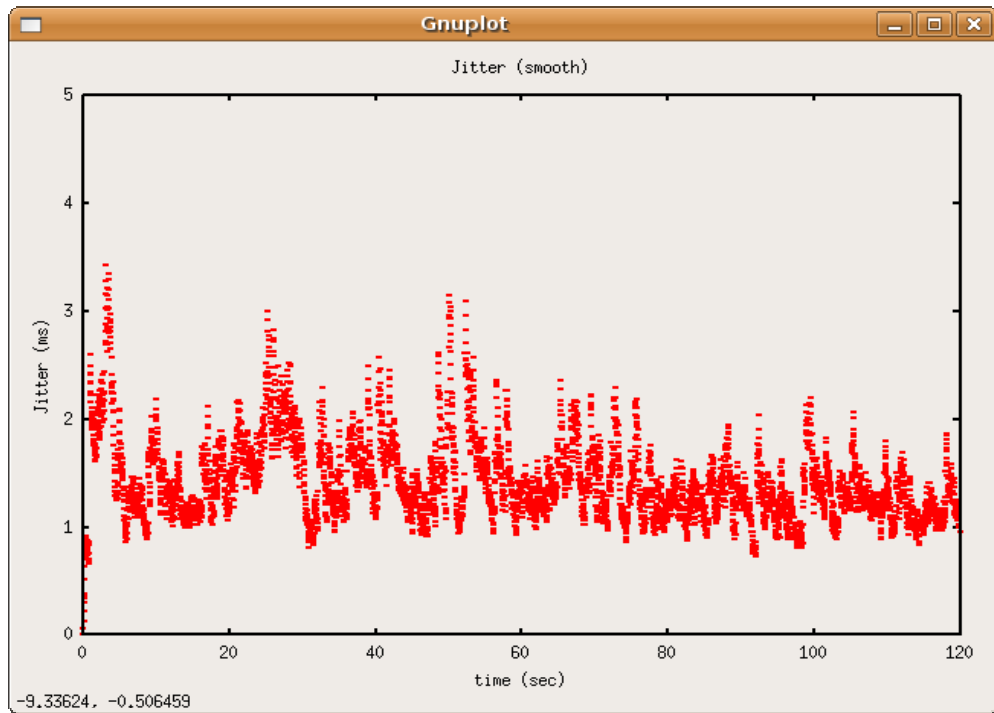
Αναλύοντας τις δύο κινήσεις, από τις οποίες αποτελείται η VoIP επικοινωνία, ξεχωριστά προκύπτει :

- Από 172.16.0.5 (Provider) -> 192.168.0.5 (χρήστης) :

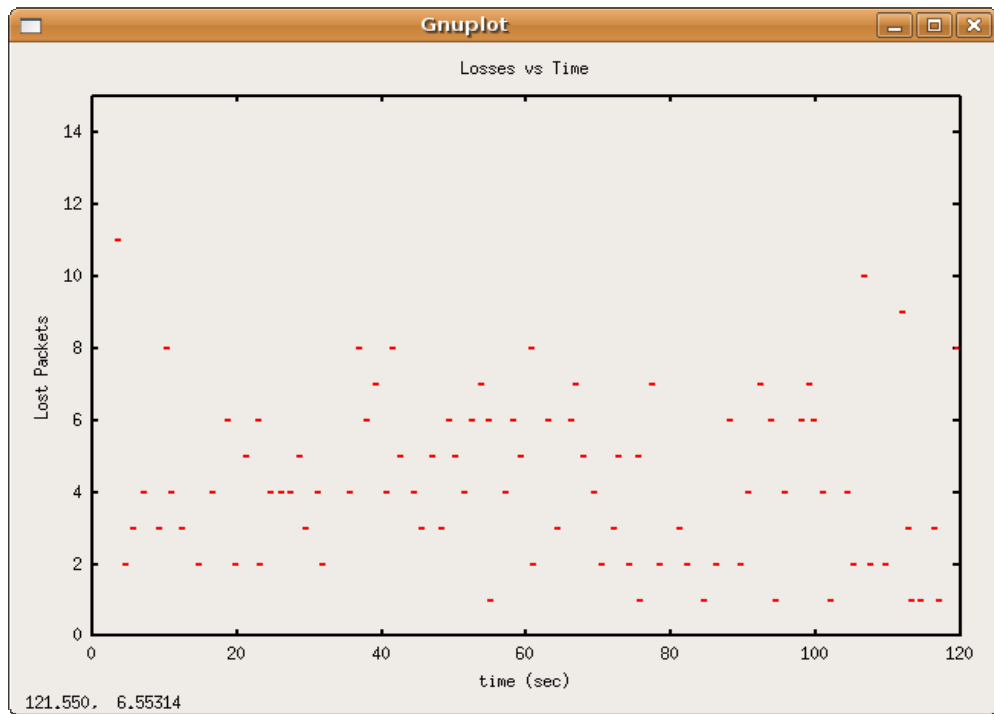
Μετρήθηκε ότι το throughput είναι 22.47 Kbps, με 6.23% απώλειες, η τιμή της one way delay είναι 173.13 ms και του jitter 1.42 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.



Σχήμα 67 : Γραφική παράσταση της One Way Delay.

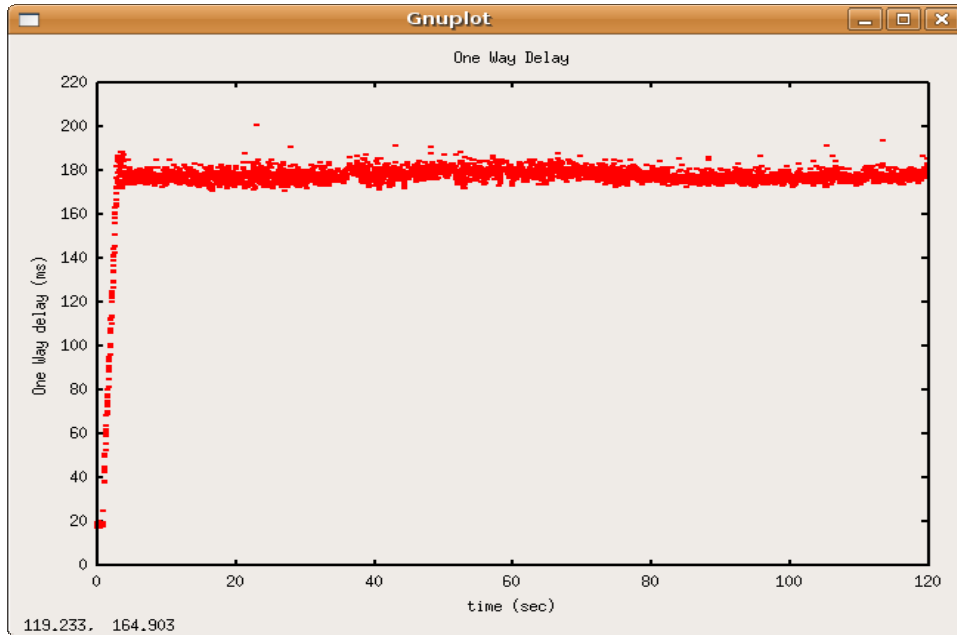


Σχήμα 68 : Γραφική παράσταση του Jitter.

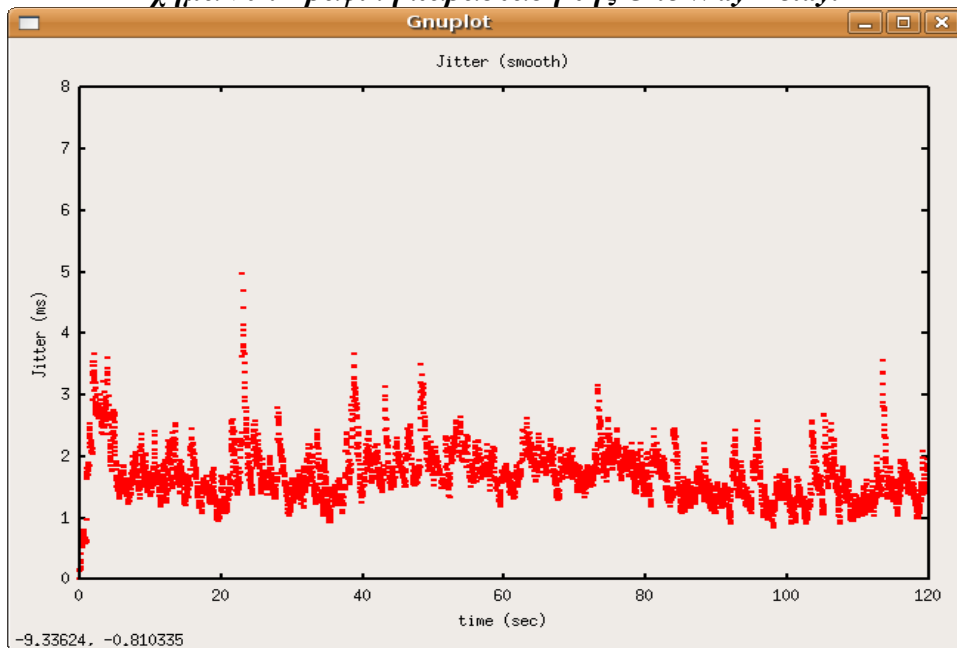


Σχήμα 69 : Γραφική παράσταση των απωλειών.

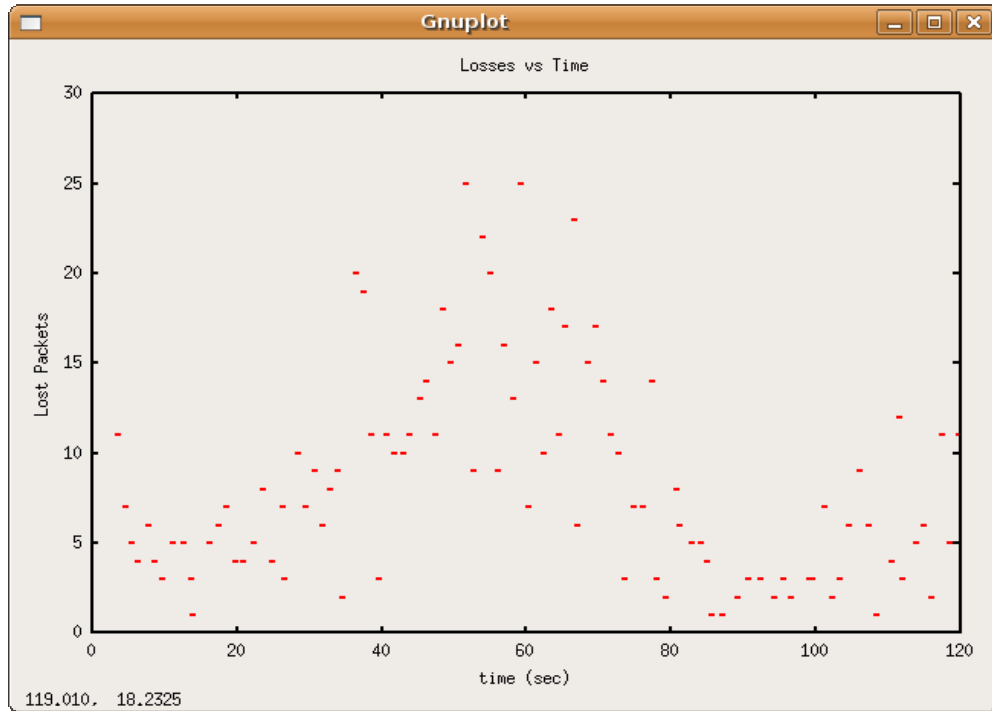
- Από 192.168.0.5 (χρήστης) -> 172.16.0.5 (Provider) :
Μετρήθηκε ότι το throughput είναι 20.64 Kbps, με 13.88% απώλειες, η τιμή της one way delay είναι 174.90 ms και του jitter 1.70 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.



Σχήμα 70 : Γραφική παράσταση της One Way Delay.



Σχήμα 71 : Γραφική παράσταση του Jitter.



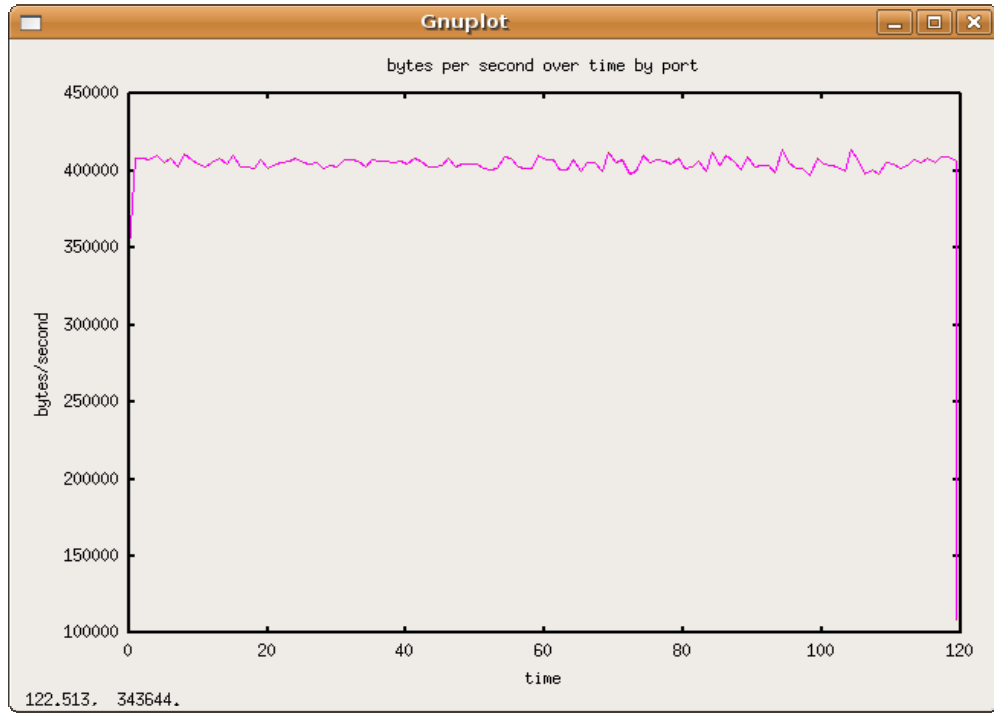
Σχήμα 72 : Γραφική παράσταση των απωλειών.

4.4.3 Μέρος 3^ο

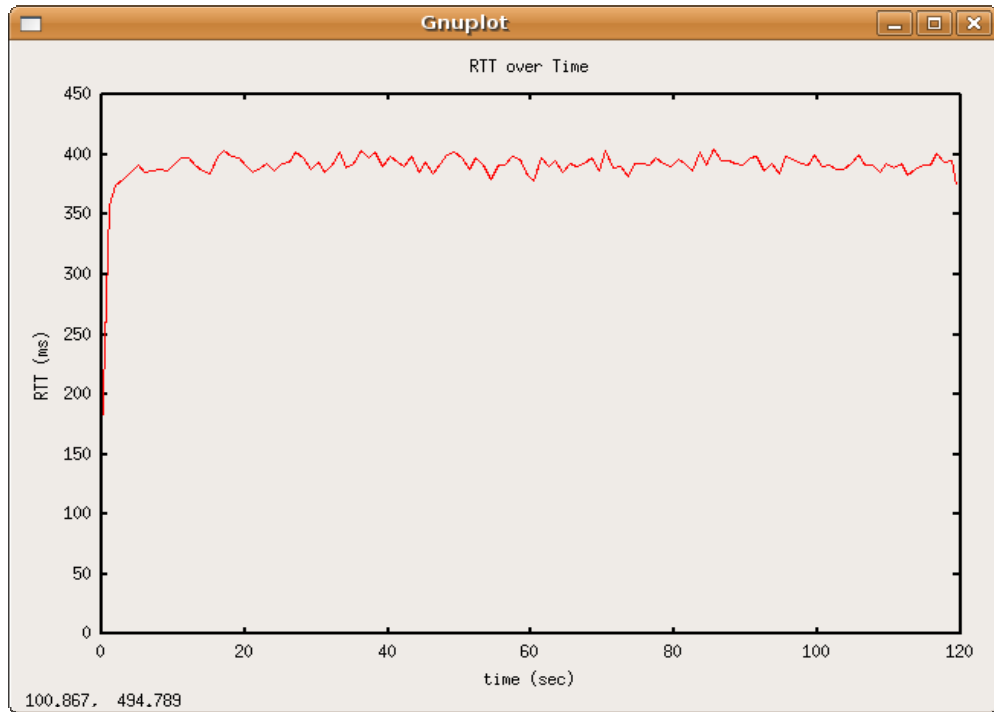
Κατά την πειραματική μέτρηση ενεργοποιείται ο μηχανισμός DiffServ και στο δίκτυο μεταδίδονται ταυτόχρονα μια TCP κίνηση, μια UDP κίνηση με ρυθμό μετάδοσης 3 Mbps και εγκαθιδρύεται μια αμφίδρομη επικοινωνία VoIP, παράλληλα με background κίνηση των 5 Mbps. Θα γίνει ανάλυση και παρουσίαση των αποτελεσμάτων για κάθε κίνηση ξεχωριστά.

4.4.3.1 TCP κίνηση

Η κίνηση έλαβε throughput 3.07 Mbps, με καμία επαναμετάδοση και RTT καθυστέρηση στα 390.9 ms. Παρακάτω παρουσιάζονται γραφικές παραστάσεις του throughput και της RTT καθυστέρησης.



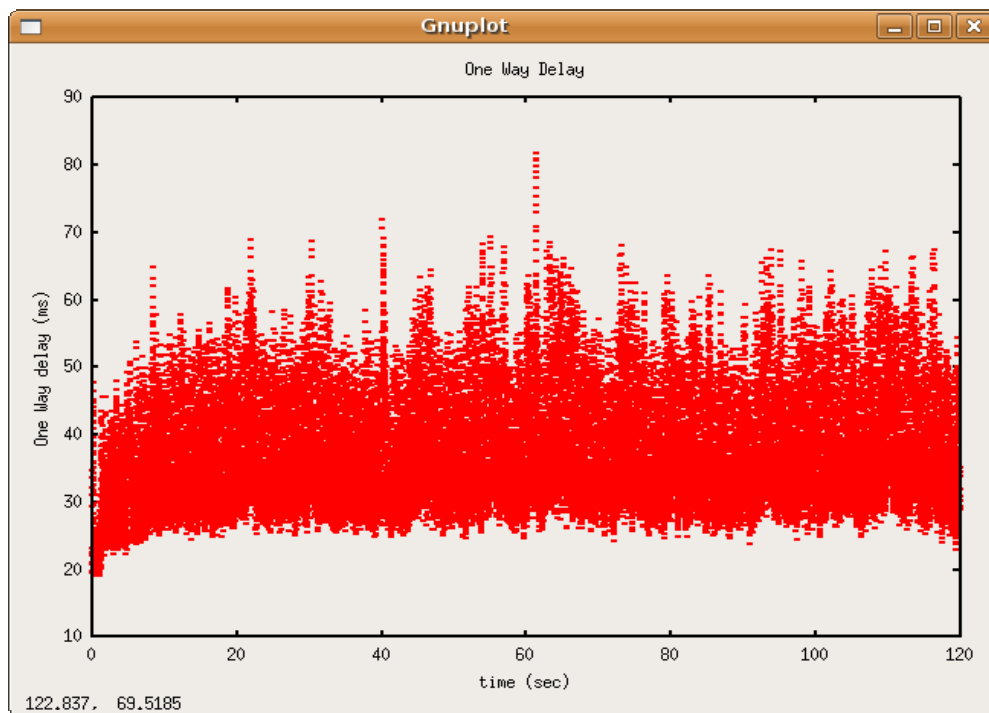
Σχήμα 73 : Γραφική παράσταση του Throughput.



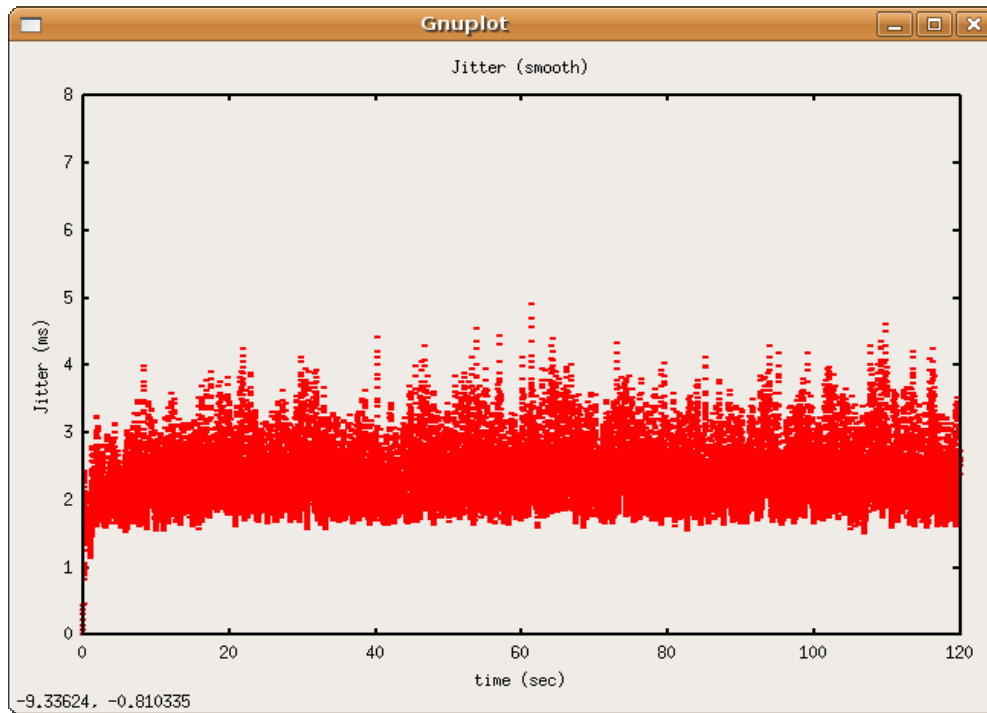
Σχήμα 74 : Γραφική παράσταση του RTT.

4.4.3.2 UDP κίνηση

Η προσομοίωση μετάδοσης video έλαβε 3.08 Mbps throughput, χωρίς απώλειες, με one way delay στα 36.86 ms και με την τιμή του jitter στα 2.34 ms. Ακολουθούν οι γραφικές παραστάσεις της one way delay και του jitter.



Σχήμα 75 : Γραφική παράσταση της One Way Delay.



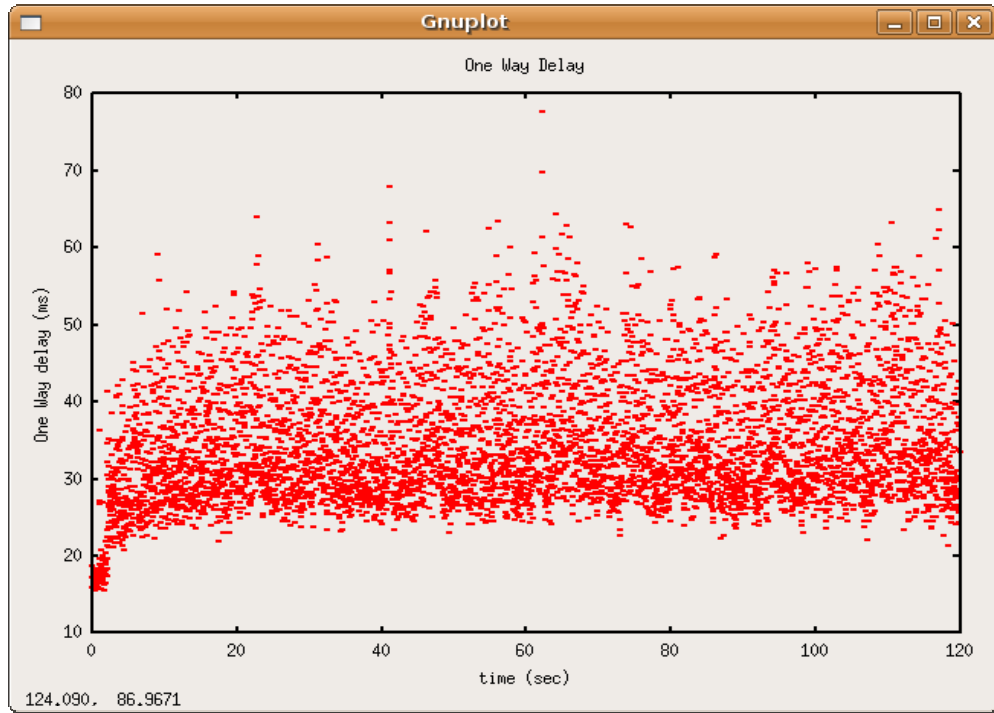
Σχήμα 76 : Γραφική παράσταση του Jitter.

4.4.3.3 VoIP επικοινωνία

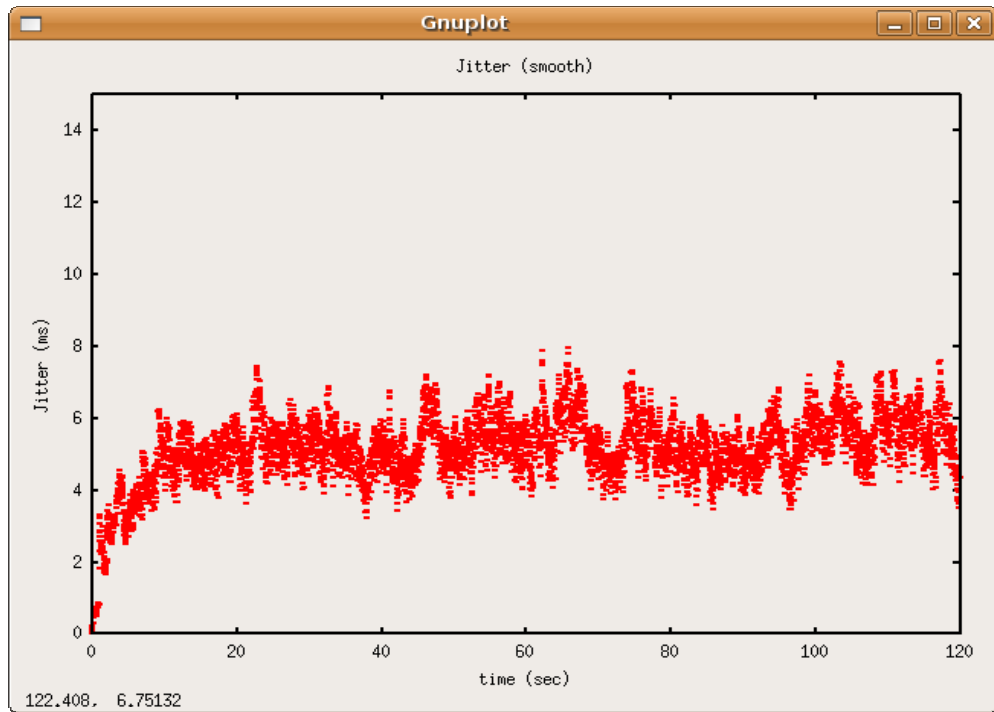
Αναλύοντας τις δύο κινήσεις, από τις οποίες αποτελείται η VoIP επικοινωνία, ξεχωριστά προκύπτει :

- Από 172.16.0.5 (Provider) -> 192.168.0.5 (χρήστης) :

Μετρήθηκε ότι το throughput είναι 24 Kbps, χωρίς απώλειες, η τιμή της one way delay είναι 34.4 ms και του jitter 5.15 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.

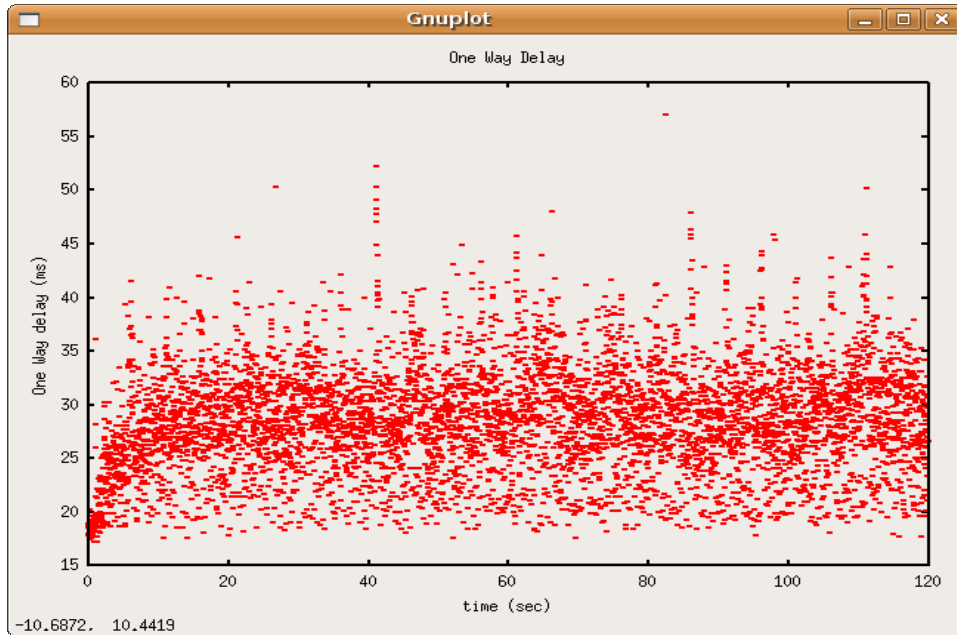


Σχήμα 77 : Γραφική παράσταση της One Way Delay.

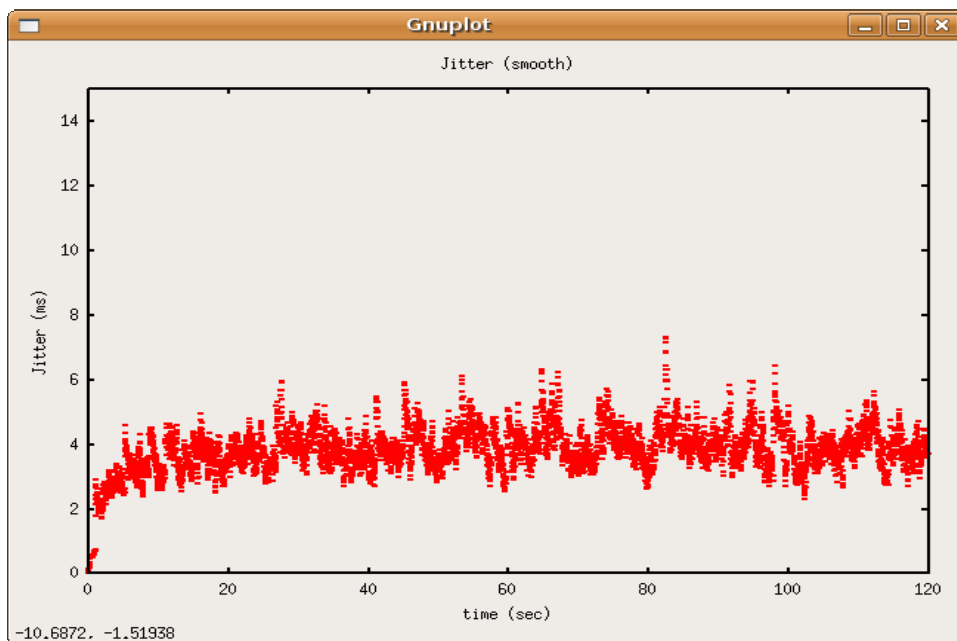


Σχήμα 78 : Γραφική παράσταση του Jitter.

- Από 192.168.0.5 (χρήστης) -> 172.16.0.5 (Provider) :
Μετρήθηκε ότι το throughput είναι 24 Kbps, δεν παρατηρήθηκαν απώλειες, η τιμή της one way delay είναι 28.49 ms και του jitter 3.84 ms. Ακολουθούν οι αντίστοιχες γραφικές παραστάσεις.



Σχήμα 79 : Γραφική παράσταση της One Way Delay.



Σχήμα 80 : Γραφική παράσταση του Jitter.

5. Συμπεράσματα - Επίλογος

Η πτυχιακή εργασία εστίασε στη μελέτη της Ποιότητα Υπηρεσιών σε ένα διαδραστικό σύστημα ψηφιακής τηλεόρασης ως ευρυζωνικό δίκτυο πρόσβασης για υπηρεσίες δεδομένων, ενσωματώνοντας σε αυτό ένα μηχανισμό διασφάλισης της ποιότητας των παρεχόμενων υπηρεσιών. Η βελτιστοποίηση αυτή επιτρέπει την παροχή διαδραστικών υπηρεσιών και την αμφίδρομη μετάδοση ψηφιακού σήματος σε ένα ικανοποιητικό επίπεδο ποιότητας, κάνοντας αποδοτική χρήση των δικτυακών πόρων και προσφέροντας εγγυήσεις για αξιόπιστη, βέλτιστη και γρήγορη επικοινωνία.

5.1 Αποτελέσματα

Από την ανάλυση των αποτελεσμάτων της πειραματικής διαδικασίας παρατηρήθηκε ότι η ενεργοποίηση του μηχανισμού διασφάλισης της ποιότητας των υπηρεσιών βελτιστοποίησε την απόδοση του δικτύου, δεσμεύοντας το απαιτούμενο εύρος ζώνης για κάθε κίνηση και εξασφαλίζοντας την αξιόπιστη μετάδοση των δεδομένων.

Πιο συγκεκριμένα, όσον αφορά την TCP κίνηση, δεσμεύτηκε το απαιτούμενο εύρος ζώνης, παρείχε μια αξιόπιστη μετάδοση χωρίς επαναμεταδόσεις πακέτων και βελτιώθηκε η RTT καθυστέρηση κατά 65.16%. Στην μετάδοση video παρατηρήθηκε εξάλειψη των απωλειών (από 48.8% σε 0%) και μείωση της one way καθυστέρησης κατά 83.5% (από 192.14 ms σε 31.96 ms), διατηρώντας παράλληλα την τιμή του jitter σε πολύ χαμηλά επίπεδα (1.42 ms). Οι τιμές αυτές καθιστούν το αναβαθμισμένο, με τον μηχανισμό DiffServ, σύστημα κατάλληλο για την παροχή υπηρεσιών IPTV streaming video. Κατά την εγκαθίδρυση της αμφίδρομης VoIP επικοινωνίας παρατηρήθηκε ότι στην κίνηση από τον Service Provider προς τον τελικό χρήστη οι απώλειες μειώθηκαν από 16.06% στο 0%, η one way καθυστέρηση μειώθηκε κατά 88.8% (από 176.9 ms σε 17.97 ms) και η τιμή του jitter κυμάνθηκε στα 1.13 ms. Στην κίνηση από τον τελικό χρήστη προς τον Service Provide οι απώλειες μειώθηκαν από 14.65% στο 0%, η one way καθυστέρηση μειώθηκε κατά 88.7% (από 178.55 ms σε 20.07 ms) και η τιμή του jitter κυμάνθηκε στα 1.17 ms.

Στην συνέχεια, στην πειραματική μέτρηση, όπου αποστέλλονται ταυτόχρονα όλες οι κινήσεις, παρατηρήθηκε εξάλειψη των επαναμεταδιδόμενων πακέτων στην TCP κίνηση (0% απώλειες), παρέχοντας παράλληλα και το απαιτούμενο εύρος ζώνης. Στην μετάδοση video παρατηρήθηκε μείωση των απωλειών από 9.24% στο 0%, ενώ η one way delay μειώθηκε κατά 79.2% (από 177.54 ms σε 36.86 ms) και η τιμή του jitter έμεινε σε ικανοποιητικό επίπεδο (2.34 ms). Κατά την εγκαθίδρυση της αμφίδρομης VoIP επικοινωνίας παρατηρήθηκε ότι στην κίνηση από τον Service Provider προς τον τελικό χρήστη οι απώλειες μειώθηκαν από 16.06% στο 0%, η one way καθυστέρηση μειώθηκε κατά 80.13% (από 173.13 ms σε 34.4 ms) και η τιμή του jitter κυμάνθηκε στα 5.15 ms. Στην κίνηση από τον τελικό χρήστη προς τον Service Provider οι απώλειες μειώθηκαν από 13.88% στο 0%, η one way καθυστέρηση μειώθηκε κατά 83.7% (από 174.9 ms σε 28.49 ms) και η τιμή του jitter κυμάνθηκε στα 3.84 ms.

5.2 Μελλοντικές Προτάσεις

Το περιβάλλον παροχής διαδραστικής ψηφιακής τηλεόρασης με ενσωματωμένο μηχανισμό διασφάλισης Ποιότητας Υπηρεσιών, που μελετήθηκε κατά την διάρκεια της συγκεκριμένης πτυχιακής εργασίας, υλοποιήθηκε με τέτοιο τρόπο, ώστε να εξυπηρετεί ένα συγκεκριμένο σύνολο SLA's, καθιστώντας την διαχείριση στατική. Επομένως, οποιαδήποτε αλλαγή χρειαστεί να γίνει λόγω τροποποιήσεων στα SLA's πρέπει να γίνει με χειροκίνητη διαδικασία, η οποία είναι δύσκολη και μπορεί να οδηγήσει σε λάθη. Συνεπώς, προτείνεται η βελτιστοποίηση του συστήματος με την εφαρμογή ενός μηχανισμού που να παρέχει μια δυναμική και σε πραγματικό χρόνο διαχείριση.

Επίσης η υλοποίηση του καναλιού επιστροφής, εκτός από την τεχνολογία IEEE 802.11g που χρησιμοποιήθηκε στο προτεινόμενο δίκτυο, μπορεί να επεκταθεί και να περιλάβει και άλλες ασύρματες και ενσύρματες τεχνολογίες ανάλογα με το είδος του τελικού χρήστη (κινητός, σταθερός). Έτσι θα υπάρχει μια πιο ολοκληρωμένη εικόνα για τα οφέλη και τις αδυναμίες της κάθε υλοποίησης.

Μελλοντικά, θα μπορούσε να επεκταθεί η συγκεκριμένη εργασία μελετώντας την εφαρμογή της τεχνολογίας DiffServ σε συνδυασμό με άλλους μηχανισμούς διασφάλισης της ποιότητας υπηρεσιών. Ο εμπλουτισμός του συστήματος που μελετήθηκε με τις

τεχνολογίες IntServ και MPLS, θα έδινε την δυνατότητα να επιτευχθεί μια πιο ολοκληρωμένη και δυναμική προσέγγιση στο ζήτημα της παροχής Ποιότητας Υπηρεσιών.

Τέλος, λαμβάνοντας υπόψη την επερχόμενη μετάβαση από το πρωτόκολλο IPv4 στο IPv6 για την αναβάθμιση των IP δικτύων, η μελέτη ενός συστήματος παροχής διαδραστικής ψηφιακής τηλεόρασης, το οποίο κάνει χρήση του πρωτοκόλλου IPv6 και η εξερεύνηση των δυνατοτήτων που αυτό παρέχει μπορεί να αποτελέσει μια άλλη πρόκληση.

6. Βιβλιογραφία - Αναφορές

1. ETS 300 744: Digital Video Broadcasting (DVB): Framing structure, channel coding and modulation for Digital Terrestrial Television (DVB-T), ETSI, 1997.
2. PSTN/ISDN (DVB-RCP: ETS 300 801), DECT (Digital Enhanced Cordless Telephony) (DVB-RCD: EN 301 193), LMDS (Local Multipoint Distribution System) (DVB-RCL: EN 301 199), GSM (Global System for Mobile) (DVB-RCG: EN 301 195), Terrestrial (DVB-RCT: Draft EN 301958).
3. Blake S., Black D., Carlson M., Davies E., Wang Z. and Weiss W., “An Architecture for Differentiated Services”, RFC 2475, December 1998.
4. Brim S., Carpenter B. and Le Faucheur F., “Per Hop Behavior Identification Codes”, RFC 2836, May 2000.
5. Information Sciences Institute University of Southern California, “INTERNET PROTOCOL”, RFC 791, September 1981.
6. “Digital Switchover: Developing Infrastructures for Broadband Access”, 6th Information Society Technologies, 6th Framework Programme (ATHENA FP6-507312), (<http://www.ist-athena.org>).
7. Huston G., “Next Steps for the IP QoS Architecture”, RFC 2990, November 2000.
8. Braden R., Clark D. and Shenker S., “Integrated Services in the Internet Architecture: an Overview”, RFC 1633, June 1994.
9. Rosen E., Viswanathan A. and Callon R., “Multiprotocol Label Switching Architecture”, RFC 3031, January 2001.
10. Nichols K., Blake S., Baker F. and Black D., “Definition of the differentiated services field (dsfield) in the ipv4 and ipv6 headers”, RFC 2474, December 1998.
11. Jacobson V., Nichols K. and Poduri K., “An Expedited Forwarding PHB”, RFC 2598, June 1999.
12. Heinanen J., Baker, W. Weiss F. and Wroclawski J., “Assured Forwarding PHB Group”, RFC 2597, June 1999.
13. A.S. Tanenbaum “Computer Networks” Fourth Edition, ISBN: 0-13-066102-3.

14. IPERF – Traffic Generator Tool
15. D-ITG – Traffic Generator Tool
16. MGEN - The Multi-Generator Toolset, (<http://mgen.pf.itd.nrl.navy.mil/>).
17. iproute2+tc notes, (<http://snafu.freedom.org/linux2.2/iproute-notes.html>).
18. Linux Advanced Routing & Traffic Control, (<http://lartc.org/>).
19. Differentiated Service on Linux HOWTO, (<http://opalsoft.net/qos/DS.htm>).
20. IP Performance Metrics (IPPM) Working group,
(<http://www.ietf.org/html.charters/ippm-charter.html>).
21. TCPdump - dump traffic on a network, (<http://www.tcpdump.org/>).
22. Gnuplot homepage, (<http://www.gnuplot.info/>).
23. Calyam P. and Lee C., “Characterizing voice and video traffic behaviour over the Internet”, in Proc. ISCIS’ 05, Istanbul, Turkey, 26-28 Oct. 2005, pp. 91-95.
(http://www.osc.edu/research/networking/PDFs/vvoip_iscis05.pdf).
24. Iptables, (<http://www.netfilter.org/projects/iptables/index.html>).
25. Tcptrace, (<http://jarok.cs.ohiou.edu/software/tcptrace/>).
26. Xplot homepage, (<http://www.xplot.org/>).

7. Παραρτήματα

7.1 Γλωσσάριο

ADC	Analog to Digital Converter
ADSL	Asymmetric Digital Subscriber Line
AF	Assured Forwarding
ATM	Asynchronous Transfer Mode
BA	Behavior Aggregate
BE	Best Effort
CMN	Cell Main Node
COFDM	Coded Orthogonal Frequency Division Multiplexing
DAC	Digital to Analog Converter
DCR	DiffServ Core Router
DER	DiffServ Edge Router
DiffServ	Differentiated Services
DS	Differentiated Services
DSCP	DiffServ Code Point
DSLAM	DSL Access Multiplexer
DVB	Digital Video Broadcasting
DVB-H	Handheld Digital Video Broadcasting
DVB-RCT	DVB Terrestrial Return Channel
DVB-S	Satellite Digital Video Broadcasting
DVB-T	Terrestrial Digital Video Broadcasting
EF	Expedited Forwarding
ETSI	European Telecommunications Standards Institute
FIFO	First In First Out
GPRS	General Packet Radio Service
GRED	Generic RED
HTB	Hierarchical Token Bucket
IDTV	Interactive Digital Television
IEEE	Institute of Electrical & Electronics Engineers
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
ISDN	Integrated Services Digital Networking
ISM	Industrial, Scientific, Medical
ISP	Internet Service Provider
ITU	International Telecommunication Union
MPE	Multi-Protocol Encapsulation
MPEG	Moving Picture Expert Group
MPLS	Multi-Protocol Label Switching
OFDM	Orthogonal Frequency Division Multiplexing
PC	Personal Computer

PHB	Per Hop Behavior
PRIO	Priority Queuing
PSTN	Public Switched Telephone Network
QAM	Quadrature Phase Shift Keying
QoS	Quality of Service
RED	Random Early Detection
RSVP	Resource Reservation Protocol
RTO	Retransmission TimeOut
RTP	Real Time Protocol
RTT	Round Trip Time
SFQ	Stochastic Fair Queuing
SLA	Service Level Agreement
SP	Service Provider
TBF	Token Bucket Filter
TCA	Traffic Conditioning Agreement
TCP	Transmission Control Protocol
ToS	Type of Service
UDP	User Datagram Protocol
UHF	Ultra High Frequency
UMTS	Universal Mobile Telecommunication System
VoIP	Voice over IP
WEP	Wired Equivalent Privacy
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network

7.2 Εντολές Εφαρμογής DiffServ

7.2.1 Σενάριο 1°

7.2.1.1 DiffServ Edge Router

```
#!/bin/bash

#delete any qdisc
tc qdisc del dev eth0 root
tc qdisc del dev wlan0 root

#clear and show iptables mangle chain
iptables -t mangle -F
iptables -t mangle -L -n

#show queueing disciplines
tc -s -d qdisc

# mark DSCP of TCP and the rest traffic
iptables -t mangle -A PREROUTING -s 0/0 -i eth0 -j DSCP --set-dscp 0x00
iptables -t mangle -A PREROUTING -p TCP -i eth0 -j DSCP --set-dscp 0x2e

# condition egress traffic
# Main htb qdisc & class
tc qdisc add dev wlan0 root handle 1:0 htb

# - EF Class (1:10)
tc class add dev wlan0 parent 1:0 classid 1:1 htb rate 8000kbit ceil 8000kbit burst 10k
tc class add dev wlan0 parent 1:1 classid 1:10 htb rate 4700kbit ceil 5000kbit burst 10k \
```

```

prio 0
tc qdisc add dev wlan0 parent 1:10 handle 10: sfq perturb 10
# - BE Class (1:20)
tc class add dev wlan0 parent 1:1 classid 1:20 htb rate 2000kbit ceil 8000kbit burst 10k \
prio 0
tc qdisc add dev wlan0 parent 1:20 handle 20: sfq perturb 10

tc filter add dev wlan0 parent 1:0 protocol ip prio 1 u32 match ip tos 0xb8 0xff flowid \
1:10
tc filter add dev wlan0 parent 1:0 protocol ip prio 1 u32 match ip tos 0x0 0xff flowid 1:20

tc -s -d qdisc
tc -s -d class

```

7.2.1.2 DiffServ Core Router

```

#!/bin/bash

#delete any policy/shaping mechanisms
tc qdisc del dev eth0 root

# - Main dsmark & classifier
tc qdisc add dev eth0 handle 1:0 root dsmark indices 64 set_tc_index
tc filter add dev eth0 parent 1:0 protocol ip prio 1 tcindex mask 0xfc shift 2

# - Main htb qdisc & class
tc qdisc add dev eth0 parent 1:0 handle 2:0 htb
tc class add dev eth0 parent 2:0 classid 2:1 htb rate 8Mbit burst 12k ceil 8Mbit

```

```
# - EF Class (2:10)
tc class add dev eth0 parent 2:1 classid 2:10 htb rate 4500kbit burst 12k ceil 5000kbit \
prio 0
tc qdisc add dev eth0 parent 2:10 pfifo limit 200
tc filter add dev eth0 parent 2:0 protocol ip prio 1 handle 0x2e tcindex classid 2:10 \
pass_on

# - BE Class (2:20)
tc class add dev eth0 parent 2:1 classid 2:20 htb rate 2Mbit burst 6k ceil 8Mbit prio 1
tc qdisc add dev eth0 parent 2:20 red limit 250000 min 42000 max 125000 burst 70 \
avpkt 1000 bandwidth 2Mbit probability 1
tc filter add dev eth0 parent 2:0 protocol ip prio 2 handle 0 tcindex mask 0 classid 2:20 \
pass_on

tc -s -d qdisc
tc -s -d class show dev eth0
```

7.2.2 Σενάριο 2°

7.2.2.1 DiffServ Edge Router

```
#!/bin/bash

#delete any qdisc
tc qdisc del dev eth0 root
tc qdisc del dev wlan0 root

#clear iptables mangle chain
iptables -t mangle -F
```

```

iptables -t mangle -L -n

#show queueing disciplines
tc -s -d qdisc

# mark DSCP of UDP and the rest traffic
iptables -t mangle -A PREROUTING -s 0/0 -i eth0 -j DSCP --set-dscp 0x00
iptables -t mangle -A PREROUTING -p UDP --dport 7001 -i eth0 -j DSCP --set-dscp \
0x2e

# - Main htb qdisc & class
tc qdisc add dev wlan0 root handle 1:0 htb
tc class add dev wlan0 parent 1:0 classid 1:1 htb rate 8000kbit ceil 8000kbit burst 10k

# - EF Class (1:10)
tc class add dev wlan0 parent 1:1 classid 1:10 htb rate 3300kbit ceil 8000kbit burst 10k \
prio 0
tc qdisc add dev wlan0 parent 1:10 handle 10: sfq perturb 10

# - BE Class (1:20)
tc class add dev wlan0 parent 1:1 classid 1:20 htb rate 4500kbit ceil 8000kbit burst 10k \
prio 0
tc qdisc add dev wlan0 parent 1:20 handle 20: sfq perturb 10

tc filter add dev wlan0 parent 1:0 protocol ip prio 1 u32 match ip tos 0xb8 0xff flowid \
1:10
tc filter add dev wlan0 parent 1:0 protocol ip prio 1 u32 match ip tos 0x0 0xff flowid 1:20

tc -s -d qdisc
tc -s -d class

```

7.2.2.2 DiffServ Core Router

```
#!/bin/bash

#delete any policy/shaping mechanisms
tc qdisc del dev eth0 root

# - Main dsmark & classifier
tc qdisc add dev eth0 handle 1:0 root dsmark indices 64 set_tc_index
tc filter add dev eth0 parent 1:0 protocol ip prio 1 tcindex mask 0xfc shift 2

# - Main htb qdisc & class
tc qdisc add dev eth0 parent 1:0 handle 2:0 htb
tc class add dev eth0 parent 2:0 classid 2:1 htb rate 8Mbit burst 12k ceil 8Mbit

# - EF Class (2:10)
tc class add dev eth0 parent 2:1 classid 2:10 htb rate 3300Kbit burst 12k ceil 8Mbit prio 0
tc qdisc add dev eth0 parent 2:10 pfifo limit 200
tc filter add dev eth0 parent 2:0 protocol ip prio 1 handle 0x2e tcindex classid 2:10
pass_on

# - BE Class (2:20)
tc class add dev eth0 parent 2:1 classid 2:20 htb rate 4500Kbit burst 6k ceil 8Mbit prio 1
tc qdisc add dev eth0 parent 2:20 red limit 562500 min 94000 max 281250 burst 157
avpkt 1000 bandwidth 4500Kbit probability 1
tc filter add dev eth0 parent 2:0 protocol ip prio 2 handle 0 tcindex mask 0 classid 2:20
pass_on

tc -s -d qdisc
tc -s -d class show dev eth0
```

7.2.3 Σενάριο 3°

7.2.3.1 DiffServ Edge Router

```
#!/bin/bash

#delete any qdisc
tc qdisc del dev eth0 root
tc qdisc del dev wlan0 root

#clear iptables mangle chain
iptables -t mangle -F
iptables -t mangle -L -n

#show queueing disciplines
tc -s -d qdisc

# mark DSCP of UDP and the rest traffic
iptables -t mangle -A PREROUTING -s 0/0 -i eth0 -j DSCP --set-dscp 0x00
iptables -t mangle -A PREROUTING -p UDP --sport 7000 -i eth0 -j DSCP --set-dscp
0x2e

# - Main htb qdisc & class
tc qdisc add dev wlan0 root handle 1:0 htb
tc class add dev wlan0 parent 1:0 classid 1:1 htb rate 8000kbit ceil 8000kbit burst 10k

# - EF Class (1:10)
tc class add dev wlan0 parent 1:1 classid 1:10 htb rate 60kbit ceil 8000kbit burst 10k \
prio 0
tc qdisc add dev wlan0 parent 1:10 handle 10: sfq perturb 10
```

```
# - BE Class (1:20)
tc class add dev wlan0 parent 1:1 classid 1:20 htb rate 7000kbit ceil 8000kbit burst 10k
prio 0
tc qdisc add dev wlan0 parent 1:20 handle 20: sfq perturb 10

tc filter add dev wlan0 parent 1:0 protocol ip prio 1 u32 match ip tos 0xb8 0xff flowid
1:10
tc filter add dev wlan0 parent 1:0 protocol ip prio 1 u32 match ip tos 0x0 0xff flowid 1:20

tc -s -d qdisc
tc -s -d class
```

7.2.3.2 DiffServ Core Router

```
#!/bin/bash

#delete any policy/shaping mechanisms
tc qdisc del dev eth0 root

# - Main dsmark & classifier
tc qdisc add dev eth0 handle 1:0 root dsmark indices 64 set_tc_index
tc filter add dev eth0 parent 1:0 protocol ip prio 1 tcindex mask 0xfc shift 2

# - Main htb qdisc & class
tc qdisc add dev eth0 parent 1:0 handle 2:0 htb
tc class add dev eth0 parent 2:0 classid 2:1 htb rate 8Mbit burst 12k ceil 8Mbit
```



```
# - EF Class (2:10)
tc class add dev eth0 parent 2:1 classid 2:10 htb rate 60Kbit burst 12k ceil 8Mbit prio 0
tc qdisc add dev eth0 parent 2:10 pfifo limit 200
tc filter add dev eth0 parent 2:0 protocol ip prio 1 handle 0x2e tcindex classid 2:10
pass_on

# - BE Class (2:20)
tc class add dev eth0 parent 2:1 classid 2:20 htb rate 7000Kbit burst 6k ceil 8Mbit prio 1
tc qdisc add dev eth0 parent 2:20 red limit 875000 min 146000 max 437500 burst 244
avpkt 1000 bandwidth 7000Kbit probability 1
tc filter add dev eth0 parent 2:0 protocol ip prio 2 handle 0 tcindex mask 0 classid 2:20
pass_on

tc -s -d qdisc
tc -s -d class show dev eth0
```

7.2.4 Σενάριο 4^ο

7.2.4.1 DiffServ Edge Router

```
#!/bin/bash

#delete any qdisc
tc qdisc del dev eth0 root
tc qdisc del dev wlan0 root

#clear iptables mangle chain
iptables -t mangle -F
iptables -t mangle -L -n
```

```

#show queueing disciplines
tc -s -d qdisc
# mark dscp of each traffic
iptables -t mangle -A PREROUTING -s 0/0 -i eth0 -j DSCP --set-dscp 0x22
iptables -t mangle -A PREROUTING -p TCP -i eth0 -j DSCP --set-dscp 0x1A
iptables -t mangle -A PREROUTING -p UDP --sport 7000 -i eth0 -j DSCP --set-dscp
0xA
iptables -t mangle -A PREROUTING -p UDP --dport 7001 -i eth0 -j DSCP --set-dscp
0x12

# main htb qdisc & class
tc qdisc add dev wlan0 root handle 1:0 htb
tc class add dev wlan0 parent 1:0 classid 1:1 htb rate 8000kbit ceil 8000kbit burst 10k

# - AF1 Class (1:10)
tc class add dev wlan0 parent 1:1 classid 1:10 htb rate 60kbit ceil 8000kbit burst 10k prio
0
tc qdisc add dev wlan0 parent 1:10 handle 10: sfq perturb 10

# - AF2 Class (1:20)
tc class add dev wlan0 parent 1:1 classid 1:20 htb rate 3200kbit ceil 8000kbit burst 10k
prio 0
tc qdisc add dev wlan0 parent 1:20 handle 20: sfq perturb 10

# - AF3 Class (1:30)
tc class add dev wlan0 parent 1:1 classid 1:30 htb rate 3000kbit ceil 3300kbit burst 10k
prio 0
tc qdisc add dev wlan0 parent 1:30 handle 30: sfq perturb 10

# - AF4 Class (1:40)

```

```
tc class add dev wlan0 parent 1:1 classid 1:40 htb rate 1000kbit ceil 8000kbit burst 10k
prio 0
tc qdisc add dev wlan0 parent 1:40 handle 40: sfq perturb 10

tc filter add dev wlan0 parent 1:0 protocol ip prio 1 u32 match ip tos 0x28 0xff flowid
1:10
tc filter add dev wlan0 parent 1:0 protocol ip prio 1 u32 match ip tos 0x48 0xff flowid
1:20
tc filter add dev wlan0 parent 1:0 protocol ip prio 1 u32 match ip tos 0x68 0xff flowid
1:30
tc filter add dev wlan0 parent 1:0 protocol ip prio 1 u32 match ip tos 0x88 0xff flowid
1:40

tc -s -d qdisc
tc -s -d class
```

7.2.4.2 DiffServ Core Router

```
#!/bin/bash

#delete any policy/shaping mechanisms
tc qdisc del dev eth0 root

# - Main dsmark & classifier
tc qdisc add dev eth0 handle 1:0 root dsmark indices 64 set_tc_index
tc filter add dev eth0 parent 1:0 protocol ip prio 1 tcindex mask 0xfc shift 2

# - Main htb qdisc & class
tc qdisc add dev eth0 parent 1:0 handle 2:0 htb
```

```

tc class add dev eth0 parent 2:0 classid 2:1 htb rate 8Mbit burst 12k ceil 8Mbit

# - AF1 Class (2:10)
tc class add dev eth0 parent 2:1 classid 2:10 htb rate 60Kbit burst 12k ceil 8Mbit prio 0
tc qdisc add dev eth0 parent 2:10 pfifo limit 200
tc filter add dev eth0 parent 2:0 protocol ip prio 1 handle 0xA tcindex classid 2:10
pass_on

# - AF2 Class (2:20)
tc class add dev eth0 parent 2:1 classid 2:20 htb rate 3200Kbit burst 12k ceil 8Mbit prio 0
tc qdisc add dev eth0 parent 2:20 pfifo limit 200
tc filter add dev eth0 parent 2:0 protocol ip prio 1 handle 0x12 tcindex classid 2:20
pass_on

# - AF3 Class (2:30)
tc class add dev eth0 parent 2:1 classid 2:30 htb rate 3000Kbit burst 10k ceil 3300Kbit
prio 0
tc qdisc add dev eth0 parent 2:30 pfifo limit 300
tc filter add dev eth0 parent 2:0 protocol ip prio 1 handle 0x1A tcindex classid 2:30
pass_on

# - AF4 - BE Class (2:40)
tc class add dev eth0 parent 2:1 classid 2:40 htb rate 1000Kbit burst 6k ceil 8Mbit prio 1
tc qdisc add dev eth0 parent 2:40 red limit 125000 min 20835 max 62500 burst 35 avpkt
1000 bandwidth 1000Kbit probability 1
tc filter add dev eth0 parent 2:0 protocol ip prio 2 handle 0x22 tcindex classid 2:40
pass_on
tc -s -d qdisc
tc -s -d class show dev eth0

```

7.3 Εντολές Δημιουργίας, “Σύλληψης” και Ανάλυσης Κίνησης

7.3.1 TCP κίνηση

7.3.1.1 Δημιουργία

Για την δημιουργία της TCP κίνησης χρησιμοποιήθηκε το εργαλείο iperf. Από την πλευρά του τελικού χρήστη εκτελείται το iperf σαν server, όπου “ακούει” στην port 5001 για με την εντολή :

```
iperf -s
```

Από την πλευρά του Service Provider εκτελείται το iperf σαν client, δηλώνοντας σαν παραμέτρους την IP διεύθυνση του server και τον χρόνο που θα διαρκέσει η επικοινωνία, με την εντολή :

```
iperf -c <ip_address_of_server> -t <time_in_seconds>  
π.χ : iperf -c 192.168.0.5 -t 120
```

7.3.1.2 Σύλληψη

Με το εργαλείο tcpdump θα “συλληφθεί” η κίνηση και θα αποθηκευτεί σε ένα dump αρχείο, ώστε να μπορέσει έπειτα να γίνει η ανάλυση της. Η εντολή είναι η εξής :

```
tcpdump -w <filename> -n -vv -i <network_interface>  
π.χ : tcpdump -w tcp_traffic -n -vv -i eth0 tcp port 5001
```

7.3.1.3 Ανάλυση

Το dump αρχείο που δημιουργήθηκε από το tcpdump θα χρησιμοποιηθεί από το εργαλείο tcptrace, ώστε να αναλυθεί η κίνηση, δίνοντας πληροφορίες για το throughput, την RTT καθυστέρηση και τις επαναμεταδόσεις. Οι εντολές δίνονται παρακάτω.

```
Throughput : tcptrace -zxy -xtraffic'-B -i1' <dump_file>
```

```
RTT : tcptrace -zxy -xtraffic'-R -i1' <dump_file>
```

```
Retransmissions : tcptrace -zxy -xtraffic'-L -i1' <dump_file>
```

7.3.1.4 Γραφική Απεικόνιση

Κάθε εντολή tcptrace δημιουργεί ένα σύνολο αρχείων με κατάληξη .xpl. Χρησιμοποιώντας το εργαλείο xplot ή το gnuplot (αφού πρώτα μετατραπούν τα .xpl σε .gpl με το xpl2gpl εργαλείο) και επιλέγοντας το κατάλληλο .xpl αρχείο εμφανίζονται οι αντίστοιχες γραφικές παραστάσεις. Η εντολή που χρησιμοποιείται είναι :

```
xplot <filename>.xpl
```

7.3.2 UDP κίνηση

7.3.2.1 Συγχρονισμός ρολογιών

Στην UDP μετάδοση, για να είναι δυνατός ο υπολογισμός την one way καθυστέρησης είναι απαραίτητος ο συγχρονισμός των ρολογιών των υπολογιστών που μετέχουν στην επικοινωνία. Ο συγχρονισμός επιτυγχάνεται χρησιμοποιώντας κάποιο πρόγραμμα που κάνει χρήση του NTP (Network Time Protocol) πρωτοκόλλου.

7.3.2.2 Δημιουργία

Για την δημιουργία της TCP κίνησης χρησιμοποιήθηκε το εργαλείο mgen. Από την πλευρά του τελικού χρήστη εκτελείται το mgen για να “ακούει” σε κάποια port που δηλώνεται για με την εντολή :

```
mgen input <script_file>
```

Το script_file περιέχει τις εντολές που θα εκτελεστούν.

```
0.0 LISTEN UDP 7000
```

Από την πλευρά του Service Provider εκτελείται το mgen για να μεταδώσει κίνηση προς τον χρήστη που “ακούει”. Οι εντολές που χρειάζεται το πρόγραμμα αυτό για τη δημιουργία της κίνησης, όπως είναι ο χρόνος εκτέλεσης και το bit rate περιέχονται στο script_file :

```
0.0 ON 1 UDP SRC 7001 DST 192.168.0.5/7000 PERIODIC [1 4096]
120.0 OFF 1
```

7.3.2.3 Σύλληψη

Με το εργαλείο tcpdump θα “συλληφθεί” η κίνηση και θα αποθηκευτεί σε ένα dump αρχείο, ώστε να μπορέσει έπειτα να γίνει η ανάλυση της. Στην UDP κίνηση θα χρειαστεί ένα tcpdump για να “πιάνει” την κίνηση που μεταδίδει ο Service Provider και ένα για να “πιάνει” την κίνηση που λαμβάνει ο τελικός χρήστης. Η εντολή είναι η εξής :

```
tcpdump -w <filename> -n -vv -i <network_interface>
π.χ : tcpdump -w udp_traffic -n -vv -i eth0 udp port 7000
```

7.3.2.4 Ανάλυση

Τα δύο dump αρχεία που δημιουργήθηκαν από το tcpdump θα μετατραπούν σε .txt αρχεία με τις εντολές :

```
tcpdump -tt -vv -n -r <dumpfile_of_server> > udp_sender.txt  
tcpdump -tt -vv -n -r <dumpfile_of_client> > udp_receiver.txt
```

Αφού ολοκληρωθεί η δημιουργία των txt αρχείων ακολουθεί, η εκτέλεση των perl προγραμμάτων του κεφαλαίου 7.4. Αρχικά εκτελείται το αρχείο παρακάτω που δημιουργεί δύο νέα αρχεία, εκ των οποίων το ένα έχει την κίνηση του αποστολέα (sender) και το άλλο του δέκτη (receiver) και εκτελείται ως εξής:

```
perl ipv4_dvstream_createendfiles.pl
```

Με το πρόγραμμα ipv4_losses.pl υπολογίζονται οι απώλειες κατά την μετάδοση της udp κίνησης.

```
perl ipv4_losses.pl
```

Με το επόμενο πρόγραμμα υπολογίζεται το data rate του αποστολέα και του δέκτη αντίστοιχα.

```
perl ipv4_sender_receiver_rate.pl
```

Ακολουθεί το πρόγραμμα όπου ευθυγραμμίζει κάθε πακέτο του αρχείου του δέκτη με κάθε πακέτο του αρχείου του αποστολέα, ώστε να υπολογιστεί αργότερα η διακύμανση της καθυστέρησης (jitter).

```
perl ipv4_align_for_delay_jitt.pl
```

Στην συνέχεια δημιουργούνται δύο αρχεία με τους χρόνους του αποστολέα και του δέκτη αντίστοιχα, για τον υπολογισμό του jitter.:

```
perl ipv4_timestamp.pl
```

Στη συνέχεια υπολογίζεται η διακύμανση της καθυστέρησης (jitter) του δικτύου. Μετά το τέλος της εκτέλεσης του προγράμματος εμφανίζονται η μέση , η μέγιστη και η ελάχιστη τιμή του jitter και δίνεται η καθυστέρηση μεταξύ των πακέτων (packet to packet Delay).

```
perl ipv4_inter_arrival_jitter.pl  
και  
perl ipv4_jitter.pl
```

Η one way καθυστέρηση, δηλαδή ο χρόνος που χρειάζεται ένα πακέτο για να μεταφερθεί από τον sender στον receiver υπολογίζεται με το πρόγραμμα :

```
perl ipv4_one_way_delay.pl
```


7.3.2.5 Γραφική Απεικόνιση

Για την απεικόνιση των γραφικών παραστάσεων των απωλειών, της one way καθυστέρησης και του jitter χρησιμοποιούνται τα παρακάτω προγράμματα με τις εντολές :

Losses :

```
gnuplot losses.gpl
```

One way delay :

```
gnuplot onewaydelay.gpl
```

Jitter :

```
gnuplot jitter.gpl
```

7.3.3 VoIP κίνηση

Για την ανάλυση της VoIP κίνησης ακολουθείται η διαδικασία που περιγράφεται και στην ανάλυση της UDP κίνησης (κεφ. 7.3.2). Για τη δημιουργία της κίνησης αντί του mgen χρησιμοποιείται το εργαλείο D-ITG.

Στον Server εκτελείται η εντολή :

```
ITGSend -a <ip_address> -sp <port> -t <time_in_msec> VoIP -x <protocol>
```

```
π.χ ITGSend -a 172.16.0.5 -sp 7000 -t 120000 VoIP -x G.729.2
```

Στον Client εκτελείται η εντολή :

```
ITGRecv
```

7.4 Προγράμματα Ανάλυσης UDP Κίνησης

7.4.1 Πρόγραμμα ipv4_dvbstream_createendfiles.pl

```
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
#Author Anargiros Sideris
#Mail irons@pasiphae.teiher.gr
#!/usr/bin/perl -w

open(INPUTFILE, "<udp_sender.txt") || die ("cannot open udp_sender file 1\n");

unless (open (OUTFILE, ">final.tx"))
{
    die ("cannot open output file outfile\n");
}

$start = time;

my (@rec_lines) = <INPUTFILE>;

foreach $rec_line (@rec_lines)
{
    chomp($rec_line);
    @line=split(/\t +/, $rec_line);

    for ($k=0;$k<@line;$k++)
    {
        $sid=0;
        if ($line[$k] eq "cid")
        {
            $sid=$line[$k+1];
            last;
        }
    }
}
```

```

for ($k=0;$k<@line;$k++)
{
    if ($line[$k] eq "seq")
    {
        $id=$id.($line[$k+1]);
        last;
    }
}
for ($k=0;$k<@line;$k++)
{
    if ($line[$k] eq "ser")
    {
        $id=$id.($line[$k+1]);
        last;
    }
}
for ($k=0;$k<@line;$k++)
{
    if ($line[$k] eq "id")
    {
        $id=$id.($line[$k+1]);
        last;
    }
}

for ($k=0;$k<@line;$k++)
{
    if ($line[$k] eq "length:")
    {
        $size=$line[$k+1];
        last;
    }
}
$time=$line[0];
#id=substr($id,0,length($id)-1);
$size=substr($size,0,length($size)-1);
#print ("id $size \n");
print OUTPUTFILE ("id\tid\ttime\tsize\n");
}

close(INPUTFILE);
close(OUTPUTFILE);

```

```

#second file

open(INPUTFILE, "<udp_receiver.txt") || die ("cannot open udp_receiver file 1\n");

unless (open (OUTFILE, ">final.rx"))
{
    die ("cannot open output file outfile\n");
}

my (@rec_lines) = <INPUTFILE>;

foreach $rec_line (@rec_lines)
{
    chomp($rec_line);
    @line=split(/[t +]/,$rec_line);
    for ($k=0;$k<@line;$k++)
    {
        $id=0;
        if ($line[$k] eq "cid")
        {
            $id=$line[$k+1];
            last;
        }
    }

    for ($k=0;$k<@line;$k++)
    {
        if ($line[$k] eq "seq")
        {
            $id=$id.($line[$k+1]);
            last;
        }
    }

    for ($k=0;$k<@line;$k++)
    {
        if ($line[$k] eq "ser")
        {
            $id=$id.($line[$k+1]);
            last;
        }
    }

    for ($k=0;$k<@line;$k++)
    {
        if ($line[$k] eq "id")

```

```

        {
            Sid=$id.($line[$k+1]);
            last;
        }
    }
for ($k=0;$k<@line;$k++)
{
    if ($line[$k] eq "length:")
    {
        $size=$line[$k+1];
        last;
    }
}
$time=$line[0];
# $id=substr($id,0,length($id)-1);
$size=substr($size,0,length($size)-1);

print OUTFILE ("Sid\tSid\tTime\tSize\n");
}

$elapsed_sec = time - $start;

my $second = $elapsed_sec%60;
my $minute = ($elapsed_sec/60)%60;
my $hour = ($elapsed_sec/(60*60))%24;
print "Total Time elapsed: $hour hours:$minute min:$second sec\n";

close(INPUTFILE);
close(OUTPUTFILE);

```

7.4.2 Πρόγραμμα `ipn4_losses.pl`

```
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
#Author Anargiros Sideris
#Mail irons@pasiphae.teiher.gr
#This file calculates the losses in a udp transmission.

#!/usr/bin/perl -w

$start = time;

& calc_loss;

$elapsed_sec = time - $start;

my $second = $elapsed_sec%60;
my $minute = ($elapsed_sec/60)%60;
my $hour = ($elapsed_sec/(60*60))%24;
print "Total Time elapsed: $hour hours:$minute min:$second sec\n";

sub calc_loss
# Simple loss calculation
{
    my ($sender_packets);
```

```
my ($receiver_packets);
my ($loss_rate);

$sender_packets = 0;
$receiver_packets = 0;
$loss_rate = 0;

# Sender file
open (SENDER, "<final.tx") || die ("cannot open input file 1\n");

while (<SENDER>)
{

    $sender_packets++;

}

close (SENDER);

# Receiver file
open (RECEIVER, "<final.rx") || die ("cannot open input file 2\n");

while (<RECEIVER>)
{

    $receiver_packets++;

}

close (RECEIVER);
```

```

#calculation
    $loss_rate =
        (($sender_packets -
            $receiver_packets) / $sender_packets) * 100;

    print "sender packets $sender_packets, receiver packets $receiver_packets, losses $loss_rate%\n";

    if (($sender_packets - $receiver_packets)!=0)
    {
        & lossvstime;
    }

}

sub lossvstime
{

unless (open (OUTFILE, ">pack_num_seq_loss_vs_time"))
{

    die ("cannot open output file outfile\n");

}

    open (SENDER, "<final.tx");
open (RECEIVER, "<final.rx");
my (@sen_lines) = <SENDER>;
my ($sen_line);
my (@rec_lines) = <RECEIVER>;
my ($rec_line);
my ($temp);
my ($temp1);
my ($lock);
my ($lock_ref_time);
my ($send_time);

```



```

my ($start_ref_time);
my ($packet_counter);
close (SENDER);
close (RECEIVER);
$size=@rec_lines;
$size1=@sen_lines;
$packet_counter=0;
$lock_ref_time=0;

foreach $sen_line (@sen_lines)
    {
$lock=0;
$packet_counter++;
    chomp($sen_line);
        @line_sender=split(/\t +/,$sen_line);
        $temp1=$line_sender[0].$line_sender[1];

        if ($lock_ref_time==0)
        {
$start_ref_time=$line_sender[2];
$lock_ref_time=1;
#print ("\n$start_ref_time\n");

        }
foreach $rec_line (@rec_lines)
    {
        chomp($rec_line);
        @line_receiver=split(/\t +/,$rec_line);
        $temp=$line_receiver[0].$line_receiver[1];
if ($temp eq $temp1)
        {

                #unless (open (OUTFILE, ">>aligned_sender"))
                #{

                #
                die ("cannot open output file outfile\n");

```

```

        #}

        #close (OUTFILE);
        $lock=1;
        last;
#print " sender $sen_line receiver $rec_line\n";
        }

    }
if ($lock==0)
{
$send_time=$line_sender[2]-$start_ref_time;
print OUTFILE "$send_time\t$packet_counter \n";

}
#if ($lock==1)
#{
#$send_time=$line_sender[2]-$start_ref_time;
#print OUTFILE "$send_time\t0 \n";

#}
}
close (OUTFILE);
& avg_lossvstime;
}
sub avg_lossvstime{

unless (open (OUTFILE, ">lossvstime"))
{

        die ("cannot open output file outfile\n");

}
}

```

```

open (LOSSES, "<pack_num_seq_loss_vs_time");

my (@sen_lines) = <LOSSES>;
my ($sen_line);
my ($temp1);
my ($temp);
my ($lock);
my ($lock1);
my ($send_time);
my ($packet_counter);
my ($line_counter);

close (LOSSES);

$size=@rec_lines;
$size1=@sen_lines;
$packet_counter=0;
$lock_ref_time=0;
$line_counter=0;
$lock=0;

$lock1=0;
foreach $sen_line (@sen_lines)
    {

$packet_counter++;
    chomp($sen_line);
        @line_sender=split(/\t +/,$sen_line);
        $temp=$line_sender[0];
        if ($lock==0)
        {
            $temp1=$line_sender[0];
            #print "$temp1\n";
            @line_sender_last=split(/\t +/,$sen_lines[(($sen_lines) -1)]);
            #print "$line_sender_last[0]\n";
            # $packet_counter=0;

```

```

$lock=1;
}

if (($temp1+1.0) < ($temp))
{
  if ($lock==0)
  {
    $pack_count=$packet_counter-1;
    print OUTFILE "$send_time\t$pack_count \n";
    #print "$send_time\t$packet_counter\t$temp \n";
    $packet_counter=1;
    $temp1=$temp;
  }
}

if (($temp1+1) > ($line_sender_last[0]))
{
  $lock=1;
  if ($temp==$line_sender_last[0])
  {
    $send_time=$temp;
    # print "$send_time\t$packet_counter\t$temp \n";
    print OUTFILE "$send_time\t$packet_counter \n";
  }

}

$send_time=$temp;
$line_counter++;
}
close (OUTFILE);
#print "$pack_count\t$packet_counter\t$line_counter\n";
}

```

7.4.3 Πρόγραμμα `ipv4_sender_receiver_rate.pl`

```
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
#Author Anargiros Sideris
#Mail irons@pasiphae.teiher.gr
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

#!/usr/bin/perl -w

my($sum_pack_size);
my($lock);
my($start_time);
my($send_time);
my($transfer_time);
my($data_rate);
$data_rate=0;
$transfer_time=0;
$sum_pack_size=0;
$lock=0;
$start_time=0;
$send_time=0;

open(SENDER, "<final.tx") || die ("cannot open input file 1\n");
```

```

$start = time;

while (<SENDER>)
{
    my($sen_line) = $_;
    chomp($sen_line);

    @line_sender=split(/\t +/,$sen_line);
    $sum_pack_size+=$line_sender[3];
    if ($lock==0)
    {
        $start_time=$line_sender[2];
        $lock=1;
    }

    #print "$papa[3]\n";

}
$end_time=$line_sender[2];
#print "stime $start_time etime $end_time\n";
$transfer_time=$end_time-$start_time;
$data_rate=$sum_pack_size/$transfer_time;
print "SENDER RESULTS\n";
print "total bytes transferred $sum_pack_size in $transfer_time sec. Sender output data rate is $data_rate
bytes/sec \n";
close(SENDER);

open(RECEIVER, "<final.rx") || die ("cannot open input file 2\n");
$data_rate=0;
$transfer_time=0;
$sum_pack_size=0;
$lock=0;
$start_time=0;
$end_time=0;
while (<RECEIVER>)

```

```

{
my($sen_line) = $_;
chomp($sen_line);

@line_receiver=split(/\t +/,$sen_line);
$sum_pack_size+=$line_receiver[3];
if ($lock==0)
{
$start_time=$line_receiver[2];
$lock=1;
}

#print "$papa[3]\n";

}
$end_time=$line_receiver[2];
print "stime $start_time etime $end_time\n";
$transfer_time=$end_time-$start_time;
$data_rate=$sum_pack_size/$transfer_time;
print "RECEIVER RESULTS\n";
print "total bytes transferred $sum_pack_size in $transfer_time sec. Receiver input data rate is $data_rate
bytes/sec \n";

close(RECEIVER);

$elapsed_sec = time - $start;

my $second = $elapsed_sec%60;
my $minute = ($elapsed_sec/60)%60;
my $hour = ($elapsed_sec/(60*60))%24;
print "Total Time elapsed: $hour hours:$minute min:$second sec\n";

```

7.4.4 Πρόγραμμα `ipn4_align_for_delay_jitt.pl`

```
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
#Author Anargiros Sideris
#Mail irons@pasiphae.teiher.gr
#!/usr/bin/perl -w
```

```
unless (open (OUTFILE, ">aligned_sender"))
{
    die ("cannot open output file outfile\n");
}
open (SENDER, "<final.tx");
open (RECEIVER, "<final.rx");
my (@sen_lines) = <SENDER>;
my ($sen_line);
my (@rec_lines) = <RECEIVER>;
my ($rec_line);
my ($temp);
my ($temp1);

close (SENDER);
close (RECEIVER);
$size=@rec_lines;
$size1=@sen_lines;
print "Receiver packets $size --- Sender packets $size1\n";

$start = time;
```



```

foreach $rec_line (@rec_lines)
{
    chomp($rec_line);
    @line_receiver=split(/\t +/,$rec_line);
    $temp=$line_receiver[0].$line_receiver[1];
    foreach $sen_line (@sen_lines)
    {
        chomp($sen_line);
        @line_sender=split(/\t +/,$sen_line);
        $temp1=$line_sender[0].$line_sender[1];
        if ($temp eq $temp1)
        {
            #unless (open (OUTFILE, ">>aligned_sender"))
            #{
            #    die ("cannot open output file outfile\n");
            #}
            print OUTFILE "$sen_line\n";
            #close (OUTFILE);
            last;
        }
        #print " sender $sen_line receiver $rec_line\n";
    }
}

$elapsed_sec = time - $start;

my $second = $elapsed_sec%60;
my $minute = ($elapsed_sec/60)%60;
my $hour = ($elapsed_sec/(60*60))%24;
print "Total Time elapsed: $hour hours:$minute min:$second sec\n";

close (OUTFILE);

```

7.4.5 Πρόγραμμα ipv4_timestamp.pl

```
# This program is free software; you can redistribute it and/or modify
```

```

# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
#Author Anargiros Sideris
#Mail irons@pasiphae.teiher.gr
#!/usr/bin/perl -w
unless (open (SENDER, "<aligned_sender"))
    {
        die ("cannot open input file outfile\n");
    }
unless (open (OUTFILE, ">sender_timestamp"))
    {
        die ("cannot open output file outfile\n");
    }
$start = time;
while (<SENDER>)
    {
        my($sen_line) = $_;
        chomp($sen_line);
        @line_sender=split(/\t +/,$sen_line);

        print OUTFILE "$line_sender[2]\n";
        #print "$papa[3]\n";

    }

close(OUTFILE);
close(SENDER);

```

```

unless (open (RECEIVER, "<final.rx"))
    {
        die ("cannot open input file outfile\n");
    }

unless (open (OUTFILE, ">receiver_timestamp"))
    {
        die ("cannot open output file outfile\n");
    }

while (<RECEIVER>)
    {
        my($sen_line) = $_;
        chomp($sen_line);

        @line_receiver=split(/[t +]/,$sen_line);

        print OUTFILE "$line_receiver[2]\n";
        #print "$papa[3]\n";

    }
close(OUTFILE);
close(RECEIVER);
$elapsed_sec = time - $start;

my $second = $elapsed_sec%60;
my $minute = ($elapsed_sec/60)%60;
my $hour = ($elapsed_sec/(60*60))%24;
print "Total Time elapsed: $hour hours:$minute min:$second sec\n";

```

7.4.6 Πρόγραμμα ipv4_inter_arrival_for_jitter.pl

```

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or

```

```
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
#Author Anargiros Sideris
#Mail irons@pasiphae.teiher.gr
#!/usr/bin/perl -w
```

```
unless (open (SENDER, "<sender_timestamp"))
{
    die ("cannot open input file outfile\n");
}
unless (open (RECEIVER, "<receiver_timestamp"))
{
    die ("cannot open input file outfile\n");
}
```

```
my (@sen_lines) = <SENDER>;
my ($sen_line);
my (@rec_lines) = <RECEIVER>;
my ($rec_line);
my($transit);
my($delta_transit);
my($last_transit);
my($jitter);
my($counter);
$counter=0;
$transit=0;
$delta_transit=0;
$last_transit=0;
$jitter=0;

close(SENDER);
```

```

close(RECEIVER);

unless (open (OUTFILE, ">final_jitter"))
    {
        die ("cannot open output file outfile\n");
    }

unless (open (OUTFILE1, ">final_pack2packdelay"))
    {
        die ("cannot open output file outfile\n");
    }

$start = time;

foreach $sen_line (@sen_lines)
{
    chomp($sen_line);

    $transit=$rec_lines[$counter]-$sen_line;
    if ($last_transit!=0)
    {
        $delta_transit=$transit-$last_transit;
        if ( $delta_transit < 0 ) {
            $delta_transit = -$delta_transit;
        }

        $jitter+=$(delta_transit-$jitter)/16.0;

    }
    $last_transit=$transit;
    $result=$jitter*1000;
    $timerec=$rec_lines[$counter];
    chomp($timerec);
    print OUTFILE "$timerec $result\n";
    $pack_delay=$delta_transit*1000;
    print OUTFILE1 "$timerec $pack_delay\n";
    #print "$sen_line $rec_lines[$counter] $result\n";

```

```

    $counter++;

}

close(OUTFILE);
close(OUTFILE1);

unless (open (INFILE, "<final_jitter"))
    {
        die ("cannot open input file outfile\n");
    }
$min=100000;
$max=0;
$counter=0;
$result=0;
$lock=0;
while (<INFILE>)
{

my($sen_line) = $_;
  chomp($sen_line);
  @values=split(/\t +/,$sen_line);
  $value=$values[1];
  $result+=$value;
  if ($counter==1)
    {
    $min=$value;

}

if ($value>$max)
{
  $max=$value;

}
if (($value<$min))
{

```

```

    $min=$value;
}
$counter++;

}

close (INFILE);
$result=$result/$counter;

print "aver jitter is $result max is $max min $min\n";

unless (open (INFILE, "<final_pack2packdelay"))
    {
        die ("cannot open input file outfile\n");
    }
$min=100000;
$max=0;
$counter=0;
$result=0;
$lock=0;
while (<INFILE>)
{
    my($sen_line) = $_;
    chomp($sen_line);

    @values=split(/\t +/,$sen_line);
    $value=$values[1];
    $result+=$value;
    if ($counter==1)
        {
            $min=$value;
        }
    if ($value>$max)
        {
            $max=$value;
        }
    if (($value<$min))

```

```

{
    $min=$value;
}
$counter++;
}

close (INFILE);
$result=$result/($counter);

print "aver pack2packdelay is $result max is $max min $min\n";

unless (open (INFILE, "<final_pack2packdelay"))
    {
        die ("cannot open input file outfile\n");
    }

unless (open (OUTFILE, ">timed_final_pack2packdelay"))
    {
        die ("cannot open input file outfile\n");
    }

my (@times) = <INFILE>;
close(INFILE);
$time=0;
$lock=0;
for ($i=0;$i<@times-1;$i++)
{
    if ($lock==0)
    {
        @valuesplits=split(/\t +/,$times[$i]);
        $valuesplit=$valuesplits[1];
chomp($valuesplit);
        print OUTFILE "$time $valuesplit\n";
        $lock=1;
    }
    #chomp($sen_line);
    @timesplit=split(/\t +/,$times[$i+1]);

```



```

$temp_time1=$timesplit[0];
@timesplit=split(/\t +/,$times[$i]);
$temp_time2=$timesplit[0];
$time=($temp_time1-$temp_time2)+$time;
#$time=$timesplit[0];
@valuesplits=split(/\t +/,$times[$i+1]);
$valuesplit=$valuesplits[1];
chomp($valuesplit);
print OUTFILE "$time $valuesplit\n";
#print "$time\n";
}

close(OUTFILE);

unless (open (INFILE, "<final_jitter"))
    {
        die ("cannot open input file outfile\n");
    }

unless (open (OUTFILE, ">timed_final_jitter"))
    {
        die ("cannot open input file outfile\n");
    }
my (@times) = <INFILE>;
close(INFILE);
$time=0;
$lock=0;
for ($i=0;$i<@times-1;$i++)
    {
        if ($lock==0)
        {
            @valuesplits=split(/\t +/,$times[$i]);
            $valuesplit=$valuesplits[1];
chomp($valuesplit);
            print OUTFILE "$time $valuesplit\n";
$lock=1;
        }
    }

```

```

#chomp($sen_line);
@timesplit=split(/\t +/,$times[$i+1]);
$stemp_time1=$timesplit[0];
@timesplit=split(/\t +/,$times[$i]);
$stemp_time2=$timesplit[0];
$time=($stemp_time1-$stemp_time2)+$time;
#$time=$timesplit[0];
@valuesplits=split(/\t +/,$times[$i+1]);
$valuesplit=$valuesplits[1];
chomp($valuesplit);
print OUTFILE "$time $valuesplit\n";
#print "$time\n";
}
close(OUTFILE);
$elapsed_sec = time - $start;
my $second = $elapsed_sec%60;
my $minute = ($elapsed_sec/60)%60;
my $hour = ($elapsed_sec/(60*60))%24;
print "Total Time elapsed: $hour hours:$minute min:$second sec\n";

```

7.4.7 Πρόγραμμα ipn4_jitter.pl

```

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
#Author Anargiros Sideris

```

```

#Mail irons@pasiphae.teiher.gr
#!/usr/bin/perl -w
unless (open (SENDER, "<sender_timestamp"))
    {
        die ("cannot open input file outfile\n");
    }
unless (open (RECEIVER, "<receiver_timestamp"))
    {
        die ("cannot open input file outfile\n");
    }
my (@sen_lines) = <SENDER>;
my (@rec_lines) = <RECEIVER>;
my($jitter);
my($counter);
my($avg_jitter);
my($sample_time);
my($min);
my($max);
$counter=0;
$jitter=0;
$avg_jitter=0;
$sample_time=0;
$max=-1;
$min=1000000;
close(SENDER);
close(RECEIVER);

unless (open (OUTFILE, ">jittervstime"))
    {
        die ("cannot open output file jittervstime\n");
    }

$start = time;

# Jitter formula is  $D_i = \text{abs}(R_{(i)} - R_{(i-1)}) - (S_{(i)} - S_{(i-1)})$ 
# Avg jitter is  $\text{Sum}(D_i)/n$ 

for ($i=0;$i<@sen_lines-1;$i++)

```

```

{
    #print("Sender line $sen_lines[$i]\n");
    #print("Receiver line $rec_lines[$i]\n");
    $jitter=abs(($rec_lines[$i+1]-$rec_lines[$i])-(($sen_lines[$i+1]-$sen_lines[$i]))*1000;
    $avg_jitter+=$jitter;
    if ($min>$jitter)
    {
        $min=$jitter;
    }
    if ($max<$jitter)
    {
        $max=$jitter;
    }
    $sample_time=$rec_lines[$i+1]-$rec_lines[0];
    #print ("$sample_time $jitter\n");
    print OUTFILE "$sample_time $jitter\n";
    $counter++;
}
$avg_jitter=($avg_jitter/$counter);
print ("Average Jitter is $avg_jitter ms. Max jitter is $max ms. Min jitter is $min ms\n");

close(OUTFILE);

$elapsed_sec = time - $start;

my $second = $elapsed_sec%60;
my $minute = ($elapsed_sec/60)%60;
my $hour = ($elapsed_sec/(60*60))%24;
    print "Total Time elapsed: $hour hours:$minute min:$second sec\n";

```

7.4.8 Πρόγραμμα `ipv4_one_way_delay.pl`

```

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or

```

```
# (at your option) any later version.
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
#Author Anargiros Sideris
#Mail irons@pasiphae.teiher.gr
```

```
#!/usr/bin/perl -w
```

```
unless (open (SENDER, "<sender_timestamp"))
{
    die ("cannot open input file outfile\n");
}
unless (open (RECEIVER, "<receiver_timestamp"))
{
    die ("cannot open input file outfile\n");
}
```

```
my (@sen_lines) = <SENDER>;
my (@rec_lines) = <RECEIVER>;
my($delay);
my($counter);
my($avg_delay);
my($sample_time);
my($min);
my($max);
$counter=0;
$delay=0;
$avg_delay=0;
$sample_time=0;
$max=-1;
$min=1000000;
close(SENDER);
close(RECEIVER);
```

```

unless (open (OUTFILE, ">one_way_delayvstime"))
    {
        die ("cannot open output file jittervstime\n");
    }

$start = time;

# One way delay formula is  $D_i = \text{abs}(R_i - S_i)$ 
#Avg One way Delay is  $\text{Sum}(D_i)/n$ 

for ( $\$i=0; \$i < @sen\_lines; \$i++$ )
{

    #print("Sender line  $\$sen\_lines[\$i]\n$ ");
    #print("Receiver line  $\$rec\_lines[\$i]\n$ ");
    $delay=abs(($rec_lines[$i])-( $sen_lines[$i]))*1000;
    $avg_delay+=$delay;
    if ( $\$min > \$delay$ )
    {
        $min=$delay;
    }
    if ( $\$max < \$delay$ )
    {
        $max=$delay;
    }
    $sample_time=$rec_lines[$i]-$rec_lines[0];
    #print (" $\$sample\_time \$jitter\n$ ");
    print OUTFILE " $\$sample\_time \$delay\n$ ";
    $counter++;
}
$avg_delay=($avg_delay/$counter);
print ("Average One way Delay is  $\$avg\_delay$  ms. Max One way Delay is  $\$max$  ms. Min One way Delay is  $\$min$  ms\n");

close(OUTFILE);

```

```

$elapsed_sec = time - $start;

my $second = $elapsed_sec%60;
my $minute = ($elapsed_sec/60)%60;
my $hour = ($elapsed_sec/(60*60))%24;
    print "Total Time elapsed: $hour hours:$minute min:$second sec\n";

```

7.4.9 Πρόγραμμα losses.gpl

```

set title "Losses vs Time"
set xlabel "time (sec)"
set ylabel "Lost Packets"
set format x "%.0f"
set format y "%.0f"
set yrange [0:*]
set xdata time
set nokey
#load "a2b_tput.labels";
plot "lossvstime" using ($1-946684800.0):2 with points pointtype 5 pointsize 1;
set terminal png color ;

set output "losses.png"
replot
pause -1;

```

7.4.10 Πρόγραμμα onewaydelay.gpl

```

set title "One Way Delay"
set xlabel "time (sec)"
set ylabel "One Way delay (ms)"
set format x "%.0f"

```

```

set format y "%.0f"
set xdata time
set nokey
#load "a2b_tput.labels";
plot "one_way_delayvstime" using ($1-946684800.0):2 with points pointtype 5 pointsize
1;
set terminal png color;
#set
set output "one_way.png"
replot
pause -1;

```

7.4.11 Πρόγραμμα jitter.gpl

```

set title "Jitter"
set xlabel "time (sec)"
set ylabel "Jitter (ms)"
set format x "%.0f"
set format y "%.0f"
set xdata time
set nokey
#load "a2b_tput.labels";
plot "jittervstime" using ($1-946684800.0):2 with points pointtype 5 pointsize 1;
set terminal png color ;

set output "jitter.png"
replot
pause -1;

```