ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΗΡΑΚΛΕΙΟΥ

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Τίτλος Πτυχιακής Εργασίας

"Χρήση DVB-T ρεύματος μεταφοράς για ασύρματη ευρυζωνική δικτύωση"

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: **ΞΥΛΟΥΡΗΣ ΧΡΗΣΤΟΣ** ΑΜ: **362**
ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ: **17/05/20007**

ΕΙΣΗΓΗΤΗΣ: **Δρ. Ζαχαρόπουλος Βασίλειος**

Σε όσους με βοήθησαν για την
υλοποίηση της πτυχιακής μου εργασίας,
με ιδιαίτερη εκτίμηση

Ευχαριστίες

Με την επιτυχή ολοκλήρωση της πτυχιακής μου εργασίας, η οποία υλοποιήθηκε στο Εργαστήριο Τηλεπικοινωνιών και Δικτύων (ΠΑΣΙΦΑΗ) του Α.Τ.Ε.Ι Κρήτης, θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν με το δικό τους τρόπο για το σκοπό αυτό. Θα ήταν παράλειψη να μην αναφερθώ σε όλους εκείνους που μου συμπαραστάθηκαν σε αυτήν την προσπάθεια.

Κατά κύριο λόγο, οφείλω να ευχαριστήσω τον επιβλέποντά μου από το Α.Τ.Ε.Ι Κρήτης και Διδάκτορα, κ. Βασίλειο Ζαχαρόπουλο ο οποίος με υποστήριξε σε όλη τη διάρκεια της πτυχιακής εργασίας. Επίσης οφείλω ένα μεγάλο ευχαριστώ στον Υποψήφιο Διδάκτορα στο Α.Τ.Ε.Ι Κρήτης κ. Γεώργιο Μαστοράκη γιατί όποτε τον χρειάστηκα βοήθησε στην διεκπεραίωση της πτυχιακής μου. Αισθάνομαι όσο λίγοι, ευνοημένος που στο διάστημα αυτό ένιωθα πάντα τη σιγουριά ότι σε κάθε βήμα είχα την υλική αλλά και ηθική βοήθεια που χρειαζόμουν για να προχωρήσω στην ολοκλήρωση αυτής της εργασίας. Ιδιαίτερες ευχαριστίες οφείλονται επίσης και στον εργαστηριακό συνεργάτη του εργαστηρίου Τηλεπικοινωνιών και Δικτύων (ΠΑΣΙΦΑΗ) του Α.Τ.Ε.Ι ΚΡΗΤΗΣ, κ. Σιδέρη Ανάργυρο για την πολύτιμη βοήθεια και υποστήριξη όποτε αυτή χρειάστηκε. Επίσης οφείλω ένα μεγάλο ευχαριστώ και στους συναδέλφους μου, τελειόφοιτους φοιτητές του Τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων, Παπαδουράκη Γεώργιο και Μπουρδένα Αθηνά.

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

Ηράκλειο, Μάιος 2007

Χρήστος Ξυλούρης

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ 8
1.1 Γενική Περιγραφή 8
1.2. Στόχοι 8
1.3. Δομή 8
1.4. Θεωρητικό υπόβαθρο για την τεχνολογία Digital Video Broadcasting-Terrestrial (DVB-T) 9
1.5. Σύνοψη 9
2. ΨΗΦΙΑΚΗ ΤΗΛΕΟΡΑΣΗ ΚΑΙ ΔΙΑΔΡΑΣΤΙΚΕΣ ΥΠΗΡΕΣΙΕΣ 10
2.1. Γενικά 10
2.2. Το πρότυπο συμπίεσης MPEG-2 11
2.2.1. Αρχές λειτουργίας MPEG-2 12
2.2.1.1. Συμπίεση 12
2.2.1.2. Σήμα βασικής ζώνης 12
2.2.1.3. Ροή μεταφοράς MPEG-2 14
2.2.1.4. Πολυπλεξία ψηφιακών τηλεοπτικών προγραμμάτων 17
2.3. Πρότυπα ψηφιακής τηλεόρασης 18
2.4. Αρχές λειτουργίας επίγειας ψηφιακής τηλεόρασης 19
2.5. Μετάδοση δεδομένων IP-Ενθυλάκωση 21
2.5.1. Ενθυλάκωση πολλαπλών πρωτοκόλλων(multi-protocol encapsulation) 22
2.5.2. Το πρωτόκολλο TCP και η σημασία του 24
2.5.3. Το πρωτόκολλο UDP και η σημασία του 25
2.6. Κωδικοποιημένη Πολυπλεξία με Διαίρεση σε Ορθογώνιες Συχνότητες (COFDM) 25
2.7. Κωδικοποίηση και Διεμπλοκή 26
2.7.1. Εξωτερική κωδικοποίηση και διεμπλοκή 27
2.7.2. Εσωτερική κωδικοποίηση και διεμπλοκή 28
2.8. Πομποί ψηφιακής τηλεόρασης 28
3 ΠΑΡΟΥΣΙΑΣΗ ΔΙΚΤΥΟΥ ΠΑΡΟΧΗΣ ΑΜΦΙΔΡΟΜΩΝ ΔΙΑΔΡΑΣΤΙΚΩΝ ΥΠΗΡΕΣΙΩΝ 30
3.1 Εισαγωγή 30
3.2. Μελέτη δικτύου επίγειας ψηφιακής τηλεόρασης με κανάλια επιστροφής τις τεχνολογίες WLAN και ISDN 30
3.3 Πρωτόκολλο IPv6 31
3.3.1 Διευθυνσιοδότηση IPv6 32
3.4. Υλοποίηση με ISDN Κανάλι Επιστροφής 32
3.5. Χαρακτηριστικά ηλεκρονικών υπολογιστών που χρησιμοποιήθηκαν για την υλοποίηση του δικτύου 34
3.6. Ρυθμίσεις της πλατφόρμας DVB-T 34
3.7 Ενδιάμεσος Κόμβος Διανομής (Cell Main Node) 35
3.7.1 Απαιτήσεις Υλικού (Hardware Requirements) 36
3.7.2. Απαιτήσεις Λογισμικού (Software Requirements) 39
3.8. Η συμπεριφορά του CMN σε δικτυακή κίνηση 41
3.8.1. Μετάδοση δεδομένων με την αξιοποίηση κίνησης TCP/IP και UDP 41
4. ΣΤΟΙΧΕΙΟΘΕΤΗΣΗ ΚΑΙ ΣΧΟΛΙΑΣΜΟΣ ΠΕΙΡΑΜΑΤΙΚΩΝ ΜΕΤΡΗΣΕΩΝ 51

4.1. Εισαγωγη 51
4.2. Προγράμματα δημιουργίας και ανάλυσης της κίνησης ενός δικτύου 51
4.2.1. Δημιουργία κίνησης 51
4.2.1.1. Πρόγραμμα Iperf 51
4.2.1.2 Πρόγραμμα Multi-Generator (MGEN) 52
4.2.2 Ανάλυση κίνησης 52
4.2.2.1 Πρόγραμμα Tcpdump 52
4.2.2.2 Πρόγραμμα Tcptrace 53
4.2.2.3. Πρόγραμμα Xpl2gpl 53
4.2.2.4. Προγράμματα της γλώσσας προγραμματισμού Perl 53
4.2.3. Παράδειγμα δημιουργίας – ανάλυσης δικτυακής κίνησης TCP 53
4.2.3.1. Δημιουργίας δικτυακής κίνησης TCP 53
4.2.3.2 Ανάλυση δικτυακής κίνησης TCP 56
4.2.4 Παράδειγμα δημιουργίας – ανάλυσης δικτυακής κίνησης UDP 61
4.2.4.1 Δημιουργία δικτυακής κίνησης TCP 61
4.2.4.2 Ανάλυση δικτυακής κίνησης UDP 63
4.3 Πειραματικές μετρήσεις δικτύου επίγειας ψηφιακής τηλεόρασης με κανάλι επιστροφής τις τεχνολογίες ISDN και WLAN 67
4.3.1 Γενικά 67
4.3.2. Μετάδοση δεδομένων κάνοντας χρήση του πρωροκόλλου TCP/IP 67
4.3.2.1 Γραφικές παραστάσεις δικτυακής κίνησης TCP 68
4.3.3 Μετάδοση δεδομένων κάνοντας χρήση του πρωροκόλλου UDP/IP 72
4.3.3.1 Γραφικές παραστάσεις δικτυακής κίνησης UDP 73
4.3.3.2 Ανάλυση ποιότητας UDP κίνησης 76
5. Συμπεράσματα και μελλοντικές προτάσεις 76

Ακρωνύμια 78

ΠΑΡΑΡΤΗΜΑ Α 81

Βιβλιογραφία 107

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

Σχ. 1 Ψηφιοποίηση και μετάδοση σήματος 10
Σχ. 2 Τύποι συμπίεσης του προτύπου MPEG-2 12
Σχ. 3 Πολυπλεξία MPEG-2 13
Σχ. 4 Δομή πακέτου μεταφοράς 14
Σχ. 5 Πολυπλέκτης MPEG-2 TS 15
Σχ. 6 Δομή ροής MPEG-2 TS 16
Σχ. 7 Δομή πακέτου MPEG-2 TS 16
Σχ. 8 Πολυπλεξία δύο ροών 17
Σχ. 9 Διάγραμμα διαμορφωτή DVB-T 20
Σχ. 10 Τυχαιοποίηση δεδομένων 21
Σχ. 11 Ενσωμάτωση της MAC address στο section header 22
Σχ. 12 Τεχνική ενθυλάκωσης πολλαπλών πρωτοκόλλων με πληροφορία stuffing 23
Σχ. 13 Τεχνική ενθυλάκωσης πολλαπλών πρωτοκόλλων (MPE) 23
Σχ. 14 Πεδία επικεφαλίδας πλαισίου TCP 24
Σχ. 15 Η επικεφαλίδα UDP 25
Σχ. 16 Φέρουσες COFDM 26
Σχ. 17 Επεξεργασία πακέτου MPEG2-TS 27
Σχ. 18 Μορφή δεδομένων μετά τις διαδικασίες: προσαρμογή, ... 28
Σχ. 19 Λειτουργικό διάγραμμα πομπού 29
Σχ. 20 Εντολές καθορισμού διευθύνσεων IPv6 32
Σχ. 21 Αρχιτεκτονική δικτύου αμφίδρομης επίγεια 33
Σχ. 22 Ο διαμορφωτής COFDM που παρουσιάζει τα στοιχεία της διαμόρφωσης 34
Σχ. 23 Ο διαμορφωτής COFDM που παρέχει το DVB-T 35
Σχ. 24 Αρχιτεκτονική CMN 36
Σχ. 25 Ενδιάμεσος Κόμβος Διανομής (CMN), κάρτα επίγειας 37
Σχ. 26 Κάρτα επίγειας ψηφιακής τηλεόρασης WinTV-NOVA-T-USB-2 38
Σχ. 27 Κάρτα Δικτύου, Intel(R) Pro/100 s Desktop Adapter 38
Σχ. 28 ISDN Network Termination Unit της INTRACOM 39
Σχ. 29 Level-One Wireless Lan PCI Adapter 11g 39
Σχ. 30 Εντολή ip_forwarding 41
Σχ. 31 Εντολή xconfig 42
Σχ. 32 Εφαρμογή xconfig 43
Σχ. 33 Επιλογή κάρτας επίγειας ψηφιακής τηλεόρασης 44
Σχ. 34 Εντολή rp-filter 45
Σχ. 35 Εντολή ip-route για την δρομολόγηση της 45
Σχ. 36 Δρομολόγηση των κινήσεων μέσα στον CMN του server 46
Σχ. 37 Δρομολόγηση των κινήσεων μέσα στον CMN του server 47
Σχ. 38 Εντολή iptables 48
Σχ. 39 Αρχείο “athena” 48
Σχ. 40 Διαθέσιμα τηλεοπτικά προγράμματα 49
Σχ. 41 Συντονισμός στο τηλεοπτικό πρόγραμμα “DSO TV” 50
Σχ. 42 Εντολή tcpdump στον Server 54
Σχ. 43 Εντολή tcpdump στον Client 54

Σχ. 44 Εντολή iperf στον Server 55
Σχ. 45 Εντολή iperf στον Client 55
Σχ. 46 Εντολή tcptrace για το TCP Delay του δικτύου 56
Σχ. 47 Εντολή xpl2gpl για το RTT 57
Σχ. 48 Εντολή perl για το RTT 57
Σχ. 49 Εντολή tcptrace για τον υπολογισμό του throughput (ανά δευτερόλεπτο) 58
Σχ. 50 Εντολή xpl2gpl για το Throughput 58
Σχ. 51 Εντολή perl για το Throughput 59
Σχ. 52 Εντολή tcptrace για τον υπολογισμό του Time sequence(ανά segments) 60
Σχ. 53 Εντολή xplot για Time sequence 60
Σχ. 54 Εντολή tcpdump στον Server (Δικτυακή κίνηση UDP) 61
Σχ. 55 Εντολή tcpdump στον Client (Δικτυακή κίνηση UDP) 62
Σχ. 56 Εντολή Mgen στον Server (Δικτυακή κίνηση UDP) 62
Σχ. 57 Εντολή Mgen στον Client (Δικτυακή κίνηση UDP) 63
Σχ. 58 Εντολή tcpdump για τα στατιστικά των αρχείων S256k και C256k 63
Σχ. 59 Perl ipv6_losses.pl 64
Σχ. 60 Perl ipv6_sender_receiver_rate.pl 64
Σχ. 61 Perl ipv6_inter_arrival_jitter.pl 65
Σχ. 62 Perl ipv6_one_way_delay.pl 66
Σχ. 63 Perl ipv6_ jitter.pl 66
Σχ. 64 Γραφική παράσταση του Round-trip Time 68
Σχ. 65 Γραφική παράσταση του Throughput 69
Σχ. 66 Γραφική παράσταση του Time sequence στα 8KByte 70
Σχ. 67 Γραφική παράσταση(με zoom) του Time sequence στα 8KByte 70
Σχ. 68 Γραφική παράσταση(με zoom) του Time sequence στα 32KByte 71
Σχ. 69 Γραφική παράσταση(με zoom) του Time sequence στα 64KByte 71
Σχ. 70 Γραφική παράσταση(με zoom) του Time sequence στα 512KByte 72
Σχ. 71 Γραφική παράσταση του Packet to Packet Delay 73
Σχ. 72 Γραφική παράσταση του Jitter 74
Σχ. 73 Γραφική παράσταση του One_Way_delay 74

1. ΕΙΣΑΓΩΓΗ

1.1 Γενική Περιγραφή

Η πτυχιακή αυτή εργασία χρησιμοποιεί την πλατφόρμα επίγειας ψηφιακής τηλεόρασης για την υλοποίηση ενός υβριδικού δικτύου παροχής αμφίδρομων διαδραστικών υπηρεσιών. Με την χρήση μιας συγκεκριμένης αρχιτεκτονικής, η οποία κάνει χρήση Digital Video Broadcasting-Terrestrial στην κατερχόμενη ζεύξη και των τεχνολογιών WLAN και ISDN στο κανάλι επιστροφής, δίνεται πρόσβαση στους χρήστες σε διαδραστικές υπηρεσίες. Αυτό σημαίνει ότι ένας χρήστης μπορεί να επιλέξει ο ίδιος τι θα ζητήσει από τον παροχέα του. Επίσης, θα υλοποιηθεί ο σχεδιασμός και η αξιολόγηση της απόδοσης του προτεινόμενου δικτύου με την πραγματοποίηση πειραματικών μετρήσεων.

1.2. Στόχοι

Στόχος της πτυχιακής είναι να συνισφέρει στη σύγκλιση της τεχνολογίας DVB-T με τα δίκτυα τηλεπικοινωνιών με τη χρήση ενσύρματων επίγειων δικτύων. Το δίκτυο που προτείνεται θα υλοποιηθεί και θα δοκιμαστεί στο εργαστήριο Έρευνας και Ανάπτυξης Τηλεπικοινωνιακών Συστημάτων (ΠΑΣΙΦΑΗ). Θα πραγματοποιηθούν μετρήσεις και παρουσίαση γραφικών παραστάσεων για την αξιολόγηση του δικτύου ενώ θα αναφερθούν τα συμπεράσματα που εξάγονται από τις αυτές καθώς και ο σχολιασμός αυτών.

1.3. Δομή

Για την κάλυψη των θεμάτων που αναφερθήκανε παραπάνω θα εξετασθούν όλες οι παράμετροι της λειτουργίας του δικτύου καθώς θα αναφερθούν και τρόποι βελτίστοποίησης του.

Στο κεφάλαιο 2 της πτυχιακής παρουσιάζονται θεωρητικά στοιχεία της ψηφιακής τηλεόρασης και η διαδικασία δημιουργίας σημάτων βασικής ζώνης. Στη συνέχεια γίνεται περιγραφή της λειτουργίας του πιο διαδεδομένου προτύπου μεταφοράς MPEG-2 με αναφορά που αφορούν τις λειτουργίες συμπίεσης και δημιουργιίας της MPEG-2 ροής μεταφοράς (MPEG-2 Transport Stream). Επίσης γίνεται μια όσο το δυνατόν λεπτομερής αναφοράς στο πρότυπο επίγειας ψηφιακής τηλεόρασης DVB-T και στις λειτουργίες αυτού. Τέλος παρουσιάζονται τα πρωτόκολλα TCP και UDP και η σημασία τους.

Στο $3^{ο}$ κεφάλαιο γίνεται εκτενής σχολιασμός των μονάδων(υπολογιστών κ.α.) που χρησιμοποιούνται για την πραγματοποίηση του δικτύου που πρόκειται για μια αρχιτεκτονική ασύμμετρων δικτύων (πελάτη-εξθπηρετητή) με τη χρήση των τεχνολογιών WLAN και ISDN τα οποία εφαρμόζονται στα κανάλια επιστροφής. Παρουσιάζεται επίσης σχεδιαστικά το δίκτυο που θα χρησιμοποιηθεί. Τέλος γίνεται η περιγραφή της δημιουργίας ενός ενδιάμεσου κόμβου διανομής (CMN) και το πως λειτουργεί μέσα στο δίκτυο.

Το αντικείμενο του κεφαλαίου 4 είναι η πραγματοποίηση των πειραματικών μετρήσεων πάνω στο δίκτυο μας. Περιγράφονται λεπτομερώς και με τη χρήση σχημάτων οι διαδικασίες της δημιουργίας και της ανάλυσης των μετρήσεων. Γίνεται επίσης η παρουσίαση των προγραμμάτων που χρησιμοποιήθηκαν για να πραγματοποιηθούν αυτές. Στη συνέχεια εξετάζεται, μελετάται και αξιολογείται η απόδοση του δικτύου που έχουμε υλοποιήσει με βάση δύο διαφορετικά είδη κίνησης(TCP και UDP). Τα αποτελέσματα των μετρήσεων αναλύθηκαν και παρουσιάστηκαν σε γραφικές παραστάσεις για την εξαγωγή χρήσιμων συμπερασμάτων της λειτουργίας του δικτύου.

Τέλος στο $5^{ο}$ κεφάλαιο γίνεται η καταγραφή των συμπερασμάτων που εξάγονται από τη μελέτη του αμφίδρομου διαδραστικού δικτύου που προτείνται στην πτυχιακή αυτή.

1.4. Θεωρητικό υπόβαθρο για την τεχνολογία Digital Video Broadcasting- Terrestrial (DVB-T)

Η εξέλιξη του προτύπου επίγειας ψηφιακής τηλεόρασης DVB-T και η εφαρμογή του πάνω σε επίγεια κανάλια είναι μία από τις πιο σημαντικές τεχνολογικές εξελίξεις στις ασύρματες τηλεπικοινωνίες. Μια πλατφόρμα DVB-T μπορεί να αξιοποιηθεί όχι μόνο για την εκπομπή "μπουκέτων" τηλεοπτικών προγραμμάτων σε ένα μεγάλο αριθμό τελικών χρηστών κατανεμημένων πάνω σε μεγάλες γεωγραφικά περιοχές αλλά και για την παροχή αμφίδρομων Διαδικτυακών υπηρεσιών. Η χαρακτηριστική ιδιότητα του DVB-T να συνδυάζει τηλεοπτικά προγράμματα MPEG-2 με υπηρεσίες IP στο ίδιο κανάλι μεταφοράς, επιτρέπει την ανάπτυξη δικτυακών υποδομών οι οποίες μπορούν να διασυνδέσουν ενδιάμεσους κόμβους διανομής για την παροχή υπηρεσιών κατά απαίτηση.

1.5. Σύνοψη

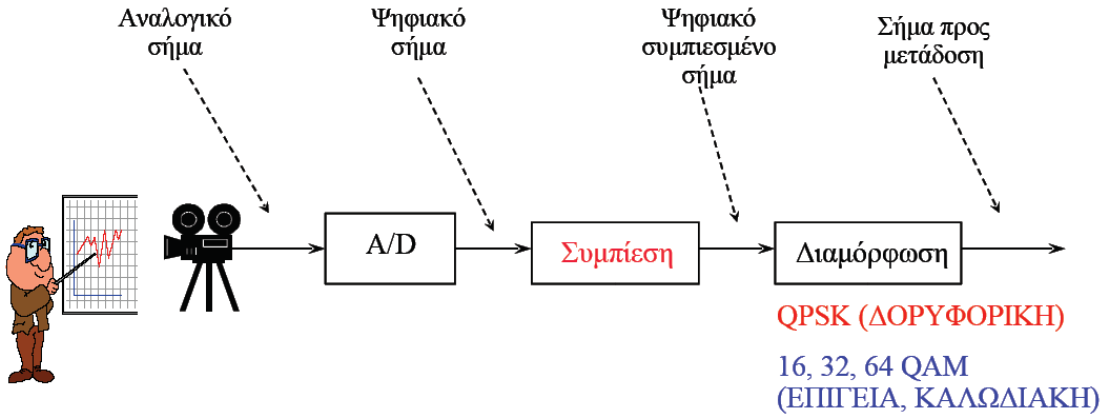
Η μελέτη αυτή υλοποιεί την ιδέα της σύγκλισης των τηλεπικοινωνιών και των τεχνολογιών εκπομπής διερευνώντας ένα μοντέλο υβριδικού ευρυζωνικού δικτύου βασισμένου στην επίγεια ψηφιακή τηλεόραση (DVB-T). Η αρχιτεκτονική που προτείνεται συνδυάζει την τεχνολογία του DVB-T με μια ήδη υπάρχουσα δικτυακή υποδομή, για να προσφέρει στον τελικό χρήστη ψηφιακά τηλεοπτικά προγράμματα μαζί με υψηλής ταχύτητας υπηρεσίες δεδομένων.

2. ΨΗΦΙΑΚΗ ΤΗΛΕΟΡΑΣΗ ΚΑΙ ΔΙΑΔΡΑΣΤΙΚΕΣ ΥΠΗΡΕΣΙΕΣ

2.1. Γενικά

Η ψηφιακή τηλεόραση έχει αρχίσει να αναπτύσσεται με γοργούς ρυθμούς σηματοδοτώντας μια νέα εποχή στις τηλεπικοινωνίες και αναμένεται να μετατρέψει τον χρήστη από παθητικό δέκτη σε ενεργό χρήστη προγραμμάτων και υπηρεσιών.

Στην ψηφιακή τηλεόραση γίνεται χρήση της ψηφιακής τεχνολογίας για την μετάδοση τηλεοπτικών προγραμμάτων. Την τελευταία δεκαετία χάρη στα πρότυπα συμπίεσης και πολυπλεξίας εικόνας της οικογένειας MPEG και του ευρωπαϊκού συνόλου προδιαγραφών DVB για διαμόρφωση και μετάδοση, υπάρχει η δυνατότητα δημιουργίας μίας πλήρης αλυσίδας εκπομπής (Σχ. 1) για την παροχή πλούσιου οπτικοακουστικού περιεχομένου και διαδικτυακών υπηρεσιών.



Σχ. 1 Ψηφιοποίηση και μετάδοση σήματος

Είναι πολλά τα πλεονεκτήματα τα οποία εισάγει η ψηφιακή τεχνολογία στη διαμόρφωση και διανομή του τηλεοπτικού σήματος. Μερικά από αυτά είναι:

- Σταθερή ποιότητα εικόνας, με μεγαλύτερη ανοχή στις ατέλειες του ασύρματου ή ενσύρματου διαύλου. Εξάλειψη φαινομένων όπως αυτά της θόλωσης, των πολλαπλών ειδώλων ή του θορύβου.

- Μειωμένος λόγος σήματος προς θόρυβο (SNR) που απαιτείται στην αναλογική μετάδοση. Αυτό επιτρέπει τη μείωση της εκπεμπόμενης ισχύος μέχρι και κατά 30 db χωρίς να αλλοιώνεται η εικόνα.

- Καλύτερη εκμετάλλευση του φάσματος λόγω της συμπίεσης του σήματος βασικής ζώνης. Για παράδειγμα., ένα ψηφιακό κανάλι UHF αντιστοιχεί σε ένα και μόνο αναλογικό τηλεοπτικό πρόγραμμα, ενώ το ίδιο μπορεί να φιλοξενήσει μέχρι και 10 ψηφιακά προγράμματα υψηλής ποιότητας.

- Ευέλικτες τεχνικές πολυπλεξίας ψηφιακών συστημάτων που επιτρέπουν την συνύπαρξη πολλών προγραμμάτων και υπηρεσιών επιλεγόμενης ποιότητας και ευκρίνειας.

- Μεταβλητή ταχύτητα εκπομπής (bit rate), ανάλογα με τις απαιτήσεις ποιότητας του προγράμματος, κάτι που έχει σαν αποτέλεσμα να αυξηθεί το κέρδος πολυπλεξίας (multiplexing gain) της ταυτόχρονης μετάδοσης πολλών προγραμμάτων μεταβλητού ρυθμού.

- Δυνατότητα επεξεργασίας της εικόνας στο δέκτη με τη βοήθεια αλγορίθμων ψηφιακής επεξεργασίας (digital image post-processing), όπως χρωματική διόρθωση, αποκοπή ορίων, αυξομείωση μεγέθους ή αφαίρεση θορύβου.

- Εμπλουτισμός των τηλεοπτικών προγραμμάτων μέσω τυποποιημένων αρχιτεκτονικών (π.χ. MHP, OpenTV) με τοπικές εφαρμογές που εκμεταλλεύονται τις δυνατότητες των σύγχρονων <<έξυπνων δεκτών>>.

- Ενσωμάτωση διαφόρων πολυμεσικών εφαρμογών και υπηρεσιών δεδομένων, όπως αμφίδρομων υπηρεσιών και διαδικτυακής πρόσβασης σε μια κοινή ψηφιακή πλατφόρμα, με προϋπόθεση ότι υπάρχει διαθέσιμο κανάλι επιστροφής (reverse path).

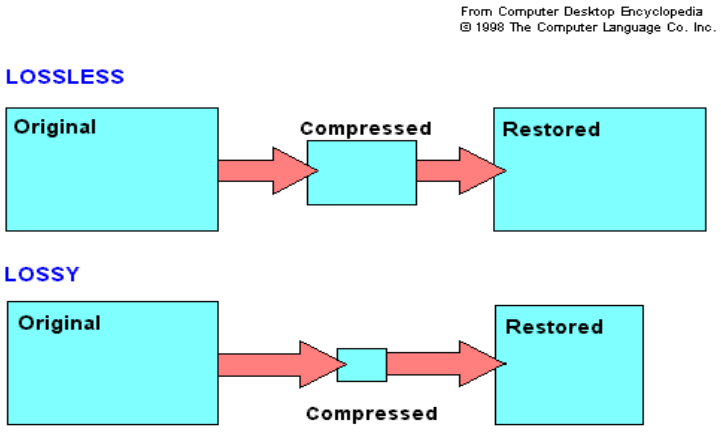
2.2. Το πρότυπο συμπίεσης MPEG-2

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

2.2.1. Αρχές λειτουργίας MPEG-2

2.2.1.1. Συμπίεση

Η συμπίεση των δεδομένων ήχου και video είναι απαραίτητη. Όταν σε ένα αναλογικό σήμα γίνεται δειγματοληψία, το μέγεθος της ψηφιακής πληροφορίας που δημιουργείται είναι πολύ μεγάλο. Με τη σωστή χρήση τεχνικών συμπίεσης μειώνεται το μέγεθος, κάτι που έχει σαν αποτέλεσμα την γρηγορότερη μεταφορά-μετάδοση του ψηφιοποιημένου σήματος ήχου και video. Υπάρχουν δύο τύποι τεχνικών συμπίεσης: η απωλεστική και η μη απωλεστική (Σχ. 2). Με την απωλεστική συμπίεση αποβάλλεται η πληροφορία που δεν είναι απαραίτητη. Με αυτή την τεχνική επιτυγχάνουμε βαθμό συμπίεσης από 2 μέχρι 250 φορές. Αντιθέτως, με την μη απωλεστική τεχνική συμπίεσης γίνεται προσαρμογή της πληροφορίας με διάφορους αλγόριθμους. Ο βαθμός συμπίεσης σε αυτή την τεχνική είναι πολύ μικρότερος(από 2 μέχρι 5 φορές).



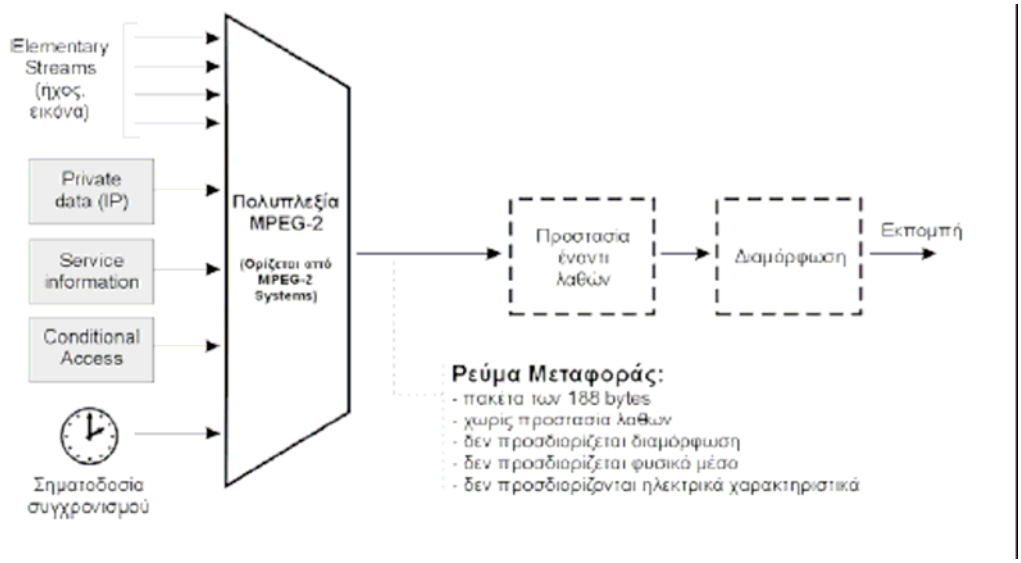
Σχ. 2 Τύποι συμπίεσης του προτύπου MPEG-2

2.2.1.2. Σήμα βασικής ζώνης

Μέχρι και την προηγούμενη δεκαετία η έννοια της ψηφιακής μετάδοσης περιοριζόταν στην απλή ψηφιοποίηση του τηλεοπτικού σήματος σε ασυμπίεστη παλμοσειρά PCM. Αυτή η μετατροπή κατέληγε σε ένα ψηφιακό σήμα με ρυθμό μεταφοράς περίπου 270 Mbps. Το ψηφιακό σήμα περιοριζόταν μόνο σε ενσύρματες υποδομές καθώς κάθε έννοια ασύρματης μετάδοσης δεν είχε κανένα πρακτικό αντίκρισμα.

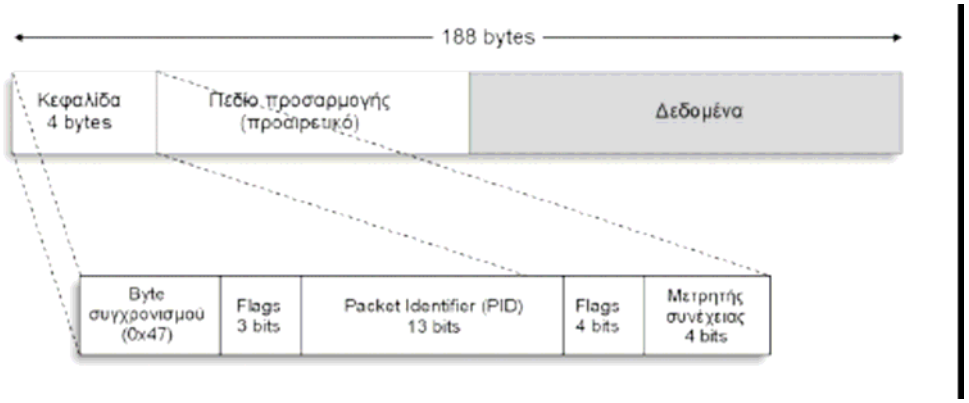
Με την εισαγωγή των αλγόριθμων συμπίεσης εικόνας MPEG-2 πετυχαίνεται συμπίεση του σήματος σε ποσοστά 1:30 ή και ακόμα περισσότερο. Αυτό επιτυγχάνεται με τον χωρισμό της εικόνας σε blocks, διακριτό μετασχηματισμό συνημίτονου και κβάντιση των συντελεστών. Επίσης χρησιμοποιούνται αλγόριθμοι ανίχνευσης και αντιστάθμισης κίνησης(motion detection/compensation) για να περιγραφούν οι αλλαγές και οι μετακινήσεις των αντικειμένων μεταξύ διαδοχικών καρέ.

Το συμπιεσμένο οπτικοακουστικό σήμα φέρει την ονομασία Βασική Ροή (Elementary Stream). Αυτό μαζί με τις Βασικές Ροές άλλων προγραμμάτων αλλά και οποιουδήποτε είδους άλλη ψηφιακή πληροφορία (π.χ δεδομένα TCP/IP), πολυπλέκεται σε μία ροή μεταφοράς (MPEG-2 Transport Stream) που αποτελεί και το σήμα βασικής ζώνης. Η ροή μεταφοράς περιέχει, εκτός από τις βασικές ροές, πληροφορίες για τις διάφορες υπηρεσίες, έλεγχο περιορισμένης πρόσβασης (Conditional Access), σηματοδοσία συγχρονισμού και ιδιωτικά δεδομένα (private data), στα οποία εντάσσονται και τα δεδομένα IP (σχήμα 3)



Σχ. 3 Πολυπλεξία MPEG-2

Η ροή μεταφοράς είναι μια μορφή πολυπλεξίας σχεδιασμένη για σύνθετες εφαρμογές που απαρτίζονται από πολλά ταυτόχρονα προγράμματα και ροές δεδομένων. Αποτελείται από μια σειρά από πακέτα σταθερού μήκους των 188 bytes που ονομάζονται Πακέτα Μεταφοράς(Transport Packets). Όταν κάποια δεδομένα είναι μεγαλύτερα σε μέγεθος (π.χ μία συμπιεσμένη εικόνα, ένα τμήμα ήχου ή ένα πακέτο IP) θα υποστούν κατάτμηση και το περιεχόμενο τους θα μοιραστεί σε περισσότερα του ενός transport packets. Η δομή ενός πακέτου μεταφοράς παρουσιάζεται στο σχήμα 4 :



Σχ. 4 Δομή πακέτου μεταφοράς

Το αναγνωστικό πακέτου(Packet Identifier-PID) χρησιμοποιείται κατά τη διαδικασία πολυπλεξίας και αποπολυπλεξίας για να ξεχωρίσει τα πακέτα που έχουν κοινό περιεχόμενο. Για παράδειγμα, τα πακέτα που φέρουν το σήμα εικόνας ενός τηλεοπτικού προγράμματος έχουνε κοινό PID. Το πεδίο προσαρμογής(adaptation field) χρησιμοποιείται για να συμπληρώσει το τελευταίο από μια σειρά πακέτων που φέρουν ένα κατακερματισμένο σύνολο δεδομένων.

Για την προστασία του ψηφιακού σήματος από λάθη κάνουμε χρήση μικρών και σταθερών μήκους πακέτων. Για παράδειγμα, στα πρότυπα DVB στις διαδικασίες διαμόρφωσης και μετάδοσης, κάθε πακέτο MPEG-2 προστατεύεται με ένα block κώδικα του τύπου Reed-Solomon πριν γίνει η περαιτέρω επεξεργασία στον πομπό. Η αυξημένη ανθεκτικότητα της ροής μεταφοράς του προσφέρει μεγαλύτερες πιθανότητες <<επιβίωσης>> σε ένα κανάλι με λάθη, όπως είναι οποιοδήποτε περιβάλλον ασύρματης εκπομπής.

2.2.1.3. Ροή μεταφοράς MPEG-2

Το πρότυπο MPEG-2 αποτελεί τη βάση για τη δημιουργία ροών μεταφοράς MPEG-2(MPEG-2 TS) οι οποίες περιέχουν πολλαπλές βασικές ροές πακέτων βίντεο, ήχου και δεδομένων. Οι ροές μεταφοράς θα πρέπει να χρησιμοποιούν ηλεκτρικές διεπαφές που να είναι συμβατές με συστήματα μετάδοσης για εκπομπή σε επίγεια ευρυ-εκπομπή. Το πρότυπο MPEG-2 δεν αναφέρεται μόνο στη διαδικασία συμπίεσης του ψηφιακού βίντεο και ήχου αλλά και στη διαδικασία συνένωσης αυτών των σημάτων μαζί με άλλες πηγές.

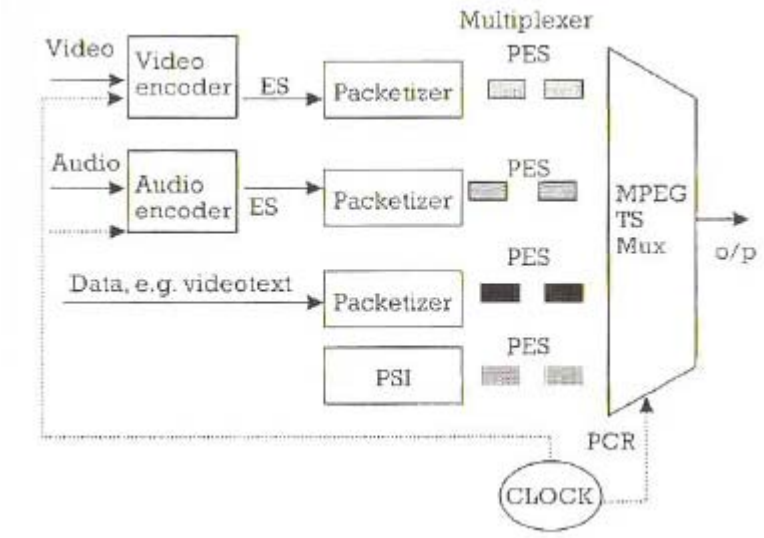
Το πρότυπο ορίζει τη ροή που προέρχεται από την κωδικοποίηση και την πολυπλεξία και όχι από τη βασική διαδικασία δημιουργίας της ροής. Το επίπεδο συστήματος MPEG-2 περιλαμβάνει ένα μηχανισμό που συνενώνει τον κωδικοποιημένο ήχο και βίντεο με άλλα σήματα δεδομένων σε ροές μεταφοράς. Επιτρέπει επίσης το συνδυασμό(με την τεχνική της πολυπλεξίας) διαφορετικού αριθμού προγραμμάτων σε μία κοινή ροή μεταφοράς MPEG-2.

Η πληροφορία αναφοράς ρολογιού χρησιμοποιείται για να επιτρέψει την ανάκτηση των ρολογιών βίντεο και ήχου. Έτσι εξασφαλίζεται ότι ο ήχος, το βίντεο

και τα δεδομένα διατηρούνται συγχρονισμένα ακόμα και όταν τα σήματα μετατρέπονται σε πακέτα.

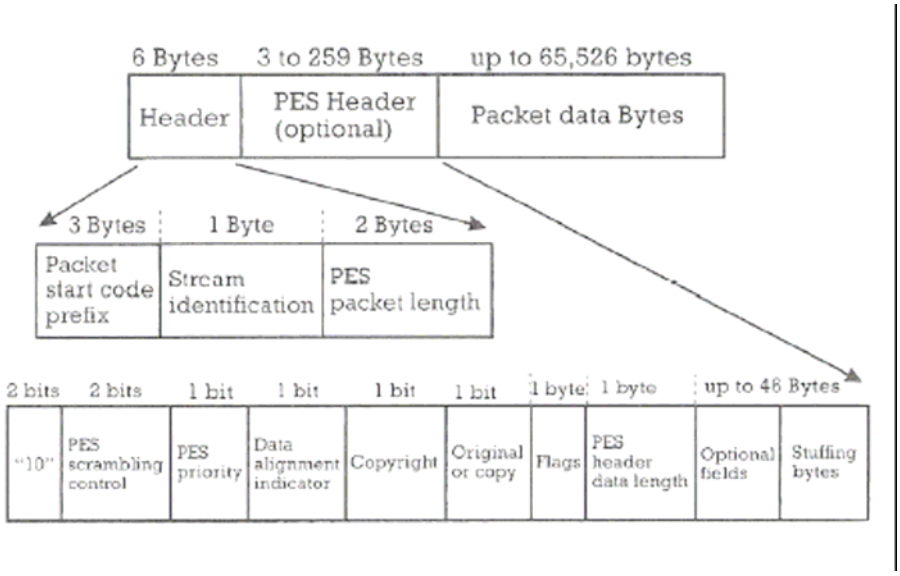
Το επίπεδο συστήματος επίσης παρέχει πληροφορίες σχετικά με το πραγματικό δίκτυο μετάδοσης(Network Information Table), σχετικά με τα προγράμματα που μεταφέρονται(Program Allocation Tables) και τους πίνακες(Conditional Access Tables) οι οποίοι δίνουν πληροφορίες σχετικά με τις κωδικοποιημένες υπηρεσίες. Με αυτές τις δυνατότητες επιτρέπεται σε έναν αποκωδικοποιητή να επιλέξει μια συγκεκριμένη τηλεοπτική υπηρεσία από αυτές που παρέχεται στους τελικούς χρήστες.

Στο παρακάτω σχήμα(σχήμα 5) παρουσιάζεται ένας πολυπλέκτης MPEG-2 TS μίας υπηρεσίες όπου τα στοιχεία του ήχου, του βίντεο και τα δεδομένα κωδικοποιούνται και δημιουργούνται σε ξεχωριστά πακέτα. Τα πακέτα που προκύπτουν είναι πολυπλεγμένα μεταξύ τους και σχηματίζουν μια τελική ροή μεταφοράς MPEG-2.



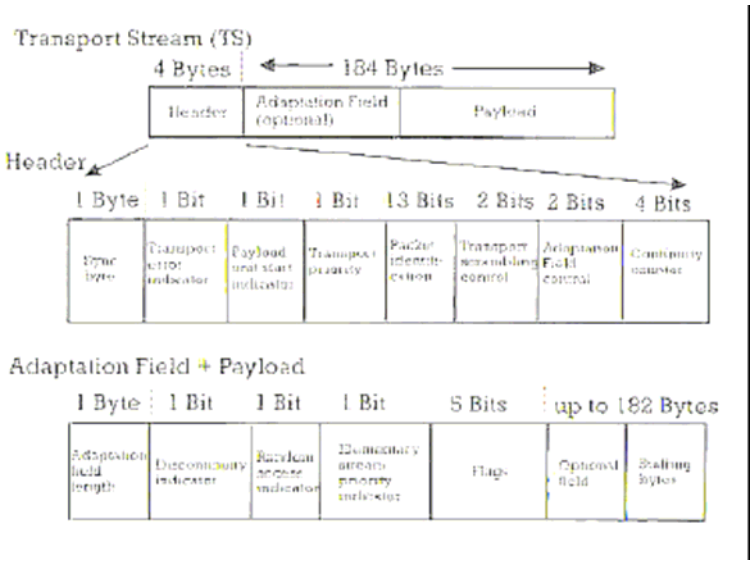
Σχ. 5 Πολυπλέκτης MPEG-2 TS

Οι βασικές ροές(ES) δημιουργούνται από τον κωδικοποιητή και είναι ξεχωριστές για τα σήματα του βίντεο, του ήχου και των δεδομένων. Οι βασικές ροές πακέτων(PES) αποτελούνται από πακέτα μεταβλητού μήκους και περιέχουν χαρακτηριστικά και πληροφορίες κεφαλίδας. Οι PES είναι ξεχωριστές για το βίντεο, τον ήχο και τα δεδομένα και η δομή τους παρουσιάζεται στο σχήμα 6 :



Σχ. 6 Δομή ροής MPEG-2 TS

Η ροή προγράμματος αποτελεί τη σύνθεση των βασικών ροών πακέτων(PES) ήχου, βίντεο και δεδομένων χρησιμοποιώντας το ίδιο ρολόι αναφοράς. Προορίζεται για εγγραφή και μετάδοση σε ασφαλή κανάλια στα οποία δεν δημιουργούνται λάθη. Επίσης η ροή προγράμματος αποτελείται από πακέτα μεταβλητού μήκους. Το πρότυπο που χρησιμοποιείται για τα πακέτα αυτά φαίνεται στο σχήμα 7 :



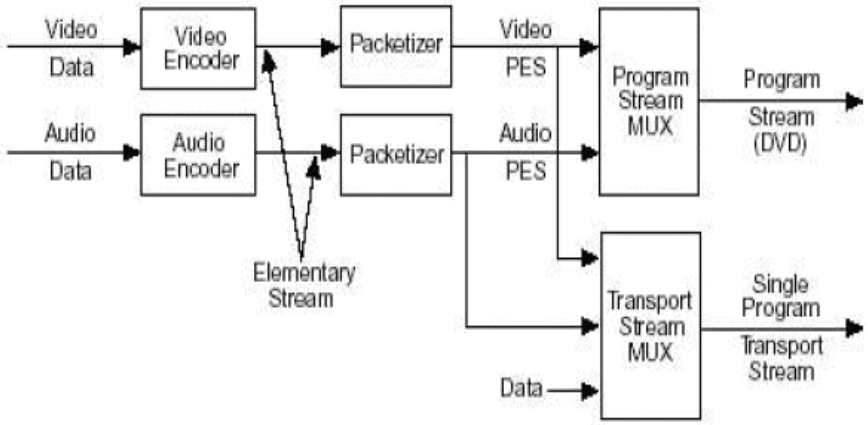
Σχ. 7 Δομή πακέτου MPEG-2 TS

Η ροή μεταφοράς μπορεί να περιέχει μία ή και περισσότερες τηλεοπτικές υπηρεσίες πολυπλεγμένες μαζί με ανεξάρτητα ρολόγια. Περιέχουν επίσης πακέτα σταθερού μήκους τα οποία προορίζονται για μετάδοση σε κανάλια που επηρεάζονται από λάθη. Στο πακέτο MPEG-2 TS περιέχεται μία κεφαλίδα που έχει μέγεθος 4 bytes στην αρχή του πακέτου μέσα στην οποία περιέχονται τα απαραίτητα δεδομένα (PID) από τον αποκωδικοιητή για την αναγνώριση των προγραμμάτων μέσα σε μια συνεχή ροή μεταφοράς. Με την εισαγωγή κενών πακέτων μέσα στη ροή μεταφοράς

εξασφαλίζεται ο σταθερός ρυθμός μεταφοράς ακόμα και αν δημιουργείται από ροές προγράμματος μεταβλητού ρυθμού.

2.2.1.4. Πολυπλεξία ψηφιακών τηλεοπτικών προγραμμάτων

Κατά την πολυπλεξία, ένας απεριόριστος αριθμός ροών μεταφοράς βίντεο, ήχου και δεδομένων πρέπει να συγχρονιστούν μεταξύ τους(π.χ. οι υπότιτλοι σε ξένες γλώσσες). Κάθε ροή μεταφοράς τεμαχίζεται σε πακέτα με χρονικές σφραγίδες. Με την πολυπλεξία πετυχαίνουμε με μία κοινή ροή μεταφοράς να μεταφέρουμε πολλές διαφορετικές τηλεοπτικές υπηρεσίες στους τελικούς χρήστες. Στο σχήμα 8 παρουσιάζεται ένα παράδειγμα πολυπλεξίας δύο ροών μεταφοράς:



Σχ. 8 Πολυπλεξία δύο ροών

Η πολυπλεξία πακέτων πρέπει να πραγματοποιείται με συγκεκριμένους κανόνες ώστε να επιβεβαιώνεται ότι μια ροή μεταφοράς επιτρέπει στον ήχο και στο βίντεο να είναι απόλυτα συγχρονισμένα. Τα πακέτα βίντεο, ήχου και δεδομένων πολυπλέκονται στην τελική ροή μεταφοράς σε σχέση με το στιγμιαίο ρυθμό μετάδοσης του κάθε ένα. Αυτή η μέθοδος διατηρεί το συγχρονισμό μεταξύ όλων των πηγών που πολυπλέκονται. Αυτό επιτυγχάνεται προγρμματίζοντας τη μεταφορά των πακέτων χρησιμοποιώντας πολυπλέκτες ροής οι οποίοι επιβλέπουν τα επίπεδα του καταχωρητή(buffer) ενός υποτιθέμενου αποκωδικοποιητή που ονομάζεται αποκωδικοποιητής συστήματος ροής μεταφοράς(TS-STD).

2.3. Πρότυπα ψηφιακής τηλεόρασης

Η ιδέα για την ίδρυση της οικογένειας προτύπων DVB(Digital Video Broadcasting) ολοκληρώνεται το 1993. Οι οικογένεια DVB αναπτύχθηκε για να προσφέρει υπηρεσίες ψηφιακής τηλεόρασης εκμεταλλευόμενη ένα ευρύ φάσμα μέσων διανομής, δορυφορικών, καλωδιακών και επίγειων. Τα πρότυπα DVB έχουν υιοθετήσει τα πρότυπα MPEG-2 για τη συμπίεση του ήχου και της κινούμενης εικόνας καθώς και για την πολυπλεξία. Με τη χρήση των πακέτων MPEG-2 TS ένα σύστημα DVB μπορεί να μεταφέρει οτιδήποτε μπορεί να ψηφιοποιηθεί, από τηλεόραση υψηλής ευκρίνειας(HDTV), πολλαπλά κανάλια PAL/SECAM/NTSC, μέχρι και υπηρεσίες πολυμέσων και δεδομένων με υψηλή ταχύτητα.

Τα πιο γνωστά πρότυπα DVB είναι: το DVB-S(Satelite) που υποστηρίζει δορυφορική μετάδοση, το DVB-C(Cable) το οποίο υποστηρίζει καλωδιακή μετάδοση, το DVB-T(Terrestrial) που αναφέρεται στην αμφίδρομη επίγεια μετάδοση και τέλος το DVB-H(Handhelds) το οποίο προσθέτει την έννοια του κινητού χρήστη στην επίγεια ψηφιακή μετάδοση.

Το δορυφορικό σύστημα DVB-S είναι το παλαιότερο και πιο διαδεδομένο πρότυπο DVB και έχει τύχει παγκόσμιας αποδοχής. Σχεδιάστηκε για να εκμεταλλεύεται πλήρως το εύρος ζώνης των δορυφορικών τηλεοπτικών αναμεταδοτών. Χρησιμοποιεί ρυθμό μεταφοράς 54Mbps με διαμόρφωση QPSK σε συνδυασμό με ένα σχήμα διπλής κωδικοποίησης και διεμπλοκής(coding/interleaving).

Το καλωδιακό σύστημα DVB-C έχει τεχνικά αρκετές ομοιότητες με το DVB-S. Η διαφορά είναι ότι χρησιμοποιεί την πιο αποδοτική αλλά παράλληλα και πιο ευαίσθητη σε παρεμβολές διαμόρφωση 64-QAM αντί για QPSK. Ένα καλωδιακό κανάλι 8MHz μπορεί να μεταφέρει 38,5Mbps.

Το επίγειο σύστημα DVB-T είναι το τελευταίο χρονικά πρότυπο της οικογένειας DVB. Επιτυγχάνει ψηφιακή μετάδοση υψηλών ταχυτήτων πάνω από το επίγειο κανάλι και χρησιμοποιεί διαμόρφωση πολλαπλών φερόντων στο σχήμα της πολυπλεξίας με ορθογωνική διαίρεση συχνότητας(Orthogonal Frequency Division Multiplexing-OFDM) Το σχήμα OFDM του DVB-T χρησιμοποιεί ένα μεγάλο αριθμό φερόντων(6817 ή 1704 για μετάδοση 8K και 2K αντίστοιχα), κάθε ένα από τα οποία διαμορφώνεται κατά QPSK, 16-QAM ή 64-QAM. Έτσι, η πληροφορία κατανέμεται ομοιόμορφα στο φάσμα και σε συνδυασμό με κωδικοποίηση και διεμπλοκή δύο στρωμάτων, το σήμα αποκτά μεγαλύτερη ευρωστία ακόμα και σε περιβάλλοντα με ισχυρές διαλείψεις και φαινόμενα πολυδιαδρομικής μετάδοσης(multipath).

Το σύστημα DVB-H είναι μια συνέχεια του DVB-T με τη διαφορά ότι αναφέρεται σε κινητούς χρήστες. Το πρότυπο αυτό λαμβάνει υπόψη του τις ιδιότητες που έχουν οι συσκευές κινητής πρόσβασης, όπως το μέγεθος της συσκευής και τη χωρητικότητα μνήμης, ανάγκη εύκολης πρόσβασης στο δίκτυο και εξοικονόμηση ενέργειας.

2.4. Αρχές λειτουργίας επίγειας ψηφιακής τηλεόρασης

Το πρότυπο DVB-T αναπτύχθηκε, όπως και τα άλλα πρότυπα της οικογένειας DVB, από το DVB forum και εγκρίθηκε από τον οργανισμό ETSI ως ευρωπαϊκό πρότυπο το 1997. Το πρώτο επίγειο ψηφιακό πρόγραμμα ξεκίνησε στην Αγγλία ένα χρόνο αργότερα. Σήμερα, συστήματα DVB-T βρίσκονται σε χρήση στην Ευρωπαϊκή Ένωση, στη Ρωσία, στην Ανατολική Ευρώπη, στην Ινδία, στην Αυστραλία και στην Σιγκαπούρη. Το DVB-T προπορεύεται του συστήματος NTSC που αναπτύχθηκε στις Ηνωμένες Πολιτείες ως προς κάποια χαρακτηριστικά, όπως στην κινητικότητα του χρήστη, στην αντοχή σε multipath μεταδόσεις και στην δημιουργία SNF ενώ το πρότυπο ISDB-T που αναπτύχθηκε στην Ιαπωνία είναι βασισμένο στο πρότυπο DVB-T.

Το DVB-T περιορίζεται αποκλειστικά στην περιγραφή των λειτουργιών του διαμορφωτή. Αυτός δέχεται από τον πολυπλέκτη την ροή μεταφοράς MPEG-2 με τις πολυπλεγμένες υπηρεσίες ήχου, βίντεο και δεδομένων υπό μορφή σήματος βασικής ζώνης και παράγει το προς μετάδοση σήμα RF. Το RF σήμα έχει εύρος ζώνης 8MHz και τοποθετείται σε ένα από τα κανάλια 21-69 της μπάντας των UHF, όπως ένα αναλογικό τηλεοπτικό σήμα.

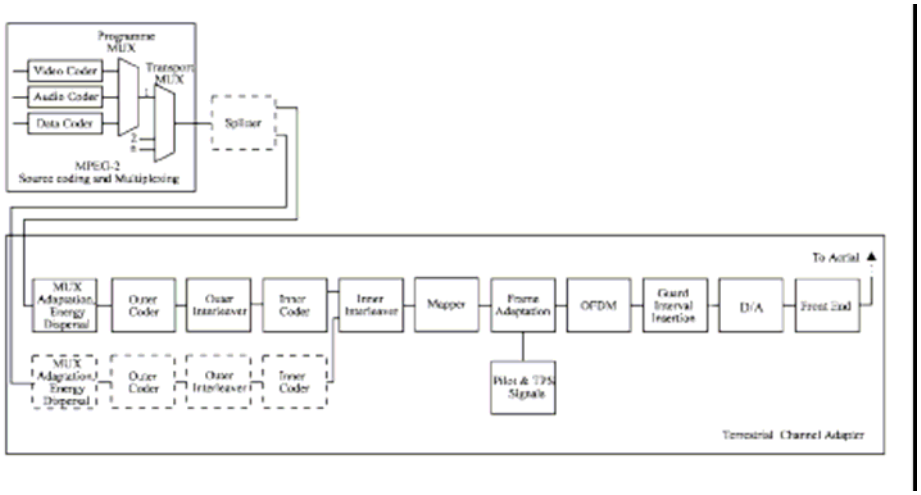
Ο διαμορφωτής DVB-T χρησιμοποιεί σχήμα OFDM για την αντιμετώπιση κατά τη μετάδοση διαλείψεων ως προς τη συχνότητα. Αυτό διευκολύνει πολύ τη λήψη δεδομένων ευρείας ζώνης και από κινητούς χρήστες. Στις τεχνολογίες εκπομπής(broadcasting) το OFDM παρέχει τη δυνατότητα κατασκευής δικτύων με διεσπαρμένους πομπούς που εκπέμπουν συγχρονισμένα τα ίδια δεδομένα και στην ίδια συχνότητα χωρίς η παρεμβολή των σημάτων να επηρεάζουν τον δέκτη. Αυτά τα δίκτυα ονομάζονται Single Frequency Networks(SFN) και έχουν την δυνατότητα να αυξήσουν τη χωρητικότητα του δικτύου μέχρι και 45 φορές καθώς εξαλείφεται η ανάγκη να υπάρχουν ελεύθερα κανάλια σε ορισμένες περιοχές για να αποφεύγονται παρεμβολές με γειτονικούς πομπούς.

Το πρότυπο DVB-T συνδυάζει το OFDM με σύνθετες τεχνικές ισοστάθμισης και κωδικοποίησης, εισάγοντας την τεχνολογία του κωδικοποιημένου OFDM(Coded-OFDM-COFDM). Με το συνδυασμό κωδικοποίησης και διεμπλοκής δύο επιπέδων, η διαδικασία διαμόρφωσης καθιστά το σήμα αρκετά ανθεκτικό σε πολυδιαδρομική μετάδοση και παρεμβολές. Το διάγραμμα που παρουσιάζεται στο σχήμα 9, δείχνει τις λειτουργίες της μετατροπής του σήματος βασικής ζώνης στο προς μετάδοση σήμα. Οι λειτουργίες αυτές που εφαρμόζονται στη ροή μεταφοράς κατά σειρά είναι οι εξής:

- Προσαρμογή MPEG-2 πακέτων και τυχαιοποίηση(randomization)
- Εξωτερική κωδικοποίηση(προστασία από λάθη με κώδικα Reed-Solomon)
- Εξωτερική συνελικτική διεμπλοκή (convolutional interleaving)
- Εσωτερική κωδικοποίηση με διάτρητο συνελικτικό κώδικα (punctured convolutional code)
- Εσωτερική διεμπλοκή (inner interleaving) στο χρόνο και στη συχνότητα

- Αντιστοίχηση και διαμόρφωση των φερόντων
- Πολυπλεξία κατά OFDM με αντίστροφο ταχύ μετασχηματισμό Fourier(IFFT) και διαμόρφωση του φέροντος IF
- Άνω μετατροπή(up-conversion) στην τελική RF συχνότητα

Η τελευταία λειτουργία δεν υποστηρίζεται από αρκετούς διαμορφωτές. Έτσι θα πρέπει να εισαχθεί μία πρόσθετη μονάδα για άνω μετατροπή.

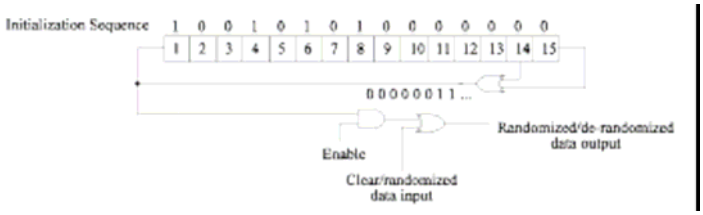


Σχ. 9 Διάγραμμα διαμορφωτή DVB-T

Στην ιεραρχική διαμόρφωση, το σήμα βασικής ζώνης προϋπάρχει διαιρεμένο σε δύο ροές μεταφοράς: μία υψηλής προτεραιότητας(high priority TS) και μία χαμηλής προτεραιότητας(low priority TS). Αυτά τα δύο σήματα διαμορφώνονται ταυτόχρονα σε ένα ιεραρχικό QAM σχήμα. Αυτό έχει ως αποτέλεσμα, ένας δέκτης με κακές συνθήκες λήψης λαμβάνει μόνο τα δεδομένα υψηλής προτεραιότητας σε αντίθεση με ένα με καλύτερες συνθήκες λήψης που λαμβάνει το σύνολο. Η λειτουργία ιεραρχικής διαμόρφωσης παρέχει σημαντική ευελιξία στο σύστημα ιδιαίτερα αν συνοδευτεί από κλιμακωτή κωδικοποίηση της κινούμενης εικόνας κατά MPEG-2(scalable MPEG-2 encoding).

Το σήμα βασικής που εισέρχεται στον διαμορφωτή έχει σταθερό ρυθμό μεταφοράς και είναι οργανωμένο σε πακέτα σταθερού μήκους 188 bytes. Κάθε πακέτο ξεκινά με το byte συγχρονισμού. Για να περιοριστεί το ενδεχόμενο να υπάρχουν μεγάλα διαστήματα χωρίς δυαδική μεταβολή(μακριές ακολουθίες από 0 ή 1) ακολουθείται η διαδικασία τυχαιοποίηση που φαίνεται στο σχήμα 10. Το πολυώνυμο για την γεννήτρια ψευδοτυχαίας ακολουθίας είναι :

$$1 + X^{14} + X^{15}$$



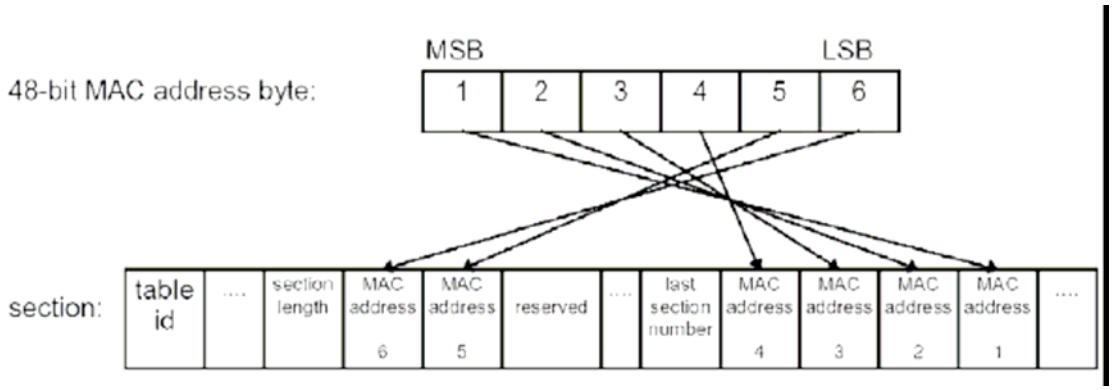
Σχ. 10 Τυχαιοποίηση δεδομένων

2.5. Μετάδοση δεδομένων IP-Ενθυλάκωση

Με την εξασφάλιση της αξιόπιστης μεταφοράς MPEG-2 πάνω από το κανάλι DVB-T, πρέπει επίσης να εξεταστεί και το πως τα πακέτα IP ενθυλακώνονται (encapsulate) πάνω στα πακέτα μεταφοράς. Τα τελευταία έχουν σταθερό μήκος 188 bytes , με τα 4 να είναι η επικεφαλίδα(header). Προκειμένου λοιπόν τα IP πακέτα αφενός να ενσωματωθούν στη ροή μεταφοράς και αφετέρου να διακρίνονται από τα πακέτα που μεταφέρουν τα προγράμματα ψηφιακής τηλεόρασης, πρέπει να υιοθετηθεί μια διαδικασία που να εκτελεί λειτουργίες αντιστοίχησης(mapping), προσαρμογής(adaptation) και κατακερματισμού(segmentation). Αυτές οι λειτουργίες έχουμε οριστεί από το πρότυπο ETSI EN 301 192. Το πρότυπο αυτό ορίζει τέσσερις τεχνικές ενθυλάκωσης :

- Διοχέτευση δεδομένων (data piping). Εδώ, τα δεδομένα των IP πακέτων ενσωματώνονται απευθείας ως φορτίο(payload) στα πακέτα μεταφοράς MPEG-2.
- Ροή δεδομένων(data streaming). Η μέθοδος αυτή είναι πιο κατάλληλη για υπηρεσίες streaming over DVB. Το data stream διαμορφώνεται σε μία συμβατή βασική ροή MPEG-2 (Elementary Stream) , η οποία στη συνέχεια οργανώνεται σε πακέτα κατά τη δομή PES (Packetized Elementary Stream). Τελικά, αυτά τα πακέτα κατακερματίζονται και διανέμονται στο φορτίο των MPEG-2 πακέτων μεταφοράς.
- Ενθυλάκωση πολλαπλών πρωτοκόλλων(Multiprotocol Encapsulation-MPE). Αυτή η μέθοδος αναπτύχθηκε για την μεταφορά πακέτων διάφορων πρωτοκόλλων(π.χ. TCP/IP) πάνω από το κανάλι DVB. Τα πακέτα πληροφορίας ενσωματώνονται σε data sections, όπως αυτά ορίζονται στο πρότυπο MPEG-2 DSM-CC. Τα data sections με τη σειρά τους είναι πλήρως συμβατά με τη δομή private_section που ορίζεται στο MPEG-2 Systems και ενσωματώνονται απευθείας στη ροή μεταφοράς. Από την πλευρά του χρήστη, τα πακέτα που προορίζονται για αυτόν διαχωρίζονται από τα υπόλοιπα δεδομένα με κριτήριο το αναγνωριστικό του πακέτου(PID), στη συνέχεια το πεδίο MAC(όπου αποθηκεύονται στα αντίστοιχα πεδία του section header, όπως δείχνει το σχήμα 11) και τέλος τη διεύθυνση IP προορισμού αν πρόκειται για πακέτα IP. Η τεχνική του Multiprotocol encaptulation είναι αυτή που χρησιμοποιείται στη μελέτη αυτή και η οποία αποτελεί την πιο διαδεδομένη τεχνική ενθυλάκωσης, παρέχοντας ακόμα και δυνατότητες κρυπτογράφησης. Επίσης η τεχνική MPE εισάγει λόγω της ενθυλάκωσης

πολλαπλών επιπέδων αρκετή πλεονάζουσα πληροφορία(overhead) για πλαισίωση και σηματοδοσία με αποτέλεσμα να μην αποτελεί τη βέλτιστη λύση για την περίπτωση του IP. Για αυτό το λόγο δημιουργήθηκε μια πιο ελαφριά έκδοση, με το όνομα ULE(Ultra-Light Encaptulation) που είναι ειδικά σχεδιασμένη για δεδομένα IP.



Σχ. 11 Ενσωμάτωση της MAC address στο section header

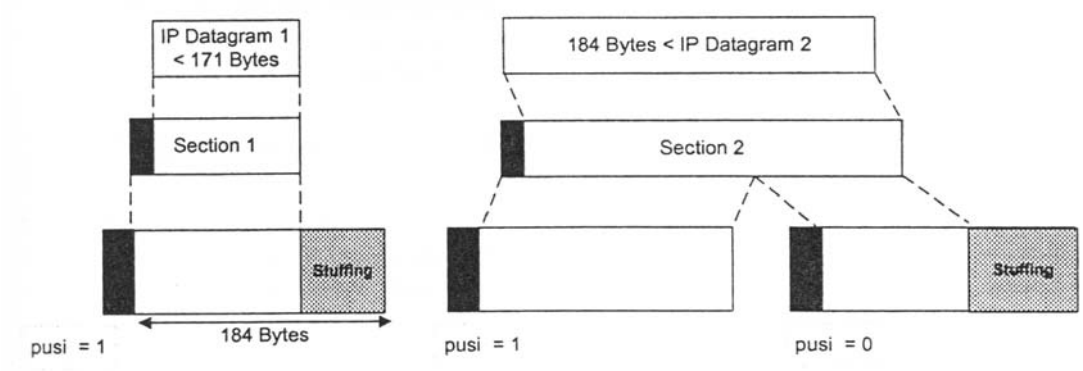
- Περιοδική εκπομπή δεδομένων(data carousel). Η τεχνική αυτή είναι κατάλληλη για μετάδοση δεδομένων χωρίς διαδραστικότητα(interaction). Τα δεδομένα που είναι για αποστολή, που αφορούν κυρίως μεγάλες ομάδες χρηστών παρά μεμονωμένους χρήστες, οργανώνονται σε ομάδες(groups) και εκπέμπονται κυκλικά ανά τακτά χρονικά διαστήματα, ώστε να είναι προσπελάσιμα από τον καθένα.

2.5.1. Ενθυλάκωση πολλαπλών πρωτοκόλλων(multi-protocol encapsulation)

Ο μηχανισμός ενθυλάκωσης πολλαπλών πρωτοκόλλων(MPE) δίνει τη δυνατότητα για τη μετάδοση πλαισίων από πολλαπλά δίκτυα δεδομένων σε κανάλια δικτύων ευρυ-εκπομπής. Η τεχνική αυτή βελτιστοποιήθηκε για τη χρήση πλαισίων IP.

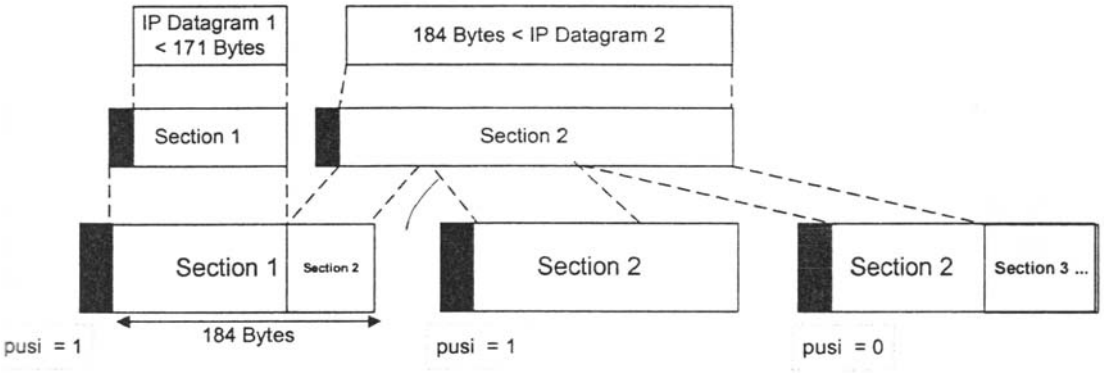
Μία δυνατότητα αξιοποίησης του μηχανισμού MPE είναι για τη μετάδοση πακέτων IP σε δίκτυα ψηφιακής τηλεόρασης. Αυτός ο μηχανισμός βασίζεται σε μία επιπλέον έκδοση της συγκεκριμένης τεχνικής ενθυλάκωσης. Αυτό έγινε για την επίλυση μερικών προβλημάτων τα οποία συνέβαιναν σε μεγάλα δίκτυα όπου υπήρχε η δυνατότητα παροχής υπηρεσιών IP. Η εισαγωγή πλαισίων IP πραγματοποιείται απευθείας στο payload των MPE πλαισίων(σχήμα 12) ενώ η τελική ροή έχει τη δυνατότητα μεταφοράς όλων των τύπων διαδικτυακών υπηρεσιών από υπηρεσίες πολυεκπομπής μέχρι υπηρεσίες διαδικτύου με υψηλούς ρυθμούς μετάδοσης. Όλες

αυτές οι υπηρεσίες συνήθως ενσωματώνονται σε μία κοινή πλατφόρμα η οποία διαχειρίζεται πολυπλεγμένες υπηρεσίες IP/DVB σε δίκτυα ψηφιακής τηλεόρασης.



Σχ. 12 Τεχνική ενθυλάκωσης πολλαπλών πρωτοκόλλων με πληροφορία stuffing

Στο παραπάνω σχήμα παρουσιάζεται η τεχνική MPE όπου stuffing πληροφορία προστίθεται στα πακέτα της ροής μεταφοράς MPEG-2. Στο σχήμα 13 φαίνεται μία βελτιωμένη έκδοση της τεχνικής αυτής σύμφωνα με την οποία κάθε πλαίσιο MP ενθυλακώνεται με τέτοιο τρόπο έτσι ώστε να μην χρειάζεται η προσθήκη stuffing πληροφορίας. Αυτή η μέθοδος θα αξιοποιηθεί για τη μετάδοση των διαδικτυακών υπηρεσιών στη μελέτη αυτή.

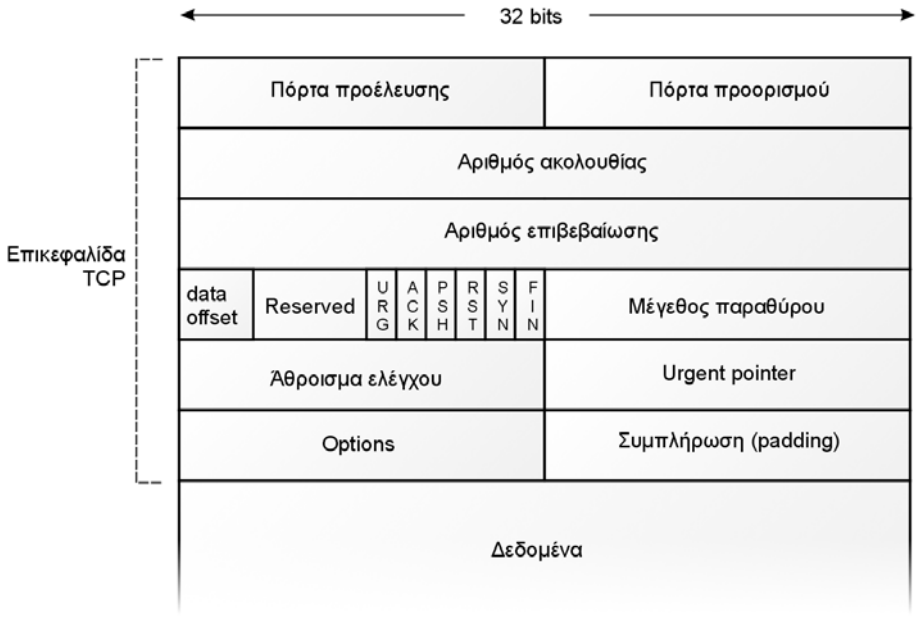


Σχ. 13 Τεχνική ενθυλάκωσης πολλαπλών πρωτοκόλλων (MPE)

2.5.2. Το πρωτόκολλο TCP και η σημασία του

Το πρωτόκολλο μεταφοράς TCP(Transmission Control Protocol) είναι το πιο δημοφιλές πρότυπο στις επικοινωνίες δεδομένων μέσω του Internet και όχι μόνο. Σχεδιάστηκε το 1981 για χρήση στο αμερικανικό DARPA για να εξασφαλίσει αξιόπιστη επικοινωνία μεταξύ δύο τερματικών. Από τότε, έχει γίνει αντικείμενο πολλών ερευνητικών προσπαθειών και έχει υποστεί αρκετές προσθήκες και βελτιώσεις. Πολλές από τις βελτιώσεις θα υιοθετηθούν σε αυτή την πτυχιακή ούτως ώστε 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(σχήμα 14).



Σχ. 14 Πεδία επικεφαλίδας πλαισίου 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 περιμένει. Τα τμήματα των οποίων η επιβεβαίωση καθυστερεί περισσότερο από ένα χρονικό διάστημα RTO(Retransmission TimeOut), τα επανεκπέμπουμε.

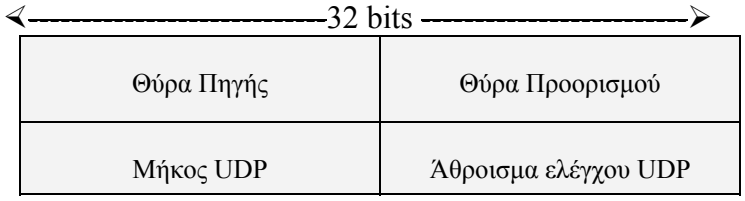
Για να μην έχουμε καθυστέρηση στην ανταλλαγή δεδομένων από τη φυσική καθυστέρηση του καναλιού, το TCP επιχειρεί να στείλει ένα συγκεκριμένο όγκο δεδομένων χωρίς να περιμένει την επιβεβαίωση για το πρώτο τμήμα. Ο όγκος αυτός

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) και αργότερα γραμμική(διαδικασία αποφυγής συμφόρησης). 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 μισό της προηγούμενης τιμής του, με αποτέλεσμα να μειώνεται αντίστοιχα και η ταχύτητα αποστολής.

2.5.3. Το πρωτόκολλο UDP και η σημασία του

Το Πρωτόκολλο Δεδομενογραφημάτων Χρήστη UDP(User Datagram Protocol) είναι ένα πρωτόκολλο μεταφοράς πληροφοριών χωρίς σύνδεση. Προσφέρει έναν τρόπο για να στέλνουν οι εφαρμογές ενθυλακωμένα ακατέργαστα δεδεδομενογραφήματα IP χωρίς να πρέπει να εγκαταστήσουν μια σύνδεση.

Ένα τεμάχιο UDP αποτελείται από μια επικεφαλίδα των 8 byte(64 bits), ακολουθούμενη από δεδομένα. Η επικεφαλίδα φαίνεται στο σχήμα 15.



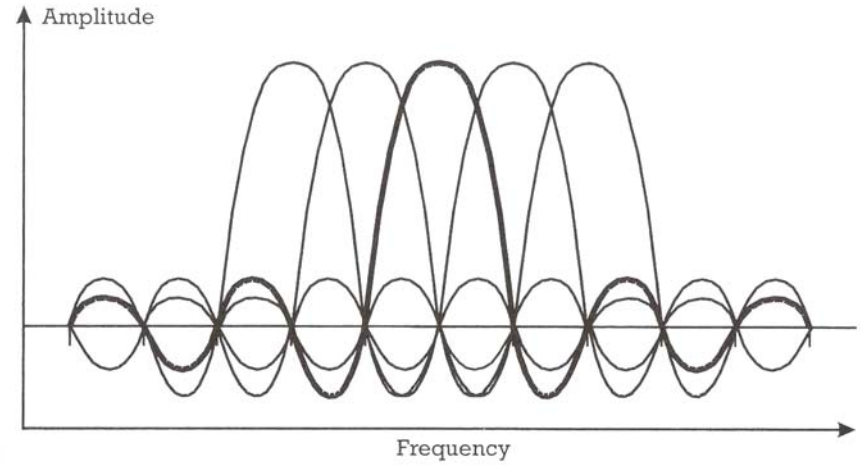
Σχ. 15 Η επικεφαλίδα UDP

Οι δύο θύρες εξυπηρετούν την ίδια λειτουργία όπως και στο TCP: την ανάγνωση των ακραίων σημείων στα μηχανήματα πηγής και προορισμού. Το μήκος UDP αφορά στην επικεφαλίδα 8 byte και στα δεδομένα. Το πεδίο άθροισμα ελέγχου περιλαμβάνει την ίδια ψευδο-επικεφαλίδα με το TCP, την επικεφαλίδα UDP και τα δεδομένα UDP, συμπληρωμένα ώστε να σχηματίζουν έναν άρτιο αριθμό. Το άθροισμα ελέγχου UDP είναι προαιρετικό και καταχωρείται ως 0 όταν δεν υπολογίζεται.

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 σε γρήγορες εφαρμογές και ερωταποκρίσεις, τύποθ πελάτη-εξυπηρετητή, όπου η άμεση παράδοση είναι σπουδαιότερη από τη σωστή παράδοση.

2.6. Κωδικοποιημένη Πολυπλεξία με Διαίρεση σε Ορθογώνιες Συχνότητες (COFDM)

Η τεχνική διαμόρφωσης COFDM(Coded Orthogonal Frequency Division Multiplexing) περιλαμβάνει μία μέθοδο για τη διαμόρφωση πολλαπλών φερόντων (σχήμα 16). Είναι αρκετά ανθεκτική στη λήψη πολυδιαδρομικών σημάτων και είναι χρήσιμη για κανάλια που παρουσιάζουν γραμμικές παραμορφώσεις. Η COFDM είναι η διαμόρφωση που επιλέχθηκε από την οικογένεια προτύπων DVB για την ανάπτυξη του Ευρωπαϊκού προτύπου επίγειας ψηφιακής τηλεόρασης (DVB-T).



Σχ. 16 Φέρουσες COFDM

Η τεχνική διαμόρφωσης COFDM χρησιμοποιεί χιλιάδες ξεχωριστές φέρουσες για τη μεταφορά του σήματος δεδομένων, μοιράζοντας τα δεδομένα σε κάθε φέρουσα. Τ ο σήμα δεδομένων έπειτα διαμορφώνει αυτές τις φέρουσες σύμφωνα με τα πρότυπα διαμόρφωσης QPSK και QAM. Οι φέρουσες είναι κοντά τοποθετημένες με τέτοιο τρόπο σε ένα κανάλι RF δίχως την ύπαρξη κενού εύρους ενώ κάθε ξεχωριστεί φέρουσα μπορεί να ληφθεί από το σύνολο τους. Σύμφωνα με τα χαρακτηριστικά μετάδοσης και τις ρυθμίσεις λειτουργίας για τη μετάδοση DVB-T είναι δυνατόν να επιτευχθεί ωφέλιμος ρυθμός μετάδοσης μεταξύ 4Mbps και 31Mbps. Αυτό επιτρέπει περισσότερη ευελιξία στο σχεδιασμό δικτύων ευρυ-εκπομπής.

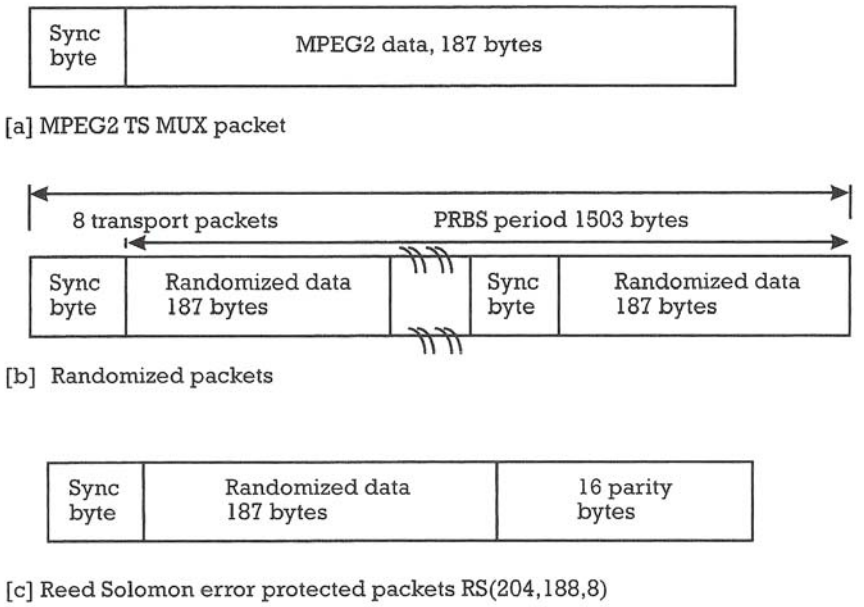
Όπως αναφέρθηκε παραπάνω, τα δεδομένα πληροφορίας που μεταφέρονται από το σύστημα μοιράζονται σε πολλαπλές φέρουσες και κάθε φέρουσα διαμορφώνεται με ένα σήμα πολύ χαμηλού ρυθμού. Έτσι η απώλεια μίας φέρουσας στο δέκτη έχει ελάχιστη επίπτωση στη ροή της συνολικής πληροφορίας ενώ τεχνικές διόρθωσης λαθών μπορούν να χρησιμοποιηθούν για τη μείωση στις απώλειες της πληροφορίας.

2.7. Κωδικοποίηση και Διεμπλοκή

Η εσωτερική κωδικοποίηση αποτελεί μία από τις παραμέτρους η οποία είναι σημαντική στη σχεδίαση δικτύων επίγειας ψηφιακής τηλεόρασης. Υπάρχουν 5 επιλογές εσωτερικής κωδικοποίησης,1/2, 2/3, 3/4, 5/6, 7/8, οι οποίες επιτρέπουν

κατά τη σχεδίαση του δικτύου τον καθορισμό του επιπέδου προστασίας και διόρθωσης λαθών για τις παρεχόμενες υπηρεσίες.

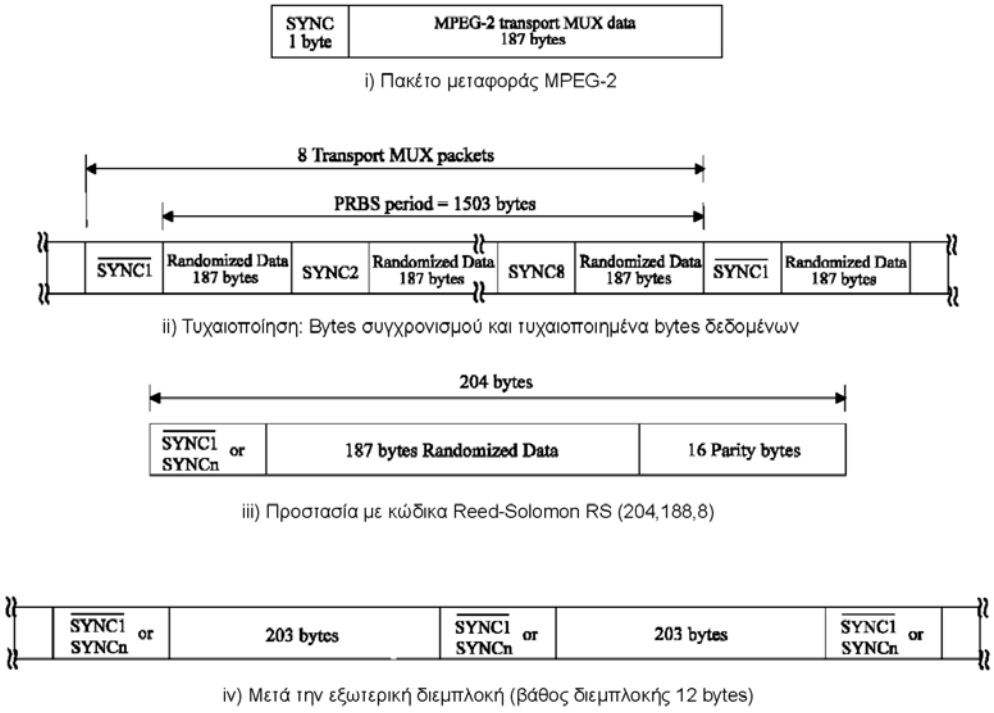
Ο διαμορφωτής COFDM δέχεται σήματα εισόδου στην μορφή MPEG-2 TS. Οι ροές μεταφοράς αποτελούνται από πακέτα σταθερού μεγέθους 188 bytes από τα οποία τα 187 bytes είναι για τα δεδομένα και το 1 byte περιέχει πληροφορίες συγχρονισμού. Η ροή πακέτων πρέπει πρώτα να τυχαιοποιηθεί πριν την μετάδοση έτσι ώστε να διαβεβαιωθεί ότι το σήμα COFDM περιέχει ίση πληροφορία σε όλο το εύρος ζώνης του. Αυτό επιτυγχάνεται διαβιβάζοντας τη ροή μεταφοράς μέσω ενός κωδικοποιητή ο οποίος χρησιμοποιεί μία γεννήτρια ψευδοτυχαίας δυαδικής ακολουθίας έτσι ώστε όλα τα δεδομένα του σήματος MPEG-2 είναι τυχαία κατανεμημένα στη ροή μεταφοράς. Μετά εφαρμόζεται ένας κώδικας Reed-Solomon σε κάθε τυχαιοποιημένο πακέτο MPEG-2. Αυτός ο κώδικας έχει μέγεθος 204 bytes, από τα οποία τα 16 είναι parity bytes και τα υπόλοιπα 188 bytes περιέχουν τα τυχαιοποιημένα δεδομένα και το byte συγχρονισμού(σχήμα 17). Ο κώδικας Reed-Solomon μπορεί να διορθώσει 8 τυχαία λανθασμένα bytes από τα 204.



Σχ. 17 Επεξεργασία πακέτου MPEG2-TS

2.7.1. Εξωτερική κωδικοποίηση και διεμπλοκή

Μετά το στάδιο της τυχαιοποίησης ακολουθεί η εξωτερική κωδικοποίηση(outer coding). Ένας συντομευμένος κώδικας Reed-Solomon που προκύπτει από τον αρχικό συστηματικό κώδικα RS εφαρμόζεται σε κάθε τυχαιοποιημένο πακέτο των 188 bytes. Ο Reed-Solomon αυξάνει το μήκος του πακέτου κατά 16 bytes, και επιτρέπει να διορθωθούν μέχρι 8 λανθασμένα bytes σε τυχαίες θέσεις μέσα στο προστατευμένο πακέτο. Μετά την κωδικοποίηση, τα προστατευμένα πακέτα υποβάλλονται σε μια συνελικτική διεμπλοκή(convolutional interleaving). Τα βήματα στη διαδικασία της προσαρμογής, τυχαιοποίησης, εξωτερικής κωδικοποίησης και διεμπλοκής παρουσιάζονται στο σχήμα 18.



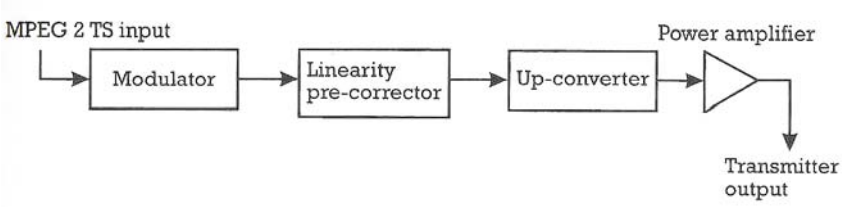
Σχ. 18 Μορφή δεδομένων μετά τις διαδικασίες: προσαρμογή, τυχαιοποίηση, εξωτερική κωδικοποίηση και διεμπλοκή

2.7.2. Εσωτερική κωδικοποίηση και διεμπλοκή

Η έξοδος του εσωτερικού κωδικοποιητή υφίσταται διεμπλοκή τόσο σε επίπεδο bits όσο και σε επίπεδο συμβόλων. Η ακριβής αντιστοιχία των bits εισόδου στα τελικά διαμορφωμένα σύμβολα εξαρτάται από τον τύπο της διαμόρφωσης που χρησιμοποιείται. Η διαδικασία της εσωτερικής κωδικοποίησης και διεμπλοκής είναι αρκετά πολύπλοκη.

2.8. Πομποί ψηφιακής τηλεόρασης

Οι πομποί επίγειας ψηφιακής τηλεόρασης χρησιμοποιούνται για τη μετάδοση ψηφιακών σημάτων σε χαμηλότερα επίπεδα ισχύος από ότι αυτά της υπάρχουσας αναλογικής τηλεόρασης. Αυτό οφείλεται στο ότι η απαιτούμενη ένταση του σήματος για τη λήψη του ψηφιακού τηλεοπτικού προγράμματος είναι μικρότερη από αυτή που απαιτείται για τη λήψη των σημάτων της αναλογικής τηλεόρασης. Η αρχιτεκτονική ενός πομπού ψηφιακής τηλεόρασης φαίνεται στο σχήμα 19 :



Σχ. 19 Λειτουργικό διάγραμμα πομπού επίγειας ψηφιακής τηλεόρασης

Αποτελείται από ένα διαμορφωτή(modulator) ο οποίος είναι συνδεδεμένος με ένα γραμμικό διορθωτή για την εξάλειψη ενδεχόμενων παραμορφώσεων στο σήμα της ψηφιακής τηλεόρασης κατά τη διάρκεια της ενίσχυσης. Μία τέτοια διάταξη θα χρησιμοποιήσουμε στην πτυχιακή αυτή έτσι ώστε να είναι δυνατή η εκπομπή όλων των υπηρεσιών ενημέρωσης μέσα από το δίκτυο επίγειας ψηφιακής τηλεόρασης.

3 ΠΑΡΟΥΣΙΑΣΗ ΔΙΚΤΥΟΥ ΠΑΡΟΧΗΣ ΑΜΦΙΔΡΟΜΩΝ ΔΙΑΔΡΑΣΤΙΚΩΝ ΥΠΗΡΕΣΙΩΝ

3.1 Εισαγωγή

Μια πλατφόρμα επίγειας ψηφιακής τηλεόρασης είναι δυνατόν να παρέχει εκτός απο ψηφιακά τηλεοπτικά προγράμματα (MPEG-2) και υπηρεσίες IP. Το ενδογενές αυτό χαρακτηριστικό σε συνδυασμό με την ύπαρξη καναλιών επιστροφής (Reverse Path), επιτρέπει την υλοποίηση ευρυζωνικών υποδομών για την παροχή αμφίδρομων διαδραστικών υπηρεσιών. Στο κεφάλαιο αυτό γίνεται η παρουσίαση ενός προτεινόμενου δικτύου το οποίο απαρτίζεται από δύο κανάλια επιστροφής που υλοποιούνται με τις τεχνολογίες WLAN και ISDN. Οι υπηρεσίες που παρέχονται από το σύστημα DVB-T της συγκεκριμένης πτυχιακής μπορούν να κατηγοριοποιηθούν σε δύο περιοχές:

- Διαδραστική Εκπομπή – Interactive Broadcasting. Περιλαμβάνει πραγματικά αμφίδρομες υπηρεσίες που συνδέονται ή είναι τελείως ανεξάρτητες με άλλες, broadcast υπηρεσίες. Απαιτείται κανάλι επιστροφής.
- Πρόσβαση στο Internet – Internet Access. Αυτή η περιοχή εστιάζεται στην παροχή πρόσβασης στο Internet μέσω μιας πλατφόρμας DVB.

3.2. Μελέτη δικτύου επίγειας ψηφιακής τηλεόρασης με κανάλια επιστροφής τις τεχνολογίες WLAN και ISDN

Όπως έχει τυποποιηθεί, ένα αμφίδρομο δίκτυο DVB αποτελείται από δύο διακριτά μέρη: το Κανάλι Εκπομπής (Broadcast Channel) και το Κανάλι Επιστροφής ή Αμφιδρομότητας (Interaction Channel). Το γενικό αυτό μοντέλο μπορεί να περιλάβει την περίπτωση του δικτύου παροχής αμφίδρομων διαδραστικών υπηρεσιών, οπότε οι κινήσεις TCP/IP ή UDP/IP που μεταφέρονται από το κανάλι εκπομπής ενθυλακώνονται σε ένα MPEG-2 Κανάλι Μεταφοράς με την τεχνική MPE (Multi Protocol Encapsulation). Η διαδικασία αυτή πραγματοποιείται από μια μονάδα που δρα ως “Πύλη IP/DVB” (IP-to-DVB Gateway). Η έξοδος της Πύλης σε μορφή σήματος βασικής ζώνης οδηγείται στον διαμορφωτή COFDM που παράγει και το προς εκπομπή RF σήμα σύμφωνα με την προδιαγραφή ETS 300 744.

Τα εκπεμπόμενα δεδομένα λαμβάνονται στη μεριά του ενδιάμεσου κόμβου διανομής (CMN) και προωθούνται στον τελικό χρήστη. Εκείνος με τη σειρά του στέλνει την δικτυακή του κίνηση πίσω, προς τον CMN. Ο ενδιάμεσος κόμβος διανομής αποστέλλει δεδομένα μέσω του καναλιού επιστροφής.

Η υλοποίηση του καναλιού επιστροφής μέσω του δημόσιου τηλεφωνικού δικτύου μεταγωγής (Integrated Services Digital Network– ISDN) εξυπηρετεί κατά κανόνα τους οικιακούς χρήστες, διότι παρέχονται προσιτές λύσεις επικοινωνίας.

Ως Μονάδα Διεπαφής χρησιμοποιείται ένα απλό ISDN modem και τα δεδομένα που προορίζονται για τον παροχέα διοχετεύονται μέσω του νοητού κυκλώματος στον Προσαρμογέα Δικτύου, που στην περίπτωση αυτή είναι ένας dial-up server.

Το δίκτυο του παροχέα είναι επίσης εξοπλισμένο με μία διεπαφή για το δίκτυο επιστροφής και η τοπολογία κορμού του μαζί με τις μονάδες μεταγωγής IP πακέτων

(routers, switches κτλ.) πρέπει να έχουν ρυθμιστεί ούτως ώστε να εξασφαλίζουν ασύμμετρη ροή πακέτων.

3.3 Πρωτόκολλο IPv6

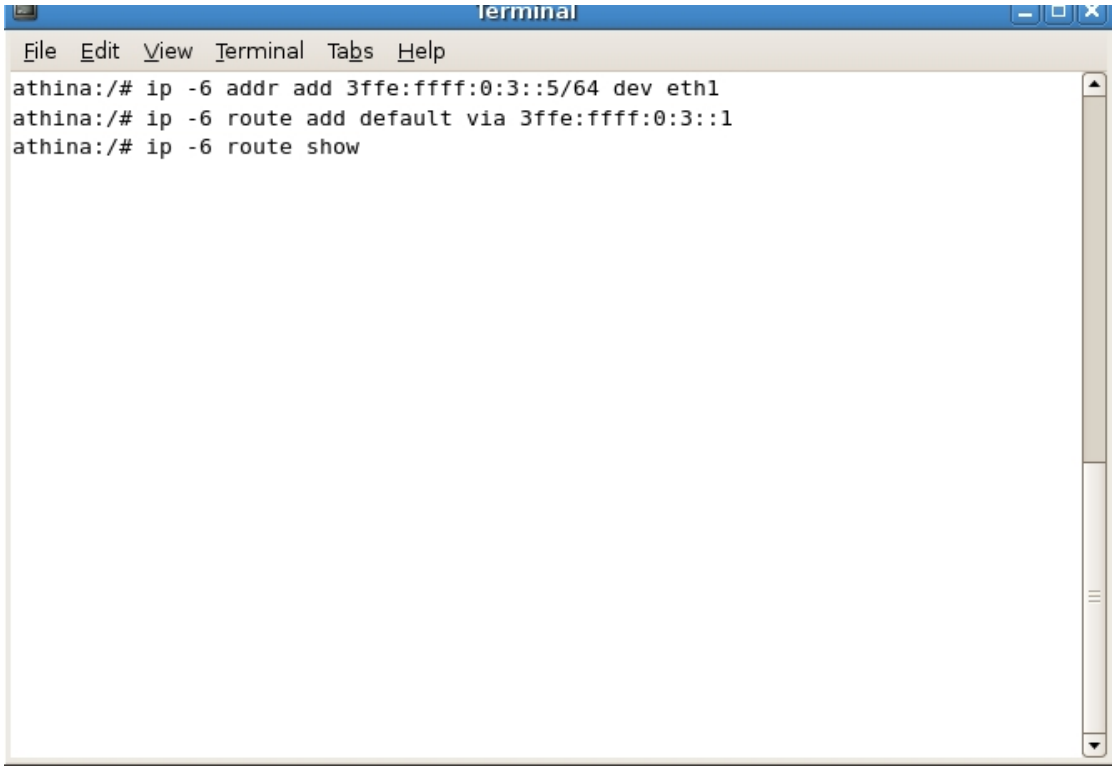
Η διευθυνσιοδότηση στην πτυχιακή αυτή είναι βασισμένη στο πρωτόκολλο IP έκδοση 6(IPv6). Πρόκειται για ένα βελτιωμένο πρωτόκολλο καθώς το πιο διαδεδομένο στις μέρες μας πρωτόκολλο IP έκδοση 4(IPv4) αδυνατεί να καλύψει τις ανάγκες της σύγχρονης διαδικτύωσης. Ένα βασικό πρόβλημα που υπάρχει με το IPv4 είναι η έλλειψη διευθύνσεων. Έτσι έρχεται το IPv6 για να λύσει το πρόβλημα. Επίσης, το IPv6 επιτυγχάνει να "δέσει" το περιβάλλον του επερχόμενου δικτύου όπου εκτός από τους συμβατικούς υπολογιστές θα αποτελείται από μυριάδες άλλες συσκευές όπως προσωπικοί επεξεργαστές δεδομένων μεγέθους παλάμης (palmtop personal data assistants-PDA), υβριδικά κινητά τηλέφωνα με υπολογιστικές δυνατότητες, έξυπνα κουτιά με ενσωματωμένους Web browsers καθώς και από φωτοτυπικά μηχανήματα ενός γραφείου έως και συσκευές που χρησιμοποιούνται στην κουζίνα ενός σπιτιού.

Το IPv6 χρησιμοποιεί ένα σχήμα διευθυνσιοδότησης μεγέθους 128 bit. Το μέγεθος των διευθύνσεων που παράγεται είναι πολύ μεγάλο. Το IPv6 κατανέμει τις διευθύνσεις με ιεραρχικό τρόπο αποφεύγοντας έτσι τα προβλήματα του IPv4 όπου και παραγόταν υπέρογκη πληροφορία για τη δρομολόγηση πάνω στα συστήματα και διευθύνσεις έμεναν αχρησιμοποίητες.

Το IPv4 δεν έχει χαρακτηριστικά που θα μπορούσαν να χρησιμοποιηθούν στην ασφάλεια των δικτύων και έτσι η προσπάθεια είχε κατευθυνθεί στη χρήση μεθόδων που βασίζονται στο network επίπεδο. Το IPv6 όμως παρέχει εγγενής δυνατότητες για παροχή ασφάλειας οι οποίες βασίζονται στις προσαρμοστικές επεκτάσεις της κεφαλής του IPv6 πακέτου. Η επέκταση της πιστοποίησης (authentication header extension) εξασφαλίζει ότι όντως το πακέτο έρχεται από τον host που δείχνει η διεύθυνση της πηγής. Αυτή η πιστοποίηση είναι σημαντική στη προστασία έναντι των εισβολέων, οι οποίοι ρυθμίζουν ένα host να παράγει πακέτα με πλαστή διεύθυνση πηγής. Αυτή η μεταμφίεση μπορεί να ξεγελάσει (IP spoofing) ένα εξυπηρέτη και να υπάρξει παράτυπη πρόσβαση σε πολύτιμα δεδομένα ή σε κρίσιμες δικτυακές λειτουργίες. Σύμφωνα με στατιστικές έρευνες αυτού του είδους το "ξεγέλασμα" (IP spoofing) ενός εξυπηρέτη είναι από τους πιο συνηθισμένους τρόπους εισβολής και δεν υπάρχει εγγενής τρόπος από το IPv4 να ελέγξει εάν το πακέτο έρχεται από εκεί που το ίδιο αναφέρει. Η σημερινή αντιμετώπιση αυτού του προβλήματος γίνεται με τους "Firewalls" η χρήση των οποίων παρουσιάζει μία σειρά από προβλήματα όπως μείωση στην απόδοση, περιοριστική δικτυακή πολιτική και περιορισμένη διασύνδεση με το Διαδίκτυο. [21]

3.3.1 Διευθυνσιοδότηση IPv6

Όπως αναφέρθηκε στην παραπάνω παράγραφο στην πτυχιακή θα χρησιμοποιηθούν IPv6 διευθύνσεις. Η εντολή για να βάλω σε ένα interface μια IPv6 διεύθυνση είναι λίγο διαφορετική από ότι είναι στο πρωτόκολλο IPv4. Στο σχήμα 20 παρουσιάζεται ενδεικτικά οι εντολές οι οποίες χρησιμοποιούνται για να δώσουμε IP:



Σχ. 20 Εντολές καθορισμού διευθύνσεων IPv6

3.4. Υλοποίηση με ISDN Κανάλι Επιστροφής

.

Στο συγκεκριμένο σενάριο ο αποδέκτης «ζητάει» υπηρεσίες διαδικτύου (internet) από τον αποστολέα κι εκείνος του αποστέλλει δεδομένα TCP. Επιλέχθηκε στον Ενθυλακωτή πολυπλέκτη (AMPER) ένα κανάλι Διαδικτυακής κίνησης (8Mbps) χωρίς την προσθήκη και μετάδοση πολλαπλών ψηφιακών τηλεοπτικών προγραμμάτων.

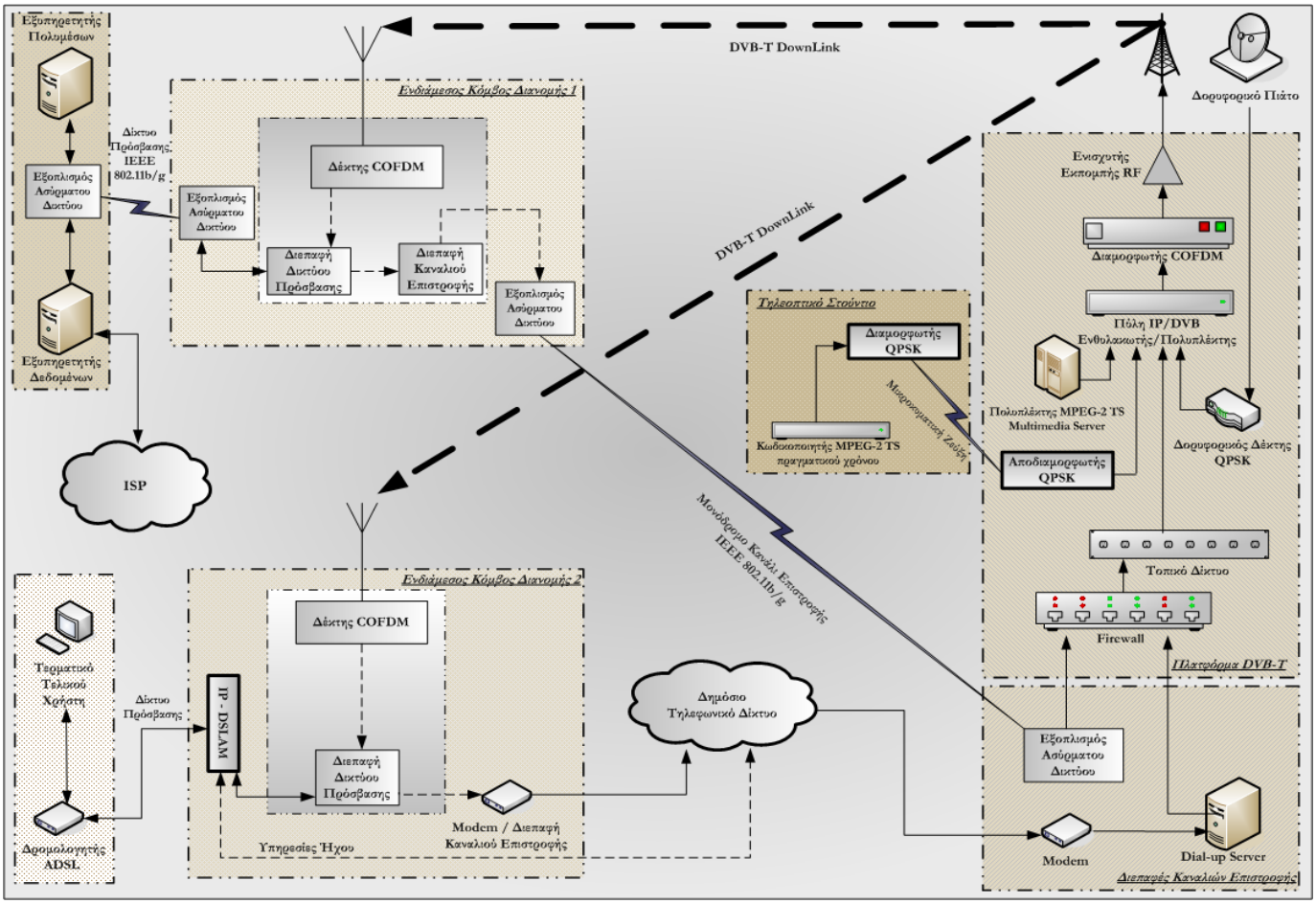
Συγκεκριμένα έχουμε έναν τελικό χρήστη (Client – IP Address: 3ffe:ffff:0:3::5) ο οποίος συνδέεται με ένα απλό δρομολογητή κι εκείνος με τη σειρά του με μια συσκευή DSLAM (Πρόκειται για εξοπλισμό ο οποίος συνδέει τους χρήστες των υπηρεσιών DSL με ένα δίκτυο μετάδοσης δεδομένων ATM. Βρίσκεται εγκατεστημένος στο κέντρο του τηλεπικοινωνιακού παρόχου (π.χ. OTE). Το DSLAM δίνει upstream 512kbps και downstream 8160kbps. Επομένως ο τελικός χρήστης του δικτύου χρησιμοποιεί ένα ADSL δίκτυο για να επικοινωνήσει αμφίδρομα με τον ενδιάμεσο κόμβο διανομής (CMN - IP Address: 3ffe:ffff:0:3::1). Η δικτυακή κίνηση στη συνέχεια στέλνεται από τον CMN, μέσω του καναλιού επιστροφής που είναι ένα ISDN δίκτυο, στον Dial-up Server κι από εκεί στην πλατφόρμα DVB-T.

Στην πλατφόρμα θα επιλεγεί η ανάλογη κωδικοποίηση και διαμόρφωση έτσι ώστε να σταλούν τα δεδομένα μέσω των UHF συχνοτήτων στον CMN (IP Address: 3ffe:ffff:0:1::1) που επικοινωνεί με τον Server (IP Address: 3ffe:ffff:0:1::5).

Ο CMN θα πάρει την κίνηση και μέσω ενός δρομολογητή θα την στείλει ασύρματα στον εξυπηρετητή του δικτύου. Οι δρομολογητές που εξασφαλίζουν ασύρματη σύνδεση μεταξύ του Server και του αντίστοιχου CMN, όπως και του CMN με την πλατφόρμα DVB-T είναι της τάξης των 54 Mbps. Ο Server επικοινωνεί αμφίδρομα με τον CMN, με αποτέλεσμα όταν απαντήσει, η δικτυακή του κίνηση να περάσει πάλι από τον δρομολογητή και να φτάσει ξανά πίσω στον CMN. Ακολουθεί μια ασύρματη σύνδεση που θα οδηγήσει την κίνηση προς την DVB-T πλατφόρμα για να κωδικοποιηθεί και διαμορφωθεί ούτος ώστε να σταλεί από την μπάντα των UHF συχνοτήτων (κανάλι 29) στον CMN του τελικού χρήστη. Εκείνος επικοινωνεί με το τερματικό του Client κι έτσι θα του προωθήσει τις απαντήσεις, του Server.

Παρατηρούμε ότι με βάση αυτή την αρχιτεκτονική του δικτύου, δίνεται η δυνατότητα στον πελάτη Client να επικοινωνήσει με τον εξυπηρετητή Server, και για την ακρίβεια να του ζητήσει TCP δεδομένα που του χρειάζονται. Ο Server, με τη σειρά του θα απαντήσει και θα πάρει πίσω τις επιβεβαιώσεις από τον Client. Έτσι δημιουργείται μια TCP δικτυακή κίνηση, ανάμεσα σ'έναν πελάτη κι έναν εξυπηρετητή με τη βοήθεια ενδιάμεσων κόμβων διανομής (CMNs) που έχουν τον ρόλο του «πυρήνα» στις κυψέλες του δικτύου, που έχουν δημιουργηθεί.

Η τοπολογία του συγκεκριμένου δικτύου δίνεται και στο σχήμα που ακολουθεί.



Σχ. 21 Αρχιτεκτονική δικτύου αμφίδρομης επίγειας ψηφιακής τηλεόρασης με κανάλι επιστροφής ISDN και WLAN

3.5. Χαρακτηριστικά ηλεκρονικών υπολογιστών που χρησιμοποιήθηκαν για την υλοποίηση του δικτύου

Στην εργαστηριακή υλοποίηση που χρησιμοποιήθηκε στην πτυχιακή εργασία, ο εξοπλισμός του τελικού χρήστη περιλαμβάνει ένα ηλεκτρονικό υπολογιστή Pentium 4 / 3.00 GHz με 512 MB RAM ο οποίος έχει υλοποιηθεί με βάση το λειτουργικό σύστημα των Linux (DEBIAN). Ο ενδιάμεσος κόμβος διανομής που επικοινωνεί με τον Client είναι ένας υπολογιστής Pentium 4/ 3GHz με 1GB RAM και κανει χρηση ιδιου λειτουργικου με αυτόν του Client. Ενώ ο δεύτερος ο ενδιάμεσος κόμβος διανομής είναι ένας LINUX (DEBIAN) υπολογιστής AMD 64bit / 2GHz με μνήμη RAM 512MB. Στο δίκτυο του παροχέα, ένας Linux server όμοιων χαρακτηριστικών με αυτόν του CMN που μόλις προαναφέραμε πραγματοποιεί την αποστολή των δεδομένων ο οποίος ομως κανει χρήση μιας διανομης των LINUX , UBUNTU.

3.6. Ρυθμίσεις της πλατφόρμας DVB-T

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 21.11Mbps. Επίσης ως φέροντα σήματα δηλώνονται 8K . Η συχνότητα του σήματος εκπομπής είναι στα 538MHz (κανάλι 29) με επίπεδο δύναμης (power level) 0dBm και το συγκεκριμένο κανάλι έχει εύρος 8MHz.



Σχ. 22 Ο διαμορφωτής COFDM που παρουσιάζει τα στοιχεία της διαμόρφωσης



Σχ. 23 Ο διαμορφωτής COFDM που παρέχει το DVB-T stream στο κανάλι 29 σε επίπεδο δύναμης 0dBm

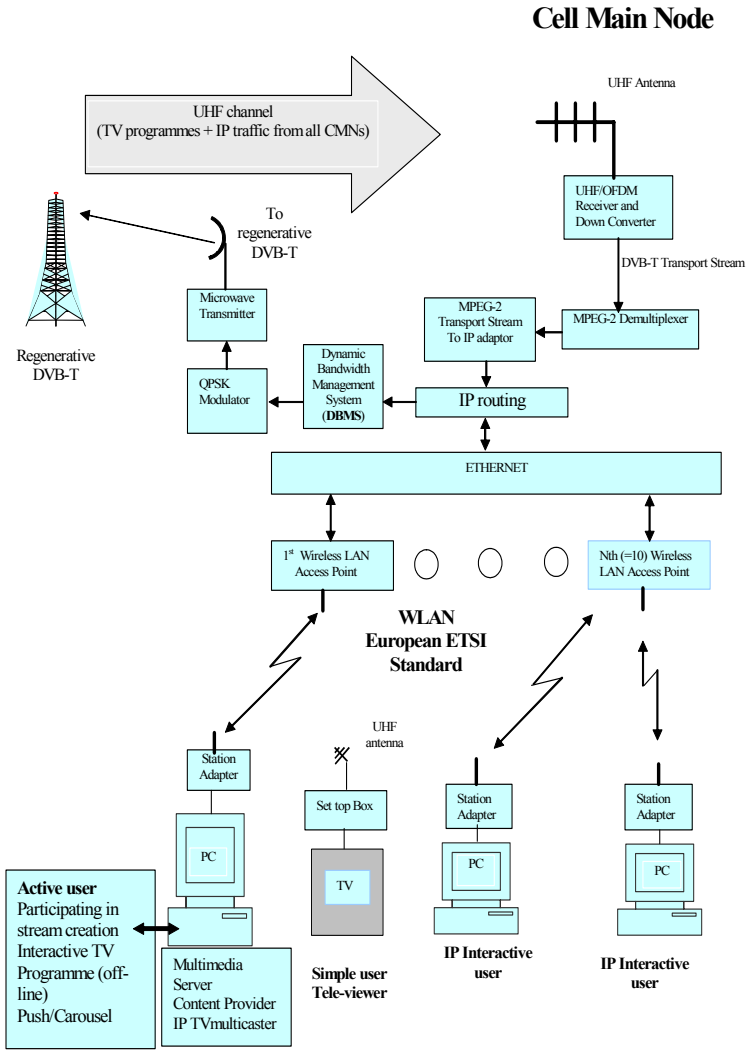
3.7 Ενδιάμεσος Κόμβος Διανομής (Cell Main Node)

Κάθε CMN καθιστά δυνατή την πρόσβαση από έναν αριθμό τελικών χρηστών (οι οποίοι βρίσκονται γεωγραφικά κοντά στο CMN) σε IP υπηρεσίες που παρέχονται από το δίκτυο. Η επικοινωνία ανάμεσα στους χρήστες και στον αντίστοιχο κόμβο επιτυγχάνεται με την χρήση ασύρματων τεχνολογιών πρόσβασης, όπως WLAN. Κάθε κεντρικός κόμβος συγκεντρώνει όλη την IP κίνηση, η οποία προέρχεται από τους χρήστες του, και την προωθεί στο κεντρικό σημείο ευρυεκπομπής (regenerative DVB-T) με τη χρήση point-to-point ζεύξεων (uplinks). Η IP κίνηση που προέρχεται από κάθε CMN, λαμβάνεται από το κεντρικό σημείο ευρυεκπομπής, όπου φιλτράρεται, αναγεννάται και πολυπλέκεται με ψηφιακά τηλεοπτικά προγράμματα (που προέρχονται από τους φορείς ευρυεκπομπής – TV Studio), σε μία κοινή ροή μεταφοράς (Transport Stream), η οποία αποτελεί το τελικό μπουκέτο DVB-T (DVB-T bouquet). Η αναγεννημένη κίνηση εκπέμπεται σε ένα κανάλι UHF (π.χ. κανάλι 29) σύμφωνα με το πρότυπο DVB-T. Κάθε χρήστης λαμβάνει τις κατάλληλες IP απαντήσεις (replies) μέσω του αντίστοιχου CMN, ενώ λαμβάνει τα ψηφιακά τηλεοπτικά προγράμματα απευθείας μέσω του καναλιού UHF.

Η κυψελωτή προσέγγιση η οποία έχει υιοθετηθεί, χρησιμοποιεί το DVB-T κανάλι μεταφοράς ως δικτυακό κορμό backbone, ο οποίος διασυνδέει όλους τους κόμβους διανομής (CMN) που βρίσκονται μέσα στην περιοχή ευρυεκπομπής (broadcasting area). Έτσι, δημιουργείται ένας μοναδικός κορμός Ethernet, που είναι διαθέσιμος στους τελικούς χρήστες (μέσω του κατάλληλου κόμβου – CMN). Η IP κίνηση του δικτύου Ethernet παρέχεται μέσω του συρμού DVB-T.

Η γενική αρχιτεκτονική ενός CMN που χρησιμοποιεί την WLAN τεχνολογία για την υλοποίηση του καναλιού επιστροφής παρουσιάζεται στο Σχήμα 24.Το συγκεκριμένο τμήμα του δικτύου υλοποιεί το δίκτυο πρόσβασης σύμφωνα με το πρότυπο IEEE 802.11b, κάνοντας χρήση τεχνικών ευρέου φάσματος (Spread

Spectrum). Ένα τέτοιο δίκτυο επιτρέπει την point-to-multipoint επικοινωνία ανάμεσα στο CMN και στους τελικούς χρήστες.



Σχ. 24 Αρχιτεκτονική CMN

Οι ενεργοί και διαδραστικοί χρήστες δημιουργούν κίνηση IP, η οποία μεταφέρεται μέσω του αντίστοιχου κόμβου στο κεντρικό σημείο ευρυεκπομπής (regenerative DVB-T) και από εκεί σε όλη την πόλη μέσω του καναλιού των UHF. [15]

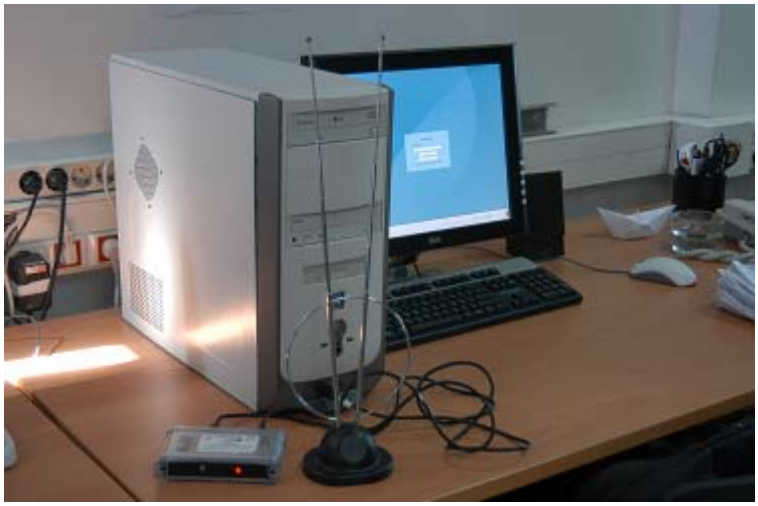
3.7.1 Απαιτήσεις Υλικού (Hardware Requirements)

Ο ενδιάμεσος κόμβος διανομής (CMN) που χρησιμοποιείται στο δίκτυο της επίγειας ψηφιακής τηλεόρασης έχει υλοποιηθεί με τη χρήση ενός ηλεκτρονικού υπολογιστή και των κατάλληλων μονάδων hardware και software. Ο Η/Υ απαιτεί ως ελάχιστα τεχνικά χαρακτηριστικά έναν επεξεργαστή Pentium 3, ένα σκληρό δίσκο χωρητικότητας 10GB και μία μνήμη RAM 256MB. Ο κόμβος για την σωστή

λειτουργία του απαιτεί επίσης τρία δικτυακά interfaces τα οποία περιγράφονται παρακάτω.

Η πλατφόρμα DVB-T εκπέμπει μία κοινή ροή μεταφοράς πακέτων MPEG-2 TS μέσα στην οποία έχει ενθυλακωθεί Διαδικτυακή κίνηση πακέτων IP στο κανάλι 29 της μπάντας συχνοτήτων UHF. Η συγκεκριμένη ροή μεταφοράς λαμβάνεται από την κάρτα επίγειας ψηφιακής τηλεόρασης που έχει τοποθετηθεί στον ενδιάμεσο κόμβο διανομής και έτσι με αυτόν το τρόπο υπάρχει η δυνατότητα λήψης της κίνησης πακέτων IP από το interface του DVB-T.

Η κάρτα αυτή διαδοχικά προωθεί τα πακέτα προς την κάρτα Ethernet του δικτύου πρόσβασης των τελικών χρηστών, όπου απορρίπτονται όσα πακέτα δεν αντιστοιχούν στο συγκεκριμένο υποδίκτυο. Η βασική λειτουργία της κάρτας δικτύου πρόσβασης είναι να προωθεί την κίνηση προς τους τελικούς χρήστες μέσω ενός δικτύου Ethernet. Με αυτόν τον τρόπο οι τελικοί χρήστες έχουν τη δυνατότητα να λάβουν τα πακέτα και να στείλουν τις επιβεβαιώσεις (ACKs) της συγκεκριμένης κίνησης μέσω του δικτύου πρόσβασης και του δικτύου επιστροφής. Η κίνηση των πακέτων αυτών προωθείται μέσω της κάρτας του δικτύου πρόσβασης στην κάρτα του δικτύου επιστροφής που διασυνδέει τον ενδιάμεσο κόμβο διανομής με το κεντρικό σημείο εκπομπής.



Σχ. 25 Ενδιάμεσος Κόμβος Διανομής (CMN), κάρτα επίγειας ψηφιακής τηλεόρασης, κεραία λήψης

Όπως έχουμε προαναφέρει για την υλοποίηση του δικτύου μας κάναμε χρήση δυο ενδιαμέσων κόμβων διανομής ,έναν με δύκτιο επιστροφής ISDN και έναν άλλο με ασύρματο 802.11b/g. Ο κάθε CMN αναλαμβάνει να προωθήσει τα δεδομένα στο αντίστοιχο υποδύκτιο. Και οι δυο CMN χρησιμοποιούν σαν interface της λήψης από μία κάρτα επίγειας ψηφιακής τηλεόρασης DVB-T. Το μοντέλο της κάρτας αυτής είναι η WinTV-NOVA-T-USB-2 (Σχήμα 26) που υποστηρίζεται από τον πυρήνα του λειτουργικού συστήματος Linux που εγκαταστάθηκε στους Η/Υ. Η κάρτα αυτή ονομάστηκε dvb_0 και χρησιμοποιείται για τη λήψη της κίνησης που εκπέμπεται από το κεντρικό σημείο εκπομπής και περιέχει τις IP απαντήσεις για τους τελικούς χρήστες. Απαραίτητη για τη λήψη είναι η χρήση μίας κεραίας UHF εσωτερικού χώρου.



Σχ. 26 Κάρτα επίγειας ψηφιακής τηλεόρασης WinTV-NOVA-T-USB-2

Άλλο κοινό χαρακτηριστικό των δύο CMN είναι το interface του δικτύου πρόσβασης το οποίο υλοποιείται με τη χρήση καρτών δικτύου, eth_1, τύπου Intel(R) Pro/100 s Desktop Adapter (Σχήμα 27) οι οποίες λαμβάνουν την προωθημένη Διαδικτυακή κίνηση από την κάρτα επίγειας ψηφιακής τηλεόρασης και την προωθούν στους τελικούς χρήστες του αντίστοιχου υποδικτύου. Οι κάρτες αυτές επίσης συγκεντρώνουν όλη την IP κίνηση, που προέρχεται από τους τελικούς χρήστες του κάθε CMN και την προωθούν προς το άλλο δυκτιακό interface του αντίστοιχου CMN.



Σχ. 27 Κάρτα Δικτύου, Intel(R) Pro/100 s Desktop Adapter

Το τρίτο δικτυακό interface (δύκτιο επιστροφής), το οποίο χρησιμοποιείται για την αποστολή των πακέτων προερχόμεψνα από τους χρήστες του αντίστοιχου υποδυκτίου από τον κεντρικό κόμβο διανομής προς το σταθμό εκπομπής, διαφέρει στους δύο CMN. Ο πρώτος CMN μέσο ενός ψηφιακού ISDN modem το οποίο κάνει χρήση του πρωτοκόλλου PPP στέλνει τα IP δεδομένα σε έναν Dial-up Server ο οποίος βρίσκεται στον κεντρικό σταθμό εκπομπής.Το modem που χρησιμοποιήθηκε είναι της ISDN Network Termination Units της INTRACOM (Σχήμα 28)



Σχ. 28 ISDN Network Termination Unit της INTRACOM

Στον δεύτερο CMN(που επικοινωνεί με τον server) για το κανάλι επιστροφής χρησιμοποιήσαμε μια ασύρματη on-board κάρτας η οποία κάνοντας χρήση του πρωτοκόλλου 802.11 αποστέλλει τα ACK στον κεντρικό σταθμό βάσης. Η κάρτα που χρησιμοποιήσαμε για να πετύχουμε τη ασύρματη ζεύξη ήταν της Level-One (Σχήμα 29)



Σχ. 29 Level-One Wireless Lan PCI Adapter 11g

3.7.2. Απαιτήσεις Λογισμικού (Software Requirements)

Οι ενδιάμεσοι κόμβοι διανομής έχουν υλοποιηθεί με βάση το λειτουργικό σύστημα των Linux το οποίο είναι ένα ελεύθερου "τύπου-Unix" λειτουργικό σύστημα και αποτελεί μία ανεξάρτητη POSIX υλοποίηση (συλλογικό όνομα μιας οικογένειας σχετικών προτύπων που διευκρινίζονται από το IEEE για να καθορίσουν τη διεπαφή προγραμματισμού εφαρμογής (API) για λογισμικό συμβατό με τις παραλλαγές του λειτουργικού συστήματος Unix). Στα χαρακτηριστικά του συμπεριλαμβάνει πολλαπλές διεργασίες, πραγματικό πολυχρηστικό περιβάλλον, εικονική μνήμη, κοινές βιβλιοθήκες, φόρτωση απαιτήσεων, δικτύωση TCP/IP και πολλά άλλα χαρακτηριστικά που δικαιολογούν τον τίτλο "τύπου-Unix". Ο πηγαίος κώδικάς του λειτουργικού συστήματος Linux είναι διαθέσιμος σε όλους τους χρήστες ενώ για τη συγκεκριμένη υλοποίηση του ενδιάμεσου κόμβου διανομής χρησιμοποιήθηκε η έκδοση Debian ETCHE.

Πριν την εγκατάσταση των εντολών για τη σωστή δρομολόγηση των πακέτων

IP έπρεπε να δοθούν οι IP διευθύνσεις των τριών δικτυακών interfaces (dvb_0, eth_1 και eth_0) σε κάθε CMN.
Στην περίπτωση του CMN που κάνει χρήση του δικτύου επιστροφής ISDN η κατανομή των IP είναι η παρακάτω. Η dvb_0, η οποία λαμβάνει τη ροή μεταφοράς MPEG-2 TS που εκπέμπεται από το κεντρικό σημείο εκπομπής που περιέχει τις Διαδικτυακές και πολυμεσικές υπηρεσίες για τους τελικούς χρήστες, τέθηκε η διεύθυνση IP 3ffe:ffff:5::2.Το interface του δικτύου πρόσβασης υλοποιήθηκε με την κάρτα eth_1. Το συγκεκριμένο interface λαμβάνει την προωθημένη κίνηση με τις δικτυακές υπηρεσίες από την κάρτα επίγειας ψηφιακής τηλεόρασης και την προωθεί έπειτα στους τελικούς χρήστες του υποδικτύου που υλοποιείται με κάποια τεχνολογία δικτύου πρόσβασης. Το συγκεκριμένο interface συγκεντρώνει επίσης όλη την IP κίνηση (αιτήσεις, επιβεβαιώσεις) που προέρχεται από τους τελικούς χρήστες του CMN και την προωθεί προς την άλλη κάρτα δικτύου eth_0. Έτσι τέθηκε ως διεύθυνση IP η 3ffe:ffff:0:3::1. Πρέπει να σημειωθεί ότι δε χρειάζεται να δοθεί συγκεκριμένη Gateway διεύθυνση και αυτό γιατί η κίνηση των πακέτων δρομολογείται με βάση συγκεκριμένων εντολών δρομολόγησης (rooting commands).Το τρίτο interface έχει υλοποιηθεί με ένα ψηφιακό modem το οποίο προωθεί όλη την IP κίνηση (αιτήσεις, επιβεβαιώσεις) που προέρχεται από την κάρτα eth_1 προς το κεντρικό σημείο εκπομπής. Στο modem αυτό τέθηκε ως IP διεύθυνση η 3ffe:ffff:0:4::2 και ως Default Gateway η 3ffe:ffff:0:4::1 η οποία αποτελεί την IP διεύθυνση του Cisco router (firewall) που βρίσκεται στο κεντρικό σημείο εκπομπής της πλατφόρμας DVB-T.

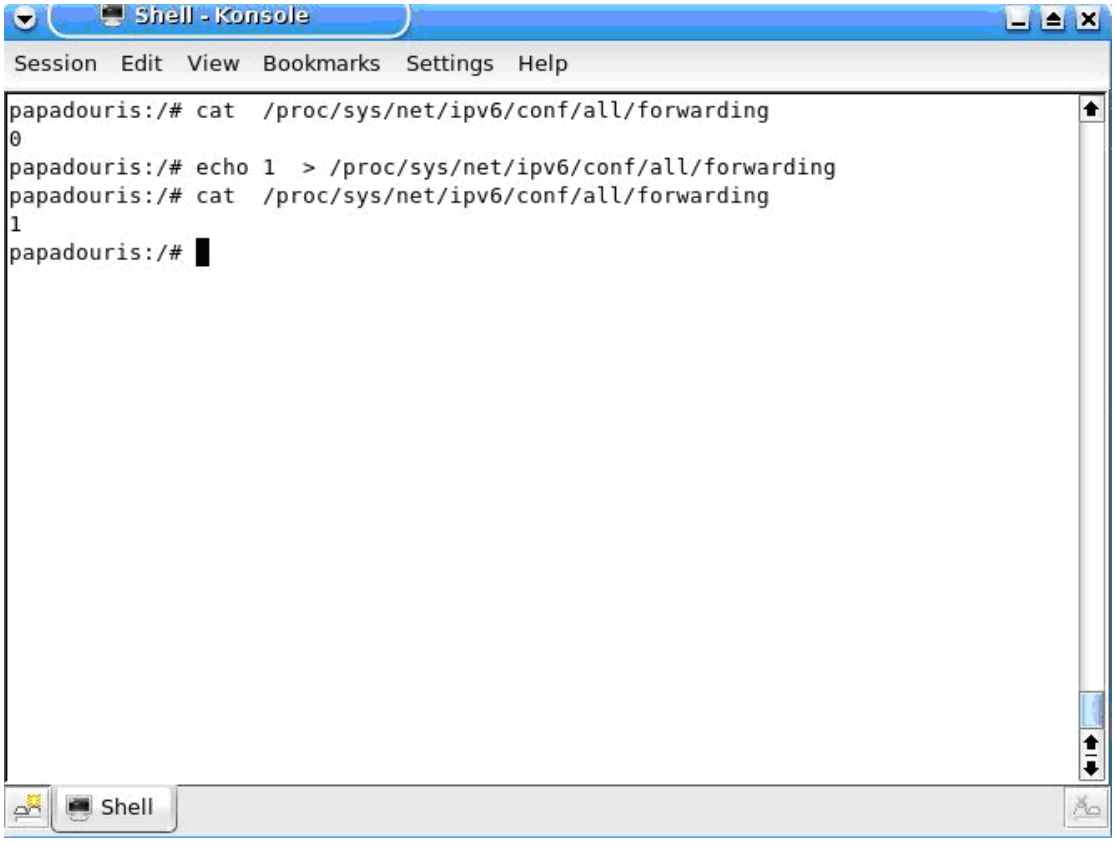
Όσον αφορά το CMN που κάνει χρήση του ασύρματου δικτύου επιστροφής 802.11g η κατανομή των ip είναι η παρακάτω. Η dvb_0, η οποία λαμβάνει τη ροή μεταφοράς MPEG-2 TS που εκπέμπεται από το κεντρικό σημείο εκπομπής που περιέχει τις Διαδικτυακές και πολυμεσικές υπηρεσίες για τους τελικούς χρήστες, τέθηκε η διεύθυνση IP 3ffe:ffff:6::2.Το interface του δικτύου πρόσβασης υλοποιήθηκε με την κάρτα eth_1. Το συγκεκριμένο interface λαμβάνει την προωθημένη κίνηση με τις δικτυακές υπηρεσίες από την κάρτα επίγειας ψηφιακής τηλεόρασης και την προωθεί έπειτα στους τελικούς χρήστες του υποδικτύου που υλοποιείται με κάποια τεχνολογία δικτύου πρόσβασης. Το συγκεκριμένο interface συγκεντρώνει επίσης όλη την IP κίνηση (αιτήσεις, επιβεβαιώσεις) που προέρχεται από τους τελικούς χρήστες του CMN και την προωθεί προς την άλλη κάρτα δικτύου eth_0. Έτσι τέθηκε ως διεύθυνση IP η 3ffe:ffff:0:1::1. Πρέπει να σημειωθεί ότι δε χρειάζεται να δοθεί συγκεκριμένη Gateway διεύθυνση και αυτό γιατί η κίνηση των πακέτων δρομολογείται με βάση συγκεκριμένων εντολών δρομολόγησης (rooting commands).Το τρίτο interface έχει υλοποιηθεί με μια ασύρματη pci κάρτα η οποία προωθεί όλη την IP κίνηση (αιτήσεις, επιβεβαιώσεις) που προέρχεται από την κάρτα eth_1 προς το κεντρικό σημείο εκπομπής. Στην κάρτα αυτή τέθηκε ως IP διεύθυνση η 3ffe:ffff:0:4::2 και ως Default Gateway η 3ffe:ffff:0:4::1 η οποία αποτελεί την IP διεύθυνση του Cisco router (firewall) που βρίσκεται στο κεντρικό σημείο εκπομπής της πλατφόρμας DVB-T.

3.8. Η συμπεριφορά του CMN σε δικτυακή κίνηση

Στα πλαίσια της υλοποίησης του ενδιάμεσου κόμβου διανομής πρέπει να προηγηθεί η εκτέλεση κάποιων εντολών που θα βοηθήσουν στην δρομολόγηση των πακέτων IP έτσι ώστε να εξασφαλιστεί η σωστή μεταφορά της Διαδικτυακής κίνησης του δικτύου. Παρακάτω παρουσιάζεται αναλυτικά η υλοποίηση του CMN που κάνει χρήση του ασύρματου δικτύου επιστροφής .Ομοίως υλοποιείται και ο CMN που κάνει χρήση ISDN καναλιού επιστροφής.

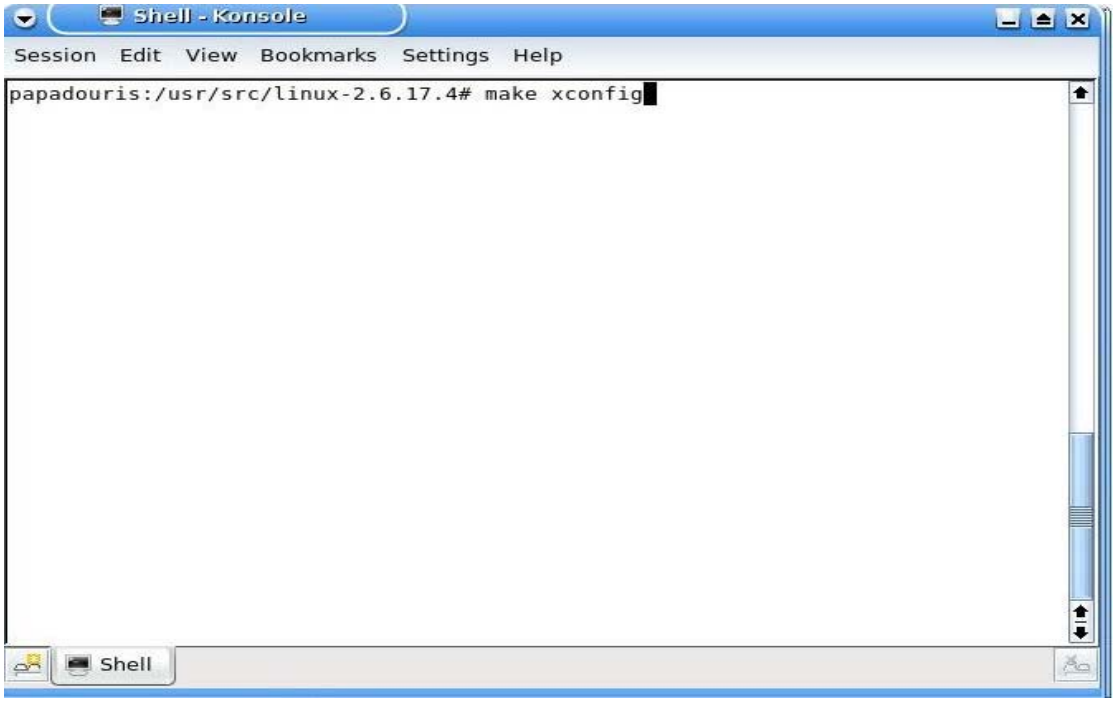
3.8.1. Μετάδοση δεδομένων με την αξιοποίηση κίνησης TCP/IP και UDP

Για να μπορέσουν να οριστούν οι εντολές δρομολόγησης (routes), βασική προϋπόθεση είναι να ενεργοποιηθεί το **ip_forwarding**, το οποίο είναι μία εφαρμογή που προωθεί τα πακέτα μεταξύ διαφορετικών interfaces (dvb_0 → eth_1 και eth_1 → eth_0). Πρώτα, πρέπει να γίνει ο έλεγχος για το αν έχει ενεργοποιηθεί η συγκεκριμένη εντολή, πληκτρολογώντας την σε ένα terminal των linux. Εφόσον το αποτέλεσμά της είναι 0, όπως φαίνεται στην πρώτη εντολή του screenshot στο σχήμα 30, στο terminal, τότε είναι απενεργοποιημένη. Αν το αποτέλεσμα είναι 1 τότε είναι ενεργοποιημένη και την αφήνουμε ως έχει. Για την ενεργοποίησή της πρέπει να πληκτρολογηθεί η δεύτερη εντολή που φαίνεται στο screenshot του ίδιου σχήματος. Τέλος, επαναλαμβάνεται ο έλεγχος, όπου φαίνεται πως ενεργοποιήθηκε κι έγινε 1.



Σχ. 30 Εντολή ip_forwarding

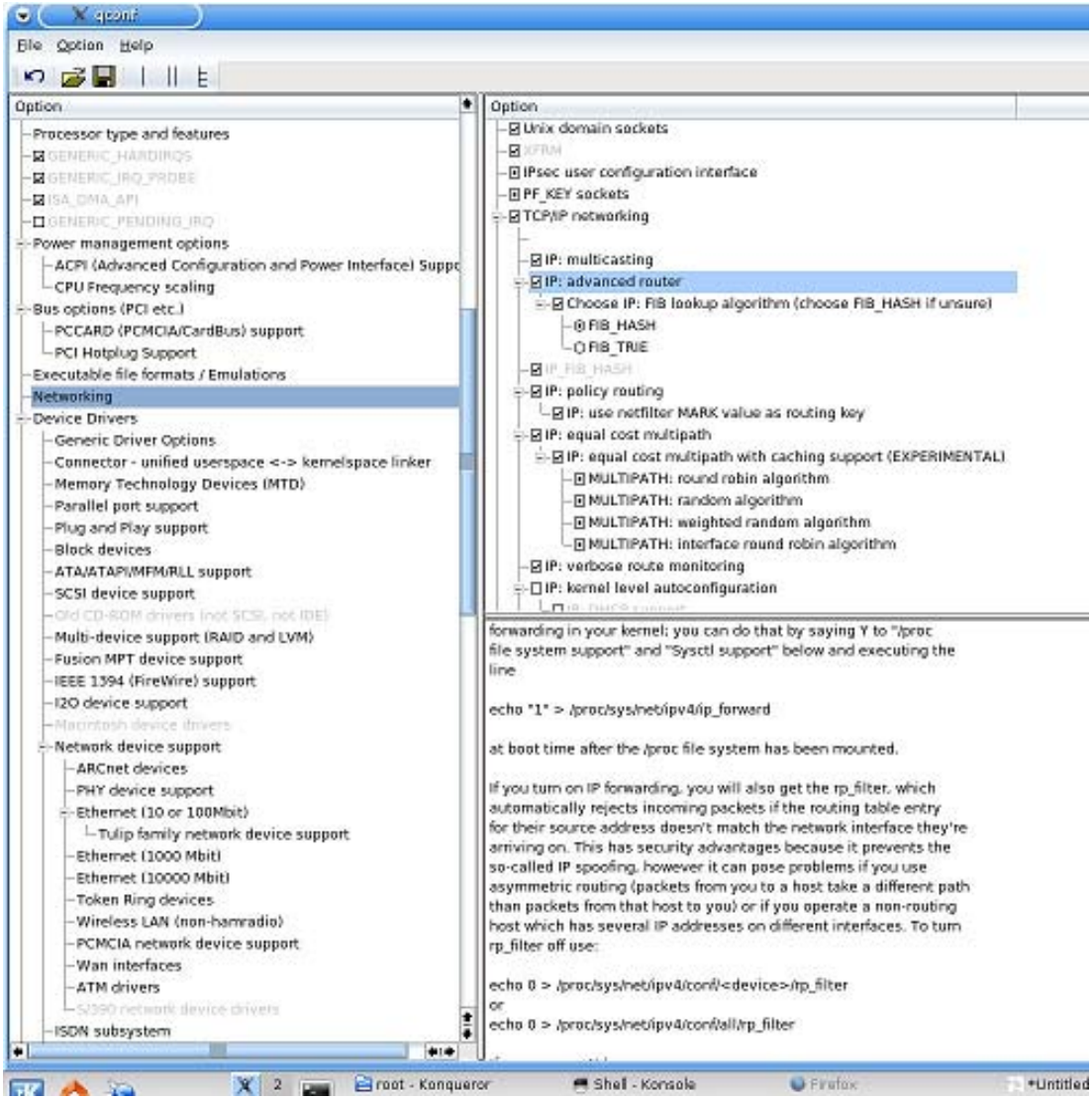
Άλλη εντολή που χρησιμοποιείται είναι η εντολή **xconfig** που παριστάνει μια γραφική προσομοίωση του πυρήνα του λειτουργικού συστήματος και παρουσιάζει όλα τα στοιχεία του. Αρχικά, πρέπει να γίνει είσοδος στο CMN ως root στο φάκελο με τον πηγαίο κώδικα του πυρήνα μας. Έπειτα πληκτρολογείται η αντίστοιχη εντολή ώστε να ενεργοποιηθεί η xconfig εφαρμογή όπως φαίνεται παρακάτω.



Σχ. 31 Εντολή xconfig

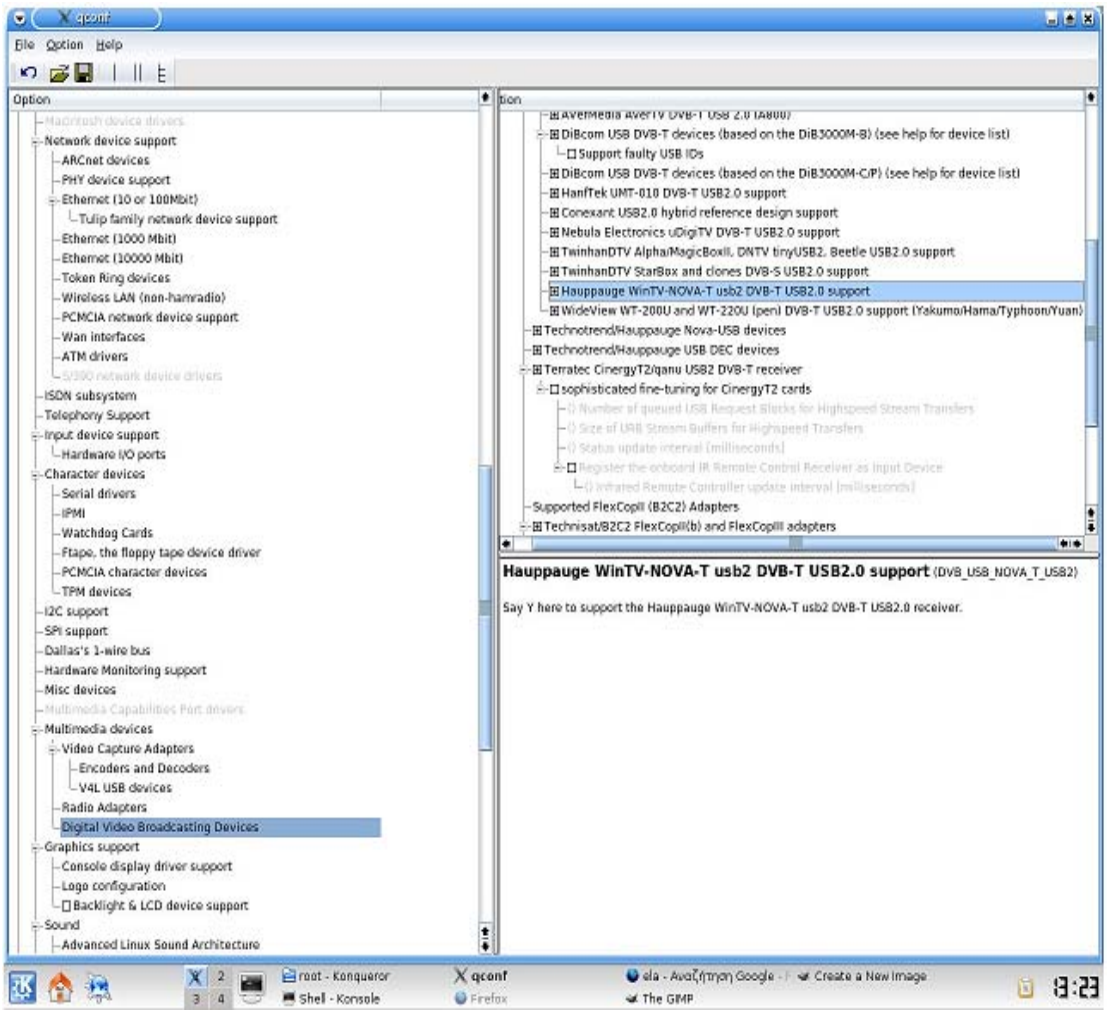
Όταν ανοίξει η εφαρμογή πρέπει να ενεργοποιηθούν οι παρακάτω επιλογές στο menu Networking.

Networking → tcp/ip networking → Ip:advanced router, όπως φαίνεται και στο screenshot του σχήματος 32 που ακολουθεί.



Σχ. 32 Εφαρμογή xconfig

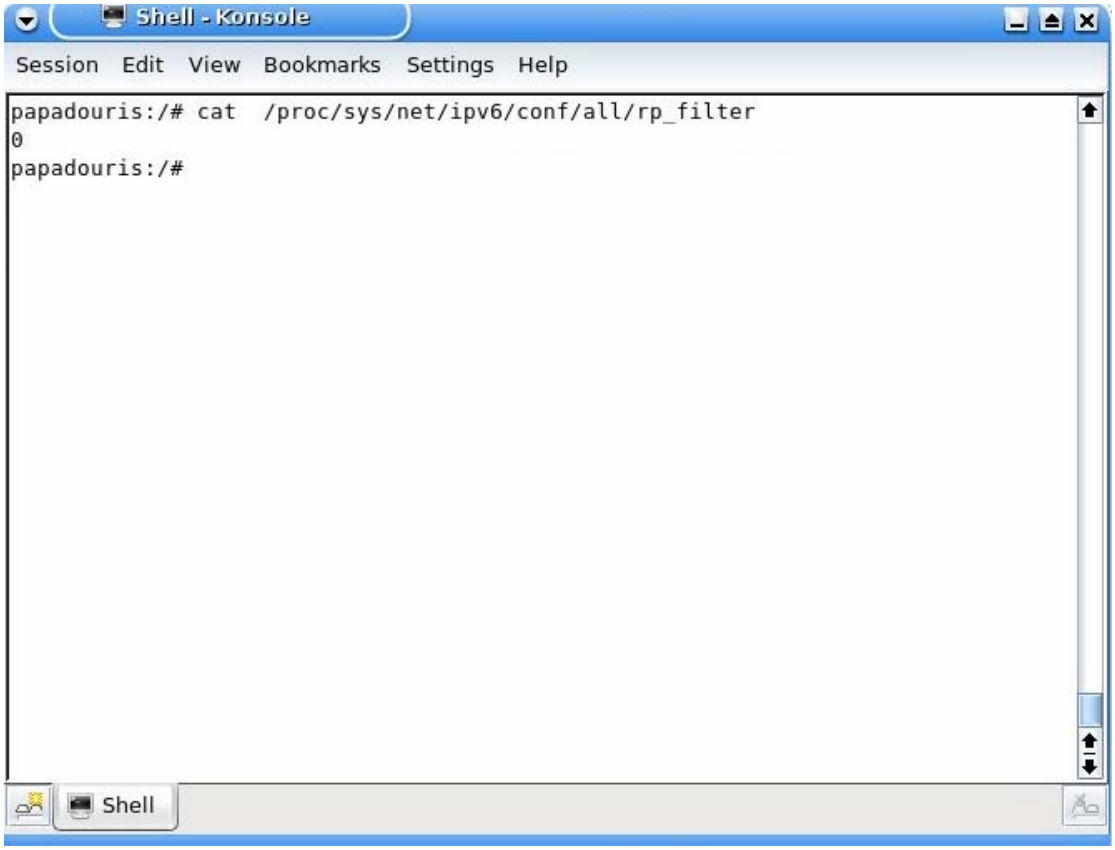
Για να ενεργοποιηθεί η κάρτα επίγειας ψηφιακής τηλεόρασης πρέπει στην ίδια εφαρμογή να βρεθεί και να επιλεγεί, όπως φαίνεται και στο screenshot του σχήματος 33 που ακολουθεί.



Σχ. 33 Επιλογή κάρτας επίγειας ψηφιακής τηλεόρασης

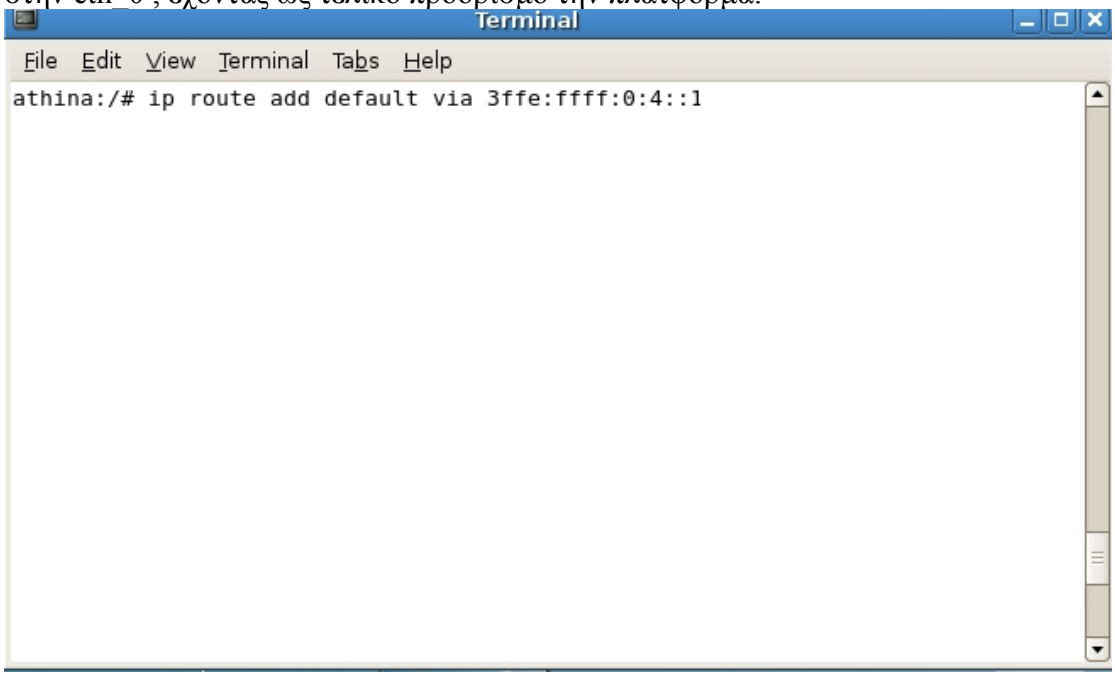
Επίσης στον ενδιάμεσο κόμβο διανομής χρησιμοποιείται η εντολή **rp-filter**, η οποία χρησιμοποιείται για να εξασφαλιστεί η αποδοχή των πακέτων μόνο από τα δικτυακά interfaces στα οποία απευθύνονται τα πακέτα IP. Ακόμα δίνει τη δυνατότητα σε αυτά τα interfaces αυτόματα να απορρίπτουν εισερχόμενα πακέτα των οποίων ο πίνακας διευθύνσεων δεν ταιριάζει με τις δικές τους διευθύνσεις.

Ο έλεγχος για το αν έχει ενεργοποιηθεί η συγκεκριμένη εντολή γίνεται πληκτρολογώντας την σε ένα terminal των linux. Εφόσον το αποτέλεσμά της είναι 0, όπως φαίνεται στην εντολή του screenshot στο σχήμα 34 στο terminal, τότε είναι ενεργοποιημένη. Αν είναι 1, τότε είναι απενεργοποιημένη και κόβει όλα τα πακέτα που έρχονται από ένα interface και φεύγουν από άλλο. Έτσι πρέπει να πληκτρολογηθεί η αντίστοιχη εντολή με την ip_forwarding, αλλά αυτή τη φορά θα έχουμε “echo 0 > proc/sys/net/ipv6/conf/all/rp_filter” .



Σχ. 34 Εντολή rp-filter

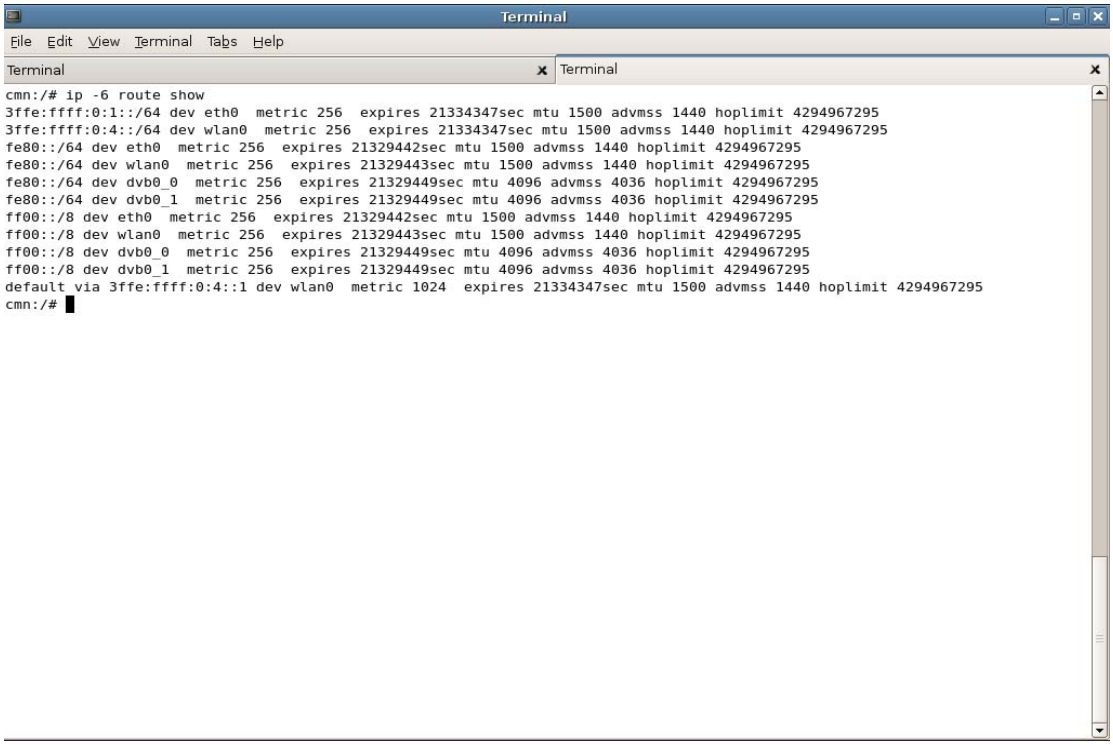
Σημαντική εντολή είναι η **ip-route** με την η οποία δίνεται η δυνατότητα να δηλωθούν default εντολές δρομολόγησης της κίνησης, μέσα στον CMN. Έτσι από το screenshot που ακολουθεί βλέπουμε ότι μέσω της εντολής ip-route προωθούνται οποιαδήποτε IP πακέτα που προέρχονται από του χρήστες, στην eth_1 κι από 'κει στην eth_0 , έχοντας ως τελικό προορισμό την πλατφόρμα.



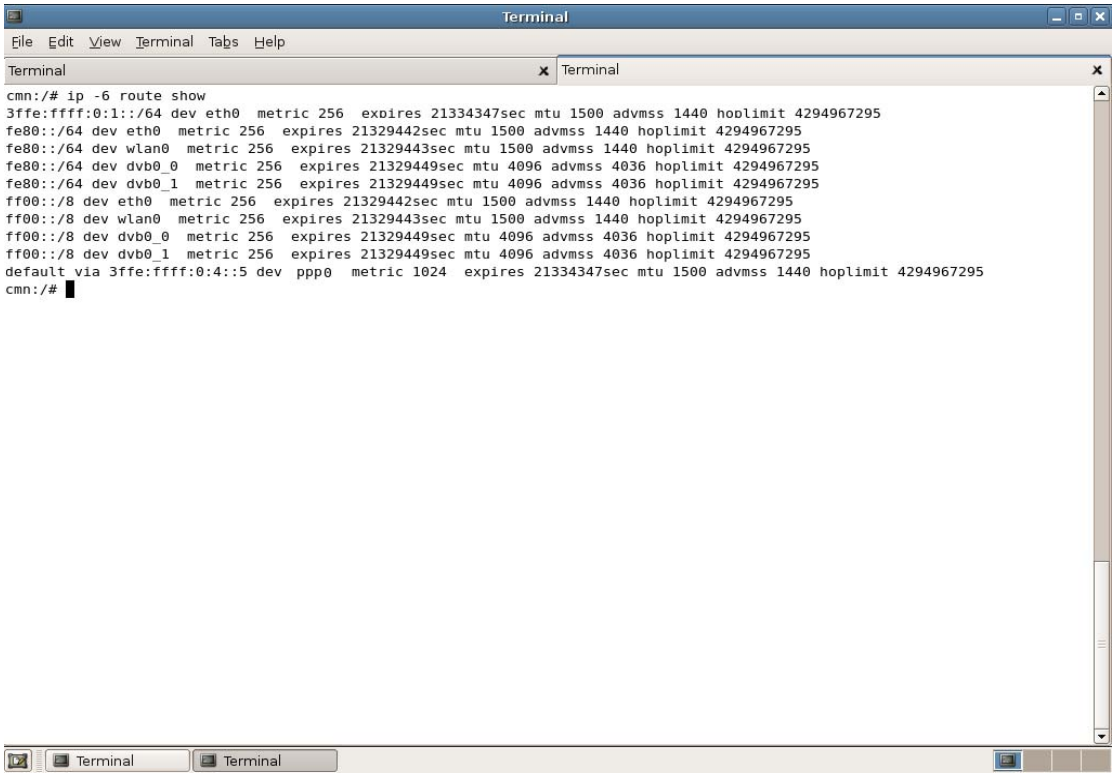
Σχ. 35 Εντολή ip-route για την δρομολόγηση της κίνησης από την eth_1 προς την eth_0

Ενώ αντίθετα η δρομολόγηση της κίνησης που έρχεται από την κάρτα ψηφιακής τηλεόρασης dvb_0 και προωθείται προς την κάρτα δικτύου eth_1 ενεργοποιείται αυτόματα, με τον καθορισμό της IP Address της κάρτας δικτύου eth_1.

Στα σχήματα που ακολουθούν παρουσιάζονται οι δρομολογήσεις των κινήσεων μέσα στους ενδιάμεσους κόμβους διανομής (CMN) με την εντολή "ip -6 route show" .

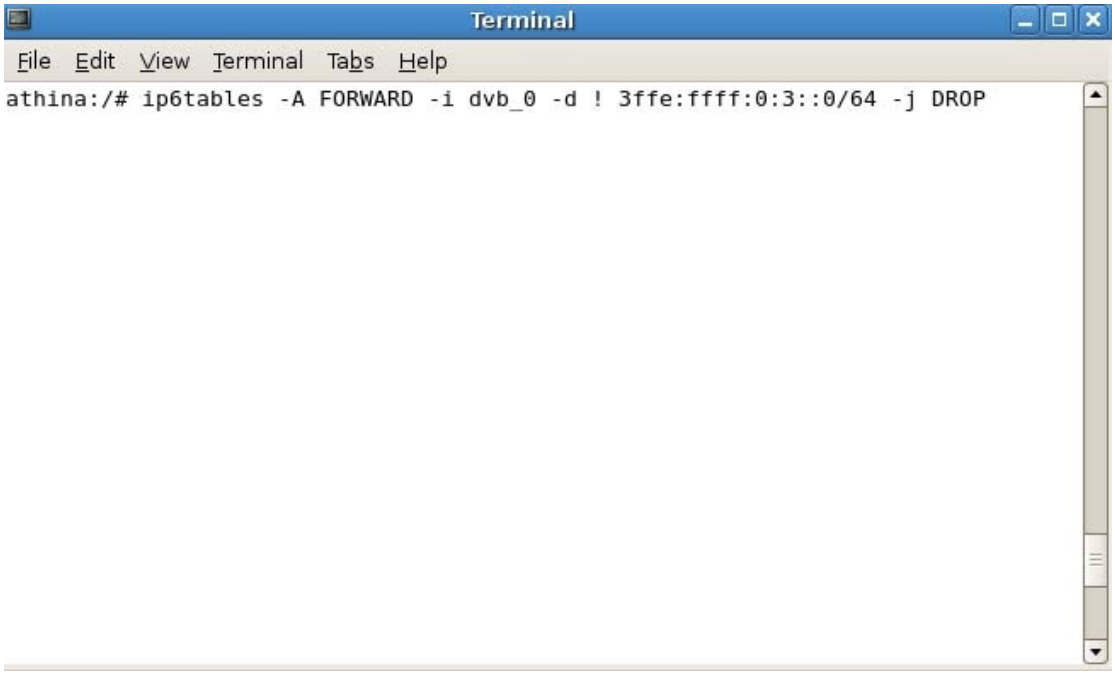


Σχ. 36 Δρομολόγηση των κινήσεων μέσα στον CMN του server



Σχ. 37 Δρομολόγηση των κινήσεων μέσα στον CMN του client

Στο συγκεκριμένο δίκτυο υπάρχει μία προεπιλογή η οποία πρέπει να αποκλειστεί. Πρόκειται για την περίπτωση που φτάσουν στον ενδιάμεσο κόμβο διανομής IP πακέτα μέσω της dvb_0 τα οποία δεν προορίζονται για αυτόν. Το αποτέλεσμα είναι να στέλνονται πίσω στον αποστολέα με αποτέλεσμα να φορτώνεται το δίκτυο με επιπλέον κίνηση. Έτσι λοιπόν για την αποφυγή τέτοιου είδους προβλημάτων χρησιμοποιείται ένα φίλτρο που «κόβει» την κίνηση η οποία φτάνει στην κάρτα dvb_0 και δεν προορίζεται για το υποδίκτυό μας. Η εντολή για την ενεργοποίηση του φίλτρου είναι η **iptables** που παρουσιάζεται στο screenshot του σχήματος 38 που ακολουθεί.

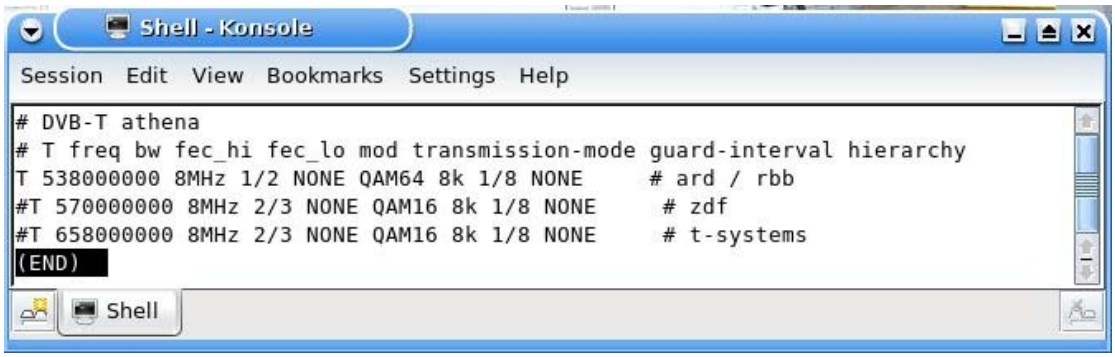


Σχ. 38 Εντολή iptables

Όταν ολοκληρωθούν όλες οι παραπάνω διαδικασίες, πρέπει να γίνει στον κόμβο ο εντοπισμός των τηλεοπτικών προγραμμάτων που εκπέμπονται από το κεντρικό σημείο εκπομπής. Για αυτό τον σκοπό χρησιμοποιείται ένα αρχείο που έχουμε ονομάσει «**athena**» το οποίο περιέχει όλα τα απαραίτητα στοιχεία του modulator της πλατφόρμας DVB-T. Τα στοιχεία αυτά είναι τα εξής:

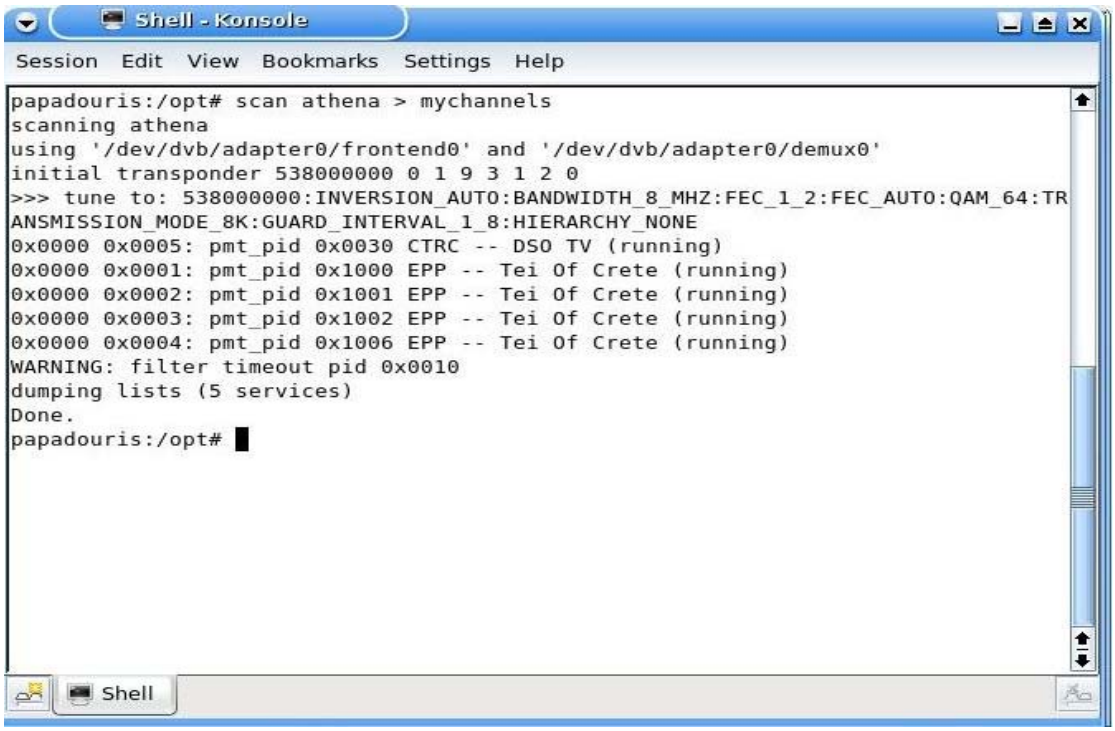
Carrier	8k
Code rate	1/2
Guard interval	1/8
Constellation	64 QAM
Frequency	538 Mhz

και τα έχουμε καθορίσει εμείς και στον modulator αλλά και στον ίδιο τον CMN. Τα παραπάνω στοιχεία φαίνονται και από το screenshot που ακολουθεί στο σχήμα 39



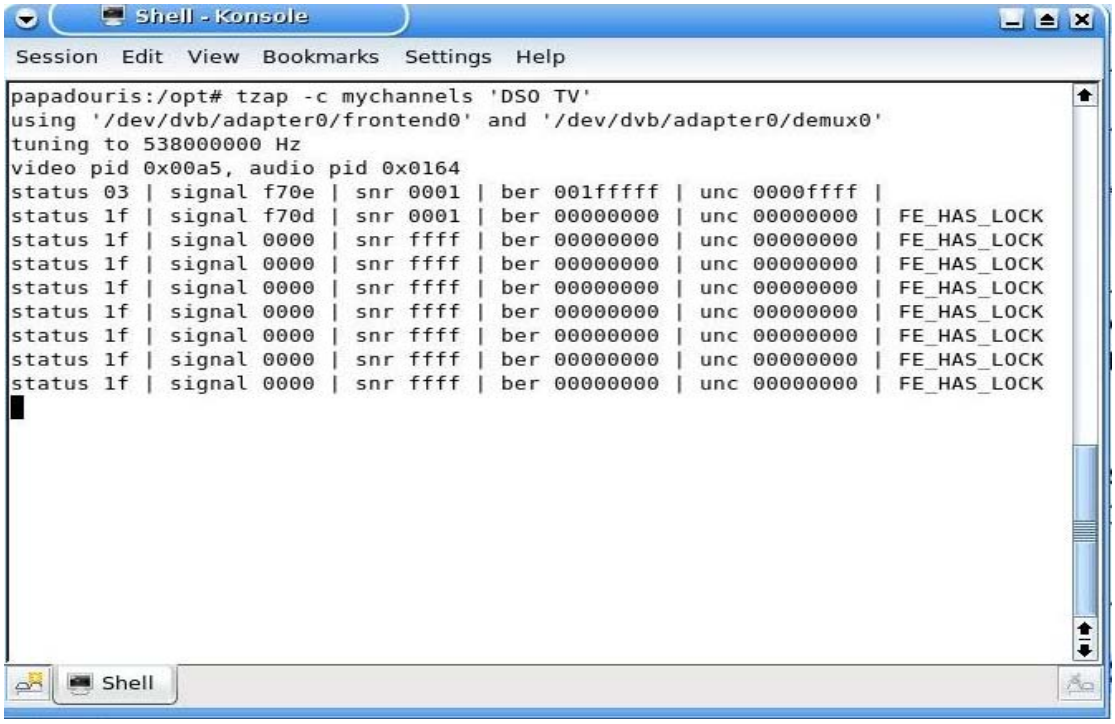
Σχ. 39 Αρχείο "athena"

Με την εντολή **scan** εντοπίζονται όλα τα τηλεοπτικά κανάλια που εκπέμπονται από την πλατφόρμα DVB-T, όπως φαίνεται και στο παρακάτω screenshot στο σχήμα 40. Παρατηρείται ότι ανιχνεύονται 5 τηλεοπτικά προγράμματα τα οποία αποθηκεύονται στη συνέχεια σε ένα αρχείο που το ονομάζουμε mychannels.



Σχ. 40 Διαθέσιμα τηλεοπτικά προγράμματα

Τέλος, με την εντολή **tzap**, παρέχεται η δυνατότητα για συντονισμό σε κάποιο από τα ανιχνεύσιμα προγράμματα σε μια συγκεκριμένη συχνότητα, όπως παρατηρείται και από το screenshot στο σχήμα 41.



Σχ. 41 Συντονισμός στο τηλεοπτικό πρόγραμμα "DSO TV"

Αυτή ήταν και η διαδικασία υλοποίησης ενός ενδιάμεσου κόμβου διανομής (CMN) με σκοπό την εξυπηρέτηση των ενεργών χρηστών-πολιτών, παροχέχοντάς τους πρόσβαση στο ευρυζωνικό δίκτυο της επίγειας ψηφιακής τηλεόρασης.

4. ΣΤΟΙΧΕΙΟΘΕΤΗΣΗ ΚΑΙ ΣΧΟΛΙΑΣΜΟΣ ΠΕΙΡΑΜΑΤΙΚΩΝ ΜΕΤΡΗΣΕΩΝ

4.1. Εισαγωγη

Κύριος σκοπός της εργασίας αυτής, είναι η αξιολόγηση των επιδόσεων του δικτύου που περιγράφηκε στο προηγούμενο κεφάλαιο. Θα πρέπει να πραγματοποιηθούν οι απαραίτητες πειραματικές μετρήσεις μέσο των οποίων θα γίνει τελικά η αξιολόγηση του δικτύου.

Πριν την περιγραφή των μετρήσεων αυτών, θα παρουσιαστούν οι εφαρμογές οι οποίες χρησιμοποιήθηκαν για τη δημιουργία κίνησης ,τόσο UDP όσο και TCP, στο δίκτυό μας.

Οι μετρήσεις που πραγματοποιήθηκαν, προσπαθούν να καλύψουν όσο το δυνατόν καλύτερα τις δυνατότητες και τα χαρακτηριστικά λειτουργίας του δικτύου. Γι' αυτόν το λόγο έπρεπε να δοθεί ιδιαίτερη προσοχή στην επιλογή των κριτηρίων αξιολόγησης του δικτύου. Τα κριτήρια λοιπόν τα οποία σε κάθε μέτρηση έκριναν τις επιδόσεις των δύο δικτύων είναι:

Για την μέτρηση δικτυακής κίνησης από σημείο σε σημείο κάνοντας χρήση του πρωτοκόλλου TCP:

1) Η μέση καθυστέρηση του δικτύου (TCP_DELAY).
2) Οι απώλειες και αναμεταδόσεις πακέτων (LOSSES AND RETRANSMISSIONS).
3) Η μέση χρήση-φόρτος του δικτύου, σε συγκεκριμένες χρονικές στιγμές, αλλά και ανά δευτερόλεπτο, οι οποίες όμως χαρακτηρίζουν σε μεγάλο βαθμό, τη συνολική χρήση του δικτύου (THROUGHPUT).
4) Sequence number

Και για την μέτρηση δικτυακής κίνησης πάλι από σημείο σε σημείο ,αλλά κάνοντας χρήση του πρωτοκόλλου UDP:

1) Η καθυστέρηση που είχαμε μεταξύ των πακέτων (Packet_to_Packet_Delay).
2) One way delay
3) Η διακύμανση της καθυστέρησης (JITTER).

4.2. Προγράμματα δημιουργίας και ανάλυσης της κίνησης ενός δικτύου

4.2.1. Δημιουργία κίνησης

4.2.1.1. Πρόγραμμα Iperf

Το Iperf είναι μια εναλλακτική λύση για τη δημιουργία TCP και UDP κίνησης καθώς και για τη μέτρηση της απόδοσης του εύρους ζώνης ενός δικτύου με TCP ή UDP κίνηση. Η εφαρμογή αυτή εφαρμόζεται τόσο σε Unix λειτουργικά όσο και σε Windows.

Επιτρέπει το συντονισμό διαφόρων παραμέτρων, μεγεθών και χαρακτηριστικών όπως μέτρηση εύρους ζώνης, απώλεια πακέτων, διακύμανση καθυστέρησης (jitter), μέγεθος αναφορών MSS/MTU και μεγέθη διαβασμένων παρατηρήσεων. Υποστηρίζει TCP window size μέσω των socket buffers, πολλαπλές ταυτόχρονες συνδέσεις μεταξύ server και client, δημιουργία UDP κίνησης με διευκρινισμένο εύρος ζώνης, multicast κίνηση και ικανότητα IPv6. Όλες οι παράμετροι πρέπει να διευκρινιστούν με το K (kilo-) και το M (mega-).

Σε καθορισμένο χρόνο μεταφοράς μπορεί να τρέξει περισσότερα από ένα σύνολα στοιχείων, επιλέγει τις καλύτερες μονάδες για τα μεγέθη της αναφοράς, και χρειάζεται πολλές περισσότερες συνδέσεις για τον server, ο οποίος μπορεί να οργανωθεί και ως daemon. Τυπώνει κατά περιόδους το μέσο εύρος ζώνης, το jitter, και τις απώλειες αναφοράς σε συγκεκριμένα χρονικά διαστήματα. Τέλος, χρησιμοποιεί αντιπροσωπευτικά streams που εξετάζουν πώς η συμπίεση στο στρώμα συνδέσεων έχει επιπτώσεις στο εύρος ζώνης του δικτύου. [16]

4.2.1.2 Πρόγραμμα Multi-Generator (MGEN)

Το πρόγραμμα MGEN είναι ένα λογισμικό ανοικτού κώδικα, και αποτελεί δημιουργία του Εργαστηρίου Έρευνας Naval (NRL) και συγκεκριμένα της ερευνητικής ομάδας που ασχολείται με τα πρωτόκολλα προηγμένων δικτύων.

Το MGEN παρέχει τη δυνατότητα να εκτελεσθούν μετρήσεις για την απόδοση δικτύων που παρέχουν IP υπηρεσίες, χρησιμοποιώντας UDP/IP κίνηση.

Το σύνολο των εργαλείων του, δημιουργεί κίνηση σε πραγματικό χρόνο έτσι ώστε το δίκτυο να μπορεί να φορτωθεί με ποικίλους τρόπους. Η κίνηση που δημιουργείται μπορεί επίσης να παραληφθεί και να καταγραφεί για τις αναλύσεις.

Χρησιμοποιούνται script αρχεία, για να καθοδηγήσουν τα αρχεία που δημιουργούνται, φορτώνοντας patterns καθ'όλη τη διάρκεια της κίνησης. Αυτά τα scripts χρειάζονται για την προσομοίωση unicast ή και multicast εφαρμογών UDP/IP.

Τα log στοιχεία του MGEN μπορούν να χρησιμοποιηθούν για να υπολογίσουν τις στατιστικές απόδοσης όσον αφορά το throughput, τα packet loss rates, το communication delay, τα ποσοστά απώλειας πακετών, την καθυστέρηση επικοινωνίας και άλλα.

Το MGEN αυτή την περίοδο εκτελείται τόσο σε διάφορα λειτουργικά τύπου Unix όσο και σε πλατφόρμες WIN32. Τέλος, κάποιες εκδόσεις του MGEN υποστηρίζουν graphical user interface. [17]

4.2.2 Ανάλυση κίνησης

4.2.2.1 Πρόγραμμα Tcpdump

Το Tcpdump είναι μια εύχρηστη εφαρμογή ανάλυσης της κίνησης ενός δικτύου και εκτελείται σε γραμμή εντολών. Επιτρέπει στους χρήστες να ανιχνεύσουν και να εμφανίσουν κάθε είδους κίνηση, στον υπολογιστή που είναι εγκατεστημένο το πρόγραμμα, καθώς επίσης και την κίνηση πακέτων από άλλα υποδίκτυα.

Το Tcpdump έχει γραφτεί από τους Van Jacobson, Craig Leres και Steven McCanne. Συνεργάζεται με τις πιο πολλές Unix εφαρμογές όπως είναι οι Linux, Solaris, BSD, Mac OS X, HP-UX και AIX. Σε τέτοια συστήματα το tcpdump

βασίζεται στην libpcap, η οποία είναι μια βιβλιοθήκη που περιέχει τις απαραίτητες ρουτίνες για να χρησιμοποιηθεί σαν φίλτρο πακέτων.

4.2.2.2 Πρόγραμμα Tcptrace

Το Tcptrace είναι ένα χρήσιμο εργαλείο, γραμμένο από τον Shawn Ostermann στο Ohio University, και χρησιμοποιείται για την ανάλυση TCP dump αρχείων. Δέχεται πακέτα που έχουν δημιουργηθεί από διάφορα packet-capture προγράμματα όπως είναι το tcpdump, snoop, etherpeek, HP Net Metrix, και WinDump. Το tcptrace μπορεί να παράγει αρχεία τα οποία περιέχουν διαφόρους τύπους πληροφοριών για κάθε υπαρκτή κίνηση. Όπως για παράδειγμα χρόνος εξομοίωσης, bytes που έχουν σταλεί και παραληφθεί, retransmissions, round trip times, window advertisements, throughput, και άλλα . Μπορεί επίσης να παράγει γραφικές παραστάσεις για κάθε μια από τις παραπάνω πληροφορίες, για περαιτέρω ανάλυση. [18]

4.2.2.3. Πρόγραμμα Xpl2gpl

Το xpl2gpl είναι ένα AWK script που μετατρέπει τα αρχεία που προκύπτουν από το tcptrace και είναι τύπου .xpl σε αρχεία τύπου .gpl. Έτσι έχουμε τη δυνατότητα να εισάγουμε τα παραγόμενα .gpl αρχεία στο Excel και να δημιουργήσουμε τις γραφικές παραστάσεις τους. [19]

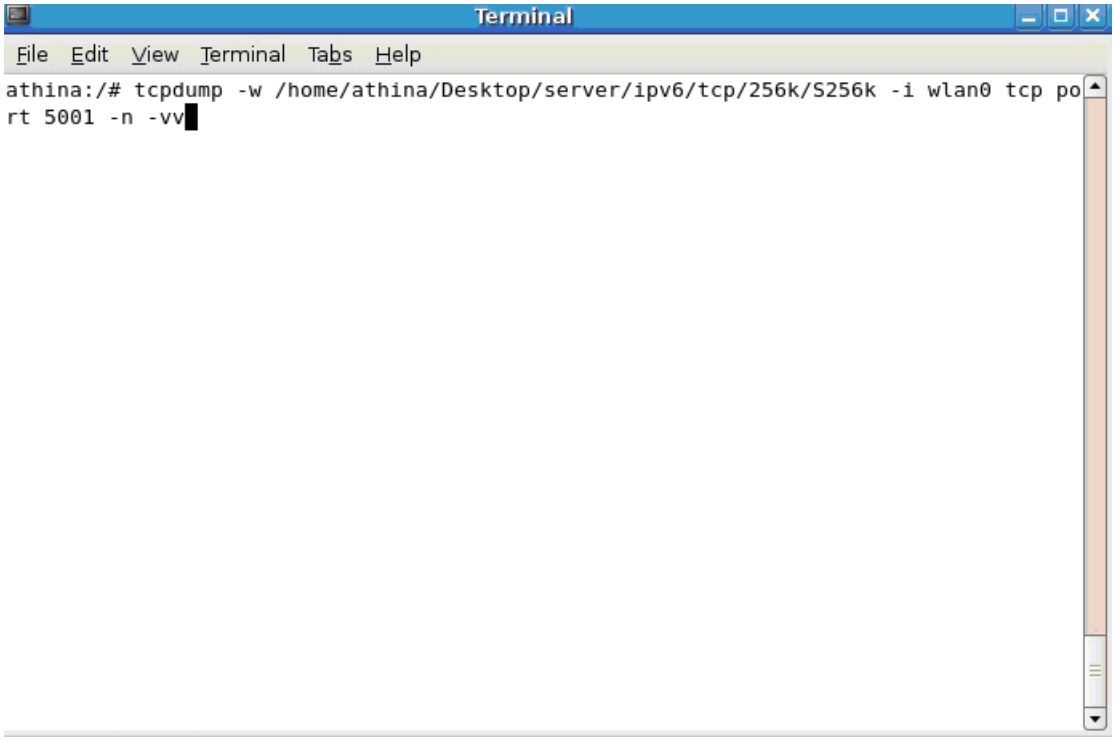
4.2.2.4. Προγράμματα της γλώσσας προγραμματισμού Perl

Η Perl είναι μια πρακτική και δυναμική, ταυτόχρονα γλώσσα προγραμματισμού που δανείζεται τα χαρακτηριστικά της γνωρίσματα από τη C, τη shell scripting (sh), AWK, sed, Lisp, και από πολλές άλλες γλώσσες προγραμματισμού. Προγράμματα, που ο αλγόριθμός τους είναι δομημένος με βάση τη γλώσσα αυτή, χρησιμοποιούνται για τον υπολογισμό της διακύμανσης της καθυστέρησης των πακέτων (jitter) .

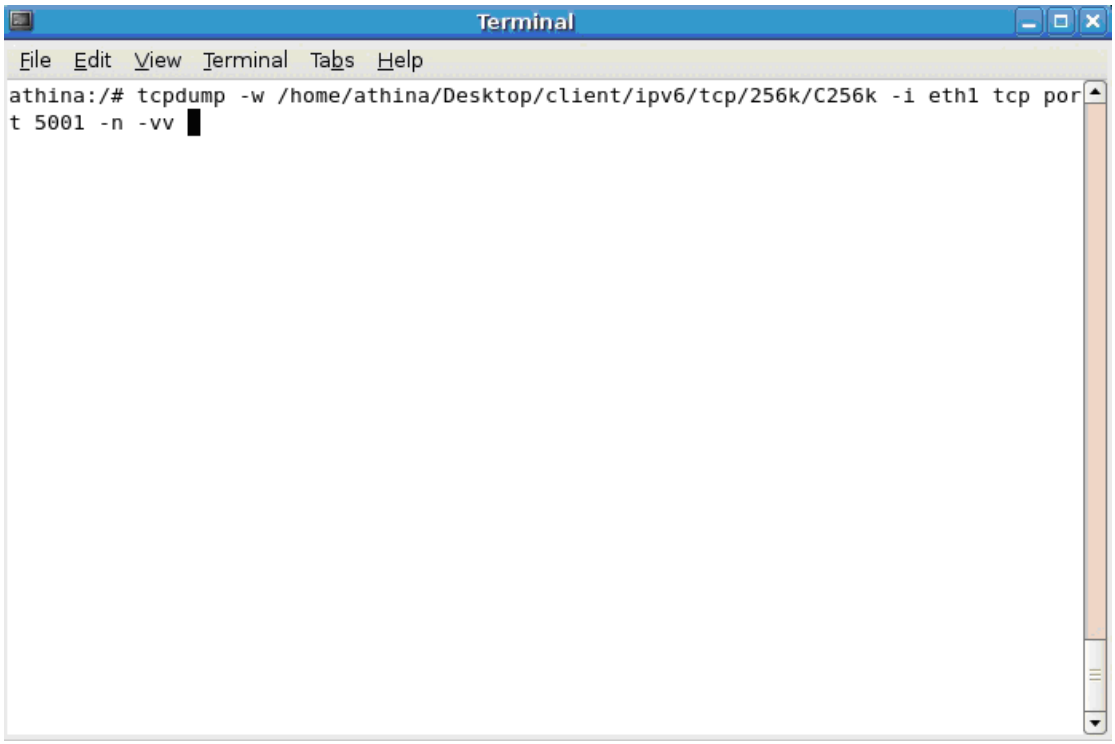
4.2.3. Παράδειγμα δημιουργίας – ανάλυσης δικτυακής κίνησης TCP

4.2.3.1. Δημιουργίας δικτυακής κίνησης TCP

Παρακάτω θα αναλυθεί η δημιουργία και η ανάλυση μιας TCP κίνησης έτσι ώστε να γίνει κατανοητή η λειτουργία των εφαρμογών που προαναφέρθηκαν. Πριν το στάδιο της δημιουργίας της κίνησης μας πρέπει να ενεργοποιήσουμε το πρόγραμμα tcpdump τόσο στον Server (Σχήμα 42) όσο και στον Client (Σχήμα 43) για να μπορεί να ανιχνεύσει όλη την κίνηση (sniffing), πριν αρχίσει η ροή της. Τα στοιχεία της κίνησης αυτής θα καταχωρούνται στο αρχείο με το όνομα S256κ και C256k για τον server και client αντίστοιχα.



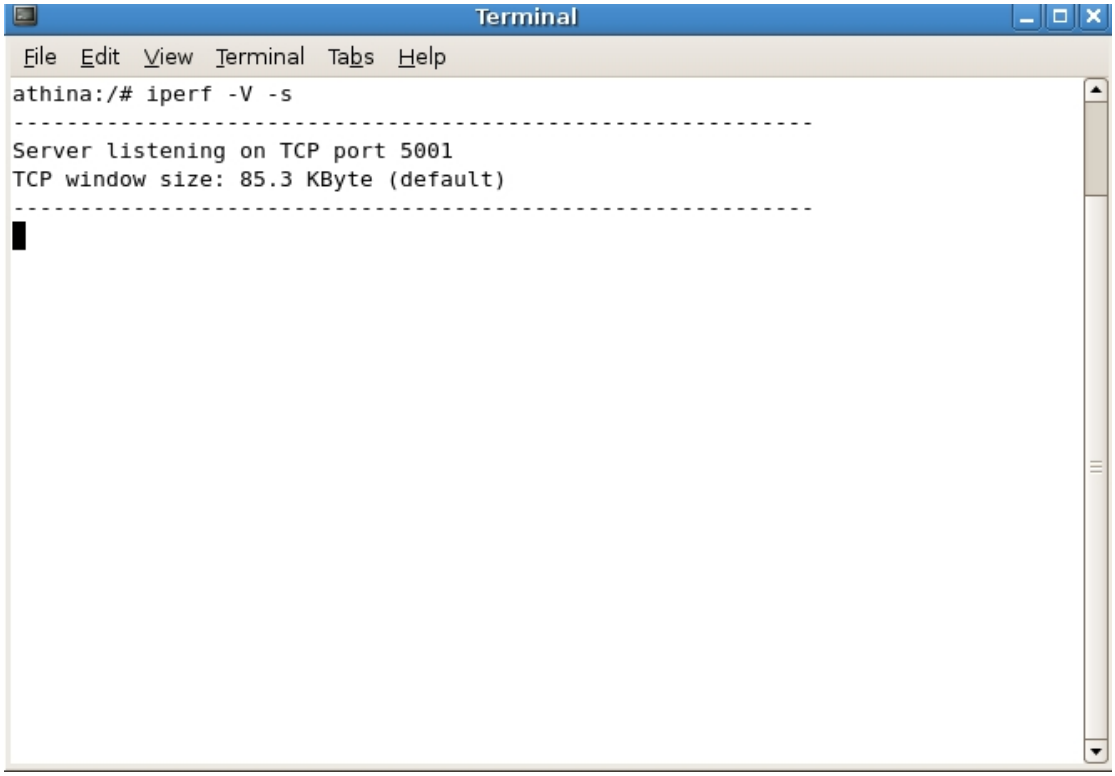
Σχ. 42 Εντολή tcpdump στον Server



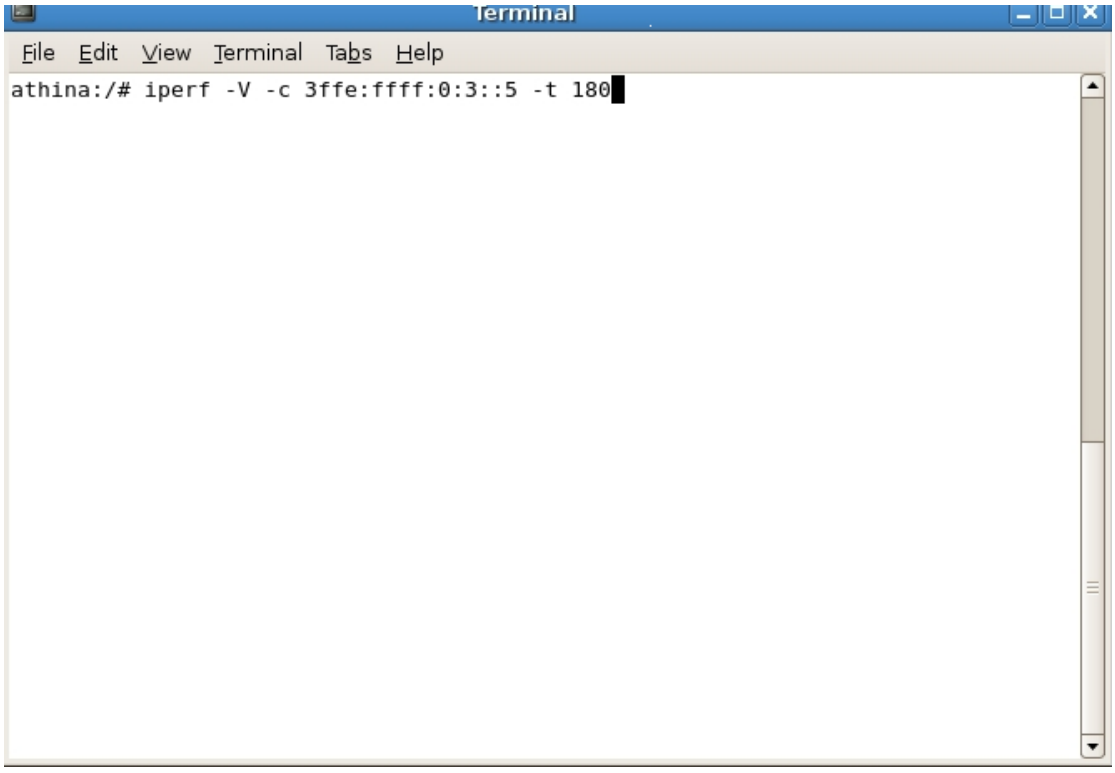
Σχ. 43 Εντολή tcpdump στον Client

Κι αφού τώρα είναι έτοιμα τα φίλτρα της κίνησης, μπορούμε να δημιουργήσουμε την ροή της. Έτσι στον Server, όπως φαίνεται και στο screenshot του σχήματος 44 εκτελείται η εντολή του iperf για να δημιουργήσει δικτυακή κίνηση

προς τον Client για χρονικό διάστημα 180 sec. Ενώ για τον client παρουσιάζεται στο screenshot του σχήματος 45



Σχ. 44 Εντολή iperf στον Server



Σχ. 45 Εντολή iperf στον Client

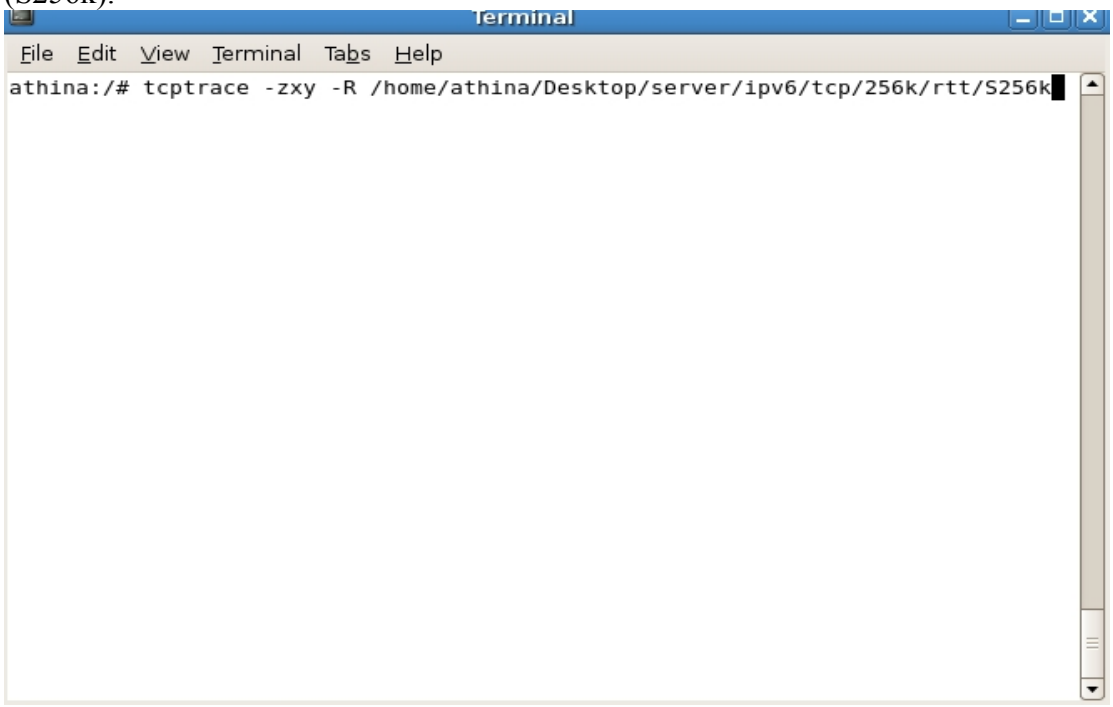
Μετά το τέλος της κίνησης αυτής, παρατηρούμε αναφέρονται τα ανιχνευμένα πακέτα τα οποία εχουν σταλθεί από τον Server.

4.2.3.2 Ανάλυση δικτυακής κίνησης TCP

Στο επόμενο στάδιο έχουμε την ανάλυση αυτής κίνησης. Η ανάλυση πρέπει να γίνει σε υπολογιστή με λειτουργικό σύστημα Linux.

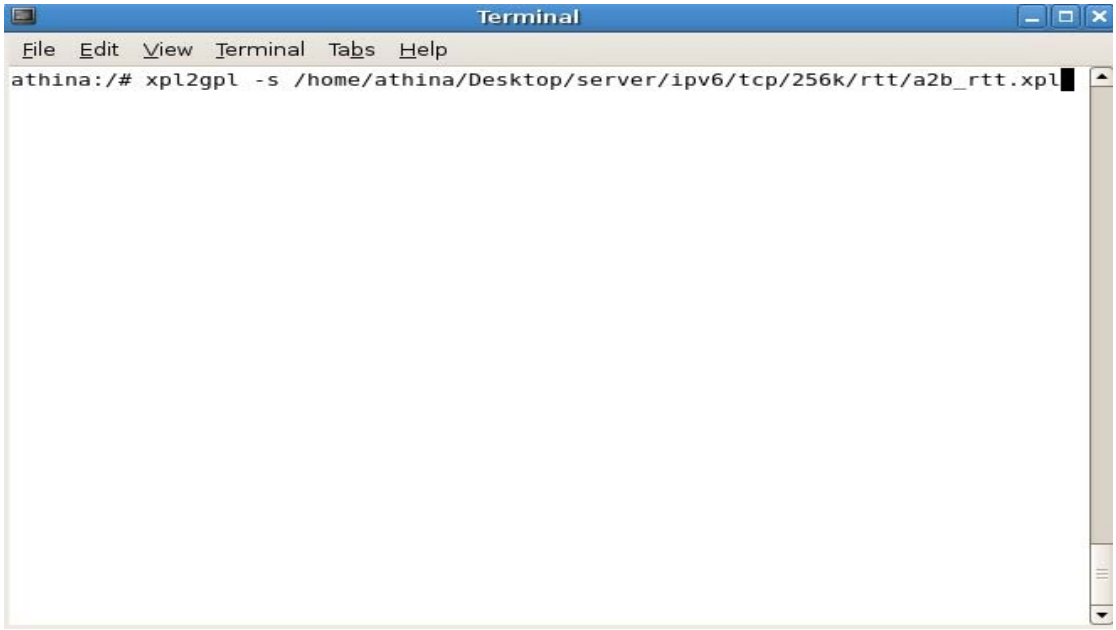
Παίρνουμε τα δύο αρχεία που δημιουργήθηκαν (S256k και C256k) και εκτελούμε για το καθένα την εφαρμογή tcptrace.

Στο σχήμα 46 ακολουθεί η εντολή του tcptrace, όπου παράγει το TCP Delay (RTT) του δικτύου. Για να υπολογιστεί αυτό χρειάζεται μόνο το αρχείο του Server (S256k).



Σχ. 46 Εντολή tcptrace για το TCP Delay του δικτύου

Ύστερα από αυτή την εντολή δημιουργούνται δύο αρχεία(a2b_rtt.xpl, b2a_rtt.xpl). Αυτά τα αρχεία τα μετατρέπουμε, όπως φαίνεται και στο σχήμα 47, σε μορφή .gpl για να γίνει στη συνέχεια η δημιουργία του αρχείου που θα εισάγουμε στο Microsoft Office Excel των Windows και να δημιουργήσουμε τις γραφικές τους.



Σχ. 47 Εντολή xpl2gpl για το RTT

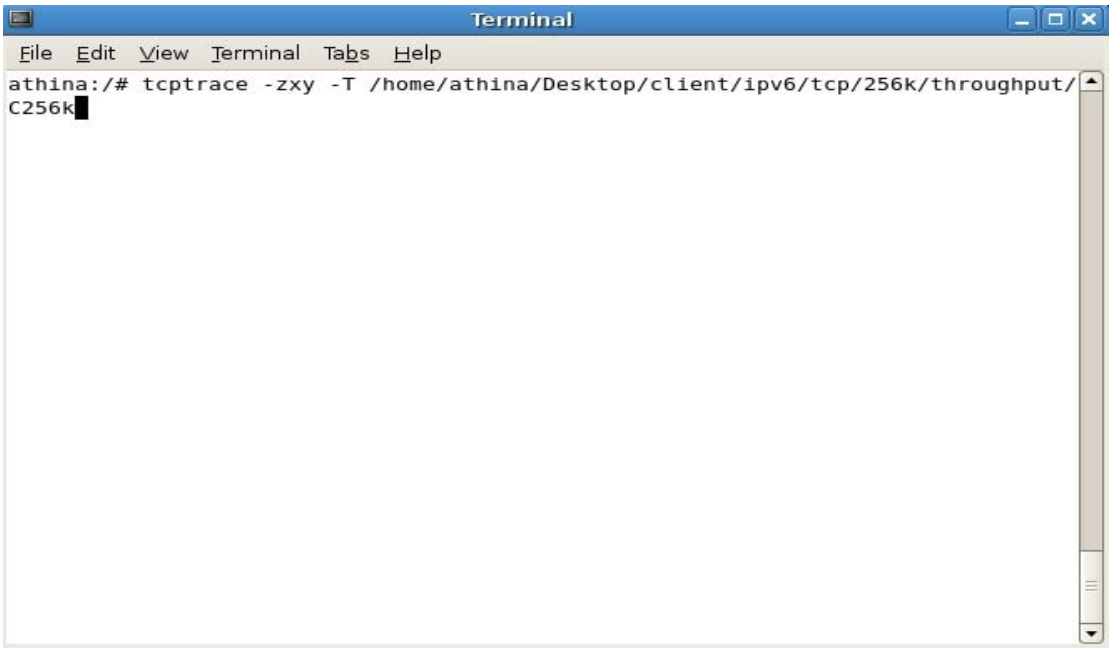
Με την εκτέλεση της παραπάνω εντολής δημιουργούνται κάποια αρχεία από τα οποία μας ενδιαφέρει ένα αρχείο με όνομα a2b_rtt.dataset.red.dot**.** Στη συνέχεια τρέχουμε αυτό το αρχείο με τη βοήθεια ενός script(ipv6_rtt.pl) σε γλώσσα Perl. Η εντολή φαίνεται στο σχήμα 48 και δέχεται τρία ορίσματα. Στο πρώτο όρισμα έχουμε το όνομα του script που θέλουμε, στο δεύτερο το όνομα του αρχείου που θέλουμε να τρέξουμε και στο τρίτο όρισμα το ανά πόσα δευτερόλεπτα θα παίρνουμε τα δείγματα που στην περίπτωση μας είναι 0,5sec.



Σχ. 48 Εντολή perl για το RTT

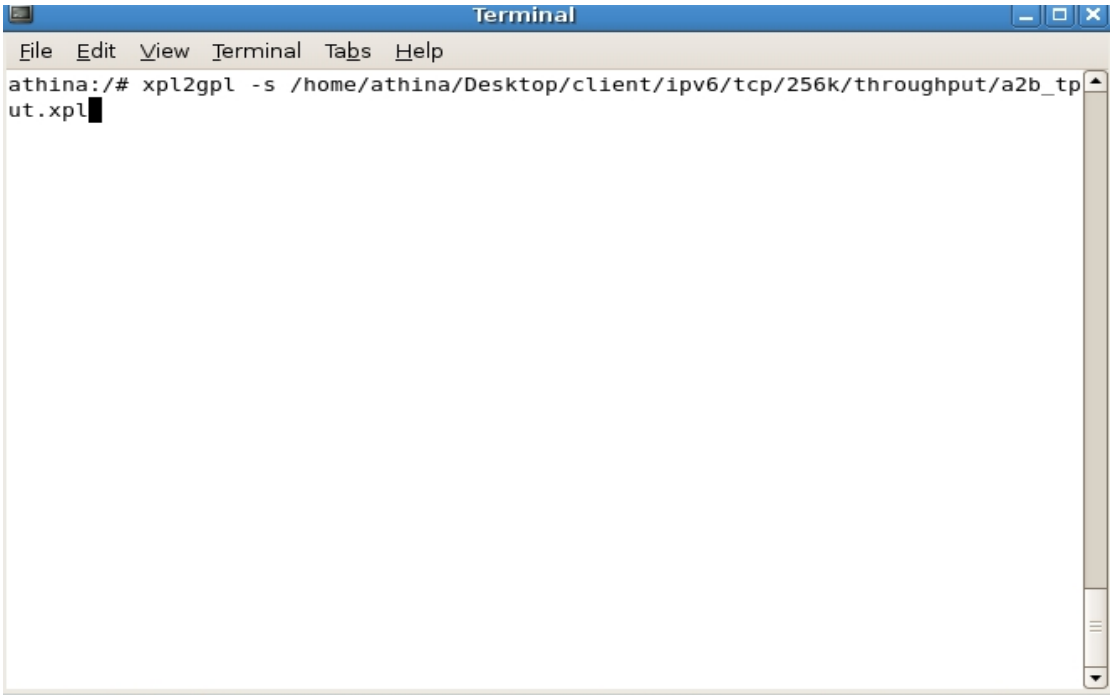
Μετά την εκτέλεση της perl παίρνουμε το τελικό αρχείο, το οποίο και εισάγουμε στο Microsoft Office Excel για να πάρουμε την γραφική παράσταση του RTT.

Η επόμενη μέτρηση έχει να κάνει με τον υπολογισμό του throughput (ωφέλιμη ταχύτητα) του δικτύου. Το throughput είναι τα Bytes που φτάνουν στο port ανά δευτερόλεπτο και η εντολή του tcptrace είναι αυτή που ακολουθεί στο σχήμα 49:



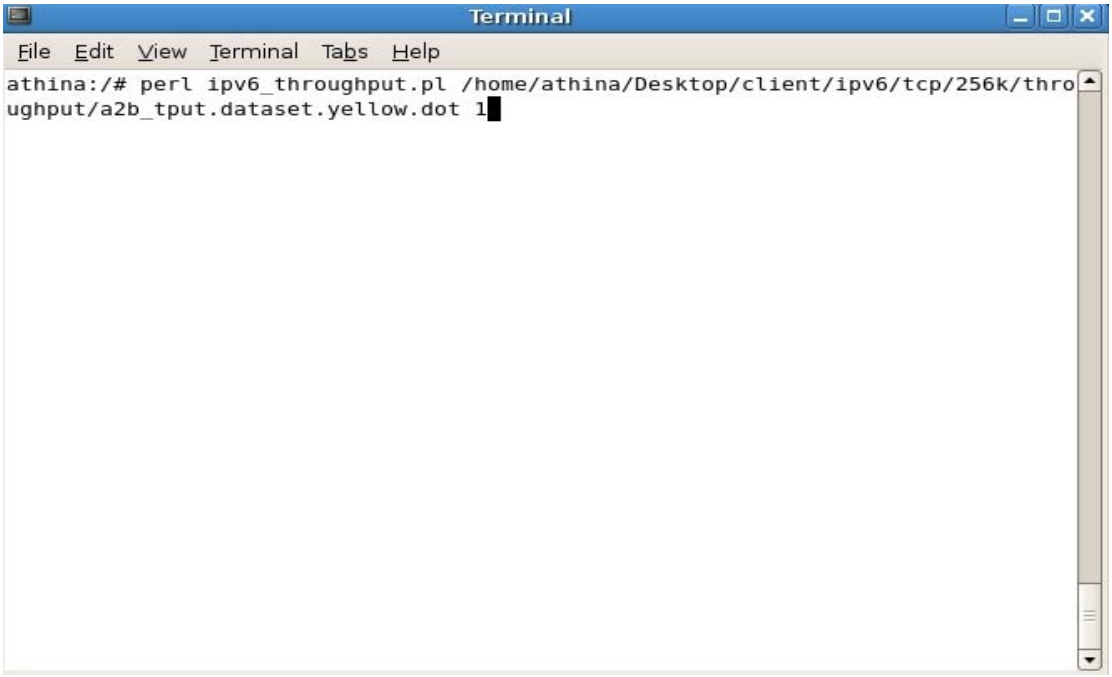
Σχ. 49 Εντολή tcptrace για τον υπολογισμό του throughput (ανά δευτερόλεπτο)

Η εντολή tcptrace δημιουργεί δύο αρχεία(a2b_tput.xpl και b2a_tput.xpl). Στη συνέχεια γίνεται η μετατροπή τους σε αρχεία με κατάληξη .gpl, η οποία φαίνεται στο σχήμα 50:



Σχ. 50 Εντολή xpl2gpl για το Throughput

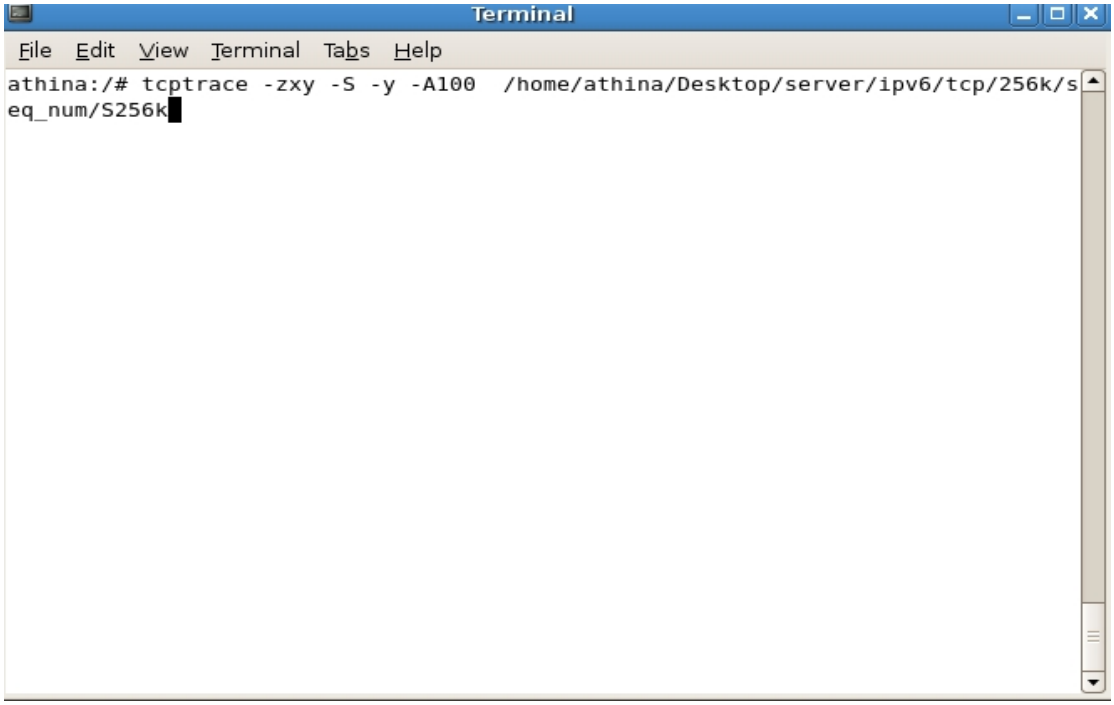
Έτσι δημιουργούνται κάποια αρχεία. Αυτό που μας ενδιαφέρει για περαιτέρω ανάλυση είναι το αρχείο a2b_tput.dataset.yellow.dot. Αυτό το αρχείο το τρέχουμε με τη βοήθεια του script ipv6_throughput.pl. Όπως έχει ήδη αναφερθεί η εντολή perl παίρνει τρία ορίσματα. Το script που θέλουμε, το όνομα του αρχείου και το ανά πόσα δευτερόλεπτα παίρνουμε τα δείγματα στη γραφική παράσταση. Στην περίπτωση του throughput τα δείγματα θα είναι ανά 1sec. Η εκτέλεση της perl φαίνεται στο σχήμα 51:



Σχ. 51 Εντολή perl για το Throughput

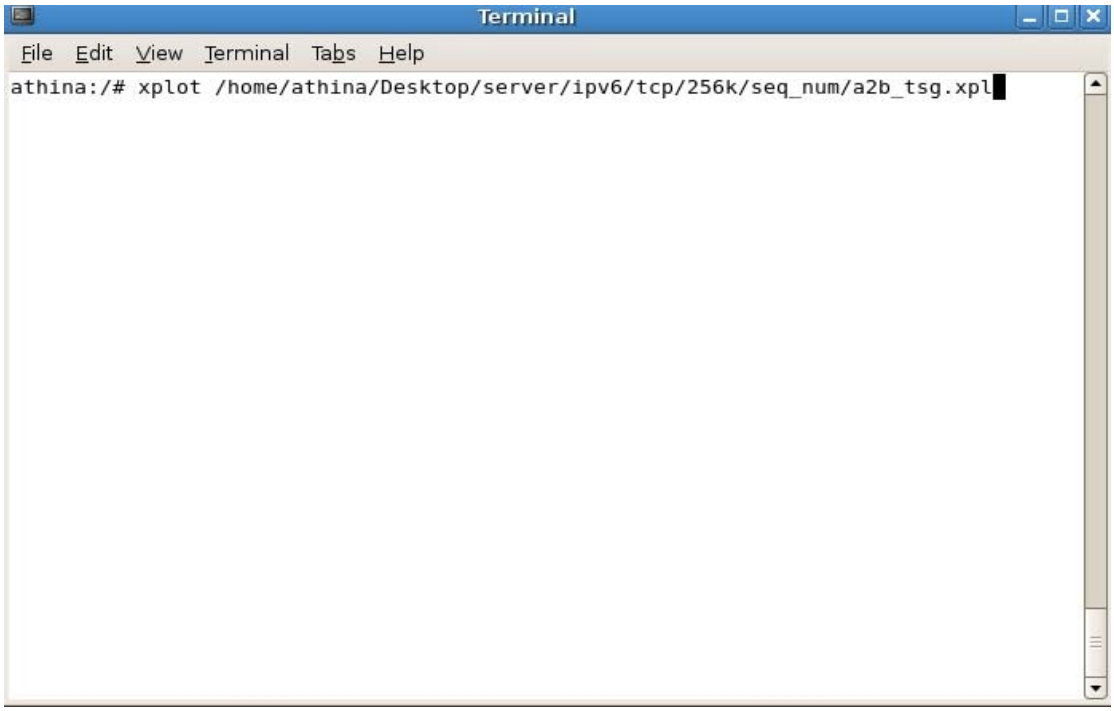
Μετά την εκτέλεση της perl παίρνουμε το τελικό αρχείο, το οποίο και εισάγουμε στο Microsoft Office Excel για να πάρουμε την γραφική παράσταση του Throughput.

Η τελευταία μέτρηση που πραγματοποιούμε είναι το Time sequence. Το time sequence είναι η ακολουθία μεταφοράς των TCP segments. Επίσης, δείχνει τα ενδεχόμενα retransmission και φαίνεται το επίπεδο των επιβεβαιώσεων και των δεδομένων τα οποία στέλνονται. Στο σχήμα 52 παρουσιάζεται η εντολή tcptrace για τη δημιουργία του αρχείου από το οποίο θα βγάλουμε την γραφική του παράσταση. Η ανάλυση γίνεται να παίρνουμε τα δείγματα ανά 100 segments.



Σχ. 52 Εντολή tcptrace για τον υπολογισμό του Time sequence(ανά segments)

Δημιουργούνται δύο αρχεία(a2b_tsg.xpl και b2a_tsg.xpl) από τα οποία με την εντολή xplot παίρνουμε τη γραφική παράσταση. Η εκτέλεση της εντολής παρουσιάζεται στο σχήμα 53:



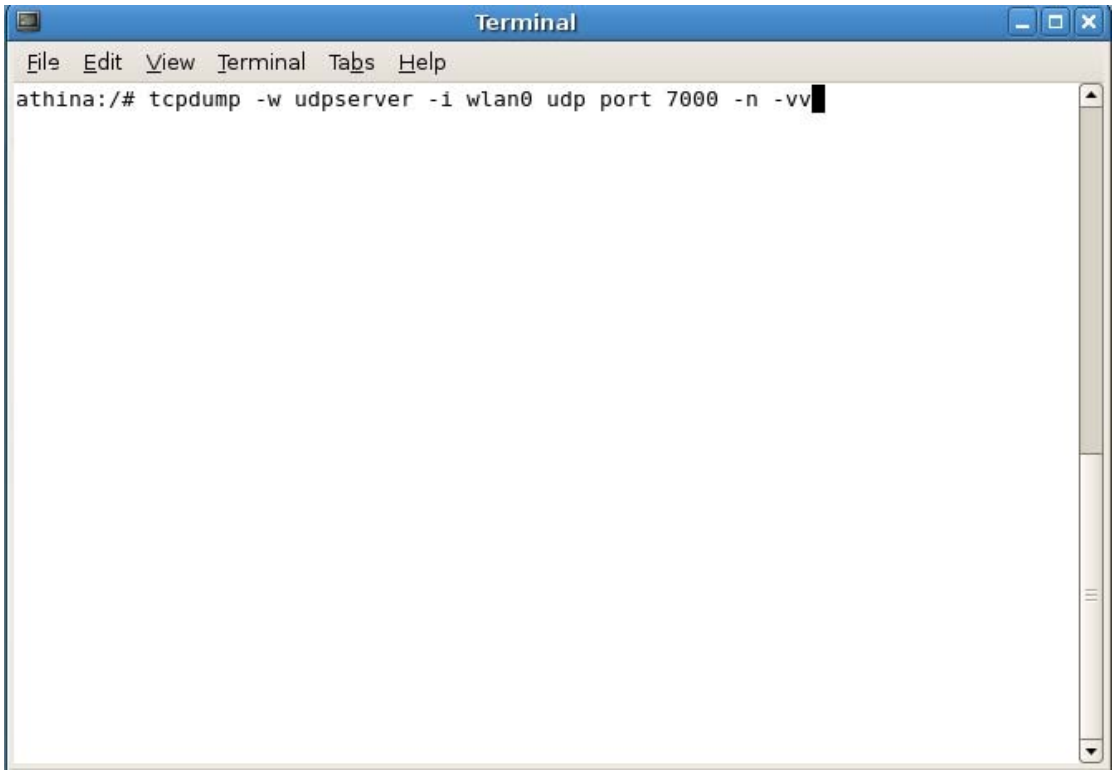
Σχ. 53 Εντολή xplot για Time sequence

4.2.4 Παράδειγμα δημιουργίας – ανάλυσης δικτυακής κίνησης UDP

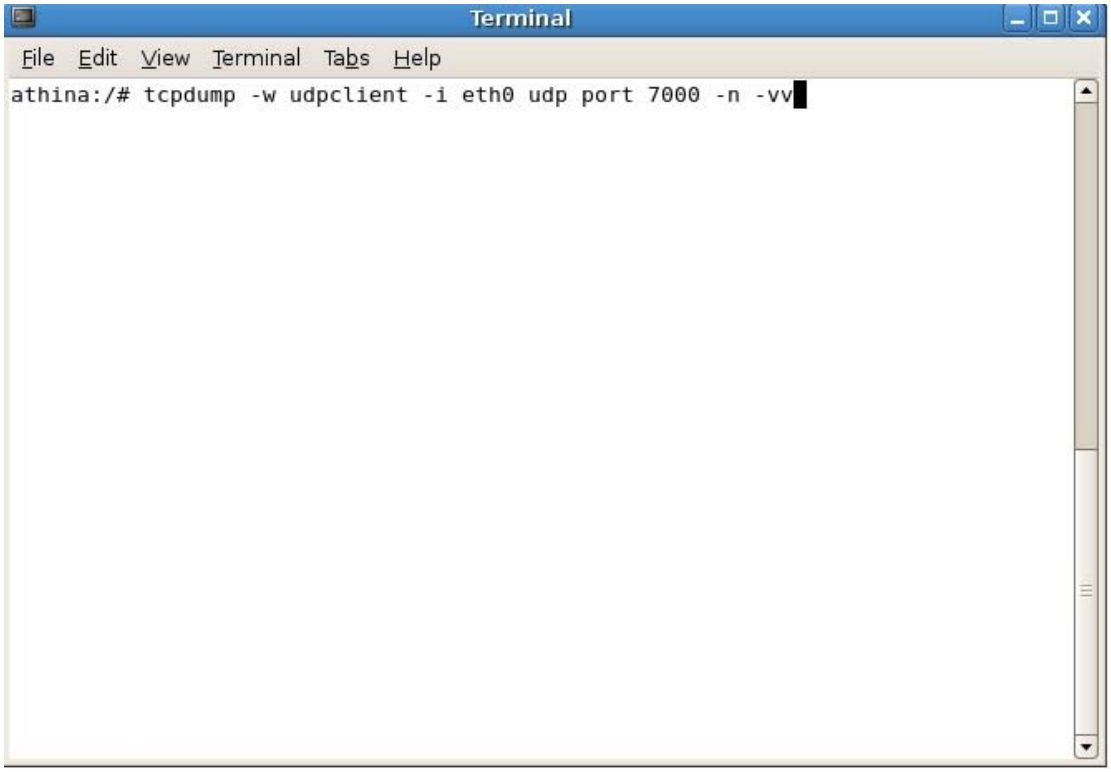
4.2.4.1 Δημιουργία δικτυακής κίνησης TCP

Για να δημιουργηθεί αυτή η δικτυακή κίνηση χρειάζεται όπως είδαμε και προηγουμένως το πρόγραμμα Mgen. Και για να ανιχνευθεί και να εμφανιστεί χρειάζεται το πρόγραμμα tcpdump στον Server και στον Client.

Αρχικά πρέπει ο Server, όπως φαίνεται και στο screenshot του σχήματος 54, να εκτελέσει την εντολή του tcpdump, για να μπορεί να ανιχνεύσει όλη την κίνηση (sniffing), πριν αρχίσει η ροή της. Τα στοιχεία της κίνησης αυτής θα καταχωρούνται στο αρχείο με το όνομα S256k. Αντίστοιχα και ο Client, όπως φαίνεται και στο screenshot του σχήματος 55, εκτελεί την εντολή tcpdump, για να μπορεί να ανιχνεύσει όλη την κίνηση (sniffing), πριν αρχίσει η ροή της. Τα στοιχεία της κίνησης αυτής καταχωρούνται στο αρχείο με το όνομα C256k.

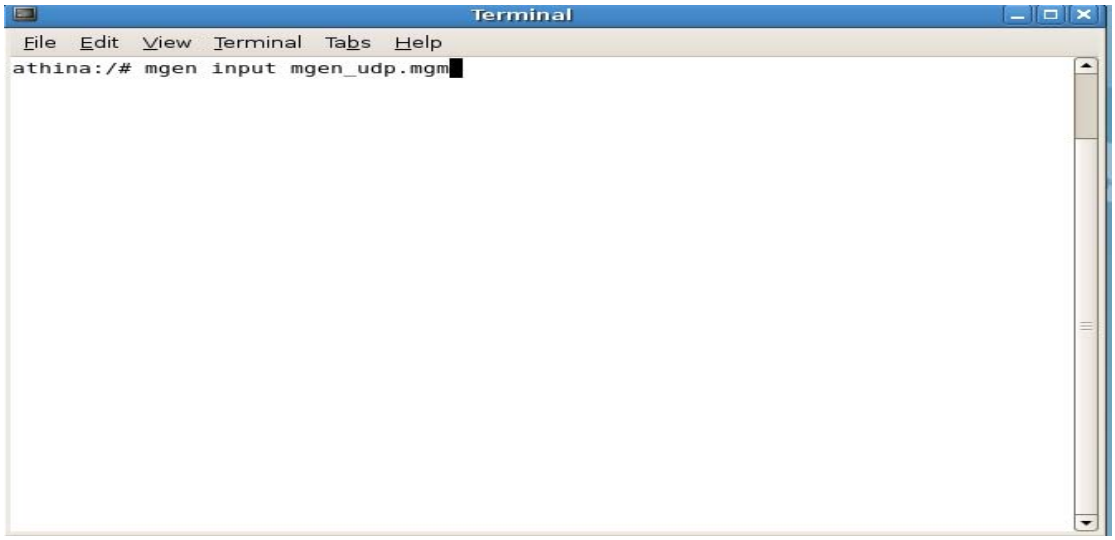


Σχ. 54 Εντολή tcpdump στον Server (Δικτυακή κίνηση UDP)

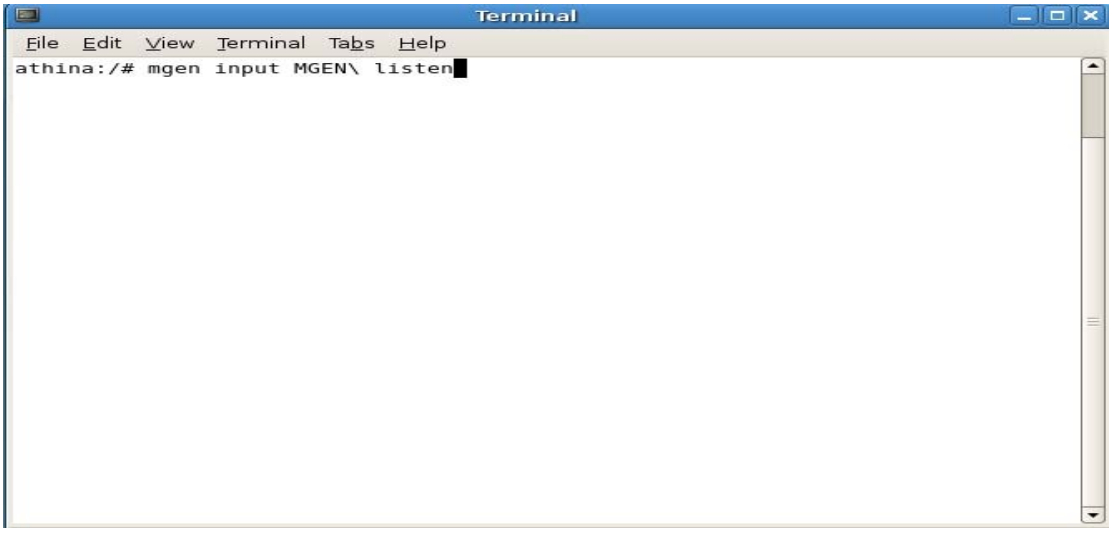


Σχ. 55 Εντολή tcpdump στον Client (Δικτυακή κίνηση UDP)

Κι αφού τώρα είναι έτοιμα τα φίλτρα της κίνησης, μπορούμε να δημιουργήσουμε την ροή της. Έτσι στον Server, όπως φαίνεται και στο screenshot του σχήματος 56, εκτελείται η εντολή του Mgen για να δημιουργήσει δικτυακή κίνηση προς τον Client για χρονικό διάστημα 180 sec. Ομοίως και στον Client, όπως φαίνεται και στο screenshot του σχήματος 57, εκτελείται η αντίστοιχη εντολή του Mgen για να δεχτεί την δικτυακή κίνηση από τον Server.



Σχ. 56 Εντολή Mgen στον Server (Δικτυακή κίνηση UDP)



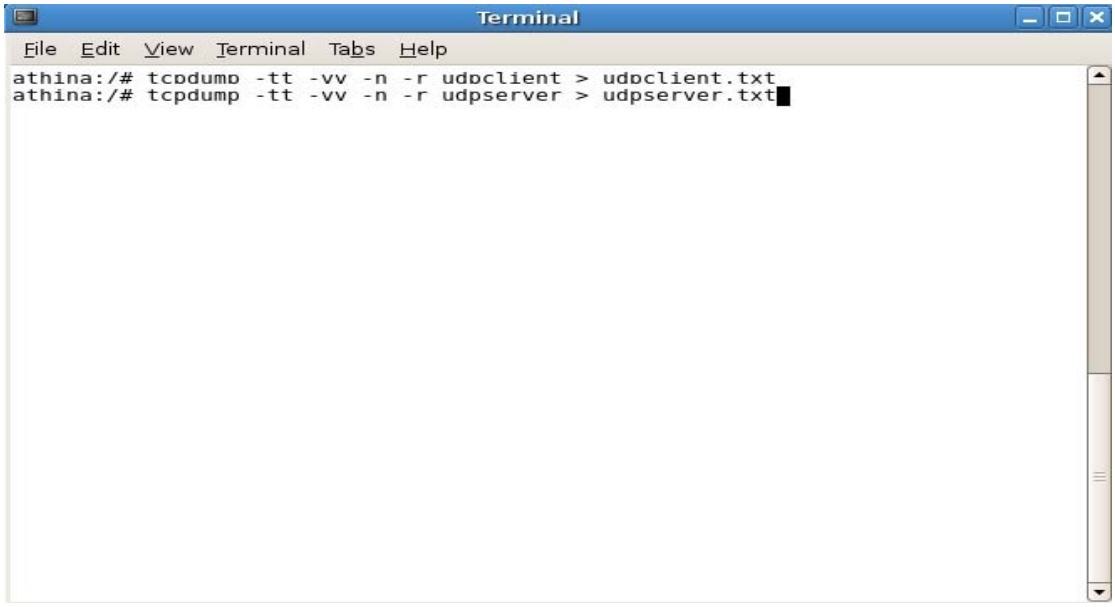
Σχ. 57 Εντολή Mgen στον Client (Δικτυακή κίνηση UDP)

Τελικά, έχουμε την δημιουργία μιας δικτυακής κίνησης από τον Server στον Client η οποία ανιχνεύεται και στους δύο υπολογιστές.

4.2.4.2 Ανάλυση δικτυακής κίνησης UDP

Στο επόμενο στάδιο έχουμε την ανάλυση αυτής κίνησης. Η ανάλυση πρέπει να γίνει σε υπολογιστή με λειτουργικό σύστημα Linux.

Παίρνουμε τα δύο αρχεία που δημιουργήθηκαν (S256k και C256k) και εκτελούμε με την εφαρμογή tcpdump. Οι εντολές του tcpdump που ακολουθούν στο σχήμα 58 παράγουν τα στατιστικά της κίνησης σε αρχεία της μορφής .txt.



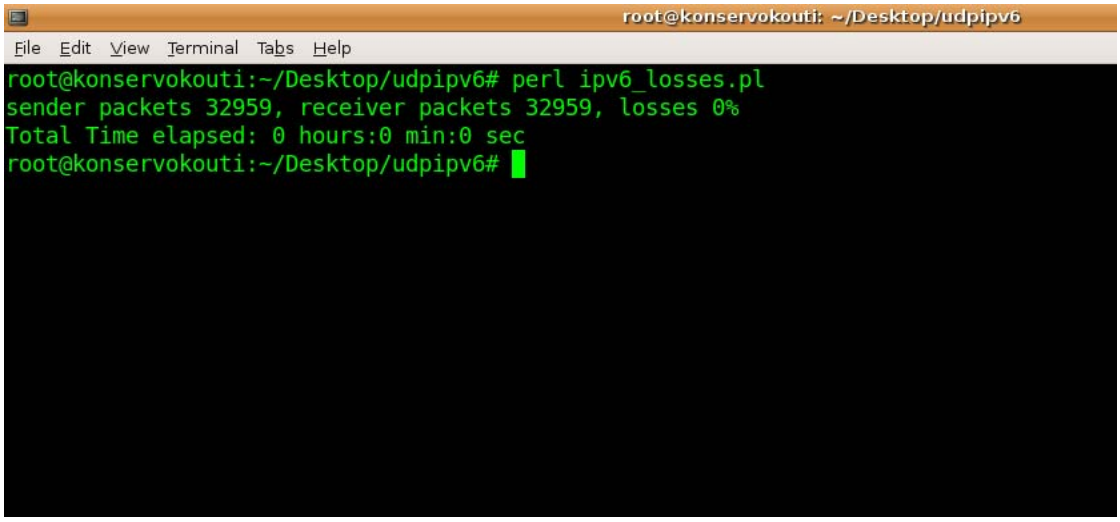
Σχ. 58 Εντολή tcpdump για τα στατιστικά των αρχείων S256k και C256k

Αφού έχουμε ολοκληρώσει και την δημιουργία των txt αρχείων μας ακολουθεί, η εκτέλεση των perl προγραμμάτων. Αρχικά εκτελούμε το createendfiles,

για τη δημιουργία δύο νέων αρχείων εκ των οποίων το ένα έχει την κίνηση του αποστολέα (sender) και το άλλο του δέκτη (receiver) και εκτελείται με την εντολή :

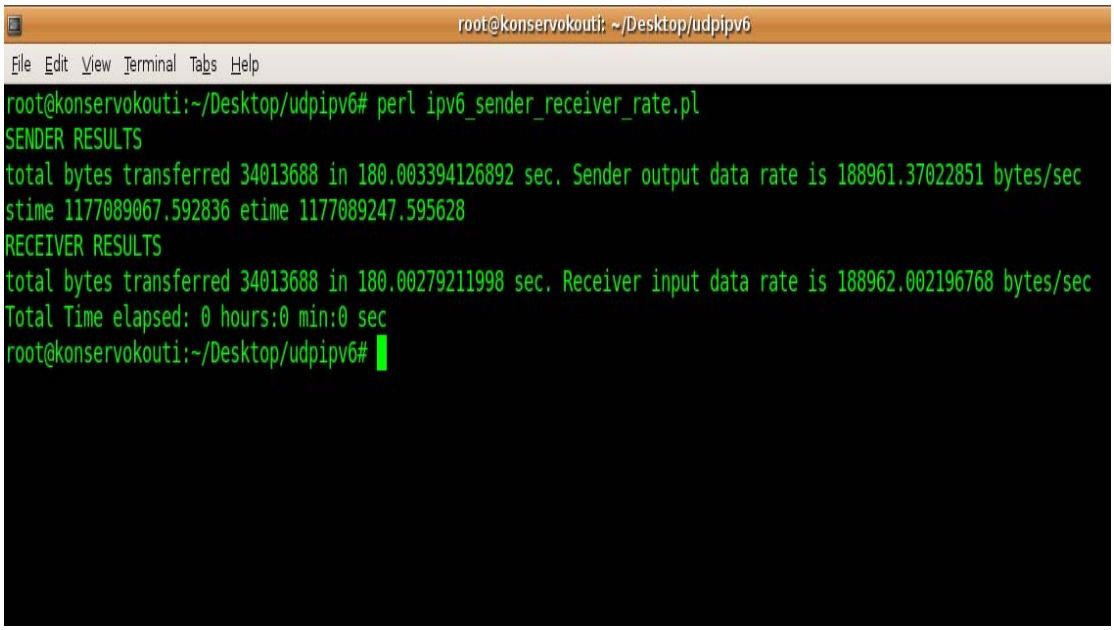
Perl ipv6_createendfiles.pl

Με το πρόγραμμα losses της perl, μπορούμε να βρούμε τις απώλειες που είχαμε κατά την ενεργό δράση του δικτύου και εκτελείται όπως φαίνεται στο σχήμα 59. Όπως παρατηρούμε στο δίκτυο μας διακινήθηκαν 32959 πακέτα και έχουμε 0% losses.



Σχ. 59 Perl ipv6_losses.pl

Με το επόμενο πρόγραμμα, το throughput της perl, μπορούμε να βρούμε το data rate του αποστολέα και του δέκτη αντίστοιχα και εκτελείται με την εντολή που φαίνεται στο σχήμα 60 :



Σχ. 60 Perl ipv6_sender_receiver_rate.pl

Ακολουθεί το πρόγραμμα align_for_delay_jitt της perl, όπου ευθυγραμμίζει κάθε πακέτο του αρχείου του δέκτη με κάθε πακέτο του αρχείου του αποστολέα, κι αυτό για να μπορέσουμε να πάρουμε αργότερα την διακύμανση της καθυστέρησης (jitter) και εκτελείται ως εξής:

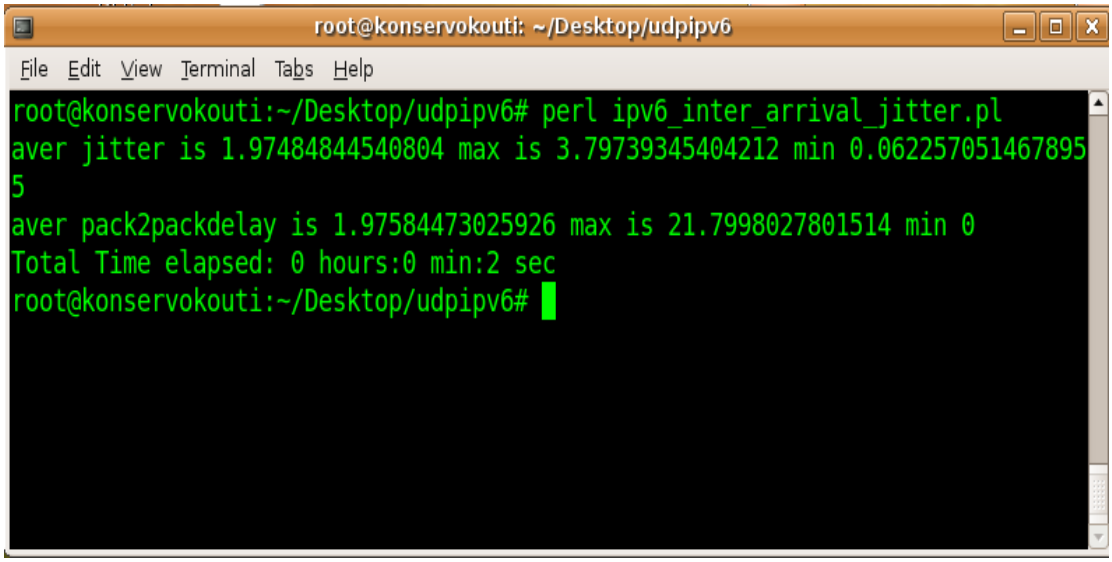
perl ipv6_align_for_delay_jitt.pl

Μετά την εκτέλεση της, βλέπουμε το πόσα πακέτα διακινήθηκαν στο δίκτυο μας από τον sender και από τον receiver. Στην μέτρηση που κάναμε διακινήθηκαν 32959 πακέτα από τον sender και 32959 πακέτα από τον receiver.

Με το πρόγραμμα timestamp της perl, έχουμε τη δημιουργία δύο αρχείων με τους χρόνους του αποστολέα και του δέκτη αντίστοιχα, για τον υπολογισμό του jitter και εκτελείται ως εξής:

perl ipv6_timestamp.pl

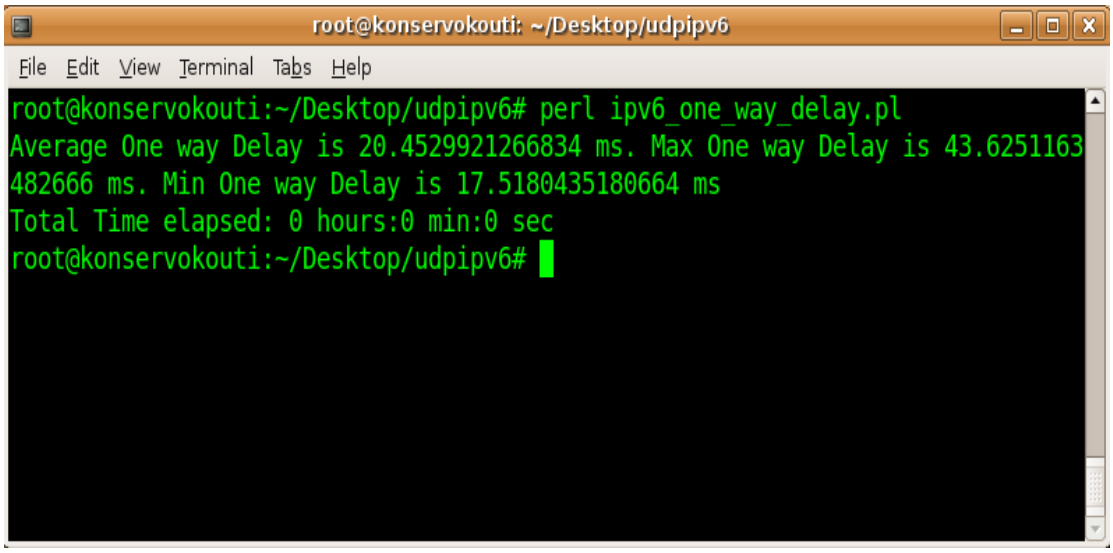
Στη συνέχεια με το πρόγραμμα inter_arival_jitter της perl υπολογίζεται η διακύμανση της καθυστέρησης (jitter) του δικτύου καθώς και η καθυστέρηση που είχαμε μεταξύ των πακέτων (packet to packet Delay). Η εκτέλεση της εντολής καθώς και τα αποτελέσματα που αυτή βγάζει παρουσιάζεται στο σχήμα 61 :



Σχ. 61 Perl ipv6_inter_arrival_jitter.pl

Παρατηρούμε ότι η εντολή αυτή μας εμφανίζει τρεις τιμές για το jitter και το packet to packet Delay. Αυτές κατά σειρά όπως εμφανίζεται στο σχήμα 61 είναι: η μέση τιμή, η μέγιστη τιμή και η ελάχιστη τιμή. Έτσι για το jitter έχουμε τις τιμές 1,98msec , 3,80msec και 0,06msec αντίστοιχα. Και οι τιμές για το pack2packdelay είναι 1,98msec, 21,80msec και 0msec.

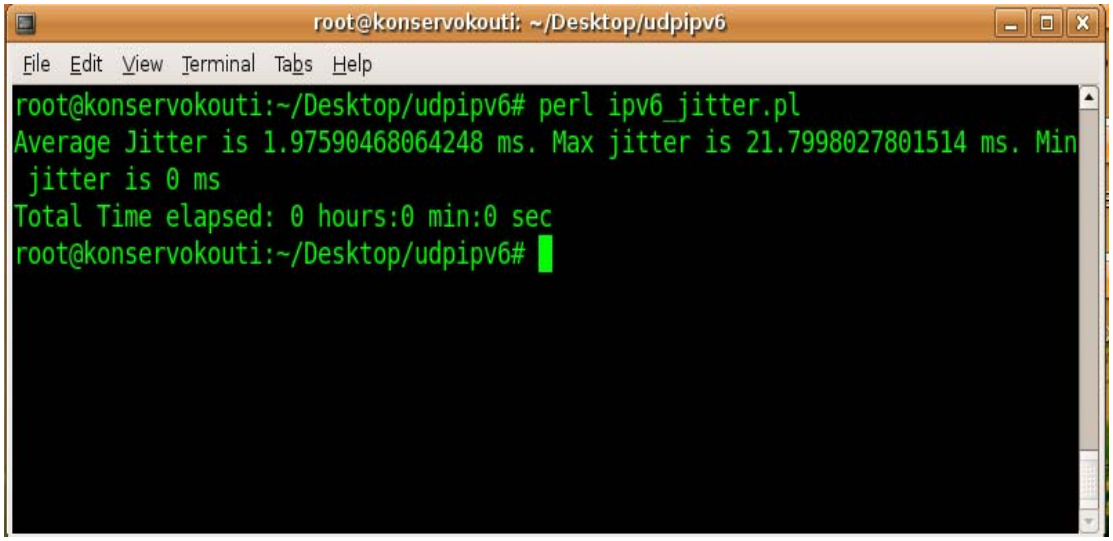
Με το πρόγραμμα one_way_delay παρουσιάζεται ο χρόνος που χρειάζεται ένα πακέτο για να μεταφερθεί από τον sender στον receiver. Ο τρόπος που εκτελείται και τα αποτελέσματα που εμφανίζει το πρόγραμμα αυτό παρουσιάζονται στο σχήμα 62 :



Σχ. 62 Perl ipv6_one_way_delay.pl

Όπως βλέπουμε μετά την εκτέλεση εμφανίζονται τρεις τιμές του one_way_delay. Η μέση, 20,45msec, η μέγιστη, 43,63msec, και η ελάχιστη τιμή, 17,52msec.

Τέλος με το πρόγραμμα ipv6_jitter.pl παίρνουμε τη διακύμανση της καθυστέρησης(jitter). Η εκτέλεση του προγράμματος φαίνεται στο σχήμα 63 :



Σχ. 63 Perl ipv6_ jitter.pl

Όπως παρατηρούμε από το σχήμα η εκτέλεση του προγράμματος μας εμφανίζει τρεις τιμές του jitter δηλαδή της καθυστέρησης στην αποστολή δύο διαδοχικών πακέτων. Την μέση τιμή, που είναι περίπου 1,98msec, την μέγιστη τιμή, η οποία είναι περίπου 21,80msec και την ελάχιστη τιμή, που είναι 0msec.

4.3 Πειραματικές μετρήσεις δικτύου επίγειας ψηφιακής τηλεόρασης με κανάλι επιστροφής τις τεχνολογίες ISDN και WLAN

4.3.1 Γενικά

Στο προηγούμενο κεφάλαιο έγινε μια λεπτομερή αναφορά πάνω στην αρχιτεκτονική του δικτύου επίγειας ψηφιακής τηλεόρασης με κανάλια επιστροφής ISDN και Wlan.

Παρακάτω αναφέρονται τα σενάρια που υιοθετήθηκαν πάνω στο προγραμματισμό των συγκεκριμένων πειραματικών μετρήσεων και για τα δύο είδη της δικτυακής κίνησης που δημιουργήθηκαν (TCP και UDP).

Επίσης γίνεται η ανάλυση και ο σχολιασμός των πειραματικών μετρήσεων με σκοπό την εξαγωγή χρήσιμων συμπερασμάτων για τις επιδόσεις των προαναφερθέντων δικτύων (σύμφωνα με τις προτεινόμενες αρχιτεκτονικές). Είναι αναμφισβήτητα το σημαντικότερο κομμάτι της πτυχιακής εξ'αιτίας του ότι κρίνεται η προσφορά της όσον αφορά τις επιδόσεις του προτεινόμενου δικτύου, σε σύγκριση με ένα δίκτυο που είναι ήδη σε λειτουργία. Στο διάστημα λοιπόν που διήρκησαν οι μετρήσεις(180sec) και σύμφωνα με τα σενάρια που αναφέρθηκαν προηγουμένως, ολοκληρώθηκαν αρκετές μετρήσεις για κάθε σενάριο έτσι ώστε να γίνουν όσο το δυνατόν αντιπροσωπευτικότερα τα δείγματα που παρουσιάστηκαν.

4.3.2. Μετάδοση δεδομένων κάνοντας χρήση του πρωροκόλλου TCP/IP

Το βασικό σενάριο είναι ότι ένας σταθερός χρήστης ζητάει υπηρεσίες internet από έναν Server, κάνοντας χρήση των δικτύων ISDN και WLAN ως κανάλια επιστροφής και του δικτύου της ψηφιακής τηλεόρασης ως κανάλι καθόδου. Η δικτυακή κίνηση από τον server προς τον χρήστη και αντίστροφα, γίνεται με τη βοήθεια ενδιάμεσων κόμβων διανομής (CMNs), όπως περιγράφηκε και στην αρχιτεκτονική του δικτύου, στο προηγούμενο κεφάλαιο.

Η συμπεριφορά και οι επιδόσεις του TCP μετρήθηκαν σε μια μαζική μεταφορά δεδομένων (TCP bulk data transfer) από τον data server στον υπολογιστή του χρήστη και αντίστροφα, μέσω του προγράμματος Iperf. Το μετρητικό πρόγραμμα πραγματοποιεί μεταφορά εικονικών δεδομένων (dummy data) από έναν υπολογιστή σε έναν άλλον για μια συγκεκριμένη χρονική διάρκεια (180 δευτερόλεπτα στη δοκιμή που περιγράφηκε) με σκοπό να προσδιοριστούν οι επιδόσεις και η συμπεριφορά των πρωτοκόλλων δικτύου πάνω από τα συγκεκριμένα κανάλια. Κατά τη διάρκεια μεταφοράς των δεδομένων, τα πακέτα που ανταλλάσσονται στις διεπαφές δικτύου του server και του client καταγράφονται από ένα πρόγραμμα σύλληψης πακέτων το Tcpdump και οι κεφαλίδες τους αποθηκεύονται σε τοπικά dump files. Τα dump files αναλύονται εκ των υστέρων από τον protocol analyzer TCPtrace προκειμένου να εξαχθούν πληροφορίες σχετικά με τον ρυθμό διαμεταγωγής (thoughput) και τους χρόνους πλήρους διαδρομής (round-trip times). Οι μετρήσεις πραγματοποιήθηκαν για διαφορετικές τιμές στο window size του χρήστη. Οι τιμές που χρησιμοποιούμε είναι: 8KByte, 32KByte, 64KByte και 512KByte. Αυτο που θέλουμε να μελετήσουμε είναι το αν και κατά πόσο βελτιστοποιείται το δίκτυο μας. Παρ' όλα αυτά, η καθυστέρηση που υπεισέρχεται τόσο από τα κανάλια επιστροφής όσο και από τις μονάδες επεξεργασίας του καναλιού εκπομπής, σε συνδυασμό με τον υψηλό ρυθμό δεδομένων

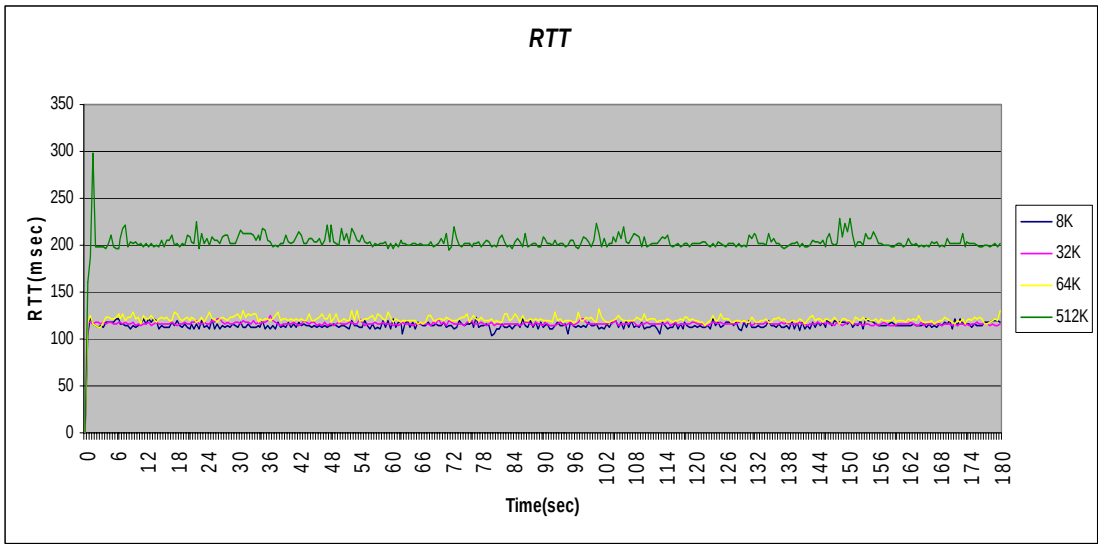
του DVB-T μπορούν μερικές φορές να προκαλέσουν σχετικά μεγάλα γινόμενα καθυστέρησης-εύρους ζώνης (bandwidth-delay products – BDP) στο δίκτυο. Συνεπώς είναι σχεδόν πάντα απαραίτητο να προσαρμοστεί το μέγιστο παράθυρο λήψης (TCP receive window) στο τερματικό του server για να επιτευχθούν οι βέλτιστες δυνατές επιδόσεις. Στη συγκεκριμένη μέτρηση το window size του server ήταν 262144.

4.3.2.1 Γραφικές παραστάσεις δικτυακής κίνησης TCP

Τα αποτελέσματα της TCP δικτυακής κίνησης δίνονται μέσα από τις γραφικές παραστάσεις που ακολουθούν:
Οι αναμεταδόσεις πακέτων (Retransmissions) ήταν μηδενικές.

4.3.2.1.1 Round-trip Time

Το round-trip time είναι ο συνολικός χρόνος που διαρκεί η κίνηση της αποστολής από τον server ενός πακέτου και της παραλαβής της επιβεβαίωσης(acknowledgment) από τον χρήστη ότι παραλήφθηκε επιτυχώς το πακέτο. Στο σχήμα 64 ακολουθεί η γραφική παράσταση, στην οποία παρουσιάζονται συνδυαστικά οι τέσσερις τιμές του window size του client :

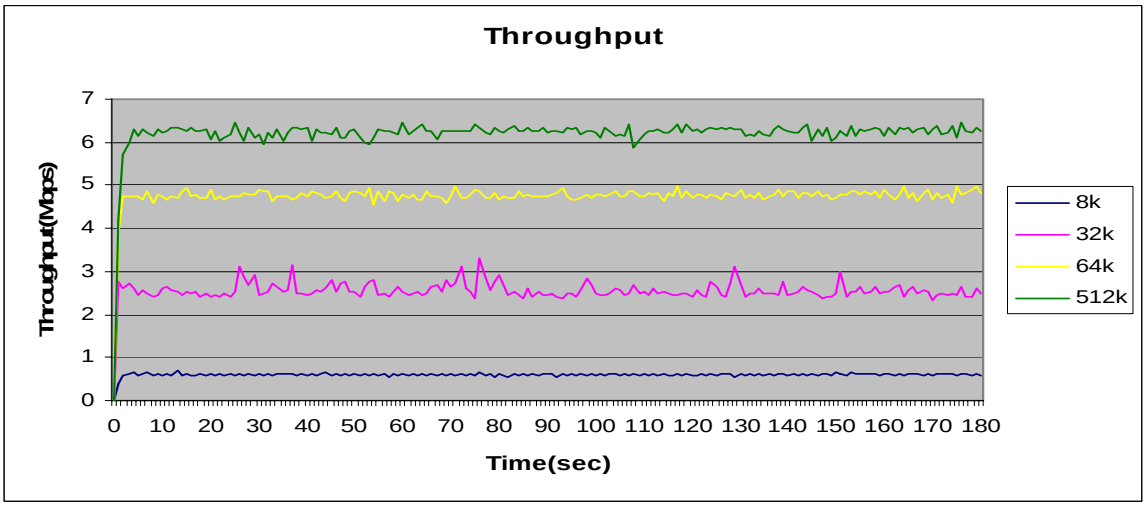


Σχ. 64 Γραφική παράσταση του Round-trip Time

Όπως παρατηρούμε από τη γραφική παράσταση, το Round-trip Time αυξάνεται καθώς αυξάνουμε το window size στον client. Αυτό είναι λογικό αφού όσο αυξάνεται το window size στο δίκτυο μας διακινούνται περισσότερα πακέτα και επιβεβαιώσεις. Έτσι σαν αποτέλεσμα έχουμε ότι τα πακέτα που φτάνουνε στον Amber είναι περισσότερα για μεγαλύτερο window size με αποτέλεσμα να καθυστερεί η διαδικασία του encapsulation, με αποτέσμα να μεγαλώνει το συνολικό Round-trip Time.

4.3.2.1.2 Throughput

Είναι ο ρυθμός με τον οποίο τα δεδομένα μεταδίδονται από τον server στον χρήστη. Η γραφική παράσταση που ακολουθεί στο σχήμα 65 δείχνει τις μετρήσεις που κάναμε:

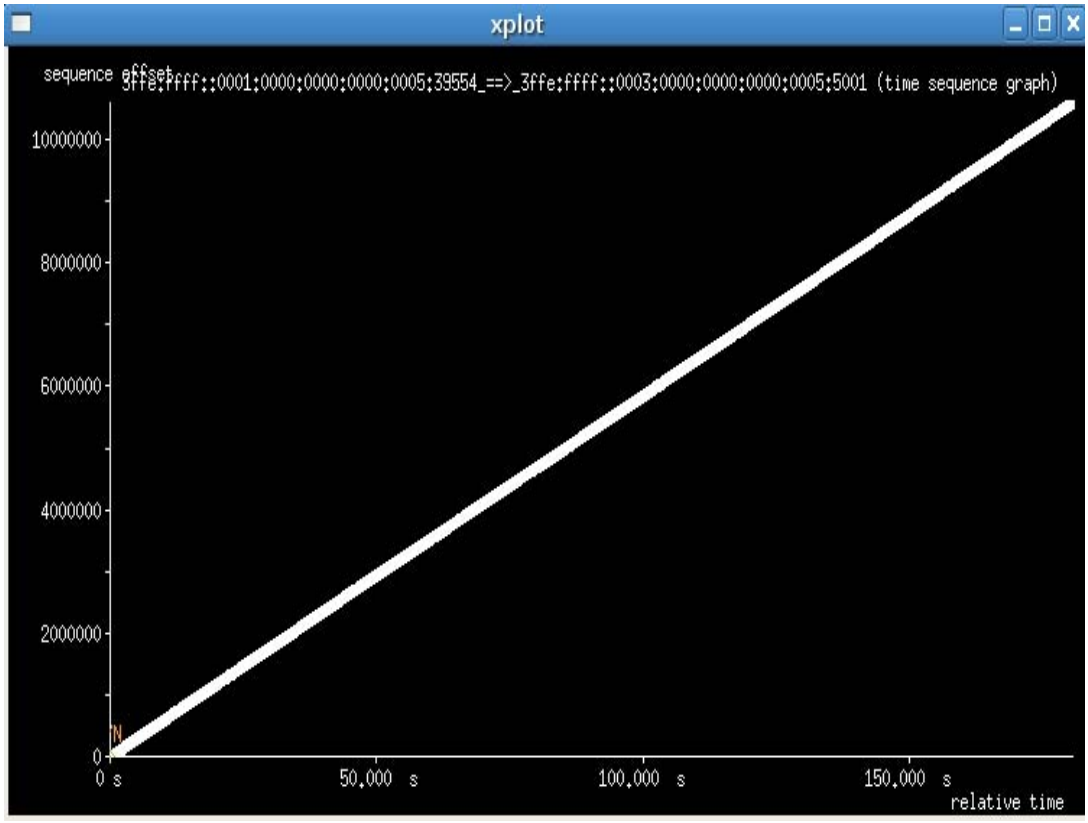


Σχ. 65 Γραφική παράσταση του Throughput

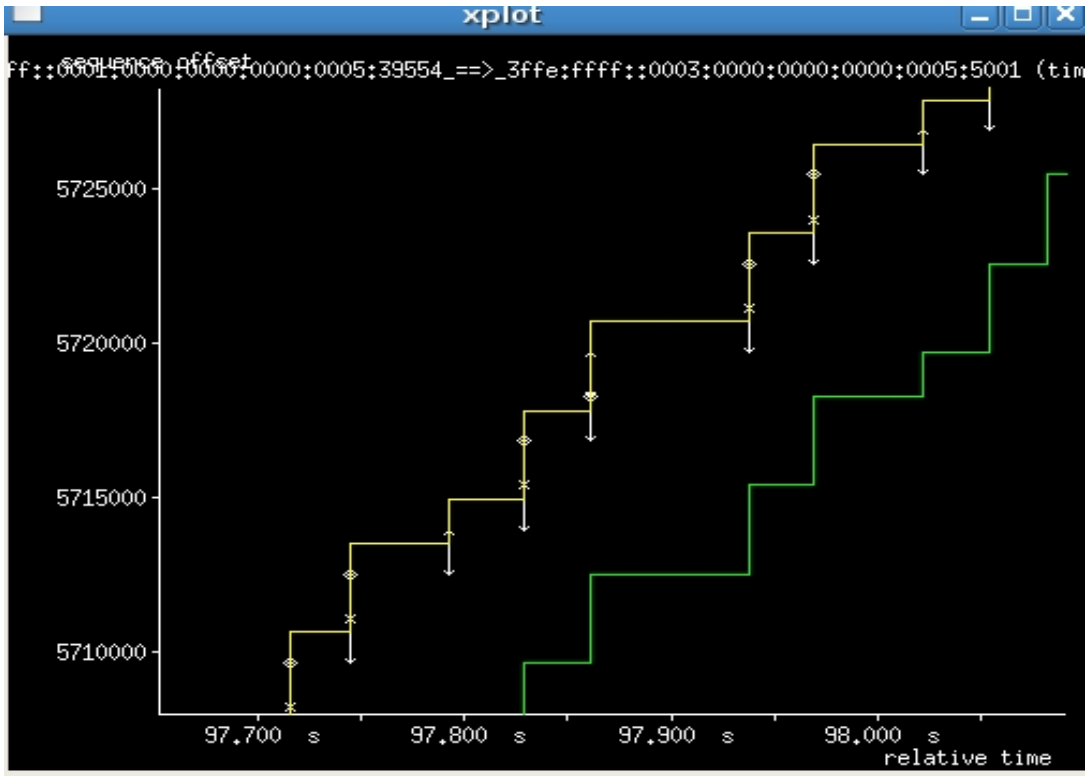
Όπως παρατηρούμε από το παραπάνω γράφημα, ο ρυθμός μετάδοσης του δικτύου είναι μεγαλύτερος για 512KByte window size στον client. Το συμπέρασμα που μπορούμε να βγάλουμε είναι ότι το δίκτυο μας αποδίδει καλύτερα όσο μεγαλώνουμε το window size στο τερματικό του χρήστη.

4.3.2.1.3 Time Sequence

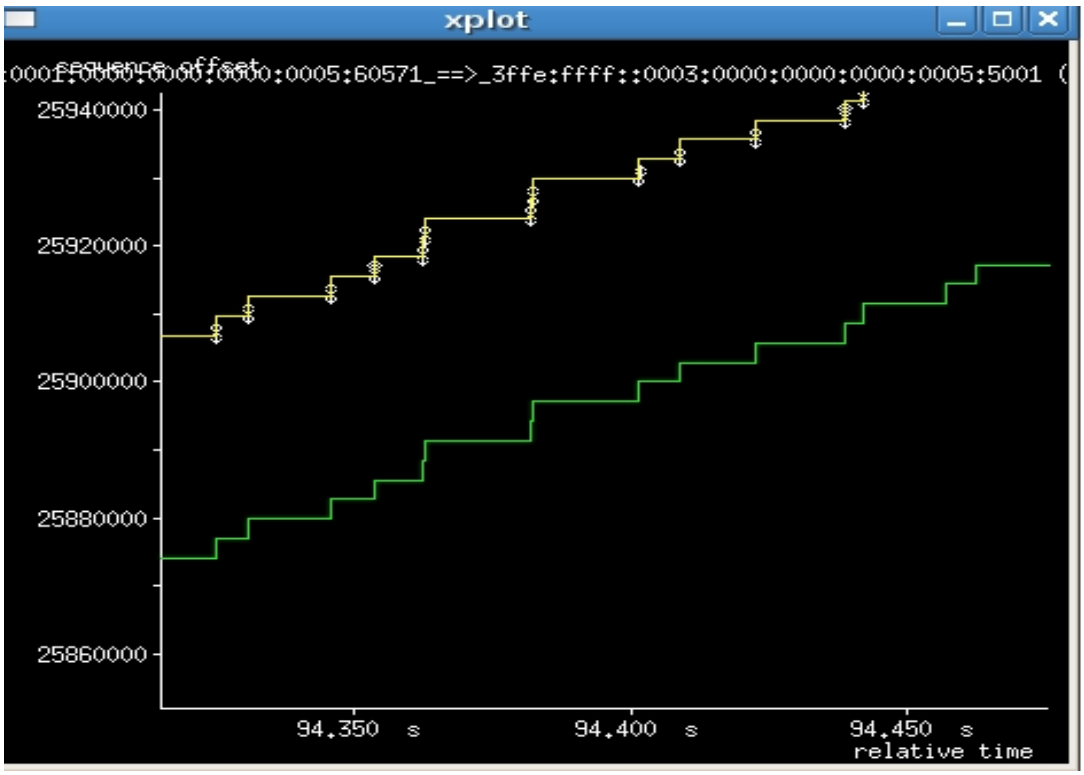
Στα παρακάτω σχήματα παρουσιάζονται οι γραφικές παραστάσεις των μετρήσεων που γίνανε για τα τέσσερα window size του client: Το πρώτο σχήμα είναι το αρχικό που βγαίνει μετά τις εντολές. Στα επόμενα τέσσερα έχει γίνει zoom ώστε να μπορέσουμε να τα αναλύσουμε.



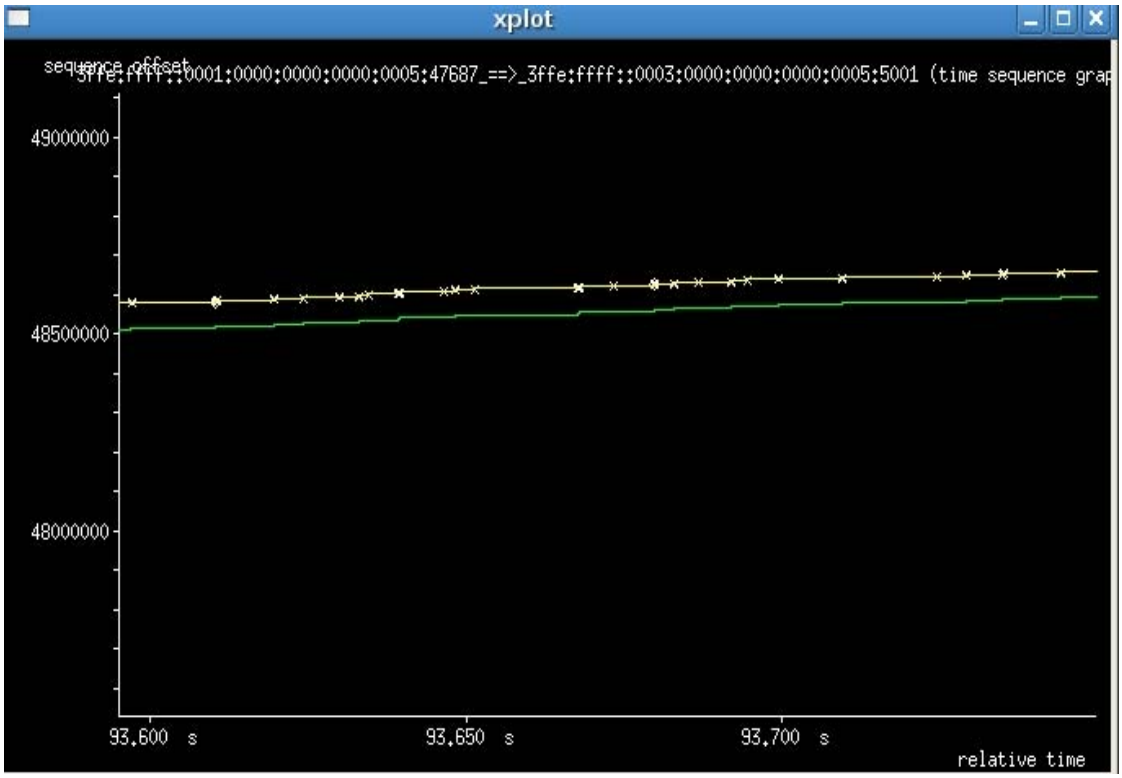
Σχ. 66 Γραφική παράσταση του Time sequence στα 8KByte



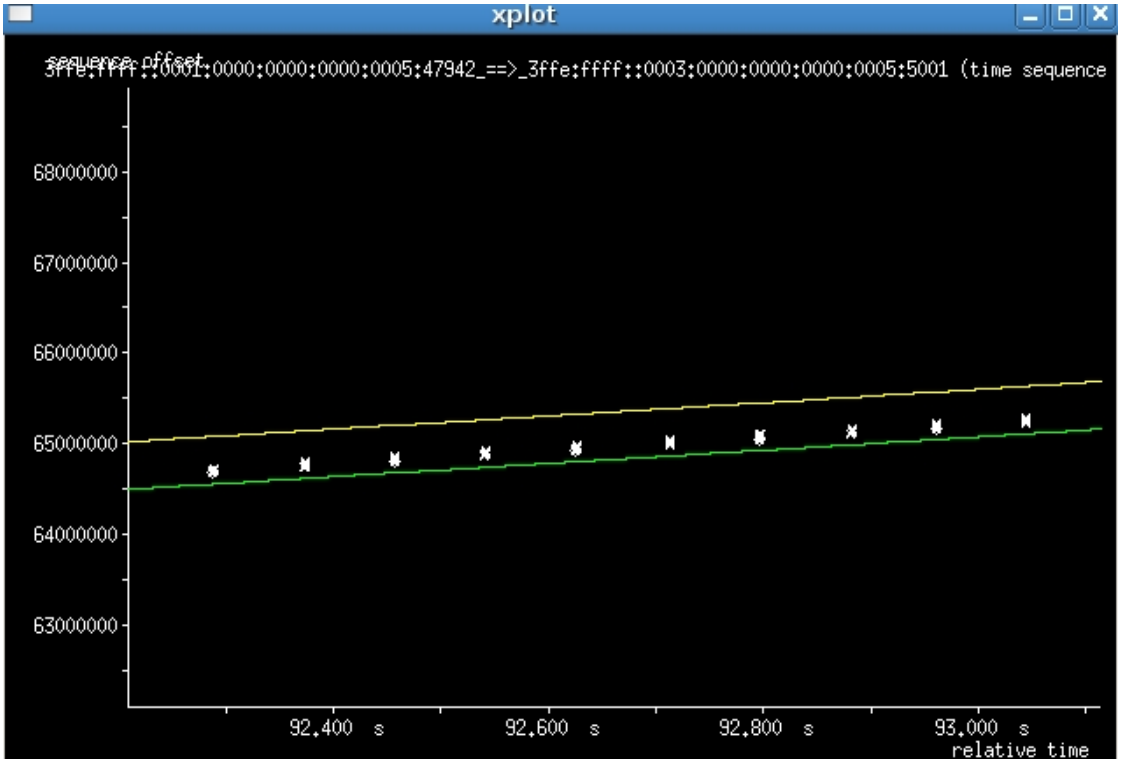
Σχ. 67 Γραφική παράσταση(με zoom) του Time sequence στα 8KByte



Σχ. 68 Γραφική παράσταση(με zoom) του Time sequence στα 32KByte



Σχ. 69 Γραφική παράσταση(με zoom) του Time sequence στα 64KByte



Σχ. 70 Γραφική παράσταση(με zoom) του Time sequence στα 512KByte

Στα παραπάνω γραφήματα φαίνεται η εξέλιξη του αριθμού ακολουθίας(sequence number) των πακέτων TCP στο πέρασμα του χρόνου για συγκεκριμένο χρονικό διάστημα που επιλέξαμε. Όπως παρατηρούμε δεν έχουμε απώλειες πακέτων και αναμεταδόσεις στο δίκτυο μας. Τα λευκά βέλη αναπαριστούν την αρχή και το τέλος του κάθε πακέτου δεδομένων που στέλνονται σε κάθε χρονική στιγμή, ενώ η κάτω γραμμή, πράσινη, δείχνει τον αριθμό ακολουθίας των δεδομένων για τα οποία μέχρι τώρα έχει φτάσει επιβεβαίωση(acknowledged data). Η άνω γραμμή, κίτρινη, υπολογίζεται αν προστεθεί στην γραμμή επιβεβαιώσεων(πράσινη) το μέγεθος παραθύρου του client(window size) και ουσιαστικά δείχνει το μέγιστο όριο της απόστασης-διαφοράς μεταξύ του αριθμού ακολουθίας του τελευταίου επιβεβαιωμένου πακέτου και αυτού που μόλις εκπέμφθηκε. Η διαφορά αυτή ισούται με το ποσό των δεδομένων που έχουν σταλεί και βρίσκονται ανεπιβεβαίωτα στο δίκτυο.

4.3.3 Μετάδοση δεδομένων κάνοντας χρήση του πρωροκόλλου UDP/IP

Το βασικό σενάριο είναι ότι ένας σταθερός χρήστης ζητάει υπηρεσίες internet από έναν Server, κάνοντας χρήση των δικτύων ISDN και WLAN ως κανάλια επιστροφής και του δικτύου της ψηφιακής τηλεόρασης ως κανάλι καθόδου. Η δικτυακή κίνηση από τον server προς τον χρήστη και αντίστροφα, γίνεται με τη βοήθεια ενδιάμεσων κόμβων διανομής (CMNs), όπως περιγράφηκε και στην αρχιτεκτονική του δικτύου, στο προηγούμενο κεφάλαιο.

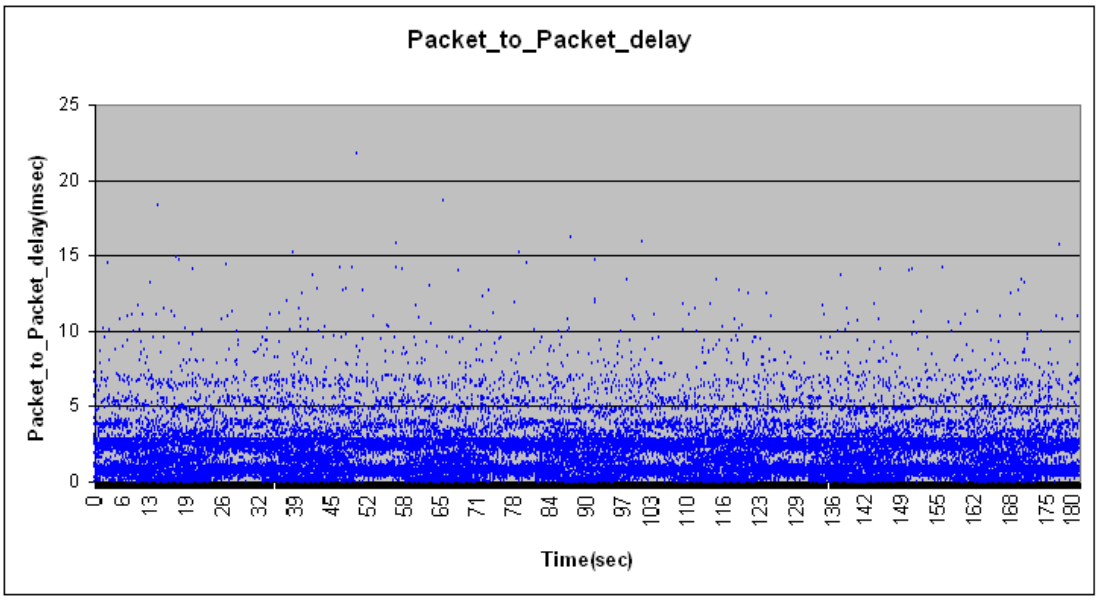
Η συμπεριφορά και οι επιδόσεις του UDP μετρήθηκαν και αναλύθηκαν στη συνέχεια με τη βοήθεια κάποιων προγραμμάτων. Τα προγράμματα που χρησιμοποιήθηκαν είναι το Mgen για τη δημιουργία της κίνησης και το tcpdump το οποίο είναι ένα πρόγραμμα σύλληψης πακέτων. Η κίνηση των πακέτων καταγράφηκε για συγκεκριμένο χρονικό διάστημα, 180 δευτερόλεπτα. Η ανάλυση των αρχείων που παίρνουμε δημιουργώντας την κίνηση γίνεται με τη βοήθεια του TCPtrace. Στη συνέχεια με τη βοήθεια script που έχουνε φτιαχτεί στη γλώσσα προγραμματισμού Perl, εξάγονται οι γραφικές παραστάσεις που παρουσιάζονται παρακάτω. Οι πληροφορίες που θα εξαχθούν αφορούν το Packet_to_Packet_delay, το Jitter και το one_way_delay. Οι εντολές των προγραμμάτων και η εκτέλεση των script που χρησιμοποιήθηκαν έχουνε ήδη παρουσιαστεί αναλυτικά στο κεφάλαιο 4.2.4.

4.3.3.1 Γραφικές παραστάσεις δικτυακής κίνησης UDP

Τα αποτελέσματα της UDP δικτυακής κίνησης δίνονται μέσα από τις γραφικές παραστάσεις που ακολουθούν:
Οι απώλειες του δικτύου (losses) είναι μηδενικές. Και ο ρυθμός μετάδοσης των δεδομένων (data rate) υπολογίστηκε γύρω στα 1.49 Mbps, δηλαδή πλησίασε κατά πολύ το 1.5 Mbps του προγράμματος IPTV που έδινε ο OPAL.

4.3.3.1.1 Packet_to_Packet_delay

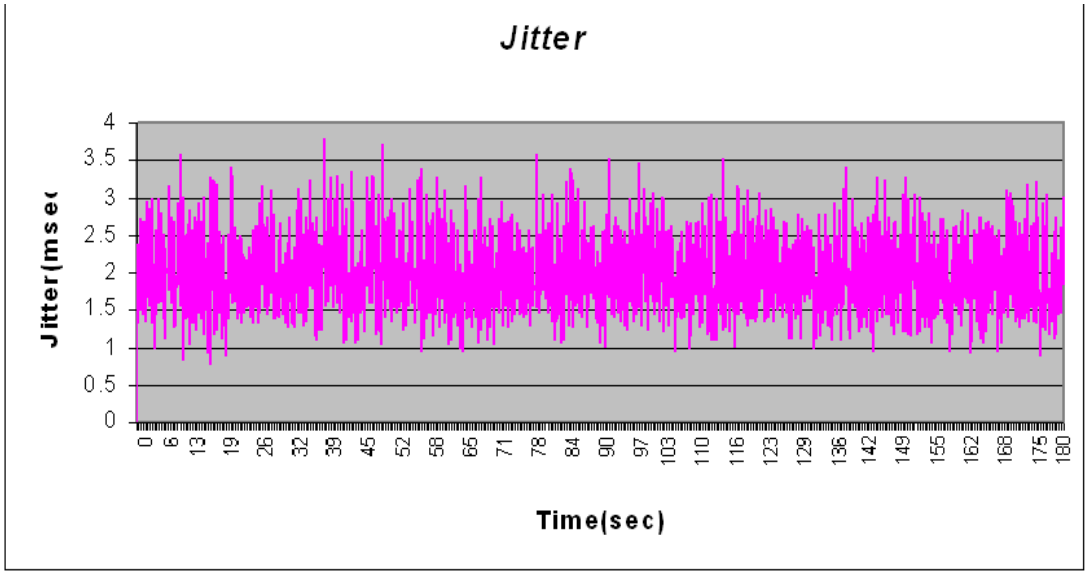
Στο σχήμα 71 παρουσιάζεται η γραφική παράσταση του Packet2Packet_delay.



Σχ. 71 Γραφική παράσταση του Packet to Packet Delay

4.3.3.1.2 JITTER

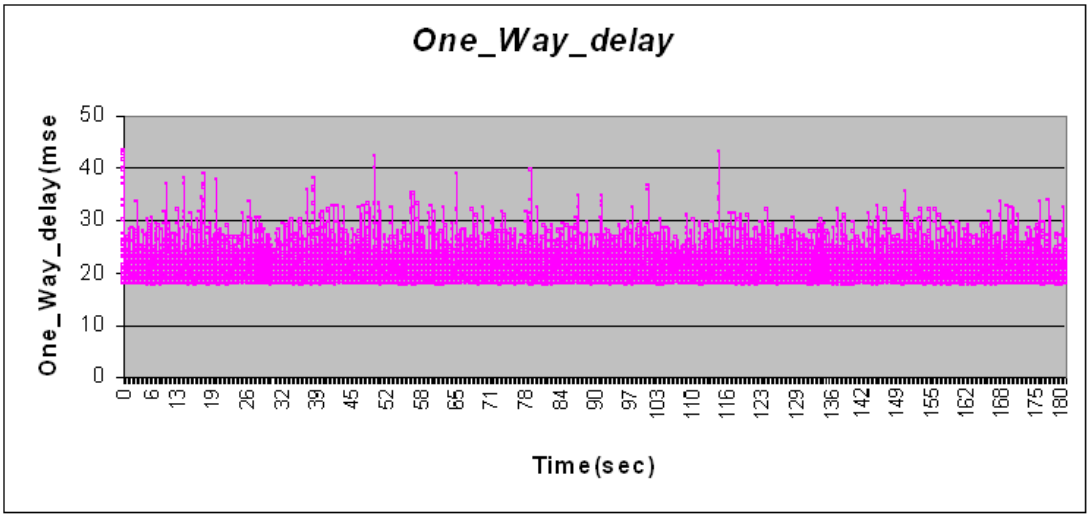
Η γραφική παράσταση του Jitter παρουσιάζεται στο σχήμα 72.



Σχ. 72 Γραφική παράσταση του Jitter

4.3.3.1.3 One_Way_delay

Στο σχήμα 73 δίνεται η γραφική παράσταση του one_way_delay:



Σχ. 73 Γραφική παράσταση του One_Way_delay

4.3.3. 1 Ανάλυση ποιότητας UDP κίνησης

Σύμφωνα με τα πρότυπα ποιότητας της Cisco, ένα Streaming Video πρέπει να πληρρει τις παρακάτω προϋποθέσεις για θεωρείται η ποιότητα του ικανοποιητική:

- Οι απώλειες του (losses) δεν πρέπει να είναι μεγαλύτερες από 5 τις εκατό.
- Η καθυστέρηση δεν θα πρέπει να ξεπερνάει τα 5 με 6 δευτερόλεπτα (αυτό εξαρτάται κυρίως από την δυνατότητα της video εφαρμογής να κάνει buffer).
-Όσον αφορά το Jitter δεν έχουν διατυπωθεί συγκεκριμένες απαιτήσεις.

Συγκρίνοντας τα αποτελέσματα που πήραμε από τις γραφικές παραστάσεις καταλήγουμε στο συμπέρασμα ότι το δίκτυο μας είναι ικανοποιητικό για video streaming, αφού η τιμή της καθυστέρησης είναι 20,5 msec (πολύ πιο χαμηλή από τα standards), και τέλος οι απώλειες μας είναι μηδενικές.

5. Συμπεράσματα και μελλοντικές προτάσεις

Η πτυχιακή εργασία εστίασε την έρευνα πάνω στη χρήση της σύγχρονης και προτυποποιημένης τεχνολογίας της επίγειας ψηφιακής τηλεόρασης (DVB-T) ως ευρυζωνικό δίκτυο πρόσβασης για υπηρεσίες δεδομένων. Μελετήθηκε μία υβριδική αρχιτεκτονική ασύμμετρου δικτύου σε δύο διαφορετικές παραλαγές, στις οποίες η κατερχόμενη ζεύξη (downlink) υλοποιήθηκε με ένα ευρυζωνικό κανάλι DVB-T ενώ η ανερχόμενη (uplink) παρεχόταν από την ενσύρματη τεχνολογία ISDN και WLAN. Σημαντικό ρόλο στην αρχιτεκτονική των δικτύων είχαν οι ενδιάμεσοι κόμβοι διανομής (CMNs). Εξετάστηκε και αξιολογήθηκε η συμπεριφορά του ενός δικτύου σε μεταδόσεις από σημείο σε σημείο κάνοντας χρήση των πρωτοκόλλων tcp/ip και udp/ip.

Όσον αφορά τα αποτελέσματα των μετρήσεων, κρίνονται αναμενόμενα με βάση την αρχιτεκτονική και τη λειτουργία του δικτύου επίγειας ψηφιακής τηλεόρασης με κανάλι επιστροφής την τεχνολογία PSTN. Παρόλο που πρόκειται για μια τεχνολογία με σημαντικά μειονεκτήματα, ο συνδιασμός της με την τεχνολογία DVB-T είναι αποτελεσματικός και δημιουργείται μια αρχιτεκτονική παροχής πολλαπλών Διαδικτυακών υπηρεσιών και υπηρεσιών πολυμέσων.

Το ερευνητικό αντικείμενο που μελετήθηκε στην παρούσα πτυχιακή εργασία μπορεί να επεκταθεί έτσι ώστε να αποτελέσει αντικείμενο μελλοντικής έρευνας στο τομέα της σύγκλισης των τεχνολογιών ευρυ-εκπομπής και τηλεπικοινωνιών. Σχετικά με το κανάλι επιστροφής τεχνολογίας ISDN που αξιοποιήθηκε στο προτεινόμενο δίκτυο, περαιτέρω διερεύνηση μπορεί να επεκταθεί και να περιλάβει και άλλες ασύρματες και ενσύρματες τεχνολογίες ανάλογα με το είδος του τελικού χρήστη (κινητός, σταθερός).

Ακρωνύμια

A

AAL ATM Adaption Layer

ACK Acknowledgement packet

ADSL Asymmetric Digital Subscriber Line

AIX Advanced Interactive eXecutive

AMD Advanced Micro Devices

AoD Audio on Demand

API Application Program(ming) Interface

ATM Asynchronous Transfer Mode

B

BDP Bandwidth Delay Products

BSD Berkeley Software Distribution

C

COFDM Coded Orthogonal Frequency Division Multiplexing

CMN Cell Main Node

D

DSL Digital Subscriber Line

DSLAM Digital Subscriber Line Access Multiplexer

DVB-C Digital Video Broadcasting-Cable

DVB-H Digital Video Broadcasting-Handover

DVB-S Digital Video Broadcasting-Satellite

DVB-T Digital Video Broadcasting-Terrestrial

DVD Digital Video Decoder

E

ES Elementary Stream

ETSI European Telecommunications Standard Institute

F

FEC Forward Error Correction

G

GOP Group Of Pictures

GPRS General Packet Radio Service

GSM Global System For Mobile Communications

H

HDTV High Definition Television

HP-UX Hewlett Packard UniX

I

IEEE Institute of Electrical & Electronics Engineers
IFFT Inverse Fast Fourier Transform
IP Internet Protocol
IPTV Internet Protocol Television
ISDB-T International Standard Digital Broadcasting-Terrestrial
ISDN Integrated Services Digital Network
ISM Industrial, Scientific, Medical
ISP Internet Service Provider
ITU-R International Telecommunications Union – Radio Communication

J

JPEG Joint Photographic Experts Group

M

MAC Media Address Controler
MAC OS X Macintosh Operating System X
MHP Multimedia Home Platform
MPE Multi Protocol Encapsulation
MPEG Motion Pictures Experts Group
MSS Maximum Segment Size
MTU Maximum Transmission Unit

N

NAT Network Address Translation
NRL Naval Research Laboratory
NTSC National TV Standards Committee

O

OPAL Omni-Purpose Apparatus at LEP

P

PAT Programm Association Table
PCR Program Clock Reference
PES Packetised Elementary Stream
PID Programm Identifier
PMT Programm Map Table
POSIX Portable Operating System Interface for uniX
PS Program Stream
PSTN Public Switched Telephone Network

Q

QAM Quadrature Amplitude Modulation

QoS Quality of Service

QPSK Quadrature Phase Shift Keying

R

RAM Random-Access Memory

RDVB-T Regenerative Digital Video Broadcasting-Terrestrial

RS Reed Solomon

RTO Retransmission Time Out

S

SFN Single Frequency Networks

SNR Signal-to-noise ratio

T

TCP Transmission Control Protocol

TPS Transactions Per Second

TS Transport Stream

TV Television

U

UDP User Datagram Protocol

UHF Ultra High Frequency

ULE Ultra-Light Encapsulation

UMTS Universal Mobile Telecommunications System

V

VCR Video Cassette Recorder

VLC VideoLAN Client

VoD Video on Demand

W

WLAN Wireless Lan

WEP Wired Equivalent Privacy

ΠΑΡΑΡΤΗΜΑ Α

1. Πρόγραμμα ipv6_mgen_createedfiles.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.

#!/usr/bin/perl -w

open(INPUTFILE, "<udpserver.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);
   $id=$line[21].$line[23];
   $size=$line[8];
   $time=$line[0];
   $id=substr($id,0,length($id));
   $size=substr($size,0,length($size)-1);
#print ("$id $size \n");
 print OUTFILE ("$id\t$id\t$time\t$size\n");
```

```
}

close(INPUTFILE);
close(OUTPUTFILE);

#second file

open(INPUTFILE, "<udpclient.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);
    $id=$line[21].$line[23];
   $size=$line[8];
   $time=$line[0];
   $id=substr($id,0,length($id));
   $size=substr($size,0,length($size)-1);

 print OUTFILE ("$id\t$id\t$time\t$size\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);
```

2. Πρόγραμμα ipv6_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 irons
#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 ($lock1==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]))
{
 $lock1=1;
    if ($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";
}
```

3. Πρόγραμμα ipv6_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
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA
02111-1307, USA.

#!/usr/bin/perl -w


my($sum_pack_size);
my($lock);
my($start_time);
my($end_time);
my($transffer_time);
my($data_rate);
$data_rate=0;
$transffer_time=0;
$sum_pack_size=0;
$lock=0;
$start_time=0;
$end_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";
$transffer_time=$end_time-$start_time;
$data_rate=$sum_pack_size/$transffer_time;
print "SENDER RESULTS\n";
print "total bytes transferred $sum_pack_size in
$transffer_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;
$transffer_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";
$transffer_time=$end_time-$start_time;
$data_rate=$sum_pack_size/$transffer_time;
print "RECEIVER RESULTS\n";
print "total bytes transferred $sum_pack_size in
$transffer_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";
```

4. Πρόγραμμα ipv6_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.

#!/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);
```

5. Πρόγραμμα ipv6_timestamp.pl

```
    #!/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";
```

6. Πρόγραμμα ipv6_inter_arrival_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.

#!/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]);
 $temp_time1=$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);

        $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. Πρόγραμμα ipv6_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.

        #!/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 Di=abs(R_i-S_i)
#Avg One way Delay is Sum(Di)/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";
```

8. Πρόγραμμα ipv6_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.

#!/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 Di=abs(R_(i)-R_(i-1))-(S_(i)-S_(i-1))
#Avg jitter is Sum(Di)/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";
```

9. Πρόγραμμα ipv6_throughput για την ανάλυση TCP κίνησης

```
#/*************************************************************
***************
# *            ipv6_throughput.pl
# *
# *  Thu Apr  19 19:16:18 2007
# *  Copyright  2007  irons
# *  Email irons@pasiphae.teiher.gr
#
*************************************************************
**************/
# 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.

#!/usr/bin/perl -w


my ($through_file)="ipv6_throughput.dat";
my ($interval_time)=$ARGV[1];

$numArgs = $#ARGV+1;
#print ("$numArgs\n");
if ($numArgs==2)
{
open(INPUTFILE, "<$ARGV[0]") || die ("cannot open file
$ARGV[0]. Does it exist?\n");

unless (open (OUTFILE, ">$through_file"))
{
```

```
    die ("cannot open output file $through_file\n");

}

my (@input_lines) = <INPUTFILE>;
my ($point_time)=0;
my ($size)=0;
my ($sum_size)=0;
my ($time)=0;
my ($temp_time)=0;
my ($temp_stop_time)=0;
my ($lock)=0;
my (@last_line)=0;
my ($throughput)=0;
my ($counter)=0;
print OUTFILE ("$time\t$size\n");

@last_line=split(/[\t +]/,$input_lines[@input_lines-1]);
for ($i=0;$i< @input_lines;$i++)
{



    if ($time>($point_time+$interval_time))
    {

    $point_time=$point_time+$interval_time;
    $counter--;
    #print ("$point_time $sum_size $counter\n");
    if ($counter!=0)
    {
    $throughput=$sum_size/$counter;

    print OUTFILE ("$point_time\t$throughput\n");
    }
    $sum_size=0;
    $counter=0;
    }
    if (($last_line[0]-$point_time)<=$interval_time)
    {

          $lock=1;

    }
    $sum_size=$sum_size+$size;

    $in_line=$input_lines[$i];
    chomp($in_line);
    @line=split(/[\t +]/,$in_line);
    $size=$line[1];
    $time=$line[0];
    $counter++;

 #print OUTFILE ("\t$time\t$size\n");

}

#print ("\n$lock\n");
if ($lock==1)
{
    if ($counter!=0)
```

```
    {

    $sum_size=$sum_size+$size;
    #print ("$time $sum_size $counter\n");

    $throughput=$sum_size/$counter;
    print OUTFILE ("$time\t$throughput\n");
    }
    $lock=0;
}
#print ("$last_line[1]\n");
close(INPUTFILE);
close(OUTPUTFILE);
}
else {
    print("dose sosta ta orismata (syntax:perl
ipv6_throughput.pl <filename> <time interval in sec>).\n");
}
```

10. Πρόγραμμα ipv6_rtt.pl για την ανάλυση TCP κίνησης

```
#/*************************************************************
***************
# *            ipv6_rtt.pl
# *
# *  Sat Apr  21 19:16:18 2007
# *  Copyright  2007  irons
# *  Email irons@pasiphae.teiher.gr
#
*************************************************************
**************/
# 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.

#!/usr/bin/perl -w


my ($rtt_file)="ipv6_rtt.dat";
my ($interval_time)=$ARGV[1];

$numArgs = $#ARGV+1;
#print ("$numArgs\n");
if ($numArgs==2)
{
open(INPUTFILE, "<$ARGV[0]") || die ("cannot open file
$ARGV[0]. Does it exist?\n");

unless (open (OUTFILE, ">$rtt_file"))
{

     die ("cannot open output file $rtt_file\n");

}

my (@input_lines) = <INPUTFILE>;
my ($point_time)=0;
my ($rtt)=0;
my ($sum_rtt)=0;
my ($time)=0;
my ($temp_time)=0;
my ($temp_stop_time)=0;
my ($lock)=0;
my (@last_line)=0;
my ($inst_rtt)=0;
my ($counter)=0;
print OUTFILE ("$time\t$inst_rtt\n");

@last_line=split(/[\t +]/,$input_lines[@input_lines-1]);
for ($i=0;$i< @input_lines;$i++)
{



     if ($time>($point_time+$interval_time))
     {

    $point_time=$point_time+$interval_time;
     $counter--;
     #print ("$point_time $sum_rtt $counter\n");
     if ($counter!=0)
     {
     $inst_rtt=$sum_rtt/$counter;

     print OUTFILE ("$point_time\t$inst_rtt\n");
     }
    $sum_rtt=0;
    $counter=0;
     }
     if (($last_line[0]-$point_time)<=$interval_time)
```

```
    {

          $lock=1;

    }
    $sum_rtt=$sum_rtt+$rtt;

    $in_line=$input_lines[$i];
    chomp($in_line);
    @line=split(/[\t +]/,$in_line);
    $rtt=$line[1];
    $time=$line[0];
    $counter++;

 #print OUTFILE ("\t$time\t$rtt\n");

}

#print ("\n$lock\n");
if ($lock==1)
{
    if ($counter!=0)
    {

    $sum_rtt=$sum_rtt+$rtt;
    #print ("$time $sum_rtt $counter\n");

    $inst_rtt=$sum_rtt/$counter;
    print OUTFILE ("$time\t$inst_rtt\n");
    }
    $lock=0;
}
#print ("$last_line[1]\n");
close(INPUTFILE);
close(OUTPUTFILE);
}
else {
    print("dose sosta ta orismata (syntax:perl ipv6_rtt.pl
<filename> <time interval in sec>).\n");
}
```

Βιβλιογραφία

[1] http://eclass.epp.teiher.gr/eclass/claroline/document/document.php
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[2] Ε.Μ.Πάλλης, "Towards the Provision of High Quality Interactive Multimedia Services via a Wireless Broadband Networking Infrastructure" PhD thesis, 2002

[3] http://lyk-malion.ira.sch.gr/mpeg.htm
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[4] Ν.Ζώτος, Γ.Τζιαχρής, "Μελέτη, σχεδίαση, υλοποίηση και αξιολόγηση ενός δικτύου παροχής αμφίδρομων διαδραστικών υπηρεσιών με τη χρήση πλατφόρμας επίγειας ψηφιακής τηλεόρασης DVB-T χρησιμοποιώντας την τεχνολογία GPRS στο κανάλι επιστροφής", Πτυχιακή Εργασία, Τ.Ε.Ι Κρήτης, Ιανουάριος, 2006

[5] http://www.ist-athena.org/Deliverables/ATHENA%20Deliverable%20D1.1.pdf
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[6] A.S.Tanenbaum, "Δίκτυα Υπολογιστών", Τρίτη έκδοση

[7] http://en.wikipedia.org/wiki/Transmission_Control_Protocol
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[8] http://en.wikipedia.org/wiki/User_Datagram_Protocol
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[9] http://www.ist-athena.org/Deliverables/ATHENA%20D3.2%20Final%20version.pdf
Ημερομηνία τελευταίας επίσκεψης: 12/02/2007

[10] http://www.ist-athena.org/Deliverables/ATHENA%20D3.1%20Final.pdf
Ημερομηνία τελευταίας επίσκεψης: 12/02/2007

[11] http://en.wikipedia.org/wiki/PSTN
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[12] http://de.wikipedia.org/wiki/Wireless_LAN
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[13] http://en.wikipedia.org/wiki/Digital_Subscriber_Line
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[14] http://www.ist-athena.org/Deliverables/ATHENA%20D3.1%20Final.pdf
Ημερομηνία τελευταίας επίσκεψης: 12/02/2007

[15] E Pallis, C. Mantakas, G. Mastorakis, V. Zacharopoulos, "Digital Switchover in UHF: the ATHENA concept for Broadband access", 14th IST Mobile & Wireless Communications Summit 2005, Dresden, 19-23 June, 2005

[16] http://dast.nlanr.net/Projects/Iperf/
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[17] http://pf.itd.nrl.navy.mil/mgen/
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[18] http://jarok.cs.ohiou.edu/software/tcptrace/
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[19] http://masaka.cs.ohiou.edu/software/tcptrace/xpl2gpl/
Ημερομηνία τελευταίας επίσκεψης: 06/04/2007

[20] Γ.Γαρδίκης, "Παροχή υπηρεσιών TCP/IP σε συστήματα επίγειας ψηφιακής τηλεόρασης (DVB-T)", PhD thesis, 2004

[21] www.grnet.gr/index.php?op=modload&modname=Sitemap&action=sitemapviewpage&pageid=246&language=el
Ημερομηνία τελευταίας επίσκεψης: 16/04/2007

[22] http://www.ciscopress.com/articles/article.asp?p=357102&seqNum=2&rl=1
Cisco system Quality of Service
Ημερομηνία τελευταίας επίσκεψης: 25/04/2007