

πραγιά τ

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΜΕΣΣΟΛΟΓΓΙΟΥ

ΤΜΗΜΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ &
ΔΙΚΤΥΩΝ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

« Μετάδοση υπηρεσιών video σε
Ασύρματα Δίκτυα »

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΤΖΟΥΜΑΣ ΕΜΜΑΝΟΥΗΛ

ΕΠΙΒΛΕΠΩΝ: Τάσος Νταγιούκλας
Ηλίας Πολίτης

ΝΑΥΠΙΑΚΤΟΣ 2009

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Ναύπακτος, .../.../2009

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Τάσος Νταγιουκλάς

2. Σπυρίδων Λούβρος

3. Αγγελική Πραγιάτη

Περίληψη

Στο παρελθόν, τα κινητά ασύρματα δίκτυα θεωρήθηκαν χαρακτηριστικά ως ομοιογενή RF προβλήματα των μέσων. Με το χρόνο και τον πολλαπλασιασμό των πολυάριθμων ιδιοκτητών ασύρματων δικτύων, η ανάγκη για ετερογενή διαλειτουργικότητα στα δίκτυα γίνεται μια πιο επικρατούσα ενδιαφέρουσα περιοχή. Οι σημερινές εφαρμογές πολυμέσων αναμένεται να τρέχουν σε ετερογενή περιβάλλοντα που αποτελούνται και από ενσύρματα και από ασύρματα υποσυστήματα. Οι ασύρματες συνδέσεις εκθέτουν τα ευδιάκριτα χαρακτηριστικά, όπως το περιορισμένο εύρος ζώνης, κ.α.. Συνεπώς, οι απαιτήσεις ποιότητας υπηρεσιών στα ασύρματα δίκτυα είναι αυστηρές και περίπλοκες. Αυτό θέτει νέες προκλήσεις ειδικά για τα τις εφαρμογές ροής πολυμέσων.

Η παρούσα πτυχιακή έχει σαν στόχο να μελετήσει το πρόβλημα της απώλειας των πακέτων στα ασύρματα δίκτυα και πως αυτές οι απώλειες επηρεάζουν την ποιότητα της υπηρεσίας βίντεο. Πιο συγκεκριμένα έχουν μελετηθεί αλγόριθμοι, οι οποίοι διαφοροποιούν τις απώλειες των πακέτων λόγω συμφόρησης από τις απώλειες λόγω λαθών στο φυσικό κανάλι. Οι αλγόριθμοι αυτοί έχουν υλοποιηθεί στο εργαλείο εξομοίωσης NS-2 και έχουν αξιολογηθεί για μετάδοση βίντεο πραγματικού χρόνου σε ασύρματα δίκτυα. Επίσης έχει υλοποιηθεί ένας μηχανισμός FEC (Forward Error Correction) για να γίνει βελτίωση της ποιότητας του βίντεο στην περίπτωση που οι αλγόριθμοι ανιχνεύσουν ότι τα σφάλματα οφείλονται σε λάθη του φυσικού καναλιού.

Abstract

In the past, mobile networks were typically looked upon as homogeneous RF media problems. With time and the proliferation of numerous proprietary wireless networks, the need for interoperability in heterogeneous networks is becoming a more prevalent interest area. Current multimedia applications are expected to run in heterogeneous environments composed of both wired and wireless components. Wireless links exhibit distinct characteristics, such as limited bandwidth, etc. Therefore, the quality of services in wireless networks is stringent and complex. This poses new challenges, especially for multimedia streaming applications.

This thesis aims to study the problem of packet loss in wireless networks and how these losses affect the quality of video. Three algorithms, which differentiate the packet losses

owing to congestion from losses due to impairments in the physical layer, have been studied. The algorithms have been implemented in NS-2 and evaluated for video-streaming in wireless networks. Furthermore, a FEC (Forward Error Correction) mechanism has been developed in order to improve video quality in case the algorithms detect that packet losses are due to physical impairments.

Ευχαριστίες

Η περάτωση της πτυχιακής μου εργασίας δεν είναι μόνο αποτέλεσμα προσωπικής μου εργασίας αλλά και συνεργασίας με ανθρώπους που με βοήθησαν πολύ.

Πρώτα απ' όλα θέλω να ευχαριστήσω τους καθηγητές μου, κ. Τάσο Νταγιούκλα και κ. Ηλία Πολίτη, για τη συνεχή, άμεση και ουσιαστική καθοδήγηση τους και την εξαιρετικά πολύτιμη βοήθεια που μου προσέφεραν, η οποία αποτέλεσε ένα βασικό συστατικό για την επιτυχή ολοκλήρωση της παρούσας εργασίας.

Ευχαριστώ θερμά την οικογένεια μου που βοηθάει και στηρίζει τις προσπάθειες μου όλα αυτά τα χρόνια.

Ευχαριστώ επίσης φίλους και καθηγητές που με στήριξαν, με ενέπνευσαν και με βοήθησαν στην προσπάθεια μου να πετύχω το στόχο μου.

Περιεχόμενα

Περιεχόμενα.....	iv
Λίστα Εικόνων.....	vi
Λίστα Πινάκων.....	vi
Λίστα Γραφημάτων.....	vi
Κεφάλαιο 1 ^ο – Εισαγωγή.....	1
Κεφάλαιο 2 ^ο - Εξέλιξη των προτύπων κωδικοποίησης βίντεο.....	5
1. Εξέλιξη των προτύπων MPEG.....	5
1.1. ISO/IEC 11172 (MPEG-1).....	5
1.2. ISO/IEC 13818 (MPEG-2).....	5
1.3. ISO/IEC 14496 (MPEG-4).....	6
1.4. Δομή βίντεο.....	6
2. Εξέλιξη του προτύπου H.26X.....	8
3. Κλιμακωτή Κωδικοποίηση Βίντεο (Video Scalability Coding).....	10
3.1. Χωρική Κλιμάκωση (Spatial Scalability).....	12
3.2. Ποιοτική Κλιμάκωση (Fidelity/SNR/Quality Scalability).....	13
3.3. Χρονική Κλιμάκωση (Temporal Scalability).....	14
3.4. Κλιμάκωση στην Πολυπλοκότητα (Complexity Scalability).....	15
Κεφάλαιο 3 ^ο – Βίντεο πάνω από ασύρματα δίκτυα.....	16
1. Ετερογενή Ασύρματα Δίκτυα.....	16
1.1. Χαρακτηριστικά ετερογενών ασύρματων δικτύων.....	16
2. Βίντεο πάνω από ετερογενή ασύρματα δίκτυα.....	17
2.1. Συμφόρηση και λάθη στο φυσικό επίπεδο σε ασύρματα δίκτυα.....	18

3. Τεχνικές βελτίωσης ποιότητας εξυπηρέτησης	20
3.1. Forward Error Correction (FEC)	20
Κεφάλαιο 4 ^ο – Εξομοίωση – Αποτελέσματα.....	22
1. Εξομοιωτής	22
2. Τοπολογία Δικτύων.....	22
2.2. Τοπολογία με χρήση CBR Traffic	24
2.3. Τοπολογία με χρήση Exponential Traffic.....	24
2.4. Αποτελέσματα	25
2.4.1 Τοπολογία με χρήση CBR κίνησης.....	25
2.4.2 Τοπολογία με χρήση Exponential κίνησης	32
Κεφάλαιο 5 ^ο – Συμπεράσματα και Μελλοντική Δουλεία.....	38
Βιβλιογραφία	40
Συνομογραφίες.....	42
Παράρτημα Α	1
Διευθύνσεις Internet	1
Παράρτημα Β.....	1
Α. Κώδικας του TCL αρχείου για την τοπολογία με χρήση CBR Traffic	1
Β. Κώδικας του TCL αρχείου για την τοπολογία με χρήση Exponential Traffic.....	10
Γ. Δημιουργία πακέτων FEC.....	20
Δ. Παραλήπτης Βίντεο.	25

Λίστα Εικόνων

Εικόνα 1: Απεικόνιση ετερογενών δικτύων	3
Εικόνα 2: Δομή Βίντεο	7
Εικόνα 3: Πρόβλεψη ψηφίων	8
Εικόνα 4: Χωρική Κλιμάκωση	13
Εικόνα 5: Ποιοτική Κλιμάκωση	14
Εικόνα 6: Χωρική, Χρονική και Ποιοτική Κλιμάκωση	15
Εικόνα 7: Ο αλγόριθμος Bias	19
Εικόνα 8: Ο αλγόριθμος Spike	19
Εικόνα 9: Διάγραμμα κωδικοποίησης – αποκωδικοποίησης δεδομένων FEC	21
Εικόνα 10: Καταστάσεις αλυσίδας Markov.	23
Εικόνα 11: Τοπολογία δικτύου με χρήση CBR Traffic	24
Εικόνα 12: Τοπολογία δικτύου με χρήση Exponential Traffic	25

Λίστα Πινάκων

Πίνακας 1: Σύντομη περιγραφή προτύπων κωδικοποίησης εικόνας και βίντεο.....	10
Πίνακας 2: Ρυθμοί Μετάδοσης.....	23
Πίνακας 3: Πίνακας πιθανοτήτων.....	23

Λίστα Γραφημάτων

Γράφημα 1: Εκτίμηση ποσοστού απωλειών σε σχέση με τις συνολικές απώλειες - Πιθανότητα pBB 0,85	26
Γράφημα 2: Overhead - Πιθανότητα pBB 0,85	27
Γράφημα 3: PSNR Αλγορίθμων - Πιθανότητα pBB 0,85	27
Γράφημα 4: Εκτίμηση ποσοστού απωλειών σε σχέση με τις συνολικές απώλειες - Πιθανότητα pBB 0,90	28
Γράφημα 5: Overhead - Πιθανότητα pBB 0,90	29
Γράφημα 6: PSNR Αλγορίθμων - Πιθανότητα pBB 0,90	29
Γράφημα 7: Εκτίμηση ποσοστού απωλειών σε σχέση με τις συνολικές απώλειες - Πιθανότητα pBB 0,94	30

Γράφημα 8: Overhead - Πιθανότητα ρBB 0,94	30
Γράφημα 9: PSNR Αλγορίθμων - Πιθανότητα ρBB 0,94	31
Γράφημα 10: Μεταβολή της πιθανότητας ρBB	31
Γράφημα 11: Εκτίμηση ποσοστού απωλειών σε σχέση με τις συνολικές απώλειες - Πιθανότητα ρBB 0,85	32
Γράφημα 12: Overhead - Πιθανότητα ρBB 0,85	33
Γράφημα 13: PSNR Αλγορίθμων - Πιθανότητα ρBB 0,85	33
Γράφημα 14: Ποσοστό απωλειών σε σχέση με τις συνολικές απώλειες - Πιθανότητα ρBB 0,90	34
Γράφημα 15: Overhead - Πιθανότητα ρBB 0,90	34
Γράφημα 16: PSNR Αλγορίθμων - Πιθανότητα ρBB 0,90	35
Γράφημα 17: Εκτίμηση ποσοστού απωλειών σε σχέση με τις συνολικές απώλειες - Πιθανότητα ρBB 0,94	36
Γράφημα 18: Overhead - Πιθανότητα ρBB 0,94	36
Γράφημα 19: PSNR Αλγορίθμων - Πιθανότητα ρBB 0,94	37
Γράφημα 20: Μεταβολή της πιθανότητας ρBB	37

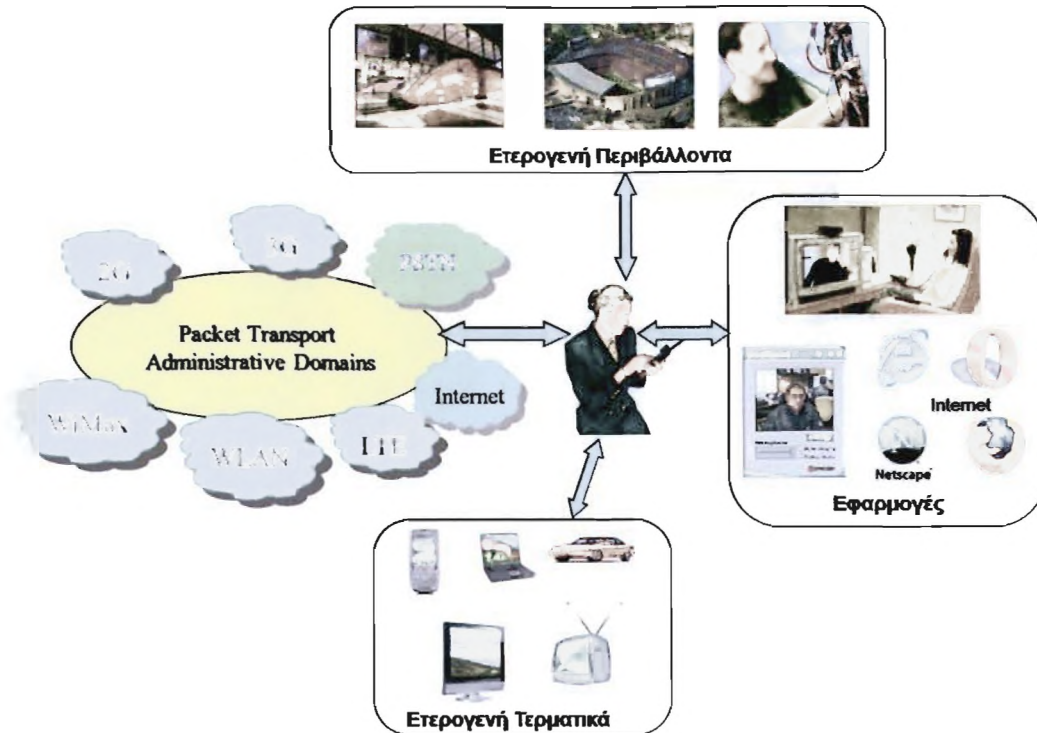
Κεφάλαιο 1^ο – Εισαγωγή

Τα τελευταία χρόνια, οι υπηρεσίες πολυμέσων αυξήθηκαν γρήγορα λόγω των ασύρματων ευρυζωνικών δικτύων και την ευέλικτη μεταφορά που βασίζεται στην IP τεχνολογία. Επιπλέον, η Web 2.0 εποχή φέρνει όλο και πιο πλούσια σε περιεχόμενο υπηρεσίες άμεσα στους καταναλωτές. Οι νέες γενιές των ασύρματων ευρυζωνικών δικτύων, όπως το 3G, το WiMAX, και τα δίκτυα πλέγματος είναι είτε προς ανάπτυξη είτε αναδυόμενα. Αυτό αυξάνει σημαντικά το εύρος ζώνης των ασύρματων επικοινωνιών και επιπλέον διευκολύνει την παράδοση και την διανομή του εύρους ζώνης σε απαιτητικές εφαρμογές πολυμέσων όπως είναι το βίντεο πραγματικού χρόνου και οι βίντεο-διασκέψεις σε κινητά τερματικά. Η μετάβαση στο πρωτόκολλο IP κάνει ακόμα πιο αποτελεσματική τη χρήση του εύρους ζώνης και βελτιστοποιεί τη συνολική end-to-end επίδοση του συστήματος. Οι επικοινωνίες πολυμέσων μπορούν να ταξινομηθούν σε πολλές διαφορετικές κατηγορίες με βάση τα σενάρια εφαρμογών τους. Από τη πλευρά του περιεχομένου, μπορούν να ταξινομηθούν σε ζωντανό περιεχόμενο, όπως η ζωντανή μετάδοση κάποιου γεγονότος (πχ αγώνας μπάσκετ), σε πραγματικό χρόνο επικοινωνία συμπεριλαμβανομένης της φωνής μέσω IP (VoIP), σε βίντεο διασκέψεις, με ήχο ή και με βίντεο chat, και σε άλλες μεθόδους στις οποίες το περιεχόμενο πολυμέσων που δημιουργείται συνεχώς και το αποθηκευμένο περιεχόμενο είναι έτοιμο πριν από την ώρα και να είναι αποθηκευμένο σε συγκεκριμένες μορφές. Από την πλευρά της παράδοσης οι επικοινωνίες πολυμέσων μπορούν να ταξινομηθούν σε πραγματικού χρόνου όπου τα περιεχόμενα των μέσων αναπαράγονται ενώ τα δεδομένα μεταφέρονται στους δέκτες. Επίσης ταξινομούνται και σε λήψεις όπου τα περιεχόμενα των πολυμέσων αναπαράγονται μόνο όταν ληφθούν όλα τα δεδομένα. Από την πλευρά του μοντέλου της διανομής οι επικοινωνίες πολυμέσων μπορούν να ταξινομηθούν σε μοντέλο πελάτη – εξυπηρετητή, όπως το βίντεο κατ' απαίτηση, όπου το περιεχόμενο των μέσων φιλοξενείται ή αναμεταδίδεται από έναν κεντρικό εξυπηρετητή και στην ισότιμων δικτύων (P2P) , όπως το P2P streaming και P2P διαμοιρασμό αρχείων όπου ο κάθε ισότιμος κόμβος μπορεί να είναι και διαμοιραστής και δέκτης περιεχομένου και τα ισότιμα συνεργάζονται μεταξύ τους για να αποκτήσουν το περιεχόμενο. Από την πλευρά της αλληλεπίδρασης, οι επικοινωνίες πολυμέσων μπορούν να ταξινομηθούν σε ανεκτικές σε καθυστέρηση συνόδου και σε διαδραστικές. Μια τυπική πολυμεσική εφαρμογή επικοινωνιών γενικά μπορεί ταυτόχρονα να ανήκει σε μερικές διαφορετικές κατηγορίες. Για παράδειγμα, μια εφαρμογή VoIP ανήκει στις κατηγορίες όπως στο ζωντανό περιεχόμενο, στη παράδοση πραγματικού χρόνου, σε ένα μοντέλο πελάτη-εξυπηρετητή και σε διαδραστικές συνόδους. Παρόμοια, ένα

P2P αρχείο μέσου ανήκει στην κατηγορία αποθηκευμένου περιεχομένου, στο P2P μοντέλο και στην ανεκτική σε καθυστέρηση σύνοδο.

Στο χώρο των κινητών δικτύων βρίσκουμε εφαρμογές από σχεδόν όλες τις κατηγορίες των πολυμεσικών επικοινωνιών που αναφέρθηκαν παραπάνω. Οι κινητές επικοινωνίες πολυμέσων βασίζονται στα ασύρματα κανάλια επικοινωνιών για την παροχή δεδομένων. Η πρόσφατη ανάπτυξη της επόμενης γενιάς ασύρματων δικτύων, όπως το 3G, το WiMax για ασύρματα δίκτυα πρόσβασης και τα Ultra-Wide Band για προσωπικά τοπικά δίκτυα, υπόσχεται μια σημαντική ενίσχυση του εύρους ζώνης, ωστόσο, συγκρινόμενη με τα ενσύρματα δίκτυα, το εύρος ζώνης στα ασύρματα δίκτυα φαίνεται να υστερεί. Ως εκ τούτου, πιο αποτελεσματικοί κωδικοποιητές μέσων και πρωτόκολλα μεταφοράς είναι πάντοτε επιθυμητά στον τομέα των κινητών επικοινωνιών. Η φύση των ασύρματων καναλιών είναι ότι έχουν την προδιάθεση λάθους εξαιτίας των τυχαίων σφαλμάτων και τις ριπές λαθών (burst errors). Επιπλέον, η κινητικότητα μιας τερματικής συσκευής καθιστά τις συνθήκες στα ασύρματα κανάλια ακόμα πιο δυναμικές και απρόβλεπτες. Το πώς να σχεδιαστεί ένα ευπροσάρμοστο σε λάθη και ακόμα πιο αποτελεσματικό σύστημα κινητών μέσων επικοινωνίας είναι ακόμη μία πρόκληση. Αυτό μπορεί να περιλαμβάνει ευέλικτη προσαρμογή σε λάθη κωδικοποίησης μέσων, ανθεκτική μεταφορά λαθών, και αποδοτική επεξεργασία απόκρυψης λάθους όταν αυτά εμφανίζονται. Τρίτον, οι συνθήκες στο εύρος ζώνης και στο ασύρματο κανάλι διαφέρουν σημαντικά από χρόνο σε χρόνο και τα κινητά τερματικά έχουν διαφορετικά τεχνικά χαρακτηριστικά (πχ η ανάλυση οθόνης, η επεξεργαστική ισχύς, η μνήμη και ο χώρος αποθήκευσης, μέθοδοι πρόσβασης στο δίκτυο, και άλλα). Επιπλέον, οι περισσότερες από τις τρέχουσες επικοινωνίες πολυμέσων σε κινητές θέσεις είναι του μοντέλου πελάτη – εξυπηρετητή στον οποίο ένας κεντρικός εξυπηρετητής φιλοξενεί το περιεχόμενο των μέσων ή αναμεταδίδει τα πακέτα μεταξύ των κινητών τερματικών.

Το διαθέσιμο εύρος ζώνης στα κυψελοειδή συστήματα είναι ακόμα μακριά από εκείνα που έχουμε στα σημερινά πολύ γρήγορα ενσύρματα δίκτυα. Επιπλέον, τα χαρακτηριστικά λάθους των ασύρματων συνδέσεων απαιτούν τα διαφορετικά σχέδια για τα βέλτιστα πρωτόκολλα. Η Εικόνα 1 παρουσιάζει ένα παράδειγμα μιας ετερογενούς υποδομής δικτύου που αποτελείται από ετερογενή ασύρματα υποδίκτυα και συνδέσεις χρησιμοποιώντας τις διαφορετικές τεχνολογίες. Πολλές εφαρμογές πολυμέσων απαιτούν μετάδοση βίντεο πάνω από συνδέσεις με ασύρματο πρώτο ή/και τελευταίο βήμα. Οι απλοί χρήστες και οι μικρές επιχειρήσεις είναι συνδεδεμένοι μέσω δικτύου πρόσβασης του παρόχου υπηρεσιών διαδικτύου τους που χρησιμοποιεί τεχνολογίες ψηφιακών συνδρομητικών γραμμών. Οι κινητοί χρήστες μπορούν να έχουν πρόσβαση στο διαδίκτυο από το κυψελοειδές δίκτυο.



Εικόνα 1: Απεικόνιση ετερογενών δικτύων.

Αυτή η διπλωματική εργασία θα εξετάσει δύο ειδών απώλειες που εμφανίζονται στα ασύρματα δίκτυα και επηρεάζουν την ποιότητα της υπηρεσίας του video. Αυτές είναι οι απώλειες λόγω παραμορφώσεων στο φυσικό μέσο (π.χ. υψηλό BER) και απώλειες λόγω συμφόρησης του δικτύου (π.χ. απλές απώλειες πακέτων και απώλειες ριπής πακέτων). Οι απώλειες αυτές και οι επιδράσεις τους στην ποιότητα του video θα μελετηθούν και θα αξιολογηθούν. Για αυτό το σκοπό, η διπλωματική μελετά τρεις αλγόριθμους οι οποίοι κάνουν μια εκτίμηση για το που οφείλονται οι απώλειες των πακέτων στο IP επίπεδο. Οι αλγόριθμοι αυτοί υπολογίζουν το ποσοστό των λαθών λόγω συμφόρησης και το ποσοστό των λαθών λόγω λαθών στο φυσικό κανάλι. Στο επόμενο βήμα, χρησιμοποιείται ένας προσαρμοστικός αλγόριθμος FEC, ο οποίος προσαρμόζει το μέγεθος των FEC πακέτων, ανάλογα με το ποσοστό εκτίμησης των απωλειών λόγω λαθών του φυσικού καναλιού.

Η διάρθρωση της διπλωματικής εργασίας είναι η ακόλουθη: Το Κεφάλαιο 2 κάνει μια σύντομη επισκόπηση των τεχνολογιών αιχμής βίντεο και τα πρότυπα που χρησιμοποιούνται σήμερα. Το Κεφάλαιο 3 εξετάζει τη μετάδοση ψηφιακού βίντεο πάνω από ασύρματα δίκτυα και παρουσιάζει τρεις αλγορίθμους οι οποίοι κάνουν εκτίμηση το ποσοστό των απωλειών σε ένα ασύρματο δίκτυο αν οφείλονται σε συμφόρηση ή σε λάθη στο φυσικό κανάλι. Το Κεφάλαιο 4 παρουσιάζει την υλοποίηση των ανωτέρων αλγορίθμων στο περιβάλλον του

εργαλείου εξομοίωσης NS-2, όπως επίσης και την απόδοση της χρήσης ενός μηχανισμού FEC τόσο στο δίκτυο όσο και στις υπηρεσίες βίντεο, στην περίπτωση ανίχνευσης από τον αλγόριθμο ότι οι απώλειες των πακέτων οφείλονται σε λάθη του φυσικού καναλιού, χρησιμοποιώντας διαφορετικά είδη όσον αφορά τη χρήση τεχνικής κίνησης στο ασύρματο δίκτυο (Exponential και CBR). Η απόδοση τόσο του δικτύου όσο και του βίντεο έχουν μετρηθεί και έχει γίνει μια σύγκριση όσον αφορά την αξιοπιστία των ανωτέρων αλγορίθμων. Τέλος το Κεφάλαιο 5 παρουσιάζει τα συμπεράσματα από τη πειραματική μελέτη καθώς και προτάσεις για μελλοντική επέκταση της εργασίας.

Κεφάλαιο 2^ο - Εξέλιξη των προτύπων κωδικοποίησης βίντεο

Οι πρόοδοι στην τεχνολογία κωδικοποίησης και τυποποίησης βίντεο μαζί με τις γρήγορες εξελίξεις και τη βελτίωση των υποδομών δικτύου, τη μεγάλη χωρητικότητα, και την υπολογιστική ισχύ επιτρέπουν έναν αυξανόμενο αριθμό εφαρμογών βίντεο. Οι τομείς εφαρμογής κυμαίνονται σήμερα από τα μήνυμα πολυμέσων, την τηλεφωνία με χρήση βίντεο, και την βίντεο-διάσκηψη πάνω από κινητή τηλεόραση, ασύρματη και ενσύρματη τηλεοπτική ροή στο Διαδίκτυο, βασικής και υψηλής ευκρίνειας τηλεοπτική αναμετάδοση ως DVD, δίσκους blue-ray, και τα μέσα οπτικής αποθήκευσης HD DVD. Για αυτές τις εφαρμογές, ποικίλα τηλεοπτικά συστήματα μετάδοσης και αποθήκευσης υιοθετούνται. Τα πρότυπα τεχνικών κωδικοποίησης βίντεο (MPEG-X και H26X) έχουν περάσει μέσω πολλών σταδίων εξέλιξης σύμφωνα με την εξέλιξη της τεχνολογίας Very Large Scale Integration (VLSI). Τα H26X πρότυπα αναπτύχθηκαν από την ομάδα ITU-T εκτός από το H264 που αναπτύχθηκε από μια συνεργασία μεταξύ της ομάδας ISO/IEC και ITU-T. Παρακάτω φαίνεται η εξέλιξη των προτύπων κάθε ομάδας:

1. Εξέλιξη των προτύπων MPEG

Το MPEG είναι μία από τις πιο διάσημες τεχνικές συμπίεσης ήχου και βίντεο γιατί δεν είναι μόνο ένα πρότυπο αλλά αντιθέτως είναι πολλά πρότυπα κατάλληλα για διαφορετικές εφαρμογές. Τα πρότυπα MPEG δημιουργήθηκαν από την ISO/IEC Moving Picture Experts Group (MPEG). Αυτά τα πρότυπα είναι το MPEG-1, το MPEG-2 και το MPEG-4 [1][2].

1.1. ISO/IEC 11172 (MPEG-1)

Το MPEG-1 είναι ένα πρότυπο κωδικοποίησης κινούμενων εικόνων και ήχου με λιγότερο από 1.5 Mbits/s εύρος ζώνης και ανάλυση εικόνας 352x240. Επίσης χρησιμοποιείται για αποθήκευση σε ψηφιακά μέσα όπως το CD-ROM.

1.2. ISO/IEC 13818 (MPEG-2)

Το MPEG-2 είναι η εξέλιξη του MPEG-1. Χρησιμοποιεί μεγαλύτερο εύρος ζώνης και ποικίλες αναλύσεις σε σύγκριση με τον προκατόχο του, 3-15 Mbits/s για κανονικής

ευκρίνειας τηλεοράσεις και 15-30 Mbits/s για υψηλής ευκρίνειας τηλεοράσεις των οποίων η ανάλυση είναι 1920x1080.

1.3. ISO/IEC 14496 (MPEG-4)

Το MPEG-4 χρησιμοποιείται για κωδικοποίηση οπτικοακουστικών αντικειμένων, παρέχοντας νέες βασισμένες στο αντικείμενο λειτουργίες, τη συνθετική και τη φυσική ολοκλήρωση, τις νέες μορφές αλληλεπίδρασης, κλπ.

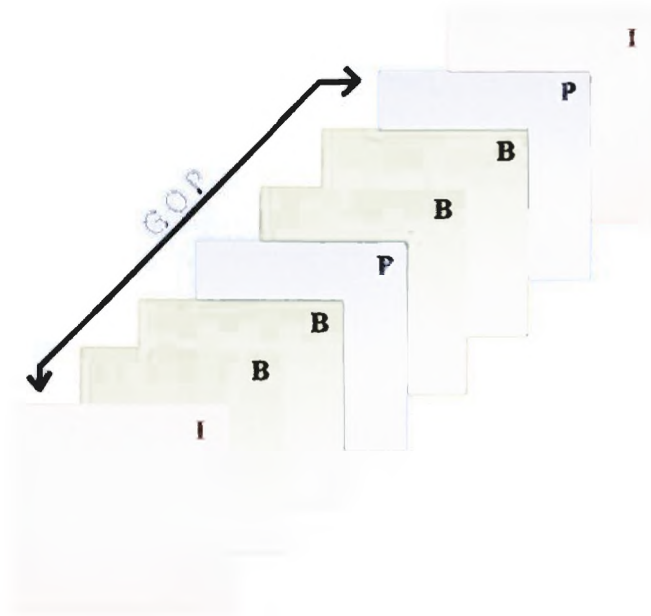
Αρχικά όλα τα πρότυπα MPEG συμπεριλαμβανομένων και των τεχνολογιών αντιπροσώπευσης των μέσων ήταν δομημένα και εξέταζαν τους στόχους στις βάσεις μιας 'τριλογίας' από τρία σημαντικά μέρη, τα τηλεοπτικά/οπτικά και ακουστικά συστήματα. Μάλλον ασυνήθιστα σε άλλους οργανισμούς τυποποίησης, παρουσιάζει πρόωρη αναγνώριση ότι τα πολυμέσα είναι για την ολοκλήρωση των μέσων και εξετάζει έτσι τα μεμονωμένα μέσα που απομονώνονται και δεν μπορούν να οδηγήσουν σε καλύτερα πρότυπα πολυμέσων. Τον Ιουλίου του 2005, το MPEG αναγνώρισε ότι η προηγούμενη προσέγγιση τυποποίησης βασισμένη στα μεγάλα προγράμματα όπως αυτά που προαναφέρθηκαν παραπάνω δεν ήταν πλέον τα πιο ικανοποιητικά, αλλά αντιθέτως διάφορα μικρότερα πρότυπα έπρεπε να αναπτυχθούν. Επίσης τα περισσότερα από αυτά τα πρότυπα δεν αφορούσαν άμεσα οποιαδήποτε από τα μεγάλα προγράμματα MPEG ήδη. Μετά από αυτήν την αναγνώριση, ένας νέος τύπος της δομής προτύπων MPEG δημιουργήθηκε για να συμπληρώσει το διαθέσιμο σύνολο μεγάλων προγραμμάτων, ειδικότερα:

- το ISO/IEC 23000 (MPEG-A), "μορφές εφαρμογών πολυμέσων", καθορίζει τις προσαρμοσμένες προς την εφαρμογή μορφές αρχείου στα πρότυπα MPEG.
- το ISO/IEC 23001 (MPEG-B), "τεχνολογίες συστημάτων MPEG", καθορίζει τα εργαλεία συστημάτων που μπορούν να χρησιμοποιηθούν σε όλα τα μεγάλα πρότυπα MPEG ή, τουλάχιστον, δεν συνδέονται άμεσα με αυτά.
- το ISO/IEC 23002 (MPEG-C), "οπτικές τεχνολογίες MPEG", καθορίζει τα οπτικά εργαλεία κωδικοποίησης που μπορούν να χρησιμοποιηθούν σε όλα τα μεγάλα πρότυπα MPEG ή δεν συνδέονται, τουλάχιστον, άμεσα με αυτά.

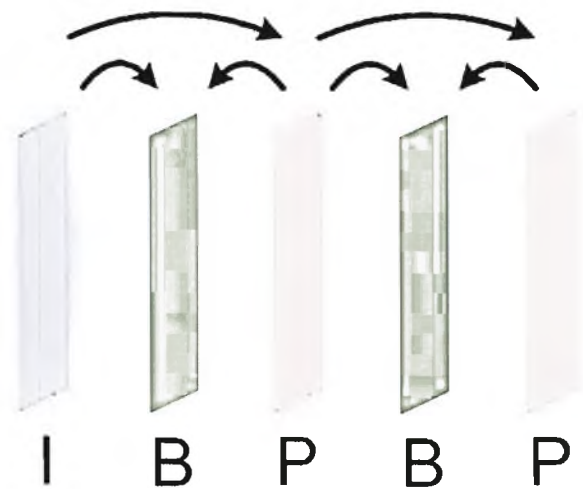
1.4. Δομή βίντεο

Το βίντεο στο πρότυπο MPEG αποτελείται από ομάδα εικόνων (Group Of Pictures - GOP) όπου το καθένα διαρκεί μεταξύ 0.5 και 1 δευτερόλεπτο. Αποτελείται από τριών ειδών

πλαίσια-εικόνες (frames), τα I (Intra), τα P (Predicted) και τα B (Bidirectional) πλαίσια, (Εικόνα 2). Τα πλαίσια I είναι ολόκληρες εικόνες οι οποίες μπορούν να κωδικοποιηθούν χωρίς να κωδικοποιήσουμε κάποια άλλη εικόνα πρώτα, είναι ίδιο με την εικόνα JPEG. Τα πλαίσια P προβλέπονται από το προηγούμενο πλαίσιο «αναφοράς» (I ή P). Εάν ο κωδικοποιητής μπορεί να βρει το συσχετισμό μεταξύ της προηγούμενης αναφοράς και του πλαισίου P. Τα macroblocks στο P πλαίσιο θα προέλθουν από την αναφορά με πληροφορίες διανύσματος και διαφοράς DCT κινήσεων. Στην περίπτωση όπου μια καλή αντιστοιχία δεν μπορεί να βρεθεί, το πλαίσιο P περιέχει κάποια ενδιάμεσα κωδικοποιημένα macroblocks. Τα B πλαίσια προβλέπονται από τα προηγούμενα και μελλοντικά πλαίσια αναφοράς, Εικόνα 3. Ο κωδικοποιητής μπορεί να χρησιμοποιήσει πληροφορίες από το macroblock για κάθε ένα από αυτά τα πλαίσια για να παραγάγει την καλύτερη αντιστοιχία για κάθε macroblock στο πλαίσιο B. Εάν καμία καλή αντιστοιχία δεν μπορεί να βρεθεί τότε το macroblock θα είναι ενδιάμεσα κωδικοποιημένο. Το κάθε GOP αρχίζει με ένα I πλαίσιο το οποίο είναι και το μοναδικό που υπάρχει, για περισσότερα I πλαίσια έχουμε μικρότερο GOP. Επίσης περιέχει δύο έως δέκα P πλαίσια και δύο έως τέσσερα B πλαίσια.



Εικόνα 2: Δομή Βίντεο



Εικόνα 3: Πρόβλεψη ψηφίων

2. Εξέλιξη του προτύπου H.26X

Τα πρότυπα H.26X περνούν από πολλά στάδια εξέλιξης δεδομένου ότι δημιουργούνται από την ITU-T Video Coding Experts Group (VCEG), αυτά τα στάδια εξέλιξης είναι:

- Το H.261 είναι τηλεοπτικό πρότυπο κωδικοποίησης που δημιουργήθηκε το 1990 από την ITU-U. Σχεδιάστηκε αρχικά για μετάδοση πάνω σε γραμμές ISDN στις οποίες ο ρυθμός δεδομένων είναι πολλαπλάσιος των 64 Kbit/s. Είναι ένα μέλος της οικογένειας των H.26x τηλεοπτικών προτύπων κωδικοποίησης της ITU-T Video Coding Experts Group (VCEG) [3].
- Το H.262 είναι ψηφιακό τηλεοπτικό πρότυπο κωδικοποίησης της ITU-T. Εμπίπτει στη δικαιοδοσία της ITU-T Video Coding Experts Group (VCEG) και υποστηρίζεται και από την ISO/IEC Moving Picture Experts Group (MPEG). Είναι ίδιο στο περιεχόμενο με το τηλεοπτικό μέρος του προτύπου ISO/IEC MPEG-2 (γνωστό και ως ISO/IEC 13818-2). Αυτό το πρότυπο αναπτύχθηκε σε μια κοινή συνεργασία μεταξύ των οργανώσεων ITU-T και ISO/IEC JTC 1, και έτσι δημοσιεύτηκαν ως πρότυπο και των δύο οργανώσεων [3].
- Το H.263 είναι πρότυπο κωδικοποίησης βίντεο που σχεδιάστηκε από την ITU-U σε ένα πρόγραμμα που τέλειωσε το 1995/1996 ως ένα χαμηλής ποιότητας συμπιεσμένο αρχείο που χρησιμοποιήθηκε για βίντεο-διάσκεψη. Είναι ένα μέλος της οικογένειας των H.26x προτύπων κωδικοποίησης βίντεο ITU-T Video Coding Experts Group (VCEG). Το H.263 αναπτύχθηκε ως εξελικτική βελτίωση βασισμένη στην εμπειρία από το H.261, τα προηγούμενα πρότυπα της ITU-T για την συμπίεση βίντεο και τα πρότυπα MPEG-1 και MPEG-2. Η πρώτη έκδοσή του ολοκληρώθηκε το 1995 και παρείχε μια κατάλληλη

αντικατάσταση για το H.261 σε όλους τους ρυθμούς μετάδοσης. Ενισχύθηκε περαιτέρω στα προγράμματα γνωστά και ως H.263v2 (επίσης γνωστό και ως H.263+ ή H.263 1998) και ως H.263v3 (επίσης γνωστό και ως H.263++ ή H.263 2000) [3].

- Το H.264 είναι ένα πρότυπο κωδικοποίησης βίντεο. Είναι επίσης γνωστό και ως MPEG-4 Part 10 ή ως MPEG-4 AVC (Advanced Video Coding). Χρησιμοποιείται κυρίως στο HD-DVD και στο Blue-Ray. Είναι ένα από τα πιο πρόσφατα πρότυπα κωδικοποίησης που αναπτύχθηκαν από την ITU-T Video Coding Experts Group (VCEG) μαζί με την ISO/IEC Moving Picture Experts Group (MPEG) ως προϊόν μιας προσπάθειας συνεργασίας γνωστής και ως Joint Video Team (JVT). Το πρότυπο ITU-T H.264 και το πρότυπο ISO/IEC MPEG-4 Part 10 (ή ISO/IEC 14496-10) διατηρούνται από κοινού έτσι ώστε να έχουν το ίδιο τεχνικό περιεχόμενο. Η πρώτη έκδοση του προτύπου ολοκληρώθηκε τον Μάιο του 2003. Το πρότυπο H.264 μειώνει την ποσότητα των πληροφοριών που απαιτούνται για την αναπαραγωγή βίντεο δηλαδή πετυχαίνει πιο αποδοτική συμπίεση. Οι κωδικοποιητές επεξεργάζονται κάθε πλαίσιο χωρίζοντας την εικόνα σε ένα πλέγμα από μπλοκ και αναζητούν προηγούμενα ή επόμενα πλαίσια για κάθε μπλοκ για να ταιριάξουν τη βασική δομή, η τεχνική αυτή είναι γνωστή ως εκτίμηση κίνησης (motion estimation). Όταν ένα κατάλληλο ταίριασμα βρεθεί, ο αποκωδικοποιητής μπορεί να αναπαράγει την δομή του μπλοκ στο τρέχον πλαίσιο χρησιμοποιώντας μόνο ένα δiάνυσμα που δείχνει την αντιστοιχία που βρέθηκε μαζί με κάποιες πληροφορίες για να διορθώσει μικρές διαφορές που μπορεί να παρατηρηθούν. Όπου η εκτίμηση κίνησης αποτυγχάνει να βρει τις κατάλληλες αντιστοιχίες, οι κωδικοποιητές χρησιμοποιούν τη δομή των κοντινών μπλοκ στο ίδιο πλαίσιο για να προβλέψουν τη σύσταση φραγμών και να αποθηκεύσουν τη διαφορά μεταξύ της πρόβλεψης και της πραγματικής δομής. Οι κωδικοποιητές ενεργούν ως «απωλεσθητικοί (lossy)» συμπίεστες γιατί ο στόχος τους δεν είναι να αναπαραγάγουν την αρχική εικόνα ακριβώς αλλά αντιθέτως να επιλέξουν το βέλτιστο μέσο για τη μείωση του ρυθμού δεδομένων, διατηρώντας την οπτική ποιότητα όσο γίνεται καλύτερη. Με τις κατάλληλες ρυθμίσεις οι διαφορές μπορεί να είναι ασήμαντες. Το πρότυπο H.264 προσφέρει ουσιαστικές βελτιώσεις απόδοσης σε σύγκριση με τους προκατόχους του. Για παράδειγμα ένα κανονικό DVD μπορεί να χωρέσει μια δίωρη ταινία συμπίεσμένη με το MPEG-2 (δημοφιλέστερο πρότυπο για ταινίες DVD) ενώ αν χρησιμοποιηθεί το H.264 τότε χωράνε δύο δίωρες ταινίες μέσα σε ένα κανονικό DVD [3][4][5][6].

Πρέπει να σημειωθεί ότι εκτός από τα πρότυπα κωδικοποίησης βίντεο που αναπτύσσονται από τα MPEG και VCEG, υπάρχουν επίσης μηχανισμοί κωδικοποίησης βίντεο όπως το VC-1 το οποίο δημιουργήθηκε από την Microsoft και την Real Video η οποία με τη σειρά της δημιουργήθηκε από την Real Networks. Μια σύνοψη αυτών των προτύπων κωδικοποίησης

βίντεο παρουσιάζεται στον πίνακα 1. Τέτοια σχήματα μέσω είναι εκτενώς χρησιμοποιημένο βίντεο ρέοντας μέσω του Διαδικτύου. Αυτά τα είδη των μέσων χρησιμοποιούνται εκτενώς στη μετάδοση βίντεο πραγματικού χρόνου μέσω του διαδικτύου.

Όνομα	Ημερομηνία ολοκλήρωσης	Βασικά Χαρακτηριστικά
H.261	1990	Για βίντεο-διάσκεψη, 64 Kbps – 1.92 Mbps.
MPEG-1	1991	Για CD-ROM, ≤1.5 Mbps.
JPEG	1992	Για την κωδικοποίηση ακίνητης εικόνας βασισμένη στον διακριτό μετασχηματισμό συνημίτονου (Discrete Cosine Transform).
MPEG-2 (H.262)	1994	Για DTV/DVD: 2-15 Mbps. Για ATSC HDTV: 19.2 Mbps, πιο χρησιμοποιούμενο πρότυπο.
H.263	1995	Για κωδικοποίηση με πολύ χαμηλό εύρος ζώνης, κάτω από 64 Kbps.
MPEG-4 Part 2	1999	Για πολυμέσα, κωδικοποίηση βασισμένη στο περιεχόμενο, το απλό και το προηγμένο προφίλ εφαρμόζονται στο βίντεο του κινητού και στη ροή.
RealVideo*	2000	Για πολλές εφαρμογές, απόδοση κωδικοποίησης παρόμοια με αυτήν του MPEG-4 part 2.
JPEG-2000	2000	Για την κωδικοποίηση ακίνητης εικόνας βασισμένη στον διακριτό μετασχηματισμό κυμάτων (Discrete Wavelet Transform – DWT).
H.264/AVC (MPEG-4 PART 10)	2005	Για πολλές εφαρμογές με σημαντικά βελτιωμένη απόδοση κωδικοποίησης πάνω από MPEG-2 και MPEG-4 part 2.
VC-1	2005	Για πολλές εφαρμογές, απόδοση κωδικοποίησης κοντά σε αυτήν του H.264.
<i>*Δεν είναι επίσημο πρότυπο αλλά σίγουρα ένα βιομηχανικό πρότυπο.</i>		

Πίνακας 1: Σύντομη περιγραφή προτύπων κωδικοποίησης εικόνας και βίντεο.

3. Κλιμακωτή Κωδικοποίηση Βίντεο (Video Scalability Coding)

Στις μέρες μας ολοένα και περισσότερο αυξάνεται ο όγκος των δεδομένων βίντεο που χειριζόμαστε καθημερινά. Μια σημαντική πρόκληση είναι η διανομή του περιεχομένου βίντεο σε χρήστες με διαφορετικά χαρακτηριστικά. Η εξάπλωση του Παγκόσμιου Ιστού άλλαξε τη φιλοσοφία της εμπορικής διανομής, με αποτέλεσμα σήμερα πάρα πολλοί χρήστες να χρησιμοποιούν υπηρεσίες βίντεο πραγματικού χρόνου (video-streaming) για να δούνε ταινίες, μουσικά βίντεο και άλλες υπηρεσίες πολυμέσων μέσω του Παγκόσμιου Ιστού. Επιπλέον, η ανάπτυξη της τεχνολογίας σύλληψης εικόνας προκάλεσε τη διείσδυση καμερών στις κινητές συσκευές, όπως τα κινητά τηλέφωνα και τα PDAs. Αυτό είχε σαν αποτέλεσμα ακόμα και οι φορητές συσκευές να λειτουργούν ως καταναλωτές ή και παραγωγοί ψηφιακού βίντεο.

Στις περισσότερες περιπτώσεις η μετάδοση βίντεο πραγματοποιείται μέσω μη αξιόπιστων δικτύων, χρησιμοποιώντας πρωτόκολλα βασιζόμενα στα πακέτα όπως για παράδειγμα το TCP/IP το οποίο δεν υποστηρίζει λειτουργίες ποιότητας εξυπηρέτησης για ανθεκτική και χρονικά ορθή διανομή περιεχομένου πολυμέσων. Το πρόβλημα διογκώνεται με την χρήση ασύρματων υπηρεσιών στο περιβάλλον του χρήστη, αφού τα ασύρματα δίκτυα εμφανίζουν απώλειες πακέτων και μεγάλες διακυμάνσεις στο εύρος ζώνης.

Για να αντιμετωπιστούν τα προβλήματα που δημιουργεί η ποικιλία των μέσων μετάδοσης βίντεο, η κοινότητα του MPEG αποφάσισε να διεξάγει έρευνα στις τεχνολογίες κωδικοποίησης βίντεο που μπορούν να χειριστούν τις απαιτήσεις μετάδοσης. Η έρευνα αυτή άρχισε το 2002 και συνεχίζεται ακόμα. Όσον αφορά το τεχνολογικό κομμάτι είναι γενικώς αποδεκτό ότι η κλιμακωτή κωδικοποίηση βίντεο (Scalable video coding - SVC) είναι η πιο βιώσιμη τεχνικά λύση στα προβλήματα της μετάδοσης βίντεο σε ετερογενή δίκτυα και τερματικά χρηστών με διαφορετικές δυνατότητες γιατί παρέχεται αντίστοιχη ποιότητα εξυπηρέτησης (Quality of Service). Η κλιμακωτή κωδικοποίηση βίντεο αποτελεί ένα πλαίσιο εργασίας όπου από μια ενιαία συμπιεσμένη ακολουθία βίντεο μπορούν να προκύψουν αναπαραστάσεις του βίντεο με διαφορετική ποιότητα, ανάλυση (resolution) και χρονικό ρυθμό πλαισίων.

Ένα σημαντικό κομμάτι της επιτυχίας ενός συγκεκριμένου κλιμακωτού αλγόριθμου κωδικοποίησης είναι το πόσο αποδοτική είναι η συμπίεση που δημιουργήσε. Συνεπώς, μια βασική απαίτηση κάθε αλγόριθμου κλιμακωτής κωδικοποίησης είναι ότι πρέπει να αποδίδει το ίδιο καλά με τα υπάρχοντα μη κλιμακωτά συστήματα κωδικοποίησης βίντεο όσον αφορά την οπτική ποιότητα. Αυτό θα πρέπει να συμβαίνει για την πλειοψηφία των απεικονίσεων, των επιπέδων ποιότητας και των χρονικών ρυθμών πλαισίων. Αυτό θα πρέπει να συμβαίνει για την πλειοψηφία των απεικονίσεων, των επιπέδων ποιότητας και των χρονικών ρυθμών πλαισίων. Μια άλλη απαίτηση αφορά τον χρόνο αποκωδικοποίησης. Γενικά για βίντεο

πραγματικού χρόνου (video streaming) ο χρόνος αποκωδικοποίησης είναι της τάξης δευτερολέπτων ενώ για εφαρμογές όπως παρακολούθηση βίντεο (video monitoring) απαιτείται χρόνος της τάξης των 100 ms. Όσον αφορά τις απαιτήσεις για κλιμάκωση (scalability) του περιεχομένου βίντεο, το MPEG ορίζει κάποιον ελάχιστο αριθμό χωρικών, χρονικών και ποιοτικών επιπέδων που πρέπει να διατίθενται. Τέλος, θα πρέπει να λάβουμε υπόψη και την πολυπλοκότητα της κωδικοποίησης και αποκωδικοποίησης. Οι απαιτήσεις αυτές αναλύονται περισσότερο στις ακόλουθες υποενότητες.

Υπάρχουν πολλά είδη ή τρόποι κλιμάκωσης. Οι πιο συνηθισμένοι είναι η χωρική κλιμάκωση (Spatial Scalability), η ποιοτική κλιμάκωση (Quality Scalability), η χρονική κλιμάκωση (Temporal Scalability) και η κλιμάκωση στην πολυπλοκότητα (Complexity Scalability).

3.1. Χωρική Κλιμάκωση (Spatial Scalability)

Η κλιμακωτή κωδικοποίηση του βίντεο πρέπει να υποστηρίζει το λιγότερο τέσσερα επίπεδα χωρικής ανάλυσης (ευκρίνειας) τα οποία κυμαίνονται από 1152 x 1408 έως 144 x 176 εικονοστοιχεία. Στο εύρος αυτό περιέχονται και δύο πολύ γνωστά πρότυπα για αναπαράσταση βίντεο, το CIF (288 x 352) και το QCIF (144 x 176). Η ικανότητα να παράγουμε πολλαπλές αναλύσεις από μία κοινή ακολουθία ψηφίων είναι σημαντική για αρκετές εφαρμογές. Μια από αυτές είναι η επιτήρηση του χώρου μέσω βίντεο και τα συστήματα παρακολούθησης, όπου η λειτουργία της χωρικής κλιμάκωσης (Εικόνα 4) έχει δύο λόγους ύπαρξης:

- Στο πρώτο σενάριο πολλές λήψεις από διαφορετικές θέσεις καταφτάνουν στον σταθμό επιτήρησης. Βασιζόμενοι στη δυνατότητα της χωρικής κλιμάκωσης κάθε λήψη μπορεί να μεγεθυνθεί. Έτσι μειώνεται το εύρος επικοινωνίας μεταξύ του δικτύου των καμερών και του σταθμού παρακολούθησης και το υψηλής ευκρίνειας περιεχόμενο μεταδίδεται σε περίπτωση συναγερμού ή από επιλογή του χρήστη.
- Την υποστήριξη αποθήκευσης βίντεο με λειτουργία διαγραφής. Στα συστήματα παρακολούθησης η σημαντικότητα των καταγεγραμμένων δεδομένων και κατ' επέκταση της απαιτούμενης ανάλυσης των ακολουθιών βίντεο, μειώνεται με το πέρασμα του χρόνου. Για παράδειγμα, το πλήρες ευκρίνειας βίντεο χρειάζεται να αποθηκευτεί μόνο για τρεις ημέρες. Οι ακολουθίες που είναι για χρονικό διάστημα μεγαλύτερο από μια εβδομάδα χρειάζονται μια ελάχιστη ευκρίνεια για την αρχειοθέτησή τους.

Τέλος, μια μεγάλη κατηγορία συστημάτων που απαιτούν κλιμάκωση στην ανάλυση είναι οι αρχιτεκτονικές επικοινωνίας που χρησιμοποιούν το πλαίσιο εργασίας MPEG-21 Digital Item Adaptation (DIA), όπου ετερογενείς υποδομές επικοινωνίας χρησιμοποιούνται για video streaming προς ένα πλήθος συσκευών. Σε αυτήν την περίπτωση, η προσαρμογή της ευκρίνειας του βίντεο είναι σημαντική για να ικανοποιηθεί το εύρος ζώνης του καναλιού και η ευκρίνεια του αποκωδικοποιητή του βίντεο.



Εικόνα 4: Χωρική Κλιμάκωση

3.2. Ποιοτική Κλιμάκωση (*Fidelity/SNR/Quality Scalability*)

Για να έχουμε κλιμάκωση στην ποιότητα (Εικόνα 5) πρέπει να έχουμε και κλιμάκωση στο εύρος ζώνης (bit rate). Πρέπει επίσης να αναφερθεί ότι για συγκεκριμένες εφαρμογές αρχειοθέτησης βίντεο, όπως η βιομηχανία ψηφιακού σινεμά, η εξερεύνηση του διαστήματος ή κάποιες ιατρικές εφαρμογές, πρέπει να υποστηρίζεται η δυνατότητα κωδικοποίησης χωρίς απώλειες (lossless). Η κλιμάκωση στην παρεχόμενη ποιότητα είναι πολύ χρήσιμη όταν η μετάδοση λαμβάνει χώρα σε μη αξιόπιστα δίκτυα, όπου οι αλλαγές στο εύρος ζώνης μπορούν να αντιμετωπιστούν αποτελεσματικά αλλάζοντας κατά τη μετάδοση (on the fly) την ποιότητα του μεταδιδόμενου βίντεο. Μία άλλη εφαρμογή είναι η διανομή περιεχομένου, όπου ο χρήστης χρεώνεται ανάλογα με την ποιότητα που επιθυμεί και έτσι με την κλιμακωτή κωδικοποίηση από μια ακολουθία μπορούν να προκύψουν τα κατάλληλα επίπεδα ποιότητας που θα μεταδοθούν στους αντίστοιχους χρήστες.

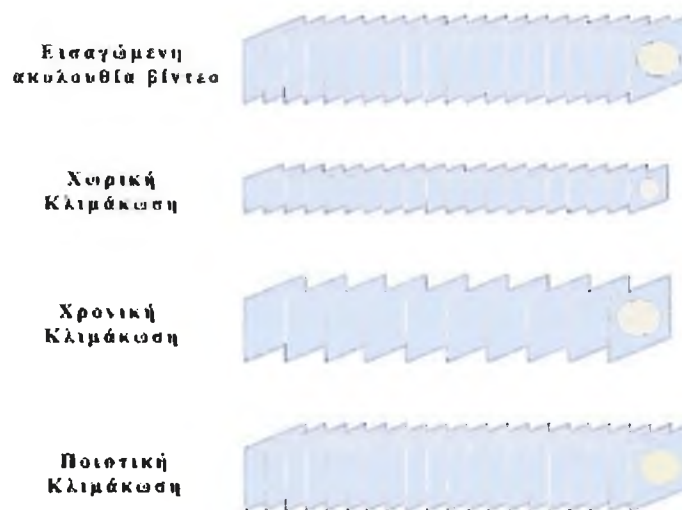


Εικόνα 5: Ποιοτική Κλιμάκωση

3.3. Χρονική Κλιμάκωση (*Temporal Scalability*)

Η χρονική κλιμάκωση (Εικόνα 6) της ακολουθίας βίντεο ισοδυναμεί με τη δυνατότητα κλιμάκωσης του ρυθμού πλαισίων (*frame rate*). Ο ρυθμός πλαισίων του βίντεο αντιστοιχεί στον αριθμό των πλαισίων που προβάλλονται στη διάρκεια ενός δευτερολέπτου. Έτσι, αλλάζοντας το ρυθμό των πλαισίων στην πραγματικότητα αλλάζουμε την ποιότητα του βίντεο που αντιλαμβάνεται το ανθρώπινο μάτι. Ο πιο συνηθισμένος ρυθμός πλαισίων είναι 30 Hz. Η κλιμακωτή κωδικοποίηση βίντεο πρέπει να παρέχει τουλάχιστον τέσσερα επίπεδα χρονικής κλιμάκωσης με ρυθμό αποκωδικοποίησης πλαισίων έως και 60 Hz.

Η κύρια εφαρμογή της χρονικής κλιμάκωσης είναι η παραγωγή και διανομή πολυκάναλου περιεχομένου, όπου η ίδια ακολουθία βίντεο προορίζεται για ένα πλήθος συσκευών που υποστηρίζουν διαφορετικές χρονικές αναλύσεις. Για παράδειγμα, 7.5 Hz, 15 Hz, 30 Hz, και 60 Hz πρέπει να υποστηρίζονται ώστε να εξυπηρετηθούν ποικίλοι χρήστες, από συσκευές στούντιο μέχρι βίντεο κατά απαίτηση (*video on demand*) σε κινητά τηλέφωνα με μικρές επεξεργαστικές ικανότητες. Επιπλέον, η χρονική κλιμάκωση μπορεί να χρησιμοποιηθεί για να εξομοιώσει λειτουργίες *fast forward/backward* που συναντούμε στις αναλογικές συσκευές, όπως τα βίντεο. Τέλος, η χρονική κλιμάκωση βρίσκει εφαρμογή σε εφαρμογές παρακολούθησης, όπου οι κάμερες καταγράφουν στατικές εικόνες και έτσι βίντεο υψηλού χρονικού ρυθμού πλαισίων είναι απαραίτητο μόνο έπειτα από κάποιο συναγερμό.



Εικόνα 6: Χωρική, Χρονική και Ποιοτική Κλιμάκωση

3.4. Κλιμάκωση στην Πολυπλοκότητα (*Complexity Scalability*)

Μια άλλη σημαντική λειτουργία είναι η κλιμάκωσης στην πολυπλοκότητα. Είναι επιθυμητό η πολυπλοκότητα αποκωδικοποίησης να μεταβάλλεται ανάλογα με την χρονική και χωρική ανάλυση. Επίσης, ανάλογα με τους διαθέσιμους πόρους στην πλατφόρμα κωδικοποίησης θα πρέπει να συμβαίνει κατά τη μετάδοση (on the fly) προσαρμογή του εύρους ζώνης, ώστε να επιλέγεται η υποακολουθία που οδηγεί σε μικρής πολυπλοκότητας αποκωδικοποίηση. Η πολυπλοκότητα σχετίζεται με αλγοριθμικά χαρακτηριστικά και με την πλατφόρμα υλοποίησης, οπότε για να έχουμε αξιόπιστα αποτελέσματα θα πρέπει να χρησιμοποιηθούν σωστά μοντέλα πολυπλοκότητας.

Σημαντικές εφαρμογές της κλιμάκωσης στην πολυπλοκότητα έχουμε στην περιοχή της ασύρματης μετάδοσης βίντεο πραγματικού χρόνου (video streaming) σε κινητές συσκευές, όπως κινητά τηλέφωνα και PDA. Επιπλέον, ανάλογα με την εφαρμογή, μπορεί να είναι απαραίτητο η κωδικοποίηση να πληρεί ορισμένα όρια πολυπλοκότητας. Αυτό είναι σημαντικό σε σενάρια όπου οι συσκευές κωδικοποίησης διαμοιράζονται στο δίκτυο, όπως στις εφαρμογές παρακολούθησης.

Κεφάλαιο 3^ο – Βίντεο πάνω από ασύρματα δίκτυα

Ε/ΘΡΚΗ Π.Α.Π. ΝΑΥΠΑΚΤΟΥ

1. Ετερογενή Ασύρματα Δίκτυα

1.1. Χαρακτηριστικά ετερογενών ασύρματων δικτύων

Μέχρι πρόσφατα, η σχεδίαση και η ανάπτυξη δικτύων επικοινωνιών βασιζόταν στο διαχωρισμό των ασύρματων και των ενσύρματων τεχνολογιών. Ο κάθε διαχειριστής ενός δικτύου ήταν υπεύθυνος για τον έλεγχο, τη διαχείριση και τη διάθεση των υπηρεσιών στους συνδρομητές ενός δικτύου. Σήμερα, υπό το φως των νέων τεχνολογικών εξελίξεων, αυτή η φιλοσοφία έχει ξεπεραστεί. Η έκρηξη στις τερματικές συσκευές κατέστησε δυνατή την παροχή υποστήριξης και πρόσβασης σε οποιασδήποτε δίκτυο, ενσύρματο ή ασύρματο από μια μόνο συσκευή. Επιπλέον, οι συνδρομητές επιθυμούν να κινούνται ελεύθερα μεταξύ των ετερογενών δικτύων και ταυτόχρονα να έχουν πρόσβαση σε υπηρεσίες ανεξαρτήτως της τεχνολογίας πρόσβασης. Η υλοποίηση μιας τέτοιας εξέλιξης περιλαμβάνει τη σύγκλιση των τεχνολογιών, την ολοκλήρωση των υπηρεσιών και το κεντρικό ολοκληρωμένο έλεγχο κάθε ενσύρματου και ασύρματου δικτύου με γνώμονα το χρήστη μέσω κάθε είδους τερματικής συσκευής. Τα ετερογενή ασύρματα δίκτυα είναι προφανές ότι διαφέρουν στον ρυθμό μετάδοσης, στο κόστος πρόσβασης, στην εμβέλεια της μετάδοσης, κ.α.. Για παράδειγμα, ο ρυθμός μετάδοσης στο 3G δίκτυο είναι πάνω από 2 Mbps ενώ στο 802.11b ασύρματο δίκτυο είναι 11Mbps και στο 802.11g είναι 54 Mbps. Η σύγκλιση των ετερογενών ασύρματων τεχνολογιών με το ευρυζωνικό δίκτυο κορμού IP θα επιτρέψει στους χρήστες-συνδρομητές την πρόσβαση σε νέες υπηρεσίες μέσω διαφορετικών δικτύων πρόσβασης, με τη χρήση ποικίλων τερματικών συσκευών. Από την πλευρά του δικτύου, μια τέτοια ενοποίηση μπορεί να πραγματοποιηθεί με την ύπαρξη τερματικών συσκευών, πρωτοκόλλων και αλγορίθμων που εξασφαλίζουν μεταγωγή μεταξύ των ετερογενών δικτύων και μηχανισμών μεταφοράς και διαθεσιμότητας. Τα κυριότερα πλεονεκτήματα που προσφέρει η διαλειτουργικότητα με βάση ένα All-IP δίκτυο κορμού είναι τα ακόλουθα:

- Οικονομικά οφέλη ιδιοκτησίας και διαχείρισης. Οι πάροχοι στα δίκτυα είναι υποχρεωμένοι να διαχειρίζονται μόνο ένα δίκτυο.
- Οικονομικά οφέλη για τη μετάδοση. Το κόστος μετάδοσης των πακέτων στο δίκτυο IP είναι χαμηλότερο σε σχέση με οποιαδήποτε άλλη λύση.
- Εξασφάλιση για το μέλλον. Είναι σχεδόν βέβαιο πως η τεχνολογία των μελλοντικών δικτύων κορμού θα εξαρτάται αποκλειστικά και μόνο από το πρωτόκολλο IP για τη

μεταφορά τόσο δεδομένων όσο και φωνής. Ένα δίκτυο πρόσβασης σχεδιασμένο με βάση το IP επιτυγχάνει ομαλή διασύνδεση με το δίκτυο κορμού IP και επιπλέον αποδοτικότερη διαχείριση των πόρων.

- Ομαλή σύγκλιση όλων των ετερογενών ασύρματων τεχνολογιών πρόσβασης.
- Αύξηση της διαθέσιμης χωρητικότητας. Το πρωτόκολλο IP επιτρέπει την ευκολότερη και τη λιγότερη αύξηση της χωρητικότητας του δικτύου μετάδοσης.
- Πλούσιες σε περιεχόμενο υπηρεσίες. Ο συνδυασμός κλήσεων φωνής/πολυμέσων εξασφαλίζει μια ισχυρή και λειτουργική πλατφόρμα για τη δημιουργία καινούργιων υπηρεσιών.

2. Βίντεο πάνω από ετερογενή ασύρματα δίκτυα

Λόγω της σύγκλισης του διαδικτύου με τα ασύρματα δίκτυα, η ετερογένεια των δικτύων στην παράδοση του βίντεο φθάνει ολόκληρο σε ένα νέο επίπεδο, έτσι είναι σημαντικό να μελετηθούν τα ζητήματα QoS για το βίντεο πάνω στα ετερογενή ασύρματα δίκτυα. Ο στόχος αυτής της σύγκλισης είναι να προσφερθούν υπηρεσίες πολυμέσων σε ποικίλες ετερογενείς τεχνολογίες πρόσβασης (πχ WiMax, WLANs, 3G και 4G συστήματα) ικανοποιώντας τις ανάγκες μιας επιχείρησης και των δημόσιων δικτύων, οπουδήποτε και οποτεδήποτε. Σαν αποτέλεσμα, η έννοια όλων των IP δικτύων οδηγεί αυτές τις εξελίξεις και ερευνά τον αντίκτυπό τους στην παροχή υπηρεσιών πολυμέσων πραγματικού χρόνου και μη πραγματικού χρόνου, όπως, της κινητής τηλεφωνίας, συνόδων πολυμέσων, ή της κινητής πρόσβασης στο διαδίκτυο στα υπάρχοντα περιβάλλοντα. Τα σφάλματα του ασύρματου καναλιού είναι μια από τις σημαντικότερες προκλήσεις για την ασύρματη μετάδοση βίντεο, επειδή πρέπει να αντιμετωπίσει την έλλειψη εγγυήσεων QoS, συμπεριλαμβανομένης της απώλειας πακέτων. Ένα επιτυχές ασύρματο σύστημα αναπαραγωγής βίντεο σε πραγματικό χρόνο πρέπει να είναι σε θέση να αναπαράγει το βίντεο στους ετερογενείς χρήστες πέρα από τις χρονικά μεταβαλλόμενες ασύρματες συνδέσεις επικοινωνίας. Στα παραδοσιακά IP δίκτυα οι απώλειες πακέτων προκαλούνται κυρίως λόγω της συμφόρησης. Η βέλτιστη στρατηγική είναι να μειωθεί ο ρυθμός μετάδοσης όταν αυξάνεται το ποσοστό απώλειας πακέτων, ενώ ένα ασύρματο περιβάλλον πάσχει χαρακτηριστικά από τις απώλειες πακέτων που προκαλούνται από τα φυσικά λάθη μετάδοσης. Σε αυτήν την περίπτωση δεν υπάρχει καμία ανάγκη για την αποφυγή της συμφόρησης και των αναμεταδόσεων. Η περιττή μεταφορά δεδομένων θα μπορούσε να χρησιμοποιηθεί για να διευκολύνει την ανάκτηση από τις απώλειες πακέτων.

Αυτές οι απώλειες μπορεί να προκαλούνται από πολλές αιτίες οι οποίες μπορεί να κατηγοριοποιηθούν σε:

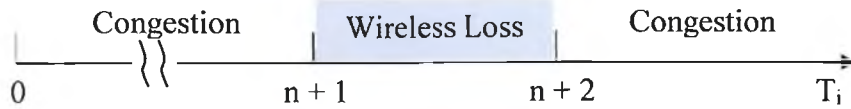
- Φυσική εξασθένηση: οι απώλειες πακέτων συμβαίνουν σύμφωνα με τα χαρακτηριστικά του ασύρματου μέρους του δικτύου λόγω παρεμβολών, θορύβου, εξασθένησης, κλπ. .
- Συμφόρηση: οι απώλειες πακέτων συμβαίνουν στο δίκτυο λόγω του περιορισμένου εύρους ζώνης και στα δύο μέρη του δικτύου.
- Κινητικότητα και κινητικότητα μεταξύ δικτύων (vertical handoff): οι απώλειες πακέτων συμβαίνουν κατά την κίνηση των χρηστών μεταξύ των ετερογενών ασύρματων δικτύων που χαρακτηρίζονται από το διαφορετικό αριθμό εξυπηρετούμενων χρηστών.

Ο αντίκτυπος όλων αυτών των παραγόντων επηρεάζει αρνητικά την απόδοση του δικτύου με συνέπεια το μικρό QoS. Επομένως είναι απαραίτητο να τεθεί ένας ευφυής μηχανισμός που να μπορεί να ανιχνεύσει τις απώλειες και να διαφοροποιήσει τις απώλειες που προκαλούνται από τη συμφόρηση και από τη φυσική εξασθένηση.

2.1. Συμφόρηση και λάθη στο φυσικό επίπεδο σε ασύρματα δίκτυα

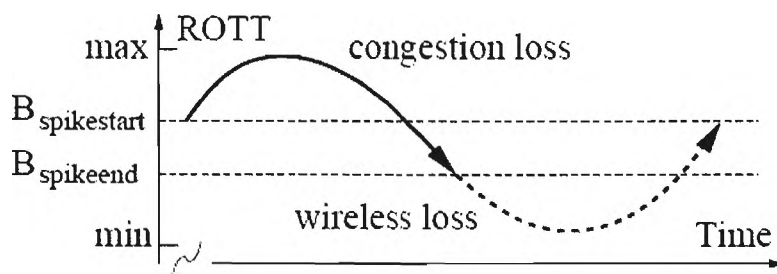
Τα τελευταία χρόνια πολλές έρευνες έχουν εκδοθεί για να αντιμετωπίσουν το πρόβλημα της λάθος διάκρισης μεταξύ των απωλειών συμφόρησης και των ασύρματων απωλειών. Οι αλγόριθμοι αυτοί προσπαθούν να εξάγουν χαρακτηριστικά από τα πακέτα από το IP επίπεδο και στη συνέχεια γίνεται μια εκτίμηση του ποσοστού που οφείλεται σε απώλειες πακέτων και το ποσοστό που οφείλεται σε λάθη του φυσικού καναλιού. Κάποιοι αλγόριθμοι οι οποίοι διακρίνουν το είδος των απωλειών είναι ο Bias, ο Spike και ο ZigZag.

- Ο **Biaz** χρησιμοποιεί τον χρόνο μεταξύ της άφιξης των πακέτων (inter-arrival time) για να διακρίνει το είδος των απωλειών. Όπως και στην Εικόνα 7 ο αλγόριθμος λειτουργεί ως εξής: Το T_{\min} υποδηλώνει τον ελάχιστο χρόνο άφιξης των πακέτων που έχει παρατηρηθεί αυτή τη στιγμή στον δέκτη κατά τη διάρκεια της σύνδεσης. Το P_i είναι το τελευταίο πακέτο που έχει φτάσει πριν χαθεί κάποιο πακέτο ενώ το P_{i+1+n} είναι το πρώτο πακέτο που έφτασε μετά από την απώλεια. Το n είναι ο αριθμός των χαμένων πακέτων. Εάν ισχύει η συνθήκη $(n+1)T_{\min} \leq T_i < (n+2)T_{\min}$ τότε τα n πακέτα χάθηκαν λόγω λαθών στο ασύρματο κανάλι.



Εικόνα 7: Ο αλγόριθμος Bias

- Σε αντίθεση με τον Bias ο Spike χρησιμοποιεί τον χρόνο που κάνει το πακέτο από τον αποστολέα στον παραλήπτη (Relative One-way Trip Time - ROTT). Το ROTT χρησιμοποιείται για να αναγνωρίσει την κατάσταση της σύνδεσης. Αν η σύνδεση είναι στην κατάσταση spike τότε οι απώλειες είναι λόγω συμφόρησης αλλιώς είναι λόγω του ασύρματου καναλιού. Όταν φτάσει ένα πακέτο i , αν η σύνδεση δεν είναι ήδη στη κατάσταση spike και το ROTT για το πακέτο i υπερβαίνει το $B_{spikestart}$, τότε ο αλγόριθμος μπαίνει στην κατάσταση spike αλλιώς αν η σύνδεση είναι σε κατάσταση spike και το ROTT για το πακέτο i είναι μικρότερο από το $B_{spikeend}$ τότε ο αλγόριθμος φεύγει από την κατάσταση spike. Όταν ο αποστολέας ανακαλύψει ένα λάθος εξαιτίας ενός κενού στην ακολουθία των πακέτων τότε αναγνωρίζει το λάθος ανάλογα με την κατάσταση που βρίσκεται, όπως φαίνεται και στην Εικόνα 8. Το $B_{spikestart}$ υπολογίζεται σύμφωνα με τον τύπο $B_{spikestart} = rott_{min} + \alpha * (rott_{max} - rott_{min})$ ενώ το $B_{spikeend}$ υπολογίζεται σύμφωνα με τον τύπο $B_{spikeend} = rott_{min} + \beta * (rott_{max} - rott_{min})$, όπου $\alpha \geq \beta$.



Εικόνα 8: Ο αλγόριθμος Spike

- Ο ZigZag χαρακτηρίζει τις απώλειες ως ασύρματες βασιζόμενος στις απώλειες n πακέτων και στη διαφορά μεταξύ του $rott_i$ και του $rott_{mean}$. Η απώλεια χαρακτηρίζεται ως ασύρματη αν: $(n = 1 \text{ ΚΑΙ } rott_i < rott_{mean} - rott_{dev})$ ή $(n = 2 \text{ ΚΑΙ } rott_i < rott_{mean} - rott_{dev} / 2)$ ή $(n = 3 \text{ ΚΑΙ } rott_i < rott_{mean})$ ή $(n > 3 \text{ ΚΑΙ } rott_i < rott_{mean} - rott_{dev} / 2)$ αλλιώς οι απώλειες

είναι λόγω συμφόρησης. Το $rott_{mean}$ υπολογίζεται σύμφωνα με τον τύπο $rott_{mean} = (1 - a) * rott_{mean} + a * rott$ ενώ το $rott_{dev}$ υπολογίζεται σύμφωνα με τον τύπο $rott_{dev} = (1 - 2a) * rott_{dev} + 2a * |rott - rott_{mean}|$. Όπου $a = 1/32$

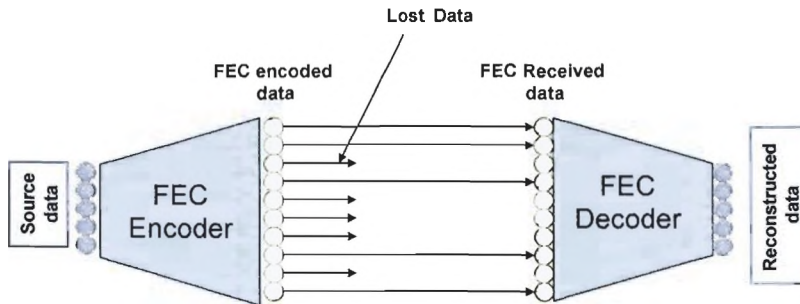
3. Τεχνικές βελτίωσης ποιότητας εξυπηρέτησης

Για να μειωθούν οι απώλειες αυτές και να βελτιωθεί η ποιότητα εξυπηρέτησης χρησιμοποιούνται κάποιες τεχνικές, όπως το Forward Error Correction (FEC) και οι προτεραιότητες (Priorities). Το FEC χρησιμοποιείται για τη διαχείριση των λαθών στο φυσικό επίπεδο, ενώ οι προτεραιότητες για τη διαχείριση των λαθών λόγω συμφόρησης.

3.1. Forward Error Correction (FEC)

Το FEC είναι μια μέθοδος που χρησιμοποιείται συνήθως για να χειριστεί τις απώλειες σε πραγματικού χρόνου επικοινωνίες. Οι τεχνικές FEC επιτρέπουν σε έναν δέκτη να διορθώνει τα λάθη ή τις απώλειες χωρίς περαιτέρω αλληλεπίδραση με τον αποστολέα. Ένας κώδικας (n, k) εξάλειψης λαθών μετατρέπει την ακολουθία k bits σε μια ομάδα κωδικοποιημένων στοιχείων n, έτσι ώστε οποιοδήποτε k των κωδικοποιημένων n στοιχείων μπορεί να χρησιμοποιηθεί για να αναδημιουργήσει τα αρχικά στοιχεία της πηγής. Συνήθως, τα πρώτα στοιχεία k σε κάθε ομάδα είναι ίδια με τα αρχικά στοιχεία πηγής k τα υπόλοιπα (n - k) στοιχεία αναφέρονται ως στοιχεία ισότητας (Εικόνα 9). Συνήθως, οι κώδικες FEC είναι σε θέση να διορθώσουν και τα λάθη και τις εξαλείψεις σε έναν φραγμό από n σύμβολα. Στη θεωρία κωδικοποίησης ένα λάθος ορίζεται ως ένα αλλοιωμένο σύμβολο σε μια άγνωστη θέση, ενώ ένα λάθος είναι ένα αλλοιωμένο σύμβολο σε μια γνωστή θέση. Στην περίπτωση της ροής πακέτων πολυμέσων, η ανίχνευση της απώλειας εκτελείται βασισμένη στους αριθμούς ακολουθίας στα RTP πακέτα (δηλ., κώδικες εξάλειψης). Ενώ το FEC μπορεί ενδεχομένως να μειώσει τα αντιληπτά αποτελέσματα των χαμένων πακέτων, αυξάνει τις απαιτήσεις του εύρους ζώνης που μπορούν στη συνέχεια να αυξήσουν το ποσοστό της απώλειας. Επομένως, είναι σημαντικό να επιλέξει, σε πραγματικό χρόνο, το κατάλληλο σχέδιο FEC για να παρασχεθεί η καλύτερη απόδοση στην εφαρμογή, επίσης το FEC από μόνο του δεν μπορεί να παρέχει την πλήρη αξιοπιστία. Ωστόσο, όταν συνδέεται με μια άλλη τεχνική, όπως το Αυτόματο Αίτημα Αναμετάδοσης (Automatic Retransmission request - ARQ) ή το interleaving, το καθιστά πιο αξιόπιστο και αποδοτικό σε μια απαραίτητη

εφαρμογή. Διάφορες έρευνες έχουν εκδοθεί οι οποίες μελετούν την μεταχείριση των χαμένων πακέτων ή/και των bit errors λόγω των λαθών μετάδοσης στα ασύρματα δίκτυα χρησιμοποιώντας το Forward Error Correction (FEC) [3].



Εικόνα 9: Διάγραμμα κωδικοποίησης – αποκωδικοποίησης δεδομένων FEC

Κεφάλαιο 4^ο – Εξομοίωση – Αποτελέσματα

Ο σκοπός της συγκεκριμένης πτυχιακής είναι να υλοποιήσει τους αλγόριθμους Biaz, Spike και ZigZag έτσι ώστε όταν παρατηρείται κάποια απώλεια ενός πλαισίου βίντεο ή και περισσότερων, να παίρνεται αμέσως απόφαση για το αν το πλαίσιο ή τα πλαίσια που χάθηκαν είναι λόγω της συμφόρησης που δημιουργείται στο δίκτυο ή είναι λάθη που δημιουργούνται εξαιτίας του ασύρματου καναλιού.

1. Εξομοιωτής

Στην συγκεκριμένη πτυχιακή έχει χρησιμοποιηθεί ο NS-2 (Network Simulator 2), επειδή είναι ανοιχτού κώδικα εξομοιωτής γραμμένος σε C++ με διεργμηνέα OTcl (Object Tool Command Language), είναι αξιόπιστος και χρησιμοποιείται ευρέως από την επιστημονική κοινότητα [14]. Για να γίνει η διάκριση μεταξύ των απωλειών χρησιμοποιούνται τρεις διαφορετικοί αλγόριθμοι, ο Biaz, ο Spike και ο ZigZag, όπου περιγράφονται παραπάνω. Όταν ανιχνεύεται κάποια απώλεια η οποία οφείλεται στο ασύρματο κανάλι τότε γίνεται η χρήση του μηχανισμού FEC. Για όσα πλαίσια χάθηκαν λόγω λαθών στο ασύρματο κανάλι τόσα πακέτα FEC στέλνονται στον παραλήπτη του βίντεο.

2. Τοπολογία Δικτύων

Στον NS-2 έχουν υλοποιηθεί δύο διαφορετικές τοπολογίες που μόνη τους διαφορά είναι το είδος της τεχνικής κίνησης που δημιουργείται και περιπλέκεται με τη κίνηση του βίντεο που μελετάμε. Στη μία γίνεται χρήση μιας πηγής παραγωγής πακέτων CBR (Constant Bit Rate) η οποία δημιουργεί με σταθερό ρυθμό UDP πακέτα τα οποία αποστέλλονται κάθε 'x' δευτερόλεπτα, ενώ στην άλλη γίνεται χρήση μιας πηγής Exponential η οποία παράγει UDP πακέτα με σταθερό ρυθμό ανά χρονικές περιόδους η οποίες δημιουργούνται από την εκθετική κατανομή. Και στις δύο τοπολογίες χρησιμοποιείται διαφορετικός ρυθμός μετάδοσης σε κάθε εξομοίωση του δικτύου, *πίνακας 2*.

Ρυθμοί Μετάδοσης (Mbps)

0,8	1	1,2	1,5	2
-----	---	-----	-----	---

Πίνακας 2: Ρυθμοί Μετάδοσης

2.1 Gilbert Elliot μοντέλο

Και στις δύο τοπολογίες έχει χρησιμοποιηθεί ένα μοντέλο λαθών που βασίζεται στο μοντέλο του Gilbert Elliot. Το μοντέλο αυτό χρησιμοποιείται για να εξομοιώσει τα φυσικά λάθη. Σύμφωνα με αυτό υπάρχουν δύο καταστάσεις στην αλυσίδα Markov, η καλή (Good) και η κακή (Bad) όπως φαίνεται και στην Εικόνα 10. Όταν το μοντέλο Gilbert βρίσκεται στην καλή κατάσταση τα λάθη συμβαίνουν μικρή πιθανότητα, p_G , ενώ όταν βρισκόμαστε στην κακή κατάσταση τότε τα λάθη συμβαίνουν με μεγάλη πιθανότητα, p_B . Επίσης η πιθανότητα να μεταβούμε στην κακή κατάσταση από την καλή είναι p_{GB} ενώ από την κακή προς την καλή είναι p_{BG} . Οι πιθανότητες να είμαστε σε σταθερή κατάσταση όταν βρισκόμαστε στην καλή και στην κακή κατάσταση είναι $\pi_G = \frac{p_{BG}}{p_{BG} + p_{GB}}$ και $\pi_B = \frac{p_{GB}}{p_{BG} + p_{GB}}$ αντίστοιχα. Το μέσο ποσοστό απώλειας πακέτων που παράγεται από αυτό το μοντέλο είναι $p = p_G \pi_G + p_B \pi_B$.



Εικόνα 10: Καταστάσεις αλυσίδας Markov.

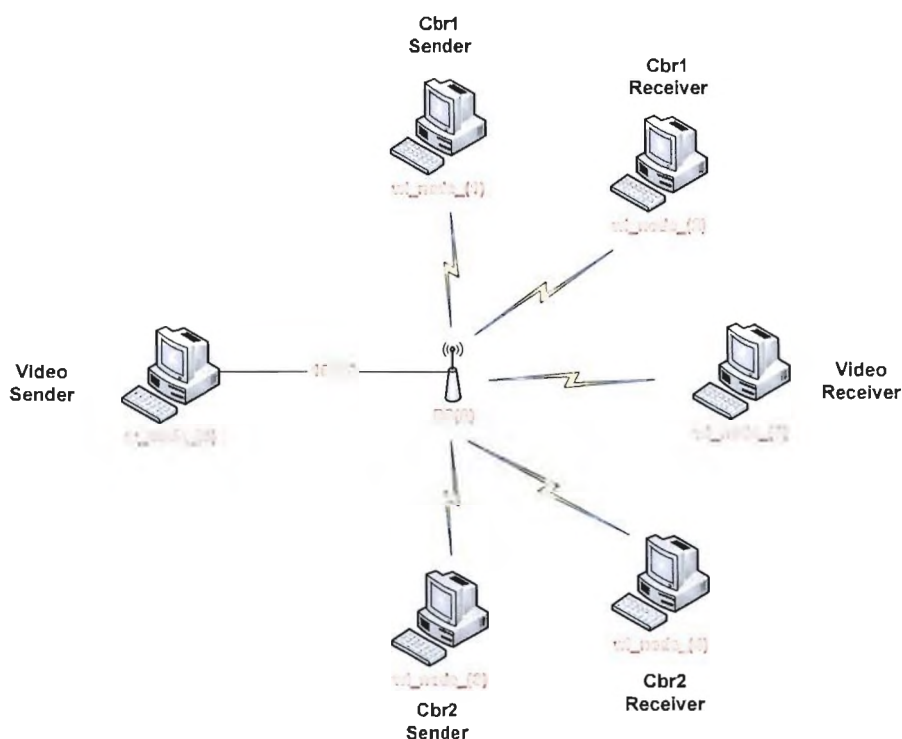
Και στις δύο τοπολογίες πέρα από τη χρήση διαφορετικών ρυθμών μετάδοσης σε κάθε εξομοίωση έχει χρησιμοποιηθεί ακόμα μία παράμετρος, η πιθανότητα, p_{BB} , να είμαστε στην κακή κατάσταση και να ξαναπάμε σε αυτήν. Για κάθε ρυθμό μετάδοσης γίνεται χρήση όλων των πιθανοτήτων, πίνακας 3.

p_{BB}:	0,85	0,90	0,94
-----------------------------	------	------	------

Πίνακας 3: Πίνακας πιθανοτήτων

2.2. Τοπολογία με χρήση CBR Traffic

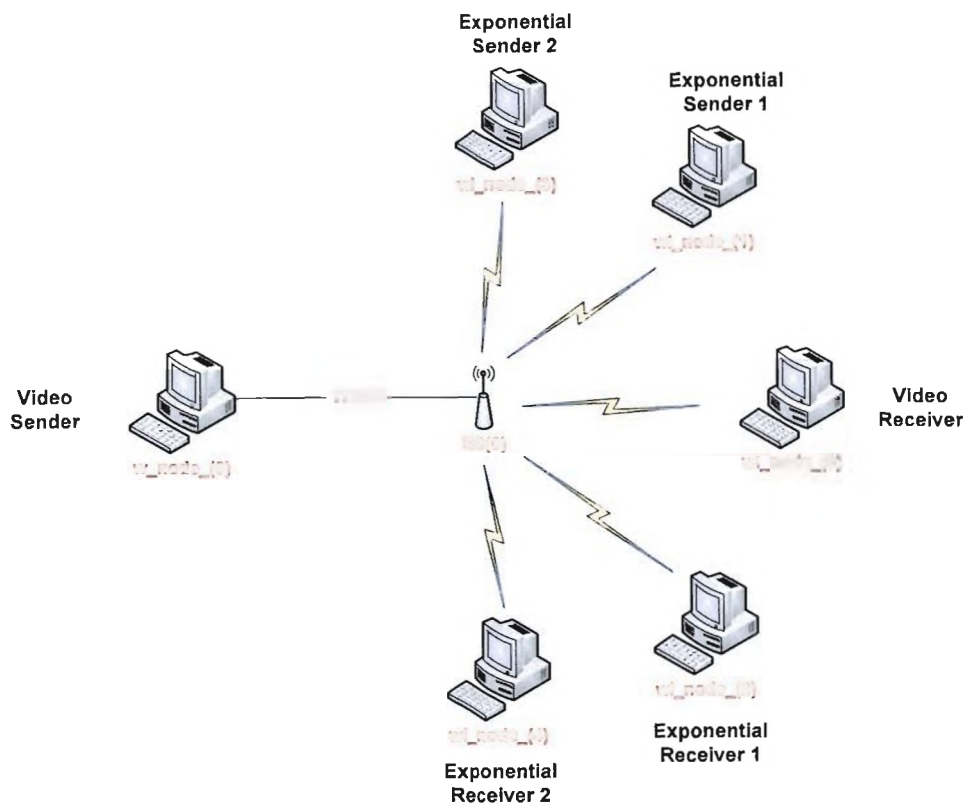
Σε αυτή την τοπολογία υπάρχει ένας ασύρματος κόμβος (Video Sender), ο οποίος συνδέεται ασύρματα με ένα κανάλι χωρητικότητας 100Mbps και καθυστέρησης 1ms με ένα access point, ο οποίος στέλνει το βίντεο σε ένα ασύρματο κόμβο (Video Receiver). Για να δημιουργηθεί συμφόρηση στο ασύρματο κανάλι χρησιμοποιούνται δύο ασύρματοι κόμβοι (CBR 1-2 Sender) οι οποίοι στέλνουν CBR κίνηση σε άλλους δύο ασύρματους κόμβους (CBR 1-2 Receiver). (Εικόνα 11)



Εικόνα 11: Τοπολογία δικτύου με χρήση CBR Traffic

2.3. Τοπολογία με χρήση Exponential Traffic

Σε αυτή την τοπολογία επίσης υπάρχει ένας ασύρματος κόμβος (Video Sender), ο οποίος συνδέεται ασύρματα με ένα κανάλι χωρητικότητας 100Mbps και καθυστέρησης 1ms με ένα access point, ο οποίος στέλνει το βίντεο σε ένα ασύρματο κόμβο (Video Receiver). Για να δημιουργηθεί συμφόρηση στο ασύρματο κανάλι χρησιμοποιούνται δύο ασύρματοι κόμβοι (Exponential 1-2 Sender) οι οποίοι στέλνουν Exponential κίνηση σε άλλους δύο ασύρματους κόμβους (Exponential 1-2 Receiver). (Εικόνα 12)



Εικόνα 12: Τοπολογία δικτύου με χρήση Exponential Traffic

2.4. Αποτελέσματα

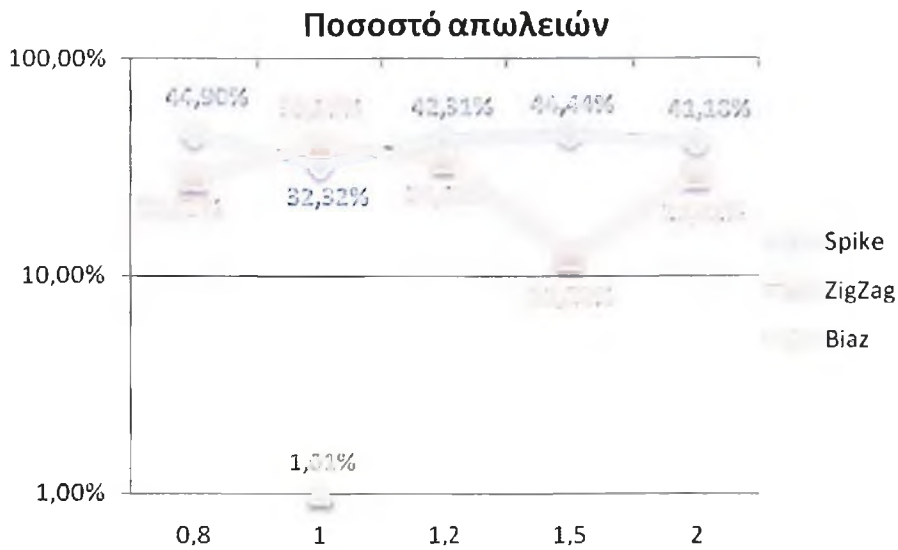
Στις παρακάτω γραφικές παραστάσεις γίνεται σύγκριση των απωλειών των πακέτων λόγω λαθών στο φυσικό κανάλι και στους τρεις αλγορίθμους σε όλα τα διαφορετικά σενάρια που εξομοιώθηκαν και στις δύο τοπολογίες. Επίσης γίνεται σύγκριση του overhead που δημιουργείται από την αποστολή FEC πακέτων σε κάθε σενάριο.

2.4.1 Τοπολογία με χρήση CBR κίνησης

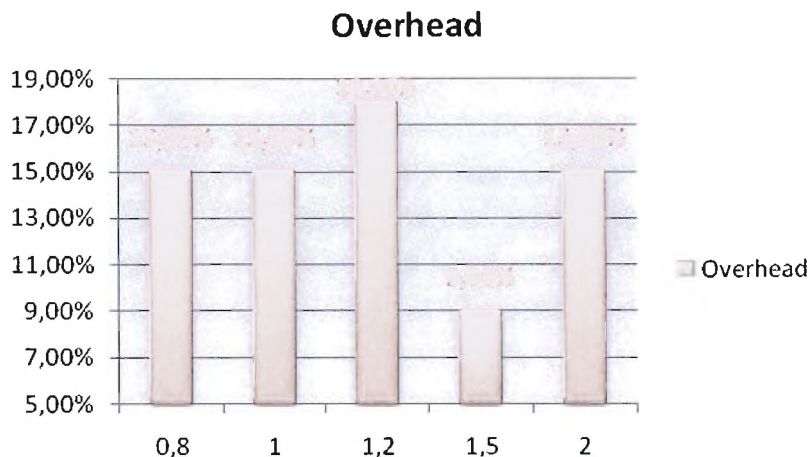
Τα γραφήματα 1, 2 και 3 αφορούν τα σενάρια που έχουν πιθανότητα να είμαστε στη κακή κατάσταση και να επιστρέψουμε σε αυτήν 0,85. Κρατώντας σταθερή την πιθανότητα μεταβάλλουμε την επιπλέον κίνηση που στέλνεται στο δίκτυο.

Στο γράφημα 1 παρατηρούμε ότι όσο αυξάνεται η κίνηση στο δίκτυο ο αλγόριθμος Spike θεωρεί ότι χάθηκαν περισσότερα πακέτα εξαιτίας των λαθών στο φυσικό κανάλι ενώ ο Bias

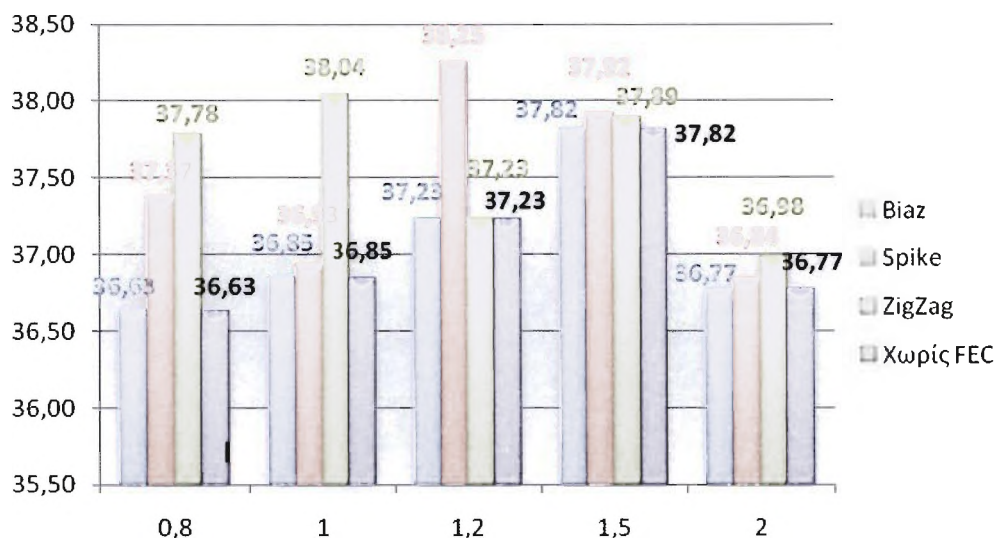
θεωρεί πως οι απώλειες δεν οφείλονται στο φυσικό κανάλι. Στο *γράφημα 2* φαίνεται το overhead που δημιουργείται στο δίκτυο εξαιτίας της αποστολής πακέτων FEC και όπως μπορούμε να δούμε είναι σχεδόν ίδιο ενώ στο *γράφημα 3* φαίνεται για κάθε αλγόριθμο η ποιότητα του βίντεο μετά τη χρήση του FEC και χωρίς τη χρήση. Όταν η κίνηση στο δίκτυο είναι 1.5 Mbps τότε ο ZigZag θεωρεί ότι χάθηκαν λιγότερα πακέτα από τον Spike και το overhead του δικτύου είναι το ελάχιστο που παρατηρείται σε όλα τα σενάρια με πιθανότητα pBB 0,85. Κατά μέσο όρο και στα πέντε σενάρια ο Spike θεωρεί ότι το ποσοστό των απωλειών στο φυσικό κανάλι είναι 41,03%, ο ZigZag θεωρεί ότι είναι 28,84% και ο Biaz μόνο 0,2% ενώ το overhead είναι 14,4%. Σύμφωνα με το *γράφημα 3* παρατηρούμε ότι η καλύτερη ποιότητα του βίντεο παρατηρείται όταν η κίνηση είναι 1.5Mbps και αυτό οφείλεται στο ότι στο σενάριο αυτό στάλθηκαν περισσότερα πακέτα FEC.



Γράφημα 1: Εκτίμηση ποσοστού απωλειών σε σχέση με τις συνολικές απώλειες - Πιθανότητα pBB 0,85



Γράφημα 2: Overhead - Πιθανότητα pBB 0,85

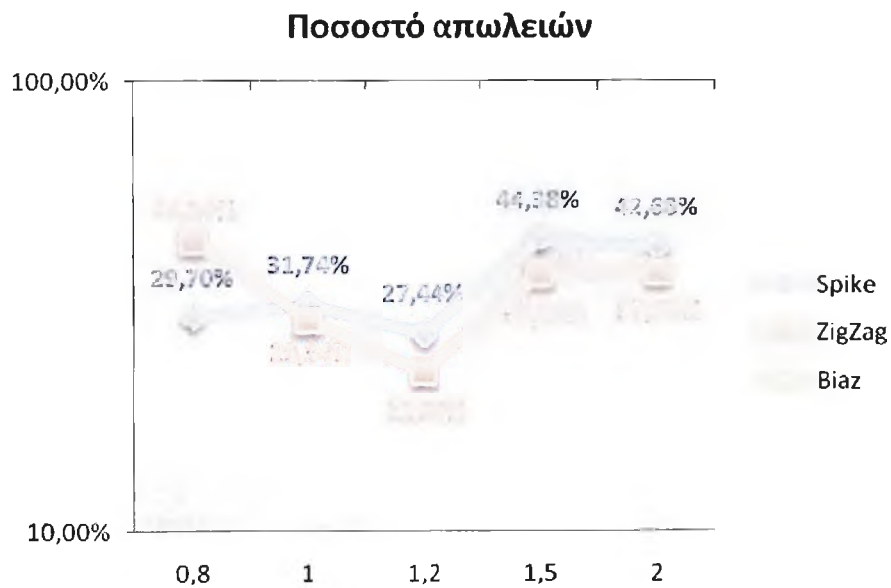


Γράφημα 3: PSNR Αλγορίθμων - Πιθανότητα pBB 0,85

Τα γραφήματα 4, 5 και 6 αφορούν τα σενάρια με πιθανότητα pBB 0,90. Κρατώντας σταθερή την πιθανότητα μεταβάλλουμε την επιπλέον κίνηση που στέλνεται στο δίκτυο.

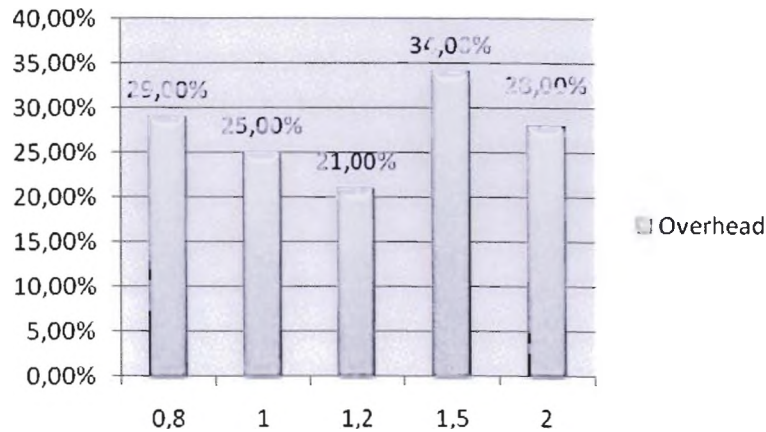
Στο γράφημα 4 φαίνεται πως ο Biaz δεν θεωρεί ότι οι απώλειες οφείλονται στο φυσικό κανάλι. Όταν η κίνηση αυξήθηκε από 0,8 Mbps σε 1,2 ο ZigZag είχε μια ραγδαία πτώση στο ποσοστό των απωλειών σε αντίθεση με τον Spike που έπεσε κατά 2%. Όταν όπως η κίνηση αυξήθηκε στα 2 Mbps τότε και οι δύο αλγόριθμοι εντόπισαν περισσότερα πακέτα αλλά ο Spike όμως από το 1 Mbps και μετά εντόπισε περισσότερα από τον Spike. Επειδή από 0,8 μέχρι τα 1,2 Mbps έχουμε λίγες απώλειες το overhead είναι μικρό ενώ όταν αυξάνεται η

κίνηση το overhead είναι υψηλότερο. Κατά μέσο όρο και στα πέντε σενάρια ο Spike θεωρεί ότι το 35,2% των πακέτων χάθηκε εξαιτίας των λαθών στο φυσικό κανάλι ενώ ο ZigZag θεωρεί το 34,2%. Επίσης το overhead που δημιουργείται εξαιτίας της αποστολής των FEC πακέτων είναι 27,4% κατά μέσο όρο. Στο γράφημα 6 παρατηρούμε πως όσο αυξάνεται η κίνηση στο δίκτυο δεν παρατηρούνται σημαντικές αλλαγές στην ποιότητα του βίντεο. Στην περίπτωση που η κίνηση είναι 1.5Mbps το PSNR δεν υπολογίζεται γιατί κατά τη λήψη του βίντεο χάθηκε ο πρώτο πακέτο και δεν ήταν δυνατό να υπολογιστεί το PSNR με το πρόγραμμα που χρησιμοποιήθηκε.

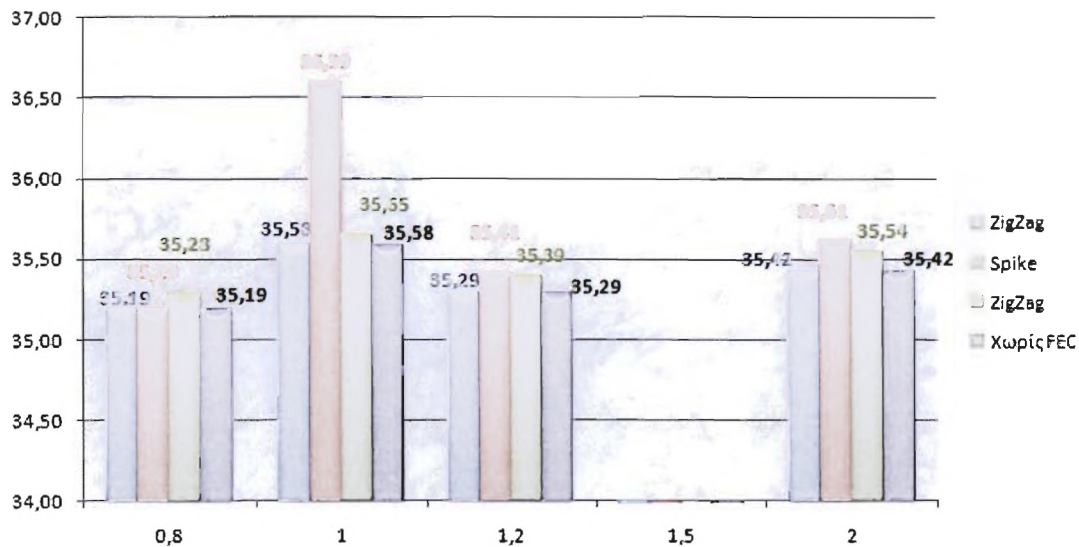


Γράφημα 4: Εκτίμηση ποσοστού απωλειών σε σχέση με τις συνολικές απώλειες -
Πιθανότητα pBB 0,90

Overhead



Γράφημα 5: Overhead - Πιθανότητα pBB 0,90



Γράφημα 6: PSNR Αλγορίθμων - Πιθανότητα pBB 0,90

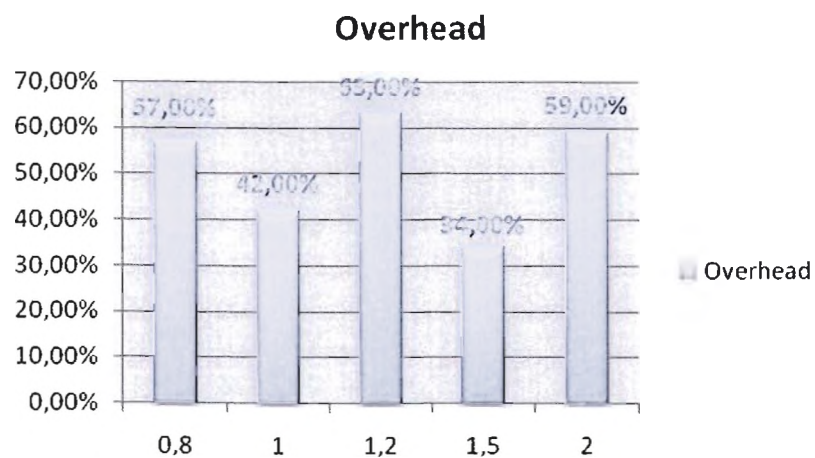
Τα γραφήματα 7, 8 και 9 αφορούν τα σενάρια με πιθανότητα pBB 0,94. Κρατώντας σταθερή την πιθανότητα μεταβάλλουμε την επιπλέον κίνηση που στέλνεται στο δίκτυο.

Στο γράφημα 7 φαίνεται πως ο Βιάζ πάλι δεν εντοπίζει καμία απώλεια εξαιτίας των λαθών στο φυσικό κανάλι. Ο Spike από τα 1 μέχρι τα 2 Mbps κίνησης εντοπίζει συνέχεια περισσότερες απώλειες από τον ZigZag ο οποίος εντοπίζει περισσότερα μόνο όταν η κίνηση είναι 0,8 Mbps. Στο γράφημα 8 φαίνεται το overhead που δημιουργείται από την αποστολή FEC πακέτων σε κάθε σενάριο ενώ στο γράφημα 9 φαίνεται η ποιότητα του βίντεο πριν και μετά τη χρήση του FEC σε κάθε αλγόριθμο. Οι αυξομειώσεις που φαίνονται οφείλονται στις

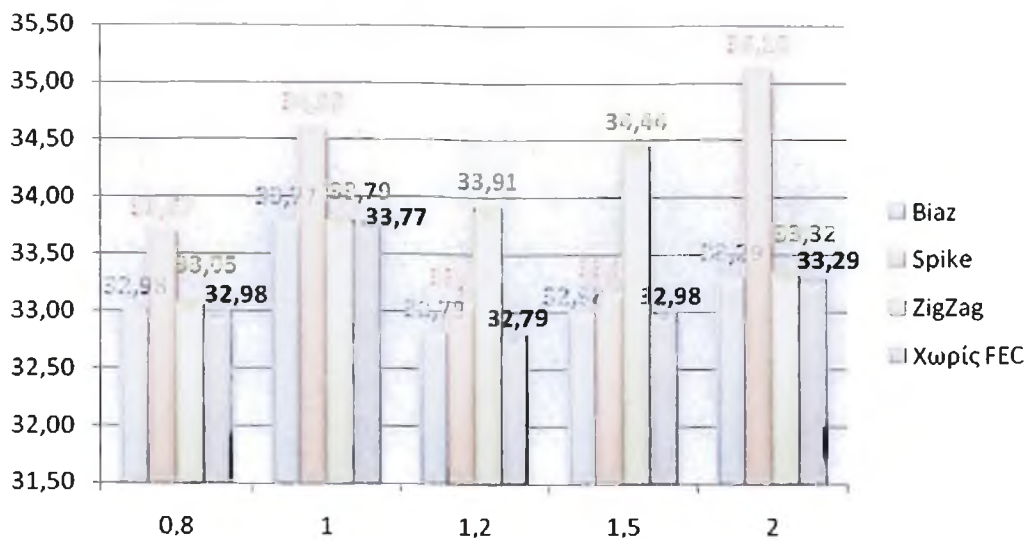
μεταπτώσεις του ποσοστού των απωλειών των πακέτων των αλγορίθμων. Κατά μέσο όρο ο Spike θεωρεί ότι χάθηκαν τα 33,6% των πακέτων εξαιτίας των λαθών στο φυσικό κανάλι, ο ZigZag θεωρεί το 27,74% των πακέτων ενώ το overhead που δημιουργείται είναι 51%.



Γράφημα 7: Εκτίμηση ποσοστού απωλειών σε σχέση με τις συνολικές απώλειες - Πιθανότητα pBB 0,94

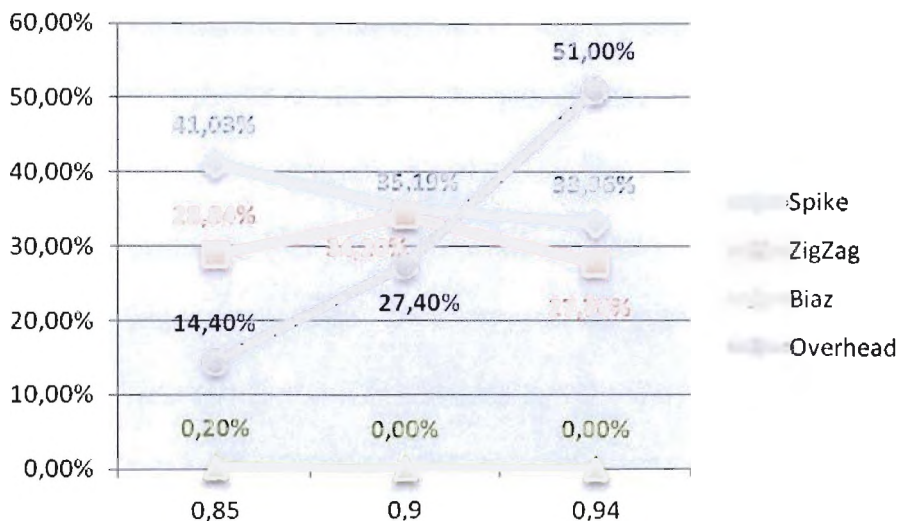


Γράφημα 8: Overhead - Πιθανότητα pBB 0,94



Γράφημα 9: PSNR Αλγορίθμων - Πιθανότητα pBB 0,94

Στο γράφημα 10 φαίνονται οι διαφορές που παρατηρούνται κατά τη μεταβολή της πιθανότητας pBB. Όταν αυξάνεται η πιθανότητα pBB παρατηρούμε πως στους αλγόριθμους Spike και Biaz η μέση τιμή των χαμένων πακέτων μειώνεται και το overhead του δικτύου αυξάνεται. Σε αντίθεση με αυτούς αλγόριθμους ο ZigZag όταν η πιθανότητα pBB αυξάνεται από 0,85 σε 0,9 τότε αυξάνεται και η μέση τιμή των χαμένων πακέτων ενώ όταν η πιθανότητα αυξάνεται από 0,9 σε 0,94 τότε η μέση τιμή ελαττώνεται.

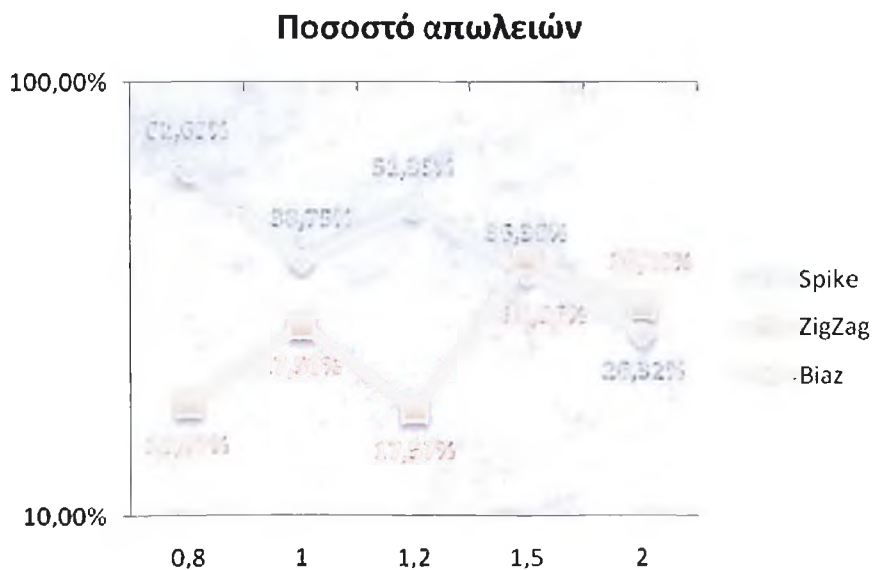


Γράφημα 10: Μεταβολή της πιθανότητας pBB

2.4.2 Τοπολογία με χρήση Exponential κίνησης

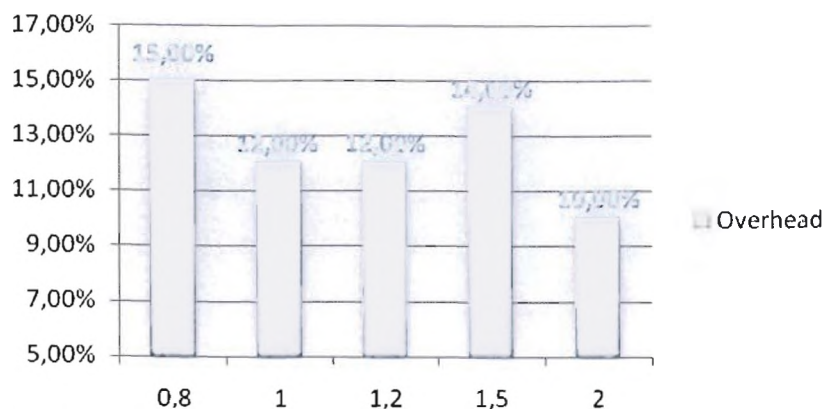
Τα γραφήματα 11, 12 και 13 αφορούν τα σενάρια που έχουν πιθανότητα να είμαστε στη κακή κατάσταση και να επιστρέψουμε σε αυτήν 0,85. Κρατώντας σταθερή την πιθανότητα μεταβάλλουμε την επιπλέον κίνηση που στέλνεται στο δίκτυο.

Στο γράφημα 11 φαίνεται πως ο Bias πως σε κανένα σενάριο δεν κατάφερε να αναγνωρίσει κάποια απώλεια που οφείλεται σε λάθη στο φυσικό κανάλι σε σύγκριση με τους άλλους αλγορίθμους. Από τα 0,8 στα 1,2 Mbps exponential κίνησης ο Spike αναγνώρισε περισσότερα πακέτα από τον ZigZag ενώ από τα 1,5 έως τα 2 Mbps ο ZigZag εντόπισε λίγο περισσότερα από τον Spike. Κατά μέσο όρο σε όλα τα σενάρια ο Spike εντόπισε πως το 43,09% των απωλειών οφειλόταν σε λάθη στο ασύρματο κανάλι ενώ ο ZigZag εντόπισε το 26,47%. Στο γράφημα 12 φαίνεται το overhead που δημιουργείται εξαιτίας της αποστολής των FEC πακέτων ενώ στο γράφημα 13 φαίνεται η ποιότητα του βίντεο πριν και μετά τη χρήση του FEC σε κάθε αλγόριθμο. Όπως φαίνεται το συνολικό overhead που δημιουργείται σε όλα τα σενάρια είναι σχεδόν το ίδιο. Κατά μέσο όρο το overhead είναι 12,60%.

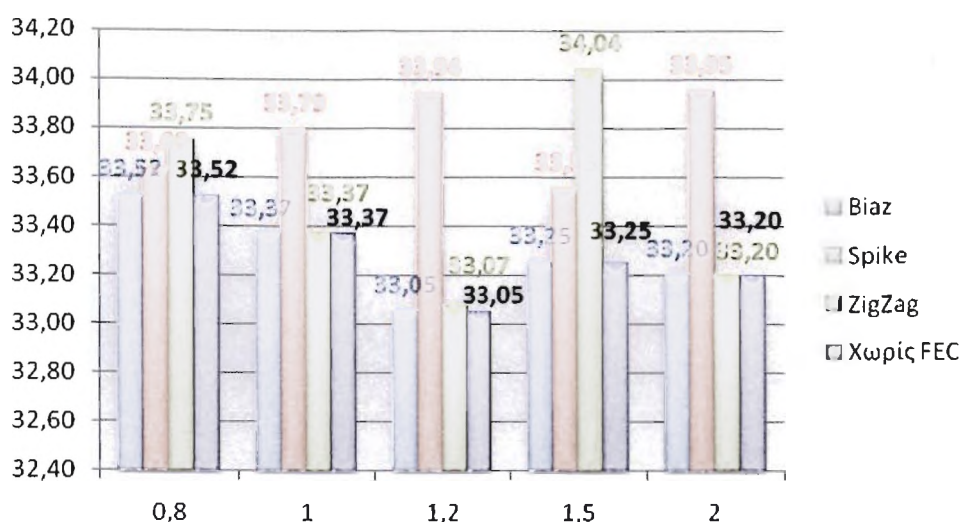


Γράφημα 11: Εκτίμηση ποσοστού απωλειών σε σχέση με τις συνολικές απώλειες - Πιθανότητα pBB 0,85

Overhead



Γράφημα 12: Overhead - Πιθανότητα pBB 0,85

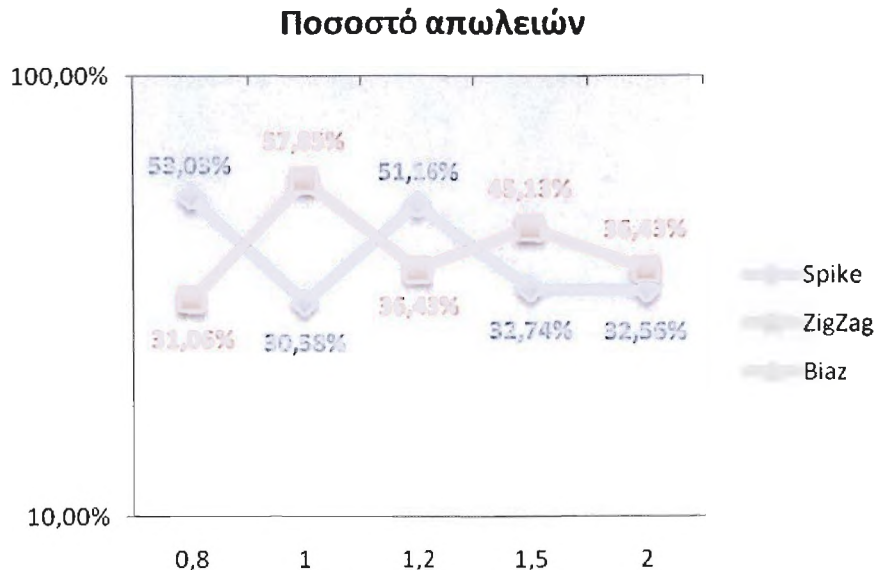


Γράφημα 13: PSNR Αλγορίθμων - Πιθανότητα pBB 0,85

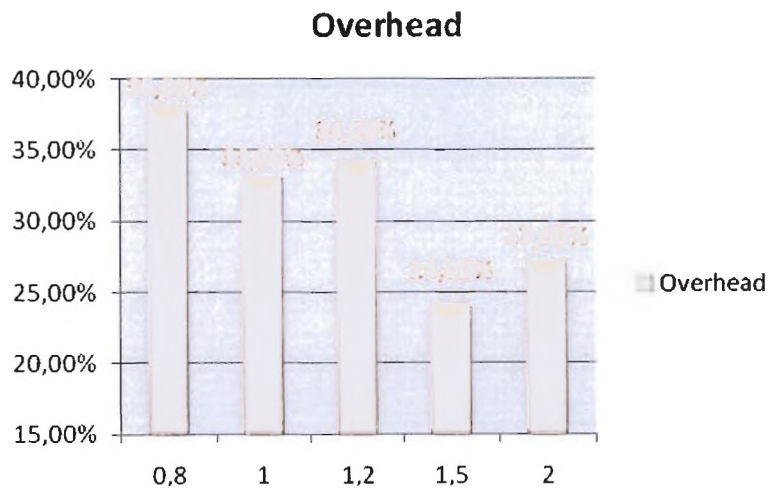
Τα γραφήματα 14, 15 και 16 αφορούν τα σενάρια που έχουν πιθανότητα pBB 0,90. Κρατώντας σταθερή την πιθανότητα μεταβάλλουμε την επιπλέον κίνηση που στέλνεται στο δίκτυο.

Στο γράφημα 14 φαίνεται πως ο Biaz δεν θεωρεί ότι τα πακέτα που χάθηκαν οφείλονται σε λάθη στο φυσικό κανάλι. Ο Spike όπως και ο ZigZag όσο αυξάνεται η κίνηση στο δίκτυο βρίσκουν άλλοτε περισσότερα και άλλοτε λιγότερα πακέτα. Κατά μέσο όρο όμως εντοπίζουν σχεδόν τον ίδιο αριθμό πακέτων ο οποίος είναι 40,01% και 41,38% αντίστοιχα. Όσο αυξάνεται η κίνηση το overhead που δημιουργείται από την αποστολή των πακέτων FEC μειώνεται (γράφημα 15). Κατά μέσο όρο σε όλα τα σενάρια το overhead είναι 31,2%. Στο

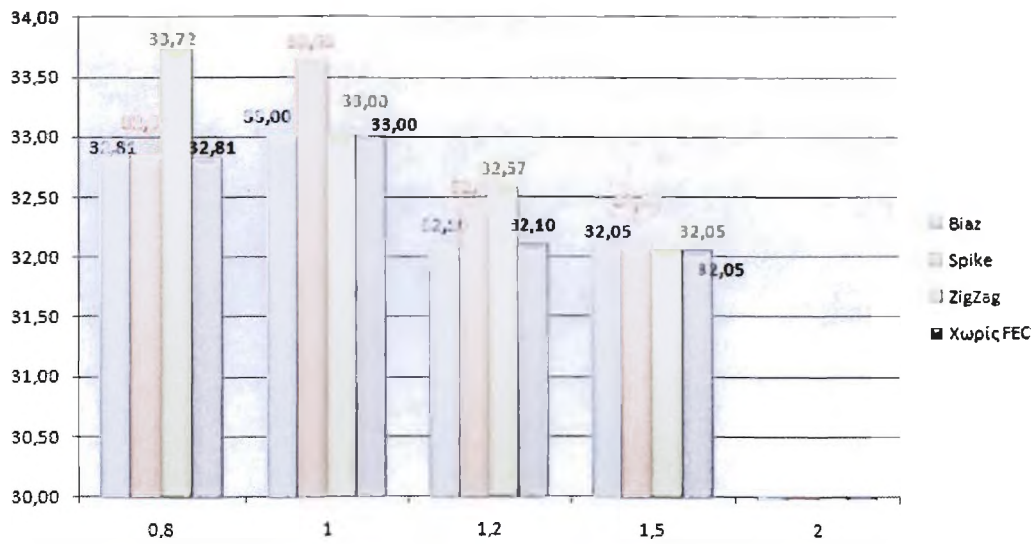
γράφημα 16 βλέπουμε πως όσο αυξάνεται η κίνηση δεν υπάρχουν μεγάλες διαφορές στο PSNR σε όλους τους αλγορίθμους.



Γράφημα 14: Ποσοστό απωλειών σε σχέση με τις συνολικές απώλειες - Πιθανότητα pBB 0,90



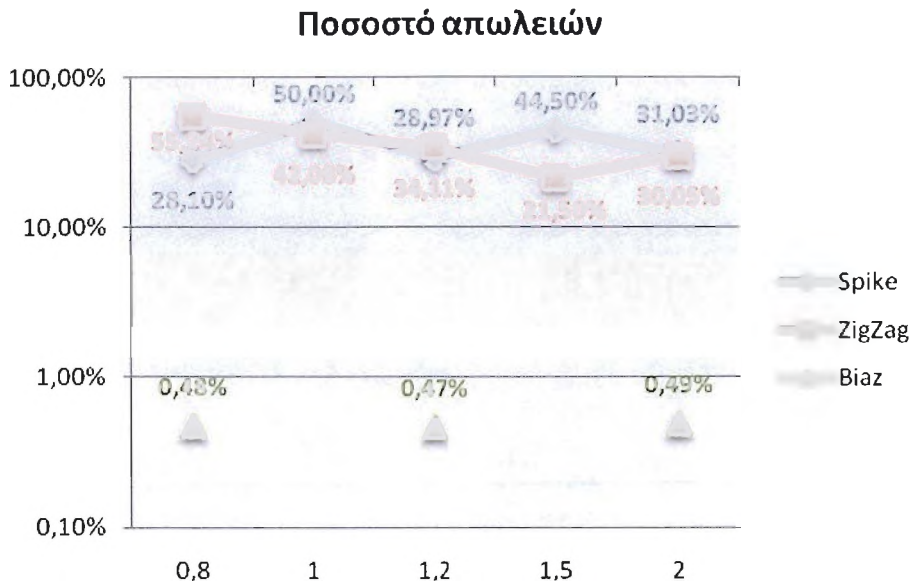
Γράφημα 15: Overhead - Πιθανότητα pBB 0,90



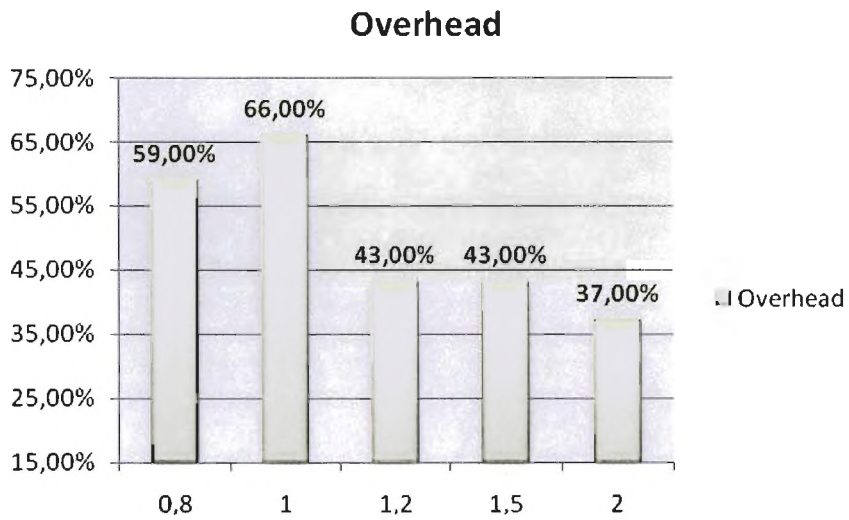
Γράφημα 16: PSNR Αλγορίθμων - Πιθανότητα pBB 0,90

Τα γραφήματα 17, 18 και 19 αφορούν τα σενάρια που έχουν πιθανότητα pBB 0,94. Κρατώντας σταθερή την πιθανότητα μεταβάλλουμε την επιπλέον κίνηση που στέλνεται στο δίκτυο.

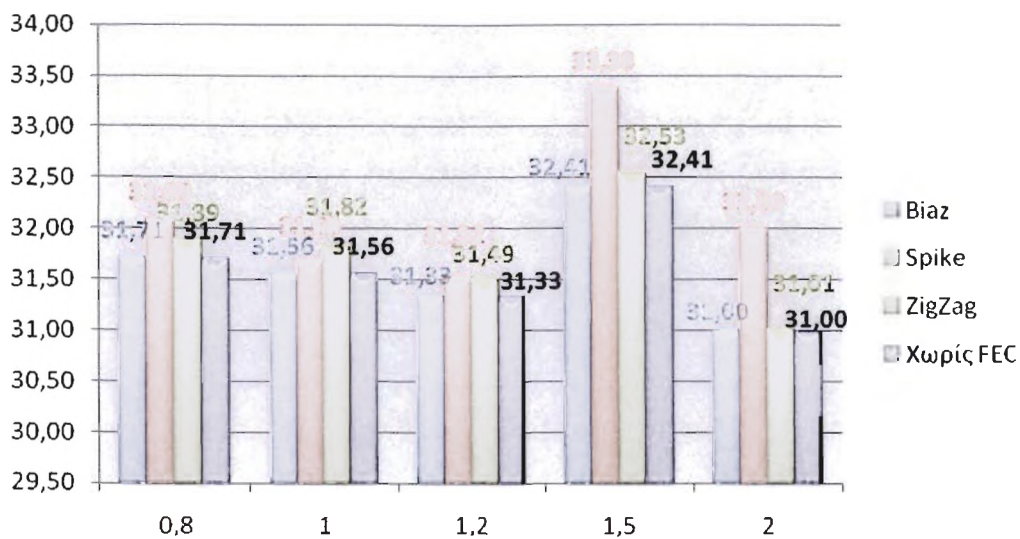
Στο γράφημα 17 βλέπουμε πως ο Spike και ο ZigZag όταν αυξάνεται η κίνηση βρίσκουν σχεδόν τον ίδιο αριθμό πακέτων ενώ ο Biaz βρίσκει για 0,8, 1,2 και 2 Mbps κίνησης. Στο γράφημα 18 φαίνεται πως στην αρχή το overhead είναι υψηλό ενώ μετά είναι πολύ χαμηλότερο. Η μέση τιμή των πακέτων που εντοπίζουν ο Spike, ο ZigZag και ο Biaz είναι 36,52%, 36,6% και 0,29% αντίστοιχα ενώ κατά μέσο όρο στέλνεται 49,6% overhead. Σύμφωνα με το γράφημα 19 παρατηρούμε πώς μόνο όταν η κίνηση είναι 1.5Mbps η ποιότητα του βίντεο είναι η καλύτερη.



Γράφημα 17: Εκτίμηση ποσοστού απωλειών σε σχέση με τις συνολικές απώλειες -
Πιθανότητα pBB 0,94



Γράφημα 18: Overhead - Πιθανότητα pBB 0,94



Γράφημα 19: PSNR Αλγορίθμων - Πιθανότητα pBB 0,94

Στο γράφημα 20 φαίνονται οι διαφορές που παρατηρούνται κατά τη μεταβολή της πιθανότητας pBB. Όταν αυξάνεται η πιθανότητα pBB παρατηρούμε πως στους αλγόριθμους Spike η μέση τιμή των χαμένων πακέτων μειώνεται ενώ στον Biaz αυξάνεται. Σε αντίθεση με αυτούς αλγόριθμους ο ZigZag όταν η πιθανότητα pBB αυξάνεται από 0,85 σε 0,9 τότε αυξάνεται και η μέση τιμή των χαμένων πακέτων ενώ όταν η πιθανότητα αυξάνεται από 0,9 σε 0,94 τότε η μέση τιμή ελαττώνεται. Επίσης το overhead του δικτύου κατά την αύξηση του pBB αυξάνεται.



Γράφημα 20: Μεταβολή της πιθανότητας pBB

Κεφάλαιο 5^ο – Συμπεράσματα και Μελλοντική Δουλειά

Ο στόχος της πτυχιακής ήταν να διατυπώσει και να επιχειρήσει να λύσει πολλά αναπάντητα ερωτήματα σχετικά με τη μετάδοση πολυμέσων σε ασύρματα δίκτυα. Για το σκοπό αυτό, μελετήθηκαν τρεις αλγόριθμοι οι οποίοι κάνουν μια εκτίμηση όσον αφορά τις απώλειες των πακέτων δηλαδή εάν αυτές οφείλονται σε συμφόρηση ή σε λάθη του φυσικού καναλιού. Οι τρεις αλγόριθμοι που μελετήθηκαν είναι οι Biaz, Spike και ZigZag. Οι συγκεκριμένοι αλγόριθμοι υλοποιήθηκαν σε περιβάλλον εξομοίωσης NS-2. Ο Biaz χρησιμοποιεί τον χρόνο μεταξύ της άφιξης των πακέτων (inter-arrival time) για να διακρίνει το είδος των απωλειών, ο Spike χρησιμοποιεί τον χρόνο που κάνει το πακέτο από τον αποστολέα στον παραλήπτη και ο ZigZag χαρακτηρίζει τις απώλειες ως ασύρματες βασιζόμενος στις απώλειες των n πακέτων και στη διαφορά μεταξύ του $roff_i$ και του $roff_{mean}$. Με βάση τα πειραματικά αποτελέσματα, ο αλγόριθμος Spike κάνει καλύτερη εκτίμηση του ποσοστού των λαθών στο φυσικό επίπεδο, με αποτέλεσμα την αποδοτικότερη απόδοση της χρήσης του μηχανισμού FEC στο βίντεο.

Μελλοντικές επεκτάσεις της παρούσης εργασίας, αφορούν το συνδυασμό του αλγόριθμου FEC με μηχανισμό χρονοπρογραμματισμού. Ένας τέτοιος μηχανισμός θα πρέπει να λαμβάνει υπόψη τον αντίκτυπο που θα έχει η απώλεια ενός πλαισίου στη συνολική αλλοίωση του βίντεο. Αυτό συμβαίνει γιατί σε μια ακολουθία βίντεο κωδικοποιημένη με βάση το πρότυπο H.264, τα πλαίσια τύπου I είναι πιο σημαντικά σε σχέση με τα πλαίσια τύπου B και P. Σε περίπτωση συμφόρησης, οι διαθέσιμοι πόροι του δικτύου είναι περιορισμένοι. Για αυτό το λόγο ο προγραμματιστής μετάδοσης πακέτων επιλέγει ποιο συνδυασμό πακέτων βίντεο θα απορρίψει πριν την μετάδοσή τους (π.χ. συνιστάται πρώτα να απορριφθούν πρώτα τα πακέτα που περιλαμβάνουν πλαίσια τύπου B και στη συνέχεια αυτά που περιλαμβάνουν πλαίσια τύπου P). Ιδιαίτερα στη περίπτωση ασύρματων δικτύων που προσφέρουν ποιότητα υπηρεσίας (π.χ. IEEE 802.11e), ο ανωτέρω χρονοπρογραμματισμός μπορεί να συνδυαστεί με τις πολλαπλές κατηγορίες πρόσβαση που υποστηρίζονται.

Ως μελλοντική εργασία, θα μπορούσε να τροποποιηθούν οι τοπολογίες που περιγράφηκαν παραπάνω έτσι ώστε οι ασύρματοι κόμβοι να κινούνται στο χώρο. Αυτή η κινητικότητα θα περιλάμβανε χρήση φυσικών εμποδίων τα οποία θα εισάγουν λάθη στα φυσικά κανάλια. Λόγω της κινητικότητας των χρηστών, οι μηχανισμοί μεταγωγής μεταξύ των ετερογενών ασύρματων δικτύων αυξάνουν την από άκρο-σε-άκρο καθυστέρηση της επικοινωνίας εφαρμογών πολυμέσων, λόγω των διαδικασιών ανακάλυψης του καλύτερου δικτύου, σύνδεσης με το δίκτυο και ενημέρωσης. Σε περιπτώσεις που ο χρήστης κινείται γρήγορα μεταξύ διαφορετικών δικτύων πρόσβασης, η διάφανη και αρραγής επικοινωνία είναι δύσκολο

να εξασφαλιστεί. Το πλαίσιο αναφοράς IEEE 802.21 περιλαμβάνει λειτουργίες που εξασφαλίζουν στο επίπεδο διαχείρισης της κινητικότητας των χρηστών, σημαντικές πληροφορίες για τα γειτονικά δίκτυα, τους διαύλους επικοινωνίας και εντολές, απαραίτητες, κατά τη διάρκεια της μεταγωγής. Η βασικότερη λειτουργία του 802.21 είναι η ανεξάρτητη του μέσου μεταγωγή (Media Independent Handover), η οποία εξασφαλίζει τις κατάλληλες διεπαφές για την ενημέρωση των ανωτέρων επιπέδων σχετικά με πληροφορίες που αφορούν τα κατώτερα επίπεδα. Στα πλαίσια αυτά, έχει ενδιαφέρον η μελέτη της μεταγωγής βίντεο πραγματικού χρόνου σε ετερογενή δίκτυα με βάση το πρότυπο 802.21.

Επιπλέον, ενδιαφέρον παρουσιάζει η πρόταση για κωδικοποίηση του βίντεο σε κλιμακωτά επίπεδα με βάση το πρότυπο H.264/Scalable Video Coding (SVC). Η κωδικοποίηση του βίντεο αποτελεί μια ελκυστική λύση σε προβλήματα που προκύπτουν από τα χαρακτηριστικά των σύγχρονων συστημάτων μετάδοσης βίντεο. Ο όρος 'κλιμακωτή κωδικοποίηση' αναφέρεται στην απόρριψη τμημάτων της ακολουθίας ψηφίων του βίντεο, ώστε να προσαρμοστεί στις ανάγκες και στις απαιτήσεις των τελικών χρηστών, στις δυνατότητες των τερματικών συσκευών και στις συνθήκες του δικτύου. Ο στόχος της επέκτασης του H.264/AVC σε κλιμακωτή συμπίεση είναι η δυνατότητα κωδικοποίησης βίντεο υψηλής ποιότητας σε παραπάνω από μια ακολουθίες ψηφίων, κάθε μια από τις οποίες μπορεί να αποκωδικοποιηθεί με ποιότητα και πολυπλοκότητα ανάλογη του H.264/AVC. Οι προτεραιότητες του H.264/SVC θα μπορούν να συνδυαστούν με τις προτεραιότητες του IEEE 802.11e για κάποιον έξυπνο χρονοπρογραμματισμό.

Βιβλιογραφία

1. Tektronix, “A Guide to MPEG Fundamentals and Protocol Analysis (Including DVB and ATSC)”.
2. Martin Jacklin, MPEG-4 Industry Forum, “MPEG-4 – The Media Standard, The landscape of advanced multimedia coding”
3. Q. Zhang, W. Zhu, Y. Zhang, “Video End-to-end QoS for video delivery over wireless Internet”, Proceedings of IEEE, January 2005.
4. Feng Wu, Guobin Shen, Kun Tan, Fan Yang, Shipeng Li, ‘Next Generation Mobile Multimedia Communications: Media Codec and Media Transport Perspectives’, China Communications Magazine, October 2007.
5. ATI, “Introduction to H.264”.
6. AXIS Communications, “Whitepaper - H.264 video compression standard”.
7. Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra, “Overview of the H.264/AVC Video Coding Standard”. IEEE Trans. On Circuits and Systems for Video Technology, July 2003.
8. Ben-Jye Chang and Shu-Yu Lin, “Mobile IPv6-based efficient vertical handoff approach for heterogeneous wireless networks”. Wireless Communications & Mobile Computing, August 2006.
9. Ekram Hossain, “Heterogeneous Wireless Access Networks Architectures and Protocols”. Springer.
10. Song Cen, Pamela C. Cosman, and Geoffrey M. Voelker, “End-to-end differentiation of congestion and wireless losses”. IEEE/ACM Transactions on Networking, October 2003.
11. Stefan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertl, Lothar Stibor, “IEEE 802.11e Wireless LAN for Quality of Service”, European Wireless Conference, 2002.
12. Eitan Altman and Tania Jimenez, “NS Simulator for beginners”, December 4, 2003.
13. Kevin Fall and Kannan Varadhan, “The ns Manual”, May 18, 2008.
14. Frank Fitzek, Patrick Seeling and Martin Reisslein, “Video Streaming in Wireless Internet”. Electrical Engineering and Applied Signal Processing Series, December 2004.
15. Παρουσίαση Κων/νος Τρούλος, “NS-2 (Network Simulator)”.

16. Jean-Pierre Ebert, Andreas Willig, "A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation", TKN Technical Report Series, March 1999.

Συντομογραφίες

FEC	Forward Error Correction
P2P	Peer to Peer
BER	Bit Error Rate
HD DVD	High-Definition/Density Digital Versatile Disc
MPEG	Moving Pictures Expert Group
VLSI	Very Large Scale Integration
GOP	Group of Pictures
ISDN	Integrated Services Digital Network
VECG	Video Coding Experts Group
JPEG	Joint Photographic Group
PDA	Personal Digital Assistant
SVC	Scalable Video Coding
QoS	Quality of Service
TCP/IP	Transmission Control Program /Internet Protocol
CIF	Common Intermediate Format
QCIF	Quarter Common Intermediate Format
DIA	Digital Image Adaptation
WLAN	Wireless Local Area Network
WiMAX	Worldwide Interoperability for Microwave Access
ROTT	Relative One-way Trip Time
ARQ	Automatic Retransmission request
NS-2	Network Simulator 2
OTCL	Object Tool Command Language
CBR	Constant Bit Rate
UDP	User Datagram Protocol
PSNR	Peak Signal to Noise Ration

Παράρτημα Α

Διευθύνσεις Internet

1. MPEG - http://en.wikipedia.org/wiki/Video_Coding_Experts_Group
2. MPEG - <http://www.chiariglione.org/mpeg/>
3. Gop Structure - <http://digvid.info/mpeg/coding3.asp.php>
4. H.264 standard- <http://www.divr.com/h264/h264/0212/h264>
5. Video Coding Experts Group - http://en.wikipedia.org/wiki/Video_Coding_Experts_Group
6. Scalable Video Coding - http://en.wikipedia.org/wiki/Scalable_Video_Coding
7. Scalable Video Coding - <http://www.elepa.com/ce.uk/mpeg/svc.html>

Παράρτημα Β

A. Κώδικας του TCL αρχείου για την τοπολογία με χρήση CBR Traffic

```
#####  
# Create simulator instance  
#####  
set ns [new Simulator]  
  
#####  
# Parameters passed to the tcl file  
#####  
proc getopt {argc argv} {  
    global opt  
    lappend optlist nn  
    for {set i 0} {$i < $argc} {incr i} {  
        set opt($i) [lindex $argv $i]  
    }  
}  
  
getopt $argc $argv  
  
if {$argc < 3} {  
    puts "\nGive pBB, rtraffic rate value and execution number"  
    puts "eg: ns ptyxiaki_cbr 0.94 1 1\n\n"  
}  
  
#####  
# Define options  
#####  
set opt(prop)    Propagation/TwoRayGround    ;# radio-propagation model  
set opt(chan)    Channel/WirelessChannel    ;# channel type  
set opt(netif)   Phy/WirelessPhy           ;# network interface type  
set opt(mac)     Mac/802_11                 ;# MAC type
```

```

set opt(ifq)      Queue/DropTail/myPriQueue      ;# interface queue type
set opt(ifql)    Queue/DropTail/PriQueue
set opt(ll)      LL                              ;# link layer type
set opt(ant)     Antenna/OmniAntenna            ;# antenna model
set opt(ifqlen)  50                              ;# max packet in ifq
set opt(adhocRouting) NOAH                      ;# routing protocol  DSDV

set opt(x)       500                             ;# x coordinate of topology
set opt(y)       500                             ;# y coordinate of topology

set opt(seed)    0.0                             ;# seed for random number gen.

set num_mobile_nodes  5                          ;# Number of Mobile Nodes
set num_wired_nodes   1                          ;# Number of Wired Nodes
set num_bs_nodes      1                          ;# Number of BS (Base Stations)
set num_nodes [expr $num_wired_nodes + $num_mobile_nodes + $num_bs_nodes]
                ;# Count all nodes

set pGG           0.96                           ;# Good ---> Good
set pBB           $opt(0)                        ;# Bad ---> Bad
set pG            0.001                          ;# Good State
set pB            0.7                            ;# Bad State

set max_fragmented_size 1024
set packetSize [expr $max_fragmented_size + 28] ;# Packet Size

#loss_model: 0 for uniform distribution, 1 for GE model
set loss_model 1

#comm_type: 0 for broadcast, 1 for unicast
set comm_type 0

# CBR traffic rate
set trafficRate $opt(1)
set exectimes $opt(2)

puts "\n\nTraffic Rate: $trafficRate, PBB: $pBB, exec: $exectimes"

```

```

Mac/802_11 set SlotTime_      0.000020      ;# 20us
Mac/802_11 set SIFS_         0.000010      ;# 10us
Mac/802_11 set PreambleLength_ 144          ;# 144 bit
Mac/802_11 set PLCPHeaderLength_ 48          ;# 48 bits
Mac/802_11 set PLCPDataRate_ 1.0e6          ;# 1Mbps
Mac/802_11 set dataRate_     11.0e6        ;# 11Mbps
Mac/802_11 set basicRate_    1.0e6          ;# 1Mbps
Mac/802_11 set ShortRetryLimit_ 7            ;# retransmissions
Mac/802_11 set LongRetryLimit_ 4            ;# retransmissions

```

```
# =====
```

```
# Set up for hierarchical routing
# Needed for routing over a basestation)

```

```
# =====
```

```

$ns node-config -addressType hierarchical
AddrParams set domain_num_ 2      ;# Number of domains
lappend cluster_num 1 1          ;# Number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel $num_wired_nodes [expr $num_mobile_nodes + $num_bs_nodes]
;# Number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# Number of nodes of each domain

```

```

#Open the traffic trace file to record all events
set ntr [open out_${Sopt(0)}_${Sopt(1)}_${Sopt(2)}.tr w]
$ns trace-all $ntr

```

```

# Create channel
set chan [new $opt(chan)]

```

```

# Create topography object
set topo [new Topography]

```

```
# Define topology
$topo load_flatgrid $opt(x) $opt(y)

# Create God
create-god [expr $num_mobile_nodes + $num_bs_nodes]
```

```
# =====
```

```
# Creating Wired nodes
```

```
# =====
```

```
for {set i 0} {$i < $num_wired_nodes} {incr i} {
    set w_node_($i) [$ns node 0.0.[expr $i]]
    #puts "Wired node $i created"
}
```

```
# =====
```

```
# Create base station node
```

```
# =====
```

```
$ns node-config -adhocRouting $opt(adhocRouting) \
    -llType $opt(ll) \
    -macType $opt(mac) \
    -ifqType $opt(ifq) \
    -ifqLen $opt(ifqlen) \
    -antType $opt(ant) \
    -propType $opt(prop) \
    -phyType $opt(netif) \
    -channel $chan \
    -topoInstance $topo \
    -wiredRouting ON \
    -agentTrace OFF \
    -routerTrace OFF \
    -macTrace OFF \
    -movementTrace OFF
```

```
# Hypothesis Base Station
```



```

set BS(0) [$ns node 1.0.0]
$BS(0) random-motion 0
set BSnetif_ [$BS(0) set netif_(0)]
$BSnetif_ set-error-level $pGG $pBB $pG $pB $loss_model

set bs_ifq [$BS(0) set ifq_(0)]
set bs_mac [$BS(0) set mac_(0)]
set bs_phy [$BS(0) set netif_(0)]

# Establishes the BS position
$BS(0) set X_ 250.0
$BS(0) set Y_ 250.0
$BS(0) set Z_ 0.0

# $bs_ifq queue type
set queuechan [open qlen_$opt(0)_$opt(1)_$opt(2).tr w]
$bs_ifq trace curq_
$bs_ifq attach $queuechan
$bs_ifq getvars $pBB $trafficRate $sexectimes

set rng [new RNG]
$rng seed $opt(seed)
set rand1 [new RandomVariable/Uniform]
$rand1 use-rng $rng
$rand1 set min_ -50.0
$rand1 set max_ 50.0

# =====
# Creating mobile nodes
# =====
$ns node-config -wiredRouting OFF
$ns node-config -ifqType $opt(ifq1)

```

```

for {set i 0} {$i < $num_mobile_nodes} {incr i} {
  set wl_node_($i) [$ns node 1.0.[expr $i + 1]]
  $wl_node_($i) random-motion 0           ;# disable random motion
  #puts "wireless node $i created"
  $wl_node_($i) base-station [AddrParams addr2id [$BS(0) node-addr]]
  set x [expr 250+[$rand1 value]]
  set y [expr 250+[$rand1 value]]
  $wl_node_($i) set X_ $x
  $wl_node_($i) set Y_ $y
  $wl_node_($i) set Z_ 0.0
  #puts "X_: $x Y_: $y"
}

# =====
# Loss model->(0:random uniform; 1:GE)
# =====
puts "\n"
puts "pGG: $pGG pBB: $pBB"
puts "pG: $pG pB: $pB \n"

set wl_phy [$wl_node_(0) set netif_(0)]
$wl_phy set-error-level $pGG $pBB $pG $pB $loss_model

# =====
# Establishes the link between Wired node $w_node_(0) and BS
# =====
$ns duplex-link $w_node_(0) $BS(0) 100Mb 1ms DropTail
set q1 [[ $ns link $w_node_(0) $BS(0)] queue]

# =====
# Video Server
# =====

```

```

# Set the sender trace file name to sd
set udp1 [new Agent/myUDP]
$ns attach-agent $w_node_(0) $udp1
$udp1 set packetSize_ $packetSize
$udp1 set_filename sd_tfile_$opt(0)_$opt(1)_$opt(2)
$udp1 getvariables $opt(0) $opt(1) $opt(2)

# Set the receiver trace file to rd
set null1 [new Agent/myUdpSink2]
$ns attach-agent $w1_node_(0) $null1
$ns connect $udp1 $null1
$null1 set_trace_filename rd_tfile_$opt(0)_$opt(1)_$opt(2)
$null1 get_variables $opt(0) $opt(1) $opt(2)

# Import video file
set original_file_name highway_qcif.txt
set trace_file_name video1_$opt(0)_$opt(1)_$opt(2).dat
set original_file_id [open $original_file_name r]
set trace_file_id [open $trace_file_name w]

set frame_count 0
set last_time 0
set pre_time 0

while {[eof $original_file_id] == 0} {

    gets $original_file_id current_line
    scan $current_line "%d%f%d%d" seq nexttime length type
    set time [expr ($nexttime-$pre_time)*1000000.0]

    puts $trace_file_id "$time $length $type"
    set pre_time $nexttime
    incr frame_count
}

```

```
#puts "frame count= [expr $frame_count - 1]"
```

```
close $original_file_id
```

```
close $trace_file_id
```

```
set end_sim_time $nexttime
```

```
puts "end_sim_time= $end_sim_time sec"
```

```
set trace_file [new Tracefile]
```

```
$trace_file filename $trace_file_name
```

```
set video1 [new Application/Traffic/myTrace3]
```

```
$video1 attach-agent $udp1
```

```
$video1 attach-tracefile $trace_file
```

```
# =====
```

```
# CBR Traffic
```

```
# =====
```

```
# wl_node_(1) ---> wl_node_(3)
```

```
set udp2 [new Agent/UDP]
```

```
$ns attach-agent $wl_node_(1) $udp2
```

```
set cbr1 [new Application/Traffic/CBR]
```

```
$cbr1 set rate_ $trafficRate Mbps
```

```
$cbr1 set interval_ 0.02
```

```
$cbr1 attach-agent $udp2
```

```
set null2 [new Agent/Null]
```

```
$ns attach-agent $wl_node_(2) $null2
```

```
$ns connect $udp2 $null2
```

```
# wl_node_(3) ---> wl_node_(4)
```

```
set udp3 [new Agent/UDP]
```

```
$ns attach-agent $wl_node_(3) $udp3
```

```
set cbr2 [new Application/Traffic/CBR]
```

```

$ubr2 set      rate_   $trafficRate Mbps
$ubr2 set      interval_ 0.02
$ubr2 attach-agent $udp3

set null3 [new Agent/Null]
$ns attach-agent $wl_node_(4) $null3
$ns connect $udp3 $null3

# =====
# Generate start times
# =====

set rng [new RNG]
$rng seed $opt(seed)

set RVstart [new RandomVariable/Uniform]
$RVstart set min_ 0
$RVstart set max_ 1
$RVstart use-rng $rng
set startT_video [expr [$RVstart value]]
puts "startT_video: $startT_video sec"

set RVstart2 [new RandomVariable/Uniform]
$RVstart2 set min_ 1
$RVstart2 set max_ 2
$RVstart2 use-rng $rng
set startT_cbr1 [expr [$RVstart2 value]]
set startT_cbr2 [expr [$RVstart2 value]]
puts "startT_cbr1: $startT_cbr1 sec"
puts "startT_cbr2: $startT_cbr2 sec \n"

# Tell nodes when the simulation ends
for {set i 0} {$i < $num_mobile_nodes} {incr i} {
    $ns at [expr $end_sim_time + 5.0] "$wl_node_($i) reset";
}

```

```

# Define a 'stop' procedure
proc stop {} {
    global ns ntr bs_ifq

    $ns flush-trace
    close $ntr

    set fecno [$bs_ifq fecno?]
    puts "Total Fec no: $fecno"
}

# =====
# Schedule events
# =====
$ns at $startT_cbr1 "$cbr1 start"
$ns at $startT_cbr2 "$cbr2 start"
$ns at $startT_video "$video1 start"
$ns at [expr $end_sim_time + 4.0] "$cbr1 stop"
$ns at [expr $end_sim_time + 4.0] "$cbr2 stop"
$ns at [expr $end_sim_time + 5.0] "$null1 closefile"
$ns at [expr $end_sim_time + 5.1] "$null1 printstatus"
$ns at [expr $end_sim_time + 5.2] "stop"
$ns at [expr $end_sim_time + 5.5] "puts \"NS EXITING...\" ; $ns halt"

# Run the simulation
$ns run

```

B. Κώδικας του TCL αρχείου για την τοπολογία με χρήση Exponential Traffic

```

# =====
# create simulator instance
# =====

```

```

set ns [new Simulator]

# =====
# Parameters passed to the tcl file
# =====

proc getopt {argc argv} {
    global opt
    lappend optlist nn
    for {set i 0} {$i < $argc} {incr i} {
        set opt($i) [lindex $argv $i]
    }
}

getopt $argc $argv

if {$argc < 3} {
    puts "\nGive pBB, rtraffic rate value and execution number"
    puts "eg: ns ptyxiaki_cbr 0.94 1 1\n\n"
}

# =====
# Define options
# =====

set opt(prop)      Propagation/TwoRayGround      ;# radio-propagation model
set opt(chan)      Channel/WirelessChannel      ;# channel type
set opt(netif)     Phy/WirelessPhy              ;# network interface type
set opt(mac)       Mac/802_11                   ;# MAC type
set opt(ifq)       Queue/DropTail/myPriQueue    ;# interface queue type
set opt(ifq1)      Queue/DropTail/PriQueue      ;# interface queue type
set opt(ll)        LL                            ;# link layer type
set opt(ant)       Antenna/OmniAntenna          ;# antenna model
set opt(ifqlen)    50                            ;# max packet in ifq
set opt(adhocRouting) NOAH                       ;# routing protocol   DSDV

set opt(x)         500                            ;# x coordinate of topology

```

```

set opt(y)          500          ;# y coordinate of topology

set opt(seed)       0.0          ;# seed for random number gen.

set num_mobile_nodes 5          ;# Number of Mobile Nodes
set num_wired_nodes 1          ;# Number of Wired Nodes
set num_bs_nodes    1          ;# Number of BS (Base Stations)
set num_nodes [expr $num_wired_nodes + $num_mobile_nodes + $num_bs_nodes]
                ;# Count all nodes

set pGG            0.96         ;# Good ---> Good
set pBB            $opt(0)      ;# Bad ---> Bad
set pG             0.001        ;# Good State
set pB             0.7          ;# Bad State

set max_fragmented_size 1024
set packetSize [expr $max_fragmented_size + 28] ;# Packet Size

#loss_model: 0 for uniform distribution, 1 for GE model
set loss_model 1

#comm_type: 0 for broadcast, 1 for unicast
set comm_type 0

# Exponential traffic rate
set trafficRate $opt(1)
set exectimes $opt(2)
puts "\n\nTraffic Rate: $trafficRate, PBB: $pBB, exec: $exectimes"

Mac/802_11 set SlotTime_      0.000020    ;# 20us
Mac/802_11 set SIFS_          0.000010    ;# 10us
Mac/802_11 set PreambleLength_ 144        ;# 144 bit
Mac/802_11 set PLCPHeaderLength_ 48        ;# 48 bits
Mac/802_11 set PLCPDataRate_  1.0e6       ;# 1Mbps
Mac/802_11 set dataRate_      11.0e6      ;# 11Mbps
Mac/802_11 set basicRate_     1.0e6       ;# 1Mbps

```



```

Mac/802_11 set ShortRetryLimit_ 7      ;# retransmissions
Mac/802_11 set LongRetryLimit_  4      ;# retransmissions

# =====
# set up for hierarchical routing
# needed for routing over a base station)
# =====

$ns node-config -addressType hierarchical
AddrParams set domain_num_ 2          ;# Number of domains
lappend cluster_num 1 1                ;# Number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel $num_wired_nodes [expr $num_mobile_nodes + $num_bs_nodes] ;#
Number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# Number of nodes of each domain

#Open the traffic trace file to record all events
set ntr [open out_ $opt(0)_ $opt(1)_ $opt(2).tr w]
$ns trace-all $ntr

# Create channel
set chan [new $opt(chan)]

# Create topography object
set topo [new Topography]

# Define topology
$topo load_flatgrid $opt(x) $opt(y)

# Create God
create-god [expr $num_mobile_nodes + $num_bs_nodes]

# =====
# Creating Wired nodes

```

```

# =====
for {set i 0} {$i < $num_wired_nodes} {incr i} {
    set w_node_($i) [$ns node 0.0.[expr $i]]
    #puts "Wired node $i created"
}

# =====
# Create base station node
# =====
$ns node-config -adhocRouting $opt(adhocRouting) \
    -llType $opt(ll) \
    -macType $opt(mac) \
    -ifqType $opt(ifq) \
    -ifqLen $opt(ifqlen) \
    -antType $opt(ant) \
    -propType $opt(prop) \
    -phyType $opt(netif) \
    -channel $chan \
    -topoInstance $topo \
    -wiredRouting ON \
    -agentTrace OFF \
    -routerTrace OFF \
    -macTrace OFF \
    -movementTrace OFF

# Hypothesis Base Station
set BS(0) [$ns node 1.0.0]
$BS(0) random-motion 0
set BSnetif_ [$BS(0) set netif_(0)]
$BSnetif_ set-error-level $pGG $pBB $pG $pB $loss_model

set bs_ifq [$BS(0) set ifq_(0)]
set bs_mac [$BS(0) set mac_(0)]
set bs_phy [$BS(0) set netif_(0)]

```

```

# Establishes the BS position
$BS(0) set X_ 250.0
$BS(0) set Y_ 250.0
$BS(0) set Z_ 0.0

# $bs_ifq queue type
set queuechan [open qlen_ $opt(0)_ $opt(1)_ $opt(2).tr w]
$bs_ifq trace curq_
$bs_ifq attach $queuechan
$bs_ifq getvars $pBB $trafficRate $sexectimes

set rng [new RNG]
$rng seed $opt(seed)
set rand1 [new RandomVariable/Uniform]
$rand1 use-rng $rng
$rand1 set min_ -50.0
$rand1 set max_ 50.0

# =====
# Creating mobile nodes
# =====
$ns node-config -wiredRouting OFF
$ns node-config -ifqType $opt(ifq1)

for {set i 0} {$i < $num_mobile_nodes} {incr i} {
    set wl_node_($i) [$ns node 1.0.[expr $i + 1]]
    $wl_node_($i) random-motion 0           ;# disable random motion
    #puts "wireless node $i created"
    $wl_node_($i) base-station [AddrParams addr2id [$BS(0) node-addr]]
    set x [expr 250+[$rand1 value]]
    set y [expr 250+[$rand1 value]]
    $wl_node_($i) set X_ $x
}

```

```

$wl_node_($i) set Y_ $y
$wl_node_($i) set Z_ 0.0
#puts "X_:$x Y_:$y"
}

# =====
# Loss model->(0:random uniform; 1:GE)
# =====

puts "\n"
puts "pGG: $pGG pBB: $pBB"
puts "pG: $pG pB: $pB \n"

set wl_phy [$wl_node_(0) set netif_(0)]
$wl_phy set-error-level $pGG $pBB $pG $pB $loss_model

# =====
# Establishes the link between Wired node $w_node_(0) and BS
# =====

$ns duplex-link $w_node_(0) $BS(0) 100Mb 1ms DropTail
set q1 [[ $ns link $w_node_(0) $BS(0) ] queue]

# =====
# Video Server
# =====

# Set the sender trace file name to sd
set udp1 [new Agent/myUDP]
$ns attach-agent $w_node_(0) $udp1
$udp1 set packetSize_ $packetSize
$udp1 set_filename sd_tfile_$opt(0)_$opt(1)_$opt(2)
$udp1 getvariables $opt(0) $opt(1) $opt(2)

# Set the receiver trace file to rd
set null1 [new Agent/myUdpSink2]

```

```

$ns attach-agent $wl_node_(0) $null1
$ns connect $udpl $null1
$null1 set_trace_filename rd_tfile_$opt(0)_$opt(1)_$opt(2)
$null1 get_variables $opt(0) $opt(1) $opt(2)

# Import video file
set original_file_name highway_qcif.txt
set trace_file_name video1_$opt(0)_$opt(1)_$opt(2).dat
set original_file_id [open $original_file_name r]
set trace_file_id [open $trace_file_name w]

set frame_count 0
set last_time 0
set pre_time 0

while {[eof $original_file_id] == 0} {

    gets $original_file_id current_line
    scan $current_line "%d%f%d%d" seq nexttime length type
    set time [expr ($nexttime-$pre_time)*1000000.0]

    puts $trace_file_id "$time $length $type"
    set pre_time $nexttime
    incr frame_count
}

#puts "frame count= [expr $frame_count - 1]"

close $original_file_id
close $trace_file_id
set end_sim_time $nexttime
#puts "end_sim_time= $end_sim_time sec"

set trace_file [new Tracefile]

```

```

Strace_file filename $trace_file_name
set video1 [new Application/Traffic/myTrace3]
$video1 attach-agent $udp1
$video1 attach-tracefile $trace_file

# =====
# Exponential Traffic1
# =====

set udp2 [new Agent/UDP]
$ns attach-agent $wl_node_(1) $udp2
set null2 [new Agent/Null]
$ns attach-agent $wl_node_(2) $null2
$ns connect $udp2 $null2
set traffic [new Application/Traffic/Exponential]
$traffic set    packetSize_    $packetSize
$traffic set    burst_time_    0.008
$traffic set    idle_time_    0.5
$traffic set    rate_    $trafficRate Mbps
#1Mbps
$traffic attach-agent $udp2

# =====
# Exponential Traffic2
# =====

set udp3 [new Agent/UDP]
$ns attach-agent $wl_node_(3) $udp3
set null3 [new Agent/Null]
$ns attach-agent $wl_node_(4) $null3
$ns connect $udp3 $null3
set traffic2 [new Application/Traffic/Exponential]
$traffic2 set    packetSize_    $packetSize
$traffic2 set    burst_time_    0.008
$traffic2 set    idle_time_    0.5
$traffic2 set    rate_    $trafficRate Mbps

```

```

#1Mbps
$traffic2 attach-agent $udp3

# =====
# Generate start times
# =====

set rng [new RNG]
$rng seed $opt(seed)

set RVstart [new RandomVariable/Uniform]
$RVstart set min_ 0
$RVstart set max_ 1
$RVstart use-rng $rng
set startT_video [expr [$RVstart value]]
puts "startT_video: $startT_video sec"

set RVstart2 [new RandomVariable/Uniform]
$RVstart2 set min_ 1
$RVstart2 set max_ 2
$RVstart2 use-rng $rng
set startT_traffic [expr [$RVstart2 value]]
set startT_traffic2 [expr [$RVstart2 value]]
puts "startT_traffic: $startT_traffic sec \n"

# Tell nodes when the simulation ends
for {set i 0} {$i < $num_mobile_nodes} {incr i} {
    $ns at [expr $end_sim_time + 5.0] "$wl_node_($i) reset";
}

# Define a 'stop' procedure
proc stop {} {
    global ns ntr bs_ifq

    $ns flush-trace

```

```

close $ntr

set fecno [$bs_ifq fecno?]
puts "Total Fec no: $fecno"
}

# =====
# Schedule events
# =====

$ns at $startT_traffic "$traffic start"
$ns at $startT_traffic "$traffic2 start"
$ns at $startT_video "$video1 start"
$ns at [expr $end_sim_time + 5.0] "$null1 closefile"
$ns at [expr $end_sim_time + 5.1] "$null1 printstatus"
$ns at [expr $end_sim_time + 5.2] "stop"
$ns at [expr $end_sim_time + 5.5] "puts \"NS EXITING...\" ; $ns halt"

# Run the simulation
$ns run

```

Γ. Δημιουργία πακέτων FEC

Ο παρακάτω κώδικας έχει υλοποιηθεί μέσα στο αρχείο myprique.cc. Με αυτό τον κώδικα μπορούμε να στείλουμε FEC πακέτα στον παραλήπτη του βίντεο για όσα πακέτα χάθηκαν εξαιτίας των ασύρματων απωλειών.

```

while(!feof(nfilepckts_spike))
{
    int rc = fgetc(nfilepckts_spike);
    if (rc=='\n')
        no_of_n_spike ++;
}
rewind(nfilepckts_spike);

```



```

pos_end_spike = no_of_n_spike * 2;

while(!feof(nfilepckts_zz))
{
    int rc = fgetc(nfilepckts_zz);
    if (rc=='\n')
        no_of_n_zz ++;
}
rewind(nfilepckts_zz);
pos_end_zz = no_of_n_zz * 2;

while(!feof(nfilepckts_biaz))
{
    int rc = fgetc(nfilepckts_biaz);
    if (rc=='\n')
        no_of_n_biaz ++;
}
rewind(nfilepckts_biaz);
pos_end_biaz = no_of_n_biaz * 2;

if(time_spike == 0)
{
    fscanf(nfilepckts_spike,"%d",&npckts_spike);
    //printf("npckts_spike= %d\n",npckts_spike);
    if(npckts_spike != 0)
    {
        rewind(nfilepckts_spike);
        for(i=0; i<no_of_n_spike; i++)
        {
            fscanf(nfilepckts_spike,"%d",&npckts_spike);
            count_l_spike += npckts_spike;
        }
        fgetpos (nfilepckts_spike,&pos_spike);
        //printf("pos_spike= %d\n",pos_spike);
        time_spike = 1;
    }
}

```

```

        tmp_lines_spike = no_of_n_spike;
    }
}

if(time_zz == 0)
{
    fscanf(nfilepckts_zz,"%d",&npckts_zz);
    if(npckts_zz != 0)
    {
        rewind(nfilepckts_zz);
        for(i=0; i<no_of_n_zz; i++)
        {
            fscanf(nfilepckts_zz,"%d",&npckts_zz);
            //printf("%d\n",npckts_zz);
            count_l_zz += npckts_zz;
        }
        fgetpos (nfilepckts_zz,&pos_zz);
        //printf("pos_zz= %d\n",pos_zz);
        time_zz = 1;
        tmp_lines_zz = no_of_n_zz;
    }
}

if(time_biaz == 0)
{
    fscanf(nfilepckts_biaz,"%d",&npckts_biaz);
    if(npckts_biaz != 0)
    {
        rewind(nfilepckts_biaz);
        for(i=0; i<no_of_n_biaz; i++)
        {
            fscanf(nfilepckts_biaz,"%d",&npckts_biaz);
            printf("%d\n",npckts_biaz);
            count_l_biaz += npckts_biaz;
        }
    }
}

```

```

        fgetpos (nfilepckts_biaz,&pos_biaz);
        //printf("pos_biaz= %d\n",pos_biaz);
        time_biaz = 1;
        tmp_lines_biaz = no_of_n_biaz;
    }
}

if(npckts_spike != 0)
{
    if(no_of_n_spike > tmp_lines_spike)
    {
        for(i=(pos_spike+1); i<pos_end_spike; i+=2)
        {
            fseek ( nfilepckts_spike , i, SEEK_SET );
            fscanf(nfilepckts_spike,"%d",&npckts_spike);
            count_l_spike += npckts_spike;
            //printf("pos_spike= %d\n",i);
        }
        fgetpos (nfilepckts_spike,&pos_spike);
        //printf("pos_spike= %d\n",pos_spike);
        tmp_lines_spike = no_of_n_spike ;
    }
}

if(npckts_zz != 0)
{
    if(no_of_n_zz > tmp_lines_zz)
    {
        for(i=(pos_zz+1); i<pos_end_zz; i+=2)
        {
            fseek ( nfilepckts_zz , i, SEEK_SET );
            fscanf(nfilepckts_zz,"%d",&npckts_zz);
            count_l_zz += npckts_zz;
        }
        fgetpos (nfilepckts_zz,&pos_zz);
    }
}

```

```

        tmp_lines_zz = no_of_n_zz ;
    }
}

if(npckts_biaz != 0)
{
    if(no_of_n_biaz > tmp_lines_biaz)
    {
        for(i=(pos_biaz+1); i<pos_end_biaz; i+=2)
        {
            fseek ( nfilepckts_biaz , i, SEEK_SET );
            fscanf(nfilepckts_biaz,"%d",&npckts_biaz);
            count_l_biaz += npckts_biaz;
        }
        fgetpos (nfilepckts_biaz,&pos_biaz);
        tmp_lines_biaz = no_of_n_biaz ;
    }
}

fclose(nfilepckts_spike);
fclose(nfilepckts_zz);
fclose(nfilepckts_biaz);

count_all = count_l_spike + count_l_zz + count_l_biaz;

for(int i=0; i<count_all; i++)
{
    Packet *pkt = p->copy();
    hdr_rtp *rh1 = hdr_rtp::access(pkt);
    hdr_ip *iph1 = hdr_ip::access(pkt);
    hdr_cmh *ch1 = hdr_cmh::access(pkt);
    hdr_myfec* mfec = hdr_myfec::access(pkt);

    ch1->ptype() = PT_myFEC;
    ch1->size() = fecpktsize;
}

```

```

ch1->frame_pkt_id_=0;

mfec->n = (8+fecpkts);
mfec->k = 8;

for(int j=0; j< 7; j++)
    mfec->redundant[j]=video_pktid[j];

mfec->redundant[7]=ch->frame_pkt_id_;

if(i<count_l_spike)
    mfec->redundant[8]= 0;
else if(i >= count_l_spike && i < (count_l_spike + count_l_zz))
    mfec->redundant[8] = 1;
else //if(i >= (count_l_spike + count_l_zz))
    mfec->redundant[8] = 2;

q1_.enqueue(pkt);
yesornot=1;
}

```

Δ. Παραλήπτης Βίντεο.

Ο παρακάτω κώδικας είναι υλοποιημένος στο αρχείο myudpsink2.cc. Χρησιμοποιώντας τον παρακάτω κώδικα μπορούμε να ανιχνεύσουμε τα χαμένα πακέτα και να κάνουμε διάκριση των απωλειών.

```

if(hdr->ptype()==PT_VIDEO)
{
    count += 1;
    check_al = 0;
    now = Scheduler::instance().clock();
    fprintf(tFile,"%-16f id %-16d udp %-16d\n", now, hdr->frame_pkt_id_, hdr->size()-
28);
}

```

```

if(hdr->frame_pkt_id_ == 0)
    packpos = 0;
else
{
    if(sendtime<10)
    {
        packpos = hdr->frame_pkt_id_ * 9;
        last_id = hdr->frame_pkt_id_;
    }
    else
        packpos = last_id * 9 + (hdr->frame_pkt_id_ - last_id) * 10;
}
fseek( sdtimes, packpos , SEEK_SET );
fscanf(sdtimes,"%lf",&sendtime);

roff = now - sendtime; //receiver - sender

if(roff_max == -1) roff_max = roff;
if(roff_min == -1) roff_min = roff;
if(roff > roff_max) roff_max = roff;
if(roff < roff_min) roff_min = roff;

if(hdr->frame_pkt_id_ != 0)
{
    if(check_zero == 1)
        if(last_pack_rcv != hdr->frame_pkt_id_ - 1)
            npackets = hdr->frame_pkt_id_ - last_pack_rcv - 1;
        else
        {
            npackets = 0;
            roff_mean = roff/32;
            roff_dev = 2*(1/32)*(fabs(roff - roff_mean));
        }
    else

```

```

        {
            npackets = hdr->frame_pkt_id_;
            check_zero = 1;
            rott_mean = (1-1/32)*rott_mean + rott/32;
            rott_dev = (1 - 1/16)*rott_dev + (fabs(rott - rott_mean))/16;
        }
    }
else
    {
        check_zero = 1;
        npackets = 0;
        rott_mean = rott/32;
        rott_dev = (fabs(rott - rott_mean))/16;
    }
last_pack_recv = hdr->frame_pkt_id_;

countn += npackets;

if(last_time == -1)
    {
        int_arr_time = now;
        min_int_arr_time = int_arr_time;
    }
else
    int_arr_time = now - last_time;

if(int_arr_time < min_int_arr_time) min_int_arr_time = int_arr_time;

spikestart = rott_min + (rott_max - rott_min)/2;
spikeend = rott_min + (rott_max - rott_min)/3;

tmp1 = fabs(rott_mean - rott_dev);
tmp2 = fabs(rott_mean - rott_dev/2);

if(npackets > 0)

```

```

{
    //Biaz Scheme
    if((((npackets+1)*min_int_arr_time) <= int_arr_time) &&
(((npackets+2)*min_int_arr_time) > int_arr_time))
    {
        biaz_count++;
        biazn+=npackets;
        sprintf(tmpbuf, "nfpcks_biaz_%.2lf_%.2lf_%d", pBB_r,
trafficRate_r, exectime_r);
        nfilepackets_biaz = fopen(tmpbuf,"a");
        fprintf(nfilepackets_biaz,"%d\n",npackets);
        fclose(nfilepackets_biaz);
    }

    //Spike Scheme
    if(rott >= rott_min && rott <= spikeend)
    {
        spike_count ++;
        spiken+=npackets;
        sprintf(tmpbuf, "nfpcks_spike_%.2lf_%.2lf_%d", pBB_r,
trafficRate_r, exectime_r);
        nfilepackets_spike = fopen(tmpbuf,"a");
        fprintf(nfilepackets_spike,"%d\n",npackets);
        fclose(nfilepackets_spike);
    }

    //ZigZag Scheme
    if((npackets == 1 && rott < tmp1) || (npackets == 2 && rott < tmp2) ||
(npackets == 3 && rott < rott_mean*500) || (npackets > 3 && rott < tmp2))
    {
        zigzag_count ++;
        zigzagn+=npackets;
        sprintf(tmpbuf, "nfpcks_zz_%.2lf_%.2lf_%d", pBB_r, trafficRate_r,
exectime_r);
        nfilepackets_zz = fopen(tmpbuf,"a");

```



```
        fprintf(nfilepackets_zz,"%d\n",npackets);
        fclose(nfilepackets_zz);
        check_al ++;
    }
}
last_time = now;
}
```