

Ατομική Διπλωματική Εργασία

**ΠΡΟΣΑΡΜΟΣΤΙΚΗ ΜΕΤΑΔΟΣΗ ΒΙΝΤΕΟ :
ΕΡΓΑΛΕΙΟ ΠΡΟΣΟΜΟΙΩΣΗΣ**

Φοίβος Κουρούγιαννης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2011

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Προσαρμοστική Μετάδοση Βίντεο:
Εργαλείο Προσομοίωσης**

Φοίβος Κουρούγιαννης

Επιβλέπων Καθηγητής
Δρ. Βάσος Βασιλείου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2011

ΕΥΧΑΡΙΣΤΙΕΣ

Στο σημείο αυτό, με το τέλος των προπτυχιακών μου σπουδών και την ολοκλήρωση της διπλωματικής μου εργασίας, θα ήθελα να ευχαριστήσω όλους εκείνους που συνέβαλαν και βοήθησαν στην πραγματοποίηση της τελευταίας.

Αρχικά θέλω να ευχαριστώ θερμά τον επιβλέποντα καθηγητή της διπλωματικής εργασίας, Δρ. Βάσο Βασιλείου, για την πολύτιμη καθοδήγηση και εμπιστοσύνη που μου έδειξε ειδικά σε στιγμές που η συνέχεια της διπλωματικής δεν ήταν ξεκάθαρη. Χωρίς τις υποδείξεις του δεν θα ήταν δυνατή η ολοκλήρωση της. Εύχομαι σε αυτόν και την οικογένειά του να έχουν πάντα υγεία και ευτυχία.

Έπειτα ευχαριστώ ιδιαίτερα τους γονείς μου για την υποστήριξη και υπομονή τους κατά τη διάρκεια των σπουδών μου. Φρόντισαν να μην μου λείπει τίποτα και δεν είχαν πολλές απαιτήσεις όταν ήμουν πιεσμένος λόγω εξετάσεων ή εργασιών που είχα να παραδώσω.

Τέλος θέλω επίσης να ευχαριστήσω επίσης τον υποψήφιο διδάκτορα κ. Παύλο Αντωνίου ο οποίος ήταν περισσότερο από πρόθυμος να με βοηθήσει όποτε χρειάστηκε να τον ρωτήσω κάτι.

ΠΕΡΙΛΗΨΗ

Καθώς αυξάνονται οι κινητοί χρήστες που ενώνονται σε ασύρματα δίκτυα αυξάνονται και οι απαιτήσεις του εύρους ζώνης που χρειάζονται. Την ίδια ώρα οι χρήστες αυτοί γίνονται όλο και πιο απαιτητικοί σε θέμα ποιότητας των υπηρεσιών που λαμβάνουν καθώς έχουν συνηθίσει στον γρήγορο ρυθμό που προχωρά η τεχνολογία σήμερα. Η μετάδοση βίντεο σε πραγματικό χρόνο είναι ένας τομέας που κερδίζει συνεχώς έδαφος με τον αριθμό των ατόμων που αφορά να αυξάνεται διαρκώς σε κινητά και σταθερά δίκτυα. Λόγω της ανάγκης αυτής είχε δημιουργηθεί ο αλγόριθμος ADaptive Internet VIdео Server (ADIVIS), ένα σύστημα προσαρμοστικής μετάδοσης βίντεο σε ασύρματο διαδικτυακό περιβάλλον.

Το σύστημα ADIVIS υλοποιήθηκε αρχικά στο εργαλείο προσομοίωσης NS2. Η αρχική υλοποίηση του συστήματος υποστήριζε ασύρματα και ενσύρματα περιβάλλοντα unicast με κύριο στόχο να δοκιμαστεί ο αλγόριθμος ADIVIS σε περιβάλλον προσομοίωσης. Αργότερα έγινε προσθήκη υποστήριξης της υπηρεσίας multicast σε ενσύρματο περιβάλλον και τώρα προστέθηκε η υποστήριξη της υπηρεσίας multicast σε ασύρματο περιβάλλον καθώς και η υποστήριξη του πρωτόκολλου MIPv4 σε ασύρματο περιβάλλον.

Το αποτέλεσμα της διπλωματικής αυτής είναι μια σουίτα την οποία μπορεί να χρησιμοποιήσει εύκολα και γρήγορα κάποιος ερευνητής για να βγάλει όσο ακριβή αποτελέσματα γίνεται χωρίς να χρειάζεται να γνωρίζει την κάθε λεπτομέρεια υλοποίησης του συστήματος. Μια σουίτα που υποστηρίζει μια γκάμα από περιβάλλοντα δικτύων και έχει δυνατότητες να υποστηρίξει ακόμη περισσότερα.

ΠΕΡΙΕΧΟΜΕΝΑ

Κεφάλαιο 1	Εισαγωγή	1
Κεφάλαιο 2	Προηγούμενη σχετική εργασία	3
2.1	Εισαγωγή	3
2.2	Προσαρμοστική μετάδοση βίντεο	4
2.3	Το σύστημα ADIVIS	4
2.3.1	Αλγόριθμος ADIVIS (ADaptive Internet VIdeo Streaming)	6
2.3.2	Κωδικοποίηση σε Στρώματα (Layered Encoding)	10
2.3.3	Μηχανισμός Ανατροφοδότησης (Feedback Mechanism)	10
2.3.4	Προσθήκη υποστήριξης multicast σε ενσύρματο περιβάλλον	11
2.4	Multicast αλγόριθμοι δρομολόγησης σε ασύρματο περιβάλλον	13
2.4.1	PUMA (Protocol for Unified Multicasting through Announcements)	13
2.4.2	MAODV(Multicast Adhoc On Demand Distance Vector).....	15
Κεφάλαιο 3	Περιβάλλον προσομοίωσης	17
3.1	Εισαγωγή	17
3.2	Σύντομη περιγραφή NS2	18
3.3	Υποστήριξη multicast αλγορίθμων σε NS2.....	20
Κεφάλαιο 4	Καινούρια σουίτα ADIVIS	21
4.1	Διαδικασία εξαγωγής αποτελεσμάτων	22
4.1.1	Εισαγωγή	22
4.1.2	Επιλογή Βίντεο	22
4.1.3	Δημιουργία Στρωμάτων & Μετατροπή σε NS2 input.....	23

4.1.4 Προσομοίωση σε NS2	24
4.1.5 Συναρμολόγηση βίντεο	25
4.1.6 Ανάλυση αποτελεσμάτων	27
4.2 Δομή Αρχαιοθέτησης	27
4.2.1 Εισαγωγή	27
4.2.2 Παλαιά δομή αρχειοθέτησης	28
4.2.3 Καινούρια δομή αρχειοθέτησης.....	29
4.2.4 Διαχείριση κατάλογων	31
4.2.5 Συμβατότητα με διαφορετικές εκδόσεις NS2	32
4.3 Δημιουργία στρωμάτων & Συναρμολόγηση βίντεο	36
4.3.1 Εισαγωγή	36
4.3.2 Παλαιά διαδικασία επεξεργασίας βίντεο	36
4.3.3 Καινούρια διαδικασία επεξεργασίας βίντεο	38
4.3.4 Αποτελέσματα με τους 2 διαφορετικούς αλγόριθμους.....	39
4.4 Διορθώσεις/Βελτιώσεις συστήματος ADIVIS.....	41
Κεφάλαιο 5 Περιβάλλοντα προσομοίωσης & Ενδεικτικά σενάρια	48
5.1 Εισαγωγή	48
5.2 Υποστηριζόμενα περιβάλλοντα προσομοίωσης	49
5.3 Ενδεικτικό σενάριο για ασύρματο περιβάλλον Unicast + MIPv4 + ADIVIS	51
5.4 Ενδεικτικό σενάριο για ασύρματο περιβάλλον Multicast+ADIVIS (PUMA)	54
5.5 Αποτελέσματα προσομοίωσης PUMA vs MAODV	57
Κεφάλαιο 6 Συμπεράσματα & Μελλοντική εργασία.....	60
6.1 Συμπεράσματα	60
6.2 Μελλοντική Εργασία	61
Βιβλιογραφία	62

Κεφάλαιο 1

Εισαγωγή

Η συνεχής εξέλιξη της τεχνολογίας οδήγησε στην ευρύτερη χρήση φορητών ασύρματων συσκευών που έχουν πρόσβαση στο διαδίκτυο. Η ανοδική τάση της χρήσης αυτής, δημιούργησε την ανάγκη εύρεσης νέων τρόπων αποστολής δεδομένων σε κινητές συσκευές με απώτερο σκοπό να ικανοποιηθούν οι χρήστες κινητών δικτύων στο ίδιο βαθμό με τους χρήστες σταθερών δικτύων. Ο σκοπός αυτός είναι αρκετά δύσκολο να επιτευχθεί λόγω της απροσδιοριστίας που χαρακτηρίζει τα ασύρματα και κινητά δίκτυα από πλευράς εύρους ζώνης, καθυστέρησης και χαμένων πακέτων. Οι προκλήσεις αυτές γίνονται ακόμα πιο αισθητές σε πιο απαιτητικές εφαρμογές.

Μια τέτοια εφαρμογή, που είναι από τις πιο διαδεδομένες σήμερα, είναι η μετάδοση βίντεο (video streaming) σε πραγματικό χρόνο στην οποία έχει μεγάλο αντίκτυπο η προαναφερόμενη απροσδιοριστία. Σε συνδυασμό με τα πιο πάνω, η αλματώδης αύξηση των ετερογενών συσκευών με δυνατότητα προβολής βίντεο και πρόσβασης σε ασύρματα δίκτυα, καθιστά ακόμα πιο αναγκαία την εξεύρεση αποδοτικών και αποτελεσματικών τεχνικών για προσαρμοσμένη μετάδοση βίντεο, ώστε να ικανοποιούνται καλύτερα οι περιορισμοί, οι απαιτήσεις και οι ικανότητες των ασύρματων και κινητών δικτύων, εφαρμογών, υπηρεσιών και τελικών χρηστών. Η προσέγγιση του ADIVIS προς το πρόβλημα αυτό συνδυάζει τεχνικές προσαρμογής περιεχομένου (Content Adaptation Techniques - CATs) και τεχνικές προσαρμογής δικτύου (Network Adaptation Techniques - NATs) με τελικό στόχο να προσαρμόσει το bit stream στις μεταβαλλόμενες παραμέτρους του δικτύου. Τα NATs είναι υπεύθυνα για την end to end προσαρμογή των αναγκών των εφαρμογών πολυμέσων πραγματικού χρόνου στις συνθήκες του δικτύου, χρησιμοποιώντας

αλγορίθμους που λαμβάνουν υπόψη την κατάσταση και τον φόρτο του δικτύου, καθώς και τους τύπους των σφαλμάτων που συμβαίνουν. Ο ρόλος των CATs είναι να προσαρμόσουν το περιεχόμενο στον επιθυμητό ρυθμό μετάδοσης χρησιμοποιώντας κυρίως προσεγγίσεις scalable video.

Σκοπός αυτής της ΑΔΕ ήταν να προσθέσει την υποστήριξη της υπηρεσίας multicast καθώς και του πρωτοκόλλου MIPv4 σε ασύρματο περιβάλλον. Έτσι θα φανεί η λειτουργία του ADIVIS σε αυτά τα περιβάλλοντα και θα ανοίξει επίσης ο δρόμος για την υποστήριξη του ασύρματου περιβάλλοντος που συνδυάζει και τα δύο (multicast+MIPv4). Επίσης θεωρήθηκε αναγκαίο ως μέρος της διπλωματικής αυτής να δημιουργηθεί μια ολοκληρωμένη σουίτα με καινούρια δομή αρχειοθέτησης, με ξεκάθαρη διαδικασία για εξαγωγή αποτελεσμάτων καθώς και με διορθώσεις/βελτιώσεις του συστήματος ADIVIS.

Η δομή αυτής της ΑΔΕ είναι η εξής:

- Κεφάλαιο 1: Περιγραφή του προβλήματος που οδήγησε στην ανάγκη δημιουργίας αυτής της ΑΔΕ.
- Κεφάλαιο 2: Περιγραφή προηγούμενης σχετικής εργασίας στην οποία βασίστηκε αυτή η ΑΔΕ.
- Κεφάλαιο 3: Περιγραφή του περιβάλλοντος προσομοίωσης που χρησιμοποιήθηκε για την υλοποίηση της σουίτας ADIVIS.
- Κεφάλαιο 4: Ανάλυση της καινούριας σουίτας ADIVIS, η οποία αιτιολογεί τις αλλαγές και προσθήκες που έχουν γίνει και επίσης επεξηγεί τον τρόπο χρήσης της σουίτας και τις δυνατότητές της.
- Κεφάλαιο 5: Καταγραφή των υποστηριζόμενων περιβάλλοντων προσομοίωσης καθώς και ανάλυση κάποιων ενδεικτικών αποτελεσμάτων που αφορούν τα περιβάλλοντα προσομοίωσης στα οποία προστέθηκε η υποστήριξη τους ως μέρος αυτής της ΑΔΕ.
- Κεφάλαιο 6: Περιγραφή των συμπερασμάτων που εξάγονται από τα αποτελέσματα του κεφαλαίου 5 και εισηγήσεις για μελλοντική εργασία που θα μπορούσε να βασιστεί πάνω σε αυτή την ΑΔΕ.

Κεφάλαιο 2

Προηγούμενη σχετική εργασία

2.1 Εισαγωγή

2.2 Προσαρμοστική μετάδοση βίντεο

2.3 Το σύστημα ADIVIS

2.3.1 Αλγόριθμος ADIVIS (ADaptive Internet Video Streaming)

2.3.2 Κωδικοποίηση σε Στρώματα (Layered Encoding)

2.3.3 Μηχανισμός Ανατροφοδότησης (Feedback Mechanism)

2.3.4 Προσθήκη υποστήριξης multicast σε ενσύρματο περιβάλλον

2.4 Multicast αλγόριθμοι δρομολόγησης σε ασύρματο περιβάλλον

2.4.1 PUMA (Protocol for Unified Multicasting through Announcements)

2.4.2 MAODV(Multicast Adhoc On Demand Distance Vector)

2.1 Εισαγωγή

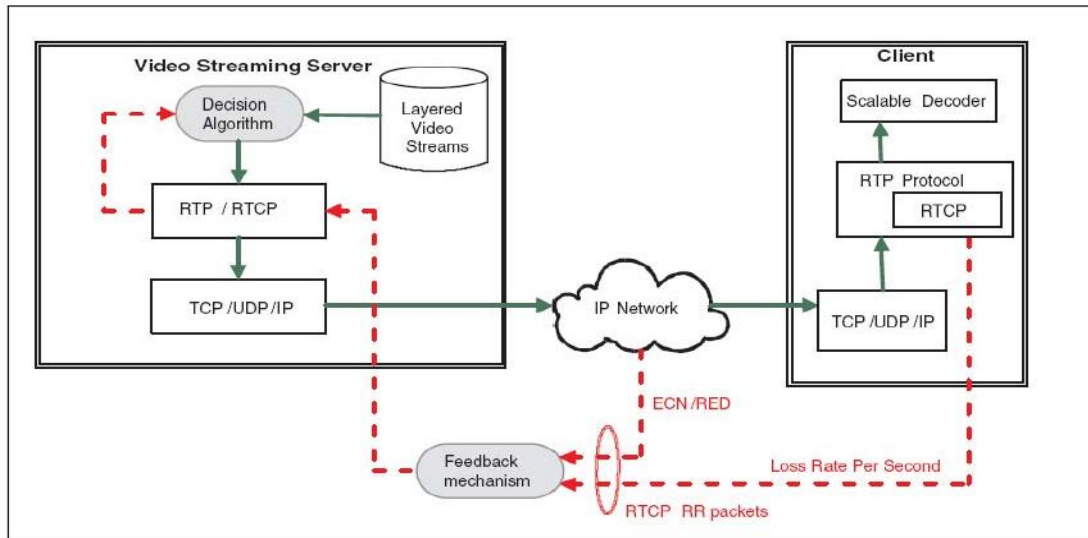
Η διπλωματική αυτή βασίστηκε σε προηγούμενη εργασία που έγινε πάνω στην προσαρμοστική μετάδοση βίντεο (από το Πανεπιστήμιο Πάτρας [4]) και στο σύστημα ADIVIS (από το Πανεπιστήμιο Κύπρου [2])

2.2 Προσαρμοστική μετάδοση βίντεο

Αρκετή δουλειά στον τομέα της προσαρμοστικής μετάδοσης βίντεο έχει γίνει από το πανεπιστήμιο Πάτρας. Συγκεκριμένα δημιούργησαν το πρωτόκολλο ASMP (Adaptive Smooth Multicast Protocol) [4]. Το πρωτόκολλο αυτό εξυπηρετεί στον έλεγχο της συμφόρησης όταν χρησιμοποιείται η υπηρεσία multicast για μετάδοση πολυμεσικού περιεχομένου σε best-effort δίκτυα. Οι ιδιότητες του που μας αφορούν είναι ότι προσαρμόζεται σε μικρές και μεγάλες ομάδες παραλήπτων, χρησιμοποιεί μεγάλο ποσοστό του εύρους ζώνης και έχει ομαλές ταχύτητες μετάδοσης που είναι σημαντικό για εφαρμογές μετάδοσης βίντεο. Για να επιτευχθούν αυτές οι ιδιότητες και στο σύστημα ADIVIS σε περιβάλλον όπου υποστηρίζεται η υπηρεσία multicast χρησιμοποιήθηκε παρόμοιος αλγόριθμος με αυτόν του ASMP ο οποίος φαίνεται στο υποκεφάλαιο 2.3.4.

2.3 Το σύστημα ADIVIS

Το ADIVIS περιλαμβάνει ένα adaptive feedback μηχανισμό για video streaming στο διαδίκτυο και ένα αλγόριθμο συγκεχυμένης απόφασης. Θεωρούμε ότι κάθε βίντεο είναι κωδικοποιημένο σε πολλαπλά στρώματα και αποθηκευμένο στην πλευρά του αποστολέα. Η μετάδοση του βίντεο γίνεται πάνω από μια σύνδεση RTP. Ο μηχανισμός ανατροφοδότησης που χρησιμοποιείται συνδυάζει τις σημαντικές πληροφορίες του παραλήπτη αναφορικά με την αντιληπτή για αυτόν ποιότητα μαζί με τις μετρήσεις που λαμβάνονται από τον πυρήνα του δικτύου με σκοπό να προβεί σε αξιολόγηση του διαθέσιμου εύρους ζώνης του μονοπατιού. Το διαθέσιμο εύρος ζώνης που προκύπτει από τον υπολογισμό, εισάγεται στον αλγόριθμο που αποφασίζει με συγκεχυμένο τρόπο τον βέλτιστο αριθμό στρωμάτων που θα πρέπει να μεταδοθούν, προσθέτοντας ή αφαιρώντας στρώματα



Σχήμα 2.1 Σύστημα Βασισμένο στο ADIVIS [2]

Όπως είναι φανερό από την πιο πάνω διάταξη, οι δύο μηχανισμοί, ανατροφοδότησης και συγκεχυμένης απόφασης, στοχεύουν στην προσαρμογή του στρωματοποιημένου περιεχομένου στο διαθέσιμο εύρος ζώνης. Με διακεκομμένες γραμμές παρατηρούμε την ροή των πακέτων ελέγχου, ενώ με τις συμπαγείς γραμμές φαίνεται η κίνηση των πακέτων δεδομένων βίντεο. Ο αλγόριθμος ανατροφοδότησης συλλέγει πληροφορίες σχετικά με την ποιότητα υπηρεσίας, τόσο από τον πυρήνα του δικτύου όσο και από τον παραλήπτη, οι οποίες θα χρησιμοποιηθούν για την εκτίμηση του διαθέσιμου εύρους ζώνης. Ο αλγόριθμος απόφασης που είναι υλοποιημένος στην πλευρά του αποστολέα, επεξεργάζεται τις πληροφορίες αυτές και αποφασίζει για τον βέλτιστο αριθμό στρωμάτων που θα αποσταλούν. Ο ρόλος των στοιχείων ανατροφοδότησης και προσαρμογής είναι να συνδέσουν τις απαιτήσεις ποιότητας της εφαρμογής βίντεο, με το δίκτυο πάνω από το οποίο εκτελείται. Η προσαρμογή στην κατάσταση του δικτύου, οφείλει να ακολουθείται και από προσαρμογή στο περιεχόμενο, που επιτυγχάνεται μέσα από κωδικοποίηση στρωματοποιημένου βίντεο.

2.3.1 Αλγόριθμος ADIVIS (ADaptive Internet Video Streaming)

Σαν εισαγωγή αναφέρεται ότι η συγκεκριμένη λογική χρησιμοποιείται για τον σχεδιασμό χειριστηρίων της ανατροφοδότησης σε περιπτώσεις όπου η θεωρητική προσέγγιση για απόλυτο έλεγχο δεν μπορεί να εφαρμοστεί πρακτικά, λόγω της μη δυνατότητας εξαγωγής κατάλληλου αυστηρού αναλυτικού μοντέλου. Ο αλγόριθμος που μελετούμε, διεξάγεται στην πλευρά του αποστολέα, ο οποίος αρχικά επεξεργάζεται την πληροφορία ανατροφοδότησης και αποφασίζει τον βέλτιστο αριθμό στρωμάτων που θα αποσταλούν, χρησιμοποιώντας, όπως προαναφέρθηκε συγκεκριμένη λογική έξοδο. Βασίζεται σε δύο γλωσσικές μεταβλητές εισόδου και μία μεταβλητή εξόδου. Όλες οι ποσότητες περιλαμβάνουν την διακριτή σταθερά kT όπου T ορίζεται σαν η περίοδος απόφασης.

Η πρώτη γλωσσική μεταβλητή σχετίζεται με την παράμετρο LRPS. Σαν LRPS(kT) ορίζεται ο ρυθμός απώλειας ανά δευτερόλεπτο εντός κάθε περιόδου απόφασης και ως LRPS($kT-T$) ο ρυθμός απώλειας ανά δευτερόλεπτο με μία καθυστέρηση T . Η γλωσσική μεταβλητή DLRPS(kT) είναι μια ένδειξη της αυξητικής ή μειωτικής τάσης του LRPS και μπορεί να υπολογιστεί με τον ακόλουθο τύπο:

$$DLRPS(kT) = LRPS(kT) - LRPS(kT - T)$$

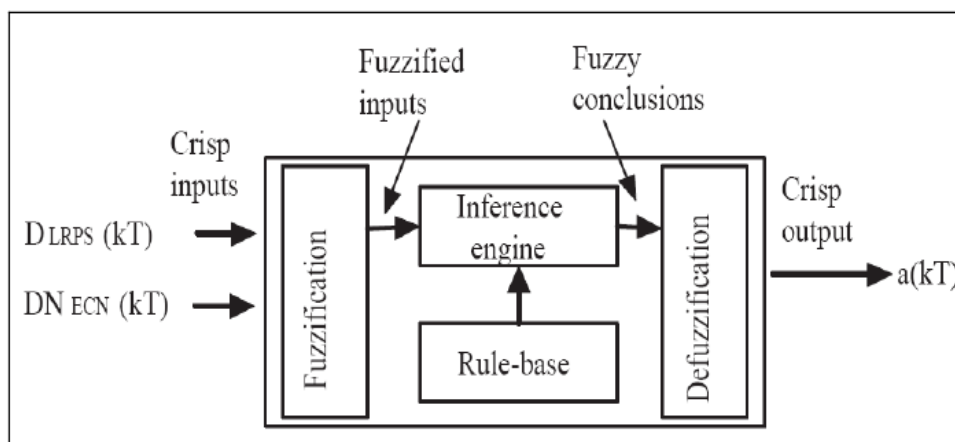
Οι τιμές της παραμέτρου LRPS κυμαίνονται από 0 έως 1 και ως εκ τούτου το κάτω και άνω φράγμα του DLRPS(kT) είναι -1 και 1 αντίστοιχα. Ως δεύτερη γλωσσική μεταβλητή χρησιμοποιείται ο αριθμός των πακέτων που έχουν ενεργοποιημένο το ECN bit εντός μιας περιόδου, σαν σαφής ένδειξη της συμφόρησης. Ο παραλήπτης περιοδικά υπολογίζει τον αριθμό αυτό που αντιπροσωπεύεται από την μεταβλητή NECN(KT) και τον αποστέλλει στα πλαίσια των RR πακέτων που εκπέμπει προς τον αποστολέα. Ο τελευταίος, στη συνέχεια, εξάγει την τιμή αυτή από τα RRs και υπολογίζει μία σταθμισμένη παράμετρο, το NECNsc(kT), που κυμαίνεται από -1 έως +1 και αναπαριστά το ποσοστό των πακέτων που είχαν ενεργοποιημένο το ECN bit εντός μίας περιόδου. Για τον υπολογισμό της σταθμισμένης παραμέτρου χρησιμοποιείται ο ακόλουθος τύπος:

$$NECNsc(kT) = NECN(kT)/Nps(kT)$$

όπου $Nps(kT)$ είναι ο αριθμός των πακέτων που αποστέλλονται εντός της ίδιας περιόδου. Συνεπώς, υπολογίζουμε την τιμή της παραμέτρου $DNECNsc(kT)$, η οποία καταδεικνύει την αυξητική ή μειωτική τάση του αριθμού των μαρκαρισμένων με ECN bit πακέτων. Η τιμή της μεταβλητής κυμαίνεται από -1 έως +1 και υπολογίζεται από τον τύπο που ακολουθεί:

$$DNECNsc(kT) = NECNsc(kT) - NECNsc(kT - T)$$

Η τιμή εξόδου έχει προεπιλεγεί να παίρνει τιμές από 0.5 έως 1.5. Έτσι, επιτρέπεται μία βαθμιαία αύξηση όταν υπάρχει διαθέσιμο εύρος ζώνης και χαμηλή συμφόρηση, ενώ αντίθετα σε περίπτωση έντονης συμφόρησης λαμβάνονται γρήγορα μέτρα ώστε να μειωθεί στο μισό ο ρυθμός μετάδοσης δεδομένων. Η τιμή εξόδου δεν έχει επιλεγεί να καθορίζει ρητά τον αριθμό στρωμάτων που θα έπρεπε να αποσταλούν, αλλά αντίθετα είναι μία καθαρή τιμή η οποία επιτρέπει στον αλγόριθμο να χρησιμοποιηθεί σε περισσότερες περιπτώσεις.



Σχήμα 2.2 Σύστημα Ελέγχου Συγκεχυμένης Λογικής [2]

Η τιμή της μεταβλητής εξόδου $a(kT)$, μπορεί να χρησιμοποιηθεί από τον αλγόριθμο απόφασης για την εκτίμηση του διαθέσιμου εύρους ζώνης βάση του τύπου που παρατίθεται στην συνέχεια:

$$\text{avail_bw}(kT) = a(kT) * \text{avail_bw}(kT - T)$$

Ο αλγόριθμος του ADIVIS, πρέπει να αποφασίσει ποια στρώματα θα αποσταλούν ανάλογα με το διαθέσιμο εύρος ζώνης του δικτύου, βασισμένος σε μία μη δυναμική προσέγγιση επιλογής στρωμάτων. Ο εξυπηρετητής θα έχει αποθηκευμένο ένα κατάλληλο αριθμό στρωμάτων, σε κάθε ένα εκ των οποίων αντιστοιχεί διαφορετικός ρυθμός μετάδοσης. Για την αποφυγή βεβιασμένων ενεργειών που θα απαιτούν άμεση αντιστροφή στην προηγούμενη κατάσταση, δεν πρέπει να υπάρχει άμεση μετάβαση σε ανώτερο στρώμα κάθε φορά που το διαθέσιμο εύρος ζώνης ξεπερνά το κατώφλι ενός συγκεκριμένου ρυθμού μετάδοσης, που αντιστοιχεί σε ψηλότερο στρώμα. Αντίθετα, το σύστημα θα πρέπει να μπαίνει σε μια φάση χρονικής υστέρησης, ώστε να αποφευχθούν τακτικές μεταβάσεις από ένα στρώμα σε άλλο. Αντίθετα, στην περίπτωση μετάβασης σε χαμηλότερο στρώμα, οι ενέργειες είναι άμεσες, καθώς προσπαθούμε να ανακάμψουμε από συμφόρηση όσο το δυνατό συντομότερα. Η χρονική υστέρηση είναι ίση με το χρονικό διάστημα μεταξύ της παραλαβής δύο διαδοχικών RR πακέτων. Αν το διαθέσιμο εύρος ζώνης υπερβεί το κατώφλι ενός συγκεκριμένου ρυθμού μετάδοσης που ανταποκρίνεται σε στρώμα ανώτερου επιπέδου, ενεργοποιείται η μεταβλητή της χρονικής υστέρησης. Όταν ένα νέο πακέτο RR παραληφθεί, αν το διαθέσιμο εύρος ζώνης παραμένει στα ίδια επίπεδα, τότε έχουμε μετάβαση σε ψηλότερο στρώμα. Οι πιο πάνω συνθήκες συνοψίζονται στον ακόλουθο ψευδοκώδικα που περιγράφει τη διαδικασία μετάβασης ανάμεσα σε στρώματα:

For all layers j up to MAX LAYER

```

    If avail_bw <= BitRate(j) and hysteresis=false {
        If layer(j) = current_layer
            break;
        hysteresis = true;
        selected layer = layer(j);
    }
    Else If avail_bw <= BitRate(j) {
        hysteresis = false;
    }

```

```

    If selected_layer < layer(j) {
        current_layer = selected_layer;
        break;
    }
else {
    current_layer = layer(j);
    break;
}
}

```

Στον πίνακα που ακολουθεί παρουσιάζονται οι γλωσσικοί κανόνες που χρησιμοποιούνται στα πλαίσια του ADIVIS και συνδέουν τις γλωσσικές μεταβλητές εισόδου $DN_{ECNsc}(kT)$ και $DLRPS(kT)$ με την μεταβλητή εξόδου $a(kT)$. Ουσιαστικά, πρέπει να ληφθούν σαν κανόνες if-then, ώστε να διατυπωθούν οι συνθήκες που αποτελούν τη συγκεχυμένη λογική.

a(kT)		DN _{ECN_{sc}} (kT)						
		NVB	NB	NS	Z	PS	PB	PVB
D _{LRPS} (kT)	NVB	H	H	B	B	Z	S	VS
	NB	H	VB	Z	Z	Z	S	VS
	NS	B	Z	B	Z	Z	S	VS
	Z	B	Z	Z	B	Z	S	VS
	PS	Z	Z	Z	Z	S	S	VS
	PB	Z	Z	Z	Z	S	S	VS
	PVB	S	S	S	S	VS	VS	VS

Σχήμα 2.3 Γλωσσικοί Κανόνες [2]

Η σημασιολογία της σημειογραφίας του πίνακα έχει ως ακολούθως: (NVB/ PVB) Negative/Positive Very Big, (NB/PB) Negative/Positive Big, (NS/ PS) Negative/Positive Small, (Z) Zero, (VS/ VB) Very Small/Big, (S/ B) Small/Big, (M) Medium και (H) Huge αναφέρονται στις τιμές που μπορούν πάρουν οι τιμές εισόδου, αλλά και η τιμή εξόδου.

2.3.2 Κωδικοποίηση σε Στρώματα (Layered Encoding)

Η τεχνική αυτή είναι κατάλληλη για την προσαρμογή της ποσότητας των δεδομένων που μεταδίδονται από ένα εξυπηρετητή βίντεο στην χωρητικότητα ενός δεδομένου μονοπατιού. Οι ροές βίντεο είναι κωδικοποιημένες με στρωματοποιημένο τρόπο, ώστε κάθε επιπρόσθετο στρώμα να αυξάνει την αντιληπτή ποιότητα της ροής. Η συνήθης τακτική είναι το στρωματοποιημένο βίντεο να αποτελείται από ένα βασικό στρώμα και επιπλέον ενισχυτικά στρώματα. Το βασικό στρώμα, εντούτοις, είναι προτιμότερο να είναι κωδικοποιημένο σε χαμηλό bitrate, ούτως ώστε να εξυπηρετεί μεγάλη γκάμα κινητών συσκευών, καθώς και τερματικά τα οποία συνδέονται στο διαδίκτυο με συνδέσεις χαμηλού εύρους ζώνης. Εξάλλου, επιπρόσθετα στρώματα προστίθενται ή αφαιρούνται, έτσι ώστε να προσαρμοστεί το bitrate του περιεχομένου στον επιθυμητό ρυθμό μετάδοσης.

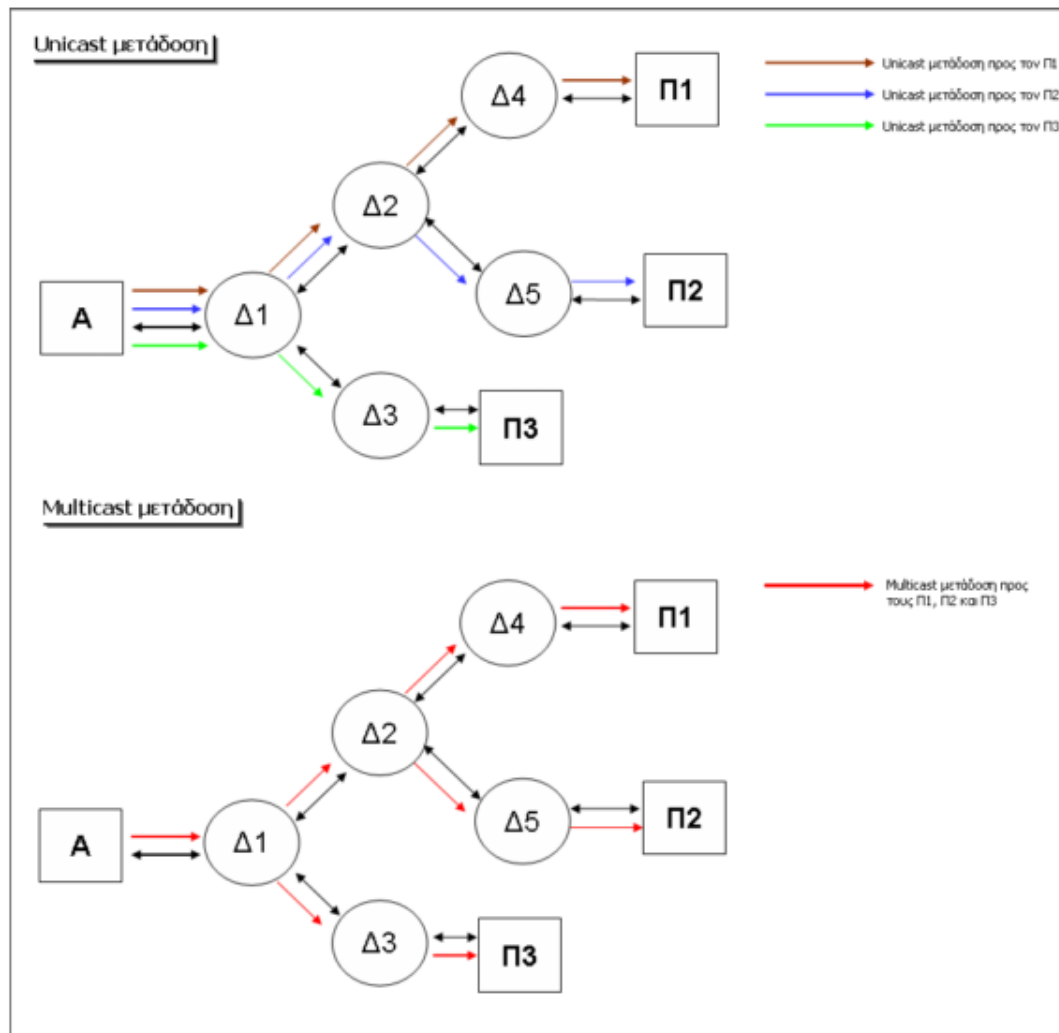
2.3.3 Μηχανισμός Ανατροφοδότησης (Feedback Mechanism)

Κάθε παραλήπτης αποστέλλει στατιστικές παραλαβής χρησιμοποιώντας πακέτα RTCP. Συγκεκριμένα, τα πακέτα που στέλνονται από τους συμμετέχοντες οι οποίοι δεν είναι ενεργοί αποστολείς, είναι Receiver Reports- RR και περιέχουν στατιστικές παραλαβής. Το ποσοστό απώλειας πακέτων μέσα σε ένα χρονικό διάστημα δίνεται από τον αριθμό των αναμενόμενων πακέτων διαιρεμένο με τον αριθμό των χαμένων πακέτων μέσα σε αυτό το χρονικό διάστημα. Ο ρυθμός απώλειας ανά δευτερόλεπτο (LRPS), προκύπτει διαιρώντας το ποσοστό απώλειας, με τη διαφορά στους χρόνους (timestamps) μεταξύ των RRs. Η διαφορά ανάμεσα σε δύο διαδοχικές τιμές LRPS, μπορεί να χρησιμοποιηθεί για την ανίχνευση αυξητικής ή μειωτικής τάσης σχετικά με το ποσοστό απώλειας πακέτων. Σαν ένα δεύτερο συστατικό του μηχανισμού αυτού, έρχεται να προστεθεί η έννοια της ρητής ειδοποίησης σχετικά με τη συμφόρηση στο δίκτυο (Explicit Congestion Notification-ECN). Σύμφωνα με την τεχνική αυτή, τα στοιχεία του δικτύου, όπως για παράδειγμα οι δρομολογητές, μπορούν να ειδοποιήσουν ρητά τον αποστολέα αναφορικά με την τρέχουσα κατάσταση της συμφόρησης εντός του πυρήνα του δικτύου, κάτι που μπορεί να χρησιμοποιηθεί αποδοτικά για την αξιολόγηση του διαθέσιμου εύρους ζώνης. Οι

ενημερώσεις ECN, μπορούν επιπλέον να εφαρμοστούν ώστε να ειδοποιηθούν έγκαιρα οι τερματικοί κόμβοι και να προληφθούν μη αναγκαίες απώλειες πακέτων.

2.3.4 Προσθήκη υποστήριξης multicast σε ενσύρματο περιβάλλον

Στο σχήμα 2.4 που παρουσιάζω πιο κάτω, φαίνεται ξεκάθαρα ο τρόπος μετάδοσης unicast και multicast. Μπορούμε να διακρίνουμε με διαφορετικό χρώμα τις διαφορετικές βιντεοροές που στέλνει ο αποστολέας στην περίπτωση της μετάδοσης unicast. Στην περίπτωση της μετάδοσης multicast παρατηρούμε ότι όλοι οι παραλήπτες λαμβάνουν από την ίδια βιντεοροή (κόκκινο χρώμα).

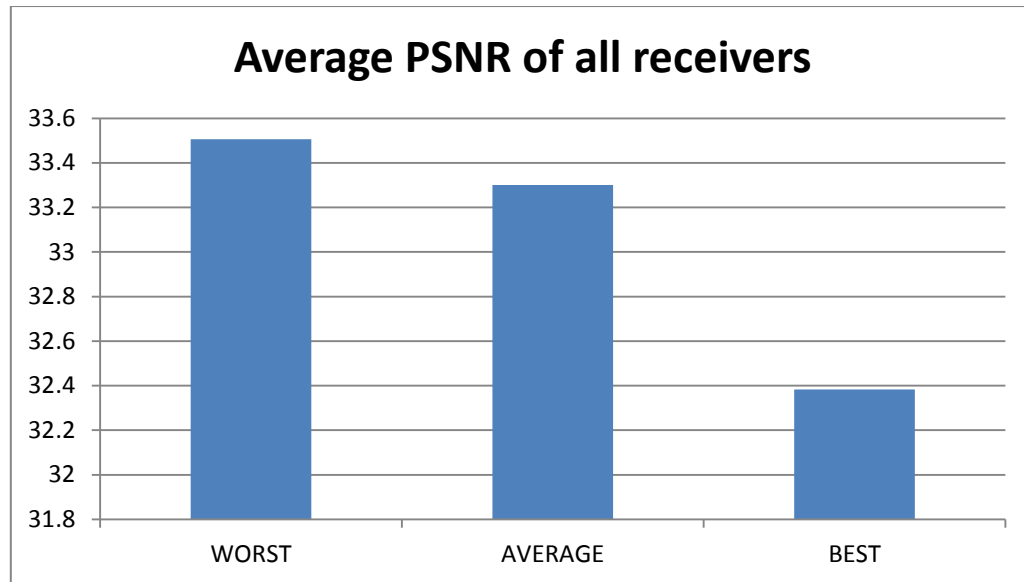


Σχήμα 2.4 Μετάδοση Unicast & Multicast [13]

Αυτή η προσθήκη επιτρέπει στο σύστημα ADIVIS να υποστηρίζει την υπηρεσία multicast σε ενσύρματο περιβάλλον. Ουσιαστικά η διαφορά από πλευράς CAT (Content Adaptation Technique) είναι το πώς λαμβάνονται υπόψη τα πακέτα RR (Receiver Reports). Προηγουμένως ο αποστολέας έστελνε διαφορετική ροή βίντεο σε κάθε παραλήπτη όποτε λάμβανε υπόψη ξεχωριστά το κάθε RR και έστελνε πακέτα από το ανάλογο κωδικοποιημένο στρώμα που ήταν διαφορετικά για τον κάθε παραλήπτη. Όταν όμως χρησιμοποιείται η υπηρεσία multicasting η ροή βίντεο που παίρνουν οι παραλήπτες είναι η ίδια, οπότε αλλαγή στο κωδικοποιημένο στρώμα που στέλνεται επηρεάζει όλους τους παραλήπτες. Έτσι πρέπει να λαμβάνονται υπόψη τα πακέτα RR από όλους τους παραλήπτες μαζί. Αυτό επιτυγχάνεται προγραμματίζοντας τους παραλήπτες να στέλνουν τα πακέτα αυτά την ίδια χρονική στιγμή κάθε 0.5 δευτερόλεπτα. Ο αποστολέας με την σειρά του αποθηκεύει τις πληροφορίες του τελευταίου πακέτου RR που έστειλε ο κάθε παραλήπτης και κάθε 0.5 δευτερόλεπτα στέλνει το LRPS (Loss Rate Per Second) που αντιστοιχεί στις ρυθμίσεις που θέσαμε πριν από την προσομοίωση. Συγκεκριμένα δίνουμε την επιλογή στον αποστολέα να βασιστεί:

- πάνω στο καλύτερο LRPS, δηλαδή η ροή βίντεο που στέλνει να εξυπηρετήσει καλύτερα αυτόν που έχει την καλύτερη λήψη
- πάνω στο χειρότερο LRPS, δηλαδή η ροή βίντεο που στέλνει να εξυπηρετήσει καλύτερα αυτόν που έχει την χειρότερη λήψη
- πάνω στον μέσο όρο LRPS, δηλαδή η ροή βίντεο που στέλνει να προσπαθεί να ικανοποιήσει όλους τους παραλήπτες

Με βάση την θεωρητική εργασία που έγινε στην ΑΔΕ του Στέφανου Χ. [13], όταν ο αποστολέας στηρίζεται στο feedback με το χειρότερο LRPS έχουμε την καλύτερη μέση ποιότητα. Στο πιο κάτω σχήμα βλέπου ότι αυτό όντως ισχύει και για το ενδεικτικό σενάριο του ασύρματου περιβάλλοντος Multicast+ADIVIS (PUMA).



Σχήμα 2.5 Μέσος όρος PSNR που αντιστοιχεί σε κάθε κριτήριο

2.4 Multicast αλγόριθμοι δρομολόγησης σε ασύρματο περιβάλλον

2.4.1 PUMA (Protocol for Unified Multicasting through Announcements)

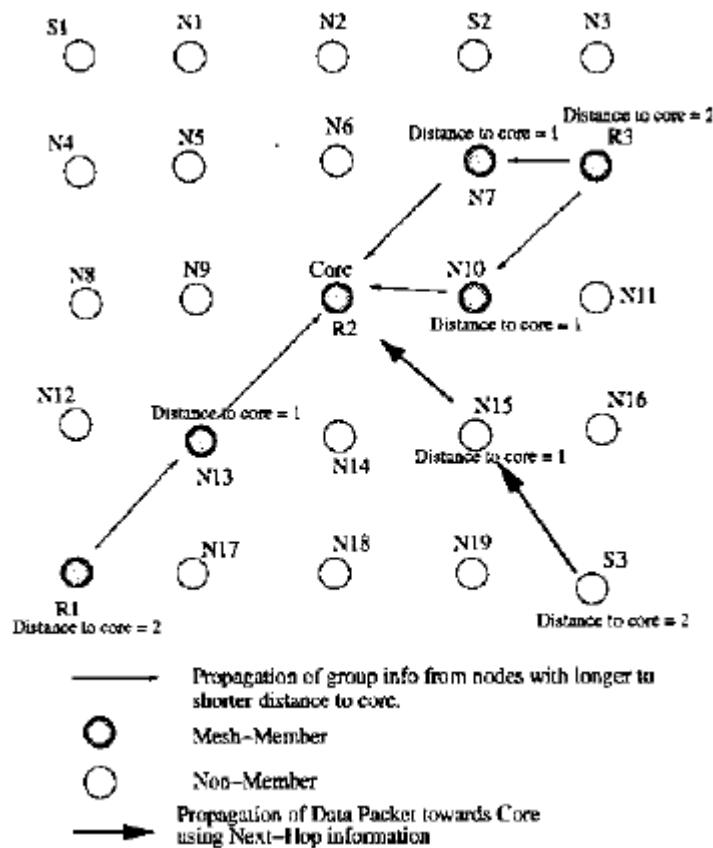
Το πρωτόκολλο PUMA [12] υποστηρίζει το μοντέλο της υπηρεσίας IP multicast αφήνοντας οποιαδήποτε πηγή να στείλει πακέτα multicast που απευθύνονται σε ένα συγκεκριμένο multicast group, χωρίς να χρειάζεται να ξέρει ποιοί αποτελούν αυτό το group. Επιπλέον, δεν είναι ανάγκη μια πηγή να είναι μέλος σε ένα multicast group για να μπορεί να στείλει πακέτα δεδομένων σε αυτό.

Όπως άλλα πρωτόκολλα που υποστηρίζουν την υπηρεσία IP multicast ασύρματα (π.χ. CAMP, MAODV), το PUMA χρησιμοποιεί μια προσέγγιση στην οποία οι παραλήπτες γίνονται μέλη ενός multicast group χρησιμοποιώντας τη διεύθυνση ενός ειδικού κόμβου, χωρίς να χρειάζεται να κάνει flood το δίκτυο με πακέτα ελέγχου ή δεδομένων από όλες τις πηγές του group αυτού. Το PUMA (όπως και το MAODV) αφαιρεί την ανάγκη χρήσης ενός πρωτοκόλλου δρομολόγησης unicast και τον προκαθορισμό πυρήνων στα multicast groups.

Υλοποιεί ένα καταναεμημένο αλγόριθμο για να επιλέξει ένα από τους παραλήπτες κάποιου group να είναι ο πυρήνας και να ενημερώνει κάθε δρομολογητή στο δίκτυο για τουλάχιστον τους κόμβους που βρίσκονται ένα hop μακριά του. Ο αλγόριθμος επιλογής που χρησιμοποιείται στο PUMA είναι ουσιαστικά ο ίδιος με τον δενδρικό αλγόριθμο επέκτασης που εισάχθηκε από τον Perlman για συνδεδεμένα δίκτυα από διαφανείς γέφυρες (internetworks of transparent bridges) [9]. Σε ένα πεπερασμένο χρονικό διάστημα ανάλογο του χρόνου που χρειάζεται για να φτάσει τον πιο μακρινό δρομολογητή από τον τελικό πυρήνα ενός group, κάθε δρομολογητής έχει ένα ή περισσότερα μονοπάτια προς τον επιλεγμένο πυρήνα.

Ο κάθε παραλήπτης ενώνεται με τον επιλεγμένο πυρήνα μέσω των πιο σύντομων μονοπατιών που τους ενώνουν. Όλοι οι κόμβοι που βρίσκονται στο πιο σύντομο μονοπάτι μεταξύ οποιουδήποτε παραλήπτη και του πυρήνα αποτελούν το πλέγμα (mesh). Ένας αποστολέας στέλνει ένα πακέτο δεδομένων στο group μέσω οποιουδήποτε από τα συντομότερα μονοπάτια που ενώνουν τον αποστολέα με τον πυρήνα. Όταν το πακέτο δεδομένων φθάσει κάποιο μέλος του πλέγματος, γίνεται flood μέσα στο πλέγμα και οι κόμβοι διατηρούν ένα cache με τα IDs των πακέτων για να απορρίπτουν πακέτα δεδομένων που έχουν λάβει ξανά.

Το PUMA χρησιμοποιεί ένα μοναδικό μήνυμα ελέγχου για όλες του τις λειτουργίες, την ανακοίνωση multicast. Κάθε ανακοίνωση multicast ορίζει ένα αριθμό ακολουθίας, την διεύθυνση ενός group (group ID), την διεύθυνση του πυρήνα (core ID), την απόσταση προς τον πυρήνα, μια σημαία που είναι ενεργοποιημένη όταν ο αποστολέας είναι μέλος του πλέγματος (mesh member flag) και ένα γονέα που δηλώνει τον γείτονα που προτιμά για να φθάσει στον πυρήνα. Συνεχόμενες ανακοινώσεις multicast έχουν ένα ψηλότερο ακολουθιακό αριθμό από τις προηγούμενες ανακοινώσεις multicast που στάληκαν από τον ίδιο πυρήνα. Με τις πληροφορίες που συμπεριλαμβάνονται σε τέτοιες ανακοινώσεις, οι κόμβοι επιλέγουν πυρήνες, αποφασίζουν τις διαδρομές για πηγές έξω από ένα multicast group σε unicast multicast πακέτα δεδομένων προς το group, ενημερώνουν άλλους για την ένταξη ή αποχώρηση τους από το πλέγμα ενός group και διατηρούν το πλέγμα του group. Η δομή του πλέγματος φαίνεται και στο σχήμα 2.6 το οποίο ακολουθεί.



Σχήμα 2.6 Δομή πλέγματος PUMA [12]

2.4.2 MAODV(Multicast Adhoc On Demand Distance Vector)

Το MAODV είναι η επέκταση multicast του πρωτοκόλλου δρομολόγησης AODV [8]. Η κάθε ομάδα multicast έχει μια μοναδική διεύθυνση. Σύμφωνα με τις προδιαγραφές του MAODV [10][11], η κάθε ομάδα multicast είναι οργανωμένη χρησιμοποιώντας μια δενδρική δομή, η οποία αποτελείται από τα μέλη της ομάδας και αρκετούς δρομολογητές, οι οποίοι δεν είναι μέλη της ομάδας αλλά πρέπει να υπάρχουν στο δέντρο για να ενώνουν τα μέλη της ομάδας. Δηλαδή τα μέλη της ομάδας και οι δρομολογητές είναι μέλη του δέντρου και ανήκουν στην ομάδα του δέντρου. Σχετικά με κάθε δέντρο multicast, το μέλος της ομάδας που κατασκευάζει πρώτο το δέντρο multicast είναι ο αρχηγός της ομάδας του

για εκείνο το δέντρο, υπεύθυνο για να διατηρεί το δέντρο της ομάδας στέλνοντας περιοδικά μέσω του μηχανισμού broadcast μηνύματα Group-Hello (GRPH) σε όλο το δίκτυο. Ο αρχηγός της ομάδας διατηρεί επίσης ο αριθμό ακολουθίας της ομάδας, ο οποίος διαδίδεται στο δίκτυο μέσω των μηνυμάτων GRPH.

Ο κάθε κόμβος στο δίκτυο μπορεί να διατηρήσει 3 πίνακες. Ο πρώτος είναι ο Unicast Route Table, που καταγράφει το επόμενο hop για διαδρομές σε άλλους προορισμούς με unicast traffic. Συνήθως, ο προορισμός είναι ένας από τους άλλους κόμβους στο δίκτυο. Η ειδική περίπτωση που υπάρχει είναι όταν ο προορισμός έχει μια διεύθυνση multicast, το οποίο συμβαίνει όταν ο κόμβος δεν είναι μέλος του δέντρου multicast αλλά έχει multicast πακέτα δεδομένων να στείλει στην συγκεκριμένη ομάδα multicast. Ο δεύτερος πίνακας είναι ο Multicast Route Table, ο οποίος κρατά τα επόμενα hops για τη δενδρική δομή της κάθε ομάδας multicast. Ο τρίτος πίνακας είναι ο Group Leader Table. Καταγράφει την παρούσα γνωστή διεύθυνση της ομάδας multicast μαζί με τον αρχηγό της ομάδας αυτής και το επόμενο hop προς τον αρχηγό αυτό όταν ο κόμβος παραλάβει ένα περιοδικό μήνυμα GRPH.

Κεφάλαιο 3

Περιβάλλον προσομοίωσης

3.1 Εισαγωγή

3.2 Σύντομη περιγραφή NS2

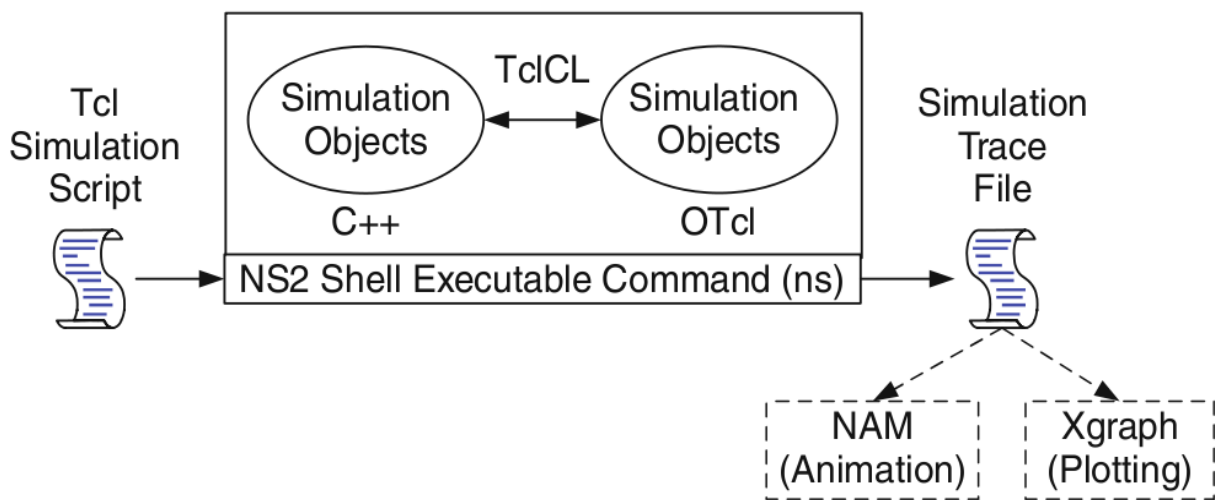
3.3 Υποστήριξη multicast αλγορίθμων σε NS2

3.1 Εισαγωγή

Ο NS2 (Network Simulator Version 2) είναι ένα εργαλείο προσομοίωσης δικτύων που στηρίζεται σε διακριτά γεγονότα. Στοχεύει να βοηθήσει την έρευνα στα δίκτυα καθώς επιτρέπει στους ερευνητές να προσομοιώσουν τις καινοτομίες τους πριν να τις δοκιμάσουν σε πραγματικό περιβάλλον κάτι το οποίο μπορεί να είναι αρκετά δαπανηρό ή χρονοβόρο και στο τέλος η καινοτομία να μην μπορεί να εφαρμοστεί. Δοκιμάζοντας την καινοτομία τους πρώτα στον NS μπορούν να πάρουν μια ιδέα πώς θα λειτουργήσει και σε πραγματικό περιβάλλον και να αποφασίσουν αν όντως επιφέρει τα επιθυμητά αποτελέσματά. Επειδή επιτρέπει στους ερευνητές να αλλάζουν την ίδια την πλατφόρμα του NS ή να προσθέσουν πάνω της ότι χρειάζεται για τις ανάγκες τους είναι πλέον από τα πιο διαδεδομένα εργαλεία για το σκοπό του. Στο κεφάλαιο αυτό θα περιγράψω σύντομα τον NS2 καθώς και την υποστήριξη που παρέχει για multicast αλγορίθμους.

3.2 Σύντομη περιγραφή NS2

Στο σχήμα 1.1 φαίνεται η βασική αρχιτεκτονική του NS2. Ο NS2 παρέχει στους χρήστες την εκτελέσιμη εντολή ns η οποία παίρνει σαν παράμετρο εισόδου, το όνομα ενός TCL script αρχείου προσομοίωσης. Αφού εκτελέσουν την εντολή αυτή ο NS2 τρέχει την προσομοίωση και στις περισσότερες περιπτώσεις δημιουργούνται κάποια αρχεία που ονομάζονται simulation trace files και έχουν αποθηκευμένα πληροφορίες για όλα τα πακέτα που σταλθήκαν και παραλήφθηκαν κατά τη διάρκεια της προσομοίωσης. Τα αρχεία αυτά χρησιμοποιούνται αργότερα για να εξαχθούν τα αποτελέσματά μέσω γραφικών παραστάσεων (οι οποίες μπορούν να δημιουργηθούν με το Xgraph) ή για να γίνει αναπαράσταση της προσομοίωσης μέσω κάποιου εργαλείου (όπως το NAM το οποίο συνοδεύει το NS2).

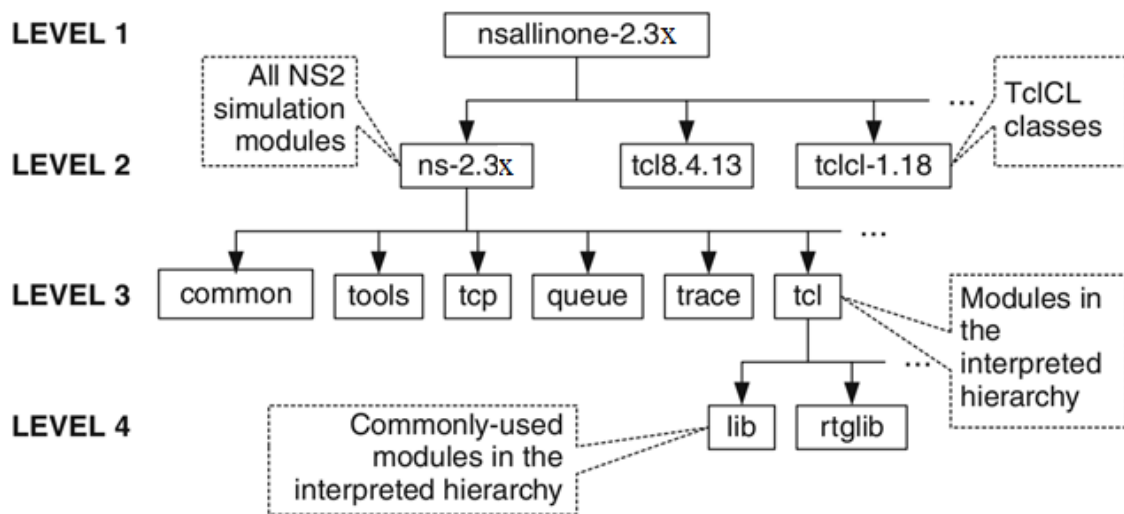


Σχήμα 3.1 Βασική αρχιτεκτονική του NS2

Το NS2 αποτελείται από 2 γλώσσες: C++ & OTcl (Object-Oriented Tool Command Language). Ενώ η C++ ορίζει τον εσωτερικό μηχανισμό των προσομοιωμένων αντικειμένων, η OTcl ρυθμίζει την προσομοίωση με το να δημιουργήσει και να αρχικοποιήσει τα αντικείμενα καθώς και να προγραμματίσει τα διακριτά γεγονότα. Η C++ και η OTcl είναι ενωμένες από την TclCL. Οι μεταβλητές σε πεδία της OTcl που

αντιστοιχούν σε ένα αντικείμενο της C++ αναφέρονται ως handles. Ένα handle στο πεδίο της OTcl είναι απλά ένα string και δεν έχει καμιά λειτουργία. Η λειτουργία του ορίζεται στο αντικείμενο C++ που του αντιστοιχεί. Με αυτό τον τρόπο χρησιμοποιείται ο ήδη υπάρχον κώδικας της C++ με διαφορετικές ρυθμίσεις χωρίς να χρειάζεται να γίνει ξανά compile όλος ο κώδικας αλλά κρατώντας την ταχύτητα που προσφέρει η C++ αφού το κομμάτι που παίρνει ο interpreter είναι μόνο οι ρυθμίσεις.

Λεπτομερείς οδηγίες για εγκατάσταση του NS2 καθώς και ο πηγαίος του κώδικας μπορούν να βρεθούν στο [6]. Είναι συμβατό με τα περισσότερα συστήματα UNIX καθώς και στα Windows κάτω από τον unix emulator Cygwin. Αφού γίνει η εγκατάστασή του, μπορεί κάποιος να τρέξει ένα από τα πολλά μοντέλα που ήδη υπάρχουν κάτω από τον κατάλογο “tcl” στους φακέλους “test” και “ex”. Στο σχήμα 1.2 φαίνεται η δομή καταλόγων που υπάρχει στον NS2 (είναι η ίδια και στην έκδοση 2.31). Αξίζει να σημειωθεί ότι σουίτα ADIVIS μπαίνει στο επίπεδο 3.



Σχήμα 3.2 Δομή καταλόγων του NS2

3.3 Υποστήριξη multicast αλγορίθμων σε NS2

Ο NS-2.31 όπως παρέχεται από την επίσημη ιστοσελίδα υποστηρίζει μόνο multicast αλγορίθμους σε ασύρματο περιβάλλον. Συγκεκριμένα υποστηρίζει τους εξής multicast αλγόριθμους:

- Centralized Multicast (CM)
- Dense Mode (DM)
- Shared Tree Mode (ST)
- Bi-directional Shared Tree Mode (BST)

Υπάρχουν όμως και έτοιμες προσθήκες (πολλές από τις οποίες μπορεί να βρεί κάποιος στο [7]) για αλγόριθμους που δεν υποστηρίζει επίσημα ο NS. Οι προσθήκες αυτές επιτρέπουν την χρησιμοποίηση επιπλέον αλγορίθμων multicast. Σημαντικό να αναφέρουμε ότι από αυτούς κάποιοι λειτουργούν σε ασύρματο περιβάλλον. Για τους σκοπούς αυτής της ΑΔΕ επιλέχθηκαν οι προσθήκες των multicast αλγορίθμων MAODV και PUMA οι οποίοι λειτουργούν σε ασύρματο περιβάλλον και όπως αναφέρεται στις μετρήσεις που έγιναν στο [12] έχουν καλύτερες επιδόσεις από άλλους αλγόριθμους multicast που υπάρχουν σε ασύρματο περιβάλλον.

Κεφάλαιο 4

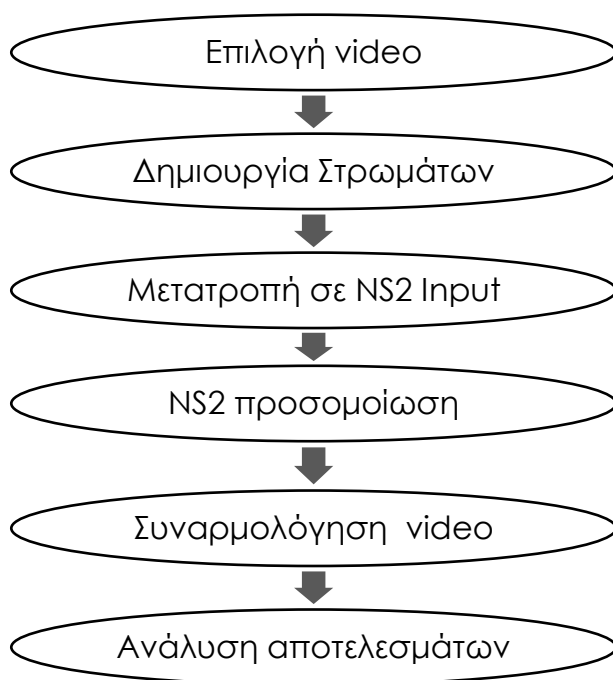
Καινούρια σουίτα ADIVIS

- 4.1 Διαδικασία εξαγωγής αποτελεσμάτων
 - 4.1.1 Εισαγωγή
 - 4.1.2 Επιλογή Βίντεο
 - 4.1.3 Δημιουργία Στρωμάτων & Μετατροπή σε NS2 input
 - 4.1.4 Προσομοίωση σε NS2
 - 4.1.5 Συναρμολόγηση βίντεο
 - 4.1.6 Ανάλυση αποτελεσμάτων
 - 4.2 Δομή Αρχαιοθέτησης
 - 4.2.1 Εισαγωγή
 - 4.2.2 Παλαιά δομή αρχειοθέτησης
 - 4.2.3 Καινούρια δομή αρχειοθέτησης
 - 4.2.4 Διαχείριση κατάλογων
 - 4.2.5 Συμβατότητα με διαφορετικές εκδόσεις NS2
 - 4.3 Δημιουργία στρωμάτων & Συναρμολόγηση βίντεο
 - 4.3.1 Εισαγωγή
 - 4.3.2 Παλαιά διαδικασία επεξεργασίας βίντεο
 - 4.3.3 Καινούρια διαδικασία επεξεργασίας βίντεο
 - 4.3.4 Αποτελέσματα με τους 2 διαφορετικούς αλγόριθμους
 - 4.4 Διορθώσεις/Βελτιώσεις συστήματος ADIVIS
-

4.1 Διαδικασία εξαγωγής αποτελεσμάτων

4.1.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα περιγράψω τα βήματα που χρειάζεται να ακολουθήσει κάποιος για να βγάλει τα αποτελέσματά που θέλει με τη σουίτα ADIVIS (δεν θα αναφερθούν συγκεκριμένες λεπτομέρειες υλοποίησης σε αυτό το κεφάλαιο καθώς αναφέρονται στα επόμενα κεφάλαια) . Η διαδικασία φαίνεται στο σχήμα 3.1



Σχήμα 4.1 Διαδικασία εξαγωγής αποτελεσμάτων

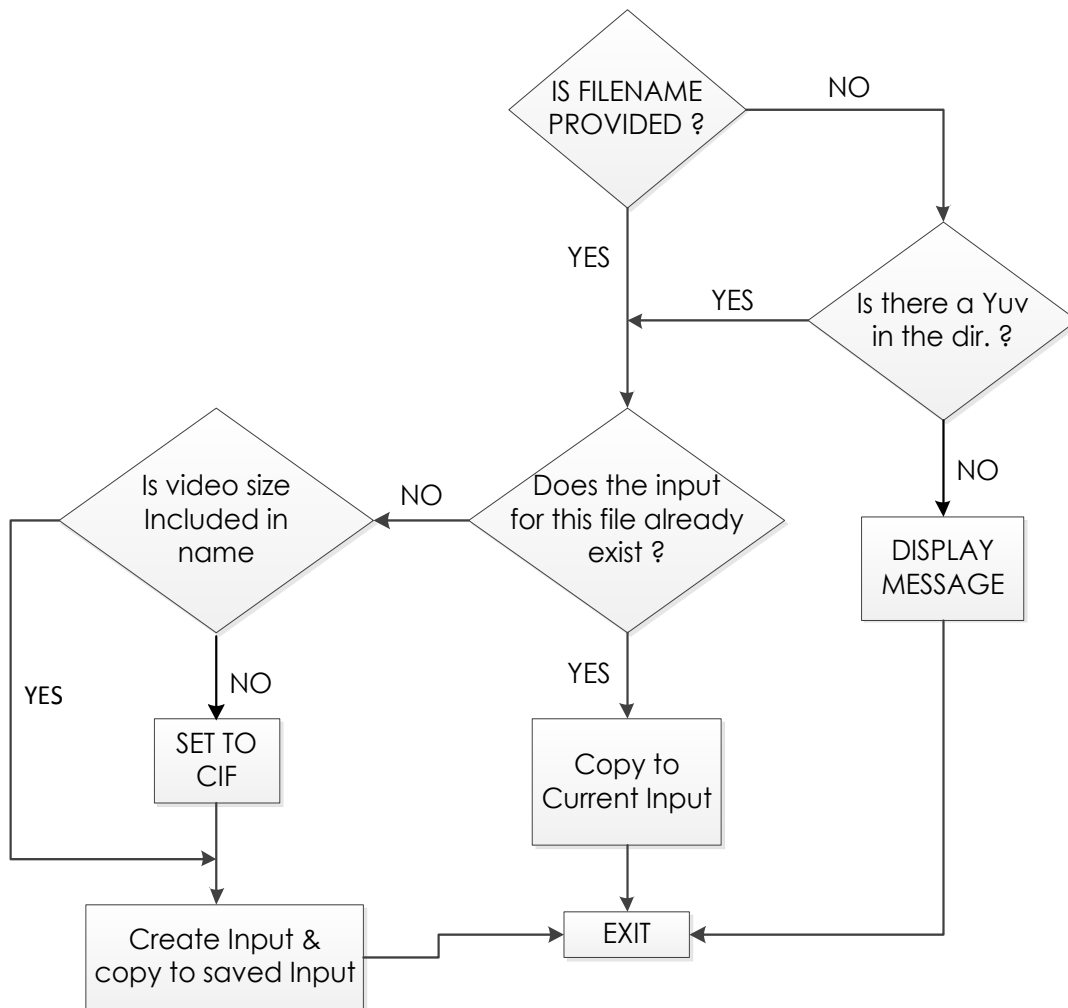
4.1.2 Επιλογή Βίντεο

Αρχικά προσπαθούσα να κωδικοποιήσω κανονικά βίντεο και να τα δώσω ως είσοδο στον NS2. Ανακάλυψα όμως ότι τα βίντεο που μπορούν να χρησιμοποιηθούν είναι ειδικά κατασκευασμένα για αυτό τον σκοπό. Υπάρχουν αρκετές ιστοσελίδες στις οποίες μπορεί να βρει κάποιος trace videos. Για το σκοπό της διπλωματικής αυτής προμηθεύτηκα κάποια

βίντεο από το [3]. Αξίζει να σημειωθεί ότι τα resolution που υποστηρίζονται προς το παρόν είναι “qcif” (176x144) και “cif” (352x288).

4.1.3 Δημιουργία Στρωμάτων & Μετατροπή σε NS2 input

Αφού επιλεγθεί και είναι διαθέσιμο το αρχείο βίντεο χρειάζεται να δημιουργήσουμε 8 κωδικοποιημένα στρώματα. Ο λόγος δημιουργίας τους είναι η ανάγκη εναλλαγής μεταξύ των στρωμάτων όταν αλλάζει το διαθέσιμο εύρος ζώνης ώστε να χάνονται λιγότερα πακέτα και να λαμβάνει καλύτερη ποιότητα εικόνας ο παραλήπτης. Το καθένα από τα 8 στρώματα είναι κωδικοποιημένα σε διαφορετικό bitrate και συγκεκριμένα χρησιμοποιούνται τα εξής bitrates: 64/96/128/192/256/384/512/768 Kbps. Για να δημιουργηθούν τα 8 κωδικοποιημένα στρώματα πρέπει να τοποθετηθεί στον φάκελο currentInput ή να δοθεί το μονοπάτι του ως παράμετρος στο script input.tcl. Δηλαδή εάν δεν δοθεί παράμετρος το script input.tcl θα ψάξει και θα φορτώσει το πρώτο αρχείο που θα βρει με κατάληξη yuv στον φάκελο currentInput. Ο χρήστης πρέπει να έχει υπόψη ότι το script έχει προκαθορισμένο το resolution cif εκτός και αν υπάρχει η λέξη “qcif” στο όνομα του αρχείου βίντεο. Σε τέτοια περίπτωση το resolution καθορίζεται σε qcif, οπότε είναι σημαντικό ο χρήστης να συμπεριλάβει τη λέξη qcif στο όνομα του βίντεο εάν επιθυμεί να χρησιμοποιήσει αυτό το resolution. Αφού τρέξει το input.tcl script δημιουργούνται τα στρώματα και όλα τα απαραίτητα αρχεία εισόδου που θα χρειαστεί ο NS2 στον φάκελο currentInput και μεταφέρονται επίσης στον φάκελο savedInput σε περίπτωση που χρειαστούν ξανά μελλοντικά. Με αυτό τον τρόπο αποφεύγεται η επαναδημιουργία των στρωμάτων ή οποία είναι αρκετά χρονοβόρα ειδικά για αρχεία με resolution “cif”. Το μόνο που έχει να κάνει ο χρήστης αν χρειαστεί να χρησιμοποιήσει το βίντεο μελλοντικά είναι να μεταφέρει τα περιεχόμενα του φάκελου στον οποίο αναγράφεται το όνομα του βίντεο που θέλει μέσα στον φάκελο currentInput. Οι πιο πάνω οδηγίες φαίνονται πιο ξεκάθαρα στο σχήμα 4.2.



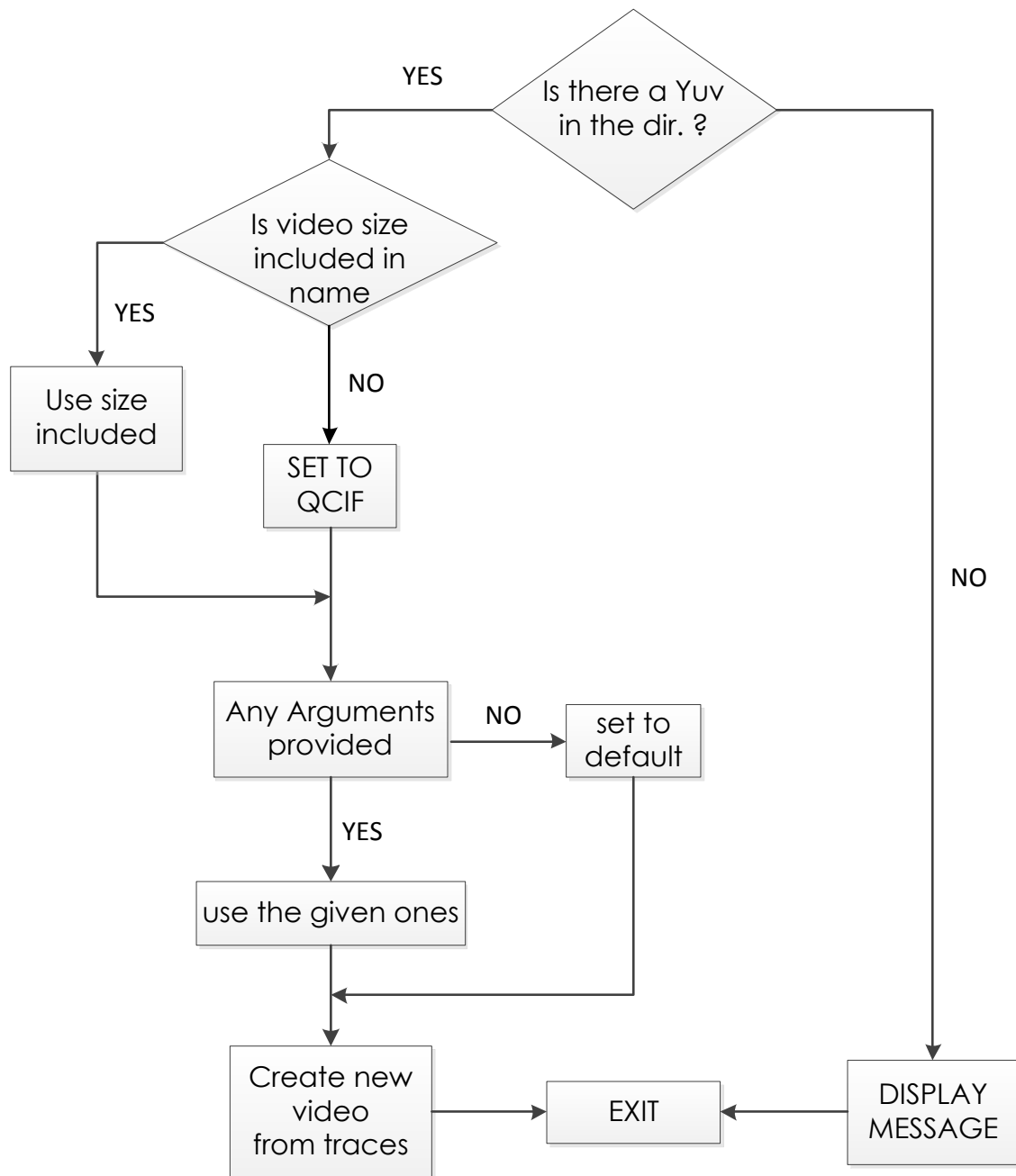
Σχήμα 4.2 Λογικό διάγραμμα αρχείου input.tcl

4.1.4 Προσομοίωση σε NS2

Ο χρήστης μπορεί να προσομοιώσει ένα από τα υποστηριζόμενα περιβάλλοντα που περιγράφονται στο Κεφάλαιο 6 μέσω των αντίστοιχων script που βρίσκονται στον ριζικό φάκελο του ADIVIS τα οποία περνούν ως παράμετρο το σενάριο που θα χρησιμοποιηθεί. Σημαντικό είναι να ακολουθείται το πρότυπο των σεναρίων που ήδη υπάρχουν για κάθε περιβάλλον.

4.1.5 Συναρμολόγηση βίντεο

Αφού γίνει η προσομοίωση, ο χρήστης μπορεί να τρέξει το script “output.tcl” το οποίο παίρνει ως προαιρετικές παραμέτρους τον αριθμό του παραλήπτη και τον αριθμό του αποστολέα. Εάν δεν δοθούν οι παράμετροι αυτοί τότε θα χρησιμοποιηθούν οι προκαθορισμένες τιμές που είναι ο πρώτος παραλήπτης καθώς και ο πρώτος αποστολέας ακολουθώντας την σειρά που ορίστηκαν στην προσομοίωση. Οι πιο πάνω οδηγίες φαίνονται πιο ξεκάθαρα στο σχήμα 4.3. Μόλις τρέξει ο χρήστης το script θα δει στην οθόνη του το συναρμολογημένο βίντεο που πήρε ο προκαθορισμένος παραλήπτης. Για να συγκρίνουμε το βίντεο αυτό με το αρχικό που έστειλε ο αποστολέας γίνεται μέτρηση του PSNR (Peak Signal-to-Noise Ratio). Η μετρική Peak Signal-to-Noise Ratio εκφράζει την αναλογία μεταξύ της μέγιστης ισχύος ενός σήματος και την ισχύ του θορύβου που επηρεάζει την ακρίβεια της αναπαράστασής του. Στην περίπτωση μας το σήμα είναι το αρχικό βίντεο και ο θόρυβος είναι το λάθος που εισάχθηκε από τις απώλειες πακέτων και την κωδικοποίηση του αρχικού βίντεο. Τα αποτελέσματα καταχωρούνται αυτόματα στον φάκελο currentOutput όπως και το βίντεο που παράχθηκε.



Σχήμα 4.3 Λογικό διάγραμμα αρχείου output.tcl

4.1.6 Ανάλυση αποτελεσμάτων

Για ανάλυση των αποτελεσμάτων έχω γράψει κάποια perl scripts για αρίθμηση των αποτελεσμάτων του PSNR στα αντίστοιχα πλαίσια, μέτρηση των χαμένων πακέτων καθώς και μέτρηση των χαμένων πλαισίων για τον κάθε παραλήπτη . Τα scripts αυτά μαζί με άλλες εφαρμογές που υπάρχουν για δημιουργία γραφικών παραστάσεων (π.χ. gnuplot) και περεταίρω ανάλυση των simulation trace files μπορούν να βρεθούν στον φάκελο software της σουίτας ADIVIS.

4.2 Δομή Αρχαιοθέτησης

4.2.1 Εισαγωγή

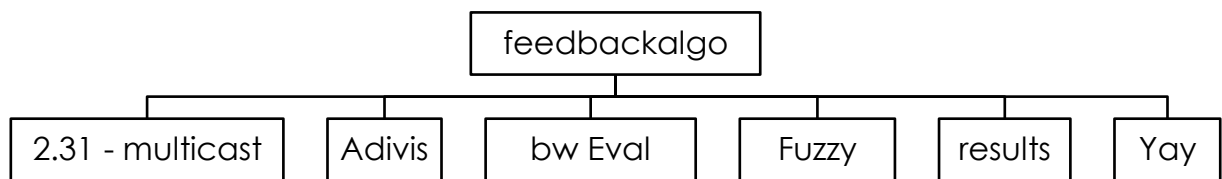
Όταν παρέλαβα την υλοποίηση του ADIVIS, δεν υπήρχε κάποια συγκεκριμένη δομή αρχειοθέτησης ούτε μπορούσε να χαρακτηριστεί ως “σουίτα”. Συγκεκριμένα ο φάκελος feedbackalgo που είναι ο ριζικός φάκελος του συστήματος συμπεριλάμβανε όλα τα αρχεία, χωρίς καμιά βασική οργάνωση. Υπήρχαν πολλά αρχεία που δεν εξυπηρετούσαν κανένα σκοπό καθώς ήταν αποτελέσματα παλαιών προσομοιώσεων ή που ήταν παλαιότερες εκδόσεις άλλων αρχείων χωρίς να υπάρχει όμως ένδειξη πιο αρχείο ήταν το πιο πρόσφατα ενημερωμένο. Επίσης κάποια από τα αρχεία λειτουργούσαν αλλά ήταν λάθος η εφαρμογή τους. Τέλος λόγω της ελλειπείας τεκμηρίωσης, δεν ήταν ξεκάθαρος ο τρόπος λειτουργίας ή ακόμη και ο σκοπός αρκετών αρχείων. Οι παράγοντες αυτοί επιβράδυναν αρκετά την κυρίως εργασία που μου είχε ανατεθεί, καθώς οι ασάφειες που υπήρχαν οδήγησαν πολλές φορές σε λάθη λόγω εσφαλμένων υποθέσεων και σε σπατάλη περισσότερου χρόνου όταν έπρεπε να ανατρέξω σε κάποιο αρχείο.

Στην πορεία της διπλωματικής μου αποφάσισα ότι ήταν σημαντικό και για μελλοντική χρήση του ADIVIS να υπάρχει μια καλή οργάνωση των φακέλων και αρχείων που το αποτελούν, έτσι ώστε ο επόμενος που θα το χρησιμοποιήσει να μπορεί να αφιερώσει περισσότερο χρόνο στο ερευνητικό κομμάτι καθώς και να έχει μια πιο ξεκάθαρη εικόνα για το ρόλο του κάθε αρχείου στο σύστημα. Θεώρησα αναγκαίο να συμπεριλάβω επίσης

κάποια ανάλυση της παλαιάς δομής για να διευκολυνθεί ο οποιοσδήποτε θελήσει να ανατρέξει σε προηγούμενες εκδόσεις του συστήματος ADIVIS.

4.2.2 Παλαιά δομή αρχειοθέτησης

Η προηγούμενη δομή που υπήρχε στον ριζικό φάκελο του συστήματος φαίνεται στο ακόλουθο σχήμα:



Σχήμα 4.4 Παλαιά δομή του ριζικού φάκελου

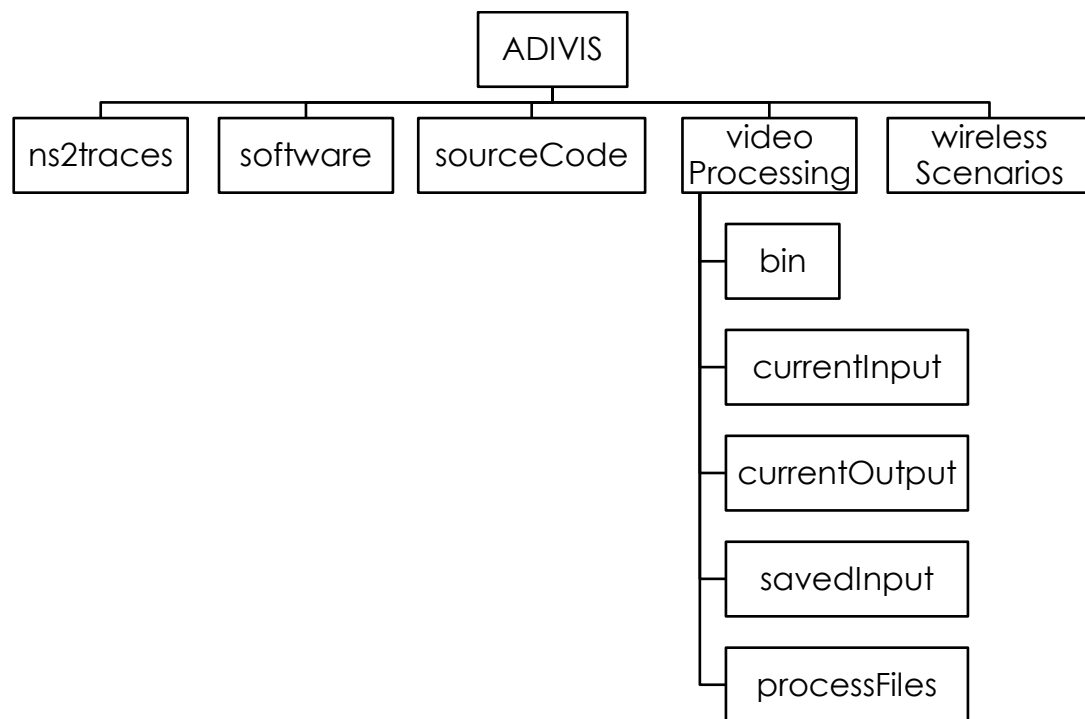
- Feedbackalgo: Παρατηρούμε ότι ο ριζικός φάκελος ονομάζεται feedbackalgo και παρόλο που υπάρχει κάποια λογική πίσω από το όνομα αυτό, κάποιος που ανοίγει τον φάκελο του NS2 δεν θα αναγνωρίσει αυτόματα ότι στον φάκελο αυτό βρίσκεται η σουίτα του ADIVIS. Όταν ανοίξει κάποιος τον φάκελο αυτό, εκτός από τους φακέλους που φαίνονται στο σχήμα θα βρει επίσης πολυάριθμα αρχεία (113 συνολικά) όπως αποτελέσματα προηγούμενων προσομοιώσεων, ns2 traces, διάφορα scripts προσομοιώσεων, εφαρμογές επεξεργασίας βίντεο, 8 στρώματα βίντεο σε κωδικοποιημένη μορφή και σε NS2 input μορφή.
- 2.31-multicast: Περιέχει μόνο πεπαλαιωμένα αρχεία πηγαίου κώδικα τα οποία αποπροσανατολίζουν κάποιον που δεν είναι ενήμερος. Επίσης αυτός ο φάκελος υπάρχει ακόμη και στις εκδόσεις της σουίτας του ADIVIS όπου δεν υποστηρίζεται το multicasting.
- Adivis: Περιέχει αποτελέσματα προηγούμενων προσομοιώσεων, 8 κωδικοποιημένα στρώματα σε μορφή βίντεο, traces προηγούμενων προσομοιώσεων (χωρίς όμως να φαίνεται από ποια προσομοίωση εξάχθηκαν) και διαφορετικά simulation scripts.
- BwEval: Είναι πεπαλαιωμένος φάκελος καθώς δεν εξυπηρετεί κανένα σκοπό στην

τρέχουσα υλοποίηση του συστήματος.

- Fuzzy: Περιέχει την εφαρμογή του αλγόριθμου Fuzzy.
- Results: Περιέχει λιγοστά αποτελέσματα προσομοιώσεων χωρίς όμως να υπάρχει σύνδεση από ποια προσομοίωση προήλθαν.
- Yay: Περιέχει μια εφαρμογή για YUV βίντεο playback και μια από τις εφαρμογές που χρειάζεται για την συναρμολόγηση του βίντεο στην μεριά του παραλήπτη. Επίσης συμπεριλαμβάνει τους αντίστοιχους πηγαίους κώδικες.

4.2.3 Καινούρια δομή αρχειοθέτησης

Η καινούρια δομή αρχειοθέτησης που δημιούργησα βασίζεται στις λειτουργίες που πρέπει να εκτελέσει κάποιος για να βγάλει αποτελέσματα με τον ADIVIS καθώς και στο ρόλο του κάθε αρχείου.



Σχήμα 4.5 Καινούρια δομή του ριζικού φάκελου

- ADIVIS: Παρατηρούμε ότι ο ριζικός φάκελος μετονομάστηκε σε ADIVIS για πιο γρήγορη εντόπιση του στον ριζικό φάκελο του NS2. Όταν ανοίξει κάποιος τον φάκελο αυτό, εκτός από τους φακέλους που φαίνονται στο σχήμα θα βρει κάποια scripts προσομοίωσης (τα οποία έχουν κατατοπιστικά ονόματα και εμπεριέχουν μια βασική τεκμηρίωση) καθώς και ένα script που βοηθά στην ανάλυση των traces που παράγει ο NS2.
- Ns2traces: Περιέχει όλα τα traces που έχουν παραχθεί από τον NS2 κατά τη διάρκεια της τελευταίας προσομοίωσης που εκτελέστηκε.
- Software: Δεν αποτελεί μέρος της υλοποίησης του συστήματος αλλά περιέχει διάφορα αρχεία που είναι σχετικά με εργασία πάνω στον ADIVIS. Συγκεκριμένα:
 - Ολοκληρωμένα πακέτα εφαρμογών που ήδη συμπεριλαμβάνονται στην υλοποίηση (χωρίς τον πηγαίο τους κώδικα)
 - Patches που χρησιμοποιήθηκαν (ή που μπορεί να εφαρμοστούν αργότερα)
 - Εφαρμογές που είναι βοηθητικές για σκοπούς αποσφαλμάτωσης,
 - Παλαιότερες εκδόσεις εφαρμογών που χρησιμοποιούνται στην υλοποίηση
- SourceCode: Περιέχει τους πηγαίους κώδικες καθώς και τα αντικείμενα που δημιουργούνται από αυτούς τα οποία χρειάζονται για τη λειτουργία του συστήματος. Αυτά αποτελούν την κυρίως υλοποίηση του συστήματος ADIVIS.
- VideoProcessing: Εκτός από τους φακέλους που φαίνονται στο σχήμα, περιέχει κάποια scripts που παράγουν το input που χρειάζεται ο NS2 και το output από τα traces του NS2 (το βίντεο που βλέπει ο προκαθορισμένος παραλήπτης). Οι φάκελοι που έχει από κάτω του:
 - Bin: Περιέχει τα executables που αφορούν την επεξεργασία βίντεο
 - CurrentInput: Το τρέχον input. Αυτό που θα διαβάσει ο NS2 για να βγάλει αποτελέσματα όταν τρέξουμε μια προσομοίωση
 - CurrentOutput: Το τελευταίο output που παράχθηκε. Συμπεριλαμβάνει όλα τα αρχεία που αφορούν την εξαγωγή αποτελεσμάτων και δημιουργία γραφικών παραστάσεων.
 - ProcessFiles: Έχει φυλαγμένα τα “ενδιάμεσα” αρχεία που δημιουργήθηκαν την τελευταία φορά που παράχθηκε input ή output. Τα αρχεία αυτά υπάρχουν μόνο για σκοπούς αποσφαλμάτωσης της επεξεργασίας βίντεο.

- SavedInput: Περιέχει έτοιμα αρχεία input για τον NS2 (περισσότερες πληροφορίες στο υποκεφάλαιο 3.1.3)
- WirelessScenarios: Περιέχει κάποια σενάρια που δημιούργησα για προσομοιώσεις που αφορούν ασύρματα δίκτυα.

4.2.4 Διαχείριση καταλόγων

Για να είναι συμβατή η σουίτα ADIVIS με τυχόν αλλαγές που χρειαστεί να γίνουν στο μέλλον στη δομή αρχειοθέτησης, δημιούργησα ένα κεντρικό script που εξυπηρετεί αποκλειστικά τη διαχείριση των φακέλων της σουίτας. Επίσης αυτό επιτρέπει τον χρήστη να έχει μια δομή αρχειοθέτησης όπως την θέλει αυτός, καθώς μπορεί να αποφασίσει ότι εξυπηρετεί καλύτερα τον σκοπό για τον οποίο θα χρησιμοποιήσει την σουίτα του ADIVIS. Συγκεκριμένα εάν ο χρήστης επιθυμεί να μετονομάσει τον φάκελο ns2traces σε traces και να τον τοποθετήσει μέσα στον φάκελο videoProcessing το μόνο που έχει να κάνει είναι να ανοίξει το αρχείο *FileManagement.tcl* και να αλλάξει την μεταβλητή που του αντιστοιχεί. Ακολουθεί ο κώδικας που περιγράφει την συγκεκριμένη εντολή που πρέπει να αλλάξει ο χρήστης:

Από ns2traces :

```
# Dir that stores all the traces generated by NS during simulation
set traceDir [set a $currDir][set b ns2traces/]
```

σε traces και σε διαφορετικό φάκελο:

```
# Dir that stores all the traces generated by NS during simulation
set traceDir [set a $currDir][set b videoProcessing/traces/]
```

Όλα τα tcl αρχεία που υπάρχουν στη δομή του ADIVIS καλούν το αρχείο αυτό στην αρχή τους μέσω της εντολής *source AdvisRootPath/FolderManagement.tcl*. Με τον τρόπο αυτό γνωρίζουν το μονοπάτι του κάθε φακέλου. Όπως φαίνεται στην εντολή που αναφέρθηκε, το *FolderManagement.tcl* πρέπει να βρίσκεται πάντα στον ριζικό φάκελο του ADIVIS. Εάν επιθυμεί κάποιος να αλλάξει την τοποθεσία του μπορεί να το πράξει εφόσον όμως αλλάξει

την εντολή που προαναφέρθηκε σε όλα τα αρχεία TCL από *source AdivisRootPath/FolderManagement.tcl* σε *source NewPathOfFileManagement /FolderManagement.tcl*. Οι οδηγίες αυτές εμπεριέχονται πιο συνοπτικά και στο ίδιο το αρχείο.

4.2.5 Συμβατότητα με διαφορετικές εκδόσεις NS2

Εάν κάποιος επιθυμεί να εγκαταστήσει την σουίτα ADIVIS σε προηγούμενες εκδόσεις του NS2 ή σε πιο καινούριες μπορεί να το πράξει εύκολα. Το μόνο που έχει να κάνει είναι να τοποθετήσει τον φάκελο ADIVIS στον ριζικό φάκελο, να κάνει overwrite το αρχείο *session-rtp.tcl* (στον φάκελο */ns-2.xx/tcl/rtp/*) με το αντίστοιχο *session-rtp.tcl* που βρίσκεται στον φάκελο *software* της σουίτας ADIVIS και να κάνει αλλαγές στα αρχεία που θα περιγράψω πιο κάτω. (το μονοπάτι αναφοράς τους είναι ο ριζικός φάκελος του NS2)

- *Makefile.in*: Αυτό το αρχείο καλύτερα να ανοιχτεί με κάποιο editor όπως τον Emacs για να διατηρηθούν σωστά τα κενά που χρειάζονται για να εκτελεστεί. Εκεί που βρίσκεται η γραμμή “*OBJ_CC = *” να προστεθούν από κάτω οι γραμμές

- *ADIVIS/sourceCode/Yuv.o *
- *ADIVIS/sourceCode/videotraffictrace.o *
- *ADIVIS/sourceCode/video_rtcp.o *
- *ADIVIS/sourceCode/session_rtp.o *
- *ADIVIS/sourceCode/video_rtp.o *

Κάτω από την ίδια γραμμή να εντοπιστεί και να μπει σε σχόλια (ή να αφαιρεθεί τελείως αν δεν πρόκειται να γίνουν προσομοιώσεις με RTP χωρίς τον ADIVIS) η γραμμή

- *common/session-rtp.o apps/rtp.o tcp/rtcp.o *

Εκεί που βρίσκεται η γραμμή “*OBJ_C = *” να προστεθεί η γραμμή

- *ADIVIS/sourceCode/video_fuzzyIe.o*

- *tcl/lib/ns-source.tcl*: Να εντοπιστεί και να μπει σε σχόλια (ή να αφαιρεθεί τελείως αν δεν πρόκειται να γίνουν προσομοιώσεις με RTP χωρίς τον ADIVIS) η γραμμή

- *Class Agent/CBR/RTP -superclass Agent/RTP*

και να προστεθεί η γραμμή

- Class Agent/CBR/RTP -superclass Agent/VideoRTP
- tcl/lib/ns-packet.tcl: Να προστεθεί από κάτω από τη γραμμή “# Transport Protocols and related protocols” (που βρίσκεται μέσα στο loop “foreach prot {“) η γραμμή
 - RTCP # RTCP
- common/Agent.h: Να προστεθούν κάτω από τη γραμμή “class Agent : public Connector {” κάτω από public, οι συναρτήσεις
 - void set_frametype(int type) { frametype_ = type; }
 - void set_frameseq(int seq) { frameseq_ = seq; }κάτω από protected, οι μεταβλητές:
 - int frametype_; // frame type for MPEG video transmission
 - int frameseq_; // frame sequence for MPEG video transmission
- common/Packet.h: Να προστεθούν κάτω από τη γραμμή “struct hdr_cmh {” οι μεταβλητές
 - double sendtime_; // send time
 - bool video_rtcp; // Flag for rtcp packets
 - u_int32_t srcid_; // Sender node id
 - int seqno_; // Packet id
- common/Agent.cc: Να προστεθούν δίπλα από τη γραμμή “Agent::Agent(packet_t pkttype) :” οι αρχικοποιήσεις
 - frametype_(0)
 - frameseq_(0)Να προστεθούν κάτω από τον ορισμό της συνάρτησης “Agent::initpkt(Packet* p) const” οι γραμμές
 - ch->video_rtcp = false;
 - ch->srcid_ = 0;
 - ch->seqno_ = 0;
 - ch->sendtime = 0;
- tcl/lib/ns-default.tcl: Να προστεθούν στο τέλος του αρχείου οι ακόλουθες γραμμές (έχει τοποθετηθεί αρχείο που τις περιέχει στον φάκελο software)
 - #Values used in scenarios that do not involve

- #real video traces
- Agent/VideoRTP set max_scale_ 8
- Agent/VideoRTP set frm_size_0_ 8000
- Agent/VideoRTP set frm_size_1_ 1200
- Agent/VideoRTP set frm_size_2_ 8000
- Agent/VideoRTP set frm_size_3_ 1200
- Agent/VideoRTP set frm_size_4_ 8000
- Agent/VideoRTP set frm_size_5_ 1200
- Agent/VideoRTP set frm_size_6_ 8000
- Agent/VideoRTP set frm_size_7_ 1200
- Agent/VideoRTP set frm_interval_0_ 1.00000000
- Agent/VideoRTP set frm_interval_1_ 0.10000000
- Agent/VideoRTP set frm_interval_2_ 0.50000000
- Agent/VideoRTP set frm_interval_3_ 0.05000000
- Agent/VideoRTP set frm_interval_4_ 0.25000000
- Agent/VideoRTP set frm_interval_5_ 0.02500000
- Agent/VideoRTP set frm_interval_6_ 0.12500000
- Agent/VideoRTP set frm_interval_7_ 0.01250000
- #RTP Session
- Session/RTP set factorC_ 1.0
- Session/RTP set wireless_ 0
- Session/RTP set debug_ 0
- Session/RTP set bytes_ 0
- Session/RTP set recvid_ 0
- for {set i 0} {\$i < 100} {incr i} { Session/RTP set loss_\$i 0.0 }
- # RTP Agent
- Agent/VideoRTP set seqno_ 0
- Agent/VideoRTP set interval_ [Agent/VideoRTP set frm_interval_0_]
- Agent/VideoRTP set random_ 0
- # packetSize includes all headers i.e 1000 + 28(IP+UDP HEADERS)

- Agent/VideoRTP set packetSize_ 1028
- Agent/VideoRTP set maxpkts_ 0x10000000
- Agent/VideoRTP set user_bw_ 0
- Agent/VideoRTP set random_ true
- Agent/VideoRTP set fragment_size_ 1000
- # Procedure that is called when RTP agent
- # finishes sending packets.
- Agent/VideoRTP instproc done {} { }
- # RTCP Agent
- Agent/VideoRTCP set seqno_ 0
- Agent/VideoRTCP set packetSize_ 100
- Agent/VideoRTCP set interval_ 0.5
- Agent/VideoRTCP set random_ 0
- Agent/VideoRTCP set class_ 32
- Agent/VideoRTCP set quality_ 2
- # Evaluation Agent
- Agent/YuvObject set videoWidth 176
- Agent/YuvObject set videoHeight 144
- Agent/YuvObject set line 0
- # Small delay between packets to avoid collisions
- Application/Traffic/videoTrace set delay_ 0.001
- # Needed to initialize these Tracefile variables
- Tracefile set debug_ false
- Tracefile set randomize_ false

Πρέπει να σημειωθεί ότι με αυτές τις αλλαγές θα δουλέψει η σουίτα του ADIVIS στις πλείστες υπάρχουσες και μελλοντικές εκδόσεις του NS2 χωρίς να λειτουργούν όμως τα scripts προσομοίωσης που χρησιμοποιούν άλλες προσθήκες του NS2 (όπως SSM και PUMA). Για να λειτουργήσουν και αυτά θα πρέπει ο χρήστης από μόνος του να εξετάσει ποιες εκδόσεις του NS2 υποστηρίζουν οι προσθήκες που επιθυμεί να χρησιμοποιήσει από

την ιστοσελίδα τους ή από την τεκμηρίωση που θα βρει. Για κάποιες προσθήκες θα υπάρχει η σχετική τεκμηρίωση στον φάκελο software της σουίτας ADIVIS.

4.3 Δημιουργία στρωμάτων & Συναρμολόγηση βίντεο

4.3.1 Εισαγωγή

Όταν δοκίμασα να βγάλω τα πρώτα αποτελέσματά σε σενάρια όπου υπήρχαν αρκετές απώλειες πακέτων αντιμετώπισα αρκετά προβλήματα με τα εργαλεία που υπήρχαν για συναρμολόγηση του βίντεο που λάμβανε ο παραλήπτης. Αυτά τα προβλήματα προέκυψαν επειδή προηγουμένως δεν είχαν δοκιμαστεί κάτω από αντίστοιχες συνθήκες οι εφαρμογές αυτές. Θεώρησα χρήσιμο να γίνει αναφορά πώς γινόταν η επεξεργασία βίντεο προηγουμένως και πώς γίνεται τώρα.

4.3.2 Παλαιά διαδικασία επεξεργασίας βίντεο

Η διαδικασία για δημιουργία των δεδομένων εισόδου του NS2 από βίντεο ήταν ανύπαρκτη. Για τη συναρμολόγηση του βίντεο υπήρχε ένα script με την ονομασία “doIt” το οποίο έπαιρνε ως είσοδο τον αριθμό του παραλήπτη και τον αριθμό του αποστολέα. Μέσα από αυτό το script αρχικά καλείτο το εκτελέσιμο αρχείο makeYUV (του οποίου έλειπε ο πηγαίος κώδικας) που έπαιρνε ως παραμέτρους το tracefile του παραλήπτη και το αρχείο psnrinfo το οποίο κρατάει τα πλαίσια στα οποία ο ADIVIS πρόσθεσε ή αφαίρεσε στρώμα κωδικοποίησης. Το αρχείο αυτό εβγάξε μήνυμα λάθους σχετικό με τη μνήμη όποτε υπήρχε ζυγός αριθμός προσθαφαιρέσεων στρωμάτων στο αρχείο psnrinfo. Ακολούθως ζήτησα τον πηγαίο κώδικα από το δημιουργό του ο οποίος ανταποκρίθηκε πολύ γρήγορα στέλνοντας μου τον κώδικα. Λύνοντας το bug που προκαλούσε το λάθος που ανέφερα πριν (ήταν μια ανάθεση λάθος), ανέλυσα προσεκτικά τον αλγόριθμο που υπήρχε στη δημιουργία ζευγαρίων tracefile.

Συγκεκριμένα ο σκοπός του κώδικα makeYUV.c ήταν να δημιουργήσει ένα ζευγάρι sender-receiver trace file για κάθε κομμάτι στρώματος που χρησιμοποιήθηκε. Δηλαδή εαν από το

πλαίσιο 143 έως το πλαίσιο 195 χρησιμοποιήθηκε το 1ο στρώμα τότε αντιστοιχεί σε αυτό το κομμάτι ένα sender trace που στέλνει ολόκληρο το 1ο στρώμα (ουσιαστικά ένα sender trace που στέλνει το βίντεο κωδικοποιημένο μόνο στο bitrate που αντιστοιχεί στο στρώμα) και ένα receiver trace που λαμβάνει μέχρι το πλαίσιο 143 ακριβώς τα ίδια πακέτα που στέλνει και το καινούριο sender trace μέχρι το πλαίσιο 143. Από το πλαίσιο 143 μέχρι το 195 το καινούριο receiver trace πρέπει να παραλάβει τα πακέτα που παραλήφθηκαν και από τον παραλήπτη κατά τη διάρκεια της προσομοίωσης. Έπειτα, από το πλαίσιο 195 μέχρι το τέλος του στρώματος το καινούριο receiver trace λαμβάνει τα ίδια πακέτα που έστειλε και το καινούριο sender trace.

Ο προβληματικός αλγόριθμος δεν άρχιζε από το πλαίσιο 143 να τοποθετεί τα αντίστοιχα πακέτα του πρώτου στρώματος στο νέο receiver trace αλλά ξεκινούσε από τον αριθμό πακέτων που είχε στείλει μέχρι το πλαίσιο 143 ο αποστολέας κατά τη διάρκεια της προσομοίωσης. Αυτό υποθέτει όμως ότι για όλα τα στρώματα αντιστοιχεί ο ίδιος αριθμός πακέτων. Αυτό όμως δεν ισχύει. Παρόλο που κάθε στρώμα έχει τον ίδιο αριθμό πλαισίων, επειδή είναι κωδικοποιημένα σε διαφορετικά bitrate το μέγεθος του αντίστοιχου πλαισίου σε κάθε στρώμα είναι διαφορετικό. Καθώς το κάθε πλαίσιο σπάει σε πακέτα που έχουν μέγιστο μέγεθος 1000 Bytes αυτό συνεπάγεται ότι ο αριθμός πακέτων που αντιστοιχεί στο πλαίσιο X είναι διαφορετικός από τον αριθμό πακέτων που αντιστοιχεί στο ίδιο πλαίσιο αλλά σε διαφορετικό στρώμα. Έτσι φθάνουμε στο συμπέρασμα ότι ο αριθμός πακέτων που έστειλε ο αποστολέας μέχρι το πλαίσιο 143 δεν είναι ο ίδιος με τον αριθμό πακέτων που θα πάρουμε από το 1ο στρώμα μέχρι το πλαίσιο 143. Είναι το σύνολο των πακέτων που στάληκαν μέχρι το πλαίσιο 143 αλλά από διαφορετικά στρώματα. Το πρόβλημα αυτό είναι πιο αισθητό σε βίντεο με πιο ψηλό resolution (π.χ cif) επειδή εκεί η διαφορά στον αριθμό πακέτων κάθε στρώματος είναι αρκετά μεγαλύτερη. Αξιοσημείωτο είναι το γεγονός ότι η μορφή του κώδικα στο αρχείο δεν ήταν καθόλου ελαστική αφού δημιουργήθηκε μόνο για να τις ανάγκες ενός συγκεκριμένου βίντεο, με συγκεκριμένο resolution, συγκεκριμένο αριθμό αλλαγών στρωμάτων.

Στη συνέχεια έπαιρνε το κάθε ζευγάρι sender-receiver trace που δημιουργήθηκε με τη διαδικασία που περιέγραψα πιο πάνω και το έδινε σαν δεδομένα εισόδου μαζί με το στρώμα

που του αντιστοιχούσε (το στρώμα σε μορφή εισόδου για NS2 και σε μορφή βίντεο) στο εκτελέσιμο αρχείο “et”. Το αρχείο αυτό είναι μέρος της σουίτας Evalvid την οποία μπορεί να βρει κάποιος στο [5]. Με τα δεδομένα εισόδου που ανάφερα παράγει το βίντεο που λαμβάνει ο παραλήπτης του ζευγαριού που δώσαμε. Από το βίντεο αυτό μας ενδιαφέρουν μόνο τα πλαίσια 143 μέχρι 195 με βάση το παράδειγμα που έδωσα. Ακολουθώντας το βίντεο αυτό αποκωδικοποιούνται σε μορφή yuv με χρήση του FFMPEG. Στο τέλος υπήρχε ο ίδιος αριθμός βίντεο με τον ίδιο αριθμό κομματιών από στρώματα που χρησιμοποιήθηκαν.

Επείτα έδινε όλα τα προηγούμενο βίντεο μαζί με το αρχείο psnrginfo (που κρατά σε πύλο πλαίσιο γίνεται η κάθε αλλαγή στρώματος) ως δεδομένο εισόδου στο εκτελέσιμο αρχείο “concatenate”. Το αρχείο αυτό για κάθε κομμάτι στρώματος που χρησιμοποιήθηκε (πληροφορία που εξάγεται από το αρχείο psnrginfo) διάβαζε από το αντίστοιχο βίντεο τό ίδιο κομμάτι (δηλαδή για το βίντεο που δημιουργήθηκε για το κομμάτι 143 – 195 θα διαβαστούν μόνο αυτά τα πλαίσια). Αφού ένωνε όλα τα πλαίσια δημιουργούσε το βίντεο που έπαιρνε ο παραλήπτης κατά τη διάρκεια της προσομοίωσης. Αξιοσημείωτο ότι και αυτό το αρχείο υποστήριζε μόνο τις ανάγκες ενός συγκεκριμένου βίντεο, με συγκεκριμένο resolution και συγκεκριμένο αριθμό αλλαγών στρωμάτων. Επίσης υπήρχε λάθος στον κώδικα που οδηγούσε στο να επιλεγθούν λανθασμένα κομμάτια από κάθε βίντεο.

4.3.3 Καινούρια διαδικασία επεξεργασίας βίντεο

Έγιναν επιπρόσθετες αλλαγές που αποσκοπούν στην εξοικονόμηση χρόνου, προσθέτουν ευελιξία και ευκολία ένταξης μελλοντικών αλλαγών. Επίσης διευκολύνουν αισθητά την αποσφαλμάτωση καθώς ο κώδικας που αφορά τη διαδικασία επεξεργασίας βίντεο είναι τώρα κτισμένος πάνω σε Tcl/Tk και από κάτω C++ οπότε μπορεί να αποσφαλματωθεί με τον ίδιο τρόπο που ακολουθείται και για τον κώδικα του συστήματος ADIVIS. Όλες οι ρυθμίσεις είναι δυναμικές, επιτρέπονται βίντεο με 2 resolution “qcif” και “cif”, επιτρέπεται οποιοδήποτε όνομα βίντεο που έχει extension .yuv, οποιοσδήποτε αριθμός αλλαγών στα στρώματα και τα αρχεία μπορούν να βρίσκονται σε όποιο κατάλογο επιθυμεί ο χρήστης. Επίσης δημιουργήθηκε ένα νέο script με το όνομα “input.tcl” το οποίο αυτοματοποιεί τη διαδικασία δημιουργίας των 8 στρωμάτων κωδικοποίησης και κομμάτι

κώδικα που εκτελείτουν προηγουμένως κάθε φορά που γινόταν κάποια προσωμοίωση τώρα τρέχει μόνο μια φορά μέσω του input.tcl. Σημειώνεται ότι η C++ χρησιμοποιείται μόνο για τις πιο απαιτητικές διαδικασίες κάτω από την Tcl/Tk.

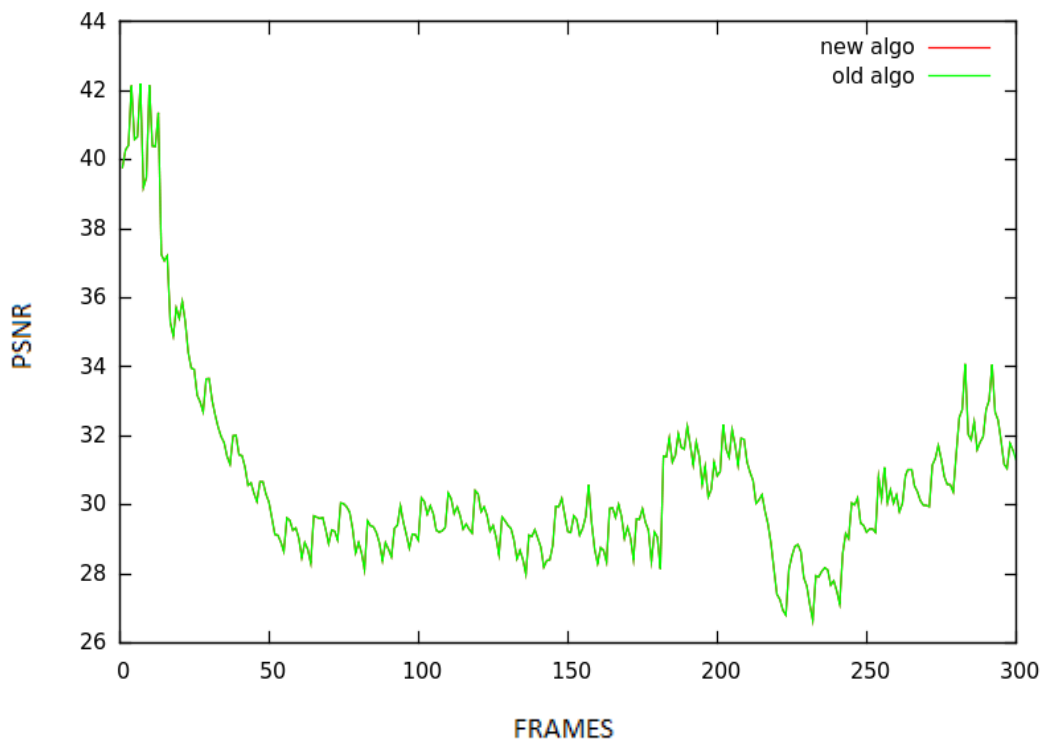
Πιο σημαντικές όμως είναι οι διορθώσεις που έγιναν στους αλγόριθμους που χρησιμοποιούνταν στα αρχεία makeYUV και concatenate καθώς τα αποτελέσματά που παίρναμε πριν ήταν ανακριβής και παρουσίαζαν ότι του average PSNR του παραλήπτη είναι καλύτερο από ότι είναι στην πραγματικότητα στα ασύρματα περιβάλλοντα όπου έχουμε απώλειες πακέτων. Θα παρουσιάσω στη συνέχεια, τη συγκριση μεταξύ των αποτελεσμάτων που πάρθηκαν με την παλαιά διαδικασία επεξεργασίας βίντεο και με την καινούρια.

4.3.4 Αποτελέσματα με τους 2 διαφορετικούς αλγόριθμους

Για το σκοπό σύγκρισης των δυο αλγορίθμων χρησιμοποιήθηκε το ίδιο ακριβώς περιβάλλον και στους δύο. Επίσης έγιναν οι απαραίτητες αλλαγές για να υποστηρίξει η παλιά διαδικασία επεξεργασίας και το “cif” resolution χωρίς όμως να αλλαχτεί οτιδήποτε στον αλγόριθμο που κάνουμε αναφορά. Χρησιμοποιούμε βίντεο με resolution “cif” επειδή όπως αναφέραμε προηγουμένως η διαφορά στον αριθμό πακέτων του κάθε στρώματος είναι μεγαλύτερη. Ο παλιός αλγόριθμος επεξεργασίας φέρνει πάντα καλύτερα αποτελέσματα (ψηλότερο PSNR) επειδή όταν δημιουργεί τα ζευγάρια των sender-receiver traces, όπως αναφέραμε πριν, ο receiver αρχίζει να λαμβάνει τα ίδια πακέτα με τον παραλήπτη σε λάθος πλαίσιο. Οπότε για παράδειγμα αν έπρεπε να διαβαστούν τα πακέτα του πρώτου στρώματος ξεκινώντας από το frame 143, ξεκινούν να διαβάζονται από το frame 130 και τελειώνουν όμως 13 frames νωρίτερα από ότι έπρεπε. Αυτό σημαίνει όμως ότι τα 13 αυτά frames που θα έπρεπε να διαβαστούν από το trace του πραγματικού παραλήπτη (trace παραλήπτη της αρχικής προσομοίωσης) διαβάζονται απευθείας από τον αποστολέα. Αν υπάρχουν οποιεσδήποτε απώλειες πακέτων σε αυτά τα 13 frames δεν θα φανούν στο τελικό βίντεο που θα συναρμολογηθεί. Το δείχνουμε στη συνέχεια με παραδείγματα αποτελεσμάτων χρησιμοποιώντας τους 2 αλγόριθμους. Αρχικά χρησιμοποιούμε ένα ενσύρματο περιβάλλον όπου δεν υπάρχουν απώλειες για να δείξουμε

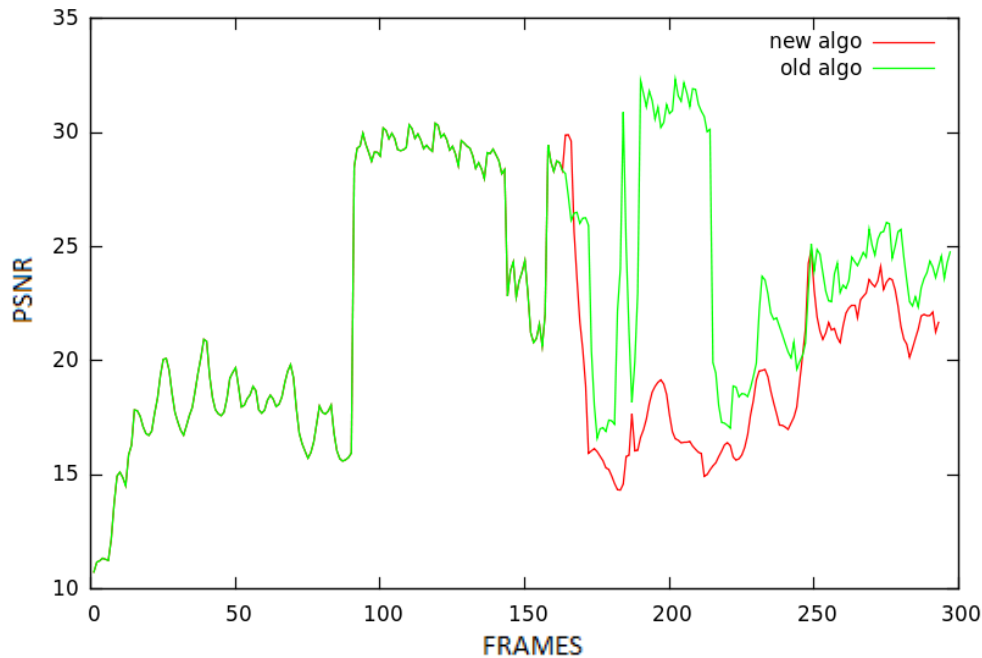
ότι σε αυτή την περίπτωση οι δύο αλγόριθμοι παράγουν το ίδιο αποτέλεσμα καθώς ο παραλήπτης της προσομοίωσης παίρνει όλα τα πακέτα που θα έπρεπε να παίρνει έτσι έχει τα ίδια πακέτα το trace file του παραλήπτη με αυτά που έχει το trace file του αποστολέα.

Όπως παρατηρούμε στο σχήμα 4.6 το PSNR των δύο αλγορίθμων είναι πανομοιότυπο.



Σχήμα 4.6 Ενσύρματο περιβάλλον

Στο σχήμα 4.7 πιο κάτω, βλέπουμε ότι ο παλιός αλγόριθμος έχει πολύ καλύτερο PSNR από ένα σημείο και μετά. Ο μέσος όρος στο PSNR είναι 22.97 για τον παλιό και 20.80 για τον καινούριο.



Σχήμα 4.7 Ασύρματο περιβάλλον

4.4 Διορθώσεις/Βελτιώσεις συστήματος ADIVIS

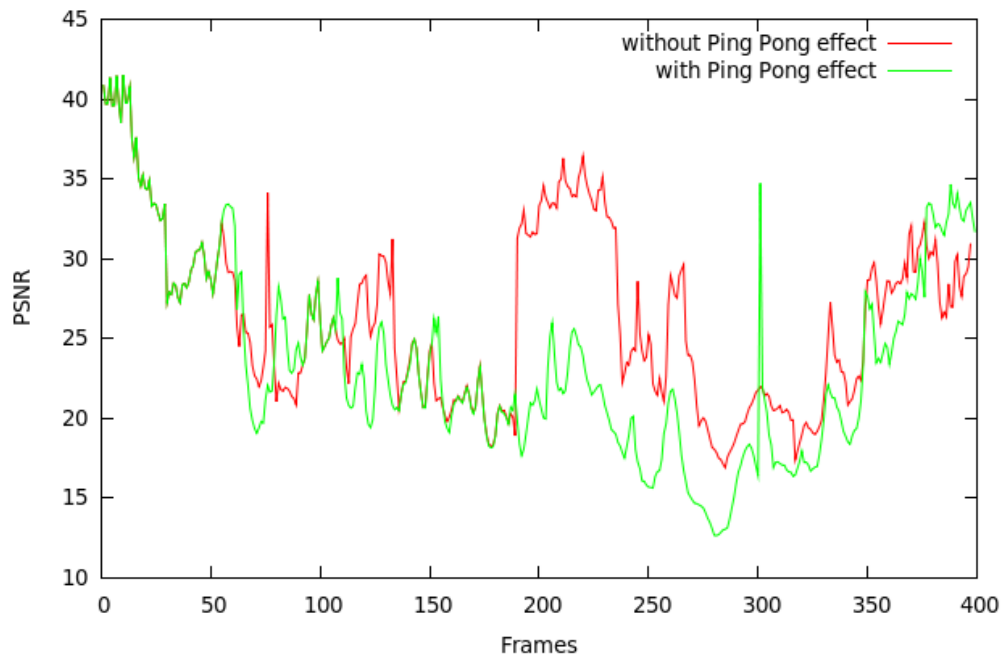
Κρίθηκε αναγκαίο να γίνουν αρκετές διορθώσεις στον κώδικα του συστήματος ADIVIS κυρίως επειδή στο υπάρχον σύστημα δεν εξετάστηκαν αρκετά τα σενάρια με απώλειες πακέτων. Προστέθηκαν ελέγχοι στον κώδικα για διευκρίνιση της ροής του προγράμματος που ακολουθείται και απαλείφθηκαν αρκετά bugs που υπήρχαν κυρίως στο κομμάτι που προστεθεί η υπηρεσία multicast. Συγκεκριμένα κάποια από τα bug που υπήρχαν και διορθώθηκαν είναι τα εξής,

- Δεν υπήρχε καταχώρηση με τα traces του παραλήπτη. Από αυτό φαίνεται ότι εξετάστηκε μόνο η λειτουργία του συστήματος στο δίκτυο και δεν έγινε καθόλου αξιολόγηση του βίντεο μετά την προσθήκη υποστήριξης της υπηρεσίας multicast.
- Υπήρχε ένας έλεγχος στο αρχείο videoFuzzyIe.c που προκαλούσε segmentation fault.

- Εάν δεν λαμβάνονταν πακέτα ελέγχου για περισσότερο από 0.5 δευτερόλεπτα ο αποστολέας δεν επεξεργαζόταν τα πακέτα ελέγχου που έφθαναν μετά.
- Δεν γινόταν σωστή καταχώρηση του μεγέθους των πακέτων σε ασύρματα περιβάλλοντα.
- Στις εναλλαγές κωδικοποιημένου στρώματος υπήρχε καθυστέρηση αποστολής του βίντεο.
- Δεν προσδιοριζόταν το destination port του RTP & RTCP agent με αποτέλεσμα να μην λειτουργούν σωστά κάποια πρωτόκολλα όπως το PUMA και το MAODV.
- Η μέθοδος αναγνώρισης του αποστολέα του κάθε πακέτου δεν λειτουργούσε σωστά σε κάποιες περιπτώσεις.
- Δεν ήταν υλοποιημένη η παρουσίαση της απώλειας πακέτων σε κάθε παραλήπτη στο τέλος της προσομοίωσης (δεν είναι bug αλλά ήταν κάτι που υποστηριζόταν πριν από την προσθήκη υποστήριξης της υπηρεσίας multicast).

Πρόσθεσα 2 ρυθμίσεις στα ενδεικτικά σενάρια του κάθε υποστηριζόμενου περιβάλλοντος που έχουν ψηλό αντίκτυπο στην ποιότητα βίντεο που λαμβάνει ο παραλήπτης,

- Στο [1] αναφέρεται ότι για να αποφευχθούν αποτελέσματα “ring-pong”, δεν θα πρέπει να υπάρχει μετάβαση σε ψηλότερου επίπεδου στρώμα κάθε φορά που το διαθέσιμο εύρος ζώνης ξεπερνά το κατώτατο όριο μιας συγκεκριμένης ταχύτητας μετάδοσης που αντιστοιχεί σε ένα ψηλότερο στρώμα βίντεο. Αυτό όμως δεν εφαρμοζόταν στο υπάρχον σύστημα του ADIVIS. Έχω προσθέσει μια απλή ρύθμιση στην υλοποίηση που επιτρέπει στον ερευνητή να διαλέξει εάν θέλει να τον χρησιμοποιήσει ή όχι. Ο περιορισμός αυτός θεωρητικά βοηθά στην βελτίωση της ποιότητας στην πλευρά του παραλήπτη σε σενάρια όπου υπάρχουν περισσότερες αυξομειώσεις του διαθέσιμου εύρους ζώνης ενώ σε περιπτώσεις όπου δεν αναμένονται ξαφνικές μειώσεις του διαθέσιμου εύρους ζώνης είναι καλύτερο να μην χρησιμοποιείται.



Σχήμα 4.8 Σύγκριση με και χωρίς το ping pong effect

Στο πιο πάνω σχήμα οι μετρήσεις έγιναν χρησιμοποιώντας τις παραμέτρους του ενδεικτικού σενάριου για ασύρματο περιβάλλον με χρήση του πρωτοκόλλου PUMA και του ADIVIS και καθυστέρηση πακέτων 0.003s. Φαίνεται ξεκάθαρα ότι στην περίπτωση που εφαρμόζουμε τον περιορισμό και αφαιρούμε το “Ping-Pong” effect έχουμε ελαφρώς καλύτερη μέση ποιότητα βίντεο στη πλευρά του παραλήπτη (26.31 μέσο όρο σε αντίθεση με 23.59 που έχουμε χωρίς τον περιορισμό). Εάν κοιτάζουμε προσεκτικά όμως, μπορούμε να δούμε ότι υπάρχουν συγκεκριμένα πλαίσια στη γραφική παράσταση όπου η ποιότητα είναι ψηλότερη στην περίπτωση που έχουμε το Ping-Pong effect. Ο λόγος είναι ότι αλλάζοντας στο σενάριο τον περιορισμό αυτό, με το που αρχίζει να στέλνει ψηλότερο στρώμα κωδικοποίησης ο αποστολέας (στο πλαίσιο 62 σε αυτό το σενάριο) έχουμε διαφορετική ροή πακέτων στις δύο περιπτώσεις. Αυτό όμως σημαίνει ότι χάνονται διαφορετικού είδους πακέτα στις δύο περιπτώσεις. Όταν λέμε διαφορετικού είδους πακέτα εννοούμε πακέτα που ανήκουν σε διαφορετικό είδος πλαισίου. Η κωδικοποίησή μας χρησιμοποιεί 4 είδη πλαισίων H, B, P και I.

- Τα πλαίσια I είναι τα πιο σημαντικά αφού αν ένα I πλαίσιο είναι διεφθαρμένο τότε αυτό θα προκαλέσει πρόβλημα στα πλαίσια τα οποία κάνουν αναφορά σε αυτό για

την πρόβλεψή τους.

- Τα πλαίσια P είναι λιγότερο σημαντικά από τα πλαίσια I αλλά περισσότερο σημαντικά από τα πλαίσια B. Ο λόγος είναι επειδή αν υπάρχει κάποιο λάθος στο πρώτο πλαίσιο P τότε αυτό μεταδίδεται στο αμέσως επόμενο πλαίσιο. Σημειώνουμε ότι η κωδικοποίησή του πλαισίου P γίνεται με βάση αναφοράς που γίνεται σε άλλα πλαίσια και συγκεκριμένα σε αναφορά περιεχομένων κάποιου πλαισίου I ή πλαισίου P το οποίο βρίσκεται πριν από αυτό (άρα επηρεάζεται αν το πλαίσιο αυτό είναι διεφθαρμένο).
- Τα πλαίσια B είναι τα λιγότερο σημαντικά καθώς δεν εμπλέκονται στην κωδικοποίηση άλλων πλαισίων (αφού δεν γίνεται αναφορά σε αυτά).
- Τα πλαίσια H δεν είναι καθόλου σημαντικά καθώς δεν επηρεάζουν το τελικό αποτέλεσμα (την εικόνα που βλέπει ο χρήστης).

Έχοντας κάνει αναφορά στη σημασία του κάθε είδος πλαισίου, θα αναφέρω σε κάθε περίπτωση τα πακέτα του κάθε είδους που χάνονται.

Στην περίπτωση που έχουμε το Ping-Pong effect στο πιο πάνω σενάριο έχουμε τις εξής απώλειες:

- πλαίσια B: 3 πακέτα
- πλαίσια P: 1 πακέτο
- πλαίσια I: 9 πακέτα
- πλαίσια H: 0 πακέτα

Στην περίπτωση που δεν έχουμε το Ping-Pong effect στο πιο πάνω σενάριο έχουμε τις εξής απώλειες:

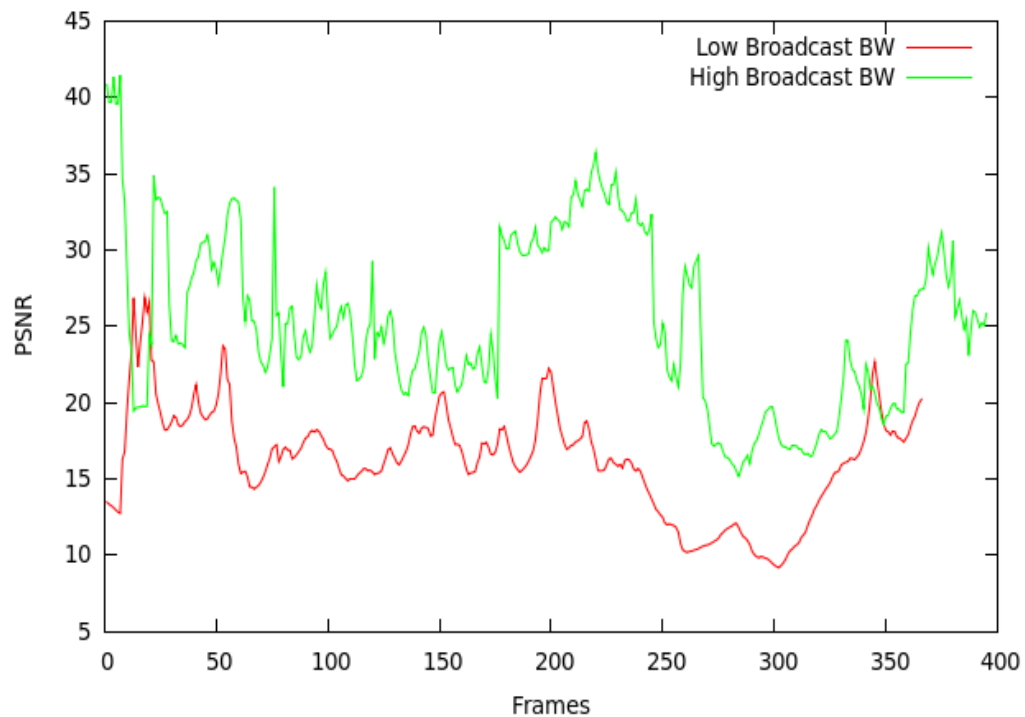
- πλαίσια B: 7 πακέτα
- πλαίσια P: 2 πακέτα
- πλαίσια I: 5 πακέτα
- πλαίσια H: 0 πακέτα

Παρατηρούμε ότι στην πρώτη περίπτωση χάνουμε συνολικά 13 πακέτα ενώ στη δεύτερη

περίπτωση χάνουμε συνολικά 14 πακέτα. Εντούτοις η δεύτερη μας δίνει καλύτερη μέση ποιότητα λόγω μικρότερου αριθμού χαμένων πακέτων που ανήκουν σε πλαίσια I. Το είδος των πακέτων που χάνονται όμως είναι τυχαίο και δεν έχει να κάνει με την εφαρμογή του περιορισμού που εξετάζουμε. Οπότε πρέπει να ληφθεί υπόψη ο αριθμός των χαμένων πακέτων για την σύγκριση των δύο περιπτώσεων. Στην περίπτωση που δεν έχουμε το Ping-Pong effect έχουμε χάσει ένα πακέτο περισσότερο ενώ θεωρητικά θα έπρεπε να έχουμε λιγότερες απώλειες καθώς χρησιμοποιούμε μικρότερο ποσοστό του διαθέσιμου εύρους ζώνης για μεγαλύτερο χρονικό διάστημα. Λόγω αυτού του αποτελέσματος δεν μπορούμε να βγάλουμε ακριβή συμπεράσματα για το αντίκτυπο που έχει ο περιορισμός αυτός στην πράξη χωρίς να γίνει εξέταση σε περισσότερο βάθος. Ο ερευνητής που θα χρησιμοποιήσει τη σουίτα ADIVIS έχει την ελευθερία να χρησιμοποιήσει ή όχι τον περιορισμό αυτό. Τα ενδεικτικά σενάρια έχουν ως προκαθορισμένη επιλογή να μην τον χρησιμοποιούν.

- ο Παρατήρησα χρησιμοποιώντας το πρωτόκολλο PUMA, ότι σε απλά σενάρια όπου έπρεπε θεωρητικά οι απώλειες πακέτων να ήταν πολύ μικρές να μην ίσχυε αυτό. Μετά από αρκετή έρευνα ανακάλυψα ότι αυτό οφειλόταν στη χρησιμοποίηση του ελάχιστου υποστηριζόμενου εύρους ζώνης από το περιβάλλον προσομοίωσης του NS2 για το πρωτόκολλο PUMA. Ο λόγος που συνέβαινε αυτό ήταν επειδή το PUMA μεταδίδει πακέτα χρησιμοποιώντας αποκλειστικά τον μηχανισμό broadcast. Στα περισσότερα πρωτόκολλα αυτός ο μηχανισμός στέλνει λιγοστά πακέτα σε κάθε interval οπότε χρειάζεται ελάχιστο εύρος ζώνης. Για να εξασφαλίσουν την αποστολή των πακέτων μέσω του μηχανισμού αυτού και σε μεγάλες αποστάσεις οι δημιουργοί του εργαλείου NS2 προκάθορισαν να χρησιμοποιεί πάντα το ελάχιστο υποστηριζόμενο εύρος ζώνης ο μηχανισμός αυτός. Για τις ανάγκες μετάδοσης του προσαρμοστικού βίντεο όμως το ελάχιστο υποστηριζόμενο εύρος ζώνης δεν είναι αρκετό όμως καθώς στέλνονται αρκετά πακέτα μεγάλου μεγέθους ανα τακτά χρονικά διαστήματα. Συγκεκριμένα για βίντεο κωδικοποιημένο σε 25 πλαίσια ανα δευτερόλεπτο πρέπει να στέλνουμε ένα πλαίσιο κάθε 0.04 δευτερόλεπτα. Ανάλογα με το bitrate, κάθε πλαίσιο αντιστοιχεί σε X πακέτα μέγιστου μεγέθους 1000 bytes. Όπως αντιλαμβάνεται κανείς είναι αρκετά απαιτητικό σε θέμα εύρους ζώνης. Μετά από έντονη συζήτηση που είχα με τους

προγραμματιστές του NS2 μου ανέφεραν ότι θα μπορούσα να αλλάξω μια ρύθμιση ώστε να χρησιμοποιεί ο μηχανισμός broadcast το επιθυμητό εύρος ζώνης. Σε ψηλότερο εύρος ζώνης όμως θα πρέπει να αποδεχτούμε τη μικρότερη εμβέλεια μετάδοσης. Αλλάζοντας την ρύθμιση αυτή οι απώλειες πακέτων μειώθηκαν αισθητά και η διαφορά φαίνεται στην ποιότητα της εικόνας που βλέπει ο παραλήπτης.

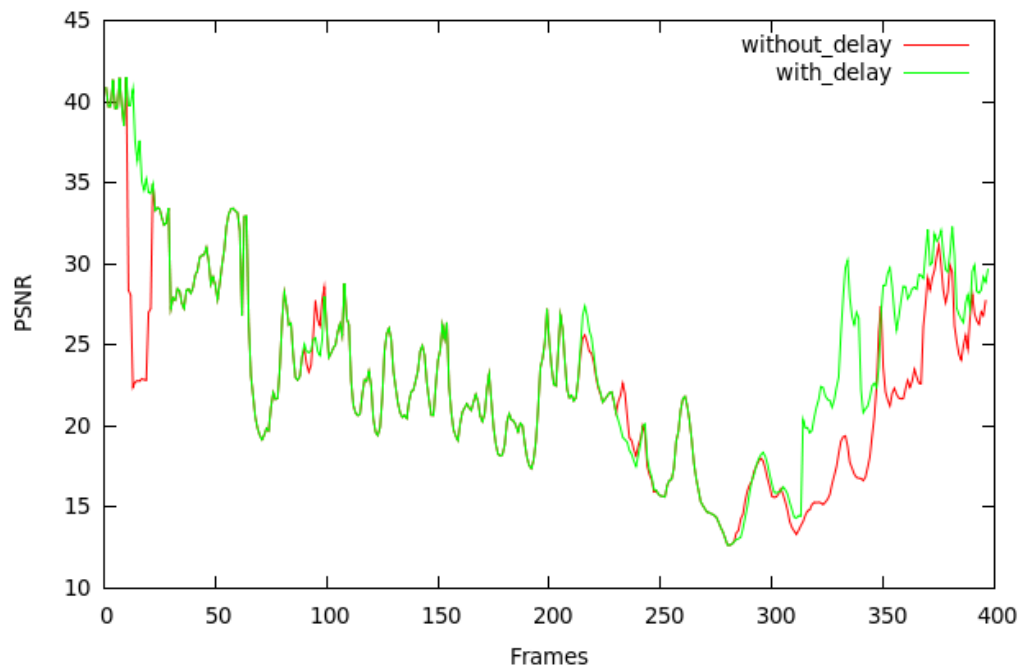


Σχήμα 4.9 Σύγκριση χρήσης χαμηλού και ψηλού BW στον μηχανισμό Broadcast

Στο πιο πάνω σχήμα οι μετρήσεις έγιναν χρησιμοποιώντας τις παραμέτρους του ενδεικτικού σενάριου για ασύρματο περιβάλλον με χρήση του πρωτοκόλλου PUMA και του ADIVIS και καθυστέρηση πακέτων 0.002s. Φαίνεται ξεκάθαρα η διαφορά της ποιότητας που λαμβάνει ο παραλήπτης όταν εφαρμοστεί η ρύθμιση του ψηλότερου εύρους ζώνης για τον μηχανισμό Broadcast.

Επίσης σκέφτηκα έναν τρόπο για να μειωθούν περαιτέρω οι απώλειες πακέτων σε ασύρματα περιβάλλοντα όταν υπάρχει μεταφορά προσαρμοστικού βίντεο. Όπως θα δείξω στη συνέχεια, ο τρόπος αυτός βοηθά αρκετά στη βελτίωση της ποιότητας βίντεο στην

πλευρά του παραλήπτη. Συγκεκριμένα όπως προαναφέρθηκε πριν για βίντεο που δείχνει 25 πλαίσια το δευτερόλεπτο, στέλνονται X πακέτα μέγιστου μεγέθους 1000 bytes κάθε 0.04 δευτερόλεπτα. Ένας από τους λόγους που έχουμε απώλειες πακέτων είναι οι συγκρούσεις που δημιουργούνται λόγω της αποστολής των X πακέτων ακριβώς την ίδια ώρα. Κάνοντας τις απαραίτητες προσθήκες στον κώδικα του συστήματος ADIVIS επιτρέπω στον ερευνήτη να εισάγει μια μικρή καθυστέρηση μεταξύ των πακέτων που στέλνονται σε κάθε burst. Με τον τρόπο αυτό μειώνονται οι συγκρούσεις που υπάρχουν με αποτέλεσμα να έχουμε λιγότερες απώλειες πακέτων και καλύτερη ποιότητα βίντεο στην πλευρά του παραλήπτη. Αυτό φαίνεται ξεκάθαρα στο πιο κάτω σχήμα. Οι μετρήσεις έγιναν χρησιμοποιώντας το ενδεικτικό σενάριο για ασύρματο περιβάλλον με χρήση του πρωτοκόλλου PUMA και του ADIVIS (γίνεται αναφορά στο επόμενο κεφάλαιο). Η καθυστέρηση που χρησιμοποιήθηκε είναι 0.002 δευτερόλεπτα η οποία φαίνεται να φέρνει τα καλύτερα αποτελέσματα σε “qcif” resolution καθώς και σε “cif” resolution.



Σχήμα 4.10 Εισαγωγή καθυστέρησης μεταξύ πακέτων ενός πλαισίου

Κεφάλαιο 5

Περιβάλλοντα προσομοίωσης & Ενδεικτικά σενάρια

5.1 Εισαγωγή

5.2 Υποστηριζόμενα περιβάλλοντα προσομοίωσης

5.3 Ενδεικτικό σενάριο για ασύρματο περιβάλλον Unicast + MIPv4 + ADIVIS

5.4 Ενδεικτικό σενάριο για ασύρματο περιβάλλον Multicast+ADIVIS (PUMA)

5.5 Αποτελέσματα προσομοίωσης PUMA vs MAODV

5.1 Εισαγωγή

Μέρος της καινούριας σουίτας ADIVIS, είναι τα αρχεία ρυθμίσεων για κάθε περιβάλλον προσομοίωσης που υποστηρίζεται και βρίσκονται τοποθετημένα στον ριζικό φάκελο του ADIVIS. Μπορεί να τα χρησιμοποιήσει κάποιος ως έχουν με δεδομένα εισόδου τα ενδεικτικά σενάρια που προσφέρονται στον φάκελο wirelessScenarios ή με καινούρια σενάρια που πιθανώς να θέλει να εξετάσει. Σε αυτό το κεφάλαιο θα περιγράψω σύντομα τα περιβάλλοντα προσομοίωσης που υποστηρίζονται από τη σουίτα ADIVIS και επίσης θα αναφέρω τα ενδεικτικά σενάρια για τα περιβάλλοντα προσομοίωσης που δεν υποστηρίζονταν προηγουμένως.

5.2 Υποστηριζόμενα περιβάλλοντα προσομοίωσης

Μετά από υπόδειξη του επιβλέποντα καθηγητή μου, δημιούργησα ένα αρχείο για κάθε περιβάλλον προσομοίωσης το οποίο περιλαμβάνει τις αναγκαίες ρυθμίσεις για να λειτουργήσει το περιβάλλον μέσω της πλατφόρμας του NS2. Έτσι ξεκαθάρισαν ποια περιβάλλοντα υποστηρίζονται όπως πρέπει και τα ρυθμισμένα αρχεία που δημιουργήθηκαν, τοποθετήθηκαν στον ριζικό φάκελο της σουίτας ADIVIS.

Ενσύρματα περιβάλλοντα

- Unicast+ADIVIS: Υποστηρίζεται. Αυτό είναι το πιο απλό και βασικό περιβάλλον όπου ο αποστολέας στέλνει βίντεο σε ένα ή περισσότερους παραλήπτες χρησιμοποιώντας τη μέθοδο αποστολής unicast και παράλληλα το σύστημα ADIVIS για προσαρμογή της βιντεοροής. Σχετικό αρχείο με δοκιμασμένη προσομοίωση που δουλεύει είναι το “unicast.tcl”.
- Unicast+MIPv4+ADIVIS: Υποστηρίζεται. Διαφέρει από το προηγούμενο περιβάλλον στο ότι χρησιμοποιείται το πρωτόκολλο MIPv4 για ενσύρματα δίκτυα το οποίο δημιουργήθηκε από τη SUN. Σε αυτό το περιβάλλον η προσομοίωση που γίνεται “ξεγέλα” τους κόμβους για να νομίσουν ότι βρίσκονται σε ασύρματο περιβάλλον ώστε να λειτουργούν ανάλογα. Για να προσομοιωθεί ο μηχανισμός Handoff, ρυθμίζουμε τον NS2 στη χρονική στιγμή που επιθυμούμε να ρίξει κάτω το link που ενώνει τον κινητό χρήστη με τον Home Agent και να ενεργοποιήσει αυτό που ενώνει τον κινητό χρήστη με τον Foreign Agent. Σχετικό αρχείο με δοκιμασμένη προσομοίωση που δουλεύει είναι το “unicast_mip.tcl”.
- MULTICAST+ADIVIS: Υποστηρίζεται. Για το σχετικό περιβάλλον έγιναν κάποιες προσθήκες/βελτιώσεις όπως αναφέρεται στο κεφάλαιο 5. Ουσιαστικά είναι το ίδιο περιβάλλον με το πρώτο με τη σημαντική διαφορά ότι ο αποστολέας στέλνει τα δεδομένα χρησιμοποιώντας τη μέθοδο αποστολής multicast. Σχετικό αρχείο με δοκιμασμένη προσομοίωση που δουλεύει είναι το “multicast.tcl”.
- MULTICAST+MIPv4: Δεν υποστηρίζεται. Το περιβάλλον αυτό δεν υποστηρίζεται μέσω του NS2 και δεν έγινε μεγάλη προσπάθεια για να λειτουργήσει καθώς το MIPv4 πρωτόκολλο της SUN θεωρείται πεπαλαιωμένο και θα αφαιρεθεί τελείως

μελλοντικά από την πλατφόρμα του NS2. Επίσης με αυτό το περιβάλλον δεν θα μπορούσαμε να εξετάσουμε τη ρεαλιστική λειτουργία του ADIVIS σε κινητά δίκτυα με το MIPv4 πρωτόκολλο καθώς το “ψεύτικο” ασύρματο περιβάλλον που χρησιμοποιείται έχει ως σκοπό την εξέταση της λειτουργίας του πρωτόκολλου MIPv4. Ως εκ τούτου πολλές παραμέτροι όπως η απόσταση των κόμβων και η μετακίνηση τους απουσιάζουν ή δεν υλοποιούνται όπως θα υλοποιούνταν σε πραγματικά περιβάλλοντα.

Ασύρματα περιβάλλοντα

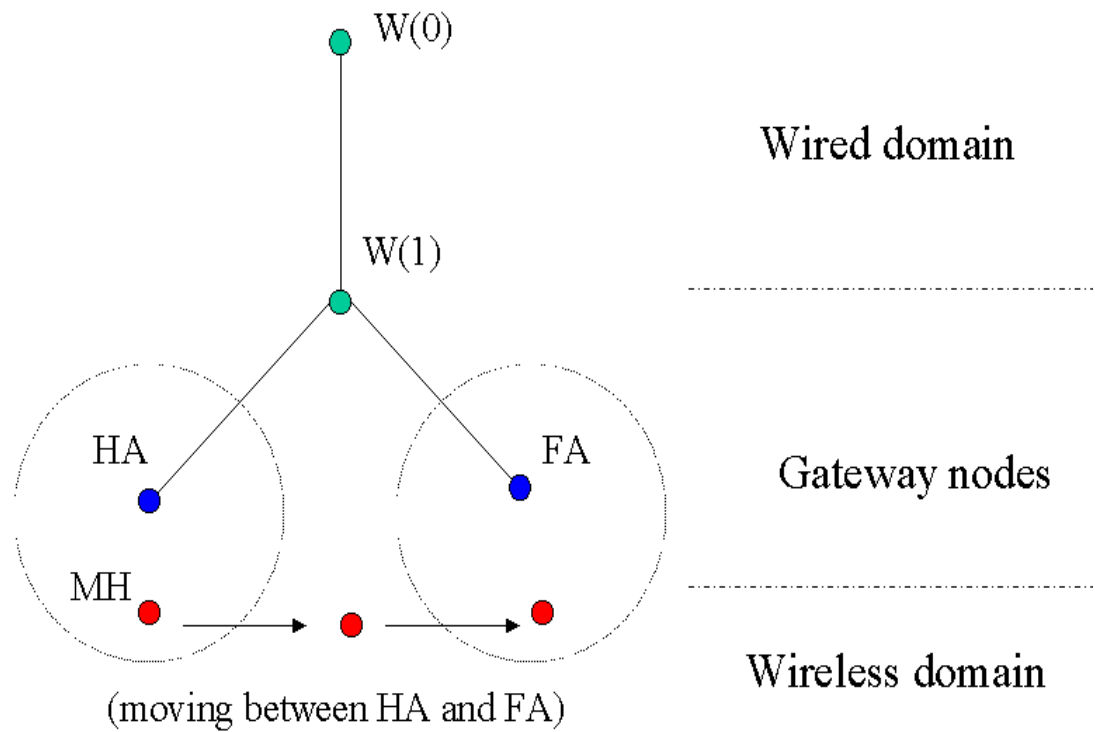
- Unicast+ADIVIS: Υποστηρίζεται. Η περιγραφή του είναι η ίδια με αυτή στο αντίστοιχο ενσύρματο περιβάλλον. Λόγω της ασύρματης φύσης του περιβάλλοντος υπάρχουν κι άλλοι παράμετροι που επηρεάζουν την προσομοίωση όπως ή απόσταση μεταξύ των κόμβων, η ταχύτητα τους, η εμβέλεια μετάδοσης καθώς και η εμβέλεια παρεμβολών (που είναι προκαθορισμένες στα 250 και 550 μέτρα αντίστοιχα) . Σχετικό αρχείο με δοκιμασμένη προσομοίωση που δουλεύει είναι το “unicast_adhoc.tcl”.
- Multicast+ADIVIS: Υποστηρίζεται. Για να υποστηριχτεί χρειάστηκε να γίνουν κάποιες προσθήκες και αλλαγές στον πηγαίο κώδικα του ADIVIS καθώς και να εφαρμοστεί προσθήκη στον NS2 ενός wireless multicast πρωτόκολλου. Οι αλλαγές αυτές δεν ήταν προφανείς και γίνεται αναφορά σε αυτές στο κεφάλαιο 5. Είναι το ίδιο περιβάλλον με το προηγούμενο με τη σημαντική διαφορά ότι ο αποστολέας στέλνει τα δεδομένα χρησιμοποιώντας τη μέθοδο αποστολής multicast. Σχετικό αρχείο με δοκιμασμένη προσομοίωση που δουλεύει είναι το “multicast_adhoc.tcl”.
- Unicast+MIPv4+ADIVIS: Υποστηρίζεται. Λειτουργεί μέσω του σεναρίου “wired-cum-wireless” όπου ο αποστολέας είναι ενωμένος ενσύρματα με τον Home Agent και Foreign Agent και στέλνει μέσω αυτόν δεδομένα σε κινητούς χρήστες. Σημαντικό να αναφερθεί ότι δεν χρησιμοποιείται το μοντέλο της SUN αλλά αυτό της CMU Monarch. Σχετικό αρχείο με δοκιμασμένη προσομοίωση που δουλεύει είναι το “unicast_wireless_mip.tcl”.
- Multicast+MIPv4+ADIVIS: Δεν υποστηρίζεται. Συγκεκριμένα για να υποστηριχτεί έπρεπε να υποστηριχτεί πρώτα το περιβάλλον wireless+multicast+ADIVIS, έργο το

οποίο εξελίχθηκε αρκετά χρονοβόρο. Παρόλα αυτά μελετήθηκε η υποστήριξη του και με τη γνώση που απέκτησα μπορώ να πω με σιγουριά ότι μπορεί να υποστηριχτεί και αυτό το περιβάλλον νοουμένου ότι θα γίνουν οι απαραίτητες αλλαγές στον πηγαίο κώδικα του NS2. Για να επιτευχθεί αυτό θα χρειαστεί αυτός που θα δοκιμάσει να έχει πρώτα αποκτήσει αρκετή τριβή με τον NS2 και να μελετήσει την τεκμηρίωση που έχω βρεί μετά από αρκετό ψάξιμο, στον φάκελο software της σουίτας ADIVIS.

Παρακάτω θα παρουσιάσουμε τον τρόπο λειτουργίας σε ένα απλό σενάριο για κάθε περιβάλλον που δεν υποστηρίζονταν σε προηγούμενες εκδόσεις της σουίτας ADIVIS .

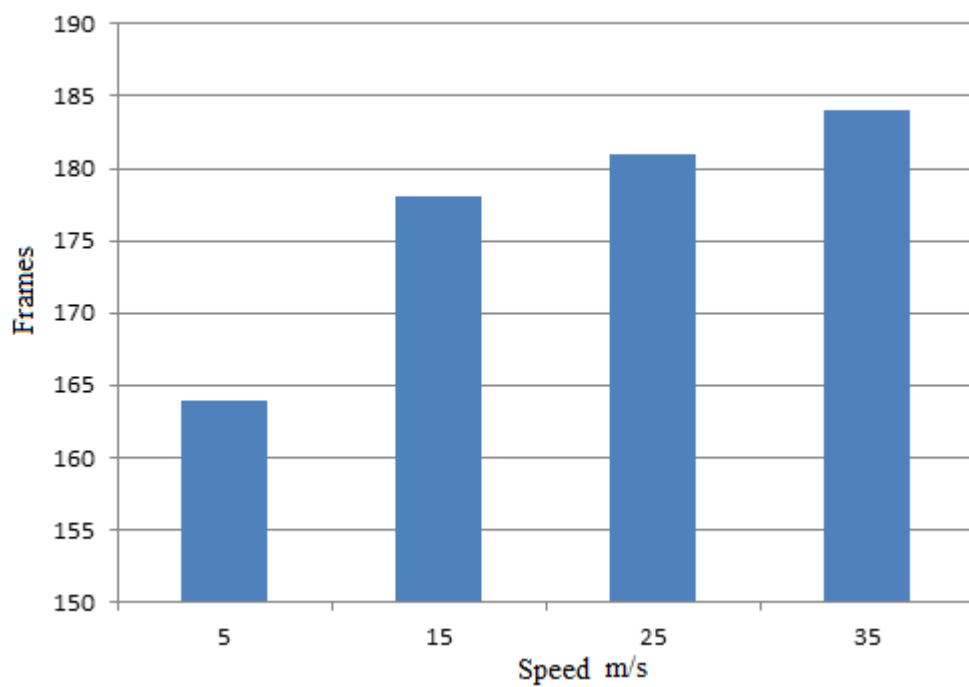
5.3 Ενδεικτικό σενάριο για ασύρματο περιβάλλον Unicast + MIPv4 + ADIVIS

Για αυτό το περιβάλλον χρησιμοποιήθηκε ένα απλό σενάριο το οποίο δείχνει τον μηχανισμό handover του MIPv4. Επίσης ως βίντεο εισόδου χρησιμοποιήθηκε το βίντεο foreman_qcif.yuv το οποίο συμπεριλαμβάνεται στη σουίτα στον φάκελο savedInput. Αξίζει να αναφερθεί ότι το σενάριο αυτό λέγεται wired-cum-wireless και για να υποστηριχτεί, υποχρεωτικά πρέπει να χρησιμοποιηθούν ιεραρχικές διευθύνσεις λόγω του τρόπου λειτουργίας του NS2. Στο σενάριο ο αποστολέας, ο οποίος αναγράφεται ως W(0) στο πιο κάτω σχήμα, είναι ενωμένος με τον Home Agent (HA) και τον Foreign Agent (FA) ενσύρματα (με 10 Mbit links) μέσω κάποιου δρομολογητή με ονομασία W(1). Οι ασύρματοι κόμβοι σε αυτή την προσομοίωση χρησιμοποιούν τις προκαθορισμένες ρυθμίσεις ασύρματης εμβέλειας, συγκεκριμένα έχουν εμβέλεια μετάδοσης 250 μέτρα και εμβέλεια παρεμβολών 550 μέτρα. Επέλεξα τις συντεταγμένες τους με βάση την εμβέλεια τους για να γίνει αναγκαστικά handover. Ο Home Agent βρίσκεται στις συντεταγμένες (X:100, Y:250, Z:0) και ο Foreign Agent στις συντεταγμένες (X:605, Y:250, Z:0). Ο παραλήπτης (MH) αρχικά βρίσκεται στις συντεταγμένες (X:345, Y:250, Z:0) και στο 8ο δευτερόλεπτο της προσομοίωσης αρχίζει να μετακινείται προς τις συντεταγμένες (X:600, Y:250, Z:0) με ευθύγραμμη κίνηση. Ο αποστολέας αρχίζει την μετάδοση βίντεο στο 5ο δευτερόλεπτο και το βίντεο διαρκεί 17 δευτερόλεπτα. Παραθέτω ένα βασικό σχέδιο του σεναρίου παρακάτω στο σχήμα 4.1.

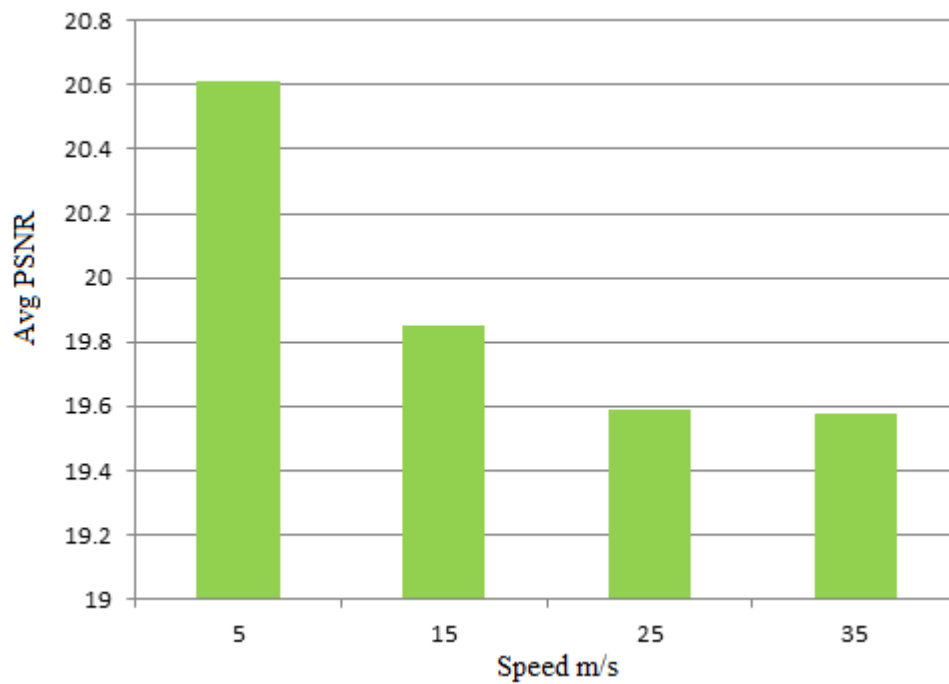


Σχήμα 5.1 Ασύρματο περιβάλλον unicast+MIPv4+ADIVIS

Αφού τρέξουμε την προσομοίωση, παρατηρούμε μελετώντας το αρχείο που παράγει ο NS2 ότι στη χρονική στιγμή 8.188931400 ο Home Agent αρχίζει να κάνει drop τα πακέτα που προορίζονται για τον παραλήπτη ενώ στη χρονική στιγμή 14.003552 παραλαμβάνει ο Foreign agent το πρώτο πακέτο. Συγκεκριμένα τα πακέτα που στέλνει ο αποστολέας, αφού γίνει το handover, αποστέλλονται στον Home Agent ο οποίος με τη σειρά του τα στέλνει στον Foreign Agent μέσω του δρομολογητή $W(1)$ για να σταλούν στον παραλήπτη. Χρησιμοποιήθηκε το ίδιο σενάριο 4 φορές με τον παραλήπτη να έχει διαφορετική ταχύτητα κάθε φορά για να συγκρίνουμε την επίδραση της ταχύτητας στην λήψη του βίντεο. Χρησιμοποιήθηκαν οι ταχύτητες 5 m/s , 15 m/s, 25 m/s και 35 m/s.



Σχήμα 5.2 Χαμένα πλαίσια για την αντίστοιχη ταχύτητα



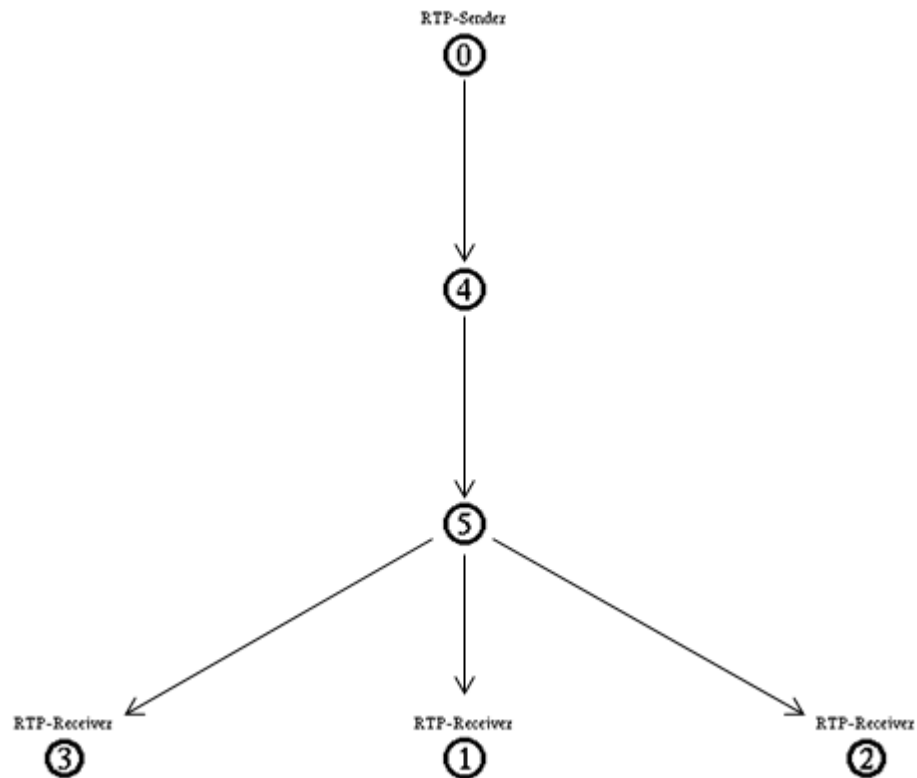
Σχήμα 5.3 Average PSNR για την αντίστοιχη ταχύτητα

Όπως βλέπουμε από τις γραφικές παραστάσεις η μεγαλύτερη διαφορά στην ποιότητα λήψης του βίντεο είναι από τα 5 m/s στα 15 m/s. Στα 5 m/s χάνονται πιο αρκετά πιο λίγα frames και το average PSNR είναι μεγαλύτερο σχεδόν κατά μια μονάδα από αυτό στα 15 m/s. Από τα 15 m/s και άνω βλέπουμε ότι η διαφορά στην ποιότητα λήψης βίντεο είναι πολύ μικρή. Αυτό συμβαίνει λόγω της υπόθεσης που γίνεται στο ασύρματο μοντέλο που χρησιμοποιεί ο NS2 ότι σε πολύ χαμηλές ταχύτητες δεν αλλάζει τόσο πολύ το περιβάλλον που περιτριγυρίζει τον ασύρματο κόμβο οπότε η πιθανότητα να δεχτεί πακέτα είναι μεγαλύτερη.

5.4 Ενδεικτικό σενάριο για ασύρματο περιβάλλον Multicast+ADIVIS (PUMA)

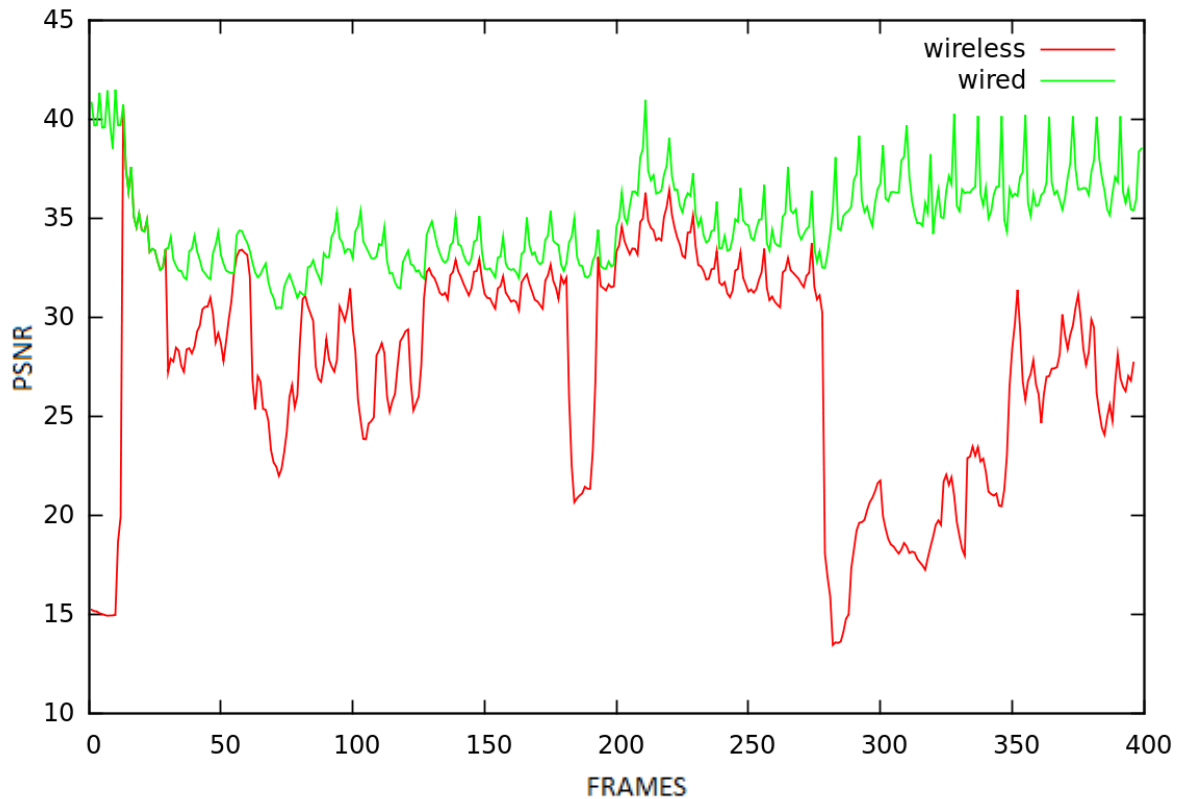
Για την προσομοίωση αυτή χρησιμοποιήθηκε το πρωτόκολλο PUMA το οποίο περιγράφεται στο πρώτο κεφάλαιο. Κατασκεύασα ένα σενάριο παρόμοιο με αυτό του αντίστοιχου ενσύρματου περιβάλλοντος για να είναι πιο ρεαλιστική η σύγκριση τους. Στο σενάριο αυτό γίνεται η υπόθεση ότι όλοι οι ασύρματοι κόμβοι έχουν την δυνατότητα χρησιμοποιήσουν εύρος ζώνης 11Mbit. Ο αποστολέας βρίσκεται στην εμβέλεια ενός μόνο κόμβου ο οποίος προωθεί τα πακέτα του αποστολέα στον επόμενο κόμβο που βρίσκεται στην εμβέλεια του. Ο τελευταίος έχοντας επικοινωνία με τους παραλήπτες αντιγράφει το πακέτο του αποστολέα όσες φορές χρειάζεται (όπως προβλέπει ο μηχανισμός multicast) και το στέλνει σε κάθε παραλήπτη ξεχωριστά. Ο αποστολέας έχει συντεταγμένες (X:530 Y:830, Z:0) και οι 2 αμέσως επόμενοι κόμβοι (X:530, Y:700, Z:0) και (X:530, Y:570, Z:0) αντίστοιχα. Συγκεκριμένα τοποθετήθηκαν με 130 μέτρα απόσταση μεταξύ τους. Οι 3 παραλήπτες έχουν συντεταγμένες (X:310, Y:440, Z:0), (X:530, Y:440, Z:0) και (X:750, Y:440, Z:0) έχοντας 220 μέτρα απόσταση μεταξύ τους. Οι αποστάσεις αυτές είναι σκόπιμες για να υπάρχουν λιγότερες παρεμβολές και κατά συνέπεια μικρότερες απώλειες πακέτων. Αξίζει να σημειωθεί ότι λόγω του τρόπου λειτουργίας του πρωτόκολλου PUMA τα αποτελέσματα μετά από κάθε προσομοίωση είναι ελαφρώς διαφορετικά. Αυτό οφείλεται κυρίως στο ότι χρησιμοποιεί τυχαία καθυστέρηση για να προσθέσει jitter όταν στέλνει πακέτα για να αποφευχθούν όσο γίνεται οι συγκρούσεις. Όμως για το σκοπό αναπαραγωγής των ίδιων αποτελεσμάτων υπάρχει η δυνατότητα να αφαιρεθεί η τυχαιότητα (randomness) αλλάζοντας τη σταθερά SEED από 0 σε 1 στο αρχείο puma.h.

Για αυτό το σενάριο θα αφαιρέσω την τυχαιότητα οπότε αν θελήσει κάποιος μπορεί να αναπαράγει τα ίδια αποτελέσματα χρησιμοποιώντας τις παραμέτρους που δίνονται εδώ. Η τοπολογία που αναφέρθηκε πριν φαίνεται και στο πιο κάτω σχήμα:



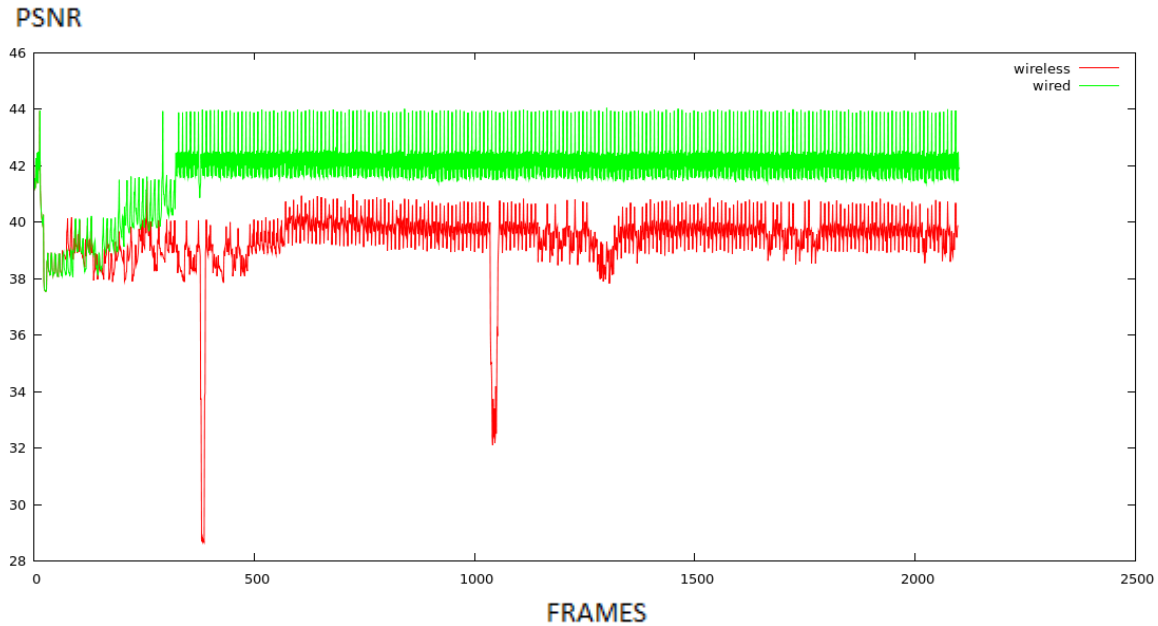
Σχήμα 5.4 Ασύρματο περιβάλλον Multicast+ADIVIS

Για το σκοπό σύγκρισης με το ενσύρματο περιβάλλον multicast θα χρησιμοποιήσουμε το βίντεο που έλαβε ο παραλήπτης 1 επειδή είναι σε πιο μικρή απόσταση από τον κόμβο 5 οπότε έχει ελαφρώς καλύτερη λήψη από τους άλλους 2. Χρησιμοποιήθηκε το ίδιο βίντεο με αυτό του προηγούμενου σενάριου.



Σχήμα 5.5 Ενσύρματο & Ασύρματο multicast + ADIVIS

Ο άξονας X αναπαριστά τον αριθμό του πλαισίου και ο άξονας Y το αντίστοιχο PSNR. Όπως φαίνεται στο πιο πάνω σχήμα η διαφορά στην ποιότητα που λαμβάνει ο χρήστης είναι αρκετά αισθητή. Στην περίπτωση του wireless multicast σε συγκεκριμένα πλαίσια υπάρχει μια δραματική διαφορά από το PSNR του wired multicast. Υποθέτουμε ότι είναι επειδή αλλάζει αρκετά η εικόνα απότομα (πράγμα το οποίο συνεπάγεται περισσότερα πακέτα την ώρα του burst άρα περισσότερες καθυστερήσεις ή απώλειες στην πλευρά του παραλήπτη) και θα εξετάσουμε την υπόθεση αυτή χρησιμοποιώντας διαφορετικό βίντεο εισόδου το οποίο έχει 2101 πλαίσια αντί 400 που είχε το προηγούμενο και υπάρχουν πολύ λίγες σκηνές που αλλάζει αισθητά η εικόνα απότομα. Το βίντεο λέγεται bridge-far_qcif.yuv και συμπεριλαμβάνεται στη σουίτα ADIVIS στον φάκελο savedInput.



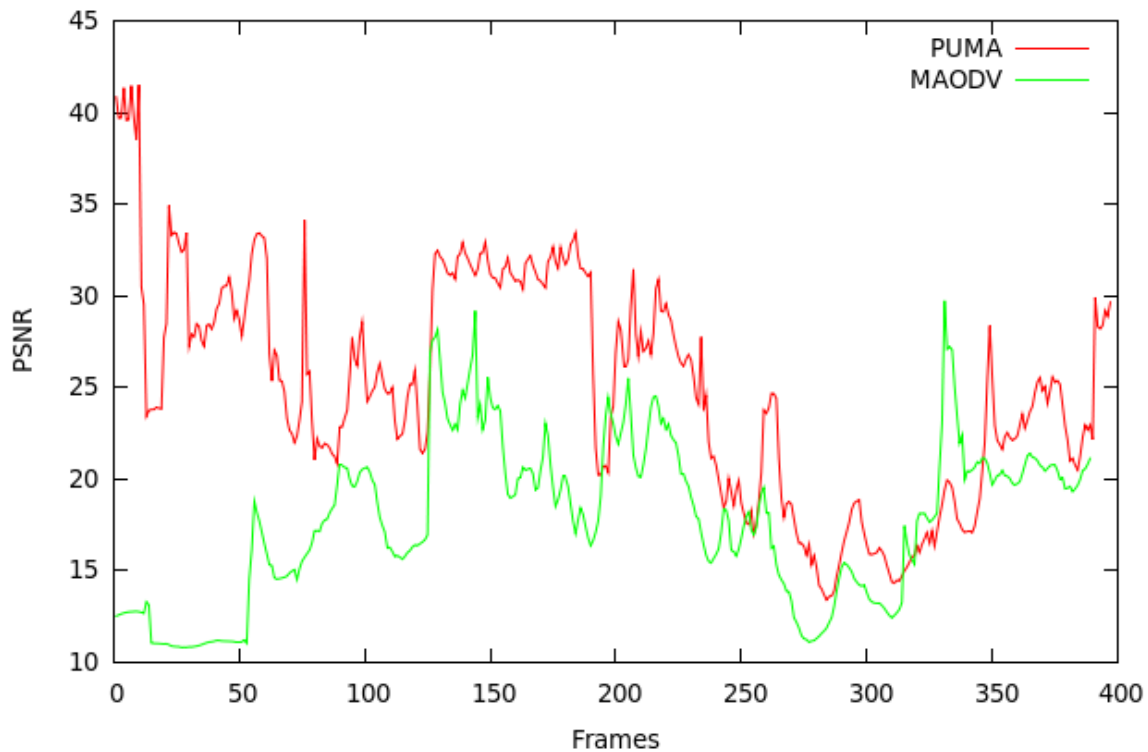
Σχήμα 5.6 Ενσύρματο & Ασύρματο multicast + ADIVIS

Όπως βλέπουμε στο σχήμα υπάρχουν 2 σημεία όπου πέφτει αισθητά το PSNR στην περίπτωση του ασύρματου περιβάλλοντος. Και στα δύο σημεία αλλάζει η εικόνα απότομα. Στο πρώτο σημείο δεν φαίνεται ξεκάθαρα αυτό αλλά αν παρατηρήσει κάποιος προσεκτικά αλλάζει κάπως η εικόνα από την κίνηση της θάλασσας στο πλαίσιο 378. Στο πλαίσιο 1037 φαίνεται ξεκάθαρα ότι αλλάζει η εικόνα από ένα πουλί που πετά πολύ γρήγορα. Στο πλαίσιο που μπαίνει το πλοίο στην εικόνα υπάρχει μόνο ελαφριά πτώση στο PSNR γιατί κινείται αργά και τα πλαίσια αλλάζουν σταδιακά. Στην περίπτωση του ενσύρματου περιβάλλοντος και στα δύο βίντεο δεν επηρεάζεται από απότομες αλλαγές της εικόνας καθώς εξακολουθεί να μην έχει απώλειες πακέτων όταν στέλνονται περισσότερα πακέτα την ίδια ώρα.

5.5 Αποτελέσματα προσομοίωσης PUMA vs MAODV

Θα συγκρίνουμε τα 2 πρωτόκολλα multicast χρησιμοποιώντας το ίδιο ενδεικτικό σενάριο που χρησιμοποιήθηκε και στην προηγούμενη προσομοίωση και ακόμη ενά σενάριο που έχει 19 παραλήπτες και 1 αποστολέα σε μορφή πλέγματος. Αξίζει να σημειωθεί ότι χρειάζεται να βγούν κι άλλα αποτελέσματα μελλοντικά σε σενάρια που έχουν μεγάλες

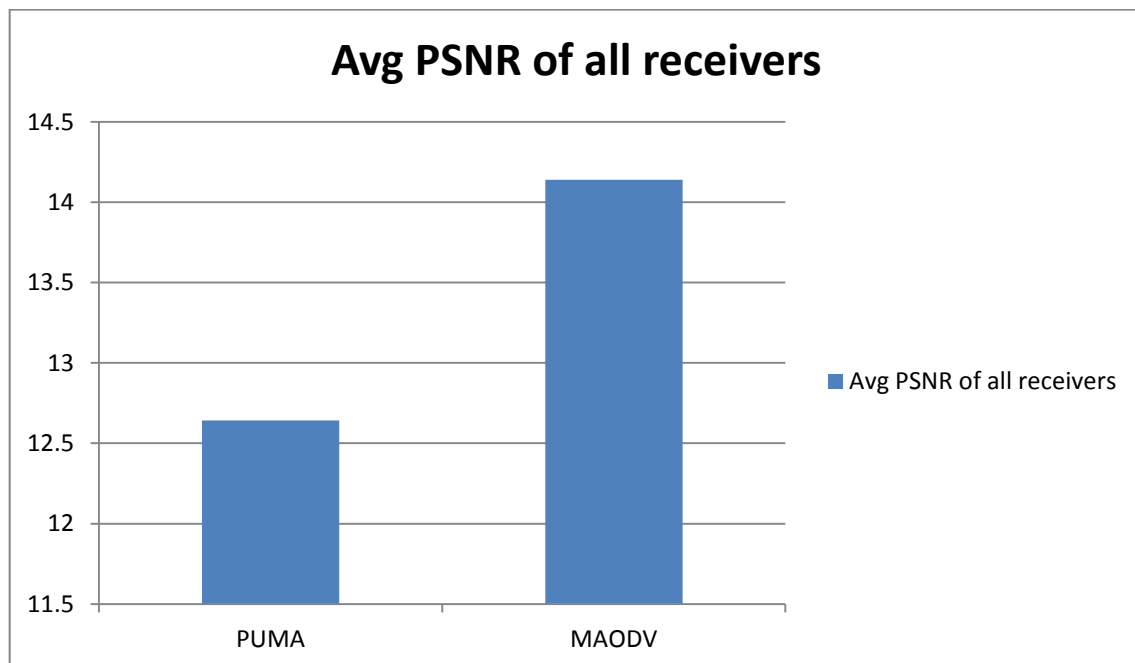
ομάδες από παραλήπτες αλλά τα συγκεκριμένα αποτελέσματα δείχνουν τον πιο αισθητό τομέα που υστερεί και υπερτερεί το κάθε πρωτόκολλο.



Σχήμα 5.7 Σενάριο με 3 παραλήπτες PUMA vs MAODV

Από το πίο πάνω σχήμα φαίνεται ότι το PUMA υπερτερεί σημαντικά έναντι του MAODV σε αυτό το σενάριο. Αυτό οφείλεται στο ότι με τον MAODV ένας σημαντικός αριθμός πακέτων που φθάνει σωστά στον παραλήπτη απρρίπτεται επειδή φθάνει “αργοπορημένα”. Ο λόγος που γίνεται αυτό είναι το τυχαίο jitter που προσθέτει ο MAODV. Αξίζει να σημειωθεί ότι και το PUMA εισάγει υπό κανονικές συνθήκες τυχαίο jitter το έχω αφαιρέσει για καλύτερη απόδοση του πρωτόκολλου. Ο MAODV δεν εξετάστηκε σε τόσο βάθος επειδή στο [12] αναφέρεται ότι έχει χειρότερες επιδόσεις. Παρολαυτά έχει προστεθεί και η υποστήριξη του MAODV επειδή επιτρέπει ταυτόχρονα και την αποστολή πακέτων μέσω της υπηρεσίας unicast. Ο λόγος που μας ενδιαφέρει αυτό είναι επειδή ιδανικά τα πακέτα RTCP στέλνονται μέσω αυτής της υπηρεσίας καθώς μόνο ο αποστολέας πρέπει να τα παραλάβει και όχι όλο το group όπως συμβαίνει με το PUMA, γεγονός που προσθέτει σημαντικό traffic στο δίκτυο ειδικά όταν έχουμε να κάνουμε με πιο μεγάλες ομάδες

παραλήπτων. Αυτό φαίνεται και στην πιο κάτω γραφική παράσταση όπου εξετάστηκε το ίδιο σενάριο αλλά με διαφορετικό αριθμό παραλήπτων (19). Μπορούμε να δούμε ότι οι παραλήπτες στο MAODV κατά μέσο όρο παίρνουν καλύτερη ποιότητα (PSNR average 14.139) από αυτούς στο PUMA (PSNR average 12.642). Ο λόγος είναι το αυξημένο RTCP traffic που παρουσιάζεται στο PUMA το οποίο προκαλεί πολύ περισσότερες συγκρούσεις, μειώνει το διαθέσιμο εύρος ζώνης στους κόμβους και τελικά τα πλείστα πακέτα RTCP σε αυτό το σενάριο δεν φθάνουν στον αποστολέα με αποτέλεσμα να μην φθάνει σχεδόν καθόλου feedback στον αποστολέα ώστε να προσαρμόσει τη ροή του βίντεο ανάλογα. Στο MAODV υπάρχουν πολύ μικρές απώλειες πακέτων RTCP επειδή χρησιμοποιεί την υπηρεσία unicast για να τα στείλει. Οι απώλειες πακέτων RTP είναι επίσης μικρότερες λόγω λιγότερου traffic που υπάρχει συγκριτικά με αυτό στο PUMA αλλά όπως αναφέραμε πριν μεγάλος αριθμός των πακέτων απορρίπτεται επειδή δεν φθάνει στην σωστή σειρά.



Σχήμα 5.8 Σενάριο με 19 παραλήπτες

Κεφάλαιο 6

Συμπεράσματα & Μελλοντική Εργασία

6.1 Συμπεράσματα

6.2 Μελλοντική Εργασία

6.1 Συμπεράσματα

Το μεγαλύτερο μέρος του χρόνου που αφιερώθηκε στην εργασία αυτή ήταν στην υλοποίηση της σουίτας ADIVIS και όχι στην εξαγωγή αποτελεσμάτων. Παρολαυτά βγήκαν κάποια πρόχειρα αποτελέσματά για ενδεικτικούς σκοπούς και για να δείξουν ότι δουλεύουν σωστά όλες οι λειτουργίες που περιλαμβάνει η σουίτα που ετοιμάστηκε. Από τα αποτελέσματά που έχουν αναφερθεί στο προηγούμενο κεφάλαιο τα σημαντικότερα είναι τα εξής :

- Το περιεχόμενο του κάθε βίντεο επηρεάζει τον φόρτο του δικτύου. Εάν για παράδειγμα υπάρχουν συνεχώς αλλαγές της εικόνας (π.χ. το βίντεο foreman) στο περιεχόμενο του βίντεο θα υπάρχει μεγαλύτερος φόρτος στο δίκτυο παρά εάν το βίντεο περιέχει μακρινές σκηνές μιας κάμερας παρακολούθησης (π.χ το βίντεο bridge-far).
- Το PUMA υπερτερεί του MAODV σε σενάρια με μικρές ομάδες παραλήπτων ενώ το MAODV υπερτερεί στα σενάρια με μεγάλες ομάδες παραλήπτων. Υπάρχει περιθώριο βελτίωσης και στα δύο. Θα μπορούσε το PUMA να συνδυαστεί με ένα unicast routing

agent ώστε να στέλνει τα πακέτα RTCP μέσω της υπηρεσίας unicast και να μειώσει έτσι το ποσοστό εύρος ζώνης που χρησιμοποιείται από τα RTCP πακέτα καθώς και τις συγκρούσεις γενικά. Για το MAODV θα μπορούσε να αφαιρεθεί η τυχαιότητα του jitter ώστε να μην απορρίπτονται πακέτα λόγω λάθος σειράς. Ακολούθως θα μπορούσε να επανεξεταστεί η επίδοση και των δύο σε συνδυασμό με τον ADIVIS.

6.2 Μελλοντική Εργασία

Με το πέρας αυτής της διπλωματικής εργασίας, θα ήθελα να παραδώσω τη σκυτάλη στον επόμενο γιατί πιστεύω ότι παρόλο που υπάρχει μια ολοκληρωμένη σουίτα ADIVIS τώρα, υπάρχει αρκετή δουλειά που μπορεί να γίνει ακόμη. Συγκεκριμένα με τη σουίτα που ήδη υπάρχει μπορούν να βγουν περισσότερα αποτελέσματά εξετάζοντας τον αλγόριθμο διαμοιρασμού πόρων (multicast) καθώς και την επίδραση που έχει ο αριθμός χρηστών στο συνολικό μέσο όρο του PSNR και στον αριθμό πακέτων feedback (Receiver Reports) που λαμβάνονται επιτυχώς από τον αποστολέα. Από πλευράς υλοποίησης μπορεί μελλοντικά να υποστηριχτεί το ασύρματο περιβάλλον multicast+MIPv4+ADIVIS (wired-cum-wireless). Συγκεκριμένα εφόσον τώρα υποστηρίζεται σε ασύρματο περιβάλλον το multicast+adivis θα μπορούσε κάποιος να αλλάξει τον κώδικα του NS2 ώστε να υλοποιηθεί το πρωτόκολλο multicast μέσω των κόμβων που κάνουν broadcast. Άλλες επιλογές μελλοντικής εργασίας είναι να δοκιμαστεί περαιτέρω η επέκταση CIMS του Pedro Estrela που ανοίγει πόρτες για πιο πολύπλοκες τοπολογίες, να χρησιμοποιηθεί νεότερη έκδοση του FFMPEG για την δημιουργία των κωδικοποιημένων στρωμάτων (η οποία επιτρέπει τον προκαθορισμό του τελικού αριθμού των πλαισίων που θα επιφέρει ακόμη πιο ακριβή αποτελέσματα) και τέλος θα μπορούσε να γίνει χρήση κάποιου πιο ρεαλιστικού μοντέλου για το 802.11 καθώς επίσης και να προστεθεί το 802.11 infrastructure.

Βιβλιογραφία

- [1] Antoniou P., Pitsillides A. and Vassiliou V., “Adaptive Feedback Algorithm for Internet Video Streaming based on Fuzzy Rate Control”, Submitted to the IEEE Symposium on Computers and Communications (ISCC'07). Aveiro, Portugal, July 1-4, 2007
- [2] Antoniou P., Vassiliou V. and Pitsillides A., “Delivering Adaptive Scalable Video over the Wireless Internet”, in 1st ERCIM Workshop on eMobility, Coimbra Portugal, May 2007, pp.23-34
- [3] Arizona State University, Video Trace Library.[Online]
Available: <http://trace.eas.asu.edu/index.html>
- [4] Bouras Ch., Gkamas A. and Kioumourtzis G., “Adaptive smooth multicast protocol for multimedia transmission: Implementation details and performance evaluation”, in *International Journal of Communication Systems*, Wiley InterScience, Vol. 23, Issue 3, 2010, pp. 299 - 333
- [5] Evalvid in NS2.[Online].
Available: http://140.116.72.80/~smallko/ns2/Evalvid_in_NS2.htm
- [6] The Network Simulator – ns-2. [Online].
Available: <http://www.isi.edu/nsnam/ns/>
- [7] The Network Simulator Wiki–Contributed Code. [Online].
Available: <http://nsnam.isi.edu/nsnam/index.php/Contributed Code>

- [8] Perkins, C. E. and Royer, E. M.; "Ad-hoc On-Demand Distance Vector Routing", Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA.99), New Orleans, LA, USA, February 1999, pp. 90-100
- [9] Perlman R., "An algorithm for distributed computation of a spanning tree in an extended Lan: ", in *ACM Special Interest Group on Data Communications (SIGCOMM)*, 1985, pp. 44-53.
- [10] Royer, E. M. and Perkins, C. E.; "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol", Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM.99), Seattle, WA, USA, August 1999, pages 207-218.
- [11] Royer, E. M. and Perkins, C. E.; "Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing", IETF, Internet Draft: draft-ietf-manet-maodv-00.txt, 2000.
- [12] Vaishampayan R. and Garcia-Luna-Aceves J.J., "Efficient and Robust Multicast Routing in Mobile Ad Hoc Networks", in *IEEE Mobile Ad-hoc and Sensor Systems International Conference*, 2004, pp. 304-313
- [13] Xristodolou S., "Use of multicast for video transmission over MIPv6 networks", Undergraduate thesis project, University of Cyprus