

Ατομική Διπλωματική Εργασία

ΠΡΟΣΑΡΜΟΣΤΙΚΗ ΜΕΤΑΔΟΣΗ H.264 ΒΙΝΤΕΟ

Άρτεμις Ηρακλέους

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Προσαρμοστική Μετάδοση H.264 Βίντεο

Άρτεμις Ηρακλέους

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 2010

ΕΥΧΑΡΙΣΤΙΕΣ

Φτάνοντας στο τέλος των προπτυχιακών μου σπουδών και στην ολοκλήρωση αυτής της διπλωματικής εργασίας, θα ήθελα να εκφράσω τις ευχαριστίες μου στα άτομα τα οποία συνέβαλαν, ο καθένας με διαφορετικό τρόπο, στην ολοκλήρωση της.

Καταρχήν, θέλω να ευχαριστήσω τους γονείς μου και τις δύο αδελφές μου για την στήριξη, κατανόηση και συμπαράσταση που μου έδειξαν τα τελευταία τέσσερα αυτά χρόνια και κυρίως τον τελευταίο χρόνο κατά την προσπάθεια πραγματοποίησης αυτής της εργασίας.

Ακολούθως, θέλω να εκφράσω τις ευχαριστίες μου στον επιβλέπων καθηγητή αυτής της διπλωματικής εργασίας, Δρ. Βάσο Βασιλείου. Οι οδηγίες, η καθοδήγηση και η βοήθειά του από την έναρξη αυτής της διπλωματικής εργασίας, ήταν πολύ σημαντικές για την ολοκλήρωση της. Εύχομαι σε αυτόν κάθε επιτυχία και ευτυχία στην επαγγελματική και προσωπική του ζωή και πάντα υγεία.

Τέλος, θέλω να ευχαριστήσω θερμά, την υποψήφια διδάκτορα Χριστιάνα Ιωάννου, για το χρόνο, το ενδιαφέρον και τη βοήθεια που μου πρόσφερε κατά τη φάση υλοποίησης του στόχου της εργασίας αυτής, ώστε να ολοκληρωθεί. Της εύχομαι κάθε επιτυχία στην επαγγελματική της ζωή.

ΠΕΡΙΛΗΨΗ

Η επικοινωνία μεταξύ των τελικών χρηστών σε ένα δίκτυο και η μεταφορά δεδομένων χρησιμοποιώντας τα διάφορα δρομολόγια και δρομολογητές σε αυτό είναι ο στόχος των δικτύων τα οποία βασίζονται στην επικοινωνία. Από τη στιγμή που υπάρχουν πολλοί χρήστες στο δίκτυο σημαίνει πως οι διαθέσιμοι πόροι του όπως είναι για παράδειγμα το εύρος ζώνης πρέπει να μοιράζεται σε όλους με στόχο πάντα να υπάρχει σταθερή ποιότητα υπηρεσίας στον κάθε χρήστη.

Σε πολλές περιπτώσεις όμως οι διαθέσιμοι πόροι ενός δικτύου δεν παραμένουν σταθεροί αφού λόγω διάφορων αιτιών μεταβάλλονται, αλλάζοντας έτσι και τις συνθήκες που επικρατούν σε αυτό. Η αλλαγή των συνθηκών στο δίκτυο επηρεάζει και κάθε οντότητα του δικτύου ξεχωριστά, σύμφωνα με τις δυνατότητες και τους περιορισμούς που έχει η κάθε οντότητα. Πρέπει όμως να γίνεται ότι είναι δυνατό να γίνει ώστε η κάθε οντότητα στο δίκτυο να λαμβάνει όσο το δυνατό καλύτερη ποιότητα υπηρεσίας. Ειδικά σε εφαρμογές δικτύων οι οποίες μεταδίδουν βίντεο και οι οποίες είναι πλέον ευρέως χρησιμοποιημένες, αυτό είναι μια μεγάλη ανάγκη.

Έτσι, υπάρχει ανάγκη για προσαρμοστική μετάδοση προς παραλήπτες και στη συγκεκριμένη περίπτωση, ανάγκη για προσαρμοστική μετάδοση βίντεο με στόχο να παρέχεται ποιότητα υπηρεσίας σε κάθε παραλήπτη ξεχωριστά. Αυτό σημαίνει πως κατά τη μετάδοση βίντεο θα λαμβάνεται υπόψη το τι συμβαίνει στο δίκτυο και στην πλευρά του παραλήπτη ώστε σε περίπτωση που κριθεί αναγκαίο, να προσαρμοστεί η μετάδοση και να υπάρχει αποδοτικότερη λειτουργία της όλης διαδικασίας. Έτσι ο παραλήπτης θα μπορεί να λαμβάνει μια ψηλότερη ποιότητα υπηρεσίας αν υπάρχει αυτή η δυνατότητα ενώ θα λαμβάνει χαμηλότερη όταν πρέπει να ληφθούν δραστικά μέτρα λόγω απώλειας δεδομένων ή συμφόρησης στο δίκτυο.

ΠΕΡΙΕΧΟΜΕΝΑ

Κεφάλαιο 1	Εισαγωγή.....	1
Κεφάλαιο 2	Mobile IPv6.....	4
	2.1 Εισαγωγή	4
	2.2 Βασική Ορολογία Mobile IPv6	5
	2.3 Επιλογές Προορισμού Mobile IPv6	7
	2.4 Δομές Δεδομένων MIPv6	8
	2.5 Τρόπος Λειτουργίας του MIPv6	8
Κεφάλαιο 3	Multicast Μετάδοση Δεδομένων.....	15
	3.1 Εισαγωγή	15
	3.2 Τρόποι Μετάδοσης Δεδομένων	15
	3.2.1 Unicast	15
	3.2.2 Broadcast	16
	3.2.3 Multicast	17
	3.3 Multicast Μετάδοση Δεδομένων	17
	3.3.1 Εισαγωγή	17
	3.3.2 Τρόπος Λειτουργίας Multicast Μετάδοσης Δεδομένων	19
	3.4 Mobile Multicast	21
	3.4.1 Remote Subscription	22
	3.4.2 Bi-directional Tunneling	22
	3.4.3 Mobile Multicast Protocol (MOM)	22
	3.4.4 Range-Based Mobile Multicast (RBMoM)	23
Κεφάλαιο 4	Κωδικοποίηση Βίντεο.....	25
	4.1 Εισαγωγή	25
	4.2 Αρχές κωδικοποίησης βίντεο	26
	4.3 Είδη κωδικοποίησης	30
	4.3.1 Intra picture ή Intra coding	30
	4.3.2 Inter picture ή Inter coding	30
	4.4. Είδη πλαισίων κατά την κωδικοποίηση	31
	4.4.1 I- πλαίσιο	31
	4.4.2 P- πλαίσιο	32

4.4.3 B- πλαίσιο	32
4.4.4 PB- πλαίσιο	33
4.4.5 D- πλαίσιο	33
4.5 Πρότυπα κωδικοποίησης	34
4.5.1 ITU-T H.261	34
4.5.2 MPEG-1 (Part 2)	34
4.5.3 MPEG-2 (Part 2)	34
4.5.4 ITU-T H.263	35
4.5.6 H264/AVC –MPEG4 (Part 10)	36
4.5.6.1 Χαρακτηριστικά του πρότυπου H264/AVC	37
4.5.6.2 Επεκτάσεις στο πρότυπο –Fidelity Range Extensions (FRExt)	41
4.5.6.3 Επέκταση Scalable Video Coding	41
Κεφάλαιο 5 Multimedia Δεδομένα.....	44
5.1 Εισαγωγή	44
5.2 Παραδείγματα Εφαρμογών Πολυμέσων	44
5.2.1 Streaming Stored Audio και Video	44
5.2.2 Streaming Live Audio και Video	46
5.2.3 Real Time Interactive Audio και Video	47
5.3 Real Time Streaming Protocol (RTSP)	47
5.3.1 Μέθοδοι RTSP	50
5.4 Μετάδοση πολυμεσικού περιεχομένου	51
5.4.1 Εισαγωγή	51
5.4.2 Real Time Πρωτόκολλα RTP και RTCP	52
5.4.2.1 Εισαγωγή	52
5.4.2.2 Πρωτόκολλο RTP	53
5.4.2.3 Πρωτόκολλο RTCP	55
5.4.2.4 RTP Σύνοδος	59
5.5 Μετάδοση Ροών Βίντεο σε Best-Effort δίκτυα	59
5.6 Multicast Μετάδοση Πολυμεσικού Περιεχομένου	60
5.6.1 Εισαγωγή	60
5.6.2 Τρόποι Multicast Μετάδοσης Δεδομένων σε Εφαρμογές	
Πολυμέσων	61

5.6.3 Κριτήρια για την αποδοτική multicast μετάδοση πολυμεσικού περιεχομένου	62
5.7 Προσαρμοστική Multicast Μετάδοση Πολυμεσικού Περιεχομένου	63
Κεφάλαιο 6 Σύστημα ADIVIS.....	66
6.1 Εισαγωγή	66
6.2 Κωδικοποίηση σε στρώματα (Layered Encoding)	67
6.3 Μηχανισμός Ανατροφοδότησης (Feedback Mechanism)	68
6.4 Αλγόριθμος Συγκεχυμένης Απόφασης (Fuzzy Decision Algorithm)	69
Κεφάλαιο 7 Darwin Streaming Server.....	73
7.1 Εισαγωγή	73
7.2. Λειτουργία του Darwin Streaming Εξυπηρετητή	74
7.3. Programming Interface of Darwin Streaming Server	75
7.4. QTSS Roles	81
7.4.1 Register Role	81
7.4.2 Rtcp Process Role	82
7.5. QTSS Objects	83
7.5.1 qtssClientSessionObjectType	83
7.5.2 qtssRTPStreamObjectType	83
7.5.3 qtssRTSPRequestObjectType	84
Κεφάλαιο 8 Προηγούμενη Εργασία.....	86
8.1 Εισαγωγή	86
8.2 Προτεινόμενη Λύση	86
8.3 Εξαγωγή Αποτελεσμάτων	89
Κεφάλαιο 9 Προσαρμοστική Μετάδοση H.264 Βίντεο.....	90
9.1 Προτεινόμενη Λύση	90
9.2 Προτεινόμενος Αλγόριθμος	92
Κεφάλαιο 10 Προβλήματα.....	101
10.1 Προβλήματα	101
10.2 Αρχική Προσπάθεια	102
Κεφάλαιο 11 Συμπεράσματα και Μελλοντική Εργασία	105
11.1 Συμπεράσματα	105

11.2 Μελλοντική Εργασία	106
Βιβλιογραφία	108
Παράρτημα Α Σημαντικά σημεία για την εγκατάσταση και χρήση του εξυπηρετητή.....	
Παράρτημα Β Κώδικας για τον Darwin Streaming Εξυπηρετητή.....	
B.1 Νέος κώδικας	
B.2 Κομμάτια κώδικα που επεκτείνουν την ήδη υπάρχων υλοποίηση	
B.2.1 Κώδικας για μεταγλώττιση του πιο πάνω module στον εξυπηρετητή	
B.2.2 Κώδικας για έλεγχο του πεδίου Range σε μια RTSP PLAY αίτηση	
B.2.3 Κώδικας για έλεγχο για νέα μετάβαση και τερματισμό της τρέχων ροής βίντεο	
Παράρτημα Γ Οθόνη DSS –ADIVIS.....	
Παράρτημα Δ Κώδικας ADIVIS.....	

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Η συνεχής εξέλιξη της τεχνολογίας επηρεάζει και τον τομέα των δικτύων και έτσι δημιουργείται μια ανάγκη για συνεχής εξέλιξη και επέκτασή τους. Οι εφαρμογές που βασίζονται στα δίκτυα για την μετάδοση δεδομένων συνεχώς αυξάνονται. Έτσι, συμβαίνει και με τη χρήση εφαρμογών μετάδοσης βίντεο αφού υπάρχουν εφαρμογές τόσο επικοινωνίας όσο και ψυχαγωγίας, όπως για παράδειγμα η βίντεο-συνδιάσκεψη και διαδραστικά παιχνίδια.

Λόγω αυτής της ευρείας χρήσης των εφαρμογών βίντεο δίνεται ιδιαίτερο ενδιαφέρον και στις τεχνικές κωδικοποίησης του βίντεο προτού αυτό μεταδοθεί προς τον επιθυμητό παραλήπτη. Όπως θα δούμε στα επόμενα κεφάλαια, οι μηχανισμοί για την κωδικοποίηση βίντεο συνεχώς βελτιώνονται. Επιπρόσθετα, έχουν αναπτυχθεί αρκετά πρότυπα κωδικοποίησης βίντεο με στόχο η εργασία αυτή να γίνεται με όσο το δυνατό αποδοτικότερο και γρηγορότερο τρόπο.

Ο παραλήπτης του βίντεο μπορεί να το λαμβάνει μέσω streaming, δηλαδή ο παραλήπτης να μπορεί να παρακολουθήσει το βίντεο μόλις αρχίσει να λαμβάνεται από αυτόν χωρίς να περιμένει να ληφθεί ολόκληρο το αρχείο για να μπορέσει να το κάνει αυτό.

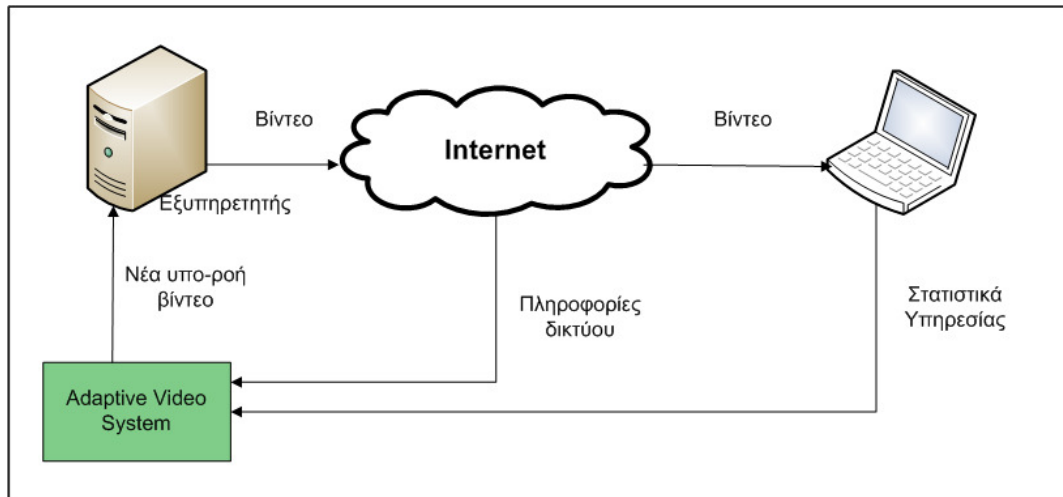
Οι εφαρμογές βίντεο έχουν την απαίτηση η μετάδοση του βίντεο να πραγματοποιείται σε πραγματικό χρόνο. Όταν όμως, το περιβάλλον στο οποίο πραγματοποιείται η μετάδοση αφορά ασύρματο ή κινητό δίκτυο τότε τα πράγματα αποτελούν μια ακόμα μεγαλύτερη πρόκληση. Αυτό συμβαίνει γιατί σε ένα τέτοιο περιβάλλον μπορεί να συμβαίνουν απρόβλεπτες αλλαγές σε ότι αφορά το εύρος ζώνης, τις καθυστερήσεις και την απώλεια δεδομένων, γεγονός που σίγουρα επηρεάζει τις ροές βίντεο που μεταδίδονται. Επιπρόσθετα, καθώς ο μεγάλος αριθμός χρηστών που έχουν πρόσβαση σε τέτοιες εφαρμογές αυξάνεται, αυξάνονται και οι ανάγκες για το βίντεο streaming. Ο λόγος που αυξάνεται ο αριθμός των χρηστών είναι λόγω της ύπαρξης διαφόρων ειδών συσκευών οι οποίες υποστηρίζουν τις εφαρμογές βίντεο όπως είναι οι ηλεκτρονικοί υπολογιστές και τα κινητά τηλέφωνα.

Επίσης, οι εφαρμογές που χρησιμοποιούν ροές βίντεο έχουν συγκεκριμένες απαιτήσεις όσον αφορά το εύρος ζώνης, τις καθυστερήσεις και το jitter. Έτσι, θα πρέπει να λαμβάνεται υπόψη η ποιότητα υπηρεσίας (Quality of Service (QoS)) που θα πρέπει να παρέχεται, αφού οι υπηρεσίες ενός δικτύου πρέπει να παρέχονται σε όσες πιο πολλές οντότητες.

Αυτά οδηγούν στην ανάγκη για την ύπαρξη ενός αποδοτικού μηχανισμού ο οποίος με κάποιον τρόπο θα μπορεί να προσαρμόζει τη ροή βίντεο που μεταδίδεται σε ένα παραλήπτη. Με αυτόν τον τρόπο θα λαμβάνονται υπόψη οι συνθήκες που επικρατούν στο δίκτυο όπως επίσης και οι απαιτήσεις και οι περιορισμοί οι οποίοι ορίζονται από αυτό, αλλά και της ίδιας της εφαρμογής και του παραλήπτη. Έτσι οι εφαρμογές θα μπορούν να προσαρμόζουν τη λειτουργία τους με βάση τα πιο πάνω χαρακτηριστικά.

Στόχος αυτής της διπλωματικής εργασίας είναι η δημιουργία ενός συστήματος το οποίο θα προσαρμόζει την μετάδοση H.264 βίντεο και το οποίο θα λειτουργεί σε συνεργασία με έναν εξυπηρετητή (server). Πιο συγκεκριμένα θα πρέπει να επιλεγεί ένας εξυπηρετητής ο οποίος θα υποστηρίζει βίντεο streaming ώστε να μπορεί να δέχεται τις αιτήσεις ενός χρήστη για κάποιο αρχείο βίντεο και να αποκρίνεται σε αυτές, αποστέλλοντας τη συγκεκριμένη ροή βίντεο. Κατά τη διάρκεια της διαδικασίας streaming θα είναι υπό λειτουργία το σύστημα προσαρμοστικής μετάδοσης βίντεο, το οποίο θα λαμβάνει πληροφορίες από τον πυρήνα του δικτύου για τις συνθήκες που επικρατούν σε αυτό, όπως επίσης και στατιστικά υπηρεσίας από τον παραλήπτη. Με βάση αυτά θα πρέπει να λαμβάνει μια απόφαση κατά πόσο μπορεί να αποσταλεί μια υπό-ροή ψηλότερης ποιότητας στον παραλήπτη ή εάν πρέπει να αποσταλεί μια υπό-ροή χαμηλότερης ποιότητας. Με βάση αυτήν την απόφαση, ο εξυπηρετητής θα αποστέλλει στη συνέχεια την νέα υπό-ροή σε περίπτωση που έτσι κρίθηκε ή θα συνεχίσει με την μετάδοση της αρχικής υπό-ροής.

Στο επόμενο σχήμα παρουσιάζεται η πιο πάνω λειτουργία και στόχος αυτής της διπλωματικής εργασίας.



Σχήμα 1.1 Σχηματική αναπαράσταση ιδανικής λειτουργίας

ΚΕΦΑΛΑΙΟ 2

Mobile IPv6

2.1. Εισαγωγή

2.2. Βασική Ορολογία Mobile IPv6

2.3. Επιλογές Προορισμού Mobile IPv6

2.4. Δομές Δεδομένων MIPv6

2.5. Τρόπος Λειτουργίας του MIPv6

2.1 Εισαγωγή

Το IP είναι η βάση για πάρα πολλές εφαρμογές και η λειτουργία πολλών τύπων δικτύου βασίζεται σε αυτό. Είναι το πρωτόκολλο το οποίο παρέχει δρομολόγηση πακέτων και παροχή υπηρεσιών που σχετίζονται με το διαδίκτυο. Αυτός είναι και ο λόγος για την ανάγκη ύπαρξης της υποστήριξης της κινητικότητας (mobility) των χρηστών στο επίπεδο του IP. Όταν ένας χρήστης είναι κινητός τότε αυτό σημαίνει πως μπορεί να μετακινείται από το ένα δίκτυο σε ένα άλλο. Με την πρόοδο της τεχνολογίας ο αριθμός των κινητών συσκευών αυξάνεται συνεχώς. Η δρομολόγηση πακέτων με βάση το πρωτόκολλο IP βασίζεται στην IP διεύθυνση που έχει κάθε κόμβος και της οποίας το πρόθεμα δικτύου καθορίζει το φυσικό υποδίκτυο στο οποίο ανήκει αυτός ο κόμβος. Χωρίς την ύπαρξη υποστήριξης της κινητικότητας, η αλλαγή του φυσικού υποδικτύου από το χρήστη θα απαιτούσε την αλλαγή της IP διεύθυνσης κάθε φορά που αυτός μετακινείται. Έτσι για να υπάρχει η υποστήριξη κινητικότητας σε ένα δίκτυο σχεδιάστηκε το Mobile IP. Αρχικά αναπτύχθηκε το Mobile IPv4 ενώ στη συνέχεια επεκτάθηκε στο Mobile IPv6 στο οποίο έχουν πραγματοποιηθεί κάποιες βελτιώσεις όσον αφορά την εμπλοκή του ίδιου του κινητού χρήστη στη διαχείριση της κινητικότητάς του. Το Mobile IP επιτρέπει έτσι στους κινητούς κόμβους του δικτύου να συνεχίσουν να λαμβάνουν όσα πακέτα προορίζονται για αυτούς, ανεξάρτητα αν αλλάζει το σημείο σύνδεσής τους στο δίκτυο. Έτσι οι συνδέσεις τις οποίες έχει ένας

κινητός κόμβος με κάποιον άλλον κόμβο θα μπορούν να διατηρούνται καθώς μετακινείται σε διαφορετικά δίκτυα. Το Mobile IPv6 δίνει τη δυνατότητα σε κάθε κόμβο IPv6 να μάθει και να αποθηκεύσει την τρέχουσα διεύθυνση (CoA) με βάση το παρών δίκτυο στο οποίο βρίσκεται ένας κινητός κόμβος. Αυτή η διεύθυνση σχετίζεται με την μόνιμη διεύθυνση του κινητού αυτού κόμβου (home address). Με αυτόν τον τρόπο τα πακέτα τα οποία προορίζονται για τον κινητό κόμβο θα προωθούνται απευθείας στην τρέχουσα διεύθυνσή του χρησιμοποιώντας IPv6 Routing Header. Όλες οι IPv6 διευθύνσεις έχουν μέγεθος 128 bits σε αντίθεση με τις διευθύνσεις οι οποίες χρησιμοποιούνταν στο IPv4 και είχαν μέγεθος 32 bits.

2.2 Βασική Ορολογία Mobile IPv6 [1],[3]

Στη συνέχεια παρουσιάζονται κάποιοι όροι οι οποίοι συνδέονται με τη λειτουργία του Mobile IPv6.

Mobile Node (MN) : Οποιοσδήποτε κόμβος στο δίκτυο, του οποίου αλλάζει το σημείο σύνδεσης του στο δίκτυο κατά το πέρασμα του χρόνου χωρίς όμως να αλλάζει η home address του.

Home Address : Η μόνιμη IP διεύθυνση η οποία ανατίθεται σε μια κινητή συσκευή και η οποία παραμένει η ίδια ανεξάρτητα από την τοποθεσία στην οποία βρίσκεται η κινητή συσκευή την παρούσα στιγμή.

Home Network (HN) : Το δίκτυο στο οποίο βρισκόταν η κινητή συσκευή όταν της έγινε η ανάθεση της Home Address της.

Foreign Network (FN): Το δίκτυο στο οποίο μια κινητή συσκευή μπορεί να ενωθεί όταν βρίσκεται μακριά από το Home network της.

Care of Address (CoA) : Μια προσωρινή διεύθυνση της κινητής συσκευής καθώς βρίσκεται μακριά από το Home network της και η οποία αποκτάται από την κινητή συσκευή μέσω IPv6 address auto configuration στο Foreign Network στο οποίο βρίσκεται την παρούσα στιγμή. Η διεύθυνση αυτή χρησιμοποιείται έτσι ώστε η κινητή συσκευή να μπορεί να λαμβάνει τα πακέτα που προορίζονται για αυτήν όταν δε βρίσκεται στο Home network της. Το πρόθεμα υποδικτύου της διεύθυνσης αυτής είναι το ίδιο με το πρόθεμα του ξένου δικτύου στο οποίο βρίσκεται η κινητή συσκευή.

Δέσιμο (Binding) : Η συσχέτιση της Home Address της κινητής συσκευής με την Care of Address της, μαζί με την εναπομείναντα διάρκεια ζωής της συσχέτισης αυτής.

Correspondent Node (CN) : Οποιοσδήποτε κόμβος ο οποίος επικοινωνεί με έναν κινητό κόμβο στο δίκτυο και μπορεί να είναι και αυτός κινητός ή μπορεί να είναι σταθερός κόμβος .

Home Agent (HA) : Ένας δρομολογητής στο Home Network του κινητού κόμβου. Διατηρεί πληροφορίες για το παρόν binding του κινητού κόμβου και λαμβάνει τα πακέτα τα οποία προορίζονται για τον κινητό κόμβο. Ακολουθώντας τα κάνει tunnel προς τον κινητό κόμβο στην παρούσα CoA του χρησιμοποιώντας IPv6 encapsulation.

Home Registration : Η εγγραφή ενός κινητού κόμβου σε μια CoA στην Binding Cache μνήμη του Home Agent. Έχει συγκεκριμένη διάρκεια ζωής.

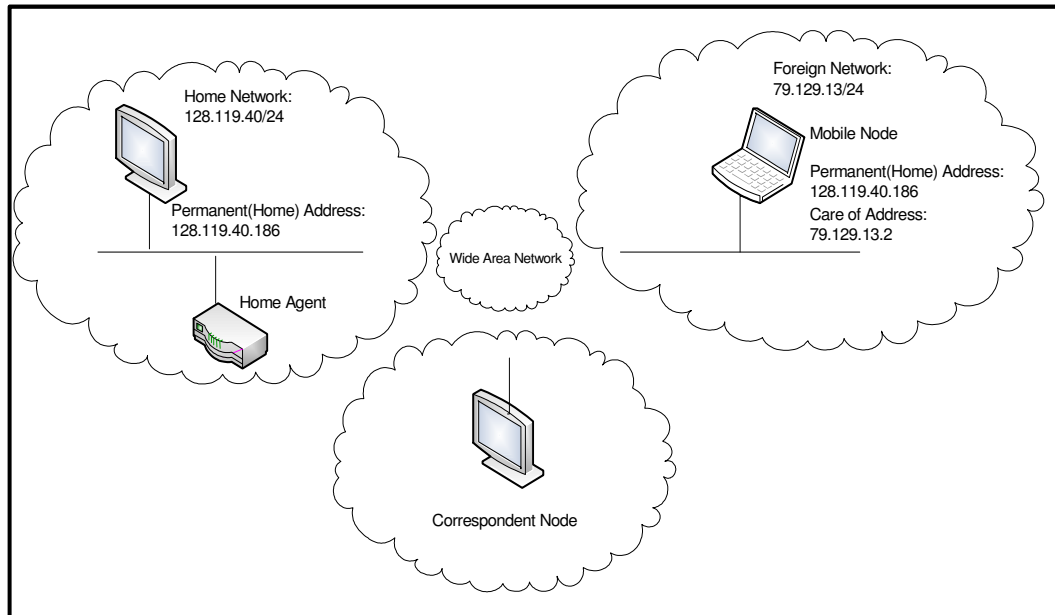
Home Link : Η σύνδεση η οποία καθορίζει το home subnet prefix.

Home Subnet Prefix : Το πρόθεμα δικτύου στην home διεύθυνση του κινητού χρήστη.

Foreign Link : Οποιαδήποτε άλλη σύνδεση διαφορετική από τη home link και η οποία καθορίζει το foreign subnet prefix.

Foreign Subnet Prefix : Οποιοδήποτε άλλο πρόθεμα δικτύου διαφορετικό από το home subnet prefix και το οποίο συνθέτει την CoA του κινητού χρήστη.

Στο επόμενο σχήμα παρουσιάζονται οι βασικές οντότητες στην τοπολογία ενός δικτύου στο οποίο υπάρχει ένας κινητός κόμβος.



Σχήμα 2.1: Τοπολογία ενός κινητού δικτύου

2.3 Επιλογές Προορισμού Mobile IPv6 [3]

Το Mobile IPv6 χρησιμοποιεί τρεις επιλογές προορισμού ώστε να δώσει τη δυνατότητα σε home agents και correspondent κόμβους να ενημερωθούν και να αποθηκεύσουν πληροφορίες που σχετίζονται με έναν κινητό κόμβο και την CoA με την οποία συσχετίζεται. Οι επιλογές αυτές είναι οι ακόλουθες :

Binding Update : Ένας κινητός κόμβος στέλλει ένα Binding Update στον home agent του και σε όποιον correspondent κόμβο με τον οποίο επικοινωνεί και ο οποίος μπορεί να γνωρίζει μια παλιά CoA του κινητού αυτού κόμβου. Το Binding Update περιέχει την παρούσα CoA του για το δίκτυο στο οποίο βρίσκεται την παρούσα στιγμή.

Binding Acknowledgement : Χρησιμοποιείται για την επιβεβαίωση της λήψης ενός Binding Update από κάποιον κινητό κόμβο εάν ζητηθεί από αυτόν τον κινητό κόμβο. Συνήθως τις περισσότερες φορές στέλλεται από έναν home agent παρά από έναν correspondent κόμβο.

Binding Request : Χρησιμοποιείται από κάποιον κόμβο όταν επιθυμεί να λάβει πληροφορίες για έναν κινητό κόμβο και την CoA με την οποία συσχετίζεται τη συγκεκριμένη στιγμή μέσω ενός Binding Update.

Οι επιλογές προορισμού Mobile IPv6 μπορούν να αποσταλούν είτε μέσα σε ένα IPv6 πακέτο το οποίο περιέχει κάποιο payload είτε ως ένα ξεχωριστό IPv6 πακέτο. Κάθε IPv6 κόμβος πρέπει να μπορεί να επεξεργάζεται τα Binding Updates τα οποία λαμβάνει και να μπορεί να στέλλει Binding Acknowledgements και Binding Requests.

2.4 Δομές Δεδομένων MIPv6 [3]

Το Mobile IPv6 καθορίζει τις εξής δομές δεδομένων για τη στήριξη της λειτουργίας του:

Binding Cache : Είναι η κεντρική δομή δεδομένων στο Mobile IPv6 και είναι μια μνήμη η οποία αποτελείται από τα bindings κινητών κόμβων και διατηρείται από κάθε κινητό κόμβο.

Binding Λίστα Ενημέρωσης : Αυτή η λίστα διατηρείται από κάθε κινητό κόμβο και περιέχει όλα τα Binding Updates τα οποία έχουν σταλεί από αυτόν στον home agent του και σε όποιους correspondent κόμβους με τους οποίους επικοινωνεί. Όταν η διάρκεια ζωής αυτών των εγγραφών λήξει τότε οι εγγραφές αυτές διαγράφονται.

Λίστα Home Agents : Αυτή η λίστα χρησιμοποιείται από έναν κινητό κόμβο όταν κάνει home agent discovery για να συνδεθεί με ένα δρομολογητή ο οποίος θα του παρέχει τις υπηρεσίες του ως home agent. Επίσης σε αυτόν θα πραγματοποιήσει την εγγραφή της CoA του έτσι ώστε να μπορεί να λαμβάνει τα πακέτα τα οποία προορίζονται για αυτόν. Κάθε home agent διατηρεί μια τέτοια λίστα η οποία του παρέχει πληροφορίες για τους άλλους home agents που βρίσκονται στην ίδια σύνδεση. Οι πληροφορίες αυτές προέρχονται από τις διαφημίσεις τις οποίες στέλλουν οι δρομολογητές.

2.5 Τρόπος Λειτουργίας του MIPv6 [3]

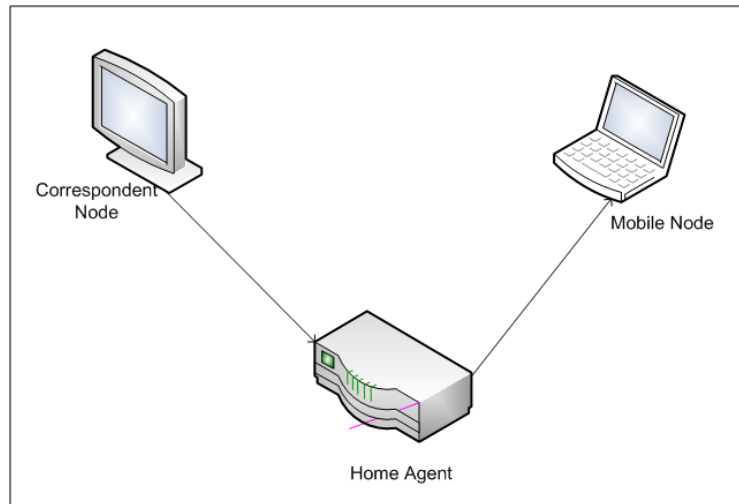
Αποστολή και Δρομολόγηση Πακέτων

Όπως έχει αναφερθεί πιο πάνω, ο home agent είναι υπεύθυνος για να προωθεί σε έναν κινητό χρήστη ο οποίος ανήκει στο home δίκτυό του αλλά τη συγκεκριμένη στιγμή βρίσκεται μακριά από αυτό, τα πακέτα που προορίζονται για αυτόν, αφού γνωρίζει την CoA του. Για να μπορεί να το πράξει αυτό πρέπει να στέλνει διαφημίσεις

στο home link του εκ μέρους του κινητού κόμβου, ώστε οι υπόλοιποι κόμβοι να γνωρίζουν ότι ο συγκεκριμένος κινητός κόμβος εξυπηρετείται από το συγκεκριμένο home agent. Αυτό επιτυγχάνεται με τη χρήση του Neighbourur Discovery Protocol.

Όταν ένας correspondent κόμβος επιθυμεί να στείλει ένα πακέτο με προορισμό έναν κινητό χρήστη τότε θα ψάξει πρώτα στην Binding Cache του για να δει εάν υπάρχει κάποια εγγραφή σε αυτήν η οποία να σχετίζεται με τον προορισμό. Αν υπάρχει τέτοια εγγραφή τότε το πακέτο με προορισμό τον κινητό αυτό χρήστη θα δρομολογηθεί απευθείας προς σε αυτόν χρησιμοποιώντας την CoA του και IPv6 Routing Header. Αν δε βρεθεί εγγραφή σχετικά με τον προορισμό τότε το πακέτο θα δρομολογηθεί στη home address του κινητού χρήστη όπου θα το παραλάβει ο home agent του και θα το κάνει tunnel στην CoA του. Για να πραγματοποιηθεί η αποστολή του πακέτου προς τον κινητό κόμβο μέσω του home agent, ο home agent χρησιμοποιεί IPv6 encapsulation. Αυτό σημαίνει πως ο κινητός κόμβος θα πρέπει να κάνει decapsulation όταν λάβει το πακέτο από τον home agent ώστε να έχει το αρχικό πακέτο. Η χρήση της Binding Cache οδηγεί στη βέλτιστη δρομολόγηση των πακέτων γιατί τα πακέτα στέλλονται απευθείας στον προορισμό τους μέσω της CoA με αποτέλεσμα να λύνεται το πρόβλημα της Τριγωνικής Δρομολόγησης, το οποίο περιγράφεται στη συνέχεια. Επίσης περιορίζεται η συμφόρηση που θα μπορούσε να προκληθεί στον home agent.

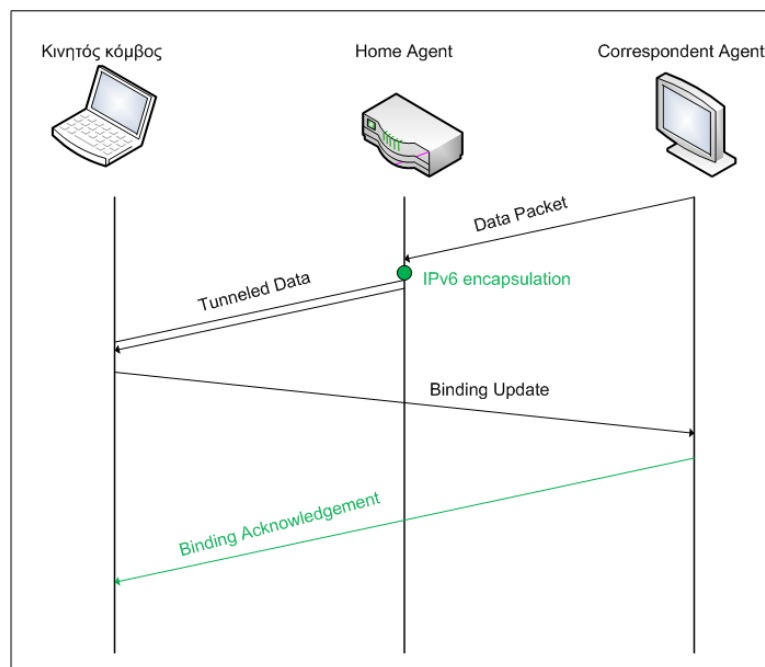
Η Τριγωνική Δρομολόγηση πραγματοποιείται όταν τα πακέτα τα οποία προορίζονται για έναν κινητό χρήστη, ο οποίος δε βρίσκεται στο home δίκτυο του, στέλλονται πρώτα στην home address του από όπου τα παραλαμβάνει ο home agent και μετά τα προωθεί προς τον κινητό χρήστη χρησιμοποιώντας την CoA του. Έτσι η αποστολή των πακέτων στον κινητό χρήστη πραγματοποιείται μέσω του home agent και όχι απευθείας σε αυτόν. Όμως αυτό δεν είναι και τόσο αποδοτικό διότι μπορεί να παρουσιαστούν διάφορα προβλήματα όπως υπερβολικό φόρτο εργασίας στο δίκτυο και καθυστέρηση στην αποστολή και λήψη των πακέτων. Επιπρόσθετα εάν συμβεί κάτι στον home agent και είναι ο μόνος ο οποίος εξυπηρετεί το συγκεκριμένο home δίκτυο με αυτόν τον ρόλο τότε θα υπάρχει σοβαρό πρόβλημα. Στο επόμενο σχήμα παρουσιάζεται η τριγωνική δρομολόγηση πακέτων.



Σχήμα 2.2: Τριγωνική Δρομολόγηση Πακέτων

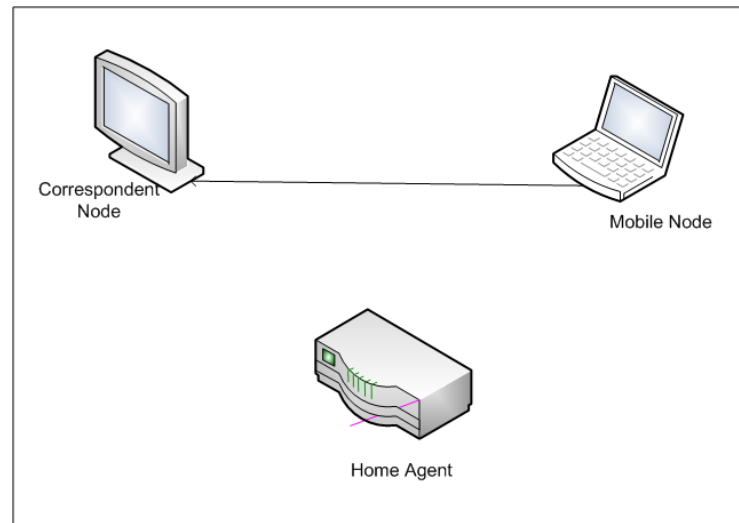
Ο κινητός κόμβος μπορεί να ενημερώσει τον correspondent κόμβο για την CoA του έτσι ώστε την επόμενη φορά που θα του στείλει κάποιο πακέτο να το δρομολογήσει απευθείας στον κινητό κόμβο.

Στο πιο κάτω σχήμα παρουσιάζεται η τριγωνική δρομολόγηση πακέτου προς έναν κινητό χρήστη και στη συνέχεια η ενημέρωσή του για την CoA του κινητού χρήστη. Ο correspondent κόμβος απαντά πίσω στον κινητό χρήστη με ένα Binding Acknowledgment σε περίπτωση που έχει ζητηθεί από τον κινητό χρήστη.



Σχήμα 2.3: Τριγωνική Δρομολόγηση – Binding Update

Όταν ένας κινητός κόμβος επιθυμεί να στείλει ένα πακέτο σε έναν correspondent κόμβο τότε το στέλλει κανονικά και απευθείας σε αυτόν. Το πακέτο θα έχει διεύθυνση αποστολέα την CoA του κινητού χρήστη ώστε να αποθηκευτεί στην Binding Cache του correspondent κόμβου. Το ακόλουθο σχήμα παρουσιάζει τη δρομολόγηση πακέτων από τον κινητό κόμβο προς τον correspondent κόμβο.



Σχήμα 2.4: Απευθείας Δρομολόγηση Πακέτων προς Correspondent

Διαχείριση Binding Μηνυμάτων

Όταν ένας κινητός κόμβος φτάσει σε ένα νέο δίκτυο και αποκτήσει μια νέα CoA πρέπει να ενημερώσει τον home agent του και όποιον correspondent κόμβο με τον οποίο επικοινωνεί μαζί του και μπορεί να γνωρίζει μια παλιά CoA για αυτόν, για την μετακίνησή του και την απόκτηση της νέας CoA. Έτσι θα στείλει Binding Update σε αυτούς, το οποίο θα περιέχει τη νέα CoA του.

Αν η CoA είναι η ίδια με την home address του τότε αυτό σημαίνει πως ο κινητός κόμβος έχει επιστρέψει στο home δίκτυό του και έτσι οποιαδήποτε εγγραφή υπάρχει σε αυτόν στη Binding Cache του home agent και των correspondent κόμβων θα πρέπει να διαγραφεί.

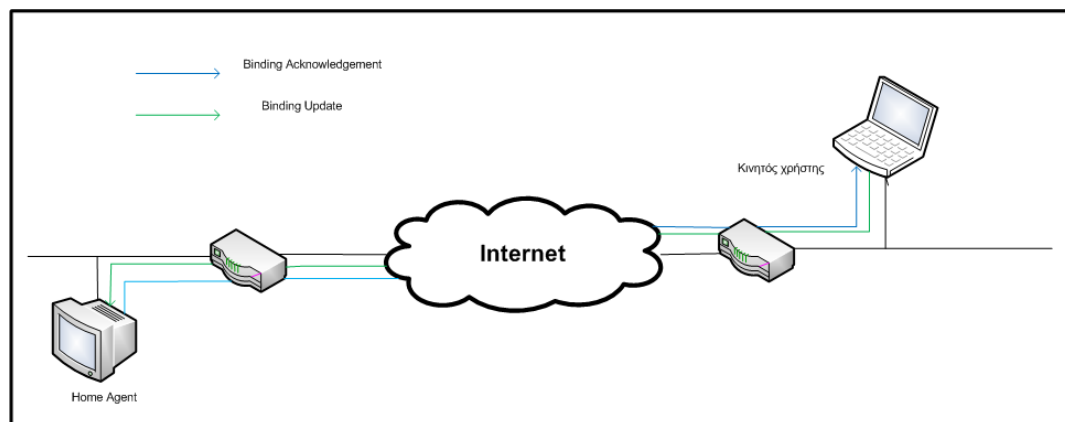
Ο κινητός χρήστης ζητά επιβεβαίωση της λήψης των Binding Updates των οποίων στέλλει ενεργοποιώντας το Acknowledge bit (A) στο Binding Update. Αν δεν

λάβει Binding Acknowledgement μέσα σε κάποιο συγκεκριμένο χρονικό διάστημα μετά την αποστολή του Binding Update τότε θα ξαναστείλει το συγκεκριμένο Binding Update. Όταν ο κινητός κόμβος στέλλει το Binding Update στον home agent πρέπει να ενεργοποιεί το Acknowledge bit (A) ενώ όταν το Binding Update προορίζεται για έναν correspondent κόμβο το κάνει μόνο αν το επιθυμεί. Όταν ένας κόμβος, είτε ένας home agent είτε ένας correspondent κόμβος λάβει ένα Binding Update το οποίο απευθύνεται σε αυτόν και το Acknowledge bit είναι ενεργοποιημένο σε αυτό, τότε θα πρέπει να στείλει πίσω στον κόμβο ο οποίος ήταν ο αποστολέας του, ένα Binding Acknowledgement.

Ένας correspondent κόμβος μπορεί να στείλει ένα Binding Request στον κινητό κόμβο ώστε να ανανεώσει την εγγραφή την οποία διατηρεί για αυτόν προτού να τερματιστεί η διάρκεια ζωής της.

Εγγραφή Care of Address στον Home Agent

Ο κινητός κόμβος πρέπει να ενημερώνει τον home agent του κάθε φορά που μετακινείται από ένα δίκτυο σε άλλο ώστε ο home agent να γνωρίζει την νέα CoA του κινητού κόμβου και να μπορεί να του στέλλει τυχόν πακέτα που απευθύνονται σε αυτόν. Αυτό πραγματοποιείται με την αποστολή Binding Update προς τον home agent με ενεργοποιημένο το Home Agent (H) bit ώστε να υπάρχει η ένδειξη ότι πρόκειται για εγγραφή της CoA στον Home Agent. Επιπρόσθετα όπως αναφέρθηκε πιο πριν, πρέπει να είναι ενεργοποιημένο το Acknowledgement bit (A) ώστε ο home agent να επιβεβαιώσει την λήψη του Binding Update.



Σχήμα 2.5: Εγγραφή Care of Address στον Home Agent

Ένας κινητός κόμβος μπορεί να ανακαλύψει τη διεύθυνση του home agent του χρησιμοποιώντας την IPv6 anycast διεύθυνση. Πιο συγκεκριμένα, ο κινητός χρήστης στέλλει ένα Binding Update στην IPv6 anycast διεύθυνση στο home δίκτυό του. Το Binding Update θα ληφθεί μόνο από έναν δρομολογητή στο δίκτυο αυτό, ο οποίος θα απορρίψει το Binding Update. Στη συνέχεια θα τοποθετήσει την IPv6 διεύθυνσή του στο Binding Update ώστε στο τέλος ο κινητός κόμβος να ξαναστείλει το Binding Update απευθείας σε αυτόν.

Ανίχνευση Κίνησης

Ένας κινητός χρήστης χρησιμοποιεί Router Discovery για να ανακαλύπτει νέους δρομολογητές. Έτσι λαμβάνει από δρομολογητές Router Advertisement μηνύματα και με βάση αυτά δημιουργεί μια λίστα με όλους τους δρομολογητές από τους οποίους λαμβάνει μήνυμα και μια λίστα με τα προθέματα των δικτύων τα οποία είναι διαθέσιμα την συγκεκριμένη στιγμή.

Όταν ο κινητός χρήστης βρίσκεται σε ένα ξένο δίκτυο, μακριά από το home δίκτυό του πρέπει να επιλέξει ένα δρομολογητή ο οποίος θα είναι ο default δρομολογητής για αυτόν και ένα πρόθεμα δικτύου από αυτά που έγιναν γνωστά από το συγκεκριμένο δρομολογητή ώστε να αποκτήσει την CoA του. Καθώς ο κινητός χρήστης χρησιμοποιεί κάποιο δρομολογητή ως τον default του πρέπει να είναι σε θέση να καταλάβει αν σε κάποια στιγμή ο πιο πάνω δρομολογητής δε θα μπορεί να του προσφέρει τις υπηρεσίες του λόγω του ότι ο ίδιος μετακινήθηκε σε μια διαφορετική σύνδεση.

Αυτό επιτυγχάνεται με τα περιοδικά Router Advertisement μηνύματα που φτάνουν στον κινητό κόμβο. Αν ο κινητός κόμβος σταματήσει να λαμβάνει διαφήμιση από το δρομολογητή που λειτουργούσε ως ο default του τότε αυτό σημαίνει πως αυτός ο δρομολογητής δεν είναι πλέον διαθέσιμος. Έτσι θα συνδεθεί με κάποιο άλλο δρομολογητή και θα αποκτήσει μια νέα CoA. Στη συνέχεια θα πρέπει να ενημερώσει τον home agent και όποιον correspondent κόμβο χρειάζεται για τη νέα του CoA αποστέλλοντάς τους Binding Update. Εναλλακτικά ο κινητός κόμβος μπορεί να στείλει Neighbor Solicitation μηνύματα στον δρομολογητή του για να ανακαλύψει ότι συνεχίζει να λειτουργεί σωστά. Αν λάβει απάντηση υπό την μορφή Neighbor

Advertisement μηνυμάτων σημαίνει πως δεν υπάρχει κάποιο πρόβλημα με τον συγκεκριμένο δρομολογητή.

IPv6 κινητός κόμβος

Ένα IPv6 κινητός κόμβος πρέπει να είναι ικανός να στέλλει Binding Updates καθώς επίσης και να λαμβάνει Binding Acknowledgements. Πρέπει επίσης να γνωρίζει πότε είναι αναγκαία η αποστολή των Binding Updates και ποιοι είναι οι κόμβοι οι οποίοι πρέπει να τα λάβουν. Τέλος πρέπει να μπορεί να διενεργεί IPv6 decapsulation σε περίπτωση που λάβει κάποιο πακέτο από τον home agent του.

IPv6 κόμβος

Ένας IPv6 κόμβος πρέπει να είναι ικανός να διατηρεί την Binding Cache του, να λαμβάνει Binding Updates και να ανανεώνει με βάση αυτά την Binding Cache του. Τέλος πρέπει να μπορεί να στέλλει Binding Request και Binding Acknowledgement όταν αυτό χρειαστεί.

ΚΕΦΑΛΑΙΟ 3

Multicast Μετάδοση Δεδομένων

3.1. Εισαγωγή

3.2. Τρόποι μετάδοσης δεδομένων

3.3. Multicast

3.4. Mobile Multicast

3.1 Εισαγωγή

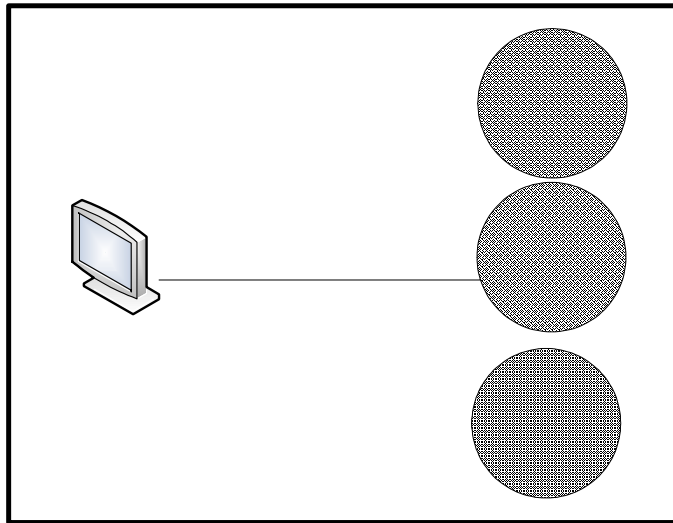
Οι εφαρμογές οι οποίες βασίζονται σε δίκτυα υπολογιστών έχουν διαφορετικές απαιτήσεις η μια από την άλλη και στο θέμα το οποίο αφορά τον αριθμό των παραληπτών και γενικότερα τους παραλήπτες των μεταδιδόμενων δεδομένων. Πιο συγκεκριμένα μπορεί να υπάρχει απαίτηση για παράδοση των δεδομένων σε έναν μόνο χρήστη ή σε περισσότερους χρήστες. Έτσι, υπάρχουν διάφοροι τρόποι για την μετάδοση δεδομένων, οι οποίοι θα περιγραφούν στη συνέχεια.

3.2 Τρόποι Μετάδοσης Δεδομένων [1]

3.2.1 Unicast

Αφορά την επικοινωνία ένα προς ένα. Ένας κόμβος ο οποίος λειτουργεί ως η πηγή αποστέλλει σε έναν μόνο άλλον κόμβο ένα πακέτο. Σε αυτήν την περίπτωση ο προορισμός αναγνωρίζεται από την IP διεύθυνσή του η οποία μεταφέρεται μέσα σε κάθε IP unicast πακέτο το οποίο προορίζεται για αυτόν.

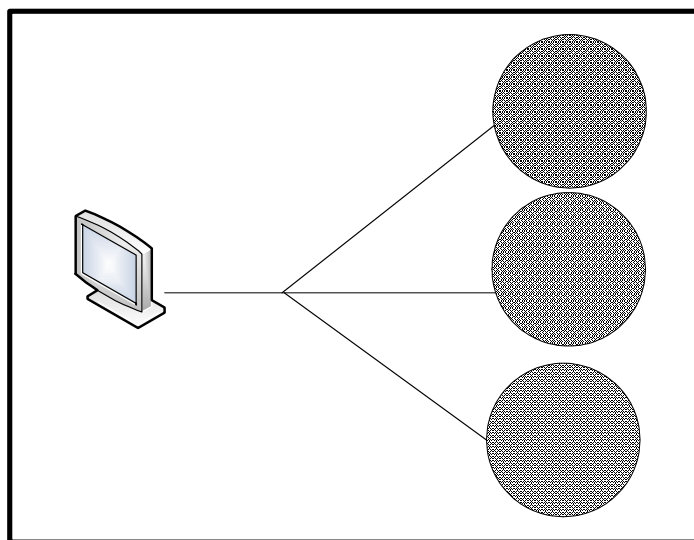
Στην περίπτωση που έχουμε streaming πολυμεσικού περιεχομένου αυτό σημαίνει πως θα δημιουργηθούν πολλές ένα προς ένα συνδέσεις μεταξύ κάθε client και server. Η σύνδεση όμως πολλών clients μέσω unicast μετάδοσης σε μια ροή σε ένα τοπικό δίκτυο μπορεί να οδηγήσει σε μεγάλη κίνηση του δικτύου.



Σχήμα 3.1: Unicast Μετάδοση Δεδομένων

3.2.2 Broadcast

Αφορά την επικοινωνία ένα προς πολλούς. Ένα κόμβος ο οποίος λειτουργεί ως η πηγή αποστέλλει ένα πακέτο σε όλους τους κόμβους που υπάρχουν μέσα στο δίκτυο στο οποίο βρίσκεται. Λόγω του ότι όλοι οι κόμβοι του δικτύου λαμβάνουν το πακέτο δε χρειάζεται αναγνώριση των προορισμών και έτσι δε χρησιμοποιείται η IP διεύθυνση του κάθε προορισμού.

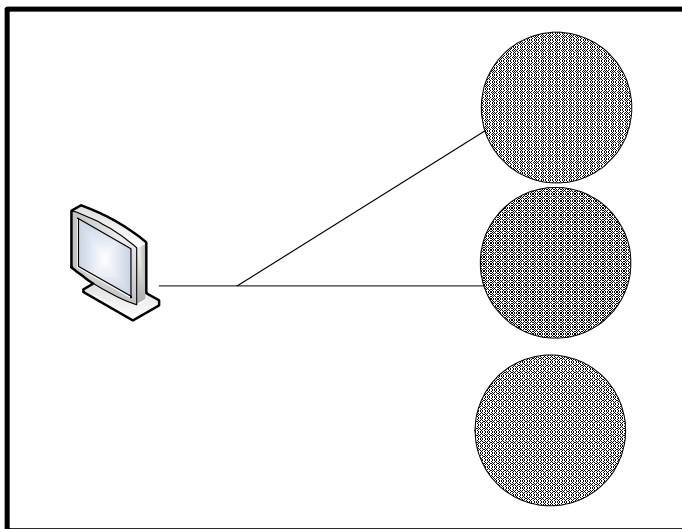


Σχήμα 3.2: Broadcast Μετάδοση Δεδομένων

3.2.3 Multicast

Αφορά την επικοινωνία ένα προς πολλούς. Σε αυτήν την περίπτωση όμως ο κόμβος ο οποίος λειτουργεί ως η πηγή αποστέλλει ένα πακέτο σε ένα υποσύνολο μόνο των κόμβων οι οποίοι βρίσκονται στο δίκτυο. Η πηγή δεν αποστέλλει ένα πακέτο σε κάθε παραλήπτη ξεχωριστά αλλά η δρομολόγηση του πακέτου από την πηγή προς τους προορισμούς γίνεται με την μεταφορά του πακέτου μόνο μια φορά σε κάθε γραμμή σύνδεσης. Αυτό προϋποθέτει την ύπαρξη κοινών γραμμών στα μονοπάτια από την πηγή σε κάθε προορισμό.

Όταν πρόκειται για streaming πολυμεσικού περιεχομένου τότε υπάρχει μια μόνο ροή την οποία και μοιράζονται όλοι οι clients.



Σχήμα 3.3: Multicast Μετάδοση Δεδομένων

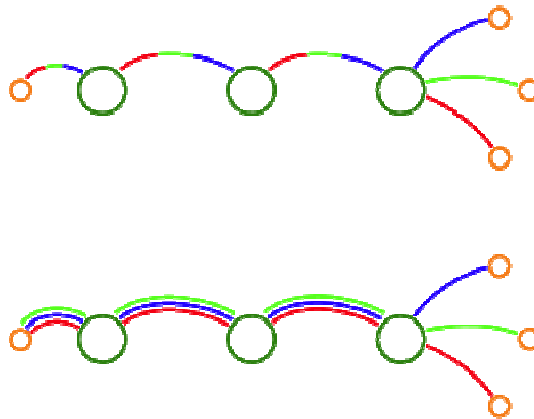
3.3 Multicast Μετάδοση Δεδομένων [1]

3.3.1 Εισαγωγή

Η συνεχής ανάπτυξη της τεχνολογίας, η εξέλιξη του διαδικτύου και η δημιουργία νέων εφαρμογών οδηγεί στην ευρεία χρήση της multicast μετάδοσης δεδομένων λόγω του ότι είναι ένας αποδοτικός τρόπος μετάδοσης δεδομένων. Αυτό οφείλεται στο γεγονός ότι

δεν δημιουργούνται και ούτε αποστέλλονται duplicate πακέτα από την πηγή στις ίδιες συνδέσεις στο δίκτυο γιατί δε δημιουργούνται ξεχωριστές ροές για την μετάδοση των δεδομένων για κάθε κόμβο-προορισμό. Έτσι σε κάθε γραμμή μεταδίδεται μόνο μια φορά το κάθε πακέτο. Αυτό οδηγεί στην αποδοτική χρήση των πόρων του δικτύου και του εύρους ζώνης του γιατί υπάρχει λιγότερο κόστος σε αυτό. Επιπρόσθετα, μειώνεται και το κόστος γενικά για όλη τη διαδικασία δρομολόγησης των πακέτων.

Στο επόμενο σχήμα παρουσιάζεται η διαφορά μεταξύ της multicast και unicast μετάδοσης δεδομένων. Πιο συγκεκριμένα, πάνω παρατηρούμε την multicast μετάδοση δεδομένων, όπου ένα πακέτο μεταδίδεται μόνο μι φορά σε κάθε γραμμή του δικτύου στο μέρος των μονοπατιών προς τους πολλαπλούς παραλήπτες τα οποία είναι κοινά. Φτάνοντας στον προτελευταίο κόμβο του δικτύου, στέλλονται τρία αντίγραφα του αρχικού πακέτου στους τρεις διαφορετικούς τελικούς παραλήπτες και τρεις διαφορετικές συνδέσεις του δικτύου. Αντίθετα στο κάτω μέρος του σχήματος παρουσιάζεται η unicast μετάδοση δεδομένων, το οποίο όπως βλέπουμε σημαίνει πως δημιουργούνται ξεχωριστές ροές για κάθε έναν παραλήπτη και επίσης σε κάθε γραμμή του δικτύου προς αυτούς μεταδίδονται τρία αντίγραφα του αρχικού πακέτου.



Σχήμα 3.4: Multicast vs. Unicast

Η ευρεία αυτή χρήση της multicast μετάδοσης δεδομένων επιβάλλει την ύπαρξη των multicast-δρομολογητών λόγω του ότι δεν είναι δυνατή η μετάδοση multicast πακέτων από όλους τους δρομολογητές σε ένα δίκτυο. Ένας multicast-δρομολογητής έχει τη δυνατότητα να κάνει encapsulate ένα multicast πακέτο σε ένα κανονικό unicast

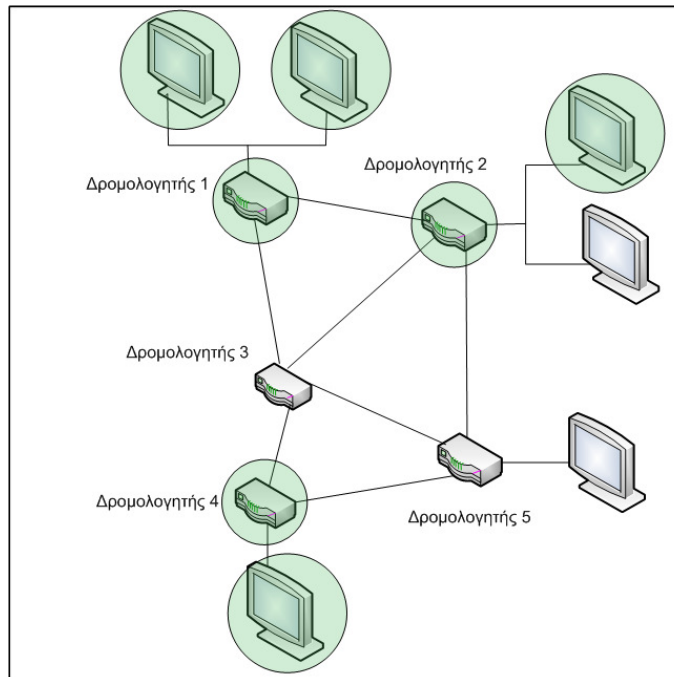
IP πακέτο ώστε αυτό να προωθείται και να λαμβάνεται από ένα δρομολογητή ο οποίος δεν υποστηρίζει multicast. Με αυτόν τον τρόπο είναι δυνατή η multicast μετάδοση δεδομένων μεταξύ δρομολογητών που δε την υποστηρίζουν. Ένας multicast δρομολογητής λαμβάνοντας ένα encapsulated πακέτο μπορεί να το κάνει decapsulate ώστε να λάβει το αρχικό multicast πακέτο. Ένα τέτοιο δίκτυο ονομάζεται Multicast Backbone (Mbone).

3.3.2 Τρόπος Λειτουργίας Multicast Μετάδοσης Δεδομένων [1], [5]

Η λειτουργία της Multicast μετάδοσης δεδομένων βασίζεται στην ύπαρξη ενός host group μοντέλου. Το host group αποτελείται από κόμβους του δικτύου οι οποίοι μοιράζονται μια κοινή multicast διεύθυνση και αντιπροσωπεύει το σύνολο των παραληπτών του multicast πακέτου. Το multicast πακέτο μεταδίδεται σε όλους τους multicast παραλήπτες οι οποίοι σχετίζονται με αυτό το group χρησιμοποιώντας αυτή τη διεύθυνση. Αυτό το σύνολο κόμβων είναι δυναμικό και οι κόμβοι έχουν τη δυνατότητα να μουν ή να αποχωρήσουν από αυτό οποιαδήποτε στιγμή επιθυμούν και να ξεκινήσουν ή να σταματήσουν να λαμβάνουν τα multicast πακέτα τα οποία στέλλονται στο group. Το δίκτυο πρέπει να έχει τους κατάλληλους μηχανισμούς ώστε να γνωρίζει ποια είναι τα μέλη από τα οποία αποτελείται το group ώστε να δρομολογεί τα multicast πακέτα σε όλα τα μέλη. Πρέπει να ανιχνεύει όλες τις τρέχουσες συμμετοχές στο multicast group ώστε να δρομολογούνται οι πληροφορίες και σε νέα μέλη αλλά και για να τερματίζεται η μετάδοση των πακέτων σε κόμβους οι οποίοι δεν ανήκουν πλέον στο multicast group.

Η δυναμική εισδοχή και έξοδος από ένα multicast group βασίζεται στο Internet Group Management Protocol (IGMP), του οποίου η λειτουργία περιγράφεται στη συνέχεια. Η παράδοση των multicast πακέτων επιτυγχάνεται με την κατασκευή ενός multicast delivery δένδρου το οποίο έχει ως φύλλα όλους τους παραλήπτες των πακέτων, δηλαδή τα μέλη του συγκεκριμένου multicast group. Το δένδρο επίσης έχει κόμβους τους multicast δρομολογητές οι οποίοι θα δρομολογήσουν τα multicast πακέτα στους τελικούς προορισμούς και οι οποίοι μπορεί να μην είναι άμεσα συνδεδεμένοι με αυτούς. Η κατασκευή του multicast delivery δένδρου βασίζεται στη χρήση αλγορίθμων για multicast δρομολόγηση.

Για παράδειγμα, παρατηρώντας το επόμενο σχήμα, ο δρομολογητής 1 και ο δρομολογητής 4 δεν μπορούν αποτελούν μέρος ενός multicast δέντρου δρομολόγησης αν δεν υπάρχει σε αυτό ένας από τους 2 δρομολογητές , 3 ή 5. Και οι δύο αυτοί δρομολογητές όμως δεν είναι άμεσα συνδεδεμένοι με έναν τελικό κόμβο προορισμό.



Σχήμα 3.5: Multicast Group

Υπάρχουν συγκεκριμένα δύο τρόποι για τον ορισμό ενός multicast δέντρου δρομολόγησης. Ο πρώτος τρόπος είναι η κατασκευή μόνο ενός τέτοιου δέντρου το οποίο θα αφορά όλους τους αποστολείς καθώς θα περιλαμβάνει όλους τους δρομολογητές οι οποίοι είναι συνδεδεμένοι με τερματικούς παραλήπτες που ανήκουν στο συγκεκριμένο multicast γκρουπ(group-shared tree). Ο δεύτερος τρόπος είναι η κατασκευή ενός δέντρου δρομολόγησης για κάθε έναν αποστολέα ξεχωριστά(source-based tree).

Internet Group Management Protocol (IGMP)

Ένας κόμβος στο δίκτυο έχει τη δυνατότητα να ενημερώσει το γειτονικό του multicast δρομολογητή για την επιθυμία του να γίνει μέλος σε ένα multicast group. Ο κόμβος στέλλει ένα IGMP join μήνυμα στο γειτονικό του multicast δρομολογητή και αν αυτός

δεν ανήκει στο group στο οποίο επιθυμεί να γίνει μέλος ο κόμβος , το μήνυμα προωθείται μέχρι να φτάσει σε κάποιον δρομολογητή ο οποίος συνδέεται στο group.

Το IGMP καθορίζει τρεις τύπους μηνυμάτων οι οποίοι περιγράφονται πιο κάτω:

Membership_query μήνυμα: Ένας δρομολογητής στέλλει σε όλους τους κόμβους με τους οποίους συνδέεται ένα membership_query message ώστε να ενημερωθεί για όλα τα multicast groups στα οποία ανήκουν οι κόμβοι αυτοί.

Membership_report μήνυμα: Χρησιμοποιείται ως απάντηση του κάθε κόμβου σε ένα membership_query message. Υπάρχει περίπτωση να γίνει αποστολή του από έναν κόμβο ο οποίος γίνεται μέλος σε ένα multicast group, χωρίς να έχει λάβει membership_query message.

Leave_group μήνυμα: Ένας κόμβος ο οποίος θέλει να αποχωρήσει από ένα multicast group στέλλει αυτό το μήνυμα ώστε να το δηλώσει και να μην λαμβάνει πλέον τις πληροφορίες που προορίζονται για αυτό το σύνολο κόμβων. Δεν είναι απαραίτητη όμως η αποστολή του από κάποιον κόμβο γιατί ένας δρομολογητής υποθέτει ότι όταν ένας κόμβος δεν απαντά σε membership_query message για συγκεκριμένο group τότε σημαίνει πως ο συγκεκριμένος κόμβος δεν ανήκει πλέον σε αυτό.

3.4 Mobile Multicast

Όπως έχουμε αναφέρει πιο πάνω, στην multicast μετάδοση δεδομένων λαμβάνεται υπόψη το multicast group το οποίο αντιπροσωπεύει τους παραλήπτες των δεδομένων και οι τρέχουσες συμμετοχές τους σε αυτό αφού αποτελούν ένα δυναμικό σύνολο. Στην mobile multicast μετάδοση δεδομένων εκτός από το δυναμικό αυτό σύνολο, δίνεται προσοχή και στην αλλαγή τοποθεσίας των παραληπτών αφού είναι κινητοί, σε άλλα δίκτυα και υπηρεσίες, γεγονός που προσθέτει πολυπλοκότητα στον καθορισμό και διατήρηση του τρόπου μετάδοσης των multicast πακέτων. Καθώς ο κινητός χρήστης μετακινείται και αλλάζει τοποθεσία, αλλάζει και η δρομολόγηση των πακέτων από και προς σε αυτόν. Η αλλαγή τοποθεσίας του χρήστη μετά την κατασκευή του multicast delivery δένδρου έχει συνέπειες και μπορεί να δημιουργήσει προβλήματα γιατί το κόστος αλλαγής του δένδρου θα είναι μεγάλο. Η IETF (Internet Engineering Task Force) έχει προτείνει δύο προσεγγίσεις ώστε να είναι δυνατή η multicast

μετάδοση πακέτων σε κινητούς χρήστες, το remote subscription και το bi-directional tunneling, των οποίων η λειτουργία θα περιγραφεί στη συνέχεια.

3.4.1 Remote Subscription [4], [5]

Ο κινητός χρήστης πρέπει να προεγγράφεται ξανά ,δηλαδή να κάνει subscribe στο multicast group στο οποίο ανήκει κάθε φορά που μετακινείται σε κάποιο άλλο νέο foreign δίκτυο. Ο αλγόριθμος αυτός όμως μπορεί να δημιουργεί προβλήματα σε κόστος στην περίπτωση που ο κινητός χρήστης μετακινείται σε νέο foreign δίκτυο συχνά λόγω του ότι το multicast delivery δένδρο πρέπει να κατασκευάζεται ξανά. Πλεονεκτεί όμως στο γεγονός ότι τα multicast πακέτα δρομολογούνται με βάση τα βέλτιστα και συντομότερα μονοπάτια προς τον προορισμό τους.

3.4.2 Bi-directional Tunnelling [4], [5]

Η λειτουργία του βασίζεται στη δημιουργία ενός tunnel διπλής κατεύθυνσης (bi-directional) μεταξύ ενός κινητού χρήστη και του home agent του, όταν ο κινητός χρήστης βρίσκεται μακριά από το home δίκτυό του, σε ένα ξένο δίκτυο. Ο κινητός χρήστης στέλλει και λαμβάνει multicast πακέτα χρησιμοποιώντας αυτό το tunnel. Με το bi-directional tunnelling δεν επηρεάζεται το multicast delivery tree από την αλλαγή τοποθεσίας του κινητού χρήστη και έτσι το κόστος είναι μικρό. Οδηγεί όμως στο tunnel convergence πρόβλημα το οποίο παρατηρείται όταν πολλά IP tunnels από διαφορετικούς home agents καταλήγουν στο ίδιο foreign δίκτυο. Αυτό συμβαίνει όταν υπάρχουν πολλοί κινητοί χρήστες οι οποίοι ανήκουν στο ίδιο multicast group και βρίσκονται στο ίδιο foreign δίκτυο. Ο κάθε ένας από τους home agents των κινητών αυτών χρηστών θα προωθεί στο foreign δίκτυο από ένα αντίγραφο του ίδιου multicast πακέτου με αποτέλεσμα να παρατηρείται packet duplication. Το πρόβλημα αυτό λύνεται με τη χρήση του Mobile Multicast Protocol (MOM), το οποίο περιγράφεται στη συνέχεια. Επιπλέον, τα multicast πακέτα μπορεί να δρομολογούνται με βάση μονοπάτια προς τον προορισμό τα οποία απέχουν πολύ από τα βέλτιστα.

3.4.3 Mobile Multicast Protocol (MOM) [4], [5]

Το πρωτόκολλο αυτό λύνει το convergence πρόβλημα το οποίο δημιουργείται στο bi-directional tunnelling με τη χρήση ενός Designated Multicast Service Provider (DMSP).

Ο DMSP είναι ο home agent ο οποίος επιλέγεται από τον foreign agent του foreign δικτύου στο οποίο παρατηρείται το packet duplication, από το σύνολο όλων των home agents που εξυπηρετούν τους κινητούς χρήστες οι οποίοι βρίσκονται στο ξένο αυτό δίκτυο και ανήκουν στο ίδιο multicast group. Αυτός ο home agent θα είναι ο μόνος που θα προωθεί το multicast πακέτο προς το ξένο δίκτυο παρόλο που μπορεί να εξυπηρετεί περισσότερους από έναν κινητούς χρήστες. Η χρήση όμως του DMSP προκαλεί ένα άλλο πρόβλημα, αυτό της απώλειας πακέτων. Η απώλεια παρατηρείται όταν ο τελευταίος κινητός χρήστης ο οποίος έχει ως home agent τον DMSP μετακινηθεί από το συγκεκριμένο ξένο δίκτυο. Αυτό σημαίνει πως ο DMSP θα σταματήσει να στέλλει multicast πακέτα στο ξένο δίκτυο με αποτέλεσμα οι υπόλοιποι κινητοί χρήστες οι οποίοι βρίσκονται σε αυτό και είναι μέλη του συγκεκριμένου multicast group να μην λάβουν τα πακέτα που προορίζονταν για αυτό. Υπάρχει μια διαφοροποίηση στη χρήση του DMSP με βάση την οποία καθορίζονται περισσότεροι από έναν DMSP κάθε φορά. Έτσι αν ο αρχικός DMSP σταματήσει να στέλει τα multicast πακέτα, οι κινητοί χρήστες να βασίζονται σε κάποιον άλλον DMSP για τη λήψη τους.

3.4.4 Rang-Based Mobile Multicast (RBMoM) [6]

Στόχος του πρωτοκόλλου αυτού είναι η μείωση της συχνότητας ανακατασκευής του multicast tree. Για την επίτευξη αυτού του στόχου εμπλέκονται δύο έννοιες, αυτή του multicast home agent (MHA) και αυτή της service range, δηλαδή εμβέλεια υπηρεσίας. Με αυτόν τον τρόπο υπάρχει περιορισμός στα δρομολόγια των multicast πακέτων καθώς επίσης και έλεγχος της συχνότητας ανακατασκευής του multicast tree. Ο MHA έχει την ευθύνη για το tunnelling των multicast πακέτων στον foreign agent του κινητού κόμβου τον οποίο εξυπηρετεί και πρέπει να είναι μέλος στο multicast group στο οποίο ανήκει και ο κινητός αυτός κόμβος. Σε αντίθεση με τον home agent του κινητού κόμβου ο οποίος παραμένει ο ίδιος, ο MHA σχετίζεται άμεσα με την τοποθεσία του κινητού χρήστη γι' αυτό και αλλάζει. Κάθε MHA έχει μια εμβέλεια υπηρεσίας (service range) η οποία καθορίζει τους κινητούς κόμβους τους οποίους μπορεί να εξυπηρετήσει. Ο MHA μπορεί να εξυπηρετήσει μόνο τους κινητούς κόμβους οι οποίοι βρίσκονται μέσα στην εμβέλειά του. Αυτό συνεπάγεται στο γεγονός ότι αν ένας κινητός κόμβος βρεθεί σε τοποθεσία η οποία δε βρίσκεται μέσα στην εμβέλεια του MHA του τότε θα πρέπει να βρει ένα νέο MHA, ο οποίος θα του προσφέρει τις υπηρεσίες του. Η ενέργεια αυτή είναι γνωστή ως MHA handoff. Ο κινητός κόμβος φτάνοντας σε ένα ξένο δίκτυο,

ανακαλύπτει και εγγράφεται μαζί με έναν FA ο οποίος θα επικοινωνήσει με τον HA του κινητού κόμβου ώστε να ενημερωθεί για τον MHA ο οποίος εξυπηρετεί τον κινητό αυτό κόμβο. Ο FA αφού λάβει τις πληροφορίες που χρειάζεται θα υπολογίσει την απόσταση προς τον MHA. Αν η απόσταση αυτή είναι μεγαλύτερη από το service range του MHA τότε θα πρέπει να επιλεγεί ένας νέος MHA. Έτσι συνήθως ορίζεται ως νέος MHA ο FA. Στη συνέχεια ο FA πρέπει να ενημερώσει τον HA για το MHA handoff το οποίο πραγματοποιήθηκε επειδή ο HA διατηρεί τις πληροφορίες για την τοποθεσία του MHA. Αν η απόσταση η οποία υπολογίζεται από τον FA είναι μέσα στα όρια του service range του MHA τότε ο FA ενημερώνει τον MHA ότι αυτός εξυπηρετεί τον κινητό κόμβο στο ξένο δίκτυο στο οποίο βρίσκεται την παρούσα χρονική στιγμή ώστε να γνωρίζει που πρέπει να προωθεί τα multicast πακέτα. Ο MHA πρέπει να διατηρεί μια λίστα με όλους τους MHs τους οποίους εξυπηρετεί. Το πρωτόκολλο αυτό μπορεί να οδηγήσει στο tunnel convergence πρόβλημα όπως παρατηρείται και στο bi-directional tunnelling. Αυτό οφείλεται στο ότι πολλοί MHAs εξυπηρετούν κινητούς κόμβους οι οποίοι ανήκουν στο ίδιο multicast group και βρίσκονται στο ίδιο ξένο δίκτυο το οποίο διαχειρίζεται ο ίδιος FA. Ο MHA παρέχει την multicast υπηρεσία στους κινητούς κόμβους ενώ ένας HA παρέχει unicast υπηρεσία στους κινητούς κόμβους τους οποίους εξυπηρετεί. Το πρωτόκολλο αυτό σε σύγκριση με το Remote Subscription παρουσιάζει λιγότερο συχνά multicast tree update και σε σύγκριση με το Bi-directional tunnelling το μέγεθος του tunnel είναι περιορισμένο, περιορίζοντας έτσι και το κόστος για το tunnelling.

ΚΕΦΑΛΑΙΟ 4

Κωδικοποίηση Βίντεο

4.1. Εισαγωγή

4.2. Αρχές κωδικοποίησης βίντεο

4.3. Είδη κωδικοποίησης

4.4. Είδη πλαισίων κατά την κωδικοποίηση

4.5. Πρότυπα κωδικοποίησης

4.1 Εισαγωγή [7]

Η ανάπτυξη και η χρήση εφαρμογών οι οποίες χρησιμοποιούν βίντεο ολοένα και αυξάνονται, αυξάνοντας και τις απαιτήσεις που πρέπει να υπάρχουν. Οι εφαρμογές αυτές για να έχουν τη δυνατότητα να λειτουργούν και να προσφέρουν τις υπηρεσίες τους στους χρήστες βασίζονται τόσο σε τεχνολογίες επικοινωνίας όσο και σε τεχνικές συμπίεσης του βίντεο.

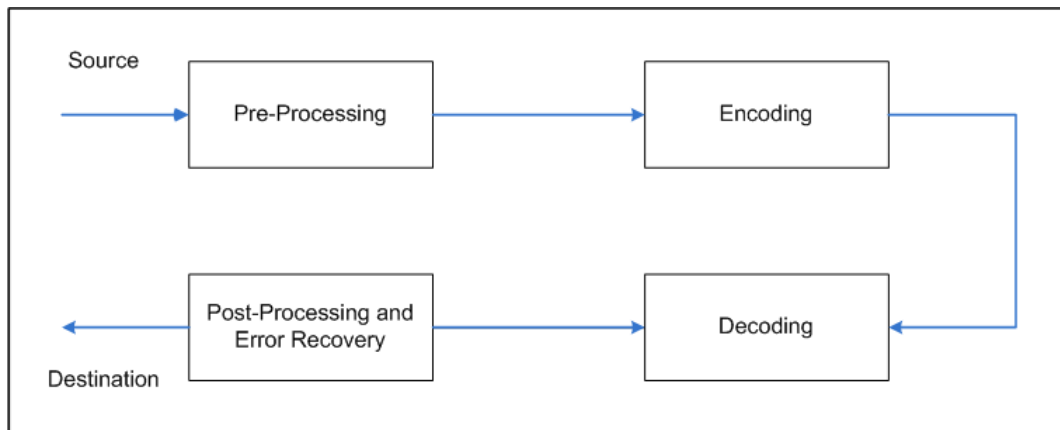
Τα δίκτυα τα οποία χρησιμοποιούνται για την υποστήριξη τέτοιων βίντεο εφαρμογών δεν μπορούν να υποστηρίξουν τους υψηλούς ρυθμούς δεδομένων που απαιτούνται για την παρουσίαση αυτού του τύπου δεδομένων σε ψηφιακή μορφή. Έτσι αυτό έχει ως αποτέλεσμα την ανάγκη της διαδικασίας της συμπίεσης του βίντεο, ώστε να μειωθεί ο ρυθμός δεδομένων, ώστε να μπορούν τα περισσότερα δίκτυα να υποστηρίξουν τις εφαρμογές αυτές.

Η συμπίεση πραγματοποιείται στα δεδομένα βίντεο πριν την μετάδοση τους σε ένα δίκτυο και βασίζεται στον πλεονασμό των πληροφοριών που υπάρχουν στο βίντεο αφού μειώνοντας ή παραλείποντας τις μπορεί να μειωθεί το μέγεθος του βίντεο και έτσι οι απαιτήσεις για την μετάδοση και την αποθήκευσή του, όσον αφορά το απαιτούμενο εύρος ζώνης, το απαιτούμενο μέγεθος δεδομένων για μετάδοση και τον απαιτούμενο διαθέσιμο χώρο αποθήκευσης.

4.2 Αρχές κωδικοποίησης Βίντεο

Ένα σύστημα κωδικοποίησης βίντεο πρέπει να βασίζεται στο ότι ο τελικός στόχος είναι η μεταφορά των δεδομένων με όσο το δυνατό μεγαλύτερη ακρίβεια με δεδομένο ένα διαθέσιμο ρυθμό μετάδοσης. Ένα τέτοιο σύστημα ονομάζεται codec και αποτελείται από έναν κωδικοποιητή και έναν αποκωδικοποιητή.

Στο επόμενο σχήμα παρουσιάζεται η διαδικασία της κωδικοποίησης και αποκωδικοποίησης βίντεο. Αρχικά το βίντεο λαμβάνει κάποια επεξεργασία καθώς βρίσκεται στην πλευρά του αποστολέα και στη συνέχεια πραγματοποιείται η κωδικοποίησή του. Στη συνέχεια πραγματοποιείται η αποκωδικοποίηση καθώς φτάνει στον παραλήπτη και ακολουθούν ενέργειες για ανάνηψη από τυχόν σφάλματα που μπορεί να παρατηρήθηκαν.

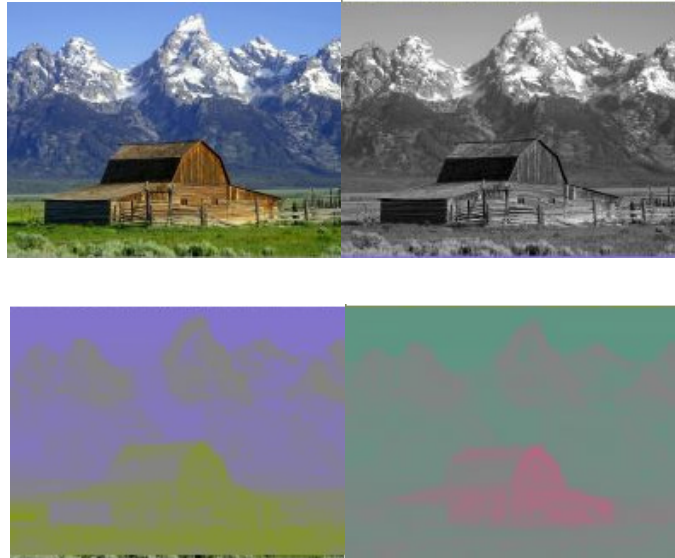


Σχήμα 4.1: Κωδικοποίηση και Αποκωδικοποίηση Βίντεο

Το βίντεο είναι μια ακολουθία από ψηφιακές εικόνες. Κάθε ψηφιακή εικόνα ή ένα frame από το ψηφιακό βίντεο αποτελείται από 3 πίνακες οι οποίοι αντιπροσωπεύουν τα τρία στοιχεία που χρησιμοποιούνται για την απεικόνιση του χρώματος. Τα τρία αυτά στοιχεία είναι το Y, Cb και Cr. Το στοιχείο Y ονομάζεται luma και σχετίζεται με τη φωτεινότητα. Τα δύο άλλα στοιχεία, τα chroma στοιχεία όπως καλούνται, αντιπροσωπεύουν τον βαθμό στον οποίο το χρώμα αποκλίνει από γκριζο σε μπλε και κόκκινο αντίστοιχα. Το ανθρώπινο μάτι είναι πιο ευαίσθητο στο luma στοιχείο παρά στα chroma στοιχεία. Ανάλογα με το πόσα δείγματα υπάρχουν για κάθε στοιχείο στους αντίστοιχους πίνακες, υπάρχουν διάφοροι τύποι δειγματοληψιών (sampling) όπως η δειγματοληψία 4:4:4, 4:2:2 και 4:2:0. Το πρώτο είδος έχει 4 Cb και Cr pixels για κάθε

ένα μπλοκ 2x2 pixels του Y στοιχείου. Η δεύτερη δειγματοληψία έχει 2 Cb και 2 Cr pixels για κάθε ένα μπλοκ 2x2 pixels του Y στοιχείου. Τέλος η δειγματοληψία 4:2:0 έχει ένα Cb και ένα Cr pixel για κάθε ένα μπλοκ 2x2 pixels του Y στοιχείου και είναι πολύ συνηθισμένη. [10]

Οι ακόλουθες εικόνες παρουσιάζουν την αυθεντική εικόνα, στη συνέχεια το στοιχείο Y, το στοιχείο Cb και το στοιχείο Cr σε αυτήν.



Σχήμα 4.2: Y, Cb και Cr εικόνες

Υπάρχουν δύο βασικές μορφές βίντεο το progressive και το interlaced. Ένας πίνακας από frames με βίντεο samples περιέχει δύο interleaved πεδία, το πεδίο top και το πεδίο bottom. Κατά το interlacing γίνεται σύλληψη μόνο ενός από τα δύο πεδία και για αυτόν τον λόγο απαιτούνται δύο περίοδοι δειγματοληψίας για να συλληφθεί ολόκληρο το frame. Αν τα δύο πεδία δεν συλληφθούν σε διαφορετικές χρονικές στιγμές τότε το frame θεωρείται progressive. [10]

Κατά τη διάρκεια της κωδικοποίησης του βίντεο πρέπει να μειωθεί το μέγεθος των δεδομένων. Αυτό πραγματοποιείται με την συμπίεση του βίντεο. Οι αλγόριθμοι συμπίεσης βίντεο μπορούν να χωριστούν σε δύο κατηγορίες στους αλγόριθμους χωρίς απώλεια (lossless compression algorithms) και στους αλγόριθμους με απώλεια (lossy compression algorithms). Όταν πραγματοποιείται συμπίεση με κάποιον αλγόριθμο της πρώτης κατηγορίας, αυτό σημαίνει πως θα μειωθεί η ποσότητα των δεδομένων τα οποία

θα μεταδοθούν με τέτοιο τρόπο ώστε όταν αυτά αποσυμπίεστούν να μην υπάρχει απώλεια δεδομένων. Αντίθετα, με έναν αλγόριθμο της δεύτερης κατηγορίας το αποτέλεσμα της αποσυμπίεσης δε θα είναι το ίδιο ακριβώς με το αρχικό σύνολο δεδομένων. Πιο συγκεκριμένα, το αποτέλεσμα θα εκλαμβάνεται από τον παραλήπτη ως ένα καλό αντίγραφο του αρχικού συνόλου δεδομένων χωρίς όμως να είναι ακριβώς το ίδιο. Όσο πιο ψηλό είναι το επίπεδο συμπίεσης στα δεδομένα βίντεο τόσο πιο ακριβή θα είναι το αποτέλεσμα. [7]

Υπάρχουν διάφορες τεχνικές οι οποίες χρησιμοποιούνται στη συμπίεση βίντεο που πραγματοποιείται σε διάφορες εφαρμογές: [7],[10]

Πρόβλεψη (Prediction)

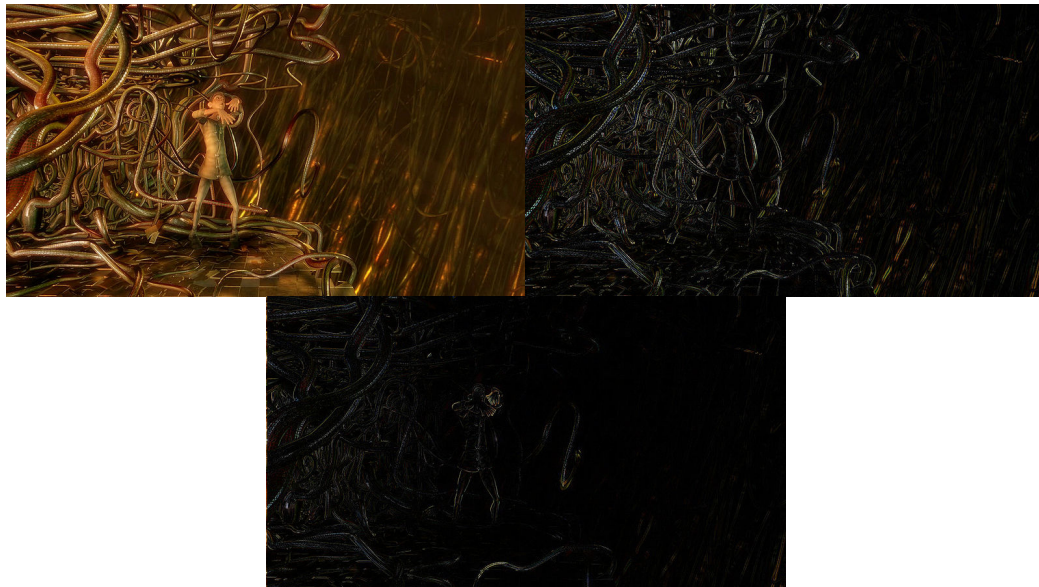
Κατά τη διάρκεια αυτής της διαδικασίας γίνεται δημιουργία ενός συνόλου από τιμές πρόβλεψης (prediction values) με βάση το οποίο προβλέπονται οι τιμές των εισερχόμενων δειγμάτων (samples), έτσι ώστε μόνο οι διαφορές από τις προβλεπόμενες τιμές να χρησιμοποιηθούν ως τις τιμές για την απεικόνιση. Αυτές οι τιμές καλούνται οι υπολειπόμενες τιμές(residual values). Αυτό διευκολύνει την κωδικοποίηση. Βασίζεται στην αναφορά που γίνεται σε προηγούμενα και σε κάποιες περιπτώσεις, επόμενα πλαίσια. Έτσι κατά την μετάδοση αντί να πραγματοποιηθεί αποστολή του αρχικού βίντεο ως ένα σύνολο από ξεχωριστά συμπίεσμένα πλαίσια, στέλλεται μόνο ένα υποσύνολο από αυτά σε αυτήν την μορφή ενώ από τα υπόλοιπα πλαίσια στέλλονται μόνο οι διαφορές μεταξύ των περιεχομένων του αρχικού πλαισίου και των περιεχομένων του προβλεπόμενου πλαισίου.

Εκτίμηση Κίνησης (Motion Estimation) και Αντιστάθμιση Κίνησης(Motion Compensation)

Ο βαθμός στον οποίο εκτιμάται κάθε κίνηση μεταξύ διαδοχικών πλαισίων καθορίζει το βαθμό ακρίβειας που θα έχει η πρόβλεψη. Η εκτίμηση κίνησης βασίζεται στο γεγονός ότι διαδοχικά πλαίσια βίντεο είναι παρόμοια με εξαίρεση τις περιπτώσεις που υπάρχει κίνηση των αντικειμένων μεταξύ των πλαισίων. Αυτό σημαίνει πως ένα μεγάλο μέρος των πληροφοριών που χρησιμοποιούνται για την απεικόνιση ενός πλαισίου θα είναι το ίδιο με τις πληροφορίες του επόμενου πλαισίου. Ένα μέρος των πληροφοριών όμως θα διαφέρει λόγω της κίνησης. Έτσι δημιουργείται ένα πλαίσιο βίντεο με βάση την

αναφορά που γίνεται σε ένα άλλο πλαίσιο. Η τεχνική αυτή συγκρίνει μικρά τμήματα σε δύο συνεχόμενα πλαίσια με στόχο την εύρεση διαφορών σε αυτά. Αν ανιχνευθεί διαφορά τότε πρέπει να καθοριστεί το γειτονικό τμήμα στο οποίο μετακινήθηκε το αρχικό, όμως μόνο μερικά γειτονικά τμήματα είναι δυνατόν να ερευνηθούν. Κατά τη διαδικασία αυτή θα πρέπει να καθοριστούν τα διανύσματα κίνησης (motion vectors (MVs)) τα οποία δείχνουν τον μετασχηματισμό που πρέπει να πραγματοποιηθεί ώστε να εφαρμοστεί η εκτιμώμενη κίνηση.

Η αντιστάθμιση κίνησης είναι η χρήση των πιο πάνω διανυσμάτων κίνησης για να διαμορφωθεί η πρόβλεψη. Η εικόνα θα μετασχηματιστεί με βάση την αναφορά που γίνεται είτε σε προηγούμενη εικόνα είτε σε επόμενη εικόνα από την εικόνα που τυγχάνει επεξεργασίας τη δεδομένη στιγμή.



Σχήμα 4.3: Motion Compensation

Από πάνω αριστερά βλέπουμε την αρχική εικόνα, στη συνέχεια δίπλα τις διαφορές μεταξύ του αρχικού πλαισίου και του επόμενου και τέλος, κάτω τις motion compensated διαφορές.

Μετασχηματισμός (Transformation)

Η διαδικασία με την οποία σχηματίζεται ένα νέο σύνολο από δείγματα από ένα συνδυασμό από εισερχόμενα δείγματα χρησιμοποιώντας linear combination. Είναι

στενά συνδεδεμένη με την διαδικασία της πρόβλεψης. Ένα γνωστό παράδειγμα είναι ο μετασχηματισμός DCT(Discrete Cosine Transform).

Κβαντοποίηση (Quantization)

Κατά τη διαδικασία αυτή, μειώνεται η ακρίβεια με βάση την οποία πραγματοποιείται η απεικόνιση μιας τιμής δείγματος ή ενός συνόλου από αυτές με σκοπό να μειωθεί η ποσότητα των δεδομένων που χρειάζεται για να πραγματοποιηθεί η κωδικοποίηση.

Κωδικοποίηση εντροπίας (Entropy coding)

Η παρουσίαση των συμβόλων τα οποία λαμβάνουν διακριτές τιμές λαμβάνει υπόψη τις σχετικές πιθανότητες για τις διάφορες δυνατές τιμές του κάθε συμβόλου. Ένα παράδειγμα είναι η κωδικοποίηση μεταβλητού μήκους(VLC- Variable Length Code).

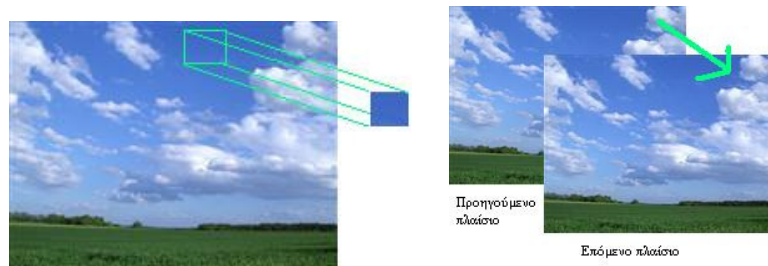
4.3 Είδη Κωδικοποίησης [10]

4.3.1 Intra picture ή Intra coding

Η εικόνα κωδικοποιείται χωρίς να γίνεται αναφορά σε άλλες εικόνες της ακολουθίας βίντεο. Η κωδικοποίηση πραγματοποιείται με βάση τις πληροφορίες που σχετίζονται με τη συγκεκριμένη εικόνα. Η ακριβής διαδικασία εξαρτάται από τον κωδικοποιητή ο οποίος χρησιμοποιείται αλλά συνήθως έχει ως εξής: γίνεται ο διαχωρισμός σε macro blocks, ο μετασχηματισμός όπως για παράδειγμα ο DCT, η κβαντοποίηση και η μετάδοση με βάση κωδικοποίηση εντροπίας.

4.3.2 Inter picture ή Inter coding

Δίνει τη δυνατότητα για βελτιωμένη απόδοση συμπίεσης χρησιμοποιώντας τη μεγάλη ποσότητα από χρονικό πλεονασμό(temporal redundancy) στο βίντεο. Με τον χρονικό πλεονασμό εννοούμε ότι δύο πλαίσια έχουν τις ίδιες τιμές σε ίδιες θέσεις. Δεν κωδικοποιούνται όλες οι περιοχές επαναληπτικά αλλά μόνο οι διαφορές που παρατηρούνται στη σύγκριση με τις προηγούμενες εικόνες στις οποίες γίνεται αναφορά. Αρκετό μέρος της κάθε σκηνής απλά επαναλαμβάνεται από εικόνα σε εικόνα χωρίς να πραγματοποιείται κάποια σημαντική αλλαγή. Εκτός από τον χρονικό πλεονασμό, ένα πλαίσιο μπορεί να χαρακτηρίζεται και από χωρικό πλεονασμό(spatial redundancy). Αυτό σημαίνει πως μέσα σε ένα πλαίσιο υπάρχουν ίδια στοιχεία που έχουν ίδιες τιμές.



Σχήμα 4.4: Χωρικός και Χρονικός Πλεονασμός

Οι δύο πιο πάνω εικόνες παρουσιάζουν τον χωρικό και τον χρονικό πλεονασμό αντίστοιχα.

4.4 Είδη πλαισίων κατά την κωδικοποίηση [7]

Υπάρχουν δύο βασικοί τύποι πλαισίων, τα πλαίσια τα οποία κωδικοποιούνται ανεξάρτητα από άλλα χωρίς καμιά αναφορά δηλαδή σε άλλα πλαίσια και τα πλαίσια για τα οποία πραγματοποιείται πρόβλεψη κάνοντας αναφορά σε άλλα πλαίσια. Τα πρώτα είναι γνωστά ως intracoded πλαίσια ή I-πλαίσια. Τα τελευταία χωρίζονται στα predictive ή P-πλαίσια και τα bidirectional ή B-πλαίσια. Τα B-πλαίσια είναι γνωστά ως interceded ή interpolation πλαίσια.

4.4.1 I-πλαίσιο

Το πλαίσιο αυτό κωδικοποιείται χωρίς να γίνεται αναφορά σε άλλα πλαίσια. Κάθε πλαίσιο λαμβάνεται ως ξεχωριστή οντότητα και η κωδικοποίηση των πινάκων σχετικά με τα στοιχεία Y, Cb, Cr πραγματοποιείται ανεξάρτητα χρησιμοποιώντας τον JPEG αλγόριθμο. Με τα I-frames λαμβάνουμε χαμηλό επίπεδο συμπίεσης. Πρέπει να δίνεται προσοχή στην εξερχόμενη ροή αφού πρέπει όλα τα I-frames να είναι παρών σε τακτά διαστήματα σε αυτήν. Αυτό συμβαίνει για την περίπτωση που κάποια από αυτά είναι διαφθαρμένα. Αν ένα I πλαίσιο είναι διεφθαρμένο τότε αυτό θα προκαλέσει πρόβλημα στα πλαίσια τα οποία κάνουν αναφορά σε αυτόν για την πρόβλεψη τους. Ο αριθμός των πλαισίων μεταξύ διαδοχικών I-frames είναι γνωστός ως group of pictures (GOP).

Όσον αφορά την αποκωδικοποίηση των I-πλαισίων, αποκωδικοποιούνται οι συμπίεσμένες πληροφορίες για το πλαίσιο μόλις αυτό ληφθεί ώστε να ξαναδημιουργηθεί το αρχικό πλαίσιο.

4.4.2 P-πλαίσιο

Το πλαίσιο αυτό κωδικοποιείται με βάση αναφορά που γίνεται σε άλλα πλαίσια και συγκεκριμένα σε αναφορά σε περιεχόμενα κάποιου I-πλαισίου ή P-πλαισίου το οποίο βρίσκεται πριν από αυτό. Με τα P-frames λαμβάνουμε ψηλότερο επίπεδο συμπίεσης λόγω του ότι η κωδικοποίηση βασίζεται στην τεχνική εκτίμησης κίνησης (motion estimation) και στην τεχνική αντιστάθμιση κίνησης (motion compensation). Ο αριθμός των P-frames μεταξύ δύο διαδοχικών ζευγών από I-frames είναι περιορισμένος γιατί αν υπάρχει κάποιο λάθος στο πρώτο P-frame τότε αυτό μεταδίδεται στο επόμενο πλαίσιο. Ο αριθμός των πλαισίων μεταξύ ενός P-frame και του αμέσως προηγούμενου I ή P-frame ονομάζεται prediction span.

Όσον αφορά την αποκωδικοποίηση του P πλαισίου, αποκωδικοποιείται η πληροφορία που φτάνει και μετά το αποτέλεσμα χρησιμοποιείται με το αποκωδικοποιημένο προηγούμενο I ή P frame για να παραχθεί το τελικό αποκωδικοποιημένο πλαίσιο.

4.4.3 B-πλαίσιο

Το πλαίσιο αυτό κωδικοποιείται και αυτό με βάση αναφορά που γίνεται σε άλλα πλαίσια και πιο συγκεκριμένα σε προηγούμενα αλλά και σε μελλοντικά πλαίσια. Με αυτόν τον τρόπο πετυχαίνεται καλύτερη εκτίμηση κίνησης. Η κωδικοποίηση λαμβάνει υπόψη το αμέσως προηγούμενο I ή P πλαίσιο και το αμέσως επόμενο I ή P πλαίσιο. Αυτό οδηγεί στην αύξηση της καθυστέρησης όσον αφορά την κωδικοποίηση (και αποκωδικοποίηση) η οποία δίνεται από τον χρόνο αναμονής για το επόμενο I ή P πλαίσιο στην ακολουθία. Τα B-πλαίσια δεν εμπλέκονται στην κωδικοποίηση άλλων πλαισίων αφού δε γίνεται αναφορά σε αυτά και γι' αυτόν τον λόγο δεν μεταδίδουν σφάλματα που μπορεί να υπάρχουν.

Σχετικά με την αποκωδικοποίηση του B πλαισίου, αποκωδικοποιείται η πληροφορία που φτάνει και μετά το αποτέλεσμα χρησιμοποιείται με το αμέσως αποκωδικοποιημένο

προηγούμενο I ή P frame και το αμέσως αποκωδικοποιημένο επόμενο I ή P frame για να παραχθεί το τελικό αποκωδικοποιημένο πλαίσιο.

Η σειρά της κωδικοποίησης και της μετάδοσης των κωδικοποιημένων πλαισίων μπορεί να αλλάξει ώστε όταν λαμβάνεται ένα B πλαίσιο να είναι διαθέσιμα το αμέσως προηγούμενο και το αμέσως επόμενο I ή P πλαίσιο και έτσι να μειωθεί και ο χρόνος που απαιτείται για την αποκωδικοποίηση κάθε B πλαισίου. Έτσι θα είναι διαθέσιμες οι αποκωδικοποιημένες πληροφορίες των δύο πλαισίων στα οποία γίνεται αναφορά.

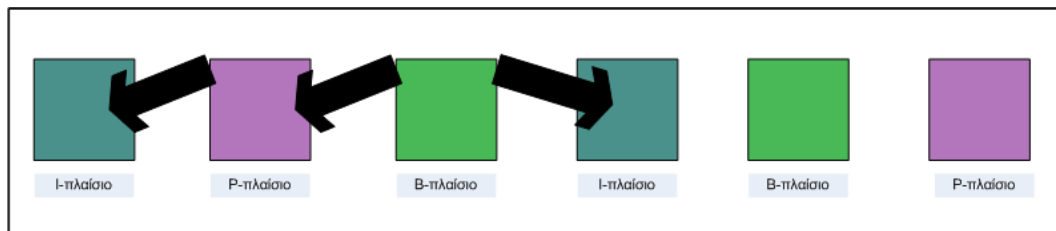
4.4.4 PB-πλαίσιο

Δεν πρόκειται για διαφορετικό τύπο πλαισίου κατά την κωδικοποίηση βίντεο αλλά αφορά τον τρόπο με τον οποίο δύο γειτονικά P ή B πλαίσια μπορούν να κωδικοποιηθούν σαν να ήταν ένα πλαίσιο.

4.4.5 D- πλαίσιο

Αυτό το πλαίσιο χρησιμοποιείται σε εφαρμογές movie/video on demand. Το βίντεο μπορεί να αποσυμπίεστεί σε πολύ ψηλότερες ταχύτητες και αυτό δίνει τη δυνατότητα για λειτουργίες όπως το rewind και το fast-forward σε μια ταινία. Τα D-πλαίσια εισάγονται σε τακτά διαστήματα μέσα στη ροή και δεν λαμβάνονται υπόψη κατά την αποκωδικοποίηση των P και B πλαισίων. Επίσης, τα συγκεκριμένα πλαίσια έχουν υποστεί ψηλό επίπεδο συμπίεσης.

Η πιο κάτω εικόνα παρουσιάζει τα τρία βασικά είδη πλαισίων κατά την κωδικοποίηση βίντεο, και την αναφορά που γίνεται από αυτά σε προηγούμενα και επόμενα πλαίσια.



Σχήμα 4.5: Είδη πλαισίων

4.5 Πρότυπα Κωδικοποίησης [7],[8]

4.5.1 ITU-T H.261

Το πρότυπο αυτό έχει καθοριστεί από την ομάδα ITU-T με σκοπό να χρησιμοποιηθεί σε υπηρεσίες video telephony και video conferencing για ενοποιημένες υπηρεσίες ψηφιακών δικτύων (ISDN). Το ITU-T H.261 ήταν η βάση για επόμενα πρότυπα κωδικοποίησης βίντεο. Είναι γνωστό και ως το πρότυπο $p \times 64$ όπου το p λαμβάνει τιμές από το 1 μέχρι και το 30. Αυτό σημαίνει πως τα διάφορα κανάλια μετάδοσης θα έχουν ρυθμούς οι οποίοι είναι πολλαπλάσια των 64kbps.

Χρησιμοποιεί μόνο I και P πλαίσια με τρία P πλαίσια μεταξύ κάθε ζεύγους από I πλαίσια και έτσι πραγματοποιείται πρόβλεψη προς τα εμπρός (forward-only) με αναφορά στην προηγούμενη εικόνα. Κάθε πλαίσιο χωρίζεται σε macro blocks των 16×16 pixels. Το digitization format που χρησιμοποιείται είναι το CIF και το quarter CIF (QCIF). Άλλα χαρακτηριστικά του είναι: 4:2:0 sampling, zig zag inverse scanning of coefficients, scalar quantization, κωδικοποίηση μεταβλητού μήκους των coefficients, αντιστάθμιση κίνησης με full pixel precision και χρήση loop filter.

4.5.2 MPEG-1 (Part 2)

Το πρότυπο αυτό αναπτύχθηκε μετά από το H.261 και προοριζόταν για την αποθήκευση ήχου και βίντεο ποιότητας VHS σε CD-ROM με ρυθμό δεδομένων μέχρι και 1.5 Mbps. Για να υπάρχει πιο γρήγορη πρόσβαση στο αποθηκευμένο υλικό χρησιμοποιούνται και ψηλότεροι ρυθμοί δεδομένων.

Όπως και στο H.261, το κάθε πλαίσιο χωρίζεται σε macro blocks των 16×16 pixels. Υποστηρίζει τέσσερις τύπους πλαισίων: I, B, P και D πλαίσια. Έτσι πραγματοποιεί επιπρόσθετα και πρόβλεψη προς τα πίσω (backward) και πρόβλεψη προς δύο κατευθύνσεις (bidirectional). Σε αντίθεση με το H.261 δεν έχει loop filter αλλά ένα απλό sample filter.

MPEG-2 (Part 2)

Το πρότυπο MPEG-2 είναι μια επέκταση στο πρότυπο MPEG-1, το οποίο επιτρέπει την κωδικοποίηση βίντεο σε διάφορες αναλύσεις όπως για παράδειγμα υψηλή ανάλυση

όπως HDTV. Υποστηρίζει έτσι διαφορετικούς ρυθμούς δεδομένων και πραγματοποιεί αποδοτική κωδικοποίηση interlaced βίντεο. Χρησιμοποιείται σε διάφορες εφαρμογές όπως το DVD-Video, HDV και DVB.

Το κάθε πλαίσιο χωρίζεται σε μπλοκ των 8x8 pixel και έτσι ένας πίνακας 2x2 από μπλοκ δημιουργεί ένα macro block. Για τη συμπίεση του βίντεο χρησιμοποιείται πρόβλεψη, μετασχηματισμός DCT και κβαντοποίηση των DCT coefficients. Υποστηρίζει I, P και B πλαίσια. Όπως αναφέρθηκε πριν, τα I πλαίσια κωδικοποιούνται χωρίς πρόβλεψη, τα P πλαίσια με πρόβλεψη κάνοντας αναφορά σε προηγούμενα πλαίσια ενώ τα B πλαίσια με πρόβλεψη κάνοντας αναφορά σε προηγούμενα και επόμενα πλαίσια.

Ορίζει τέσσερα επίπεδα τα οποία καθορίζουν χαρακτηριστικά όπως είναι ο μέγιστος ρυθμός δεδομένων. Αυτά είναι το επίπεδο low, main, high 1440 και high. Το επίπεδο low παράγει βίντεο ποιότητας VHS. Το επίπεδο main προορίζεται για βίντεο ποιότητας studio. Το επίπεδο high 1440 είναι κατάλληλο για High Definition Television(HDTV) ενώ το επίπεδο high για wide-screen HDTV. Κάθε επίπεδο μπορεί να συσχετιστεί με προφίλ τα οποία ορίζονται από αυτό το πρότυπο.

4.5.4 ITU-T H.263 [9]

Το πρότυπο αυτό έχει αναπτυχθεί από την ομάδα ITU-T με σκοπό να χρησιμοποιηθεί σε εφαρμογές βίντεο όπως το video telephony, videoconferencing και interactive games, εφαρμογές δηλαδή πραγματικού χρόνου. Στόχος ήταν να χρησιμοποιηθεί η διαθέσιμη τεχνολογία ώστε να παραχθεί ένα πρότυπο το οποίο να έχει χαμηλή πολυπλοκότητα, να παρουσιάζει ευρωστία σε τυχόν σφάλματα και να είναι ευέλικτο σε μελλοντικές επεκτάσεις. Είναι βασισμένο στο H.261 αλλά προσφέρει βελτιωμένη απόδοση στη συμπίεση και πιο προχωρημένες επιλογές όσον αφορά την κωδικοποίηση, όπως unrestricted motion vectors, syntax based arithmetic coding, advanced prediction και PB πλαίσια. Έτσι πετυχαίνεται καλύτερη ποιότητα εικόνας.

Συνδυάζει inter-picture πρόβλεψη και DCT transform coding του prediction residual με σκοπό να μειώσει το χρονικό και χωρικό πλεονασμό αντίστοιχα. Μετά από τον μετασχηματισμό, το prediction error signal γίνεται quantize με ένα scalar quantize και τα τελικά σύμβολα κωδικοποιούνται με κωδικοποίηση μεταβλητού μήκους και

μεταδίδονται. Η αντιστάθμιση κίνησης πραγματοποιείται με half pixel precision. Το κάθε πλαίσιο χωρίζεται σε macroblocks των 16x16 pixels. Τα κύρια digitization formats που χρησιμοποιούνται είναι το QCIF και το sub-QCIF. Χρησιμοποιεί I, P και B πλαίσια. Γειτονικά ζεύγη από P και B πλαίσια μπορούν να κωδικοποιηθούν ως μια οντότητα, λειτουργώντας δηλαδή ως ένα PB πλαίσιο το οποίο προσφέρει ψηλότερο frame rate.

Άλλα χαρακτηριστικά του είναι: error resilience, error tracking, independent segment decoding.

4.5.6 H.264/AVC –MPEG4 (Part 10) [10], [11]

Το πρότυπο αυτό έχει αναπτυχθεί από την ομάδα ITU-T Video Coding Experts Group και την ομάδα ISO/IEC Moving Picture Experts Group και είναι το πιο πρόσφατο πρότυπο κωδικοποίησης. Σε σύγκριση με τα προηγούμενα πρότυπα κωδικοποίησης βίντεο παρέχει βελτιωμένη απόδοση όσον αφορά τη συμπίεση. Μπορεί να παρέχει καλή ποιότητα βίντεο σε χαμηλότερους ρυθμούς δεδομένων από αυτούς που χρησιμοποιούν προηγούμενα πρότυπα. Σχεδιάστηκε για χρήση σε διάφορες περιοχές εφαρμογών όπως video on demand, multimedia streaming εφαρμογές, multimedia messaging υπηρεσίες και video telephony. Παρέχει διάφορους μηχανισμούς που στοχεύουν στην βελτίωση της δυνατότητας πρόβλεψης της εικόνας που θα κωδικοποιηθεί, στην αποδοτικότητα σχετικά με την κωδικοποίηση και την ευρωστία σε σφάλματα και απώλειες όπως επίσης και για τη λειτουργία σε διαφορετικά δίκτυα. Το H.264/AVC μπορεί να χρησιμοποιηθεί για την κωδικοποίηση βίντεο το οποίο περιέχει progressive ή interlaced πλαίσια ή συνδυασμό των δύο.

Η λειτουργία του βασίζεται στο Network Abstraction Layer (NAT) και στο Video Coding Layer(VCL). Το VCL κωδικοποιεί τα αρχικά δεδομένα ενώ το NAT είναι υπεύθυνο για τη διαμόρφωση των κωδικοποιημένων αυτών δεδομένων και την παροχή για τις πληροφορίες της επικεφαλίδας του πακέτου(header information) ώστε τα VCL δεδομένα να μπορούν να χρησιμοποιηθούν από διαφορετικά συστήματα.

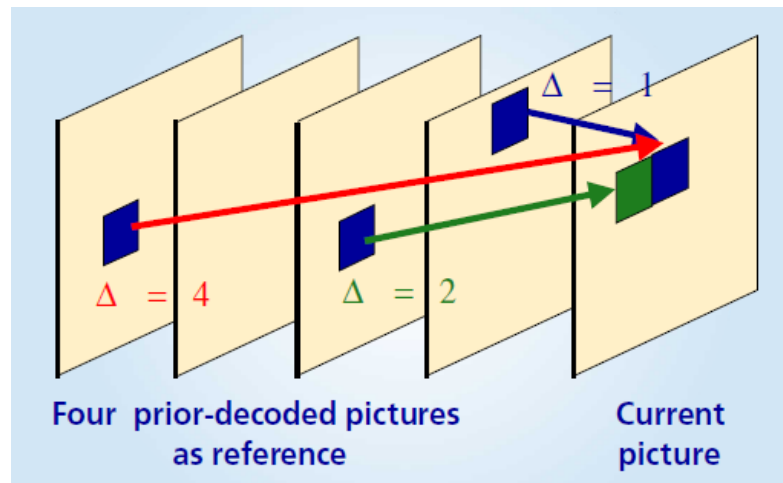
Το πρότυπο αυτό δίνει τις απαιτούμενες δυνατότητες για την ανάπτυξη νέων εφαρμογών. Επιπρόσθετα, σημαντικό είναι το γεγονός ότι πρόκειται για ένα ανοικτό πρότυπο.

4.5.6.1 Χαρακτηριστικά του πρότυπου H264/AVC

Το πρότυπο αυτό σε σύγκριση με προηγούμενα πρότυπα κωδικοποίησης βίντεο παρουσιάζει διάφορα χαρακτηριστικά και μηχανισμούς οι οποίοι βελτιώνουν διάφορες πτυχές της λειτουργίας της κωδικοποίησης.

Καταρχήν υπάρχει βελτίωση κατά την διαδικασία της πρόβλεψης για μια εικόνα. Το πρότυπο αυτό είναι ευέλικτο σε ότι αφορά την αντιστάθμιση κίνησης αφού υπάρχει δυνατότητα για χρήσης μπλοκ μεταβλητού μεγέθους και με μπλοκ τα οποία έχουν μεγέθη αρκετά μικρά. Επιτρέπει επίσης να υπάρχει ακρίβεια quarter-sample στη διαδικασία αυτή παρά την half-sample ακρίβεια που μπορεί να φτάσουν τα προηγούμενα πρότυπα. Κατά τη διαδικασία της αντιστάθμισης κίνησης είναι δυνατόν να χρησιμοποιηθούν πολλαπλές αναφορές σε εικόνες για να πραγματοποιηθεί η πρόβλεψη μιας συγκεκριμένης εικόνας.

Η αναφορά σε πολλαπλές εικόνες κατά την διαδικασία της αντιστάθμισης κίνησης παρουσιάζεται στο επόμενο σχήμα.



Σχήμα 4.6: Αναφορά σε πολλαπλές εικόνες κατά την αντιστάθμιση κίνησης

Βελτιώσεις έχουν πραγματοποιηθεί και όσον αφορά την αποδοτικότητα της κωδικοποίησης. Ο μετασχηματισμός γίνεται με χρήση μπλοκ που έχουν μικρό μέγεθος αλλά λόγω του ότι μπορεί να είναι πιο αποδοτικό σε κάποιες περιπτώσεις η χρήση μεγαλύτερου μεγέθους μπλοκ, μπορεί να έχουμε ιεραρχικό μετασχηματισμό (hierarchical block transform). Χρησιμοποιούνται επίσης μέθοδοι κωδικοποίησης

εντροπίας όπως η αριθμητική κωδικοποίηση (arithmetic coding) και η προσαρμοστική με βάση το περιεχόμενο κωδικοποίηση (content-adaptive coding). Με αυτόν τον τρόπο βελτιώνεται η επίδοση.

Το H.264/AVC επιτρέπει στον κωδικοποιητή να αποστέλλει πλεονάζουσες πληροφορίες όσον αφορά τις εικόνες σε περίπτωση που παρατηρηθεί απώλεια δεδομένων. Έτσι θα υπάρχει ευρωστία όσον αφορά τις απώλειες δεδομένων.

Network Abstraction Layer (NAL)

Διευκολύνει το mapping των H.264/AVC VCL δεδομένων στα διάφορα επίπεδα δικτύου. Υπάρχει η έννοια των NAL units στα οποία οργανώνονται τα κωδικοποιημένα δεδομένα βίντεο. Κάθε NAL unit είναι ένα πακέτο το οποίο αποτελείται από έναν ακέραιο αριθμό από bytes και ξεκινά με μια επικεφαλίδα η οποία έχει μέγεθος ένα byte και αντιπροσωπεύει τον τύπο των δεδομένων. Τα υπόλοιπα bytes αποτελούν τα payload δεδομένα του πακέτου σύμφωνα με τον τύπο της επικεφαλίδας. Μια σειρά από NAL units αποτελεί ένα NAL unit stream.

Το πρότυπο H.264/AVC καθορίζει ένα byte stream format όσον αφορά τα NAL units κατά την οποία κάθε NAL unit έχει ένα πρόθεμα μήκους τριών bytes, το οποίο καλείται start code prefix. Το πρόθεμα αυτό είναι μοναδικό για κάθε NAL unit και μπορεί να χρησιμοποιηθεί για την αναγνώριση και εύρεση των ορίων του κάθε NAL unit. Αυτή η διαμόρφωση είναι χρήσιμη σε συστήματα στα οποία πρέπει να πραγματοποιείται η παράδοση ενός NAL unit stream ως μια διατεταγμένη ροή από bytes και στα οποία είναι απαραίτητη η αναγνώριση των ορίων των NAL units. Υπάρχουν συστήματα στα οποία η αναγνώριση των ορίων των NAL units μπορεί να επιτευχθεί χωρίς τη χρήση του προθέματος start code. Στα συστήματα αυτά τα δεδομένα τα οποία έχουν κωδικοποιηθεί μεταφέρονται σε πακέτα με τη βοήθεια του επιπέδου μεταφοράς.

Ένα σύνολο από συνεχόμενα NAL units καλείται access unit και όταν αυτό αποκωδικοποιηθεί τότε έχει ως αποτέλεσμα μια αποκωδικοποιημένη εικόνα. Έτσι μια κωδικοποιημένη εικόνα αποτελείται από ένα σύνολο από συνεχόμενα access units. Τα NAL units χωρίζονται σε VCL και non VCL units. Τα δεδομένα τα οποία αντιπροσωπεύουν τις τιμές των δειγμάτων του βίντεο βρίσκονται στα VCL units ενώ οι επιπρόσθετες πληροφορίες όπως τα parameter sets αποτελούν τα non VCL units.

Τα parameter sets περιέχουν σημαντικές πληροφορίες επικεφαλίδας και διακρίνονται σε sequence parameter sets και picture parameter sets. Το sequence set αφορά μια κωδικοποιημένη ακολουθία βίντεο, η οποία είναι στην ουσία μια ακολουθία από κωδικοποιημένες εικόνες βίντεο. Το picture set σχετίζεται με την αποκωδικοποίηση μιας ή περισσότερων εικόνων οι οποίες αποτελούν την ακολουθία βίντεο. Για κάθε NAL unit υπάρχει ένα αναγνωριστικό το οποίο το συνδέει με το picture parameter set του, και κάθε picture parameter set έχει ένα αναγνωριστικό ώστε να μπορεί να υπάρχει η αναφορά στο σχετικό sequence parameter set. Αν υπάρχουν οι απαραίτητες πληροφορίες από τα parameter sets τότε μια ακολουθία βίντεο μπορεί να αποκωδικοποιηθεί ανεξάρτητα από οποιαδήποτε άλλη.

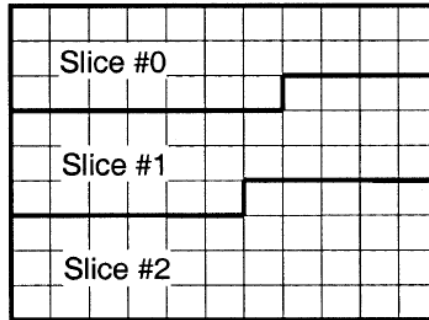
Video Coding Layer (VCL)

Βασίζεται στην τεχνική block-based hybrid video coding. Παρόλο που έχει αρκετά κοινά στοιχεία με τα προηγούμενα πρότυπα κωδικοποίησης όσον αφορά το σχεδιασμό του, παρουσιάζει κάποιες διαφορές αφού είναι πιο ευέλικτο από τα άλλα. Ο βασικός αλγόριθμος ο οποίος ακολουθείται κατά την κωδικοποίηση είναι η inter picture κωδικοποίηση μαζί με μετασχηματισμό του prediction residual. Κάθε πλαίσιο στο βίντεο αποτελείται από δύο interleaved πεδία, τα οποία καλούνται το top και το bottom πεδίο αντίστοιχα. Σε ένα interlaced πλαίσιο τα δύο αυτά πεδία μπορούν είτε να κωδικοποιηθούν ξεχωριστά είτε μαζί. Αντίθετα, ένα progressive πλαίσιο κωδικοποιείται ως ένα.

Μια ακολουθία από κωδικοποιημένες εικόνες αποτελεί μια κωδικοποιημένη ακολουθία βίντεο. Χρησιμοποιείται και πάλι η έννοια του macro block, με το κάθε macro block να περιέχει μια εικόνα των 16x16 samples για το στοιχείο Y και 8x8 samples για το καθένα από τα στοιχεία Cb και Cr. Τα samples του κάθε macro block προβλέπονται είτε χωρικά είτε χρονικά και το τελικό prediction residual signal μεταδίδεται πραγματοποιώντας transform coding. Κάθε στοιχείο χρώματος από αυτό το σήμα χωρίζεται σε μικρότερα μπλοκ και το κάθε μπλοκ μετασχηματίζεται. Στα coefficients του μετασχηματισμού πραγματοποιείται κβαντοποίηση και στη συνέχεια κωδικοποίηση πραγματοποιώντας κωδικοποίηση εντροπίας. Μια εικόνα μπορεί να χωρίζεται σε ένα ή περισσότερα slices, όπως παρουσιάζεται στο επόμενο σχήμα, και το κάθε slice μπορεί να αποκωδικοποιηθεί ανεξάρτητα από τα υπόλοιπα. Έτσι τα slices μιας εικόνας

μπορούν να αποστέλλονται και να λαμβάνονται με οποιαδήποτε σειρά. Αυτός ο μηχανισμός καλείται Arbitrary slice ordering (ASO) και μπορεί να βοηθήσει στην μείωση της end-to-end καθυστέρησης σε εφαρμογές πραγματικού χρόνου.

Ο διαχωρισμός μιας εικόνας σε slices παρουσιάζεται στο σχήμα που ακολουθεί .



Σχήμα 4.7: Διαχωρισμός εικόνας σε slices

Το H/264/AVC έχει τη δυνατότητα της ιδιότητας Flexible macroblock ordering(FMO). Με βάση αυτή ορίζονται τα slice groups τα οποία αποτελούνται από slices. Η δυνατότητα αυτή είναι χρήσιμη γιατί προσφέρει ευρωστία και ευελιξία στις τυχόν απώλειες δεδομένων που μπορεί να παρατηρηθούν. Στην περίπτωση που δε χρησιμοποιείται η FMO τότε μια εικόνα αποτελεί ένα slice group.

Υπάρχουν πέντε κατηγορίες στις οποίες χωρίζονται τα slices, οι οποίες είναι οι εξής:

I slice: Πραγματοποιείται intra-prediction για την κωδικοποίηση των macroblocks στο slice.

P slice: Επιπρόσθετα στην intra-prediction, μπορεί να πραγματοποιηθεί και inter-prediction για κάποια macroblocks στο slice, χρησιμοποιώντας ένα motion-compensated σήμα για την πρόβλεψη.

B slice: Σε κάποια macroblocks του slice μπορεί να χρησιμοποιηθεί inter-prediction για κάποια macroblocks στο slice, χρησιμοποιώντας δύο motion-compensated σήματα για την πρόβλεψη επιπρόσθετα στην πρόβλεψη που πραγματοποιείται στα P slices.

SP slice (Switching P): Η κωδικοποίηση ενός P slice με τέτοιον τρόπο ώστε να είναι αποδοτική η μετάβαση σε διαφορετικές προηγούμενα κωδικοποιημένες εικόνες.

SI slice: Χρήσιμο για να είναι δυνατή η τυχαία πρόσβαση και η ανάνηψη από τυχόν σφάλματα.

Το πρότυπο H.264/AVC όπως αναφέρθηκε σε προηγούμενο σημείο, υποστηρίζει κωδικοποίηση εντροπίας και πιο συγκεκριμένα καθορίζονται 2 μέθοδοι: η μέθοδος CAVLC(Context Adaptive Variable Length Coding) και η μέθοδος CABAC(Context Adaptive Binary Arithmetic Coding). Η μέθοδος CAVLC χρησιμοποιεί variable length codes και έχει περιορισμένη προσαρμοστικότητα. Η μέθοδος CABAC χαρακτηρίζεται από ψηλότερη πολυπλοκότητα από την πρώτη μέθοδο αλλά αποδίδει καλύτερα όσον αφορά την κωδικοποίηση. Χρησιμοποιεί arithmetic codes.

Επίσης, αυτό το πρότυπο καθορίζει τρία προφίλ, το baseline, το main και το extended. Το baseline εστιάζεται στην αποδοτικότητα της κωδικοποίησης και την ευρωστία με χαμηλή υπολογιστική πολυπλοκότητα. Το main δίνει προσοχή μόνο στην αποδοτικότητα της κωδικοποίησης ενώ το extended στην ευελιξία με υψηλή αποδοτικότητα κωδικοποίησης.

4.5.6.2 Επεκτάσεις στο πρότυπο-Fidelity Range Extensions (FRExt)

Το πρότυπο H.264 μπορούσε να εφαρμοστεί σε αρκετές εφαρμογές όμως δεν υποστήριζε κάποιες ιδιότητες που αφορούσαν εφαρμογές πιο απαιτητικές. Έτσι δημιουργήθηκαν κάποιες επεκτάσεις στο αρχικό πρότυπο, γνωστές ως Range Extensions (FRExt).

Οι επεκτάσεις που έγιναν αφορούσαν μεταξύ άλλων, ψηλότερους ρυθμούς δεδομένων και ψηλότερη ανάλυση όσον αφορά το χρώμα. Έτσι υπάρχει η δυνατότητα για 4:2:2 και 4:4:4 sampling και για προσαρμογή του μεγέθους του μπλοκ που χρησιμοποιείται στον μετασχηματισμό.

Με βάση τις επεκτάσεις αυτές δημιουργήθηκαν τέσσερα επιπρόσθετα προφίλ τα οποία είναι: High profile(HP), High 10 profile(Hi10P), High 4:2:2 profile(H422P) και High 4:4:4(H444P) υποστηρίζοντας 4:2:0, 4:2:0, 4:2:2 και 4:4:4 τύπο δειγματοληψίας αντίστοιχα για τα τρία στοιχεία Y, Cb και Cr.

4.5.6.3 Επέκταση Scalable Video Coding [12]

Η επέκταση αυτή πραγματοποιήθηκε ώστε να είναι δυνατή η κωδικοποίηση μιας ροής βίντεο η οποία είναι υψηλής ανάλυσης και η οποία αποτελείται από υπό-ροές χαμηλότερης ανάλυσης. Η κάθε υπό-ροή εξάγεται απορρίπτοντας ένα αριθμό πακέτων από την αρχική ροή βίντεο. Έτσι είναι δυνατή η μετάδοση των ροών με χαμηλότερη ανάλυση σε περίπτωση που έχουμε ένα δίκτυο στο οποίο παρατηρείται απώλεια δεδομένων ή στο οποίο οι συνθήκες οι οποίες επικρατούν μεταβάλλονται. Η κωδικοποίηση αυτή, του scalable video προσφέρει με αυτόν τον τρόπο προσαρμογή όσον αφορά είτε τον ρυθμό δεδομένων, είτε το format είτε το power.

Κάθε υπό-ροή βίντεο θα έχει μια χαμηλότερη ανάλυση είτε χωρικά(spatial scalability) είτε χρονικά(temporal scalability) είτε χαμηλότερη ακρίβεια και ποιότητα (fidelity scalability) σε σύγκριση με την αρχική ροή βίντεο. Στην περίπτωση που υπάρχει χρονική κλιμάκωση τότε αυτό σημαίνει πως κάθε υπό-ροή έχει μειωμένο ρυθμό δεδομένων και ολόκληρες εικόνες μπορεί να απορρίπτονται από την αρχική ροή βίντεο. Πιο συγκεκριμένα υπάρχει ένα βασικό temporal επίπεδο και ένας αριθμός από ενισχυτικά temporal επίπεδα. Όταν το μέγεθος εικόνας είναι μειωμένο τότε έχουμε χωρική κλιμάκωση. Αυτό σημαίνει πως τα δεδομένα όπως επίσης και τα δείγματα τα οποία έχουν αποκωδικοποιηθεί και είναι χαμηλότερης ακρίβειας, μπορούν να χρησιμοποιηθούν στην πρόβλεψη των δεδομένων και των δειγμάτων που παρουσιάζουν υψηλότερη ακρίβεια. Και πάλι υπάρχουν πολλαπλά επίπεδα από τα οποία το καθένα αντιπροσωπεύει ένα βαθμό χωρικής ανάλυσης. Στην κλιμάκωση όσον αφορά την ποιότητα, γνωστή και ως quality ή SNR κλιμάκωση, η χρονική και χωρική ανάλυση παραμένουν τα ίδια όπως στην αρχική ροή με μειωμένη όμως ποιότητα. Όπως και στη χωρική κλιμάκωση, στην πρόβλεψη δεδομένων υψηλότερης ακρίβειας μπορούν να χρησιμοποιηθούν δεδομένα και δείγματα χαμηλότερης ακρίβειας. Επιπρόσθετα η κλιμάκωση που παρουσιάζει μια ροή μπορεί να συνδυάζει τα τρία πιο πάνω είδη κλιμάκωσης.

Με βάση αυτή την επέκταση δημιουργήθηκαν ακόμα τρία προφίλ: το Scalable Baseline, το Scalable High και το Scalable High Intra. Το πρώτο χρησιμοποιείται σε εφαρμογές όπως βίντεο-συνδιασκέψεις, το δεύτερο προφίλ για εφαρμογές broadcasting και streaming και το τελευταίο για εφαρμογές production.

Η συγκεκριμένη επέκταση στο πρότυπο H.264/AVC είναι σημαντική αφού η scalable κωδικοποίηση βίντεο είναι ιδιαίτερα χρήσιμη σε αρκετές εφαρμογές πολυμέσων όπως το streaming και το broadcasting.

ΚΕΦΑΛΑΙΟ 5

Multimedia Δεδομένα

5.1. Εισαγωγή

5.2. Παραδείγματα Εφαρμογών Πολυμέσων

5.3. Real Time Streaming Protocol (RTSP)

5.4. Μετάδοση πολυμεσικού περιεχομένου

5.5. Multicast Μετάδοση Πολυμεσικού Περιεχομένου

5.6. Προσαρμοστική Multicast Μετάδοση Πολυμεσικού Περιεχομένου

5.1 Εισαγωγή

Τα πολυμέσα είναι ένας συνδυασμός από διάφορους τύπους μέσων όπως είναι το κείμενο, ο ήχος, η εικόνα και το βίντεο. Οι εφαρμογές πολυμέσων χρησιμοποιούνται ευρέως σε διάφορους τομείς όπως η ψυχαγωγία για παράδειγμα και έτσι η όλο και ευρύτερη εξάπλωσή τους οδηγεί στην ανάγκη για ικανοποίηση των απαιτήσεων που μπορεί να έχουν. Υπάρχουν απαιτήσεις οι οποίες αφορούν χρονικούς περιορισμούς και απώλεια δεδομένων. Όσον αφορά τους χρονικούς περιορισμούς οι εφαρμογές πολυμέσων είναι ευαίσθητες στις καθυστερήσεις ενώ στην περίπτωση της απώλειας δεδομένων παρουσιάζουν μεγαλύτερη ανοχή.

5.2 Παραδείγματα Εφαρμογών Πολυμέσων [1]

5.2.1 Streaming Stored Audio και Video

Τα δεδομένα ήχου και βίντεο είναι συμπιεσμένα και είναι αποθηκευμένα σε έναν εξυπηρετητή από τον οποίο μπορούν να πραγματοποιούνται αιτήσεις για αυτά από το χρήστη που τα χρειάζεται. Αυτού του είδους εφαρμογές είναι πολύ γνωστές και χρησιμοποιούνται πολύ συχνά σήμερα μέσω του διαδικτύου. Υπάρχουν πολλές

ιστοσελίδες οι οποίες προσφέρουν ροές αποθηκευμένου ήχου ή βίντεο όπως για παράδειγμα είναι το YouTube.

Οι εφαρμογές αυτές έχουν τρία κύρια χαρακτηριστικά:

Stored media

Όπως αναφέρθηκε προηγουμένως, το πολυμεσικό περιεχόμενο βρίσκεται αποθηκευμένο σε έναν εξυπηρετητή. Αυτό δίνει τη δυνατότητα στο χρήστη να εκτελέσει διάφορες λειτουργίες πάνω στο πολυμεσικό περιεχόμενο όπως να κάνει παύση, rewind ή fast-forward.

Streaming

Το πολυμεσικό περιεχόμενο μπορεί να παρουσιαστεί στο χρήστη μερικά δευτερόλεπτα μετά από την στιγμή που ο χρήστης αρχίζει να λαμβάνει τα δεδομένα. Έτσι αυτό σημαίνει πως ο χρήστης μπορεί να παίζει τον ήχο ή το βίντεο το οποίο λαμβάνει και παράλληλα να συνεχίζει να λαμβάνει μέρος από το πολυμεσικό αυτό περιεχόμενο. Δεν είναι απαραίτητο να πραγματοποιηθεί πλήρης λήψη του αρχείου για να μπορεί αυτό να παρουσιαστεί στο χρήστη. Με αυτόν τον τρόπο είναι δυνατόν να αποφευχθούν καθυστερήσεις που τυχόν να σημειωθούν. Αυτή η τεχνική ονομάζεται streaming.

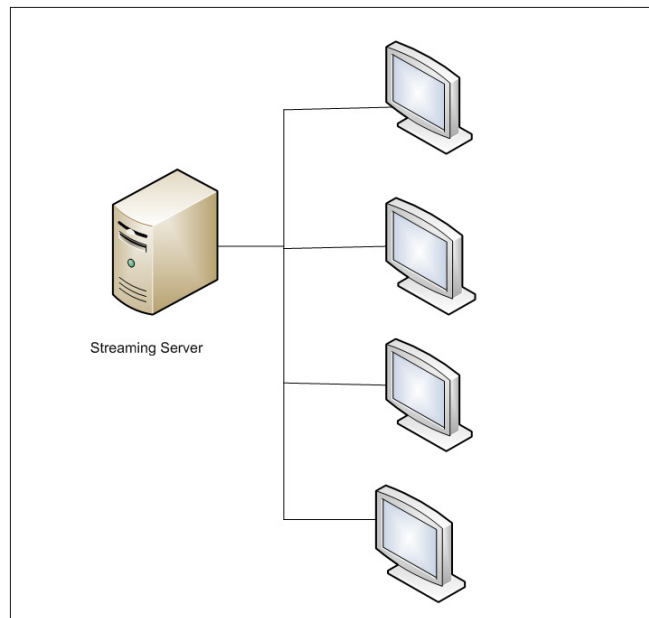
Η τεχνική αυτή για μετάδοση media επιτρέπει σε ένα streaming εξυπηρετητή να στείλει δεδομένα μέσω ενός δικτύου σε έναν client σε πραγματικό χρόνο. Ο client δε χρειάζεται να κατεβάσει κανένα αρχείο στο σκληρό του δίσκο για να μπορέσει να παρακολουθήσει τα media. Κάποια παραδείγματα client που λαμβάνουν και παρουσιάζουν ροές ήχου ή βίντεο είναι η εφαρμογή QuickTime Player από την Apple και η εφαρμογή Windows Media Player από την Microsoft.

Έτσι οι ροές του πολυμεσικού περιεχομένου στέλνονται μέσω του streaming εξυπηρετητή στο διαδίκτυο ή σε ένα τοπικό δίκτυο. Ένας client ζητά από τον streaming εξυπηρετητή ροές βίντεο ή ήχου και αυτός τα μεταδίδει ως ανταπόκριση στις αιτήσεις αυτές.

Το πολυμεσικό περιεχόμενο κωδικοποιείται και συμπιέζεται ώστε να μεταδοθεί στη συνέχεια. Αυτό συμβαίνει γιατί η μετάδοση των δεδομένων αυτών θα πρέπει να

συνυπάρχει μαζί με άλλες μεταδόσεις στο διαδίκτυο και αυτό σημαίνει πως το εύρος ζώνης για την μετάδοση αυτή θα είναι περιορισμένο.

Στο επόμενο σχήμα παρουσιάζεται μια τοπολογία δικτύου η οποία αποτελείται από τον streaming εξυπηρετητή ο οποίος ανταποκρίνεται στις αιτήσεις των τελικών χρηστών για τις ροές βίντεο που επιθυμούν.



Σχήμα 5.1: Streaming

Continuous playout

Οι εφαρμογές πολυμέσων είναι ευαίσθητες όσον αφορά τις καθυστερήσεις και για αυτόν τον λόγο πρέπει το πολυμεσικό περιεχόμενο να φτάνει στο χρήστη έγκαιρα ώστε να μην παρουσιάζονται καθυστερήσεις. Η παρουσίαση του ήχου ή του βίντεο θα πρέπει να γίνεται σύμφωνα με το χρονισμό του περιεχομένου το οποίο είναι αποθηκευμένο στον εξυπηρετητή. Αν τα δεδομένα δεν ληφθούν από τον εξυπηρετητή την κατάλληλη στιγμή τότε θα δημιουργηθεί πρόβλημα αφού θα παρατηρηθεί καθυστέρηση.

5.2.2 Streaming Live Audio και Video

Αυτού του είδους εφαρμογή εξυπηρετεί τους ίδιους σκοπούς και έχει πολλές ομοιότητες με την παραδοσιακή υπηρεσία ραδιοφωνου και τηλεόρασης. Σε αυτήν την περίπτωση όμως η μετάδοση πραγματοποιείται μέσω του διαδικτύου, δίνοντας έτσι τη

δυνατότητα σε οποιοδήποτε χρήστη να λαμβάνει τη συγκεκριμένη μετάδοση από οποιοδήποτε και αν βρίσκεται. Τέτοιου είδους εφαρμογές είναι γνωστές και ως Internet radio και IPTV.

Διαφέρουν από τις εφαρμογές που περιγράψαμε πιο πριν και οι οποίες μεταδίδουν αποθηκευμένο ήχο και βίντεο γιατί σε αυτήν την περίπτωση τα δεδομένα δεν είναι αποθηκευμένα. Αυτό σημαίνει πως ο χρήστης δεν μπορεί για παράδειγμα να πραγματοποιήσει fast-forward στο πολυμεσικό περιεχόμενο. Ο χρήστης όμως μπορεί να κάνει παύση ή rewind. Ένα κοινό χαρακτηριστικό που μοιράζεται μαζί με τις streaming εφαρμογές αποθηκευμένου ήχου και βίντεο είναι το continuous play out.

5.2.3 Real Time Interactive Audio και Video

Οι χρήστες μπορούν να επικοινωνήσουν μεταξύ τους σε πραγματικό χρόνο χρησιμοποιώντας ήχο και βίντεο. Οι εφαρμογές οι οποίες χρησιμοποιούν μόνο ήχο στο διαδίκτυο είναι γνωστές ως Internet telephony και δίνει τη δυνατότητα για ομαδική επικοινωνία μεταξύ άλλων. Οι εφαρμογές οι οποίες χρησιμοποιούν βίντεο είναι γνωστές ως βίντεο-τηλεδιασκέψεις (video conferencing).

5.3 Real Time Streaming Protocol (RTSP) [1], [13]

Το πρωτόκολλο αυτό δίνει τη δυνατότητα στο χρήστη να μπορεί να ελέγχει και να διαχειρίζεται το playback με το να του επιτρέπει να πραγματοποιεί ενέργειες όπως η παύση, η γρήγορη μετάβαση προς τα εμπρός και η μετάβαση προς τα πίσω. Μπορεί να ελέγχει μια ή περισσότερες συγχρονισμένες ροές από βίντεο χωρίς όμως να σχετίζεται με την παράδοση των δεδομένων αυτών. Το RTSP χρησιμοποιείται από τον media player και τον εξυπηρετητή (server) για να ανταλλάζουν playback πληροφορίες ελέγχου.

Είναι ανεξάρτητο από τον τρόπο μεταφοράς των δεδομένων και το πώς αυτά θα αποθηκεύονται από τον media player. Τα βίντεο δεδομένα μπορούν να συμπιεστούν χωρίς να επηρεάζουν το πρωτόκολλο αυτό, όπως επίσης δεν επηρεάζεται από τον τρόπο που πραγματοποιείται η ενθυλάκωση τους σε πακέτα ώστε να μεταδοθούν.

Αυτό το πρωτόκολλο κάνει δυνατή την ανάκτηση των media από τον κατάλληλο εξυπηρετητή, όπως επίσης και τη συμμετοχή ενός media εξυπηρετητή στο λεγόμενο

conference, το οποίο είναι μια ροή πολυμεσικού περιεχομένου στην οποία συμμετέχουν πολλές οντότητες.

Ο εξυπηρετητής και η πλευρά του client ανταλλάζουν RTSP μηνύματα για να είναι δυνατή η επικοινωνία τους. Τα RTSP μηνύματα αυτά χρησιμοποιούν το port 544, το οποίο διαφέρει από αυτό που χρησιμοποιείται για τη ροή δεδομένων. Κάποια βασικά παραδείγματα μηνυμάτων που λειτουργούν είτε ως αιτήσεις είτε ως αποκρίσεις και τα οποία σχετίζονται με τις συγκεκριμένες μεθόδους που ορίζει το πρωτόκολλο RTSP είναι τα ακόλουθα: RTSP DESCRIBE, RTSP SETUP, RTSP OK, RTSP PLAY, RTSP PAUSE και RTSP TEARDOWN. Πιο συγκεκριμένα ένα μήνυμα αίτησης μπορεί να βασίζεται σε μια από τις εξής μεθόδους : DESCRIBE, ANNOUNCE, GET_PARAMETER, OPTIONS, PAUSE , PLAY, RECORD, REDIRECT, SETUP, SET_PARAMETER και TEARDOWN. Μια συνοπτική περιγραφή του ρόλου κάθε μεθόδου θα παρουσιαστεί στη συνέχεια. Ένα μήνυμα απόκρισης αποτελείται από έναν κωδικό κατάστασης ο οποίος δείχνει το αποτέλεσμα της αίτησης στην οποία αντιστοιχεί. Στη συνέχεια παρουσιάζεται μια μορφή μηνύματος RTSP PLAY.

```
PLAY rtsp://192.168.2.14/sample_100kbit.mov RTSP/1.0
CSeq: 4
Range: npt=0.000000-70.000000
x-prebuffer: maxtime=2.000000
x-transport-options: late-tolerance=10
Session: 130197638632959
User-Agent: QuickTime/7.6.6 (qtver=7.6.6;os=Windows NT 5.1Service Pack 3)
```

Σχήμα 5.2: RTSP Μήνυμα PLAY

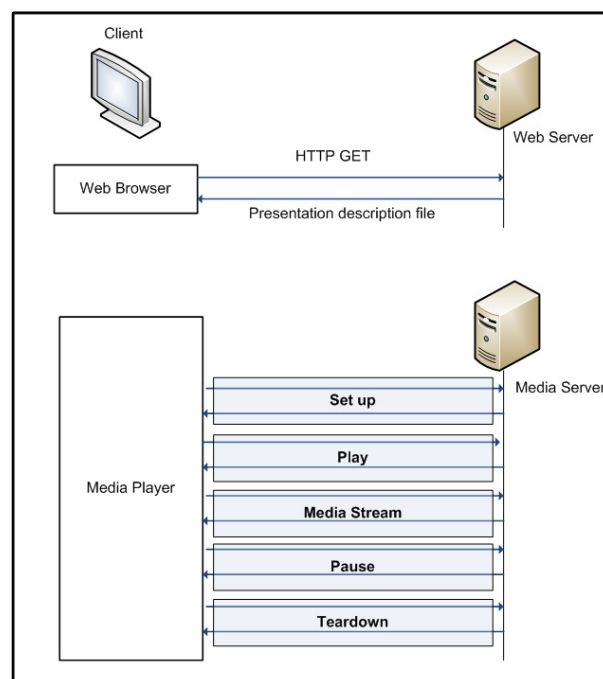
Όπως βλέπουμε, το RTSP PLAY μήνυμα, όπως και τα υπόλοιπα RTSP μηνύματα αποτελούνται από διάφορες επικεφαλίδες. Στο συγκεκριμένο όμως, παρατηρούμε ανάμεσα σε άλλα στοιχεία, το βίντεο που ζητά ένας χρήστης, τη διάρκεια του αρχείου η οποία δίνεται από την επικεφαλίδα Range υπό την μορφή χρονικού διαστήματος και επίσης η εφαρμογή που τρέχει ως η client πλευρά για να παρουσιάσει τη ροή βίντεο που ζητήθηκε.

Η κάθε ροή δεδομένων καθορίζεται από ένα RTSP URL το οποίο έχει τη μορφή rtsp://. Ο client μπορεί να ανακτήσει ένα αρχείο περιγραφής παρουσίασης (presentation description file) το οποίο αφορά τη συγκεκριμένη παρουσίαση και τις διάφορες

ιδιότητες οι οποίες τη χαρακτηρίζουν. Το αρχείο αυτό μπορεί να περιέχει πληροφορίες για τις ροές που αφορούν την παρουσίαση όπως για παράδειγμα κωδικοποιήσεις. Αυτή η διαδικασία μπορεί να πραγματοποιηθεί χρησιμοποιώντας έναν web browser για παράδειγμα. Μια παρουσίαση μπορεί να αποτελείται από διάφορες ροές πολυμεσικού περιεχομένου οι οποίες μπορεί να βρίσκονται σε διαφορετικούς εξυπηρετητές με βάση πάντα το τι ορίζεται στο URL που αντιστοιχεί στην κάθε μια. Αφού ζητηθεί το presentation description file, στη συνέχεια ένας web εξυπηρετητής θα ανταποκριθεί σε αυτήν την αίτηση ενθυλακώνοντας το συγκεκριμένο αρχείο σε ένα HTTP μήνυμα απόκρισης. Όταν το λάβει ο web browser τότε ο media player ο οποίος είναι εγκατεστημένος στην πλευρά του client θα ενεργοποιηθεί ώστε να είναι δυνατό να παρουσιάσει το ζητούμενο πολυμεσικό περιεχόμενο.

Αυτή η λειτουργία παρουσιάζεται στο επόμενο σχήμα. Όπως θα παρατηρήσουμε ο media player και ο media εξυπηρετητής ανταλλάζουν μια ακολουθία από τα RTSP μηνύματα τα οποία αναφέρθηκαν πιο πάνω.

Αρχικά πραγματοποιείται η εγκαθίδρυση της RTSP συνόδου αφού ο εξυπηρετητής πρέπει να δεσμεύσει τους απαραίτητους πόρους για την ροή και να ξεκινήσει τη σύνοδο. Στη συνέχεια ξεκινά η μετάδοση των δεδομένων. Στο τέλος οι πόροι οι οποίοι δεσμεύτηκαν πρέπει να αποδεσμευτούν.



Σχήμα 5.3: Λειτουργία RTSP πρωτοκόλλου

5.3.1 Μέθοδοι RTSP

Το πρωτόκολλο RTSP ορίζει κάποιες μεθόδους για να είναι δυνατή η λειτουργία του. Οι μέθοδοι αυτές χρησιμοποιούνται στα RTSP μηνύματα και συγκεκριμένα παρουσιάζονται στην πρώτη γραμμή κάθε μηνύματος. Στη συνέχεια περιγράφεται η κάθε μέθοδος συνοπτικά.

DESCRIBE: Χρησιμοποιείται για την ανάκτηση της περιγραφής μιας παρουσίασης από τον εξυπηρετητή με βάση το URL που ακολουθεί τη μέθοδο μέσα στο μήνυμα.

ANNOUNCE: Μπορεί να χρησιμοποιηθεί είτε από τον εξυπηρετητή είτε από την άλλη πλευρά. Στην πρώτη περίπτωση θα ανανεώσει την περιγραφή της συνόδου σε πραγματικό χρόνο ενώ στην δεύτερη περίπτωση αφορά την περιγραφή μιας παρουσίασης .

GET_PARAMETER: Χρησιμοποιείται για την ανάκτηση της τιμής κάποιας παραμέτρου η οποία σχετίζεται με μια ροή δεδομένων η οποία καθορίζεται από το URL.

OPTIONS: Χρησιμοποιείται για την ένδειξη των παραμέτρων που γίνονται αποδεκτές από τις δύο πλευρές.

PAUSE: Χρησιμοποιείται με σκοπό να διακοπεί προσωρινά η ροή των δεδομένων.

PLAY: Χρησιμοποιείται έτσι ώστε ο εξυπηρετητής να ξεκινήσει την αποστολή των δεδομένων σύμφωνα με τα όσα καθορίστηκαν με το SETUP.

RECORD: Χρησιμοποιείται για την αποθήκευση των δεδομένων με βάση την περιγραφή παρουσίασης.

REDIRECT: Χρησιμοποιείται ώστε να ενημερωθεί ο παραλήπτης για την ανάγκη να συνδεθεί με τον εξυπηρετητή σε διαφορετική τοποθεσία.

SETUP: Χρησιμοποιείται για τη δέσμευση των απαραίτητων πόρων, την έναρξη της RTSP συνόδου αλλά και για να καθοριστεί το πώς θα γίνει η μεταφορά των δεδομένων.

SET_PARAMETER: Χρησιμοποιείται για τον καθορισμό της τιμής κάποιας παραμέτρου η οποία σχετίζεται με μια ροή δεδομένων η οποία καθορίζεται από το URL.

TEARDOWN: Χρησιμοποιείται για τον τερματισμό της ροής δεδομένων απελευθερώνοντας τους πόρους που είχαν δεσμευτεί αρχικά.

5.4 Μετάδοση πολυμεσικού περιεχομένου

5.4.1 Εισαγωγή [1], [14], [15]

Υπάρχουν πολλές εφαρμογές σήμερα οι οποίες βασίζονται στην μετάδοση πολυμεσικού περιεχομένου όπως το Voice over IP (VoIP), οι βίντεο-τηλεδιασκέψεις και το video on demand. Αυτές οι εφαρμογές όμως έχουν συγκεκριμένες απαιτήσεις από τα κατώτερα επίπεδα σχετικά με έννοιες όπως είναι η απώλεια δεδομένων, η καθυστέρηση και το jitter. Όπως είναι γνωστό δύο ευρέως χρησιμοποιημένα πρωτόκολλα μεταφοράς δεδομένων είναι το TCP και το UDP. Όμως, τα πρωτόκολλα αυτά δεν είναι σχεδιασμένα για να καλύπτουν ανάγκες όπως αυτές που προκύπτουν από τη χρήση των πιο πάνω εφαρμογών. Αυτό οδηγεί στην ανάγκη για πρωτόκολλα τα οποία να μπορούν να μεταδίδουν και να παραδίδουν δεδομένα με ψηλό εύρος ζώνης, να υποστηρίζουν την multicast μετάδοση δεδομένων και να μπορούν να διαχειρίζονται την ποιότητα της μετάδοσης. Το πρωτόκολλο το οποίο θα είναι υπεύθυνο για την μετάδοση του πολυμεσικού περιεχομένου θα πρέπει να είναι αρκετά γρήγορο ώστε να μπορεί να καλύψει τις ανάγκες της χρησιμοποιημένης εφαρμογής. Πολλές multimedia εφαρμογές καθορίζουν μεγάλο αριθμό από παραλήπτες και όπως αναφέραμε σε προηγούμενο κεφάλαιο, η μετάδοση των δεδομένων με χρήση unicast συνδέσεων προς τους παραλήπτες δεν χρησιμοποιεί αποδοτικά τους διαθέσιμους πόρους του δικτύου. Έτσι η υποστήριξη της multicast μετάδοσης δεδομένων είναι πολύ σημαντική. Επιπρόσθετα είναι σημαντικό να υπάρχει η δυνατότητα για τον καθορισμό των απαιτήσεων που αφορούν την ποιότητα υπηρεσίας.

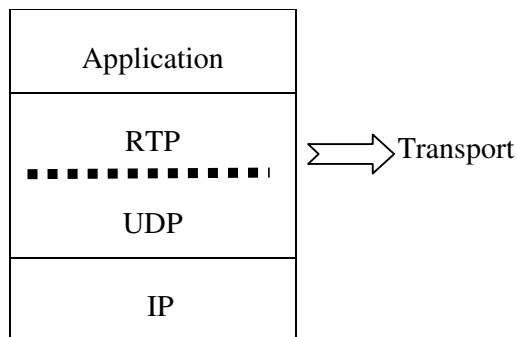
Έτσι αναπτύχθηκαν τα πρωτόκολλα RTP (Real-Time Transport Protocol) / RTCP (RTP Control Protocol) τα οποία θα παρουσιαστούν στη συνέχεια.

5.4.2 Real Time Πρωτόκολλα RTP και RTCP [1],[14],[15]

5.4.2.1 Εισαγωγή

Η μετάδοση των multimedia δεδομένων βασίζεται στα πρωτόκολλα RTP και RTCP, τα οποία αναπτύχθηκαν από την IETF. Το RTP χρησιμοποιείται για την μετάδοση του πολυμεσικού περιεχομένου από τον αποστολέα προς τον παραλήπτη ενώ το RTCP χρησιμοποιείται από τον παραλήπτη για να δώσει αναφορά όσον αφορά την ποιότητα της μετάδοσης στον αποστολέα. Τα δύο αυτά πρωτόκολλα δίνουν τη δυνατότητα για one way επικοινωνία όπως συμβαίνει για παράδειγμα στο video on demand, αλλά και για two way επικοινωνία όπως παρατηρείται στις βίντεο-τηλεδιασκέψεις. Έχουν σχεδιαστεί ώστε να λειτουργούν σε συνεργασία το ένα με το άλλο και τρέχουν πάνω από UDP. Αυτό συμβαίνει λόγω του ότι το UDP είναι πιο γρήγορο πρωτόκολλο μεταφοράς αφού δεν χρειάζεται οποιαδήποτε σύνδεση, δεν προσφέρει αξιόπιστη μεταφορά δεδομένων και δεν πραγματοποιεί έλεγχο συμφόρησης. Αντίθετα το TCP είναι ένα connection-oriented πρωτόκολλο το οποίο προσφέρει αξιόπιστη μετάδοση δεδομένων και έλεγχο συμφόρησης. Η αξιόπιστη μετάδοση δεδομένων βασίζεται στην επαναμετάδοση των χαμένων πακέτων γεγονός όμως που οδηγεί σε καθυστερήσεις. Έτσι αυτό που είναι πιο σημαντικό είναι το πολυμεσικό περιεχόμενο να μεταδοθεί την κατάλληλη στιγμή και χωρίς καθυστερήσεις παρά να έχουμε μια αξιόπιστη μετάδοση. Το RTP και το RTCP χρησιμοποιούν συνεχόμενα transport-layer ports, με το RTP να χρησιμοποιεί το μικρότερο και το οποίο είναι περιττός αριθμός.

Το πρωτόκολλο RTP μπορεί να ληφθεί και ως ένα υπό-επίπεδο του επιπέδου μεταφοράς όπως φαίνεται και από το ακόλουθο σχήμα.



Data Link
Physical

Σχήμα 5.4: Επίπεδα

5.4.2.2 Πρωτόκολλο RTP

Το RTP χρησιμοποιείται για την ανταλλαγή πολυμεσικού περιεχομένου αφού επιτρέπει τη μετάδοση ήχου με μορφή mp3 για παράδειγμα, αλλά και τη μετάδοση βίντεο σε μορφές όπως MPEG και H.263.

Στην μετάδοση του πολυμεσικού περιεχομένου, αρχικά ο αποστολέας τοποθετεί το περιεχόμενο μέσα σε ένα RTP πακέτο - με χρήση ενθυλάκωσης – και στη συνέχεια τοποθετεί το RTP πακέτο μέσα σε ένα UDP πακέτο (UDP segment). Αυτό μεταδίδεται και φτάνει τελικά στον παραλήπτη ο οποίος θα εξάγει το RTP πακέτο από το UDP segment και ακολούθως από αυτό θα εξάγει το πολυμεσικό περιεχόμενο. Το πολυμεσικό περιεχόμενο θα αποκωδικοποιηθεί ώστε να μπορεί στη συνέχεια να παρουσιαστεί με τη βοήθεια ενός media player. Το RTP πακέτο αποτελείται από το πολυμεσικό περιεχόμενο και την RTP επικεφαλίδα η οποία αποτελείται από σημαντικά πεδία, από τα οποία για κάποια θα δοθεί μια συνοπτική περιγραφή στη συνέχεια. Τα RTP πακέτα χρησιμοποιούνται και σε unicast αλλά και σε multicast μεταδόσεις αφού όπως ήδη αναφέρθηκε είναι πολύ σημαντική η υποστήριξη της multicast μετάδοσης δεδομένων. Οι RTP multimedia ροές οι οποίες ανήκουν μαζί διαμορφώνουν μια RTP σύνοδο.

Το πρωτόκολλο RTP δεν εγγυάται για την έγκαιρη παράδοση των δεδομένων και για την σωστή σειρά παράδοσης τους. Επίσης, δεν υπάρχουν εγγυήσεις όσον αφορά την ποιότητα υπηρεσίας.

Αυτό το πρωτόκολλο προσφέρει διάφορες υπηρεσίες στις εφαρμογές πραγματικού χρόνου αφού δίνει τη δυνατότητα για τον καθορισμό και την αναγνώριση του payload type, την σειριακή αρίθμηση των πακέτων (sequential numbering) και τη χρήση timestamps.

Στη συνέχεια δίνεται μια συνοπτική περιγραφή των σημαντικών πεδίων από τα οποία αποτελείται η RTP επικεφαλίδα:

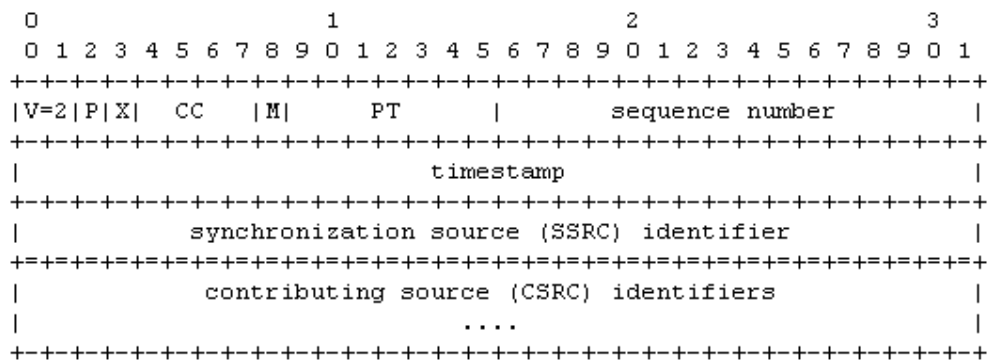
Payload type: Έχει μέγεθος 7 bits και στην περίπτωση που το πολυμεσικό περιεχόμενο είναι βίντεο, δείχνει τον τύπο της κωδικοποίησης βίντεο που χρησιμοποιείται, όπως για παράδειγμα MJPEG και H.261. Υπάρχει δυνατότητα για αλλαγή κατά τη διάρκεια μιας συνόδου.

Sequence number: Έχει μέγεθος 16 bits. Η τιμή του πεδίου αυτού αυξάνεται κατά ένα κάθε φορά που θα πραγματοποιηθεί αποστολή ενός RTP πακέτου. Το RTP τρέχει πάνω από UDP και έτσι δεν υπάρχει εγγύηση ότι τα πακέτα θα μεταδοθούν με τη σωστή σειρά. Ο παραλήπτης όμως χρησιμοποιώντας αυτό το πεδίο μπορεί να βρει τη σωστή σειρά των πακέτων. Επιπρόσθετα, μπορεί να το χρησιμοποιήσει για να ανιχνεύσει τυχόν απώλεια στα δεδομένα.

Timestamp: Έχει μέγεθος 32 bits και παρέχει χρήσιμες πληροφορίες στις real time εφαρμογές. Το timestamp εισάγεται από τον παραλήπτη και καθορίζει το πως πρέπει να παρουσιαστούν τα δεδομένα. Με βάση αυτό ο παραλήπτης μπορεί να ξανακατασκευάσει τα αρχικά δεδομένα, να αφαιρέσει το packet jitter και να συγχρονίσει ξεχωριστές ροές δεδομένων όπως για παράδειγμα ήχο και βίντεο μαζί.

Synchronization source identifier (SSRC): Έχει μέγεθος 32 bits και δείχνει τον αποστολέα του RTP stream. Είναι στην ουσία ένας τυχαίος αριθμός ο οποίος ανατίθεται στην αρχή μιας νέας ροής. Κάθε RTP σύνοδος έχει μοναδικό SSRC.

Στο ακόλουθο σχήμα παρουσιάζεται η δομή της επικεφαλίδας ενός RTP πακέτου.



Σχήμα 5.5: Δομή της RTP επικεφαλίδας

5.4.2.3 Πρωτόκολλο RTCP

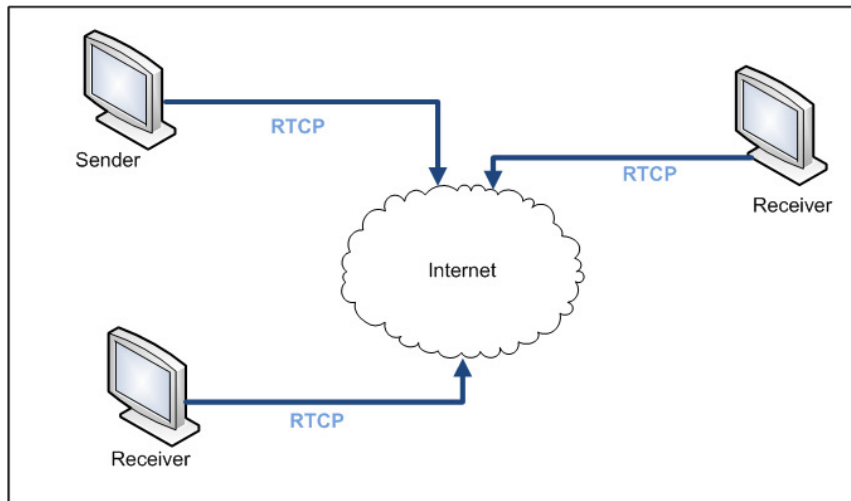
Το RTCP είναι το πρωτόκολλο ελέγχου του RTP. Κάθε συμμετέχοντας σε μια RTP σύνοδο μεταδίδει σε όλους τους άλλους συμμετέχοντες στη σύνοδο RTCP πακέτα χρησιμοποιώντας μια IP multicast διεύθυνση η οποία ανταποκρίνεται στην συγκεκριμένη σύνοδο.

Τα RTCP πακέτα δεν περιέχουν πολυμεσικό περιεχόμενο όπως συμβαίνει με τα RTP πακέτα. Περιέχουν χρήσιμες πληροφορίες και στοιχεία όσον αφορά την εφαρμογή για να μπορεί να ενημερωθεί ο αποστολέας για την ποιότητα της υπηρεσίας. Πιο συγκεκριμένα δίνει πληροφορίες σχετικές με τις συνθήκες που επικρατούν στο εν λόγω δίκτυο και σχετικές με την ποιότητα των δεδομένων που λαμβάνονται από τον παραλήπτη. Έτσι παρέχεται ένα είδος ανατροφοδότησης στον αποστολέα σχετικά με την επίδοση της εφαρμογής και του δικτύου. Αυτό είναι χρήσιμο για παράδειγμα όταν χρησιμοποιούνται εφαρμογές οι οποίες προσαρμόζουν είτε το είδος συμπίεσης είτε την ποιότητα της ροής ανάλογα με την συμφόρηση που παρατηρείται στο δίκτυο ή σε περιπτώσεις που υπάρχει κάποιο πρόβλημα εκ μέρους του δικτύου.

Επίσης το RTCP πρωτόκολλο παρέχει τρόπους για να γίνεται ο συγχρονισμός μεταξύ διαφορετικών media ροών οι οποίες προέρχονται από τον ίδιο αποστολέα. Αυτό πραγματοποιείται με τη χρήση του “canonical name” (CNAME). Κάθε αποστολέας έχει ένα CNAME και σχετίζεται με τα διάφορα SSRC τα οποία χρησιμοποιεί αυτός ο αποστολέας

Επιπρόσθετα, όπως θα δούμε και στη συνέχεια, το RTCP προσφέρει τη δυνατότητα για την αναγνώριση όλων των οντοτήτων που συμμετέχουν σε ένα RTP session καθώς επίσης και για την παροχή πληροφοριών σχετικά με αυτούς.

Στην περίπτωση που στέλλονται πολλά RTCP πακέτα συνεχώς από κάθε μέλος ενός multicast group τότε θα δημιουργηθεί πρόβλημα εφόσον θα καταναλώνεται ένα σημαντικό μέρος από το διαθέσιμο εύρος ζώνης για αυτού του είδους κίνησης δεδομένων. Το RTCP καθορίζει όμως τρόπο ώστε να μπορεί να μεταβάλλεται ο ρυθμός με τον οποίο αποστέλλονται αυτά τα πακέτα όταν ο αριθμός των μελών που ανήκουν σε ένα multicast group αυξάνεται.



Σχήμα 5.6: RTCP Πακέτα

Το RTCP καθορίζει διάφορους τύπους πακέτων που χρησιμοποιούνται και μπορεί να στέλλονται είτε από τον αποστολέα είτε από κάποιον παραλήπτη, όπως φαίνεται και από το πιο πάνω σχήμα:

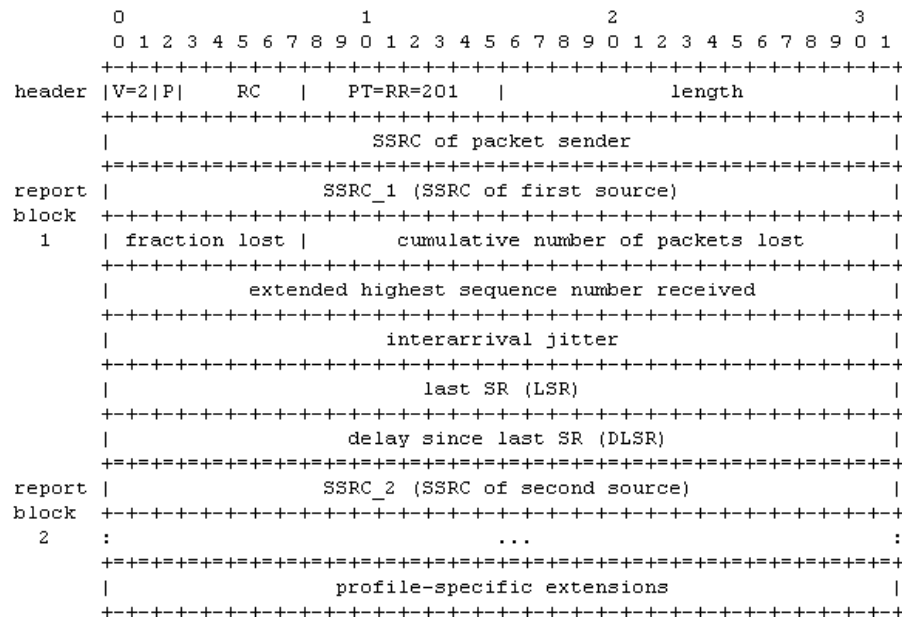
Αναφορές παραλήπτη (Receiver reports): Αποστέλλονται από κάθε παραλήπτη για κάθε RTP stream το οποίο λαμβάνει, στο multicast δένδρο το οποίο έχει όλους τους συμμετέχοντες της RTP συνόδου. Η αναφορά παραλήπτη, όπως θα δούμε στη συνέχεια, είναι αυτή που παρέχει μια από τις πληροφορίες που χρειάζεται ο αλγόριθμος που θέλουμε να υλοποιήσουμε. Τα πιο σημαντικά πεδία από τα οποία αποτελείται είναι:

- Το SSRC του RTP stream για το οποίο δημιουργήθηκε η συγκεκριμένη αναφορά.
- Fraction lost: Είναι το fraction των RTP πακέτων τα οποία έχουν χαθεί από τη χρονική στιγμή στην οποία στάλθηκε το προηγούμενο Receiver Report πακέτο. Η τιμή του πεδίου αυτού ορίζεται από τη διαίρεση του αριθμού των πακέτων που έχουν χαθεί δια τον αριθμό των πακέτων τα οποία είναι αναμενόμενο να φτάσουν. Αν η τιμή αυτού του πεδίου δίνει ενδείξεις ότι υπάρχει μεγάλη απώλεια πακέτων τότε ο αποστολέας μπορεί να μειώσει τον ρυθμό κωδικοποίησης τον οποίο χρησιμοποιεί.
- Cumulative number of packets lost: Ο συνολικός αριθμός των RTP πακέτων τα οποία έχουν χαθεί. Ο αριθμός αυτός λαμβάνεται με την αφαίρεση του αριθμού

των πακέτων τα οποία έχουν ληφθεί από τον συνολικό αριθμό από παλέτα τα οποία ήταν αναμενόμενα. Ο αριθμός των αναμενόμενων πακέτων προκύπτει από το τελευταίο extended sequence number το οποίο λήφθηκε πλν το αρχικό sequence number το οποίο έχει ληφθεί.

- Το τελευταίο sequence number το οποίο λήφθηκε στην ροή των RTP πακέτων
- Το jitter το οποίο είναι μια εκτίμηση της διαφοροποίησης μεταξύ των χρόνων άφιξης των RTP πακέτων.

Η δομή και τα πεδία από τα οποία αποτελείται ένα RTCP Receiver Report παρουσιάζονται στο επόμενο σχήμα.



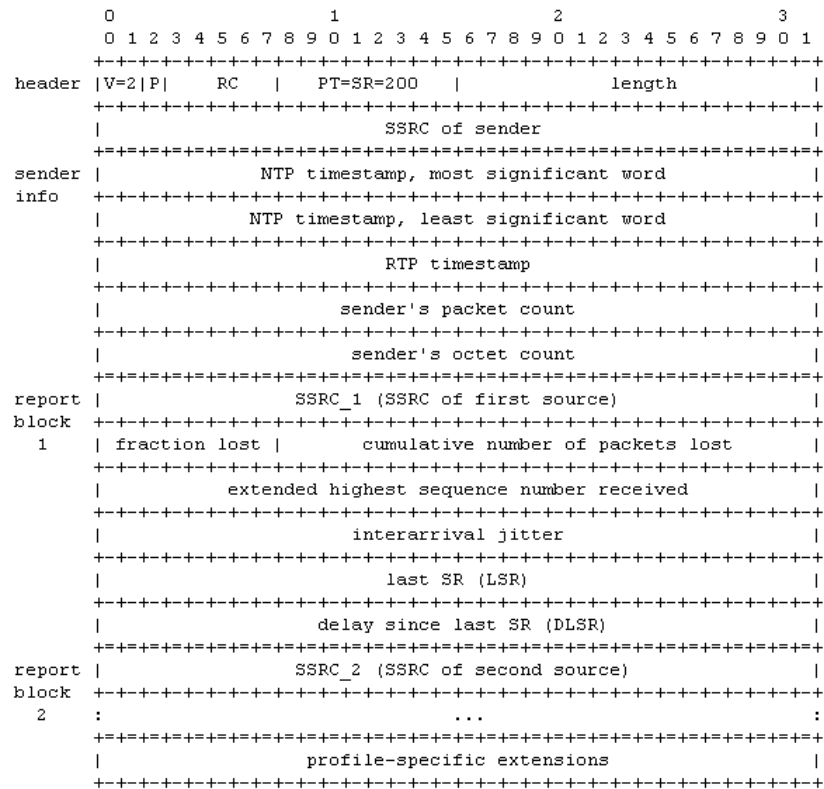
Σχήμα 5.7: Δομή RTCP Receiver Report

Αναφορές αποστολέα (Sender Report): Αποστέλλονται από τον αποστολέα για κάθε RTP ροή που μεταδίδει και μπορούν να χρησιμοποιηθούν για το συγχρονισμό διαφορετικών media ροών σε μια RTP σύνοδο. Τα πιο σημαντικά πεδία από τα οποία αποτελείται είναι:

- Το SSRC του RTP stream για το οποίο δημιουργήθηκε η συγκεκριμένη αναφορά.

- Ο αριθμός των πακέτων που στέλλονται στη ροή
- Ο αριθμός των bytes που στέλλονται στη ροή
- Το timestamp και το wall clock time του πιο πρόσφατα δημιουργημένου RTP πακέτου στη ροή.

Η δομή και τα πεδία από τα οποία αποτελείται ένα RTCP Sender Report παρουσιάζονται στο επόμενο σχήμα.



Σχήμα 5.8: RTCP Sender Report

Source Descriptions: Αποστέλλονται από τον αποστολέα για κάθε RTP ροή που μεταδίδει και δίνουν πληροφορίες για την πηγή όπως τη διεύθυνση ηλεκτρονικού ταχυδρομείου του αποστολέα και πληροφορίες για την εφαρμογή η οποία χρησιμοποιείται.

Πακέτο RTCP BYE: (Goodbye RTCP Packet)

Το συγκεκριμένο πακέτο υποδεικνύει ότι κάποιοι από τους αποστολείς, πηγές δεν είναι πλέον διαθέσιμοι.

Πακέτο Application Defined RTCP (APP)

Το είδος RTCP πακέτου αυτού είναι χρήσιμο για πειραματική χρήση κατά την ανάπτυξη νέων χαρακτηριστικών ή εφαρμογών .

5.4.2.4 RTP σύνοδος

Θα πρέπει να καθοριστούν τα χαρακτηριστικά της συνόδου τα οποία είναι η IP διεύθυνση δικτύου και τα 2 ports, το ένα για χρήση από το RTP και το άλλο για χρήση από το RTCP. Η διεύθυνση μπορεί να είναι μια unicast ή multicast διεύθυνση δικτύου. Το RTP port χρησιμοποιείται από όλους τους συμμετέχοντες στη σύνοδο για την αποστολή των πακέτων που θέλουν να μεταδώσουν δεδομένα ενώ το RTCP χρησιμοποιείται από όλους τους συμμετέχοντες στη σύνοδο για την αποστολή των RTCP μηνυμάτων ελέγχου. Μέσα σε μια multimedia σύνοδο, κάθε medium μεταδίδεται σε ξεχωριστή RTP σύνοδο. Τα RTCP πακέτα δίνουν πληροφορίες για κάθε σύνοδο ξεχωριστά.

5.5 Μετάδοση Ροών Βίντεο σε Best-Effort δίκτυα [1]

Το βίντεο είναι ένας τύπος πολυμεσικού περιεχομένου το οποίο χρησιμοποιείται σε μεγάλο βαθμό από πολλές εφαρμογές πλέον στο διαδίκτυο. Έτσι αναπτύχθηκε η μετάδοση βίντεο μέσω best effort δικτύων. Η υπηρεσία best effort προσπαθεί να προσφέρει την μετάδοση των δεδομένων από την πηγή προς τον προορισμό όσο πιο γρήγορα είναι δυνατόν χωρίς όμως να δίνει εγγυήσεις σε θέματα όπως καθυστερήσεις, απώλεια πακέτων και packet jitter. Αυτό δίνει έξτρα πολυπλοκότητα σε αυτή την διαδικασία μετάδοσης. Επιπρόσθετα υπάρχουν και θέματα fairness όσον αφορά την κατανομή των διαθέσιμων πόρων του δικτύου στις διάφορες ροές.

Το πρόβλημα με την απώλεια πακέτων θα μπορούσε να επιλυθεί με την μετάδοση των πακέτων με TCP αντί UDP. Το TCP όμως πραγματοποιεί αναμεταδόσεις των χαμένων πακέτων γεγονός που σε εφαρμογές πραγματικού χρόνου δεν θα είναι αποδοτικό και επιτρεπτό.

Το packet jitter είναι το φαινόμενο κατά το οποίο το διάστημα από την χρονική στιγμή κατά την οποία παράγεται ένα πακέτο μέχρι τη χρονική στιγμή που αυτό φτάνει στον προορισμό, στον παραλήπτη διαφέρει από ένα πακέτο σε άλλο. Αυτό οφείλεται στις διαφορετικές καθυστερήσεις που αντιμετωπίζει το κάθε πακέτο καθώς βρίσκεται στην ουρά για να αρχίσει η μετάδοσή του (queuing delay). Η παράμετρος αυτή μπορεί να επηρεάσει σημαντικά την ποιότητα του βίντεο κατά την παρουσίασή του στον παραλήπτη.

Σε περίπτωση που το περιβάλλον στο οποίο θα πραγματοποιηθεί η μετάδοση των δεδομένων είναι δυναμικό, δηλαδή ένα δίκτυο ασύρματο ή ένα δίκτυο με κινητούς χρήστες τότε αυτό αποτελεί μεγαλύτερη πρόκληση λόγω των απροσδιόριστων και μεταβαλλόμενων συνθηκών που επικρατούν στο δίκτυο. Επιπλέον σε ένα ασύρματο δίκτυο η τυχόν απώλεια πακέτων η οποία μπορεί να πραγματοποιηθεί μπορεί να οφείλεται σε κάποιο σφάλμα σε ένα συγκεκριμένο πακέτο και να μην αποτελεί έτσι καθαρή ένδειξη για ύπαρξη συμφόρησης στο δίκτυο. Αντίθετα, σε ένα ενσύρματο δίκτυο η απώλεια πακέτων εκλαμβάνεται ως ένδειξη συμφόρησης και οδηγεί στην χρήση κατάλληλων μηχανισμών για έλεγχο της.

5.6 Multicast Μετάδοση Πολυμεσικού Περιεχομένου

5.6.1 Εισαγωγή

Η υποστήριξη της multicast μετάδοσης δεδομένων στις εφαρμογές πολυμέσων είναι απαραίτητη λόγω του ότι πολλές εφαρμογές καθορίζουν μεγάλο αριθμό από παραλήπτες. Έτσι η χρήση multicast για τη μετάδοση των δεδομένων των εφαρμογών αυτών οδηγεί στην αποδοτικότερη χρήση των πόρων του δικτύου. Είναι πολύ σημαντική για εφαρμογές στο διαδίκτυο όπως οι βίντεο-τηλεδιασκέψεις και η κατανομή βίντεο. Όμως λόγω του ότι οι διάφοροι παραλήπτες του ίδιου πολυμεσικού περιεχομένου μπορεί να διαφέρουν μεταξύ τους όσον αφορά για παράδειγμα την ανοχή τους σε απώλεια και το διαθέσιμο εύρος ζώνης των συνδέσεων τους, η χρήση multicast μετάδοσης δεδομένων σε εφαρμογές πολυμέσων μπορεί να αποτελέσει μια πρόκληση.

Ο αποστολέας του πολυμεσικού περιεχομένου θα πρέπει να έχει τη δυνατότητα να προσαρμόζει το ρυθμό μετάδοσης δεδομένων με βάση τις συνθήκες που επικρατούν στο δίκτυο και στους παραλήπτες. Επίσης, θα πρέπει να λαμβάνεται υπόψη η

ετερογένεια που υπάρχει στο διαδίκτυο έτσι ώστε να μπορούν να ικανοποιούνται όσο πιο πολλοί παραλήπτες είναι δυνατόν.

5.6.2 Τρόποι Multicast Μετάδοσης Δεδομένων σε Εφαρμογές Πολυμέσων[18]

Υπάρχουν διάφοροι τρόποι για την πραγματοποίηση multicast μετάδοσης δεδομένων στις εφαρμογές πολυμέσων. Ο αποστολέας μπορεί να μεταδώσει μια multicast ροή δεδομένων για όλους τους παραλήπτες αφού πρώτα καθορίζει το ρυθμό μετάδοσης ο οποίος θα αφήσει ικανοποιημένο τον μεγαλύτερο αριθμό από παραλήπτες. Ένας άλλος τρόπος είναι να μεταδώσει πολλές multicast ροές δεδομένων οι οποίες θα έχουν διαφορετικό ρυθμό μετάδοσης και θα απευθύνονται σε διαφορετική ομάδα από παραλήπτες. Μια άλλη προσέγγιση είναι η χρήση κωδικοποίησης σε επίπεδα και η μετάδοση κάθε επιπέδου σε διαφορετική multicast ροή.

Με τον πρώτο τρόπο επιτυγχάνεται αποδοτική χρήση των διαθέσιμων πόρων που διαθέτει το δίκτυο και αποτελεί μια απλή λύση. Όμως μειονεκτεί στο γεγονός ότι προκαλεί προβλήματα όσον αφορά το fairness. Αυτό προκαλείται λόγω του ότι οι παραλήπτες μπορεί να έχουν διαφορετικές δυνατότητες. Η αποστολή μόνο μιας multicast ροής δεδομένων θα οδηγήσει τους παραλήπτες οι οποίοι έχουν χαμηλό εύρος ζώνης να λαμβάνουν μια υψηλού εύρους ζώνης ροής αν οι περισσότεροι παραλήπτες αυτής έχουν ψηλό εύρος ζώνης και αντίστροφα. Αυτό το πρόβλημα καλύπτεται με την αποστολή πολλών multicast ροών δεδομένων. Επιπρόσθετα, σε περίπτωση που θα πρέπει να πραγματοποιηθεί προσαρμοστική μετάδοση των δεδομένων ο αποστολέας θα πρέπει να επιλέξει το ρυθμό μετάδοσης που θα ικανοποιήσει τον μεγαλύτερο αριθμό από παραλήπτες.

Με τον δεύτερο αυτό τρόπο μετάδοσης κάθε ροή δεδομένων είναι κωδικοποιημένη σε διαφορετικό βαθμό ποιότητας. Κάθε ροή περιέχει το ίδιο βίντεο περιεχόμενο αλλά κάθε μια από αυτές έχει διαφορετικό ρυθμό κωδικοποίησης και ακόμα και διαφορετική μορφή βίντεο. Ο κάθε παραλήπτης θα λάβει μια ροή δεδομένων ανάλογα με τις δυνατότητες που έχει. Παρόλο που αυτός ο τρόπος για multicast μετάδοση δεδομένων λύνει το πρόβλημα με την αποστολή μιας μόνο multicast ροής, η αποστολή πολλών multicast ροών δεδομένων μειονεκτεί στο ότι η ίδια πληροφορία μεταδίδεται πολλές φορές στο δίκτυο.

Ένα βίντεο μπορεί να κωδικοποιηθεί σε επίπεδα, το βασικό επίπεδο και τα υπόλοιπα επιπρόσθετα επίπεδα. Το κάθε επίπεδο μεταδίδεται σε ξεχωριστή multicast ροή δεδομένων. Το βασικό επίπεδο δίνει τη βασική ποιότητα του βίντεο ενώ κάθε επιπρόσθετο επίπεδο προσφέρει μεγαλύτερη ακρίβεια και ποιότητα. Ο κάθε παραλήπτης μπορεί να λάβει μια ή περισσότερες multicast ροές δεδομένων ανάλογα και πάλι με τις δυνατότητες που έχει.

5.6.3 Κριτήρια για την αποδοτική multicast μετάδοση πολυμεσικού περιεχομένου

Η αποδοτική χρήση των διαθέσιμων πόρων του δικτύου είναι ο κύριος στόχος κατά την multicast μετάδοση πολυμεσικού περιεχομένου. Για να επιτευχθεί αυτό χρησιμοποιούνται κάποια κριτήρια ώστε να καθορίζεται αν πράγματι έχουμε τη βέλτιστη χρήση των πόρων του δικτύου. Τα κριτήρια αυτά είναι τα εξής:

Συμφόρηση στο δίκτυο

Πρέπει να γίνεται όσο πιο καλή χρήση του διαθέσιμου εύρους ζώνης στο δίκτυο ώστε να μην δημιουργείται συμφόρηση αλλά και να μην υποβαθμίζεται η λειτουργία του δικτύου. Επίσης δεν πρέπει να παρατηρείται μεγάλη απώλεια δεδομένων όσον αφορά τη συγκεκριμένη εφαρμογή πολυμέσων αλλά και τις εφαρμογές οι οποίες μεταδίδουν δεδομένα χρησιμοποιώντας τα ίδια δρομολόγια με αυτήν.

Επεκτασιμότητα

Η επίδοση του δικτύου δεν πρέπει να επηρεάζεται από τον αριθμό των παραληπτών των δεδομένων της εφαρμογής πολυμέσων. Αυτό σημαίνει πως όταν ο αριθμός αυτός αυξηθεί, το δίκτυο θα πρέπει πάλι να λειτουργά με αποδοτικό τρόπο.

Ικανοποίηση του χρήστη

Η απώλεια πακέτων επηρεάζει σε μεγάλο βαθμό την ικανοποίηση του τελικού χρήστη όσον αφορά τα δεδομένα που λαμβάνει και την ποιότητά τους.

Όσον αφορά την ποιότητα του βίντεο που λαμβάνεται υπάρχουν διάφορες μέθοδοι για την εκτίμηση της τόσο σε αντικειμενικό επίπεδο όσο και σε υποκειμενικό. Σχετικά με την αντικειμενική εκτίμηση της ποιότητας υπάρχουν μέθοδοι που είτε συγκρίνουν την αρχικά ακολουθία βίντεο μαζί με την αποκωδικοποιημένη είτε απλά βασίζονται στην

τελευταία μόνο. Παράδειγμα α αυτής της προσέγγισης είναι η τεχνική PSNR(Peak Signal to Noise Ratio) . Η αντικειμενική αντίληψη για την ποιότητα δε μπορεί να δώσει μια ολοκληρωμένη εικόνα για τον αν ο χρήστης μένει ικανοποιημένος από την ποιότητα που λαμβάνει ή όχι. Έτσι είναι απαραίτητα και η δική του αντίληψη για αυτό που λαμβάνει.

5.7 Προσαρμοστική Multicast Μετάδοση Πολυμεσικού Περιεχομένου [16], [17],[19]

Ένα ετερογενές δικτυακό περιβάλλον έχει απρόβλεπτη φύση λόγω του ότι οι πόροι που διαθέτει για να χρησιμοποιηθούν από τις διάφορες εφαρμογές μπορούν να αλλάζουν πολύ γρήγορα και είναι περιορισμένοι. Το εύρος ζώνης, η από άκρο σε άκρο καθυστέρηση και η απώλεια πακέτων οδηγούν τις συνθήκες που επικρατούν σε ένα δίκτυο να μεταβάλλονται συνεχώς. Αυτό όμως έχει αντίκτυπο στην μετάδοση δεδομένων.

Ένα δίκτυο αποτελείται από πολλούς χρήστες και έτσι οι υπηρεσίες οι οποίες προσφέρονται από αυτό πρέπει να μοιράζονται σε όσο το δυνατό πιο μεγάλο αριθμό χρηστών. Κάθε χρήστης όμως έχει διαφορετικούς περιορισμούς, ικανότητες όπως και απαιτήσεις, γεγονός που πρέπει να λαμβάνεται υπόψη. Έτσι υπάρχει ανάγκη για αποδοτική προσαρμοστική μετάδοση δεδομένων ώστε να επιτυγχάνεται σταθερή ποιότητα υπηρεσίας.

Σήμερα υπάρχουν πολλές συσκευές όπως ηλεκτρονικοί υπολογιστές και κινητά τηλέφωνα οι οποίες υποστηρίζουν πολυμεσικό περιεχόμενο όπως είναι το βίντεο και έτσι το video streaming χρησιμοποιείται όλο και πιο πολύ. Οι μηχανισμοί προσαρμοστικής μετάδοσης θα πρέπει να προσαρμόζουν την μετάδοση του πολυμεσικού περιεχομένου σύμφωνα κάθε φορά με τις αλλαγές που συμβαίνουν στο δίκτυο. Αυτό σημαίνει πως θα πρέπει να παρακολουθούνται οι συνθήκες που επικρατούν σε ένα δίκτυο ανά πάσα χρονική στιγμή.

Η διαδικασία της προσαρμοστικής μπορεί να βασίζεται σε αλλαγές οι οποίες σχετίζονται με το εύρος ζώνης, την καθυστέρηση και την απώλεια πακέτων. Όσον αφορά τις βίντεο εφαρμογές, το frame rate, picture resolution size και colour depth μπορούν να προσαρμόζονται ανάλογα με την κατάσταση του δικτύου. Μια άλλη προσέγγιση είναι η

προσαρμογή του εύρους ζώνης ή ακόμη και της καθυστέρησης, εάν αυτό είναι εφικτό χωρίς να ξεπερνιούνται τα απαιτούμενα όρια που ορίζει η εφαρμογή.

Για να μπορεί να υλοποιηθεί και να πραγματοποιείται αυτού του είδους μετάδοση πρέπει να υπάρχουν οι ανάλογοι μηχανισμοί οι οποίοι να την υποστηρίζουν. Πρώτον, θα πρέπει να υπάρχει μηχανισμός ο οποίος θα είναι υπεύθυνος για την μετάδοση του πολυμεσικού περιεχομένου. Επίσης, πρέπει με κάποιον τρόπο να παρακολουθείται η κατάσταση του δικτύου ώστε να αναγνωρίζονται αμέσως οι αλλαγές που γίνονται σε αυτό. Η απώλεια πακέτων για παράδειγμα μπορεί να χρησιμοποιηθεί ως ένδειξη για ύπαρξη συμφόρησης και έχει άμεση σχέση μαζί με την ποιότητα που θα λαμβάνει ο χρήστης κατά την παρουσίαση των δεδομένων. Μπορεί έτσι να χρησιμοποιηθεί ο ρυθμός απώλειας πακέτων ο οποίος ορίζεται ως το ποσοστό των πακέτων τα οποία χάθηκαν και δεν έφτασαν στον προορισμό τους. Στη συνέχεια πρέπει να υπάρχει ο μηχανισμός ο οποίος θα προσαρμόζει την μετάδοση σύμφωνα με τις συνθήκες που παρατηρούνται στο δίκτυο. Υπάρχουν διάφορες τεχνικές προσαρμογής όπως η κωδικοποίηση σε στρώματα, όπως αναφέρθηκε σε προηγούμενο σημείο, η διαμόρφωση του ρυθμού με τον οποίο στέλλονται τα δεδομένα ή συνδυασμός από τεχνικές. Διαμορφώνοντας το ρυθμό αποστολής δεδομένων μεταβάλλονται κάποιες παράμετροι όπως ο ρυθμός πλαισίων (frame rate) ώστε να ελέγχεται ο ρυθμός μετάδοσης των δεδομένων. Κατά την μετάδοση του πολυμεσικού περιεχομένου υπάρχει πιθανότητα να συμβούν λάθη και έτσι είναι απαραίτητο να υπάρχει κάποιος μηχανισμός ο οποίος να ελέγχει αυτά τα λάθη και να προσπαθεί για την μείωση των απωλειών των πακέτων. Αυτό όμως έχει επιπτώσεις αφού μειώνεται το throughput και παρατηρούνται αυξημένες καθυστερήσεις. Συγκεκριμένα υπάρχουν 2 τεχνικές για τον διαχείριση των λαθών κατά την μετάδοση των δεδομένων και είναι οι εξής:

ARQ (Automatic Repeat Request)

Ο αποστολέας μεταδίδει ξανά τα χαμένα πακέτα δεδομένων στον παραλήπτη αφού ο παραλήπτης τον ενημερώνει για αυτό. Η ενημέρωση προς τον αποστολέα πραγματοποιείται με αποστολή επιβεβαιώσεων (acknowledgements).

FEC (Forward Error Correction)

Βασίζεται στην επιπρόσθετη πληροφορία την οποία μεταδίδει ο αποστολέας όταν μεταδίδει τα δεδομένα που θέλει. Αυτή η πληροφορία χρησιμοποιείται από τον παραλήπτη για να διορθωθούν τα λάθη και η απώλεια δεδομένων με αποτέλεσμα την ανακατασκευή του χαμένου πακέτου.

Υπάρχουν δύο τρόποι με τους οποίους είναι εφικτή αυτή η λειτουργία. Κατά τον πρώτο τρόπο, πραγματοποιείται αποστολή ενός επιπρόσθετου μέρους δεδομένων μετά από έναν συγκεκριμένο αριθμό από πακέτα τα οποία έχουν σταλεί. Έτσι αν χαθεί κάποιο πακέτο το οποίο ακολουθία στη συνέχεια θα χρησιμοποιηθεί το επιπρόσθετο για την ανακατασκευή του. Κατά τον δεύτερο τρόπο η επιπρόσθετη πληροφορία η οποία αποστέλλεται είναι μια ροή των αντίστοιχων δεδομένων είτε βίντεο είτε ήχου σε χαμηλότερη ποιότητα και ανάλυση.

ΚΕΦΑΛΑΙΟ 6

Σύστημα ADIVIS

6.1. Εισαγωγή

6.2. Κωδικοποίηση σε στρώματα (Layered Encoding)

6.3. Μηχανισμός Ανατροφοδότησης (Feedback Mechanism)

6.4. Αλγόριθμος Συγκεχυμένης Απόφασης (Fuzzy Decision Algorithm)

6.1 Εισαγωγή [20]

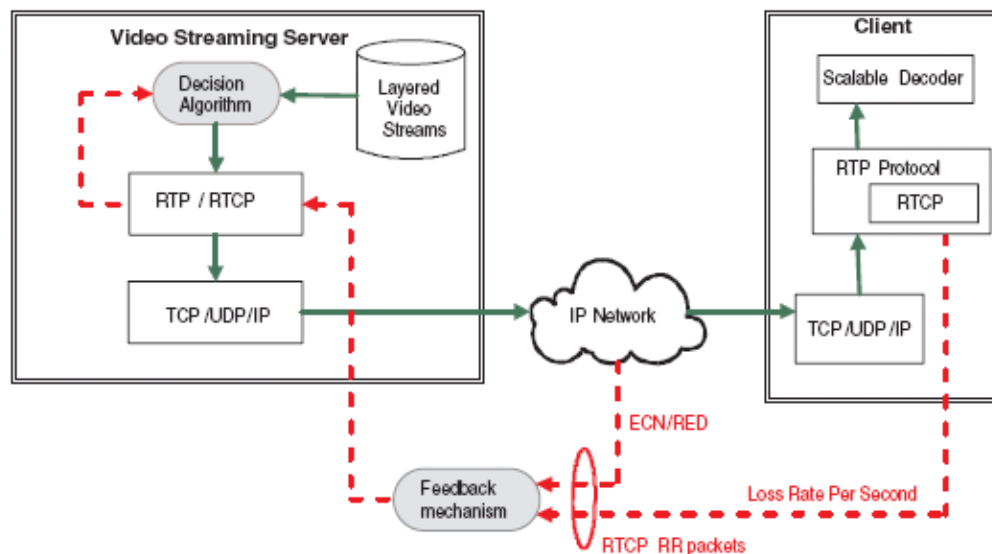
Ο αλγόριθμος Adivis είναι ένας τρόπος με τον οποίο είναι δυνατή η προσαρμοστική μετάδοση βίντεο σε ένα δίκτυο με βάση τις συνθήκες που επικρατούν σε αυτό και πιο συγκεκριμένα την απώλεια πακέτων που παρατηρείται καθώς ο φόρτος εργασίας του δικτύου μεταβάλλεται. Οι απαιτήσεις και οι περιορισμοί που χαρακτηρίζουν κάθε χρήστη ο οποίος επιθυμεί να παραλάβει μια συγκεκριμένη ροή βίντεο καθιστούν αναγκαία την δυνατότητα υποστήριξης τέτοιου είδους προσαρμογής ώστε να έχουμε καλή απόδοση.

Για να πραγματοποιηθεί ο πιο πάνω στόχος, ο Adivis συνδυάζει δύο είδη τεχνικών, τις τεχνικές προσαρμογής δικτύου (Network Adaptation Techniques(NATs)) και τις τεχνικές προσαρμογής περιεχομένου (Content adaptation Techniques(CATs)). Οι NATs, με βάση το δίκτυο και τυχόν σφάλματα που μπορεί να παρατηρηθούν, προσπαθούν να πετύχουν την προσαρμογή των αναγκών που έχουν οι εφαρμογές σύμφωνα με το τι συμβαίνει στο δίκτυο. Οι CATs χρησιμοποιούν προσεγγίσεις οι οποίες βασίζονται σε scalable βίντεο ώστε να πραγματοποιηθεί η προσαρμογή των δεδομένων στο ρυθμό μετάδοσης που πρέπει.

Η λειτουργία του αλγόριθμου ADIVIS βασίζεται σε ένα μηχανισμό ανατροφοδότησης και έναν αλγόριθμο συγκεχυμένης απόφασης. Κάθε βίντεο ροή κωδικοποιείται σε

πολλαπλά στρώματα τα οποία είναι αποθηκευμένα στον αποστολέα και κάθε ένα από αυτά μεταδίδεται μέσω μιας RTCP σύνδεσης. Ο μηχανισμός ανατροφοδότησης χρησιμοποιεί τις πληροφορίες τις οποίες λαμβάνει από τον παραλήπτη και αφορούν την αντιληπτή ποιότητα για αυτόν και τις μετρήσεις από το πυρήνα του δικτύου, για να κάνει την εκτίμηση για το διαθέσιμο εύρος ζώνης του μονοπατιού. Στη συνέχεια ο αλγόριθμος απόφασης λαμβάνει το υπολογισμένο εύρος ζώνης και αποφασίζει με συγκεκριμένο τρόπο τον βέλτιστο αριθμό από στρώματα τα οποία θα στείλει. Αυτό γίνεται με την πρόσθεση ή την αφαίρεση στρωμάτων. Έτσι γίνεται προσαρμογή του περιεχομένου του στρωματοποιημένου βίντεο στο διαθέσιμο εύρος ζώνης του δικτύου.

Το σχήμα που ακολουθεί παρουσιάζει τη λειτουργία ενός συστήματος το οποίο βασίζεται στον αλγόριθμο ADIVIS. Όπως αναφέρθηκε πιο πάνω, βλέπουμε πως ο μηχανισμός ανατροφοδότησης λαμβάνει τις κατάλληλες πληροφορίες από το δίκτυο και από την πλευρά του client και στη συνέχεια δίνει το αποτέλεσμα της επεξεργασίας τους στον αλγόριθμο απόφασης.



Σχήμα 6.1: Σύστημα βασισμένο στον αλγόριθμο ADIVIS

6.2 Κωδικοποίηση σε στρώματα (Layered Encoding) [20]

Η κωδικοποίηση σε στρώματα στοχεύει στην προσαρμογή της ποσότητας δεδομένων που μεταδίδονται από έναν video server στην χωρητικότητα ενός δεδομένου

μονοπατιού στο δίκτυο. Οι βίντεο ροές κωδικοποιούνται σε στρώματα έτσι ώστε κάθε επιπρόσθετο στρώμα να αυξάνει την αντιληπτή ποιότητα του παραλήπτη που λαμβάνει από την βίντεο ροή. Μια στρωματοποιημένη βίντεο ροή αποτελείται από ένα βασικό στρώμα (base layer) και αρκετά άλλα επιπρόσθετα ενισχυτικά στρώματα. Το βασικό στρώμα πρέπει να κωδικοποιείται σε χαμηλό rate. Αυτό πρέπει να γίνεται έτσι ώστε να εξυπηρετούνται περισσότεροι κινητοί χρήστες αλλά και για όσους είναι συνδεδεμένοι στο δίκτυο με συνδέσεις οι οποίες έχουν χαμηλό εύρος ζώνης. Για να πραγματοποιηθεί η προσαρμογή του rate του περιεχομένου στο επιθυμητό ρυθμό μετάδοσης μπορούν να προστεθούν ή να αφαιρεθούν και άλλα στρώματα.

6.3 Μηχανισμός Ανατροφοδότησης (Feedback Mechanism) [20]

Ο μηχανισμός αυτός λαμβάνει ως είσοδο πληροφορίες σχετικά με τη ποιότητα υπηρεσίας όπως για παράδειγμα ρυθμό απώλειας και jitter, οι οποίες προέρχονται από τον πυρήνα του δικτύου και τον παραλήπτη. Οι πληροφορίες αυτές χρησιμοποιούνται ώστε να εκτιμηθεί το διαθέσιμο εύρος ζώνης του μονοπατιού από τον αποστολέα προς τον παραλήπτη. Κάθε παραλήπτης στέλλει στατιστικές οι οποίες αφορούν την παραλαβή. Αυτό πραγματοποιείται με τη χρήση RTCP πακέτων, τα οποία περιέχουν μεταξύ άλλων στοιχεία το ποσοστό απώλειας πακέτων, τον αριθμό των πακέτων που χάθηκαν και το jitter. Οι στατιστικές παραλαβής μεταφέρονται μέσα σε Receiver Report (RR) πακέτα τα οποία στέλλονται από μη ενεργούς αποστολείς. Το ποσοστό απώλειας πακέτων σε ένα χρονικό διάστημα υπολογίζεται από τη διαίρεση του αριθμού των αναμενόμενων πακέτων δια τον αριθμό των χαμένων πακέτων μέσα στο χρονικό αυτό διάστημα. Ο ρυθμός απώλειας πακέτων ανά δευτερόλεπτο (loss rate per second – LRPS) υπολογίζεται από τη διαίρεση του ποσοστού απώλειας πακέτων δια τη διαφορά που υπάρχει ανάμεσα στα RRs timestamps:

$LRPS = \text{ποσοστό απώλειας πακέτων} / \text{διαφορά στα RRs timestamps}.$

Με βάση αυτή τη διαφορά μπορεί να πραγματοποιηθεί η εύρεση της αύξησης ή μείωσης του ποσοστού απώλειας πακέτων. Όπως αναφέρθηκε πιο πάνω, ο πυρήνας του δικτύου λαμβάνει μέρος σε αυτή την ανατροφοδότηση προς τον αποστολέα. Ο αποστολέας μπορεί να ενημερώνεται για την κατάσταση συμφόρησης που επικρατεί στο δίκτυο τη συγκεκριμένη στιγμή μέσω των δρομολογητών. Χρησιμοποιείται ο μηχανισμός Explicit Congestion Notification (ECN) για την ενημέρωση όσον αφορά τη

συμφόρηση που υπάρχει σε τελικούς κόμβους έτσι ώστε να μην παρατηρείται απώλεια πακέτων.

6.4 Αλγόριθμος Συγκεχυμένης Απόφασης (Fuzzy Decision Algorithm) [20], [21]

Ο αλγόριθμος αυτός λαμβάνει την απόφαση για τον βέλτιστο αριθμό των στρωμάτων που πρέπει να αποσταλούν με βάση την εκτίμηση που έγινε για το διαθέσιμο εύρος ζώνης του δικτύου από τον μηχανισμό ανατροφοδότησης. Η λειτουργία του βασίζεται σε ένα fuzzy σύστημα το οποίο χρησιμοποιεί 2 γλωσσικές μεταβλητές (linguistic terms) εισόδου και μια εξόδου.

Η σταθερά kT , όπου το T είναι η περίοδος απόφασης, χρησιμοποιείται σε όλες τις ποσότητες. Η πρώτη γλωσσική μεταβλητή εισόδου σχετίζεται με την παράμετρο LRPS. Το $LRPS(kT)$ είναι ο ρυθμός απώλειας ανά δευτερόλεπτο σε κάθε περίοδο απόφασης και το $LRPS(kT-T)$ είναι ο ρυθμός απώλειας ανά δευτερόλεπτο με καθυστέρηση T . Οι τιμές που μπορεί να λάβουν κυμαίνονται από το 0 μέχρι και το 1. Η γλωσσική μεταβλητή $D_{LRPS}(kT)$ δίνει την τάση αύξησης ή μείωσης του LRPS. Ο υπολογισμός της πραγματοποιείται από την εξίσωση:

$$D_{LRPS}(kT) = LRPS(kT) - LRPS(kT-T) \text{ η οποία κυμαίνεται από } -1 \text{ μέχρι και } +1.$$

Η δεύτερη γλωσσική μεταβλητή λαμβάνει υπόψη τον αριθμό των πακέτων τα οποία είναι μαρκαρισμένα με το ECN bit, $N_{ECN}(kT)$, μέσα σε μια περίοδο, ως ένδειξη για ύπαρξη συμφόρησης. Ο παραλήπτης υπολογίζει περιοδικά αυτόν τον αριθμό, ο οποίος στη συνέχεια μπαίνει σε ένα RR πακέτο. Ο αποστολέας αφού εξάγει αυτόν τον αριθμό από το πακέτο, υπολογίζει την scaled παράμετρο $N_{ECNsc}(kT)$ η οποία λαμβάνει τιμές από το -1 μέχρι και το $+1$. Αυτή η παράμετρος δίνει το ποσοστό των πακέτων που είναι μαρκαρισμένα σε αυτήν την περίοδο και λαμβάνεται από τον εξής υπολογισμό:

$$N_{ECNsc}(kT) = \frac{N_{ECN}(kT)}{N_{ps}(kT)}$$

Όπου το $N_{ps}(kT)$ είναι ο αριθμός των πακέτων τα οποία έχουν σταλεί μέσα στην ίδια περίοδο.

Στη συνέχεια υπολογίζεται η τάση αύξησης ή μείωσης του αριθμού των μαρκαρισμένων πακέτων, $DN_{ECNsc}(kT)$:

$$DN_{ECNsc}(kT) = N_{ECNsc}(kT) - N_{ECNsc}(kT - T), \text{ η οποία κυμαίνεται από } -1 \text{ μέχρι και } +1.$$

Η τιμή εξόδου επιλέγεται να λαμβάνει τιμές από 0.5 μέχρι 1.5. Όταν υπάρχει διαθέσιμο εύρος ζώνης και μειωμένη συμφόρηση επιτρέπεται μια σταδιακή αύξηση ενώ αντίθετα αν παρατηρηθεί συμφόρηση μπορούν να πραγματοποιηθούν γρήγορες κινήσεις ώστε να μειωθεί ο ρυθμός στο μισό.

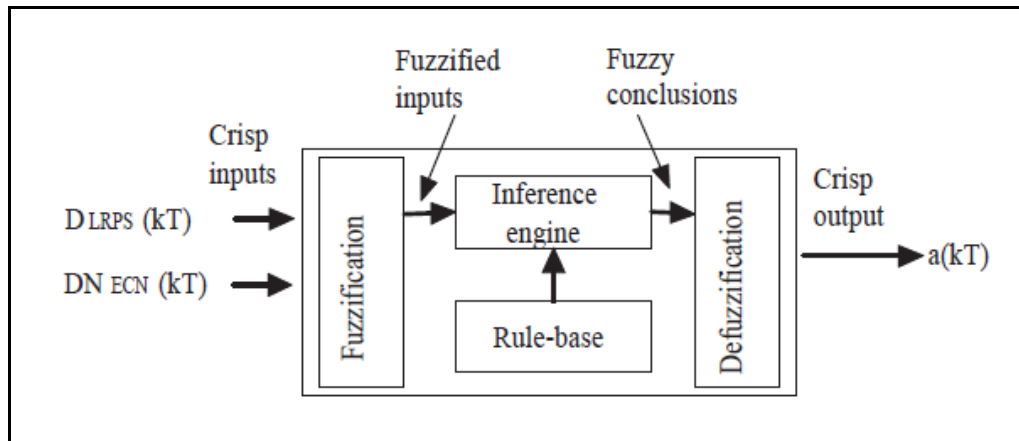
Η μεταβλητή εξόδου $a(kT)$ υπολογίζεται με βάση τις τιμές των 2 μεταβλητών $D_{LRPS}(kT)$ και $DN_{ECNsc}(kT)$ και χρησιμοποιώντας κάποιους κανόνες. Αυτοί οι κανόνες παρουσιάζονται στον πίνακα που ακολουθεί.

$a(kT)$		$DN_{ECNsc}(kT)$						
		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

Πίνακας 6.1: Κανόνες για την μεταβλητή εξόδου

Όπου NVB/PVB (Negative/Positive Very Big), NB/PB (Negative/Positive Big), NS/PS(Negative/Positive Small), Z(Zero), VS/VB(Very Small/Big), S(Small), B(Big) και H(High).

Η λειτουργία του αλγόριθμου συγκεχυμένης απόφασης ώστε να υπολογιστεί η μεταβλητή εξόδου παρουσιάζεται στο πιο κάτω σχήμα. Μπορούμε να δούμε τις 2 μεταβλητές που λαμβάνονται αρχικά ως είσοδος σε αυτό το σύστημα και στη συνέχεια με βάση την επεξεργασία που πραγματοποιείται σε αυτά σύμφωνα με τους πιο πάνω κανόνες, εξάγεται η μεταβλητή εξόδου.



Σχήμα 6.2: Αλγόριθμος Συγκεχυμένης Απόφασης

Ο αλγόριθμος απόφασης για να εκτιμήσει το διαθέσιμο εύρος ζώνης χρησιμοποιεί την μεταβλητή εξόδου $a(kT)$ με τον εξής τρόπο:

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

Ανάλογα με το διαθέσιμο εύρος ζώνης που υπάρχει στο δίκτυο αποφασίζονται ποια στρώματα πρέπει να σταλούν. Ο εξυπηρετητής έχει έναν αριθμό από στρώματα και το καθένα από αυτά είναι συσχετισμένο με ένα διαφορετικό ρυθμό μετάδοσης. Η μετάβαση σε ένα υψηλότερο στρώμα δεν πρέπει να πραγματοποιείται κάθε φορά που το διαθέσιμο εύρος ζώνης ξεπερνά μια συγκεκριμένη τιμή (threshold), η οποία τιμή σηματοδοτεί το ρυθμό μετάδοσης για ένα ψηλότερο στρώμα. Με αυτόν τον τρόπο αποφεύγονται οι συνεχείς μεταβάσεις σε άλλα στρώματα. Για αυτόν τον λόγο υπάρχει η χρονική υστέρηση η οποία είναι ίση με το χρονικό διάστημα μεταξύ της λήψης δύο διαδοχικών RR πακέτων. Έτσι αν το διαθέσιμο εύρος ζώνης ξεπεράσει το threshold του συγκεκριμένου ρυθμού μετάδοσης κάποιου στρώματος που βρίσκεται πιο ψηλά τότε θα ενεργοποιηθεί η χρονική υστέρηση. Αν το διαθέσιμο εύρος ζώνης βρίσκεται στα ίδια

επίπεδα όταν φτάσει το επόμενο RR πακέτο τότε μπορεί να πραγματοποιηθεί η μετάβαση σε πιο ψηλό στρώμα. Αντίθετα η μετάβαση σε ένα χαμηλότερο στρώμα πραγματοποιείται άμεσα λόγω του ότι γίνεται προσπάθεια για αποφυγή απώλειας πακέτων και συμφόρησης.

.....

ΚΕΦΑΛΑΙΟ 7

Darwin Streaming Server

7.1. Εισαγωγή

7.2. Λειτουργία του Darwin Streaming Εξυπηρετητή

7.2. Programming Interface of Darwin Streaming Server

7.3. QTSS Roles

7.4. QTSS Objects

7.1 Εισαγωγή [22]

Ο Darwin Streaming εξυπηρετητής (Darwin Streaming Server (DSS)) είναι η έκδοση ανοικτού κώδικα του QuickTime Streaming εξυπηρετητή, προϊόν από την εταιρεία Apple. Όπως κάθε άλλος streaming εξυπηρετητής έχει τη δυνατότητα να αποστέλλει streaming media μέσω του διαδικτύου. Η μετάδοση των δεδομένων πραγματοποιείται με βάση τα πρωτόκολλα RTP και RTSP. Όσον αφορά θέματα κώδικα βασίζεται σε μεγάλο βαθμό στον κώδικα που χρησιμοποιείται για τον QuickTime Streaming εξυπηρετητή. Αρχικά ήταν διαθέσιμος για χρήση μόνο σε Mac OS X αλλά στη συνέχεια δημιουργήθηκε η δυνατότητα για χρήση σε διάφορες πλατφόρμες όπως Linux, Solaris και Windows. Μπορεί να υποστηρίζει διάφορους τύπους από media όπως H.264/MPEG-4 AVC, 3GP και MPEG-4 Part 2 και όσον αφορά τη μετάδοση των δεδομένων μπορεί να χρησιμοποιηθεί και για multicast και για unicast μετάδοση.

Ένας streaming εξυπηρετητής μπορεί να δημιουργήσει ροές βίντεο από QuickTime ταινίες οι οποίες είναι αποθηκευμένες σε δίσκο ή μπορεί να μεταδίδει live stream όταν υπάρχει η ανάλογη πρόσβαση. Όπως έχουμε ήδη αναφέρει σε προηγούμενο κεφάλαιο, τα δεδομένα τα οποία θα πρέπει να αποσταλούν κωδικοποιούνται και στη συνέχεια συμπιέζονται. Τα τελικά αρχεία μετά από αυτήν τη διαδικασία αποθηκεύονται σε

συγκεκριμένη τοποθεσία. Έτσι ένας streaming εξυπηρετητής όπως είναι ο Darwin, θα μεταδίδει αυτά τα δεδομένα σε όσους τα έχουν ζητήσει.

Κάποιες δυνατότητες του Darwin Streaming εξυπηρετητή είναι το streaming MPEG-4 βίντεο, το mp3 audio streaming, η βελτιωμένη ποιότητα των ροών, οι βελτιώσεις σε θέματα επίδοσης, η παροχή ασφάλειας μέσω πιστοποίησης (authentication) και η υποστήριξη relay.

Από την πλευρά του client, πρέπει να υπάρχει το κατάλληλο λογισμικό ώστε να μπορεί να λάβει και να παίξει τα media. Μια τέτοια εφαρμογή στην περίπτωση μας είναι ο QuickTime Player ή κάποια άλλη εφαρμογή η οποία να μπορεί να υποστηρίξει τους τύπους δεδομένων που υποστηρίζει ο QuickTime Player. Πιο συγκεκριμένα ο QuickTime Player 6 ή μια πιο πρόσφατη έκδοση του θα προσφέρει καλύτερα αποτελέσματα από μια προηγούμενη έκδοση. Μια εναλλακτική λύση είναι η χρήση web browser για να γίνονται διαθέσιμα τα media προς τους χρήστες με την προϋπόθεση να είναι εγκατεστημένο το QuickTime plug-in.

Ο Darwin Streaming εξυπηρετητής και ο QuickTime Player είναι δύο από τα προϊόντα QuickTime τα οποία σχετίζονται με το streaming των media. Υπάρχουν επίσης τα ακόλουθα QuickTime Pro, QuickTime Broadcaster και QuickTime Streaming Server.

7.2 Λειτουργία του Darwin Streaming Εξυπηρετητή [22]

Η πρόσβαση στον εξυπηρετητή για παρακολούθηση της λειτουργίας του και για καθορισμό κάποιων ρυθμίσεων πραγματοποιείται εισάγοντας το ακόλουθο URL σε έναν web browser στον υπολογιστή στον οποίο είναι εγκατεστημένος ο εξυπηρετητής : `http://hostname:1220` όπου το hostname είναι το hostname ή η IP διεύθυνση του συγκεκριμένου υπολογιστή και 1220 είναι το port number.

Στη συνέχεια υπάρχουν διάφορες επιλογές όπως ρυθμίσεις για την τοποθεσία στην οποία θα αποθηκεύονται τα media αρχεία, για τον μέγιστο αριθμό από συνδέσεις που μπορεί να πραγματοποιεί ο εξυπηρετητής και για το streaming στο port με αριθμό 80. Το port 80 είναι το port εξ'ορισμού το οποίο χρησιμοποιείται για την web κίνηση δεδομένων. Ενεργοποιώντας την επιλογή αυτή, οι χρήστες οι οποίοι έχουν

συγκεκριμένα firewall για λόγους προστασίας και ασφάλειας, θα έχουν τη δυνατότητα να συνδεθούν με τον εξυπηρετητή για να ζητήσουν τα media δεδομένα που επιθυμούν.

Ένας client μπορεί να ζητήσει κάποιο βίντεο από τον Darwin Streaming εξυπηρετητή χρησιμοποιώντας τον QuickTime Player για παράδειγμα και ζητώντας το URL `rtsp://hostname/samplemovie.mov` όπου το hostname είναι το hostname ή η IP διεύθυνση του υπολογιστή στον οποίο είναι εγκατεστημένος ο Darwin Streaming εξυπηρετητής και `samplemovie.mov` είναι το όνομα του αρχείου το οποίο περιέχει το βίντεο που επιθυμούμε να παρακολουθήσουμε. Ο QuickTime Player συνδέεται με αυτόν τον τρόπο με τον streaming εξυπηρετητή και αν ο streaming εξυπηρετητής λειτουργεί χωρίς πρόβλημα τότε προβάλλεται το βίντεο.

Ο Darwin Streaming Server χρησιμοποιεί συγκεκριμένα ports ώστε να μπορεί να λαμβάνει και να επεξεργάζεται τις RTSP αιτήσεις από κάποιον client και να πραγματοποιεί streaming των επιθυμητών δεδομένων σε αυτόν. Ο επόμενος πίνακας συνοψίζει τα ports τα οποία χρησιμοποιεί ο εξυπηρετητής.

Ports used to communicate with client	554, 7070 TCP or 80 TCP
Ports used to send media	6970-6999 UDP, or 80 TCP
Ports used to receive broadcast	10000-65635 UDP
Ports server will stream through	554 RTSP 7070 TCP or 80 TCP
Default port typically used by MP3 broadcasters	8000 TCP
Port used for remotely managing QTSS or DSS	1220 TCP

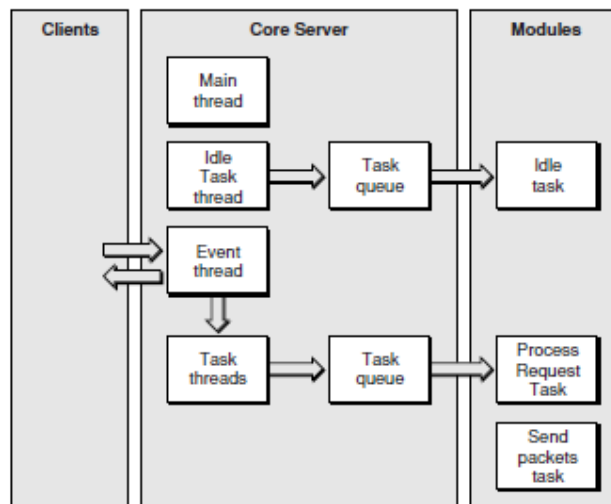
Σχήμα 7.1: Ports που χρησιμοποιεί ο DSS

7.3 Programming Interface of Darwin Streaming Server [23]

Το programming interface του Darwin Streaming εξυπηρετητή βασίζεται στην εκτέλεση διάφορων modules που υλοποιούνται σε αυτόν καθώς επίσης και στη δημιουργία νέων modules τα οποία επεκτείνουν τη βασική λειτουργία του εξυπηρετητή.

Ο εξυπηρετητής χρησιμοποιεί τα modules ώστε να μπορεί να ανταποκρίνεται στις διάφορες αιτήσεις που γίνονται σε αυτόν για τα media αρχεία, τα οποία είναι αποθηκευμένα σε αυτόν. Κάθε instance ενός module είναι υπεύθυνο για την φόρτωση(loading), την αρχικοποίηση και την εκτέλεση ενός στατικού ή δυναμικού API module.

Ο πυρήνας του εξυπηρετητή λειτουργεί ως μια ενδιάμεση οντότητα μεταξύ των clients και των modules τα οποία αποτελούν τον εξυπηρετητή. Οι clients στέλλουν τις αιτήσεις τους και λαμβάνουν σχετικές αποκρίσεις από τον εξυπηρετητή μετά την επεξεργασία που γίνονται σε αυτές από τα διάφορα modules. Όπως παρατηρείται και στο επόμενο σχήμα, υπάρχουν διάφορα είδη από threads τα οποία χρησιμοποιούνται ώστε να λειτουργεί ο εξυπηρετητής αποδοτικά. Καταρχήν υπάρχει το κυρίως thread (main thread) το οποίο πραγματοποιεί κάποιους ελέγχους σχετικά με τον εξυπηρετητή. Το Idle Task thread είναι υπεύθυνο για τη διαχείριση μιας ουράς από περιοδικά γεγονότα (events). Γεγονότα όπως είναι η λήψη μιας RTSP αίτησης ή ενός RTP πακέτου λαμβάνονται από το Event thread, το οποίο τα προωθεί στο Task thread ώστε και αυτό με τη σειρά του να τα προωθήσει στο κατάλληλο module για περαιτέρω επεξεργασία. Μπορεί να υπάρχουν περισσότερα από ένα Task thread.



Σχήμα 7.2: Αρχιτεκτονική του εξυπηρετητή

Υπάρχουν τρεις κατηγορίες από modules τα οποία χρησιμοποιεί ο Darwin Streaming εξυπηρετητής. Η πρώτη κατηγορία αφορά τα content-managing modules τα οποία είναι

υπεύθυνα για τον χειρισμό των RTSP αιτήσεων και ανταποκρίσεων για media δεδομένα. Τα server-support modules είναι τα modules τα οποία σχετίζονται με λειτουργίες log-in και data gathering. Τέλος, η τρίτη κατηγορία από modules είναι τα access control modules που εκτελούν λειτουργίες όπως είναι το authentication και το authorization . Τα modules τα οποία ορίζονται από τον DSS είναι τα ακόλουθα:

Content-Managing Modules	QTSSFileModule, QTSSReflectorModule, QTSSRelayModule, QTSSMP3StreamingModule
Server-Support Modules	QTSSErrorLogModule, QTSSAccessLogModule, QTSSWebStatsModule, QTSSWebDebugModule, QTSSAdminModule, QTSSPOSIXFileSystemModule.
Access Control Modules	QTSSAccessModule, QTSSHomeDirectoryModule, QTSSHttpFileModule, QTSSSpamDefenseModule.

Πίνακας 7.1: Modules

Ο κώδικας του εξυπηρετητή αυτού είναι υλοποιημένος σε γλώσσα προγραμματισμού C++ και οργανώνεται ως εξής:

Server.tproj	Βασικός source κώδικας.
CommonUtilitiesLib	Περιέχει διάφορες λειτουργίες χρήσιμες για thread management, data structure, networking και text parsing.
QTFileLib	Περιέχει τον source κώδικα για τη βιβλιοθήκη QTFile η οποία περιέχει τον κώδικα για την επεξεργασία των hinted QuickTime αρχείων.
APICommonCode	Περιέχει το source κώδικα για κλάσεις σχετικές με το API.
APIModules	Περιέχει ένα directory για κάθε module του streaming εξυπηρετητή.
RTSPClientLib	Περιέχει τον source κώδικα που υλοποιεί τον RTSP client του εξυπηρετητή.
RTCPUtilitiesLib	Περιέχει τον source κώδικα που χρησιμοποιείται για ην επεξεργασία των RTCP αιτήσεων προς τον εξυπηρετητή.
APIStubLib	Περιέχει αρχεία για τον καθορισμό του API και αρχεία

	στήριξης.
HTTPUtilitiesLib	Περιέχει τον source κώδικα που χρησιμοποιείται για ην επεξεργασία των HTTP αιτήσεων προς τον εξυπηρετητή.

Πίνακας 7.2: Οργάνωση Κώδικα Εξυπηρετητή

Το κάθε module για να έχει τη δυνατότητα να επεξεργάζεται διάφορα δεδομένα μέσω των συνδέσεων μεταξύ του εξυπηρετητή και των clients χρειάζεται τα QTSS roles. Μπορεί κάθε module να έχει διάφορα roles και με βάση το κάθε role, το module μπορεί να έχει πρόσβαση σε διάφορα δεδομένα μέσω των QTSS αντικειμένων(objects), τα οποία ορίζει ο Darwin Streaming εξυπηρετητής.

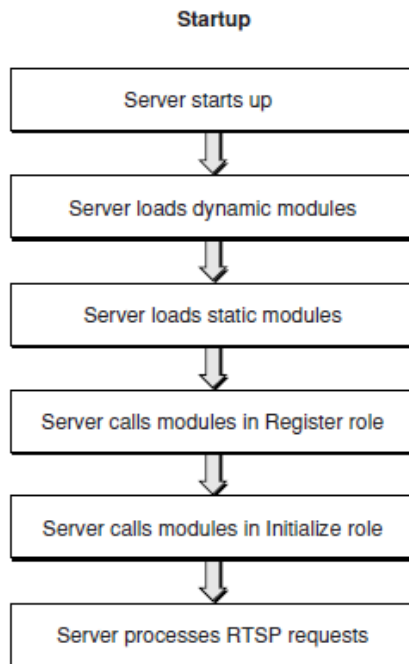
Κάποια από τα βασικά QTSS roles περιγράφονται συνοπτικά στον ακόλουθο πίνακα.

Όνομα Role	Constant	Περιγραφή
Register Role	QTSS_Register_Role	Εγγράφει τα roles για κάθε module.
Initialize Role	QTSS_Initialize_Role	Αρχικοποιεί το module.
Shutdown Role	QTSS_Shutdown_Role	Εκτελεί λειτουργίες clean up.
Error Log Role	QTSS_ErrorLog_Role	Υπεύθυνο για τα error logs.
RTSP Filter Role	QTSS_RTSPFilter_Role	Πραγματοποιεί αλλαγές τα περιεχόμενα των RTSP αιτήσεων.
RTSP Preprocessor Role	QTSS_RTSPPreProcessor_Role	Επεξεργάζεται τις αιτήσεις από κάποιον client πριν να τις επεξεργαστεί ο εξυπηρετητής.
RTSP Request Role	QTSS_RTSPRequest_Role	Επεξεργάζεται μια αίτηση από τον client εάν δεν ανταποκριθεί κάποιο άλλο role σε αυτήν.
RTP Send Packets Role	QTSS_RTPSendPackets_Role	Υπεύθυνο για την αποστολή των πακέτων.

Client Session Closing Role	QTSS_ClientSessionClosing_Role	Εκτελεί λειτουργίες όταν τερματίζεται. το client session
RTCP Process Role	QTSS_RTCPProcess_Role	Επεξεργάζεται RTCP Receiver Reports.
Open File Preprocess Role	QTSS_OpenFilePreProcess_Role	Επεξεργάζεται αιτήσεις για άνοιγμα αρχείων
Open File Role	QTSS_OpenFile_Role	Επεξεργάζεται αιτήσεις για άνοιγμα αρχείων σε περίπτωση που δεν γίνει αυτό από το προηγούμενο role
Advise File Role	QTSS_AdviseFile_Role	Ενεργοποιείται όταν πραγματοποιηθεί κλήση στη συνάρτηση QTSS_Advise
Read File Role	QTSS_ReadFile_Role	Υπεύθυνο για διάβασμα αρχείου
Request Event File Role	QTSS_RequestEventFile_Role	Διαχειρίζεται τις ειδοποιήσεις διαθεσιμότητας ενός αρχείου για διάβασμα
Close File Role	QTSS_CloseFile_Role	Υπεύθυνο για το κλείσιμο ενός αρχείου το οποίο ήταν ανοικτό για διάβασμα

Πίνακας 7.3: QTSS Roles

Στη συνέχεια παρουσιάζεται στο επόμενο σχήμα η σειρά με την οποία φορτώνονται τα modules κατά την έναρξη λειτουργίας του εξυπηρετητή, όπως επίσης και ποια roles ενεργοποιούνται αρχικά.



Σχήμα 7.3: Έναρξη λειτουργίας του εξυπηρετητή

Όπως βλέπουμε, πρώτα φορτώνονται τα δυναμικά modules, modules δηλαδή που δεν έχουν μεταγλωττιστεί στον εξυπηρετητή ενώ στη συνέχεια φορτώνονται τα στατικά modules, τα modules τα οποία έχουν μεταγλωττιστεί στον εξυπηρετητή. Ακολούθως, ενεργοποιείται το Register role για κάθε module, το οποίο θα περιγραφεί συνοπτικά στη συνέχεια και μετά το Initialize role για τυχόν αρχικοποιήσεις. Κατά τη διάρκεια λειτουργίας του εξυπηρετητή θα ενεργοποιούνται τα διάφορα άλλα roles κάθε module ανάλογα με τα διάφορα γεγονότα (events) που πραγματοποιούνται.

Όπως αναφέρθηκε πιο πριν, για να υπάρχει μια επικοινωνία μεταξύ των modules και του εξυπηρετητή υπάρχουν τα QTSS αντικείμενα και έτσι είναι δυνατή η ανταλλαγή δεδομένων μεταξύ τους. Το κάθε QTSS αντικείμενο έχει τις δικές του ιδιότητες και δεδομένα. Ο Darwin Streaming εξυπηρετητής ορίζει τους ακόλουθους τύπους αντικειμένων:

qtssAttrInfoObjectType	Περιέχει τα χαρακτηριστικά που περιγράφουν μια ιδιότητα ενός module.
qtssClientSessionObjectType	Περιέχει τα χαρακτηριστικά όσον αφορά μια client σύνοδο(session).

qtssConnectedUserObjectType	Περιέχει τα χαρακτηριστικά σχετικά με έναν συνδεδεμένο χρήστη.
qtssDynamicObjectType	Χρησιμοποιείται για τη δημιουργία ενός αντικειμένου το οποίο δεν έχει στατικά χαρακτηριστικά.
qtssFileObjectType	Περιέχει τα χαρακτηριστικά σχετικά με ένα ανοικτό αρχείο.
qtssModuleObjectType	Περιέχει τα χαρακτηριστικά τα οποία περιγράφουν ένα QTSS module.
qtssPrefsObjectType	Περιέχει τα χαρακτηριστικά τα οποία περιγράφουν το internal preference storage σύστημα του εξυπηρετητή.
qtssRTPStreamObjectType	Περιέχει τα χαρακτηριστικά τα οποία περιγράφουν μια συγκεκριμένη RTP ροή δεδομένων.
qtssRTSPHeaderObjectType	Περιέχει τα χαρακτηριστικά σχετικά με όλες τις επικεφαλίδες των RTSP αιτήσεων.
qtssRTSPRequestObjectType	Περιέχει τα χαρακτηριστικά που περιγράφουν μια συγκεκριμένη RTSP αίτηση.
qtssRTSPSessionObjectType	Περιέχει τα χαρακτηριστικά τα οποία είναι σχετικά με μια RTSP client- server σύνδεση.
qtssServerObjectType	Περιέχει τα χαρακτηριστικά σχετικά με πληροφορίες που αφορούν τον εξυπηρετητή.
qtssTextMessageObjectType	Περιέχει τα χαρακτηριστικά των οποίων οι τιμές προορίζονται για το χρήστη.

Πίνακας 7.4: Τύποι Αντικειμένων

7.4 QTSS Roles [23]

7.4.1 Register Role

Το κάθε module χρησιμοποιεί αυτό το role ώστε να καταγραφούν όλα τα roles για τα οποία θα καλείται όταν πραγματοποιείται η κατάλληλη ενέργεια. Το κάθε role που θα υποστηρίξει το module εγγράφεται με τη χρήση της συνάρτησης QTSS_AddRole. Το

register role είναι το πρώτο role το οποίο καλείται από τον Darwin Streaming εξυπηρετητή και καλείται μόνο μια φορά στην έναρξη της λειτουργίας του εξυπηρετητή.

Το role αυτό μπορεί να χρησιμοποιηθεί και για άλλους σκοπούς όπως για παράδειγμα για πρόσθεση στατικών χαρακτηριστικών στους τύπους QTSS αντικειμένων.

7.4.2 RTCP Process Role

Το role αυτό είναι χρήσιμο όταν θέλουμε να επεξεργαστούμε RTCP receiver reports, οι οποίες αποστέλλονται από κάθε χρήστη ο οποίος είναι συνδεδεμένος με τον εξυπηρετητή ώστε να λάβει το βίντεο, παρακολουθώντας έτσι την κατάσταση που επικρατεί στο δίκτυο και τη συγκεκριμένη σύνδεση προς τον εξυπηρετητή. Καλείται από τον εξυπηρετητή μόλις ληφθεί από αυτόν ένα RTCP receiver report.

Για την λειτουργία αυτού του role χρησιμοποιείται μια δομή δεδομένων η οποία επιστρέφεται σε αυτό κάθε φορά που καλείται. Η δομή αυτή ονομάζεται QTSS_RTCPProcess_Params και έχει ως εξής:

```
typedef struct{  
  
    QTSS_RTPStreamObject inRTPStream;  
  
    QTSS_ClientSessionObject inClientSession;  
  
    void* inRTCPPacketData;  
  
    UInt32 inRTCPPacketDataLen;  
  
} QTSS_RTCPProcess_Params;
```

Στη συνέχεια περιγράφονται συνοπτικά το κάθε πεδίο της πιο πάνω δομής. Το πεδίο **inRTPStream** είναι ένα αντικείμενο τύπου QTSS_RTPStreamObject το οποίο αφορά την RTP ροή στην οποία ανήκει το συγκεκριμένο RTP πακέτο. Το πεδίο **inClientSession** είναι ένα αντικείμενο τύπου QTSS_ClientSessionObject για το συγκεκριμένο client session. Το πεδίο inRTCPPacketData αντιπροσωπεύει έναν δείκτη σε έναν buffer ο οποίος περιέχει τα πακέτα τα οποία μπορούμε να επεξεργαστούμε και τέλος το πεδίο inRTCPPacketDataLen αντιπροσωπεύει το μέγεθος των έγκυρων

δεδομένων που βρίσκονται στο buffer αυτό. Ένα module το οποίο θα υποστηρίζει αυτό το role τότε πρέπει να εγγράψει το RTCP Process role χρησιμοποιώντας το Register role.

7.5 QTSS Objects [23]

7.5.1 qtssClientSessionObjectType

Αυτός ο τύπος αντικειμένων καθορίζει ένα αντικείμενο το οποίο αποτελείται από τα χαρακτηριστικά που αφορούν ένα client session και το QTSS_ClientSessionObject, το οποίο αποτελεί αντικείμενο της δομής δεδομένων σχετικά με το RTCP Process role, είναι ένα instance του τύπου αυτού. Η πρόσβαση στα χαρακτηριστικά του αντικειμένου πραγματοποιείται με μια από τις τρεις συναρτήσεις οι οποίες ορίζονται στο source κώδικα του Darwin και είναι οι εξής: QTSS_GetValue, QTSS_GetValueAsString και QTSS_GetValuePtr.

Ανάμεσα στα χαρακτηριστικά που αποτελούν αυτό τον τύπο αντικειμένου είναι τα ακόλουθα:

qtssCliSesPacketLossPercent: είναι το ποσοστό των πακέτων που χάθηκαν.

qtssCliSesPacketSent: είναι ο αριθμός των RTP πακέτων που έχουν σταλεί σε αυτό το session.

qtssCliSesBytesSent: είναι ο αριθμός των RTP bytes που έχουν σταλεί σε αυτό το session.

7.5.2 qtssRTPStreamObjectType

Αυτός ο τύπος αντικειμένων καθορίζει ένα αντικείμενο το οποίο αποτελείται από τα χαρακτηριστικά που σχετίζονται με μια RTP ροή και το QTSS_RTPStreamObject, το οποίο και αυτό ανήκει στη δομή δεδομένων σχετικά με το RTCP Process role, είναι ένα instance αυτού του τύπου.

Ένα αντικείμενο αυτού του τύπου μπορεί να σχετίζεται με μόνο ένα αντικείμενο τύπου Client Session ενώ αντίθετα ένα αντικείμενο τύπου Client Session μπορεί να σχετίζεται με ένα ή περισσότερα αντικείμενα RTP Stream. Η πρόσβαση στα χαρακτηριστικά του

αντικειμένου πραγματοποιείται με μια από τις τρεις συναρτήσεις οι οποίες ορίζονται στο source κώδικα του Darwin και είναι οι εξής: QTSS_GetValue, QTSS_GetValueAsString και QTSS_GetValuePtr.

Ανάμεσα στα χαρακτηριστικά που αποτελούν αυτό τον τύπο αντικειμένου είναι τα ακόλουθα:

qtssRTPStrFractionLostPackets: είναι το fraction των πακέτων που χάθηκαν στη συγκεκριμένη ροή μέχρι τη συγκεκριμένη στιγμή.

qtssRTPStrPercentPacketsLost: είναι το ποσοστό των πακέτων που έχουν χαθεί για τη συγκεκριμένη ροή.

qtssRTPStrTotalLostPackets: είναι ο συνολικός αριθμός από πακέτα τα οποία έχουν χαθεί για τη συγκεκριμένη ροή μέχρι τη συγκεκριμένη στιγμή.

qtssRTPStrTotPacketsLost: είναι τα συνολικά πακέτα τα οποία χάθηκαν.

7.5.3 qtssRTSPRequestObjectType

Αυτός ο τύπος αντικειμένων καθορίζει ένα αντικείμενο το οποίο αποτελείται από τα χαρακτηριστικά που έχουν σχέση με μια RTSP αίτηση και το qtssRTSPRequestObject είναι ένα instance αυτού του τύπου αντικειμένων. Το συγκεκριμένο αντικείμενο υπάρχει από τη στιγμή που λαμβάνεται μια RTSP αίτηση από τον εξυπηρετητή μέχρι της στιγμή της απόκρισης.

Ένα αντικείμενο τύπου RTSP αίτησης πρέπει να σχετίζεται με ένα μόνο αντικείμενο RTSP συνόδου.

Ανάμεσα στα χαρακτηριστικά που αποτελούν αυτό τον τύπο αντικειμένου είναι τα ακόλουθα:

qtssRTSPReqFullRequest: είναι η ολοκληρωμένη μορφή της RTSP αίτησης η οποία στάλθηκε από τον client προς τον εξυπηρετητή.

qtssRTSPReqMethod: είναι η RTSP μέθοδος η οποία χαρακτηρίζει τη συγκεκριμένη RTSP αίτηση.

qtssRTSPReqStartTime: είναι η χρονική στιγμή της ροής βίντεο από όπου θα αρχίσει η αναπαραγωγή της, όπως καθορίστηκε στο πεδίο Range της RTSP PLAY αίτησης.

qtssRTSPStopTime: είναι η χρονική στιγμή της ροής βίντεο στην οποία θα τερματιστεί η αναπαραγωγή της ροής βίντεο, όπως καθορίστηκε στο πεδίο Range της RTSP PLAY αίτησης.

ΚΕΦΑΛΑΙΟ 8

Προηγούμενη Σχετική Εργασία

8.1. Εισαγωγή

8.2. Προτεινόμενη λύση

8.3. Εξαγωγή Αποτελεσμάτων

8.1 Εισαγωγή [24]

Η προηγούμενη σχετική εργασία σε σχέση με τον στόχο αυτής της μελέτης και εργασίας αφορούσε την δημιουργία ενός συστήματος το οποίο θα μπορεί να προσαρμόζει το περιεχόμενο και στη συγκεκριμένη περίπτωση πολυμεσικό περιεχόμενο τύπου βίντεο, το οποίο ζητά ένας χρήστης από μια πηγή. Πιο συγκεκριμένα, το σύστημα σύμφωνα με μετρήσεις που λαμβάνει από τον πυρήνα του δικτύου καθώς επίσης και πληροφορίες από τον παραλήπτη του περιεχομένου που αφορούν την απώλεια δεδομένων, θα πρέπει να λαμβάνει απόφαση ως προς το τι πρέπει να συμβεί με την τρέχουσα μετάδοση του βίντεο όσον αφορά την ποιότητά του

Για τη δημιουργία και υλοποίηση του πιο πάνω συστήματος χρησιμοποιήθηκε καταρχήν βίντεο τύπου H.264. Η προσαρμοστική μετάδοση του βίντεο έγινε με την υλοποίηση του συστήματος ADIVIS σε συνδυασμό με τη χρήση του λογισμικού JSVM. Επιπρόσθετα πραγματοποιήθηκε η δημιουργία μιας διαπροσωπείας μεταξύ των δυο.

8.2 Προτεινόμενη λύση [24]

Ο αλγόριθμος ADIVIS χρησιμοποιήθηκε λόγω του ότι μπορεί να προσαρμόζει την μετάδοση του βίντεο με βάση τις πληροφορίες που λαμβάνει από τον πυρήνα του δικτύου αλλά και τις στατιστικές που λαμβάνει από τον παραλήπτη, όπως είδαμε σε

προηγούμενο κεφάλαιο. Για την υλοποίησή του χρησιμοποιήθηκε η γλώσσα προγραμματισμού C# και το περιβάλλον Microsoft Visual Studio 2008.

Το λογισμικό JSVM, το οποίο αφορά το project Scalable Video Coding από την Join Video Team, κρίθηκε κατάλληλο για χρήση λόγω των διάφορων δυνατοτήτων που μπορεί να παρέχει. Πιο συγκεκριμένα ανάμεσα στις λειτουργίες που υποστηρίζει είναι η δημιουργία στρωματοποιημένου βίντεο και η εξαγωγή της κατάλληλης υπό-ροής η οποία περιέχει τον κατάλληλο αριθμό στρωμάτων ανάλογα με τις επιλεγμένες παραμέτρους. Αυτές οι λειτουργίες υποστηρίζονται με τις κατάλληλες βιβλιοθήκες και εκτελέσιμα αρχεία τα οποία δημιουργούνται με την μεταγλώττιση του κώδικα του λογισμικού, ο οποίος είναι γραμμένος σε C++. Ένας ακόμα λόγος επιλογής του ήταν η ευκολία χρήσης του. Η χρήση του λογισμικού πραγματοποιείται μέσω της γραμμής εντολών.

Αρχικά, δίνεται ένα αρχείο βίντεο στον κωδικοποιητή του JSVM ώστε να παραχθεί ένα στρωματοποιημένο αρχείο βίντεο. Με βάση ένα αρχικό εύρος ζώνης εξάγεται η κατάλληλη υπό-ροή με τον κατάλληλο αριθμό στρωμάτων, η οποία μετατρέπεται σε AVC ροή. Αφού ξεκινήσει η αποστολή της, αρχίζει η λειτουργία του αλγόριθμου ADIVIS.

Το σύστημα το οποίο υλοποιήθηκε βασίζεται στην προσομοίωση της αποστολής των πληροφοριών που λαμβάνονται ως είσοδο από τον αλγόριθμο ADIVIS. Όπως περιγράψαμε σε προηγούμενο κεφάλαιο, ο αλγόριθμος για να λάβει την απόφαση που πρέπει, χρησιμοποιεί δύο μεταβλητές εισόδου οι οποίες σχετίζονται με τις πληροφορίες που λαμβάνονται από τον παραλήπτη (RTCP πακέτα) και αυτές που προέρχονται από τον πυρήνα του δικτύου (με χρήση του μηχανισμού ECN), $D_{LRPS}(kT)$ και $DN_{ECNsc}(kT)$ αντίστοιχα. Οι τιμές των δύο αυτών παραμέτρων προκύπτουν μετά τον υπολογισμό κάποιων πράξεων που ορίζει ο αλγόριθμος βασισμένος στις πιο πάνω πληροφορίες. Έτσι στην υλοποίηση του συστήματος, προσομοιώθηκε η παράμετρος LRSP(kT) και η παράμετρος NECN(kT), οι οποίες αντιστοιχούν στις πιο πάνω πληροφορίες. Ο λόγος που δεν πραγματοποιήθηκε η λήψη των πληροφοριών σε πραγματικές συνθήκες είναι επειδή υλοποιείται το σύστημα μόνο από την πλευρά του αποστολέα οπότε και δεν είναι δυνατή η επικοινωνία με τον παραλήπτη και λόγω του ότι έπρεπε το δίκτυο να υποστηρίζει την αποστολή ECN πακέτων.

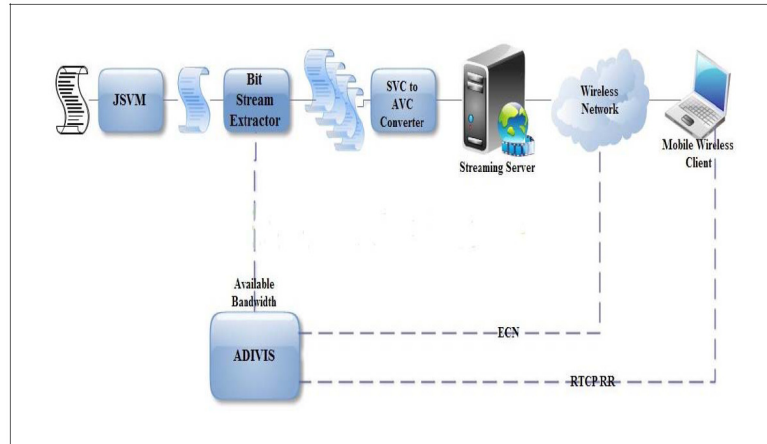
Στη συνέχεια υπολογίζεται το διαθέσιμο εύρος ζώνης και επίσης το αν θα πρέπει να εξαχθεί νέα υπό-ροή βίντεο ή όχι. Η λήψη αυτής της απόφασης γίνεται κάθε ένα συγκεκριμένο χρονικό διάστημα, την περίοδο απόφασης η οποία ορίστηκε $T=0.5$ δευτερόλεπτα. Έτσι εάν πρέπει να πραγματοποιηθεί εξαγωγή νέας υπό-ροής βίντεο, δίνεται ως είσοδο στο JSVM bit stream extractor το διαθέσιμο εύρος ζώνης. Με βάση αυτό θα εξαχθεί ένα βίντεο με λιγότερα ή περισσότερα στρώματα, ανάλογα με το αν είχαμε μείωση στο εύρος ζώνης ή εάν αυτό διατηρήθηκε στα ίδια επίπεδα μετά την εισδοχή σε φάση υστέρησης. Η αποστολή της νέας υπό-ροής βίντεο θα πρέπει ξεκινά από το σημείο στο οποίο είχε μείνει η προηγούμενη ροή βίντεο και έτσι πραγματοποιούνται κάποιοι υπολογισμοί ώστε να βρεθεί το σημείο αυτό, χρησιμοποιώντας την ποσότητα δεδομένων που στέλλονται κάθε φορά που ξεκινά μια νέα υπό-ροή βίντεο μέχρι την χρονική στιγμή που θα πραγματοποιηθεί μια μετάβαση σε άλλα υπό-ροή.

Αφού εξαχθεί η νέα υπό-ροή βίντεο σε κάθε περίοδο απόφασης, το εξαγόμενο SVC bit stream μετατρέπεται σε AVC bit stream μέσω του κατάλληλου εργαλείου του JSVM, ώστε να είναι δυνατή η αποστολή της ροής και από εξυπηρετητές οι οποίοι δεν υποστηρίζουν scalable βίντεο.

Η πιο πάνω λειτουργία του αλγόριθμου ADIVIS πραγματοποιείται μέχρι να ολοκληρωθεί το streaming ολόκληρου του βίντεο.

Η αποστολή της AVC ροής δεν πραγματοποιείται αφού όπως αναφέρθηκε, δεν υπάρχει η επικοινωνία με παραλήπτη. Άρα απλά πραγματοποιείται μια προσομοίωση της αποστολής των πακέτων.

Στο επόμενο σχήμα παρουσιάζεται η λειτουργία του συστήματος αυτού.



Σχήμα 8.1: Λειτουργία συστήματος προηγούμενης εργασίας

8.3 Εξαγωγή αποτελεσμάτων [24]

Με το τέλος κάθε εκτέλεσης της λειτουργίας του συστήματος, όπως περιγράψαμε πιο πάνω, λαμβάνεται ένα αρχείο (log file) το οποίο περιέχει πληροφορίες για κάθε περίοδο απόφασης, σχετικά με τις τιμές των δύο μεταβλητών εισόδου του αλγόριθμου ADIVIS, την μεταβλητή εξόδου, το διαθέσιμο εύρος ζώνης και το αν πρέπει να πραγματοποιηθεί ή όχι εξαγωγή νέας υποροής βίντεο.

Με βάση τα παραδείγματα εκτέλεσης που πραγματοποιήθηκαν, παρατηρήθηκε πως το πρόγραμμα συμπεριφέρεται όπως ήταν αναμενόμενο και πως πραγματοποιούνται σωστά οι μεταβάσεις που τυχόν να έπρεπε να γίνουν, σύμφωνα με το διαθέσιμο εύρος ζώνης.

ΚΕΦΑΛΑΙΟ 9

Προσαρμοστική Μετάδοση H.264 Βίντεο

9.1. Προτεινόμενη Λύση

9.2 Προτεινόμενος Αλγόριθμος

9.1 Προτεινόμενη Λύση

Έχοντας μελετήσει τις διάφορες έννοιες οι οποίες εμπλέκονται στην πραγματοποίηση αυτής της διπλωματικής εργασίας, κατέληξα στην πρόταση μιας λύσης για την πραγματοποίηση της προσαρμοστικής μετάδοσης H.264 βίντεο σε ασύρματα δίκτυα. Πιο συγκεκριμένα, έγινε έρευνα σε θέματα όπως ο τρόπος λειτουργίας των MIPv6 ασύρματων δικτύων, η διαδικασία κωδικοποίησης βίντεο όπως επίσης και τα υπάρχον πρότυπα κωδικοποίησης βίντεο και οι εφαρμογές πολυμεσικού περιεχομένου τύπου βίντεο και η μετάδοσή του. Λαμβάνοντας υπόψη όλα αυτά, έγινε κατανοητή η ανάγκη που υπάρχει για προσαρμοστική μετάδοση βίντεο. Έτσι για να πραγματοποιηθεί αυτός ο στόχος έγινε επιλογή του Darwin Streaming Server ως τον εξυπηρετητή ο οποίος θα διαχειρίζεται τις αιτήσεις ενός χρήστη για το βίντεο streaming και θα πραγματοποιεί το streaming και ο οποίος θα λειτουργεί σε συνεργασία με το σύστημα το οποίο θα πραγματοποιεί την απόφαση που αφορά την προσαρμογή της μετάδοσης βίντεο. Το σύστημα αυτό θα είναι μια υλοποίηση του συστήματος ADIVIS, για το οποίο έχει ήδη πραγματοποιηθεί μια υλοποίηση σε προηγούμενη εργασία.

Ο Darwin Streaming Server, όπως περιγράφηκε στο προηγούμενο κεφάλαιο είναι ένας εξυπηρετητής ο οποίος έχει τη δυνατότητα αποστολής streaming multimedia δεδομένων στο διαδίκτυο και πιο συγκεκριμένα υποστηρίζει το πρότυπο κωδικοποίησης βίντεο H.264, το οποίο είναι αυτό που έχει στόχο αυτή η εργασία. Αυτός ο εξυπηρετητής δέχεται RTSP αιτήσεις (RTSP requests) από clients και στη συνέχεια αν όλα πάνε σωστά παρουσιάζει την ροή βίντεο που ζήτησε ο client. Ο client πρέπει να έχει τον κατάλληλο player για να μπορεί να παρακολουθήσει το βίντεο που ζήτησε. Ο κώδικας

υλοποίησης για την λειτουργία του εξυπηρετητή αυτού είναι ανοικτός κώδικας, γεγονός που τον καθιστά διαθέσιμο σε κάποιον που επιθυμεί να ασχοληθεί με μετάδοση πολυμεσικού περιεχομένου.

Ο εξυπηρετητής αυτός βασίζεται στα πρωτόκολλα RTP και RTSP, γεγονός που ήταν σημαντικό για την επιλογή ώστε να χρησιμοποιηθεί στα πλαίσια της εργασίας αυτής. Ο πρώτος λόγος που συμβαίνει αυτό είναι γιατί και το σύστημα το οποίο βασίζεται στον αλγόριθμο ADIVIS υποθέτει την ύπαρξη RTCP σύνδεσης μεταξύ εξυπηρετητή και χρήστη. Ο δεύτερος λόγος που συμβαίνει αυτό είναι το ότι αφού χρησιμοποιεί το πρωτόκολλο RTP και επομένως το πρωτόκολλο RTCP, θα υπάρχει δυνατότητα λήψης των RTCP πακέτων και συγκεκριμένα των RTCP αναφορών παραλήπτη. Η λήψη αυτών των αναφορών θα παρέχει την πρόσβαση σε πληροφορίες οι οποίες είναι πολύ χρήσιμες για να μπορεί να πραγματοποιηθεί η λειτουργία του συστήματος ADIVIS και η προσαρμοστική μετάδοση του βίντεο. Ο ρυθμός απώλειας δεδομένων είναι αυτή η πληροφορία η οποία θα ενημερώνει για την απώλεια δεδομένων που παρατηρείται στον παραλήπτη και η οποία θα μπορεί να λαμβάνεται από το σύστημα ADIVIS ώστε να πραγματοποιούνται οι κατάλληλες ενέργειες από αυτό.

Για αυτήν τη διπλωματική εργασία εγκαταστάθηκε η έκδοση του Darwin Streaming εξυπηρετητή για Windows και συγκεκριμένα χρησιμοποιήθηκε σε Microsoft Windows XP. Η συγκεκριμένη έκδοση του εξυπηρετητή που χρησιμοποιήθηκε είναι ο Darwin Streaming Server 5.5.5. Επίσης, η υλοποίηση του σχετικού κώδικα στον εξυπηρετητή έγινε χρησιμοποιώντας τη γλώσσα προγραμματισμού C++ σε περιβάλλον Microsoft Visual Studio 6.0.

Ο αλγόριθμος ADIVIS είναι απαραίτητος για την επίτευξη του στόχου μας διότι με βάση τον μηχανισμό ανατροφοδότησης και τον αλγόριθμο συγκεκριμένης απόφασης στα οποία βασίζεται η λειτουργία του, μπορεί να αποφασίζει εάν πρέπει να πραγματοποιηθεί αλλαγή στην υπό-ροή βίντεο η οποία αποστέλλεται από τον εξυπηρετητή ή όχι. Άρα αυτό το σύστημα το οποίο θα βασίζεται σε αυτόν θα διαχειρίζεται την προσαρμογή της μετάδοσης του βίντεο. Η αποστολή των υπό-ροών βίντεο, σύμφωνα με τον ADIVIS πραγματοποιείται μέσω RTCP σύνδεσης.

Ο παραλήπτης στέλλει RTCP αναφορές παραλήπτη για τα δεδομένα που λαμβάνει από κάποιον εξυπηρετητή, οι οποίες περιέχουν πληροφορίες όπως είναι το ποσοστό

απώλειας πακέτων. Επιπρόσθετα, οι δρομολογητές στον πυρήνα του δικτύου ενημερώνουν τον αποστολέα για τις συνθήκες που επικρατούν στο δίκτυο μέσω του μηχανισμού ECN ο οποίος δίνει ενδείξεις για ύπαρξη συμφόρησης. Ο αλγόριθμος συγκεχυμένης απόφασης λαμβάνει ως είσοδο το ποσοστό απώλειας πακέτων όπως επίσης και τον αριθμό των πακέτων τα οποία έχουν ενεργοποιημένο το ECN bit σε κάθε περίοδο απόφασης. Στη συνέχεια επεξεργάζεται αυτές τις δύο πληροφορίες πραγματοποιώντας κάποιους υπολογισμούς. Μετά από αυτήν την επεξεργασία υπολογίζεται μια μεταβλητή εξόδου με βάση την οποία θα υπολογιστεί στη συνέχεια το διαθέσιμο εύρος ζώνης. Το διαθέσιμο εύρος ζώνης θα χρησιμοποιηθεί για να ληφθεί η απόφαση του κατά πόσο πρέπει να πραγματοποιηθεί μετάβαση σε μια υπό-ροή ψηλότερης ακρίβειας ή μια υπό-ροή χαμηλότερης ακρίβειας ή κατά πόσο θα συνεχιστεί η μετάδοση της αρχικής υπό-ροής. Η διαδικασία αυτή, της ανατροφοδότησης από τον παραλήπτη και τον πυρήνα του δικτύου και της απόφασης για την μετάδοση πραγματοποιείται μέχρι το τέλος του streaming του βίντεο.

Στα πλαίσια αυτής της διπλωματικής εργασίας, πραγματοποιήθηκε μελέτη μιας προηγούμενης σχετικής εργασίας της οποίας ο στόχος ήταν η δημιουργία συστήματος προσαρμοστικού βίντεο H.264 για χρήση σε ασύρματα κινητά δίκτυα. Έτσι έγινε μια υλοποίηση του συστήματος ADIVIS, όπως περιγράφηκε στο Κεφάλαιο 8. Η υλοποίηση αυτή αποτέλεσε την βάση για την χρήση του αλγόριθμου ADIVIS σε αυτήν τη διπλωματική εργασία. Η συγκεκριμένη υλοποίηση βασίστηκε στη χρήση της γλώσσας προγραμματισμού C# σε περιβάλλον Microsoft Visual Studio 2008.

9.2 Προτεινόμενος Αλγόριθμος

Το σύστημα ADIVIS και ο Darwin Streaming Server θα είναι εγκατεστημένοι στην πλευρά του εξυπηρετητή ενώ στην πλευρά του client θα είναι εγκατεστημένη κατάλληλη εφαρμογή για την παρουσίαση του βίντεο. Ξεκινά την λειτουργία του ο Darwin Streaming Server και καθώς βρίσκεται σε λειτουργία η client εφαρμογή ζητά μέσω του πρωτοκόλλου RTSP το βίντεο που επιθυμεί ο client. Μόλις ο εξυπηρετητής αρχίσει το streaming της ροής βίντεο τότε ξεκινά τη λειτουργία του ο αλγόριθμος ADIVIS. Η έναρξη λειτουργίας του αλγόριθμου ADIVIS ξεκινά μόλις ληφθεί το κατάλληλο σήμα από τον Darwin εξυπηρετητή. Αυτό πραγματοποιείται με τη βοήθεια ενός αρχείου το οποίο δημιουργείται όταν ξεκινήσει η αποστολή πακέτων από τον

εξυπηρετητή. Η δημιουργία του αρχείου είναι στην ουσία το σήμα προς το σύστημα του ADIVIS.

Ο αλγόριθμος ADIVIS θα λειτουργεί για όση ώρα διαρκεί η αναπαραγωγή της συγκεκριμένης ροής βίντεο. Η συγκεκριμένη πληροφορία, δηλαδή η διάρκεια της ροής λαμβάνεται από τον ADIVIS μέσω ενός αρχείου που δημιουργεί ο Darwin, όταν λαμβάνει την RTSP αίτηση από τον client για αναπαραγωγή της συγκεκριμένης ροής (όνομα αρχείου=MovieDuration.txt).

Η υπάρχουσα υλοποίηση του ADIVIS πραγματοποιεί μια προσομοίωση της λήψης της πληροφορίας που αφορά τον ρυθμό απώλειας δεδομένων αφού δεν υπήρχε επικοινωνία με τον client. Όμως, χρησιμοποιώντας τώρα τον Darwin Streaming Server έχουμε επικοινωνία με τον client χωρίς πρόβλημα, έτσι σε αυτό το σημείο θα πραγματοποιηθεί η αλλαγή στην υπάρχουσα υλοποίηση του ADIVIS ώστε ο ADIVIS να λαμβάνει τη συγκεκριμένη πληροφορία σε πραγματικό χρόνο. Ο Darwin πρέπει να έχει κάποιο μηχανισμό ώστε να μπορεί να λαμβάνει και να διαχειρίζεται τις RTCP αναφορές παραλήπτη τις οποίες αποστέλλει ο παραλήπτης, ώστε στη συνέχεια με κάποιο τρόπο να τις δίνει στο σύστημα ADIVIS. Αυτό έγινε εφικτό με την δημιουργία ενός νέου module, το οποίο προστέθηκε στα ήδη υπάρχον modules του εξυπηρετητή. Σε αυτό το module έγινε χρήση ενός QTSS Role από τα QTSS Roles, τα οποία ορίζει ο Darwin Streaming Server (Πίνακας 7.3) και συγκεκριμένα του RTCP Process Role (QTSS_RTCPProcess_Role).

Το role αυτό είναι χρήσιμο όταν θέλουμε να επεξεργαστούμε RTCP receiver reports, οι οποίες αποστέλλονται από κάθε χρήστη ο οποίος είναι συνδεδεμένος με τον εξυπηρετητή ώστε να λάβει το βίντεο, παρακολουθώντας έτσι την κατάσταση που επικρατεί στο δίκτυο και τη συγκεκριμένη σύνδεση προς τον εξυπηρετητή. Καλείται από τον εξυπηρετητή μόλις ληφθεί από αυτόν ένα RTCP receiver report. Με βάση τη δομή δεδομένων που επιστρέφεται όταν καλείται αυτό το role, παίρνουμε τον ρυθμό απώλειας δεδομένων τον οποίο χρειαζόμαστε. Συγκεκριμένα η δομή η οποία επιστρέφεται περιλαμβάνει τα εξής πεδία:

QTSS_RTPStreamObject in RTPStream;

QTSS_ClientSessionObject in ClientSession;

void* inRTCPPacketData;

UInt32 inRTCPPacketDataLen

Το αντικείμενο τύπου QTSS_ClientSessionObject συμπεριλαμβάνει στα πεδία του το πεδίο **qtssCliSesPacketLossPercent** το οποίο είναι το ποσοστό των πακέτων που χάνονται.

Το module αυτό καταγράφει αυτήν την πληροφορία σε ένα αρχείο .txt από το οποίο το σύστημα ADIVIS θα την λάβει. Στον κώδικα του ADIVIS πραγματοποιήθηκαν οι κατάλληλες αλλαγές ώστε να λαμβάνεται η πληροφορία από το αρχείο που δημιουργεί ο Darwin κατά την περίοδο απόφασης.

Όσον αφορά την αποστολή των μαρκαρισμένων πακέτων με το ECN bit ενεργοποιημένο, με βάση την υπάρχουσα υλοποίηση του ADIVIS, πραγματοποιείται και αυτή με προσομοίωση. Συγκεκριμένα ο αριθμός των πακέτων που είχαν ενεργοποιημένο το ECN bit υπολογιζόταν με τυχαίο τρόπο. Λόγω όμως του ότι το δίκτυο μας δεν υποστηρίζει την αποστολή πακέτων με βάση αυτόν τον μηχανισμό για ένδειξη συμφόρησης δεν είναι δυνατή η λήψη πραγματικών πληροφοριών και για αυτόν τον λόγο θα παραμείνει η συγκεκριμένη προσομοίωση. Αυτό όμως που έχει αλλαχθεί είναι το γεγονός ότι και αυτή η πληροφορία ια καταγράφεται στο ίδιο αρχείο με τον ρυθμό απώλειας δεδομένων, ώστε ο ADIVIS να λαμβάνει και τις δύο πληροφορίες από αυτό.

Οι πληροφορίες αυτές, δηλαδή ο ρυθμός απώλειας των δεδομένων όπως και ο αριθμός των πακέτων που έχουν ενεργοποιημένο το ECN bit, πρέπει να λαμβάνονται κάθε περίοδο απόφασης (=T). Η περίοδος απόφασης είναι το χρονικό διάστημα μεταξύ δύο διαδοχικών RTCP αναφορών παραλήπτη. Παρακολουθώντας σε λειτουργία τον Darwin Streaming εξυπηρετητή και καταγράφοντας σε ένα αρχείο log την πληροφορία της απώλειας δεδομένων κάθε φορά που λαμβάνεται μια RTCP αναφορά παραλήπτη, παρατήρησα ότι ο αριθμός των RTCP αναφορών ήταν πολύ μεγάλος αφού η συχνότητα λήψης τους ήταν μεγάλη. Αυτό έκανε αδύνατο να λαμβάνονται υπόψη από το προσαρμοστικό σύστημα ADIVIS κατά την φάση της απόφασης, όλες οι RTCP αναφορές παραλήπτη. Αυτό γιατί αν λαμβάνονταν υπόψη όλες οι RTCP αναφορές παραλήπτη τότε η περίοδος απόφασης θα ήταν πολύ μικρή και έτσι θα

πραγματοποιούνταν αποφάσεις για μετάβαση σε διαφορετική ροή βίντεο πολύ συχνά, γεγονός που δεν θα ήταν αποδοτικό. Επίσης, οι συχνές μεταβάσεις θα ήταν ακριβάστες αφού δε θα βελτίωναν την ποιότητα υπηρεσίας και επιπλέον θα προκαλούσαν και μεγαλύτερες καθυστερήσεις. Έτσι, η περίοδος απόφασης πρέπει να είναι τέτοια ώστε να δίνει τη δυνατότητα ενημέρωσης για την κατάσταση που επικρατεί στο δίκτυο και στον παραλήπτη σε σύντομα χρονικά διαστήματα ώστε να λαμβάνεται η απόφαση έγκαιρα όσον αφορά την μετάδοση της ροής. Επίσης, όμως πρέπει να δίνει το περιθώριο ώστε να πραγματοποιηθεί η τυχόν μετάβαση σε άλλη ροή βίντεο και να ξεκινήσει η λήψη των RTCP αναφορών παραλήπτη που θα αναφέρονται στην νέα ροή βίντεο. Για τον καθορισμό της, πραγματοποιήθηκαν δοκιμές με διάφορες τιμές ώστε να βρεθεί η κατάλληλη η οποία να πληροί τις πιο πάνω προϋποθέσεις. Η τιμή στην οποία κατέληξα είναι η 8 δευτερόλεπτα.

Αφού ληφθούν οι δύο τιμές των πληροφοριών από το αρχείο που δημιουργεί ο Darwin Streaming εξυπηρετητής, το προσαρμοστικό σύστημα εκτελεί τους κατάλληλους υπολογισμούς ώστε να εξάγει τις δύο μεταβλητές εισόδου για τον μηχανισμό απόφασης. Με βάση τις δύο αυτές μεταβλητές εισόδου, καθορίζεται η μεταβλητή εξόδου $a(kT)$ όπως ορίζεται στον αλγόριθμο ADIVIS με βάση ένα πίνακα κανόνων. Στη συνέχεια υπολογίζεται το διαθέσιμο εύρος ζώνης το οποίο χρησιμοποιείται σε κάποιους ελέγχους για να διαπιστωθεί εάν έχει μεταβληθεί σε σχέση με το προηγούμενο. Με βάση το αποτέλεσμα μπορεί να πραγματοποιηθεί η απόφαση για μετάβαση σε ψηλότερο ή χαμηλότερο στρώμα βίντεο. Όπως καθορίζεται με τον ADIVIS, η μετάβαση σε χαμηλότερο στρώμα είναι άμεση ενώ η μετάβαση σε ψηλότερο στρώμα θα πραγματοποιηθεί μετά από τη χρονική υστέρηση και εφόσον το διαθέσιμο εύρος ζώνης παραμείνει στα ίδια επίπεδα. Αν ληφθεί απόφαση για μετάβαση σε νέο στρώμα βίντεο τότε υπολογίζεται το ανάλογο στρώμα προσθέτοντας ή αφαιρώντας ένα στρώμα από τον τρέχον αριθμό στρωμάτων. Μετά τον υπολογισμό του νέου αριθμού στρωμάτων πραγματοποιείται έλεγχος αν είναι εφικτός ο συγκεκριμένος αριθμός σε περίπτωση που το αποτέλεσμα βγει εκτός του αριθμού στρωμάτων που έχει το συγκεκριμένο βίντεο. Σε τέτοια περίπτωση δεν θα πραγματοποιηθεί αλλαγή και μετάβαση σε διαφορετική ροή βίντεο.

Στη συνέχεια, αφού ληφθεί η απόφαση για μετάβαση σε νέα ροή βίντεο πρέπει να τερματιστεί το streaming της προηγούμενης ροής και να ξεκινήσει το streaming της

νέας ροής. Έτσι, το προσαρμοστικό σύστημα δίνει σήμα στον Darwin Streaming εξυπηρετητή να τερματίσει το streaming της ροής που βρίσκεται σε εξέλιξη. Αυτό πραγματοποιείται με τον τερματισμό της συγκεκριμένης RTSP συνόδου. Επιπλέον, το προσαρμοστικό σύστημα στέλλει στον εξυπηρετητή μια νέα RTSP αίτηση για την νέα ροή βίντεο. Αυτό δίνει έναν περιορισμό όσον αφορά το που είναι εγκατεστημένη η εφαρμογή client, αφού με βάση το πιο πάνω τρόπο επιβάλλεται η πλευρά του client να είναι η ίδια με την πλευρά του εξυπηρετητή.

Ο Darwin Streaming εξυπηρετητής καθώς πραγματοποιεί το streaming μιας ροής βίντεο ελέγχει για τυχόν σήμα από το προσαρμοστικό σύστημα ADIVIS για ανάγκη αλλαγής της ροής. Αυτή η επικοινωνία πραγματοποιείται και πάλι μέσω ενός αρχείου (όνομα αρχείου= Adapt.txt). Αν υπάρχει ένδειξη ότι πρέπει να πραγματοποιηθεί μετάβαση σε νέα ροή βίντεο τότε τερματίζει την RTSP σύνοδο η οποία βρίσκεται σε εξέλιξη. Αυτό πραγματοποιείται με την κλήση της μεθόδου QTSS_Teardown, η οποία ορίζεται από τον Darwin. Στη συνέχεια λαμβάνει μια νέα RTSP αίτηση για μια άλλη ροή βίντεο και έτσι λειτουργεί όπως θα λειτουργούσε σε μια λήψη μιας RTSP αίτησης για μια ροή βίντεο.

Η νέα ροή βίντεο πρέπει να ξεκινήσει από το σημείο στο οποίο σταμάτησε η προηγούμενη. Έτσι, ο εξυπηρετητής θα πρέπει με κάποιον τρόπο να γνωρίζει αυτήν τη στιγμή, δηλαδή πόσα δευτερόλεπτα της ροής έχουν αναπαραχθεί ήδη, ώστε να ξεκινήσει από αυτό το σημείο. Για να γίνει αυτό δυνατό χρησιμοποίησε το πεδίο Range το οποίο βρίσκεται σε μια RTSP αίτηση με μέθοδο PLAY. Το συγκεκριμένο πεδίο δίνει ένα διάστημα, αρχή-τέλος όσον αφορά από το που πρέπει να ξεκινήσει η ροή και που πρέπει να σταματήσει. Στο επόμενο σχήμα παρουσιάζεται η RTSP αίτηση με μέθοδο PLAY όπου μπορούμε να δούμε το πεδίο Range.

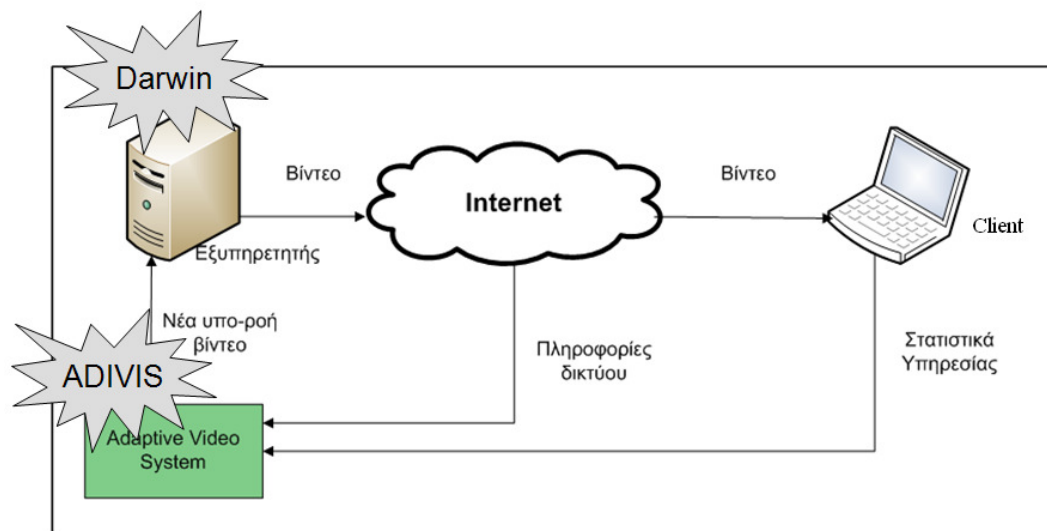
```
PLAY rtsp://192.168.2.14/sample_100kbit.mov RTSP/1.0
CSeq: 4
Range: npt=0.000000-70.000000
x-prebuffer: maxtime=2.000000
x-transport-options: late-tolerance=10
Session: 130197638632959
User-Agent: QuickTime/7.6.6 (qtver=7.6.6;os=Windows NT 5.1Service Pack 3)
```

Με βάση το πιο πάνω πεδίο Range, η συγκεκριμένη ροή βίντεο θα αναπαραχθεί από το 0 δευτερόλεπτο μέχρι και τα 70 δευτερόλεπτα. Αυτή η μέθοδος όμως, ίσως να

περιορίζει σε κάποιο βαθμό την εφαρμογή client, αφού θα πρέπει οι RTSP PLAY αιτήσεις να περιέχουν τις τιμές στο συγκεκριμένο πεδίο και ειδικά αυτή της χρονικής στιγμής για έναρξη της ροής.

Έτσι, στο σημείο όπου η RTSP αίτηση λαμβάνει επεξεργασία για καθορισμό των διάφορων παραμέτρων με βάση τα πεδία που την αποτελούν, πραγματοποιείται ένας έλεγχος ώστε σε περίπτωση που έχουμε μετάβαση να ξεκινήσει η συγκεκριμένη ροή από το σωστό σημείο. Ο έλεγχος πραγματοποιείται με την βοήθεια ενός αρχείου το οποίο περιέχει το σημείο από το οποίο πρέπει να ξεκινήσει η ροή (όνομα αρχείου=Start.txt, Ο κώδικας βρίσκεται στο αρχείο RTSPRequest.cpp στα αρχεία κώδικα του εξυπηρετητή).

Η πιο πάνω διαδικασία παρουσιάζεται στο επόμενο σχήμα.



Σχήμα 9.1: Σχηματική αναπαράσταση προτεινόμενης λύσης

Όπως βλέπουμε, στη θέση του συστήματος προσαρμοστικής μετάδοσης βίντεο βρίσκεται το σύστημα το οποίο βασίζεται στον αλγόριθμο ADIVIS και στη θέση του streaming εξυπηρετητή βρίσκεται ο Darwin Streaming εξυπηρετητής. Ο client λαμβάνει την ροή βίντεο από τον εξυπηρετητή και καθώς την λαμβάνει στέλλει περιοδικά στατιστικά υπηρεσίας που αφορούν την ποιότητα υπηρεσίας που λαμβάνει. Από αυτά τα χαρακτηριστικά χρησιμοποιείται ο ρυθμός απώλειας δεδομένων από το σύστημα ADIVIS. Το σύστημα ADIVIS λαμβάνει και τις πληροφορίες από τον πυρήνα του

δικτύου και συγκεκριμένα τον αριθμό πακέτων με το ECN bit ενεργοποιημένο. Με βάση αυτές τις πληροφορίες γίνονται οι κατάλληλες ενέργειες οι οποίες έχουν περιγραφεί πιο πάνω. Σε περίπτωση απόφασης για μετάβαση σε νέα υπό-ροη βίντεο τότε ο ADIVIS στέλλει μια νέα RTSP αίτηση στον εξυπηρετητή.

9.3 Εξαγωγή Αποτελεσμάτων

Κατά τη διάρκεια λειτουργίας του προσαρμοστικού συστήματος ADIVIS και του Darwin Streaming εξυπηρετητή και συγκεκριμένα κατά τη διάρκεια μιας μετάδοσης μιας ροής βίντεο, καταγράφονται διάφορες χρήσιμες πληροφορίες σε διάφορα αρχεία. Τα συγκεκριμένα αρχεία χρησιμοποιούνται ώστε να μπορεί να πραγματοποιηθεί μια ανάλυση μετά το τέλος μιας μετάδοσης ώστε να διαπιστωθεί ότι δικαιολογημένα παρατηρήθηκε η συγκεκριμένη συμπεριφορά του ADIVIS και του Darwin. Τα συγκεκριμένα αρχεία αποθηκεύονται στο ακόλουθο μονοπάτι στον υπολογιστή όπου είναι εγκατεστημένος ο εξυπηρετητής και ο ADIVIS: C:/Documents and Settings/ .

Θα ακολουθήσει μια συνοπτική εξήγηση για τις πληροφορίες που περιέχει κάθε αρχείο. Το αρχείο timeChanged δημιουργείται από τον Darwin (RTSPRequest.cpp) και περιέχει τις χρονικές στιγμές κατά τις οποίες η νέα ροή βίντεο (μετά από μετάβαση) πρέπει να ξεκινά. Όπως, αναφέρθηκε και πιο πάνω η νέα ροή βίντεο ξεκινά από το σημείο που σταμάτησε η αρχική ροή. Το αρχείο Request δημιουργείται από τον Darwin (QTSSMyModule1.cpp) και περιέχει τις διάφορες RTSP αιτήσεις που πραγματοποιούνται. Το RTCPinfo δημιουργείται και αυτό από τον Darwin και περιέχει τις πληροφορίες LRPS και NECN για κάθε RTCP αναφορά παραλήπτη που λαμβάνεται, όπως επίσης και το όνομα του αρχείου βίντεο για το οποίο ισχύουν. Το αρχείο LRPS δημιουργείται από το σύστημα ADIVIS και περιέχει τις τιμές LRPS για κάθε περίοδο απόφασης. Τέλος το σύστημα ADIVIS δημιουργεί ένα log αρχείο (με βάση την προηγούμενη υλοποίηση), το οποίο περιέχει πληροφορίες όπως είναι το προηγούμενο και επόμενο στρώμα βίντεο. Σε περίπτωση που θα πραγματοποιηθεί μια μετάβαση σε νέα ροή βίντεο τότε καταγράφεται η συγκεκριμένη μετάβαση, καταγράφοντας το όνομα της νέας ροής.

Στη συνέχεια, παρουσιάζεται ένα μέρος από το αρχείο RTCPinfo, το οποίο δημιουργείται από τον Darwin κατά τη συνολική διάρκεια του streaming της ροής βίντεο που ζήτησε ο client. Παρατηρούμε ότι οι RTCP αναφορές παραλήπτη

λαμβάνονται με μεγάλη συχνότητα αφού έχουμε πληροφορίες πάνω από μια φορές όταν ο αριθμός των πακέτων είναι ο ίδιος. Ο εξυπηρετητής λαμβάνει με επιτυχία τις αναφορές που στέλλει ο παραλήπτης και εμείς έχουμε πρόσβαση σε αυτές.



```
RTCPinfo - Notepad
File Edit Format View Help
LRSF 0.0000
NECN 0.5000
/sample_300kbit.mp4
RTPPacketSent:827
LRSF 0.0000
NECN 0.0000
/sample_300kbit.mp4
RTPPacketSent:827
LRSF 0.0000
NECN 0.0000
/sample_300kbit.mp4
RTPPacketSent:827
LRSF 0.0000
NECN 0.0000
/sample_300kbit.mp4
RTPPacketSent:831
LRSF 0.0000
NECN 0.5000
/sample_300kbit.mp4
RTPPacketSent:831
LRSF 0.0000
NECN 0.0000
/sample_300kbit.mp4
RTPPacketSent:836
LRSF 0.0000
NECN 0.8000
/sample_300kbit.mp4
RTPPacketSent:841
LRSF 0.0000
NECN 0.6000
/sample_300kbit.mp4
RTPPacketSent:841
LRSF 0.0000
NECN 0.0000
/sample_300kbit.mp4
RTPPacketSent:841
LRSF 0.0000
NECN 0.0000
/sample_300kbit.mp4
RTPPacketSent:849
LRSF 0.0000
NECN 0.0000
/sample_300kbit.mp4
RTPPacketSent:849
LRSF 0.0000
NECN 0.0000
/sample_300kbit.mp4
RTPPacketSent:849
LRSF 0.0000
NECN 0.0000
/sample_300kbit.mp4
RTPPacketSent:856
LRSF 0.0000
NECN 0.1429
/sample_300kbit.mp4
RTPPacketSent:856
LRSF 0.0000
NECN 0.0000
```

Σχήμα 9.2 RTCP info αρχείο

Η υλοποίηση που αναπτύχθηκε πετυχαίνει το βασικό στόχο της προτεινόμενης λύσης. Ο Darwin Streaming εξυπηρετητής λαμβάνει τις πληροφορίες από τις RTCP αναφορές παραλήπτη και δίνει τον ρυθμό απώλειας δεδομένων στον ADIVIS. Ο ADIVIS λαμβάνει την πληροφορία αυτή από το ανάλογο αρχείο και μαζί με την πληροφορία σχετικά με τον ECN μηχανισμό μετά από κατάλληλη επεξεργασία, υπολογίζει το διαθέσιμο εύρος ζώνης. Με βάση αυτό υπολογίζεται αν θα έχουμε μετάβαση σε νέα ροή. Σε περίπτωση που έχουμε, η αρχική RTSP σύννοδος τερματίζεται και ξεκινά μια νέα για την νέα ροή βίντεο. Όμως, παρόλο που τερματίζεται η αρχική σύννοδος, το

συγκεκριμένο παράθυρο της εφαρμογής για την αναπαραγωγή της ροής συνεχίζει να παραμένει ανοικτό, χωρίς όμως να συνεχίζεται η αναπαραγωγή της αρχικής ροής. Σε κάποιες περιπτώσεις από τις φορές που τρέξαμε τους κώδικες και θέσαμε σε λειτουργία τον εξυπηρετητή και το προσαρμοστικό σύστημα, παρατηρήθηκε μια μικρή καθυστέρηση μέχρι να ανοίξει το νέο παράθυρο της νέας RTSP συνόδου και να ξεκινήσει η αναπαραγωγή της νέας ροής. Κατά τη φάση στην οποία δημιουργείται το κατάλληλο σήμα προς τον εξυπηρετητή για μετάβαση σε νέα ροή και στη συνέχεια η δημιουργία και αποστολή της νέας RTSP συνόδου, ο ADIVIS δεν συνεχίζει να μετρά για να φτάσει σε περίοδο απόφασης. Αυτό πραγματοποιείται ώστε να δοθεί το κατάλληλο μικρό χρονικό διάστημα για την μετάβαση και κατά συνέπεια η λήψη των πληροφοριών που αφορούν τη νέα ροή.

ΚΕΦΑΛΑΙΟ 10

Προβλήματα

10.1 Προβλήματα

10.2 Αρχική προσπάθεια

10.1 Προβλήματα

Κατά τη φάση της υλοποίησης του στόχου της διπλωματικής αυτής εργασίας συνάντησα αρκετά προβλήματα όσον αφορά το κομμάτι σχετικά με την πραγματοποίηση της μετάβασης σε διαφορετική ροή βίντεο, όταν αυτό ήταν αναγκαίο. Η αρχική ιδέα, η οποία θα ήταν και η πιο ιδανική και αποδοτική ήταν με το που λαμβάνεται η απόφαση για μετάβαση σε μια νέα ροή βίντεο, να πραγματοποιείται με τέτοιο τρόπο ώστε η νέα ροή να ξεκινά την αναπαραγωγή από το σημείο που έμεινε η αρχική ροή. Επίσης, η RTSP σύνοδος μεταξύ εξυπηρετητή και client για τη συγκεκριμένη μετάδοση βίντεο δε θα τερματιζόταν αφού θα παρέμενε η ίδια. Με λίγα λόγια, ο εξυπηρετητής θα ήταν υπεύθυνος να ανακατευθύνει την αρχική ροή στην νέα ροή, έχοντας ως δεδομένη την νέα ροή βίντεο. Παρόλο που έγινε προσπάθεια πραγματοποίησης αυτής της διαδικασίας δεν κατέστη δυνατό να ολοκληρωθεί με επιτυχία.

Ο λόγος που συνέβηκε αυτό ήταν ότι για να πραγματοποιηθεί ο πιο πάνω στόχος έπρεπε να γίνουν αλλαγές ή και επεκτάσεις στον ήδη υλοποιημένο κώδικα του εξυπηρετητή. Αυτό είχε ως προϋπόθεση την κατανόηση της ροής λειτουργίας του εξυπηρετητή ώστε να καθοριστούν οι αναγκαίες αλλαγές που έπρεπε να πραγματοποιηθούν. Όμως, λόγω του ότι τα αρχεία κώδικα του εξυπηρετητή ήταν πολλά και έτσι υπήρχαν πολλές αναφορές από ένα αρχείο σε άλλο, ήταν δύσκολη η παρακολούθηση της ακριβούς ροής λειτουργίας του εξυπηρετητή. Ακόμα ένας παράγοντας ο οποίος επηρέαζε την πολυπλοκότητα του κώδικα ήταν η multithreaded ασύγχρονη φύση του εξυπηρετητή.

Στη συνέχεια θα γίνει μια αναφορά στην αρχική προσπάθεια πραγματοποίησης της προσαρμογής της ροής βίντεο και τα προβλήματα που παρατηρήθηκαν.

10.2 Αρχική Προσπάθεια

Όπως αναφέρθηκε στο κεφάλαιο 7, η λειτουργία του εξυπηρετητή βασίζεται στην εκτέλεση των διαφόρων modules. Το module το οποίο μας ενδιαφέρει είναι το module το οποίο πραγματοποιεί το streaming του βίντεο, δηλαδή το module το οποίο έχει εγγραφεί για τον ρόλο QTSS RTPSendPackets_Role, ο οποίος ρόλος είναι υπεύθυνος για την αποστολή των πακέτων. Έτσι, σε αυτό το module έπρεπε να γίνουν οι βασικές αλλαγές οι οποίες θα έκαναν δυνατή την προσαρμοστική μετάδοση.

Το συγκεκριμένο module εκτελεί διάφορες διαδικασίες ανάλογα με την RTSP μέθοδο η οποία καθορίζει την RTSP αίτηση. Έτσι μελετώντας τον κώδικα, παρατηρούμε ότι η μέθοδος DoSetup(...) αναφέρεται σε μια RTSP αίτηση με μέθοδο SETUP, η μέθοδος DoDescribe(...) αναφέρεται σε μια RTSP αίτηση με μέθοδο DESCRIBE και η μέθοδος DoPlay(...) αναφέρεται σε μια RTSP αίτηση με μέθοδο RTSP PLAY. Αν έχουμε για παράδειγμα την ακολουθία από RTSP αιτήσεις που παρουσιάζεται στο επόμενο σχήμα τότε θα πραγματοποιηθεί κλήση των μεθόδων DoDescribe, DoSetup, DoSetup και DoPlay.

```
DESCRIBE rtsp://192.168.2.17/sample_300kbit.mp4 RTSP/1.0
CSeq: 1
Accept: application/sdp
Bandwidth: 512000
Accept-Language: en-US
User-Agent: QuickTime/7.6.6 (qtver=7.6.6;os=Windows NT 5.1Service Pack 3)
```

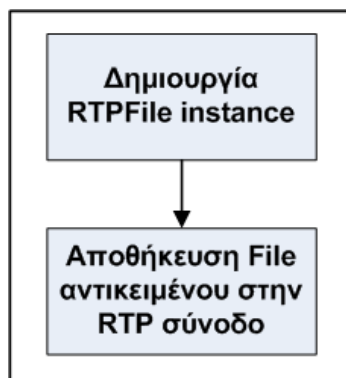
```
SETUP rtsp://192.168.2.17/sample_300kbit.mp4/trackID=3 RTSP/1.0
CSeq: 2
Transport: RTP/AVP;unicast;client_port=6974-6975
x-retransmit: our-retransmit
x-dynamic-rate: 1
x-transport-options: late-tolerance=1.400000
User-Agent: QuickTime/7.6.6 (qtver=7.6.6;os=Windows NT 5.1Service Pack 3)
Accept-Language: en-US
```

```
SETUP rtsp://192.168.2.17/sample_300kbit.mp4/trackID=4 RTSP/1.0
CSeq: 3
Transport: RTP/AVP;unicast;client_port=6972-6973
x-retransmit: our-retransmit
x-dynamic-rate: 1
x-transport-options: late-tolerance=1.400000
Session: 138663019159997
User-Agent: QuickTime/7.6.6 (qtver=7.6.6;os=Windows NT 5.1Service Pack 3)
Accept-Language: en-US
```

```
PLAY rtsp://192.168.2.17/sample_300kbit.mp4 RTSP/1.0
CSeq: 4
Range: npt=0.000000-70.000000
x-prebuffer: maxtime=2.000000
x-transport-options: late-tolerance=10
Session: 138663019159997
User-Agent: QuickTime/7.6.6 (qtver=7.6.6;os=Windows NT 5.1Service Pack 3)
```

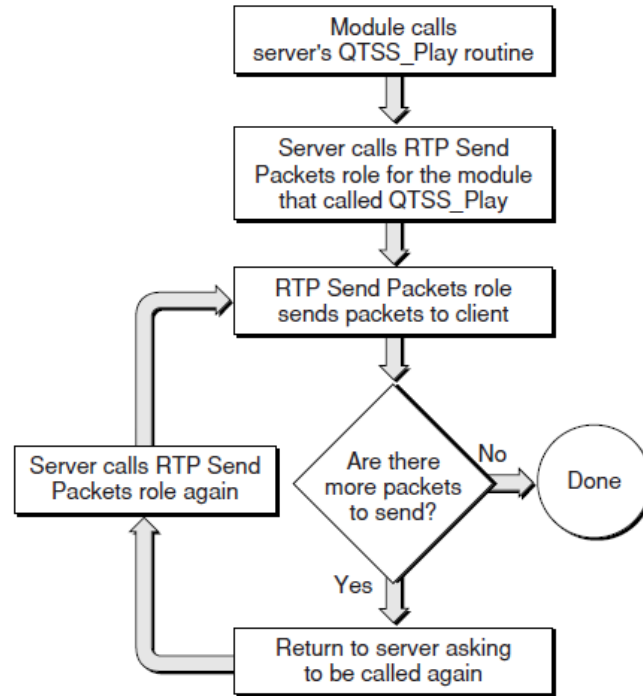
Σχήμα 10.1: RTSP αιτήσεις

Το FileModule με βάση αυτές τις μεθόδους δημιουργεί καταρχήν ένα RTPFile instance, δημιουργώντας έτσι ένα file αντικείμενο το οποίο αναφέρεται στη συγκεκριμένη ροή βίντεο για τη οποία γίνεται η RTSP αίτηση. Κατά τη φάση δημιουργίας του file αντικειμένου το περιεχόμενο του αρχείου φυλάσσεται στην cache. Το file αντικείμενο αφού δημιουργηθεί αποθηκεύεται στην RTP σύνοδο και έτσι τα χαρακτηριστικά τα οποία το αφορούν βρίσκονται σε ένα αντικείμενο τύπου QTSS_ClientSessionObject.



Σχήμα 10.2: Μια βασική λειτουργία στο FileModule

Η μέθοδος DoPlay είναι η μέθοδος η οποία είναι υπεύθυνη για το streaming της ροής βίντεο, αφού σε αυτήν υπάρχει η κλήση της QTSS_Play(...) η οποία χρησιμοποιείται για την παρουσίαση των δεδομένων, σύμφωνα με το εγχειρίδιο χρήσης. Με την κλήση αυτή ενεργοποιείται έτσι ο ρόλος QTSS RTPSendPackets_Role. Ο ρόλος αυτός ενεργοποιείται επαναληπτικά μέχρι να φτάσει στο τέλος η παρουσίαση της ροής, όπως παρουσιάζεται στο επόμενο σχήμα.



Σχήμα 10.3: Streaming

Με βάση τα πιο πάνω, κατά τη διαδικασία του streaming, και δηλαδή κάθε φορά που ενεργοποιείται ο ρόλος QTSS_RTPSendPackets_Role τοποθέτησα έναν έλεγχο ώστε σε περίπτωση που υπήρχε σήμα από τον ADIVIS, η λειτουργία να προχωρούσε προσπαθώντας να αλλάξει την ροή βίντεο. Συγκεκριμένα, σκέφτηκα πως το ιδανικό θα ήταν να πραγματοποιείται η διαδικασία δημιουργίας του file αντικείμενου ξανά λαμβάνοντας υπόψη την νέα ροή βίντεο. Όμως, παρόλο που το νέο αντικείμενο δημιουργείτο σωστά, στη συνέχεια στις επόμενες ενέργειες παρουσιάζονταν συνεχώς σφάλματα τα οποία δεν κατέστη δυνατό να λυθούν ώστε να δοκιμαστεί η συγκεκριμένη λύση. Αυτό οφείλεται στο ότι οι κλήσεις που υπήρχαν σε μεθόδους άλλων αρχείων βίντεο ήταν πάρα πολλές, γεγονός που έκανε δύσκολη την αποσφαλμάτωση αφού δεν ήταν δυνατή η παρακολούθηση της ροής λειτουργίας. Το νέο file αντικείμενο έπρεπε να αντικαταστήσει το παλιό αντικείμενο και σε αυτό το σημείο είναι που παρουσιάστηκαν τα πιο πάνω προβλήματα.

ΚΕΦΑΛΑΙΟ 11

Συμπεράσματα και Μελλοντική Εργασία

11.1 Συμπεράσματα

11.2 Μελλοντική Εργασία

11.1 Συμπεράσματα

Μελετώντας τις διάφορες έννοιες που αναπτύχθηκαν σε λεπτομέρεια σε προηγούμενα κεφάλαια, ήταν εμφανής η ανάγκη και η σημασία της προσαρμοστικής μετάδοσης βίντεο σε δίκτυα. Ένα δίκτυο που μπορεί να υποστηρίξει την προσαρμοστική μετάδοση βίντεο πρέπει να έχει τους κατάλληλους μηχανισμούς για την επίτευξη της αφού πρέπει να λαμβάνει τις διάφορες πληροφορίες σε πραγματικό χρόνο και να τις αξιοποιεί ώστε να βγάλει τα ανάλογα συμπεράσματα.

Κατά την εκτέλεση και αξιολόγηση της λειτουργίας του συστήματος ADIVIS σε συνεργασία με τον εξυπηρετητή streaming Darwin, συμέρανα πως ο αλγόριθμος ADIVIS χρησιμοποιώντας την κωδικοποίηση σε στρώματα, τον μηχανισμό ανατροφοδότησης και τον μηχανισμό απόφασης μπορεί να συνδυαστεί με έναν streaming εξυπηρετητή ώστε να πετυχαίνεται η προσαρμοστική μετάδοση βίντεο. Βασική προϋπόθεση φυσικά είναι να υπάρχει η κατάλληλη επικοινωνία μεταξύ των δύο όπως και ο σωστός συγχρονισμός. Όπως εξηγήθηκε στο κεφάλαιο της παρουσίασης της προτεινόμενης λύσης, στην δική μας περίπτωση η επικοινωνία μεταξύ του ADIVIS και του εξυπηρετητή πραγματοποιήθηκε με τη χρήση αρχείων. Είτε η μια οντότητα είτε η άλλη δημιουργούσε ένα αρχείο με συγκεκριμένη πληροφορία, το οποίο αρχείο διάβαζε η άλλη οντότητα.

Παρατηρήθηκε ότι όταν χρειάζεται και δηλαδή όταν αποφασιστεί από τον ADIVIS, πραγματοποιούνται οι σωστές μεταβάσεις σε διαφορετικές ροές βίντεο από την αρχική. Όπως ειπώθηκε όμως, η μια από τις δύο πληροφορίες που λαμβάνει ως είσοδο ο αλγόριθμος του ADIVIS δεν αντιπροσώπευε πραγματικές συνθήκες αφού δεν ήταν δυνατή η λήψη της σε πραγματικό χρόνο. Αυτό σίγουρα επηρεάζει τη συμπεριφορά της

όλης διαδικασίας αφού δεν ελέγξαμε την λειτουργία της προσαρμοστικής μετάδοσης σε πραγματικές συνθήκες. Αρχικά η συγκεκριμένη πληροφορία λάμβανε τυχαίες τιμές και έτσι δεν μπορούσαμε να γνωρίζουμε εκ των προτέρων ποια ήταν η αναμενόμενη συμπεριφορά. Για αυτόν τον λόγο, για να μπορέσουμε να βάλουμε συμπεράσματα εισάγαμε σε ένα αρχείο κάποιες τιμές οι οποίες θα οδηγούσαν σε συγκεκριμένα αποτελέσματα και μεταβάσεις, δεδομένου ότι ο ρυθμός απώλειας των δεδομένων παραμένει σταθερός χωρίς ξαφνικές αλλαγές.

11.2 Μελλοντική Εργασία

Μπορούν να πραγματοποιηθούν διάφορες αλλαγές στην παρούσα υλοποίηση ώστε το όλο σύστημα και η λειτουργία του να είναι πιο αποδοτική. Οι βελτιώσεις που μπορούν να γίνουν αφορούν και το κομμάτι του αλγόριθμου ανατροφοδότησης του αλγόριθμου ADIVIS αλλά και το κομμάτι του πως πραγματοποιείται στην τελική φάση η προσαρμοστική μετάδοση του βίντεο όταν αυτό κριθεί αναγκαίο.

Για να γίνεται σωστά η λήψη των πληροφοριών από το σύστημα το οποίο βασίζεται στον αλγόριθμο ADIVIS πρέπει να εξασφαλιστεί ότι η λήψη των πληροφοριών πραγματοποιείται σε πραγματικό χρόνο με βάση πραγματικές συνθήκες. Όπως είδαμε, το σύστημά μας λαμβάνει την πληροφορία της απώλειας δεδομένων, η οποία σχετίζεται με την πρώτη μεταβλητή εισόδου του ADIVIS, σε πραγματικό χρόνο από τον εξυπηρετητή. Η πληροφορία αυτή ανταποκρίνεται σε πραγματική τιμή αφού λαμβάνεται μετά από επεξεργασία των RTCP αναφορών παραλήπτη από τον εξυπηρετητή. Η δεύτερη όμως πληροφορία, ο αριθμός των πακέτων που έχουν ενεργοποιημένο το ECN bit δε βασίζεται σε πραγματικές τιμές, αφού πραγματοποιείται μια προσομοίωση της λήψης της. Έτσι, σε μελλοντικό στάδιο θα ήταν καλό να ενσωματωθεί ο μηχανισμός ECN στο περιβάλλον δικτύου που χρησιμοποιείται για πειραματισμό ώστε να είναι δυνατή η λήψη των συγκεκριμένων πακέτων όπως ορίζεται στον ADIVIS.

Επιπρόσθετα, παρόλο που αρχικός στόχος ήταν η προσαρμοστική μετάδοση να πραγματοποιηθεί σε κινητά MIPv6 ασύρματα δίκτυα, αυτό δεν ήταν εφικτό. Έτσι, σίγουρα σε μελλοντική εργασία η επέκταση και δοκιμή του συστήματος σε τέτοια δίκτυα θα ήταν μια καλή επιλογή ώστε να διερευνηθούν καλύτερα οι προκλήσεις που ανάγονται από την κινητικότητα των χρηστών.

Το παρών σύστημα λειτουργεί για εξυπηρέτηση ενός μόνο παραλήπτη σε μια συγκεκριμένη χρονική στιγμή λόγω του ότι το προσαρμοστικό σύστημα, δηλαδή το σύστημα το οποίο είναι βασισμένο στον ADIVIS λαμβάνει τις πληροφορίες που αφορούν μόνο ένα παραλήπτη. Θα ήταν μια καλή πρόκληση να γίνει προσπάθεια για επέκταση του συστήματος ώστε να μπορεί να ανταποκριθεί σε ένα σύνολο από παραλήπτες μιας συγκεκριμένης ροής βίντεο.

Η επικοινωνία μεταξύ του συστήματος ADIVIS και του εξυπηρετητή μπορεί να είναι πιο άμεση, δηλαδή χωρίς τη χρήση αρχείων με συγκεκριμένες πληροφορίες ώστε να υπάρχει πρόσβαση σε αυτές ή με πληροφορίες που λειτουργούν ως κάποια σήματα. Η νέα λύση μπορεί να είναι πιο αποδοτική και πιο γρήγορη αφού δε θα υπάρχουν οι διάφορες καθυστερήσεις δημιουργίας, γραψίματος και διαβάσματος του αρχείου.

Όμως αναφέρθηκε, η ιδανική λύση θα ήταν οι μεταβάσεις στις διάφορες ροές βίντεο να πραγματοποιούνται χωρίς να αλλάζει η RTSP σύνοδος, όπως συμβαίνει στην παρούσα λύση. Συγκεκριμένα, η αρχική RTSP σύνοδος τερματίζεται και γίνεται RTSP αίτηση για άλλη ροή βίντεο, δημιουργώντας έτσι μια άλλη RTSP σύνοδο. Έτσι, αυτό θα ήταν καλό σημείο για μελλοντική εργασία, δηλαδή να πραγματοποιείται προσαρμοστική μετάδοση βίντεο χωρίς να αλλάζει η RTSP σύνοδος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] J.F.Kurose and K.W.Ross, *Computer Networking A Top-Down Approach*, 4th ed.: Addison Wesley, 2008.
- [2] L.L.Peterson and B.S.Davie, *Computer Network : A Systems Approach* ,Morgan Kaufmann Series in Networking, 3rd ed.Boston:Elsevier, 2003.
- [3] C.E.Perkins , D.B.Johnson, "Mobility Support in IPv6", Proceedings of the 2nd annual international conference on Mobile computing and networking, New York, pp.27-37, Jun.1996
- [4] V.Chikarmane, C.L.Williamson, R.B.Bunt, W.L.Mackrell, "Multicast support for mobile hosts using Mobile IP: Design issues and proposed architecture", Mobile Networks and Applications 3, pp.365-379, 1998
- [5] H. Gossain, C.de M.Coreiro, D.P. Agrawal, "Multicast: Wired to Wireless", IEEE Communications Magazine, pp.116-123, Jun.2002
- [6] C.R.Lin, K.M.Wang, "Mobile Multicast Support in IP Networks", pp.1664-1672, 2000
- [7] F.Halsall, " Multimedia Communications Applications, Networks, Protocols and Standards", Addison Wesley, 2001, pp.193-235
- [8] "Implementation of Current Standards", Mar. 23, 2010. [Online]. Available: <http://msdn.microsoft.com/en-us/library/ff567651%28v=VS.85%29.aspx>
- [9] K.Rijkse, KPN Research, "H.263: Video Coding for Low-Bit-Rate Communication", IEEE Communications Magazine, pp.42-45, Dec.1996
- [10] G.J.Sullivan, T.Wiegand, "Video Compression—From Concepts to the H.264/AVC Standard", Proceedings of the IEEE , vol. 93, no. 1, pp.18-31, Jan.2005
- [11] T.Wiegand, G.J.Sullivan, Gisle Bjontegaard, A.Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, vol.13, no.7, pp.56-576, Jul.2003
- [12] H.Schwarz, D.Marpe, T.Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard"
- [13] "Real Time Streaming Protocol (RTSP)", Apr.1998 [Online]. Available: <http://www.ietf.org/rfc/rfc2326.txt>
- [14] C.Bouras, A.Gkamas, D.Primpas, K.Stamos, "Real-Time Protocols(RTP/RTCP) ", *Encyclopedia of Internet Technologies and Applications*, pp. 463 - 468, 2008.

- [15]”RTP: A Transport Protocol for Real-Time Applications”,Jul.2003 [Online]. Available: <http://www.ietf.org/rfc/rfc3550.txt>
- [16]V.Vassiliou, P.Antoniou, I.Giannakou, A.Pitsillides,”Requirements for the Transmission of Streaming Video in Mobile Wireless Networks”, *International Conference on Artificial Neural Networks*, Athens, Greece, pp.528-537, Oct.2006
- [17]M.Neophytou, K.Stavrou, V.Vassiliou, A.Pitsillides, “The Importance of Adaptive Applications in Mobile Wireless Networks”, *Proceedings of the International Conference on Software in Telecommunications and Computer Networks*, 2006
- [18] C.Bouras, A.Gkamas, D.Primpas, K.Stamos, “Multicast of Multimedia Data”, *Encyclopedia of Internet Technologies and Applications*, pp. 316 - 322, 2008.
- [19]C.Bouras, A.Gkamas, D.Primpas, K.Stamos, “Adaptive Transmission of Multimedia Data over the Internet”, *Encyclopedia of Internet Technologies and Application*,pp.16-22,2008.
- [20]P.Antoniou, V.Vassiliou, A.Pitsillides,”ADIVIS:A Novel Adaptive Algorithm for Video Streaming Over the Internet”, *The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2007
- [21]P.Antoniou, A.Pitsillides, V.Vassiliou, ”Adaptive Feedback Algorithm for Internet Video Streaming based on Fuzzy Rate Control”, 2007
- [22]”Administrator’s Guide”,
Available:http://developer.apple.com/opensource/server/streaming/qtss_admin_guide.pdf
- [23]”QuickTime Streaming Server Modules Programming Guide”, Available: http://images.apple.com/quicktime/pdf/QTSS_Modules.pdf
- [24]Kyrou A., “System for H.264 video adaptive transmission for use in wireless cellular networks”, Undergraduate thesis project, University of Cyprus, 2009.
- [25]Christodoulou S., “Use of Multicast for multimedia transmission in cellular IPv6 networks”, Undergraduate thesis project, University of Cyprus,2008
- [26]Argyrou A., “Adaptive Video Transmission”, Undergraduate thesis project, University of Cyprus, 2009

[27]Michael L.,” Use of Multicast for multimedia transmission in cellular IPv6 networks”, Undergraduate thesis project, University of Cyprus, 2007

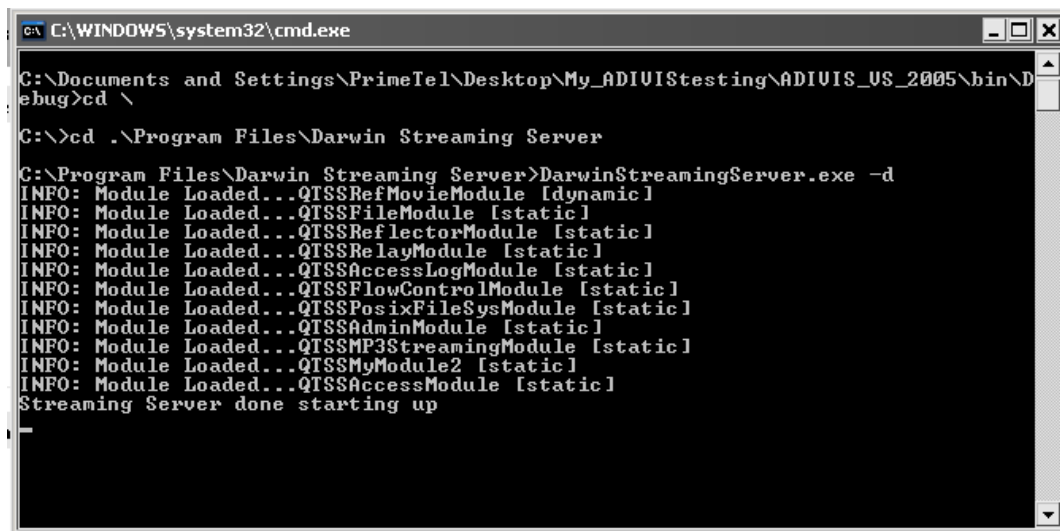
Παράρτημα Α

Σημαντικά σημεία για την εγκατάσταση και χρήση του εξυπηρετητή

Για να είναι δυνατή η λειτουργία του Darwin Streaming εξυπηρετητή πρέπει να είναι εγκατεστημένη στον ηλεκτρονικό υπολογιστή που λειτουργεί ο εξυπηρετητής μια πρόσφατη έκδοση του διερμηνέα Perl (ActivePerl 5.6 ή πιο πρόσφατη έκδοση).

Τα εκτελέσιμα αρχεία του εξυπηρετητή όπως και βασικά στοιχεία του, βρίσκονται :
c:\Program Files\Darwin Streaming Server\, μετά την εγκατάστασή (install) τους.

Για την εγκατάσταση τους WinSupport/Darwin Streaming Server → Install.bat

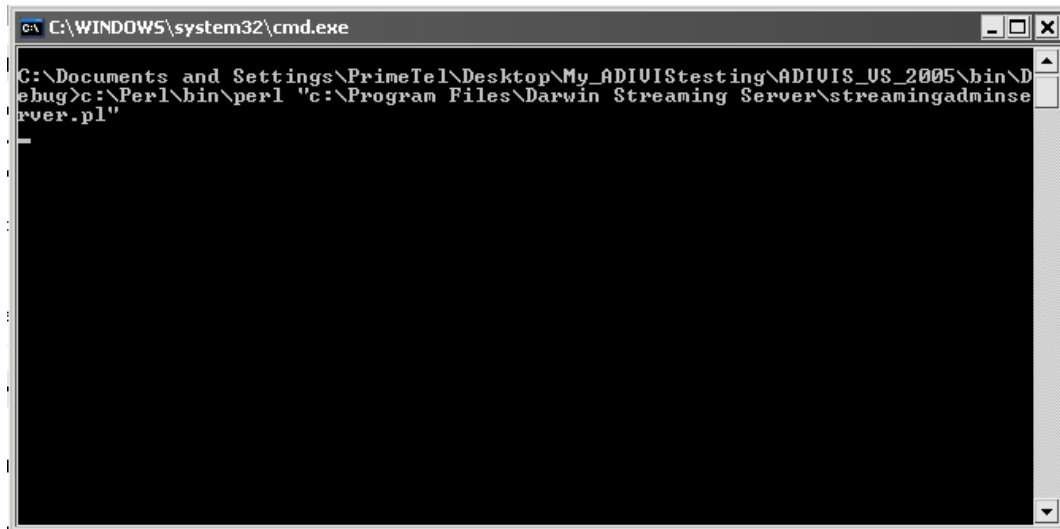


```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\PrimeTel\Desktop\My_ADIVISTesting\ADIVIS_US_2005\bin\Debug>cd \
C:\>cd .\Program Files\Darwin Streaming Server
C:\Program Files\Darwin Streaming Server>DarwinStreamingServer.exe -d
INFO: Module Loaded...QTSSRefMovieModule [dynamic]
INFO: Module Loaded...QTSSFileModule [static]
INFO: Module Loaded...QTSSReflectorModule [static]
INFO: Module Loaded...QTSSRelayModule [static]
INFO: Module Loaded...QTSSAccessLogModule [static]
INFO: Module Loaded...QTSSFlowControlModule [static]
INFO: Module Loaded...QTSSPosixFileSysModule [static]
INFO: Module Loaded...QTSSAdminModule [static]
INFO: Module Loaded...QTSSMP3StreamingModule [static]
INFO: Module Loaded...QTSSMyModule2 [static]
INFO: Module Loaded...QTSSAccessModule [static]
Streaming Server done starting up
```

Σχήμα A-1: Έναρξη λειτουργίας Darwin Streaming εξυπηρετητή από γραμμή εντολών

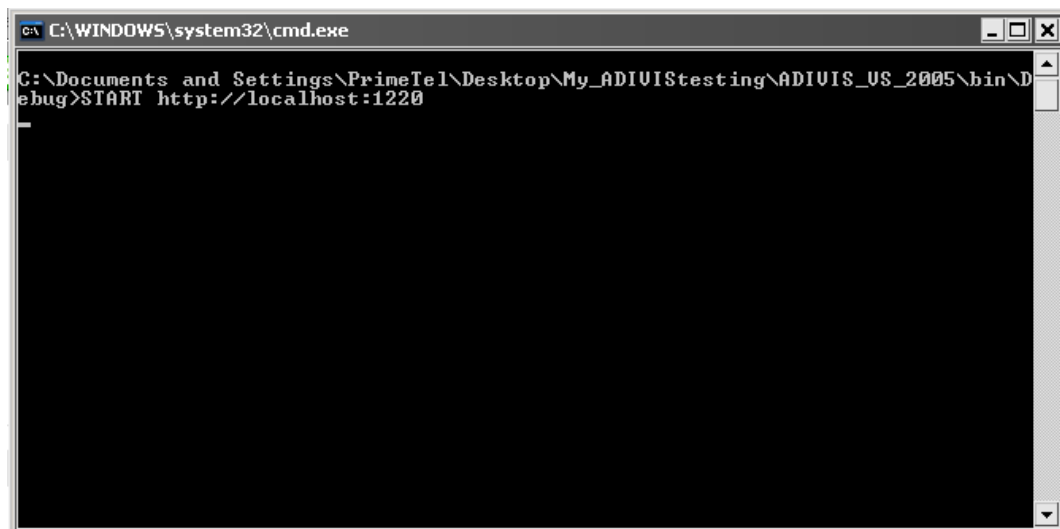
Διαχείριση εξυπηρετητή (Configure and Administrate Server)

Πρώτο βήμα:



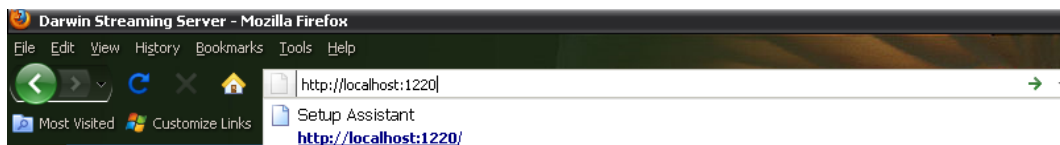
Σχήμα A-2: Διαχείριση εξυπηρετητή

Δεύτερο βήμα:

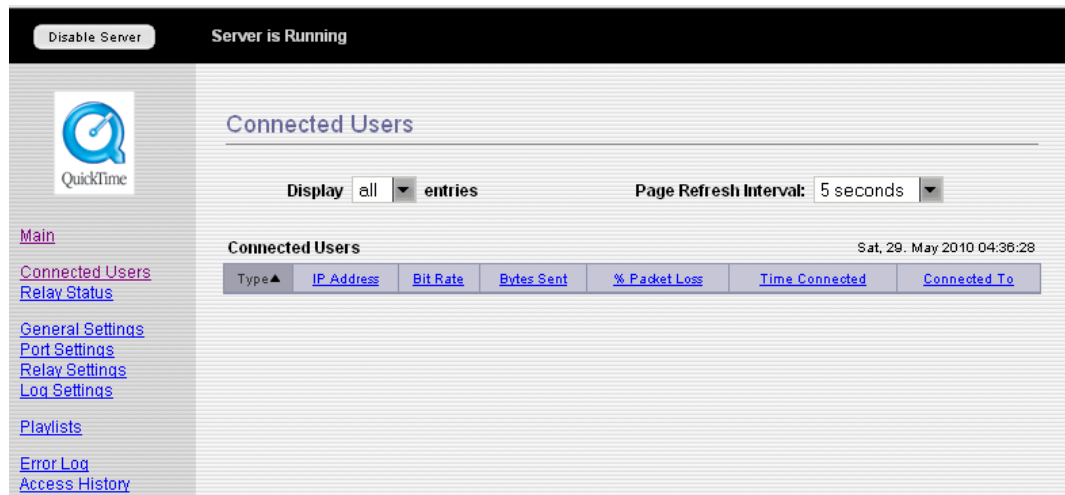


Σχήμα A-2: Διαχείριση εξυπηρετητή 2

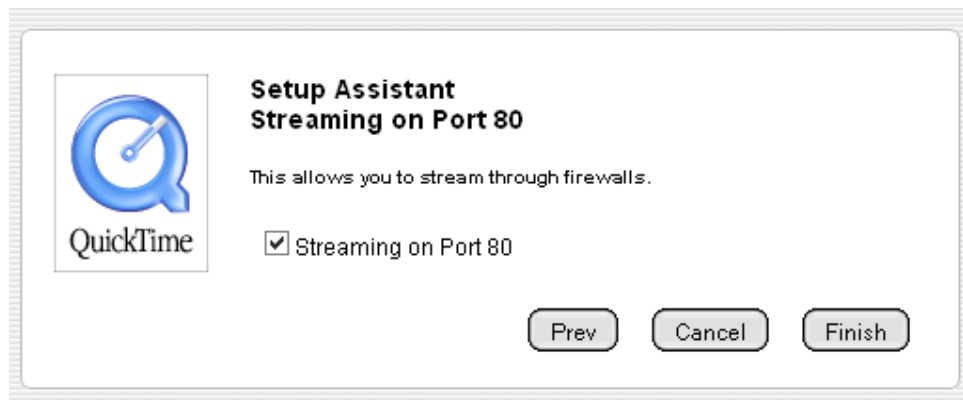
Ή εναλλακτικά



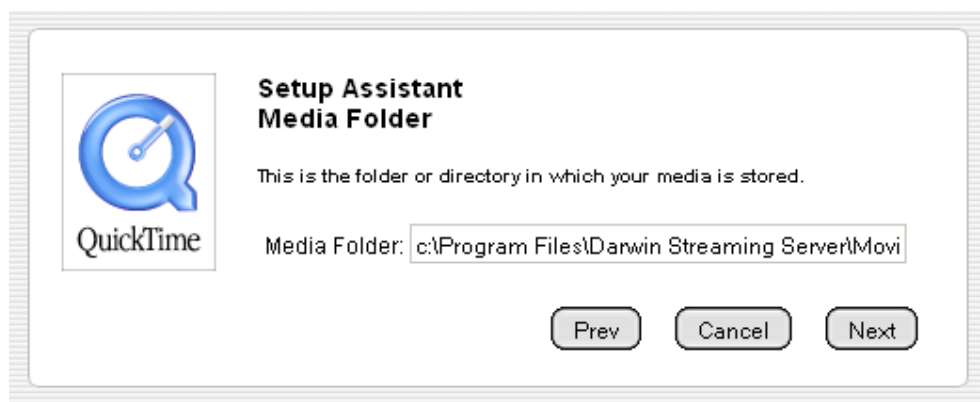
Σχήμα A-3: Διαχείριση εξυπηρετητή 3



Σχήμα A-4: Παρακολούθηση συνδεδεμένων παραληπτών



Σχήμα A-5: Streaming στο port 80



Σχήμα A-6: Επιλογή τοποθεσίας των αρχείων βίντεο τα οποία θα γίνονται streaming

Παράρτημα Β

Κώδικας για τον Darwin Streaming Εξυπηρετητή

B.1 Νέος κώδικας (QTSSModule1.cpp(στον φάκελο Server.tproj))

```
#include "QTSS.h"
#include <iostream.h>
#include <fstream.h>
#include <iomanip.h>
#include <string.h>
#include "QTSSRTSPProtocol.h"

static signed int lastNpackets=0;
static int temp;

QTSS_Error QTSSMyModuleDispatch(QTSS_Role inRole, QTSS_RoleParamPtr
inParams);
static QTSS_Error Register(QTSS_Register_Params* inParams);
static QTSS_Error CalculateLoss(QTSS_RTCPPProcess_Params* inParams);
static QTSS_Error GetFileName2(QTSS_StandardRTSP_Params* inParams);

extern "C"
{
    EXPORT QTSS_Error QTSSMyModule_Main(void* inPrivateArgs)
    {
        return _stublibrary_main(inPrivateArgs, QTSSMyModuleDispatch);
    }
}

QTSS_Error QTSSMyModuleDispatch(QTSS_Role inRole, QTSS_RoleParamPtr
inParams)
{
    switch (inRole)
    {
        case QTSS_Register_Role:
        {
            return Register(&inParams->regParams);
        }
        case QTSS_RTCPPProcess_Role:
            return CalculateLoss(&inParams-
>rtcpProcessParams);
        case QTSS_RTSPRoute_Role:
        {
            GetFileName2(&inParams-
>rtspRouteParams);
            return QTSS_NoErr;
        }
    }
}
```

```

        }
        return QTSS_NoErr;
    }

QTSS_Error Register(QTSS_Register_Params* inParams)
{
    // Do role & attribute setup
    //RTCP Process Role so that we can get LRPS and other info
    (void)QTSS_AddRole(QTSS_RTCPProcess_Role);
    //RTSP Route Role so that we can get duration, RTSP requests info
    (void)QTSS_AddRole(QTSS_RTSPRoute_Role);

    // Tell the server our name!
    static char* sModuleName = "QTSSMyModule";
    ::strcpy(inParams->outModuleName, sModuleName);

    return QTSS_NoErr;
}

QTSS_Error CalculateLoss(QTSS_RTCPProcess_Params* inParams)
{
    ofstream myfile,myfile2;
    float packetLossPercentData;
    float *packetLossPercentPointer;
    signed int *RTPPacketSentPointer;
    unsigned long theLength1,theLength2=0;
    signed int RTPPacketSent;
    char* theURLpointer=NULL;

    //Get PacketLossPercent(LRPS)
    (void)QTSS_GetValuePtr(inParams->inClientSession,
qtssCliSesPacketLossPercent, 0, (void**)&packetLossPercentPointer, &theLength1);
    //Geth the number of packets
    (void)QTSS_GetValuePtr(inParams->inClientSession,
qtssCliSesRTPPacketsSent, 0, (void**)&RTPPacketSentPointer, &theLength2);
    //Get movie file name
    QTSS_Error theError5=QTSS_GetValueAsString(inParams-
>inClientSession, qtssCliSesPresentationURL,0,&theURLpointer);
    packetLossPercentData=*packetLossPercentPointer;
    RTPPacketSent=*RTPPacketSentPointer;

    //Write the nessary information in files so that ADIVIS read them
    myfile.open ("C:/Documents and Settings/RTCPinfo.txt",ios::app);
    myfile2.open ("C:/Documents and Settings/LRSPandECN.txt");

    if(theError5==QTSS_NoErr)

```

```

    {
        myfile<<theURLpointer;
        myfile<<"\n";
    }
    myfile << "RTPPacketSent:";
    myfile.setf(ios::fixed);
    myfile.precision(4);
    myfile<<RTPPacketSent;
    myfile<<"\n";
    myfile << "LRSP ";
    myfile.setf(ios::fixed);
    myfile.precision(4);
    myfile<<packetLossPercentData;
    myfile<<"\n";

    //Calculate NECN
    double NECN;;
    signed int packetsArrived;
    packetsArrived=RTPPacketSent-lastNpackets;
    if(packetsArrived!=0)
    { temp = rand() % packetsArrived + 0;}
    lastNpackets=RTPPacketSent;
    if(packetsArrived!=0)
    { NECN=double(temp)/(double)packetsArrived;}
    else
        NECN=0;

    myfile2.setf(ios::fixed);
    myfile2.precision(4);

    myfile2<<packetLossPercentData;
    myfile2<<"\n";
    myfile2<<NECN;

    myfile<<"NECN ";
    myfile<<NECN;

    myfile<<"\n";
    myfile.close();
    myfile2.close();

    return QTSS_NoErr;
}

```

```

QTSS_Error GetFileName2(QTSS_StandardRTSP_Params* inParams)
{

```

```

        ofstream file1,file2,file3;
        unsigned long theLength,theLength2;
        double *timePointer;
        double theTime;
        char* theRequest=NULL;

        //Get the RTSP requests
        QTSS_GetValueAsString(inParams->inRTSPRequest,
qtssRTSPReqFullRequest,0,&theRequest);
        //Get the duration of the movie
        QTSS_Error theError4=QTSS_GetValuePtr(inParams->inRTSPRequest,
qtssRTSPReqStopTime, 0, (void**)&timePointer, &theLength);
        theTime=*timePointer;

        file2.open("C:/Documents and Settings/MovieDuration.txt",ios::app);
        file3.open("C:/Documents and Settings/Request.txt",ios::app);

        //The duration must be written in a file so that ADIVIS will read it
        //The RTSP requests will be written in a file so that we can see them

        QTSS_RTSPMethod* theMethod=NULL;
        QTSS_Error theErr=QTSS_GetValuePtr(inParams-
>inRTSPRequest,qtssRTSPReqMethod,0,(void**)&theMethod,&theLength2);

        //Use RTSP PLAY method in order to get the duration of the movie
        if(*theMethod==qtssPlayMethod)
        {
            file2<<theTime;

        }

        file3<<theRequest;
        file3<<"\n";

        return QTSS_NoErr;
    }

```

B.2 Κομμάτια κώδικα που επεκτείνουν την ήδη υπάρχων υλοποίηση

B.2.1 Κώδικας για μεταγλώττιση του πιο πάνω module στον εξυπηρετητή

Αρχείο QTSServer.cpp

```
QTSSModule* the = new QTSSModule("QTSSMyModule2");  
(void)theMyModule->SetupModule(&sCallbacks, &QTSSMyModule_Main);  
(void)AddModule(theMyModule);
```

B.2.2 Κώδικας για έλεγχο του πεδίου Range σε μια RTSP PLAY αίτηση (με έντονο μαύρο οι επιπρόσθετες γραμμές κώδικα)

Αρχείο QTSSRequest.cpp

```
void RTSPRequest::ParseRangeHeader()  
{ //Open the file with possibly the new start time  
    ifstream file1;  
    file1.open("C:/Documents and Settings/Start.txt");  
    float startTime;  
  
    StringParser theRangeParser(fHeaderDictionary.GetValue(qtssRangeHeader));  
  
    // Setup the start and stop time dictionary attributes  
    this->SetVal(qtssRTSPReqStartTime, &fStartTime, sizeof(fStartTime));  
    this->SetVal(qtssRTSPReqStopTime, &fStopTime, sizeof(fStopTime));  
  
    theRangeParser.GetThru(NULL, '=');//consume "npt="  
    theRangeParser.ConsumeWhitespace();  
    fStartTime = (Float64)theRangeParser.ConsumeNPT();  
  
    //Check if the video must start from a point other than first second  
    if((file1>>startTime)!=0)  
    {fStartTime=startTime;  
    ofstream file2;  
    file2.open("C:/Documents and Settings/timeChanged.txt",ios::app);  
    file2<<startTime<<endl;}  
  
    //see if there is a stop time as well.  
    if (theRangeParser.GetDataRemaining() > 1)  
    {  
        theRangeParser.GetThru(NULL, '-');  
        theRangeParser.ConsumeWhitespace();  
        fStopTime = (Float64)theRangeParser.ConsumeNPT();  
    }  
}
```

B.2.3 Κώδικας για έλεγχο για νέα μετάβαση και τερματισμό της τρέχων ροής βίντεο

Αρχείο QTSSFileModule.cpp, μέθοδος SendPackets

```
.....
//Open the file which was created by ADIVIS to check if we should stop streaming
ifstream fileRead;
fileRead.open("C:/Documents and Settings/Adapt.txt");
//Get the decision
char decision;
fileRead>>decision;
fileRead.close();

//If the decision is 'y' then we should stop streaming
if(decision=='y')
{
    (void)QTSS_Pause(inParams->inClientSession);
    inParams->outNextPacketTime =
    qtssDontCallSendPacketsAgain;
    (void)QTSS_Teardown(inParams->inClientSession);
    ofstream fileTemp;
    fileTemp.open("C:/Documents and Settings/Ready.txt");
    fileTemp.close();
    //Change decision to 'n',the streaming has stopped
    ofstream fileWrite;
    fileWrite.open("C:/Documents and Settings/Adapt.txt");
    decision='n';
    fileWrite<<decision;
    fileWrite.close();
    return QTSS_RequestFailed;
}

while (true)
{
    //Open the file which was created by ADIVIS to check if we should stop
streaming
    ifstream fileRead;
    fileRead.open("C:/Documents and Settings/Adapt.txt");
    //Get the decision
    char decision;
    fileRead>>decision;
    fileRead.close();

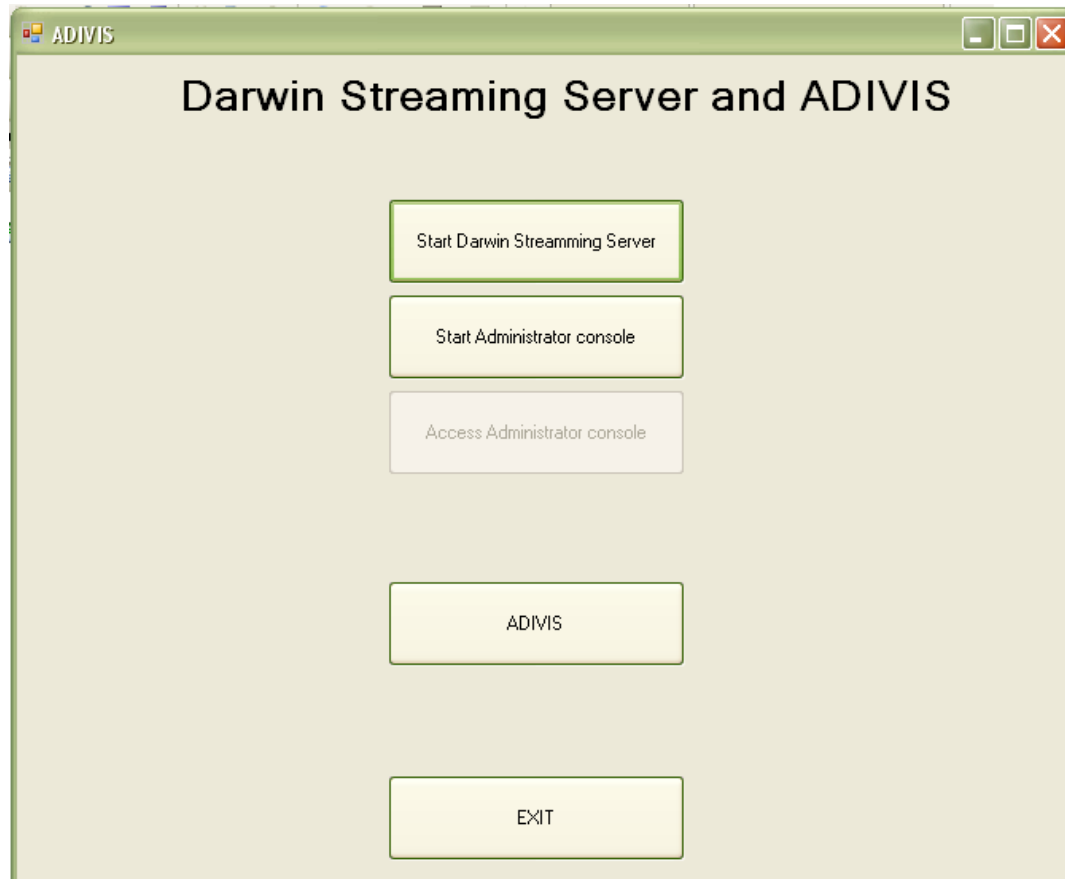
    //If the decision is 'y' then we should stop streaming
    if(decision=='y')
    {
        (void)QTSS_Pause(inParams->inClientSession);
        inParams->outNextPacketTime =
        qtssDontCallSendPacketsAgain;
        (void)QTSS_Teardown(inParams->inClientSession);
```

```
//Change decision to 'n',the streaming has stopped  
ofstream fileWrite;  
fileWrite.open("C:/Documents and Settings/Adapt.txt");  
decision='n';  
fileWrite<<decision;  
fileWrite.close();  
return QTSS_RequestFailed;  
}
```

.....

Παράρτημα Γ

Βασική Οθόνη ADIVIS-DSS



Παράρτημα Δ

Κώδικας ADIVIS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using System.Timers;
using System.IO;

namespace ADIVIS
{
    public partial class ADIVISProg : Form
    {

        System.Timers.Timer timerClock = new System.Timers.Timer();
        System.Timers.Timer streamingClock = new System.Timers.Timer();
        Random RandomClass = new Random(Environment.TickCount);

        double[,] array=new double[7,7]{{ 1.5,1.5,1.15,1.15,0,0.75,0.5},
                                           { 1.5,1.35,0,0,0,0.75,0.5},
                                           { 1.15,0,1.15,0,0,0.75,0.5},
                                           { 1.15,0,0,1.15,0,0.75,0.5},
                                           { 0,0,0,0,0.75,0.75,0.5},
                                           { 0,0,0,0,0.75,0.75,0.5},
                                           { 0.75,0.75,0.75,0.75,0.5,0.5,0.5}};

        int clockTime = 0;
        double defuzziedA=0;
        double previousLRPS = 0;
        double currentLRPS = 0;
        double currentNECN = 0;
        double previousNECN = 0;
        double DLRPS = 0;
        double DNECN = 0;

        double NECNSC = 0;
        bool hysteresis = false, prevHysteresis=false;
        double currentBW = 0, previousBW = 0, toCheckBW = 0;

        int InitialBW=0;
        bool extractNew = false;
        TextWriter tw;
```

```

string[] video_layers = new string[4]{ "sample_100kbit.mp4",
                                         "sample_300kbit.mp4",
                                         "sample_h264_100kbit.mp4",
                                         "sample_h264_1mbit.mp4"};

int previous_layer = 3;
int current_layer = 3;
bool transition = false;
int period = 0;
StreamReader ECNReader = new StreamReader(@"C:\Documents and
Settings\ECN.txt");

public ADIVISProg()
{
    InitializeComponent();
}

private void ADIVISProg_Load(object sender, EventArgs e)
{
}

//Function to calculate defuzzified output variable a(kt) base on DLRPS(kT) and
DNECNsc(kT)
void calculateDefuzzied(double DLRPS, double DNECNsc)
{
    int row=0,column=0;

    //Determine row for defuzzified array
    if (DLRPS >= 0.85)
        row = 6;
    else if (DLRPS >= 0.35)
        row = 5;
    else if (DLRPS > 0)
        row = 4;
    else if (DLRPS == 0)
        row = 3;
    else if (DLRPS >= -0.35)
        row = 2;
    else if (DLRPS >= -0.85)
        row = 1;
    else
        row = 0;

    //Determine column for defuzzified array
    if (DNECNsc >= 0.85)
        column = 6;
    else if (DNECNsc >= 0.35)
        column = 5;

```

```

        else if (DNECNESC > 0)
            column = 4;
        else if (DNECNESC == 0)
            column = 3;
        else if (DNECNESC >= -0.35)
            column = 2;
        else if (DNECNESC >= -0.85)
            column = 1;
        else
            column = 0;

        //Get defuzzified output
        defuzziedA = array[row, column];
    }

    public void Finished(Object source, ElapsedEventArgs e)
    {
        streamingClock.Stop();
        streamingClock.Enabled = false;
    }

    public void OnTimer(Object source, ElapsedEventArgs e)
    {
        timerClock.Stop();
        string line, str, strNECN;
        prevHysteresis = hysteresis;

        //Read LRPS from the file which Darwin has created
        //Create a copy of that file and get the value
        string sourceFile = @"c:\Documents and Settings\LRSPandECN.txt";
        string destFile = @"c:\Documents and Settings\tempFile.txt";
        File.Delete(destFile);
        File.Copy(sourceFile, destFile);
        FileStream s2 = new FileStream(destFile, FileMode.Open, FileAccess.Read,
FileShare.None);
        StreamReader MyStreamReader = new StreamReader(s2);
        str = MyStreamReader.ReadLine();

        //Convert to double
        currentLRPS = (Convert.ToSingle(str)) / 100;

        //Write LRPS in a log file
        StreamWriter SWnew;
        SWnew = File.AppendText(@"c:\Documents and Settings\LRPS.txt");
        SWnew.WriteLine("currentLRPS " + currentLRPS);
        SWnew.Close();
        DLRPS = currentLRPS - previousLRPS;
    }

```

```

        //Calculate scaled NECN and DNECN only if any packets were sent during this
period
        // strNECN = MyStreamReader.ReadLine();
        // currentNECN = Convert.ToSingle(strNECN);
        // NECNSC = currentNECN;
        // DNECN = NECNSC - previousNECN;

        string tempECN;
        tempECN = ECNReader.ReadLine();
        DNECN = Convert.ToSingle(tempECN);

        MyStreamReader.Close();
        s2.Close();
        period = period + 1;
        //Increment time
        this.clockTime+=8000;
        //Write parameters to log file
        line = "Time(in ms)= " + this.clockTime.ToString() + " DLRPS= " +
DLRPS.ToString() + " DNECN= " + DNECN.ToString();
        tw.WriteLine(line);

        //Calculate defuzzied output
        calculateDefuzzied(DLRPS, DNECN);

        if (defuzziedA > 1.0 && hysteresis == false) //Bandwidth increases, enter
hysteresis phase
        {
            hysteresis = true;
            toCheckBW = currentBW;
            previousBW = currentBW;
            currentBW = defuzziedA * currentBW;
            if (currentBW > InitialBW)
                currentBW = InitialBW;
            extractNew = false;
        }
        else if (defuzziedA < 1.0) //Bandwidth decreases
        {

            previousBW = currentBW;
            //if defuzzied output == 0 do not multiply since bandwidth becomes 0
            if (defuzziedA != 0.0)
                currentBW = currentBW * defuzziedA;
            if (currentBW == 0.0)
                currentBW = previousBW;
            if (currentBW > toCheckBW && hysteresis == true)
            { //Still if currentBW exceeds toCheckBW and already in hysteresis phase,
move up layer
                extractNew = true; //flag to signal main to proceed to new extraction

```

```

        hysteresis = false;
        current_layer = previous_layer + 1;
    }
    if (currentBW < toCheckBW)
        hysteresis = false;

    if (currentBW < previousBW)
    { //Transition to lower layer
        extractNew = true;
        hysteresis = false;
        current_layer = previous_layer - 1;
    }
}
else if ((defuzziedA == 0.0 || defuzziedA > 1.0) && hysteresis == true) //if
bandwidth increases and already in hysteresis phase, move up layer
{
    previousBW = currentBW;
    if (defuzziedA != 0.0)
        currentBW = defuzziedA * currentBW;

    extractNew = true; //flag to signal main to proceed to new extraction
    hysteresis = false;
    current_layer = previous_layer + 1;
    if (currentBW > InitialBW)
        currentBW = InitialBW;
}

if (extractNew == true)
{
    if (current_layer < 0 || current_layer > 3)
    { current_layer = previous_layer; }
    else transition = true;
}

line = "Current BW= " + currentBW.ToString() + " Defuzzied Output= " +
defuzziedA.ToString() + "Previous Hysteresis= " + prevHysteresis.ToString() + "
Current Hysteresis= "+hysteresis.ToString()+ " New Extraction= " +
extractNew.ToString() + " Previous Layer= "+previous_layer.ToString()+" Current
Layer= " +current_layer.ToString();
tw.WriteLine(line);
if (transition == true)
    tw.WriteLine("Transition to " + video_layers[current_layer]);
tw.WriteLine("\n");

//Change variables accordingly
previousNECN = NECNSC;
previousLRPS = currentLRPS;

```

```

previous_layer = current_layer;

timerClock.Start();
}

//Main function of current form
private void btnMeasure_Click(object sender, EventArgs e)
{
    double ending;
    string strTime;
    //Timer initialization, enabled when streaming begins
    timerClock.Elapsed += new ElapsedEventHandler(OnTimer);
    timerClock.Interval = 8000;
    timerClock.Enabled = false;

    //Wait until an RTSP request
    while (!File.Exists(@"C:\Documents and Settings\streaming.txt")) ;

    //Read the duration of the streaming from the file which Darwin has created
    string sourceFileTime = @"c:\Documents and Settings\MovieDuration.txt";
    string destFileTime = @"c:\Documents and Settings\DurationTemp.txt";
    File.Delete(destFileTime);
    File.Copy(sourceFileTime, destFileTime);
    FileStream timeFile = new FileStream(destFileTime, FileMode.Open,
FileAccess.Read, FileShare.None);
    StreamReader timeReader = new StreamReader(timeFile);
    strTime = timeReader.ReadLine();
    timeFile.Close();
    timeReader.Close();
    ending = (Convert.ToSingle(strTime)) *1000;

    //Initialize the clock for streaming
    streamingClock.Elapsed += new ElapsedEventHandler(Finished);
    streamingClock.Interval = ending;
    streamingClock.Enabled = false;

    tw = new StreamWriter("ADIVIS.log");
    string command = "";
    string toTW = "";

    btnStream.Enabled = false;

    InitialBW = 1000000;

    //Set the variables for the current and previous bandwidth
    currentBW = InitialBW;
    previousBW = currentBW;
    toCheckBW = currentBW;

```

```

//Write Data for initial sequence in file
tw.WriteLine();
toTW = "Time(in ms)= " + this.clockTime.ToString() + " DLRPS= " +
DLRPS.ToString() + " DNECN= " + DNECN.ToString();
tw.WriteLine(toTW);
toTW = "Current BW= " + currentBW.ToString() + "Previous Hysteresis= " +
prevHysteresis.ToString() + " Current Hysteresis= " + hysteresis.ToString() + " New
Extraction= " + extractNew.ToString();
tw.WriteLine(toTW);
tw.WriteLine(command);
tw.WriteLine();
int num_periods;
num_periods = (int)(ending / 8000);

//Beginning of the streaming
streamingClock.Enabled = true;

while ((streamingClock.Enabled==true)&&(period<num_periods) )
{
    //Begin timer to check every period
    timerClock.Enabled = true;

    //If transition is true then termiante RTSP session and send a new RTSP
    request for the new video file
    //Only if the transition is true, it will occur a change in the streaming
    (current layer is between 0-numberofLayers)
    if(transition==true)
    {
        timerClock.Stop();

        TextWriter adaptW = new StreamWriter(@"c:\Documents and
Settings\Adapt.txt");
        adaptW.Write('y');
        adaptW.Close();

        //Tell Darwin the start time of the new file
        TextWriter start = new StreamWriter(@"c:\Documents and
Settings\Start.txt");
        float time;
        time = period * 8;
        start.Write(time);
        start.Close();

        File.Delete(@"C:\Documents and Settings\streaming.txt");

        // Send the new RTSP request
        //First create the batch file
        TextWriter tw1 = new StreamWriter(@"c:\Documents and
Settings\RTSP.bat");

```

```

        // write a line of text to the file
        tw1.WriteLine("cd \\");
        tw1.WriteLine("cd .\\Program Files\\QuickTime\\");
        tw1.WriteLine("QuickTimePlayer.exe rtsp://localhost/" +
video_layers[current_layer]);
        tw1.Close();

        System.Threading.Thread.Sleep(4000);

        //Then run batch file
        System.Diagnostics.ProcessStartInfo p1= new
System.Diagnostics.ProcessStartInfo(@"C:\Documents and Settings\RTSP.bat");
        System.Diagnostics.Process proc1 = new System.Diagnostics.Process();
        proc1.StartInfo = p1;
        proc1.Start();

        //Wait until an RTSP request
        while (!File.Exists(@"C:\Documents and Settings\streaming.txt")) ;
        timerClock.Start();//Start the timer again

        //Set transition and extractNew to false
        transition = false;
        extractNew = false;
    }
}
//Stop Timer after the completion of the streaming
timerClock.Enabled = false;
clockTime = 0;
ECNReader.Close();

tw.Close();

File.Delete(@"C:\Documents and Settings\streaming.txt");
File.Delete(@"c:\Documents and Settings\MovieDuration.txt");
File.Delete(@"c:\Documents and Settings\Start.txt");
this.Close();
    }
}
}

```