

Ατομική Διπλωματική Εργασία

**ΣΥΣΤΗΜΑ ΜΕΤΑΔΟΣΗΣ
ΠΡΟΣΑΡΜΟΣΤΙΚΟΥ ΒΙΝΤΕΟ H.264 ΓΙΑ
ΧΡΗΣΗ ΣΕ ΑΣΥΡΜΑΤΑ ΚΙΝΗΤΑ ΔΙΚΤΥΑ**

Αχιλλέας Κύρου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ιούνιος 2009

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Σύστημα Μετάδοσης Προσαρμοστικού Βίντεο
H.264 για Χρήση σε Κινητά Ασύρματα Δίκτυα**

Αχιλλέας Κύρου

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

Ιούνιος 2009

ΕΥΧΑΡΙΣΤΙΕΣ

Έχοντας πλέον ολοκληρώσει το σημαντικότερο και πιθανώς πιο απαιτητικό μέρος της μέχρι τώρα ακαδημαϊκής μου σταδιοδρομίας θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες προς τα άτομα χωρίς των οποίων την βοήθεια κάτι τέτοιο δεν θα ήταν εφικτό.

Κατ' αρχάς θα ήθελα φυσικά να ευχαριστήσω τους γονείς και τις αδελφές μου, όχι μόνο για την συμπαράστασή τους τα τελευταία τέσσερα χρόνια, αλλά για την συνολική υποστήριξη τους μέχρι σήμερα και την κατανόηση που επιδεικνύουν στα όποια λάθη κατά καιρούς κάνω.

Συνεχίζοντας, θέλω να ευχαριστήσω τον επιβλέπων καθηγητή μου, Δρ. Βάσο Βασιλείου για τις οδηγίες και συμβουλές που μου προσέφερε καθόλη τη διάρκεια της μελέτης αυτής, όπως επίσης και τον συνεπιτηρητή και ταυτόχρονα σύμβουλο καθηγητή μου κατά τα τέσσερα χρόνια φοίτησής μου στο Πανεπιστήμιο Κύπρου, Δρ. Κωνσταντίνο Παττίχη. Και στους δύο εύχομαι κάθε επιτυχία και ευτυχία, τόσο επαγγελματική, όσο και οικογενειακή.

Τέλος, οι θερμές μου ευχαριστίες πάνε, φυσικά, και στον υποψήφιο διδάκτορα Ανδρέα Παναγίδη, για την πολύτιμη βοήθεια που μου έχει προσφέρει και το ενδιαφέρον που έχει επιδείξει τον τελευταίο χρόνο, για την εκπλήρωση των στόχων της μελέτης αυτής. Του εύχομαι καλή συνέχεια και κάθε επιτυχία.

ΠΕΡΙΛΗΨΗ

Η τεράστια ανάπτυξη που έχει επέλθει στην τεχνολογία των ασύρματων δικτύων σε συνδυασμό με την αλματώδη εξέλιξη στην παραγωγή κινητών συσκευών με δυνατότητα πρόσβασης στο Διαδίκτυο είναι δύο από τους κύριους λόγους που καθιστούν τις υπηρεσίες video μεταξύ ασύρματων χρηστών τόσο δημοφιλείς. Όπως, είναι ήδη γνωστό τέτοιες εφαρμογές απαιτούν εγγυημένα υψηλή ποιότητα υπηρεσίας, κυρίως από πλευράς εύρους ζώνης, αλλά και από άποψης απωλειών, καθυστέρησης και διακύμανσης καθυστέρησης.

Οι παράγοντες που συντελούν στη μη ύπαρξη εγγυήσεων σχετικά με την ποιότητα υπηρεσίας στα ασύρματα δίκτυα είναι πολλοί, όπως οι παρεμβολές στο σήμα των ασύρματων δικτύων, από τεχνολογίες που λειτουργούν σε παρακείμενα φάσματα συχνοτήτων, ο διαρκώς μεταβαλλόμενος αριθμός των χρηστών που έχουν πρόσβαση από συγκεκριμένη πηγή καθώς επίσης και η συνεχής κίνηση του ασύρματου κινητού χρήστη που διαφοροποιεί την ένταση του λαμβανόμενου σήματος.

Καθώς οι ρυθμοί μετάδοσης δεδομένων που υποστηρίζονται από την τεχνολογία IEEE 802.11, που είναι και η πιο διαδεδομένη όσο αφορά ασύρματα τοπικά δίκτυα, είναι περιορισμένοι, κρίνεται αναγκαία η εύρεση κάποιας τεχνικής η οποία να μεγιστοποιεί την αξιοποίηση των διαθέσιμων πόρων του εκάστοτε δικτύου, χωρίς να επηρεάζει σε μεγάλο βαθμό την ποιότητα του βίντεο που αντιλαμβάνεται ο χρήστης. Αυτό πιθανό να μπορεί να επιτευχθεί μέσω της χρήσης adaptive video streaming: δεδομένης μίας ήδη κωδικοποιημένης ακολουθίας στρωματοποιημένου βίντεο υπό πολλές διαφορετικές μορφές όσο αφορά αριθμό πλαισίων ανά δευτερόλεπτο και ευκρίνεια, γίνεται μία αξιολόγηση των περιοδικών δικτυακών συνθηκών ανά τακτά χρονικά διαστήματα και μεταβάλλεται η μεταδιδόμενη ακολουθία με την πρόσθεση ή αφαίρεση στρωμάτων. Με αυτό τον τρόπο προσαρμόζονται οι ανάγκες της εφαρμογής, στο διαθέσιμο εύρος ζώνης ώστε να παρέχεται αρκετά καλή ποιότητα υπηρεσίας στον τελικό χρήστη.

Η πρόκληση όσο αφορά την επιτυχία της τεχνικής αυτής είναι αρκετά μεγάλη, αφού θεωρητικά αναμένεται βελτίωση όσο αφορά το προσφερόμενο Quality of Service(QoS), εντούτοις αναμένεται η μελέτη των αποτελεσμάτων στην πράξη για να εξαχθούν ακριβή συμπεράσματα.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ	iii
ΠΕΡΙΛΗΨΗ	iv
ΠΕΡΙΕΧΟΜΕΝΑ	v
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ	1
ΚΕΦΑΛΑΙΟ 2 MOBILE IPv6	4
2.1 Βασικές Αρχές του Πρωτοκόλλου Mobile IPv6	4
2.2 Βασική Ορολογία Mobile IPv6	5
2.3 Δομές Δεδομένων MIPv6	7
2.4 Λειτουργία του MIPv6	7
2.5 MIPv6 Roaming between IPv6 Networks	14
2.6 MIPv6 Roaming between IPv4 Networks	15
ΚΕΦΑΛΑΙΟ 3 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΙΝΗΤΩΝ ΚΑΙ ΑΣΥΡΜΑΤΩΝ ΔΙΚΤΥΩΝ	17
3.1 Κυψελοειδή Δίκτυα Δεύτερης Γενιάς 2G	17
3.2 Κυψελοειδή Δίκτυα 2.5G (GPRS)	18
3.3 Κινητά Δίκτυα 3G	18
3.4 Ασύρματα Τοπικά Δίκτυα (WLANs) IEEE 802.11	19
ΚΕΦΑΛΑΙΟ 4 ΘΕΜΕΛΕΙΩΔΕΙΣ ΑΡΧΕΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΨΗΦΙΑΚΟΥ ΒΙΝΤΕΟ	20
4.1 Εισαγωγή	20
4.2 Θεμελιώδεις Αρχές Μείωσης Πλεονασμού	22
4.2.1 Μείωση Χωρικού Πλεονασμού	22
4.2.2 Μείωση Χρονικού Πλεονασμού	22
4.2.3 Εντροπική Κωδικοποίηση	23
4.3 Φάσεις της Τεχνικής Συμπίεσης	23
4.3.1 Ενδοπλαισιακή Κωδικοποίηση (Intra-frame Coding)	23
4.3.2 Διαπλαισιακή Κωδικοποίηση (Inter-frame Coding)	23
4.4 Είδη Πλαισίων κατά την Κωδικοποίηση Βίντεο	24
4.4.1 I- Frames (Intra- frames)	24
4.4.2 P- Frames (Predicted- Frames)	25
4.4.3 B- Frames (Bidirectional- Predicted Frames)	26
4.5 Τεχνικές Κωδικοποίησης	27
4.5.1 Εκτίμηση Κίνησης	27
4.5.2 Αντιστάθμιση Κίνησης	28
ΚΕΦΑΛΑΙΟ 5 VIDEO STREAMING ΚΑΙ ΠΡΟΤΥΠΑ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΨΗΦΙΑΚΟΥ ΒΙΝΤΕΟ	30
5.1 Video Streaming	30
5.2 Τρόποι Ανάκτησης Δεδομένων μετά από Απώλεια Πακέτου	32
5.2.1 Forward Error Correction (FEC)	32
5.2.2 Ανάκτηση Δεδομένων μέσω Παρεμβολής (Interleaving)	34
5.3 Πρότυπα Συμπίεσης Ψηφιακού Βίντεο	35
5.3.1 Η ανάγκη για συμπίεση	35
5.4 Ομάδες Εργασίας στην Ανάπτυξη Προτύπων Ψηφιακού Βίντεο	36
5.4.1 ITU-T VCEG	36
5.4.2 ISO/MPEG	36

5.4.3 JVT.....	38
5.5 Πρότυπα Κωδικοποίησης	39
5.5.1 ITU H.261	39
5.5.2 ITU H.263	40
5.5.3 MPEG-1 Part 2.....	40
5.5.4 MPEG-2 Part 2.....	42
5.5.5 MPEG-3	43
5.5.6 MPEG-4 Part 2.....	43
5.5.7 MPEG-4 Part 10/ Advanced Video Coding.....	46
ΚΕΦΑΛΑΙΟ 6 ΒΑΣΙΚΑ ΠΡΩΤΟΚΟΛΛΑ ΔΙΑΔΙΚΤΥΟΥ ΓΙΑ ΤΗ ΜΕΤΑΔΟΣΗ ΠΟΛΥΜΕΣΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ	58
6.1 Επίπεδα και Αρχιτεκτονική του Διαδικτύου.....	58
6.2 Επίπεδο Εφαρμογών	61
6.2.1 Πρωτόκολλο Session Initiation Protocol (SIP).....	61
6.2.2 Πρωτόκολλο Real Time Streaming Protocol (RTSP)	70
6.2.3 Πρωτόκολλο Real- Time Transport Protocol (RTP)	75
6.2.4 Πρωτόκολλο RTP Control Protocol (RTCP).....	77
6.3 Επίπεδο Μεταφοράς	80
6.3.1 Πρωτόκολλο Transmission Control Protocol (TCP)	81
6.3.2 Το Πρωτόκολλο User Datagram Protocol (UDP)	89
6.4 Επίπεδο Δικτύου	91
ΚΕΦΑΛΑΙΟ 7 ΣΥΣΤΗΜΑ ADIVIS (ADAPTIVE INTERNET VIDEO STREAMING) ..	94
7.1 Εισαγωγή	94
7.2 Αλγόριθμος ADIVIS.....	95
7.3 Κωδικοποίηση σε Στρώματα (Layered Encoding)	96
7.4 Μηχανισμός Ανατροφοδότησης (Feedback Mechanism)	97
7.5 Μηχανισμός Συγκεχυμένης Απόφασης (Fuzzy Decision Mechanism).....	97
ΚΕΦΑΛΑΙΟ 8 JSVM (JOINT SCALABLE VIDEO MODE)	102
8.1 Εισαγωγή	102
8.2 Βιβλιοθήκες και Εκτελέσιμα	103
8.3 Χρησιμότητα.....	106
8.4 Κωδικοποιητής H264AVCEncoderLibTestStatic	108
8.5 Εξαγωγή Bitstream BitStreamExtractorStatic	111
8.6 Μετατροπή SVC Bitstream σε AVC AvcRewriterStatic	112
8.7 Μέτρηση Ποιότητας PSNRStatic	112
ΚΕΦΑΛΑΙΟ 9 ΣΥΣΤΗΜΑ ΜΕΤΑΔΟΣΗΣ ΠΡΟΣΑΡΜΟΣΤΙΚΟΥ ΒΙΝΤΕΟ ΓΙΑ ΧΡΗΣΗ ΣΕ ΚΙΝΗΤΑ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ	114
9.1 Προτεινόμενη Λύση.....	114
9.2 Προτεινόμενος Αλγόριθμος.....	117
9.3 Σχηματική Αναπαράσταση Προτεινόμενης Λύσης	121
9.4 Ακολουθίες Video που Χρησιμοποιήθηκαν	122
9.5 Εξαγωγή Αποτελεσμάτων.....	124
9.6 Αποτελέσματα Προγράμματος	126
9.7 Παρατηρήσεις.....	131
ΚΕΦΑΛΑΙΟ 10 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ	133
10.1 Συμπεράσματα	133

10.2 Μελλοντική Εργασία	135
ΒΙΒΛΙΟΓΡΑΦΙΑ	137
ΠΑΡΑΡΤΗΜΑ Α ΒΑΣΙΚΟΣ ΚΩΔΙΚΑΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΕ ΣΤΗΝ ΥΛΟΠΟΙΗΣΗ.....	140
ΠΑΡΑΡΤΗΜΑ Β ΒΑΣΙΚΕΣ ΟΘΟΝΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ.....	160
ΠΑΡΑΡΤΗΜΑ Γ ΑΡΧΕΙΑ ΡΥΘΜΙΣΕΩΝ ΓΙΑ ΚΩΔΙΚΟΠΟΙΗΣΗ SCALABLE VIDEO	164
Γ.1 Κύριο Αρχείο Ρυθμίσεων	164
Γ.2 Αρχεία Ρυθμίσεων Στρωμάτων	165
Γ.2.1 Στρώμα 0	165
Γ.2.2 Στρώμα 1	165
Γ.2.3 Στρώμα 2	166
Γ.2.4 Στρώμα 3	166

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Από την έναρξη της επανάστασης του διαδικτύου και την καθιέρωσή του σαν σημαντικό μέρος της καθημερινής μας ζωής, η ανάπτυξη τεχνολογιών σχετικές με αυτό υπήρξε αλματώδης.

Ουσιώδες μέρος της ανάπτυξης αυτής, οφείλεται και στην αύξηση της χρήσης κινητών συσκευών με πρόσβαση στο διαδίκτυο. Αυτό όμως είχε και σαν φυσικό επακόλουθο την ανακάλυψη τεχνικών που να επιτρέπουν στους χρήστες των συσκευών αυτών να ικανοποιούνται στον ίδιο βαθμό με τους αντίστοιχους σταθερούς, κατά την εμπειρία τους στο διαδίκτυο. Κάτι τέτοιο όμως είναι αρκετά δύσκολο καθώς οι συσκευές αυτές υπόκεινται σε διάφορους περιορισμούς λόγω του μεγέθους της οθόνης τους είτε λόγω άλλων χαρακτηριστικών τους.

Μία από τις πλέον διαδεδομένες χρήσεις του διαδικτύου στις μέρες μας είναι η μεταφορά πολυμεσικού περιεχομένου και κυρίως βίντεο. Η μετάδοση βίντεο μέσω best-effort δικτύων είναι μια περίπλοκη διαδικασία. Άγνωστα και μεταβαλλόμενα με τον χρόνο εύρος ζώνης, καθυστέρηση και απώλειες καθώς επίσης και ο καθορισμός του πώς θα διαμοιραστούν δίκαια σε όλες τις ροές οι πόροι του δικτύου καθιστούν την διαδικασία ασταθή και επίπονη. Επιπλέον, σε δυναμικά περιβάλλοντα, όπως τα ασύρματα δίκτυα, η εφαρμογή αυτή είναι ακόμη πιο δύσκολη καθώς οι παράγοντες που αναφέρθηκαν πιο πάνω (εύρος ζώνης, καθυστέρηση και απώλειες πακέτων) είναι άγνωστα εκ των προτέρων με την αλλαγή θέσης του χρήστη.

Οι σημαντικότερες προκλήσεις όσο αφορά τη μεταφορά βίντεο μέσω ασύρματων δικτύων είναι το μεγάλο Round Trip Time, που δυνατό να καταστήσει τους μηχανισμούς ελέγχου λαθών αναποτελεσματικούς, και η αδυναμία εφαρμογής του ελέγχου συμφόρησης, καθώς οι απώλειες πακέτων λόγω σφαλμάτων στα ίδια τα πακέτα είναι συχνό φαινόμενο στα ασύρματα δίκτυα.

Τα πιο πάνω χαρακτηριστικά ενδυναμώνουν σε μεγάλο βαθμό το πρόβλημα ετερογένειας που παρατηρείται τόσο στις απαιτήσεις των χρηστών κινητών συσκευών, όσο και στις

δυνατότητες ανταποκρίσεις των ασύρματων δικτύων σε αυτές. Επιπλέον, το περιορισμένο μέγεθος ζώνης συχνοτήτων το οποίο μπορούν να χρησιμοποιηθούν από αυτά τα δίκτυα, καθώς επίσης και ο ανταγωνισμός ανάμεσα στους παροχείς υπηρεσιών μεγάλωνουν τον βαθμό δυσκολίας εκτέλεσης της εφαρμογής αυτής. Λύση δεν μπορεί να θεωρηθεί η ενδυνάμωση όλων των δικτύων σε κάποιο μονοπάτι επικοινωνίας, αφού η απόδοση θα εξαρτάται από το πιο αδύναμο από αυτά, ενώ παράλληλα θα υπάρχει υποχρησιμοποίηση των πόρων των υποδικτύων αφού η συμφόρηση παρατηρείται μόνο σε ορισμένα χρονικά διαστήματα.

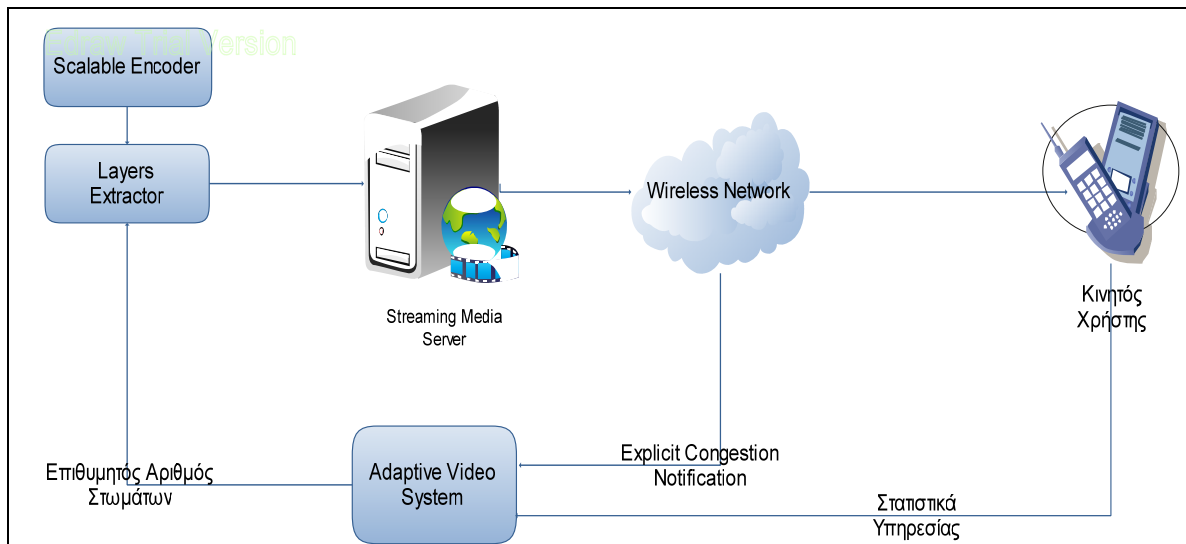
Έχοντας υπόψη τα πιο πάνω δεδομένα, καταλήγουμε στο συμπέρασμα ότι μία λύση για το πρόβλημα αυτό θα ήταν η χρήση προσαρμοστικού Quality of Service (QoS). Με αυτό τον τρόπο επιτυγχάνεται τόσο καλύτερη διαχείριση των πόρων του δικτύου, όσο και αυξημένη δυνατότητα του δικτύου. Το πλαίσιο εργασίας του προσαρμοστικού QoS μπορεί να χωριστεί σε επίπεδο εφαρμογών και επίπεδο δικτύου. Από πλευράς εφαρμογών αναμένεται η εφαρμογή να προσαρμόζεται στο διαθέσιμο QoS, ενώ το δίκτυο να προσαρμόζεται στις QoS απαιτήσεις της ίδιας της εφαρμογής. Πιο συγκεκριμένα, για την περίπτωση που η εφαρμογή είναι το βίντεο μπορούν να μεταβληθούν οι παράγοντες λόγος πλαισίου, βάθους χρώματος, διαστάσεις εικόνας και ιεραρχικής κωδικοποίησης, ενώ το δίκτυο μπορεί να κρίνει κατά πόσο πρέπει να μειώσει τους πόρους που προσφέρει στην εφαρμογή αν χρειαστεί, ή αντίθετα να προσφέρει περισσότερους. Με αυτό τον τρόπο καθίσταται δυνατή η επίτευξη μεγαλύτερης ικανοποίηση του χρήστη από το τελικό αποτέλεσμα, δηλαδή από την χωρίς διακοπές λήψη και αναπαραγωγή υψηλής ποιότητας βίντεο.

Τέλος, ακόμα ένας παράγοντας που μπορεί να βοηθήσει στην επίλυση του προβλήματος είναι η επιλογή της σωστής μορφής βίντεο που θα μεταφέρεται. Συγκεκριμένα, στα πλαίσια της μελέτης αυτής θα χρησιμοποιηθεί βίντεο τύπου H.264, που προσφέρει την καλύτερη δυνατή μέχρι τώρα συμπίεση, ψηλής ποιότητας βιντεοροή με χαμηλό bitrate. Επιπλέον, παρέχει την δυνατότητα scalability τόσο χωρικά, όσο και χρονικά.

Στόχος της έρευνας αυτής είναι, μέσα από την μελέτη των διαθέσιμων πρωτοκόλλων για μεταφορά πολυμεσικού περιεχομένου και των τεχνικών μέσω των οποίων η εφαρμογή αυτή υλοποιείται, η δημιουργία ενός συστήματος το οποίο να μπορεί να προσαρμόζει το περιεχόμενο που αποστέλλεται βάσει των μετρήσεων που λαμβάνονται από τον πυρήνα

του δικτύου και κατόπιν αξιολόγησης στατιστικών σχετικά με την ποιότητα της υπηρεσίας που παρουσιάζεται.

Μία πλήρης σχηματική αναπαράσταση της αρχιτεκτονικής που είναι δυνατό να επιτύχει τον πιο πάνω στόχο φαίνεται στο Σχήμα 1.1.



Σχήμα 1-1: Αναπαράσταση Πλήρους Συστήματος για Προσαρμοστική Μετάδοση Βίντεο

ΚΕΦΑΛΑΙΟ 2

MOBILE IPv6

2.1 Βασικές Αρχές του Πρωτοκόλλου Mobile IPv6

2.2 Βασική Ορολογία Mobile IPv6

2.3 Δομές Δεδομένων MIPv6

2.4 Λειτουργία του MIPv6

2.5 MIPv6 Roaming between IPv6 Networks

2.6 MIPv6 Roaming between IPv4 Networks

2.1 Βασικές Αρχές του Πρωτοκόλλου Mobile IPv6

Ως συνέπεια της αυξημένης ζήτησης των χρηστών για χρήση κινητών συσκευών για πρόσβαση στο διαδίκτυο, ήταν η ανάπτυξη του πρωτοκόλλου MIPv6, που να υποστηρίζει τις κινητές συνδέσεις, ως υποσύνολο του πρωτοκόλλου IPv6. Επίσης, πρόκειται για την αναβαθμισμένη έκδοση του Mobile Internet Protocol version 4 (MIPv4) και σχεδιάστηκε για τον διαχωρισμό των κινητών συσκευών χρησιμοποιώντας διευθύνσεις IPv6.

Στην κλασσική δρομολόγηση IP, όπου ουσιαστικά κάθε διεύθυνση αντιστοιχεί σε μια τοπολογία, οι μηχανισμοί δρομολόγησης βασίζονται στο γεγονός ότι κάθε κόμβος θα έχει πάντα το ίδιο σημείο σύνδεσης στο δίκτυο, καθώς και το ίδιο network link μέσω του οποίου θα συνδέεται, πράγμα που εξακριβώνεται χάρις στην IP διεύθυνση. Αυτό έχει σαν αποτέλεσμα την ανάγκη ρύθμισης της IP διεύθυνσης μιας συσκευής καθώς επίσης και του subnet mask και του default router της, σε περίπτωση αποσύνδεσής της από το δίκτυο και επανασύνδεσής της σε κάποιο διαφορετικό. Αυτό κρίνεται αναγκαίο ώστε να καθίσταται δυνατή η αποστολή πακέτων στους κόμβους αυτούς μέσω των δρομολογητών του δικτύου, αφού σε αντίθετη περίπτωση δεν θα ήταν γνωστές στους δρομολογητές οι απαραίτητες πληροφορίες τοποθεσίας της συσκευής.

Για αυτό τον λόγο ήταν αναγκαία η ανάπτυξη του MIPv6, αφού το πρωτόκολλο αυτό επιτρέπει σε ένα κόμβο να διατηρεί συνδέσεις, ενώ κινείται από ένας υποδίκτυο σε ένα άλλο. Η αναγνώριση κάθε κινητής συσκευής επιτυγχάνεται βάσει του home address της, ανεξαρτήτως του δικτύου στο οποίο ενώνεται. Σε περίπτωση σύνδεσης σε foreign network, ο home agent λαμβάνει τα στοιχεία της τοποθεσίας της κινητής συσκευής από την ίδια τη συσκευή και ακολούθως αυτός είναι υπεύθυνος να μεταφέρει τα πακέτα που προορίζονται για την συσκευή αυτή στην τρέχουσα τοποθεσία της. Η ορολογία που χρησιμοποιήθηκε, καθώς και επιπλέον ορισμοί επεξηγούνται με σαφήνεια στη συνέχεια.

2.2 Βασική Ορολογία Mobile IPv6

Home Address (HAdd): Πρόκειται για την μόνιμη IP διεύθυνση που ανατίθεται σε μια κινητή συσκευή και παραμένει αναλλοίωτη ανεξάρτητα από την παρούσα τοποθεσία της συσκευής. Είναι χρήσιμη και στην περίπτωση αποστολής πακέτων από τον κινητό κόμβο, έτσι ώστε να ενημερώνεται ο παραλήπτης για την HA του αποστολέα.

Home Network (HN): Το δίκτυο από το οποίο η κινητή συσκευή έχει πάρει την μόνιμη IP διεύθυνσή της.

Care-of Address (CoA): Μια IP διεύθυνση που συσχετίζεται με έναν κινητό κόμβο όταν αυτός βρίσκεται σε ένα ξένο υποδίκτυο και κατ' επέκταση σε μια ξένη σύνδεση. Το πρόθεμα της διεύθυνσης αυτής είναι το πρόθεμα του ξένου δικτύου. Είναι ουσιαστικά μια προσωρινή IP διεύθυνση και επιτρέπει την μεταφορά πακέτων προς τη συσκευή αυτή όταν είναι συνδεδεμένη σε δίκτυο διαφορετικό από το Home Network.

Δέσιμο (Binding): Η συσχέτιση του Home Agent (HA) ενός κινητού κόμβου με μια CoA, μαζί με την εναπομείνασα διάρκεια ζωής της σύνδεσης αυτής.

Correspondent Node (CN): Ένας κόμβος με τον οποίο ο κινητός κόμβος είναι σε επικοινωνία. Ο CN μπορεί να είναι είτε κινητός είτε σταθερός.

Foreign Link (FL): Οποιαδήποτε άλλη σύνδεση εκτός από το Home Link του κόμβου.

Foreign Subnet Prefix (FSP): Οποιοδήποτε άλλο IP πρόθεμα υποδικτύου εκτός από το home πρόθεμα του δικτύου.

Foreign Network (FN): Οποιοδήποτε δίκτυο διαφορετικό από το Home Network μίας κινητής συσκευής στο οποίο είναι δυνατό η συσκευή να συνδεθεί.

Home Link (HL): Η σύνδεση στην οποία καθορίζεται το Home Subnet Prefix ενός κινητού κόμβου. Οι τυπικοί μηχανισμοί δρομολόγησης, θα παραδώσουν πακέτα που προορίζονται για κάποιο κόμβο στο HL του.

Home Registration: Εγγραφή ενός κινητού κόμβου σε μια CoA.

Home Subnet Prefix: Το IP subnet prefix που αντιστοιχεί στο home address ενός κινητού κόμβου.

Home Agent (HA): Ένας δρομολογητής στο home network της κινητής συσκευής, ο οποίος διατηρεί πληροφορίες σχετικά με την παρούσα τοποθεσία της συσκευής, όπως αυτές αναγράφονται στην care-of address της. Ο κάθε κινητός κόμβος είναι υπεύθυνος να εγγράψει την CoA του στον δρομολογητή αυτό. Όταν ο κινητός κόμβος βρίσκεται εκτός του home network του, ο HA δέχεται πακέτα που προορίζονται για την Home Address του κόμβου αυτού, τα ενθυλακώνει (tunneling) και τα προωθεί στην CoA που του αντιστοιχεί. Με αυτό τον τρόπο αποφεύγεται η ανάγκη αλλαγής της IP διεύθυνσης της συσκευής κάθε φορά που αυτή ενώνεται από διαφορετικό δίκτυο.

Επιλογές Προορισμού Mobile IPv6

Σαν IPv6 Destination Options (Επιλογές Προορισμού), ορίζονται όλα τα μηνύματα του MIPv6. Αυτές οι επιλογές χρησιμοποιούνται στο IPv6 για την παροχή και την μεταφορά των επιπρόσθετων πληροφοριών που εξετάζονται μόνο στον τερματικό κόμβο δηλαδή από τον παραλήπτη ενός πακέτου. Ακολουθώς, αναφέρονται και εξηγούνται οι Επιλογές Προορισμού του MIPv6.

- **Binding Update:** Χρησιμοποιείται από ένα MN για να ενημερώσει το HA ή κάθε άλλο CN για την τρέχουσα CoA.
- **Binding Acknowledgement:** Χρησιμοποιείται για την επιβεβαίωση λήψης ενός μηνύματος Binding Update.
- **Binding Request:** Χρησιμοποιείται από κάποιο κόμβο για να ζητήσει από τον MN να στείλει πληροφορίες με την μορφή Binding Update που να περιέχουν την τρέχουσα CoA του κινητού κόμβου.

2.3 Δομές Δεδομένων MIPv6

Οι δομές δεδομένων που χρησιμοποιούνται από το πρωτόκολλο αυτό είναι οι ακόλουθες.

Binding Cache: Κάθε Mobile IPv6 κόμβος έχει μια binding Cache μνήμη η οποία χρησιμοποιείται για την αποθήκευση των bindings για άλλους κόμβους. Για παράδειγμα όταν ένας κόμβος λάβει ένα Binding Update, θα το φυλάξει στην Binding Cache του. Κάθε φορά που ένα πακέτο αποστέλλεται, γίνεται μια αναζήτηση στη Binding Cache για μια εγγραφή που αφορά τον παραλήπτη. Αν βρεθεί μια εγγραφή, το πακέτο αποστέλλεται στη CoA του παραλήπτη και όχι στον HA του.

Binding Λίστα Ενημέρωσης: Κάθε κινητός κόμβος έχει μια binding λίστα ενημέρωσης η οποία χρησιμοποιείται για να αποθηκεύονται πληροφορίες σχετικά με κάθε Binding Update που έχει σταλεί από αυτό τον κινητό κόμβο και του οποίου η διάρκεια ζωής δεν έχει λήξει ακόμη. Περιέχει όλα τα Binding Updates που έχουν σταλεί σε οποιοδήποτε CN, είτε σταθερό είτε κινητό και στον HA του.

Home Agents List: Για κάθε HL όπου ένας κόμβος λειτουργεί σαν HA, δημιουργεί μια λίστα η οποία περιέχει πληροφορίες για όλους τους άλλους HA σε αυτή τη σύνδεση. Οι πληροφορίες λαμβάνονται από τις διαφημίσεις των δρομολογητών που στέλνουν οι HAs. Στις διαφημίσεις αυτές, είναι ενεργοποιημένο το home agent bit, αν ο αποστολέας είναι HA σε αυτή τη σύνδεση. Οι πληροφορίες για τους άλλους HAs χρησιμοποιούνται από τον μηχανισμό Dynamic Home Agent Discovery. Ο μηχανισμός αυτός επιτρέπει σε ένα κινητό κόμβο να ανακαλύπτει δυναμικά την IP διεύθυνση ενός HA στο HL του, στον οποίο μπορεί να εγγράψει την primary care-of address του κατά την διάρκεια που βρίσκεται μακριά από το home δίκτυό του.

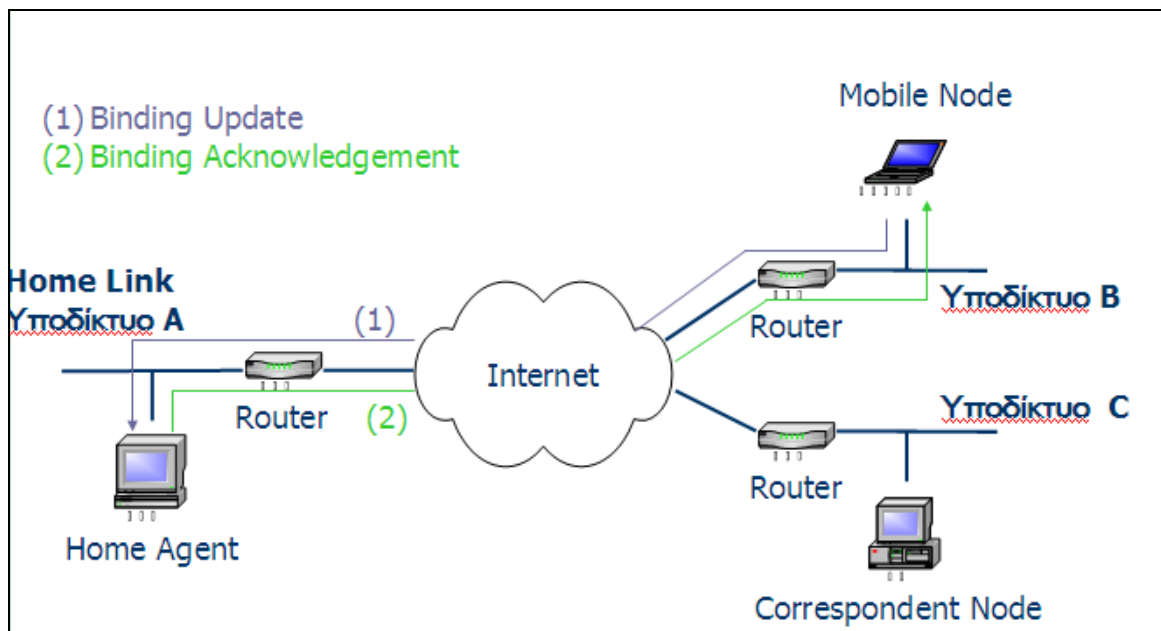
2.4 Λειτουργία του MIPv6

Για να περιγράψουμε την βασική λειτουργία του Mobile IPv6, χρησιμοποιούμε ένα δίκτυο το οποίο περιέχει τρία υποδίκτυα (Υποδίκτυο A, Υποδίκτυο B, Υποδίκτυο C). Στο Υποδίκτυο A υπάρχει ένας δρομολογητής ο οποίος λειτουργεί σαν Home Agent. Αυτή η σύνδεση είναι επίσης το HL κάποιου κινητού κόμβου ο οποίος έχει μετακινηθεί από την

σύνδεση Α σε κάποια άλλη σύνδεση (έστω σύνδεση Β). Επιπλέον, υπάρχει ένας CN (κινητός ή σταθερός) στην τρίτη σύνδεση (έστω σύνδεση Γ).

Διαδικασία Εγγραφής Home Agent (Home Agent Registration)

Μόλις ένας κινητός κόμβος αναγνωρίσει ότι έχει μετακινηθεί από μια σύνδεση σε άλλη και ανακαλύψει ένα καινούριο default δρομολογητή, κάνει τη λειτουργία του address autoconfiguration (stateful ή stateless). Αυτή η νέα διεύθυνση χρησιμοποιείται σαν η CoA του. Το πρόθεμα της CoA είναι το πρόθεμα της σύνδεσης στην οποία είναι τώρα ο κινητός κόμβος. Έτσι όλα τα πακέτα που προορίζονται για τον κόμβο αυτό, αποστέλλονται στην σύνδεση στην οποία βρίσκεται. Ο κινητός κόμβος εγγράφει την CoA του στον HA του, μέσω του HL.



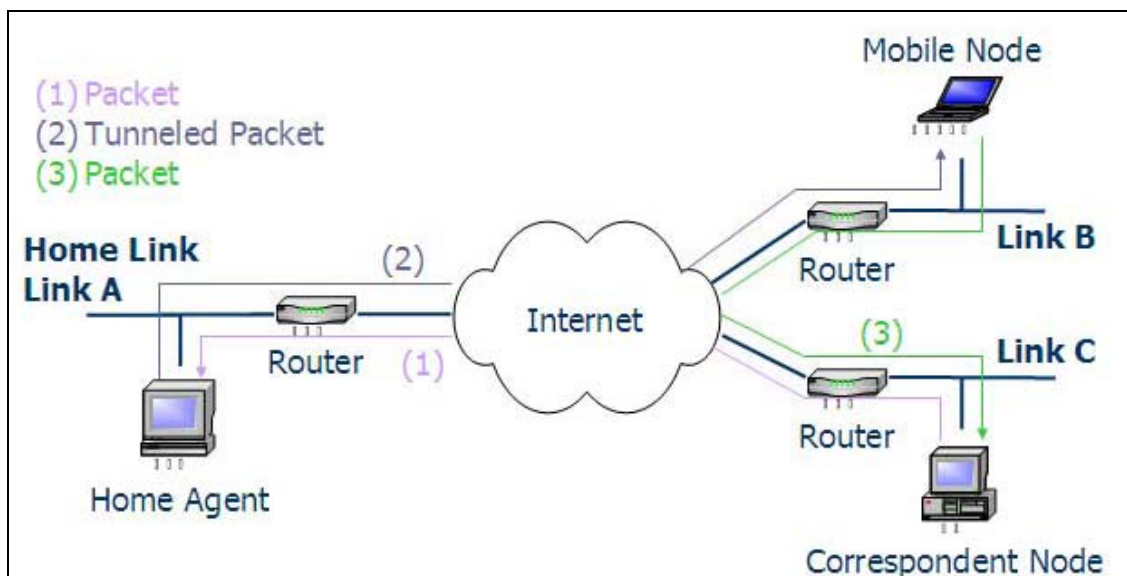
Σχήμα 2-1: Διαδικασία Εγγραφής Home Agent

Ο κινητός κόμβος στο υποδίκτυο Β στέλνει ένα πακέτο στον HA του που περιέχει μια Binding Update επιλογή προορισμού(destination option). Ο HA με τη σειρά του εγγράφει αυτό το Binding και επιστρέφει ένα πακέτο με μια Binding Acknowledgement επιλογή προορισμού.

Βελτιστοποίηση Διαδρομής (Route Optimization)

Αφού ο HA έχει εγγράψει την CoA του κινητού κόμβου, είναι υπεύθυνος για πακέτα που προορίζονται στο home address του κινητού δικτύου. Έτσι χρησιμοποιεί την μέθοδο Proxy Neighbor Discovery. Με την μέθοδο αυτή, ο HA στέλνει μια διαφήμιση γείτονα (Neighbor Advertisement) στο HL εκ μέρους του κινητού κόμβου. Ο HA απαντά και σε γειτονικούς ερεθισμούς (Neighbor Solicitations) εκ μέρους του κινητού κόμβου. Κάθε πακέτο που αποστέλλεται στο home address του κινητού κόμβου, παραλαμβάνεται από τον HA και αποστέλλεται στην CoA του κόμβου, με IPv6 encapsulation.

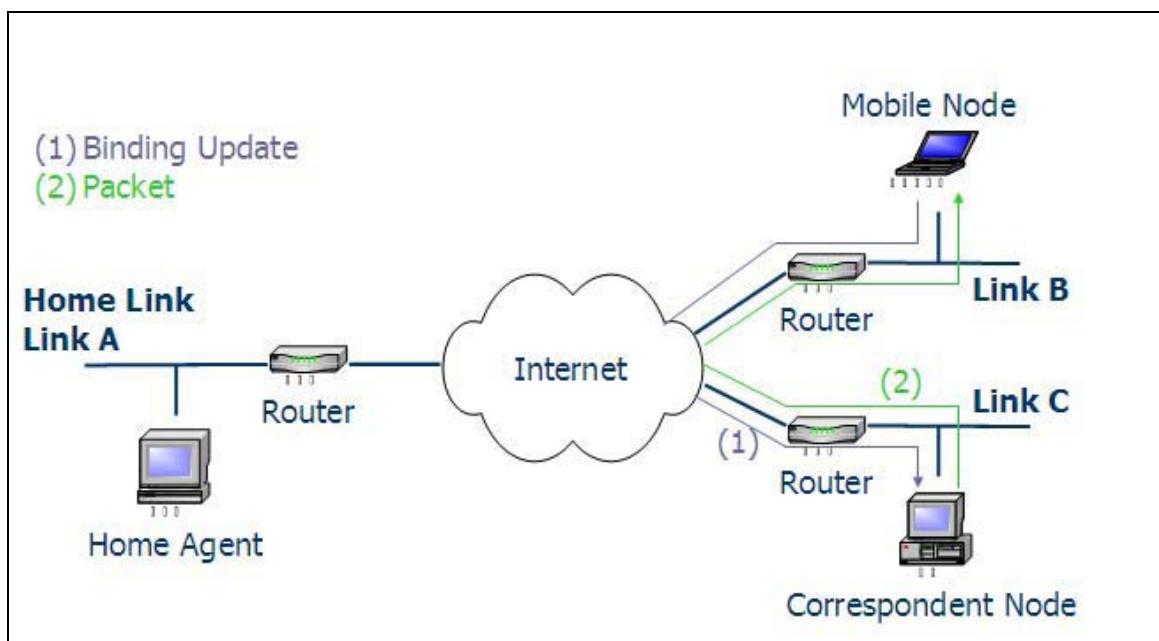
Αν ο κινητός κόμβος στείλει πακέτα σε ένα οποιοδήποτε άλλο κόμβο, τα στέλνει απευθείας στον προορισμό τους. Ο κινητός κόμβος θέτει σαν τη διεύθυνση αποστολέα στα πακέτα που στέλνει την CoA του και συμπεριλαμβάνει μια Home Address επιλογή προορισμού. Επειδή η home address είναι στατική, σε αντίθεση με την CoA, επιτρέπει σε κάθε CN τη «διαφανή» χρήση της CoA για στρώματα πάνω από την υποστήριξη του IPv6. Τα πιο ψηλά επίπεδα, συμπεριλαμβανομένου και των προγραμμάτων εφαρμογών δεν προσέχουν την CoA, μόνο την home address.



Σχήμα 2-2: Δρομολόγηση Πακέτου

Αν ένας κινητός κόμβος επικοινωνεί με ένα CN καθώς είναι μακριά από το home subnet του, τα πακέτα δρομολογούνται από τον CN στον HA, από τον HA στον κινητό κόμβο και από τον κινητό κόμβο στον CN. Αυτή η «ανωμαλία» δρομολόγησης ονομάζεται Τριγωνική Δρομολόγηση.

Για την αποφυγή του προβλήματος αυτού, ένας κινητός κόμβος μπορεί να στείλει Binding Updates σε οποιοδήποτε CN, είτε κινητό είτε σταθερό. Αυτό επιτρέπει σε IPv6 CN να φυλάζουν την τρέχουσα CoA του κόμβου αυτού και να στέλνουν πακέτα απευθείας στην κινητή συσκευή χωρίς την ενδιάμεση μεσολάβηση του HA.



Σχήμα 2-3: Διαδικασία Binding Update

Οποιοσδήποτε IPv6 κόμβος που πρόκειται να στείλει ένα πακέτο, ελέγχει πρώτα την Binding Cache του για την συγκεκριμένη διεύθυνση προορισμού. Αν υπάρχει μια εγγραφή, θα στείλει το πακέτο στον κινητό κόμβο χρησιμοποιώντας ένα routing header. Η διαδρομή που καθορίζεται από αυτό το routing header έχει δυο hops. Το πρώτο hop είναι η CoA και το δεύτερο το home address του κινητού κόμβου. Έτσι το πακέτο πηγαίνει απευθείας στην CoA του κόμβου. Στην συνέχεια, αφού ο κινητός κόμβος λάβει το πακέτο, το προωθεί στο

επόμενο hop που καθορίζεται στο routing header. Αφού το τελευταίο hop είναι η home address του κινητού κόμβου, το πακέτο θα σταλεί στο home address.

Αν η Binding Cache δεν έχει καμία εγγραφή, το πακέτο θα σταλεί κανονικά. Μετά θα δρομολογηθεί στο συγκεκριμένο δίκτυο και θα παραληφθεί από τον κόμβο προορισμού. Στην περίπτωση που ο προορισμός είναι ένας κινητός κόμβος μακριά από το home subnet του, το πακέτο θα παραληφθεί από τον HA του στο HL και θα σταλεί στον κινητό κόμβο. Με την παραλαβή του πακέτου αυτού, ο κινητός κόμβος θα στείλει στον CN ένα Binding Update με την CoA του.

Διαχείριση Binding Μηνυμάτων

Ένας κινητός κόμβος ο οποίος έχει κάνει configure ένα νέο CoA σαν την primary CoA του, πρέπει να εγγράψει αυτή τη διεύθυνση στον HA του και στους CN που έχουν ήδη ενημερωθεί για το binding του κινητού κόμβου. Για αυτό τον σκοπό ο κινητός κόμβος στέλνει ένα Binding Update που περιέχει τη νέα του διεύθυνση. Για επιβεβαίωση ότι ο παραλήπτης πράγματι παρέλαβε το Binding Update, ο κινητός κόμβος μπορεί να ζητήσει αναγνώριση ενεργοποιώντας το Acknowledgment bit στο Binding Update (μέχρι την παραλαβή της επιβεβαίωσης, ο κινητός κόμβος θα στέλνει περιοδικά το Binding Update). Ένας κινητός κόμβος πρέπει να θέτει σαν ενεργοποιημένο το Acknowledgment bit σε Binding Updates που προορίζονται προς τον HA του. Μπορεί ακόμη να ενεργοποιήσει το Acknowledgment bit όταν στέλνει σε CNs αλλά δεν είναι αναγκαίο γιατί αν το Binding Update δεν παραληφθεί για οποιοδήποτε λόγο από τον CN, ο κινητός κόμβος το καταλαβαίνει όταν εξακολουθεί να λαμβάνει πακέτα από CNs, μέσω του HA του και όχι απευθείας από τον CN.

Πριν τη λήξη της διάρκειας ζωής μιας καταχώρησης στην Binding Cache για ένα κινητό κόμβο, ο CN μπορεί να ανανεώσει την καταχώρηση στέλνοντας ένα Binding Request στον κινητό κόμβο. Κατά συνέπεια, ο κινητός κόμβος θα απαντήσει με ένα Binding Update.

Ανίχνευση Κίνησης

Κατά την διάρκεια που ένας κινητός κόμβος είναι μακριά από το home υποδίκτυό του, επιλέγει ένα δρομολογητή σαν τον default δρομολογητή και ένα πρόθεμα που διαφημίζεται από αυτόν τον δρομολογητή για να το χρησιμοποιήσει στην primary CoA του.

Στην συνέχεια, ο κινητός κόμβος μπορεί να χρησιμοποιήσει οποιοδήποτε συνδυασμό από διαθέσιμους μηχανισμούς για να ανιχνεύσει πότε έχει κινηθεί από μια σύνδεση σε μια άλλη. Μια δυνατότητα είναι ο κόμβος να περιμένει για τα Router Advertisements που στέλνονται περιοδικά. Αν δεν παραλάβει διαφήμιση για ένα συγκεκριμένο χρονικό διάστημα θα υποθέσει ότι ο default δρομολογητής δεν είναι πλέον διαθέσιμος και συνδέεται με άλλον δρομολογητή από τον οποίο είχε λάβει διαφήμιση αυτό το διάστημα.

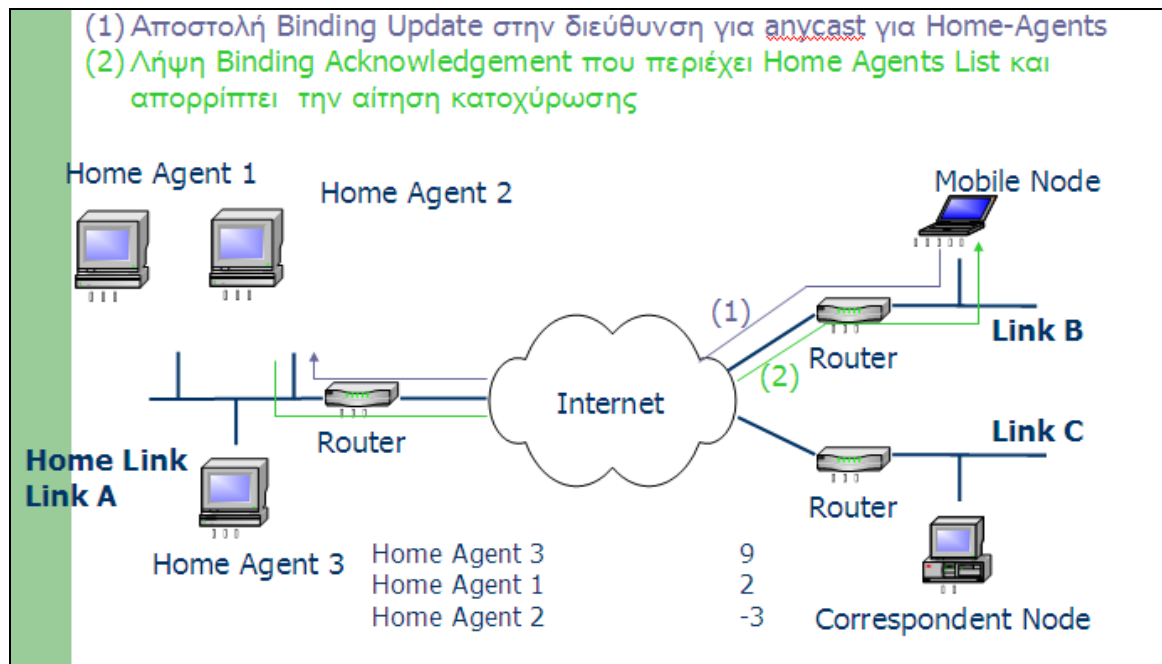
Όταν ο κινητός κόμβος καταλάβει ότι έχει κινηθεί σε άλλη σύνδεση, στέλνει ένα Binding Update στον HA του και στους CN τους οποίους έχει καταχωρημένους στην Binding Λίστα Ενημέρωσής του. Έτσι ο κινητός κόμβος τους ενημερώνει για την νέα CoA του και κατά συνέπεια για την μετακίνησή του.

Μηχανισμός Εξεύρεσης Home Agent

Υποθέτοντας ότι ο κινητός κόμβος δε γνωρίζει το IP του HA του, το πρωτόκολλο Mobile IPv6 προσφέρει ένα μηχανισμό που επιτρέπει στον κινητό κόμβο να ανακαλύπτει δυναμικά την IP διεύθυνση κάποιου HA στο HL του, στον οποίο θα μπορεί να εγγράψει την CoA του όταν θα είναι μακριά από το home subnet του.

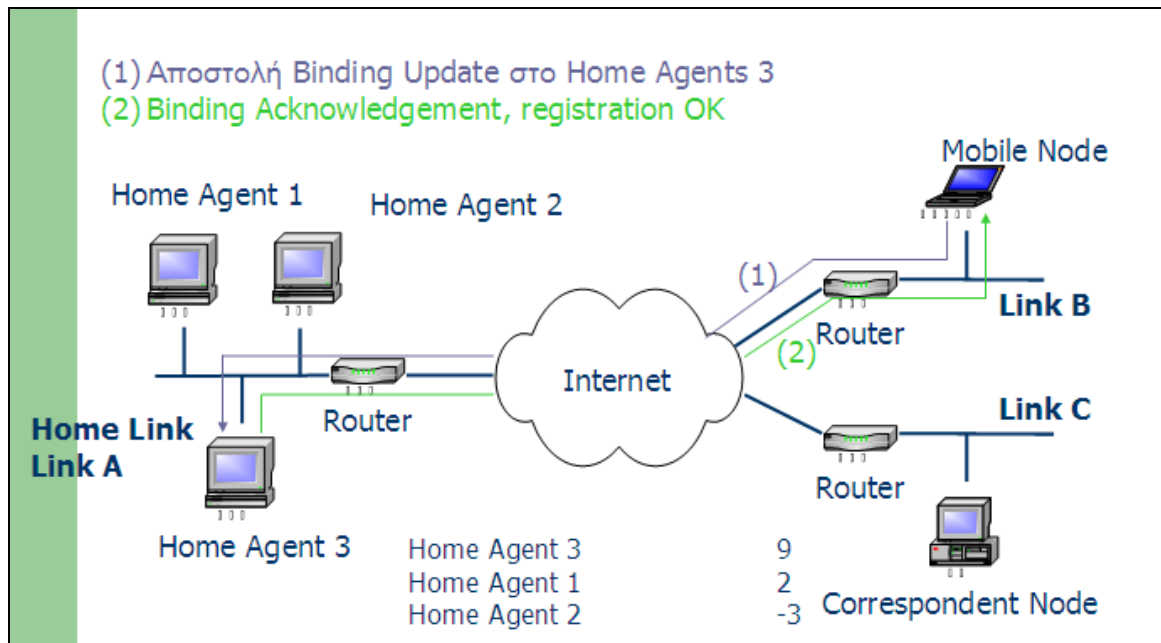
Ο κινητός κόμβος στέλνει ένα Binding Update στο “Home Agents’ anycast address” για το δικό του home subnet πρόθεμα και κατά συνέπεια καταφέρνει να επικοινωνήσει με ένα από τους δρομολογητές στο HL του, που εκείνη τη στιγμή λειτουργεί σαν HA. Αν ο HA απορρίψει το Binding Update, θα επιστρέψει μια λίστα με όλους τους HA στο HL. Αυτή η λίστα διατηρείται από κάθε HA και δημιουργείται μέσω των περιοδικών αποστολών Routing Advertisements. Ο κινητός κόμβος στέλνει ένα Binding Update σε μια από τις διευθύνσεις στην λίστα και περιμένει για το ανάλογο Binding Acknowledgement. Αν δεν το λάβει, ή αν απορριφθεί, δοκιμάζει να εγγραφεί σε ένα άλλο HA της λίστας. Η σειρά

επιλογής των HAs στην λίστα, γίνεται με την σειρά που καταγράφονται σε αυτή, αφού η πρώτη διεύθυνση είναι του πιο διαθέσιμου HA και η τελευταία του λιγότερου διαθέσιμου.



Σχήμα 2-4: Επιλογή Home Agent από την λίστα των HAs στο HL

Με βάση το σενάριο που περιγράφεται από την πιο πάνω εικόνα, ο κινητός κόμβος τώρα θα στείλει το Binding Update του στον Home Agent 3 γιατί έχει απορριφθεί το Binding Update που είχε στείλει στον Home Agent 2. Ο Home Agent 2 του απαντά, αποστέλλοντας μια λίστα με διευθύνσεις, από την οποία ο Home Agent 3 καλείται να επιλέξει μια από αυτές. Η επιλογή γίνεται με την σειρά που οι πιθανοί Has καταγράφονται στην λίστα, γιατί η πρώτη διεύθυνση είναι του πιο διαθέσιμου HA και η τελευταία του λιγότερου διαθέσιμου.



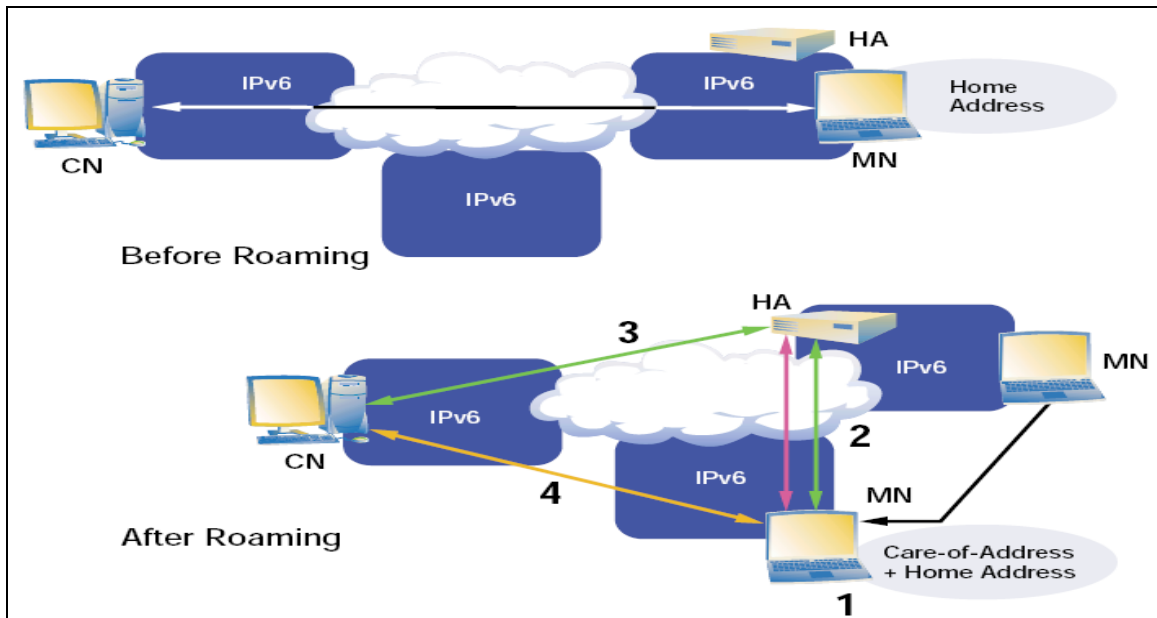
Σχήμα2-5: Επιτυχής Ολοκλήρωση Εγγραφής στον Home Agent 3

2.5 MIPv6 Roaming between IPv6 Networks

Καθώς μία κινητή συσκευή μετακινείται ανάμεσα σε διαφορετικά IPv6 δίκτυα, ακολουθείται πάντα η ίδια διαδικασία που περιγράφεται στη συνέχεια.

1. Καθώς η κινητή συσκευή εισέρχεται σε ένα IPv6 foreign network, λαμβάνει την νέα CoA του με την χρήση μηχανισμού autoconfiguration. Αυτό μπορεί να προκύψει είτε από router advertisements, είτε από εξυπηρετητή DHCP στο δίκτυο.
2. Ακολούθως ενημερώνει τον Home Agent της στέλλοντας πληροφορίες σχετικά με την νέα CoA της μέσω binding update και ο home agent απαντά στο μήνυμα αυτό, αποστέλλοντας στον κινητό κόμβο επιλογή προορισμού binding acknowledgment.
3. Από αυτό το σημείο και έπειτα ο home agent παρεμβάλλεται στην δρομολόγηση των πακέτων προς την home address της συσκευής και τα επαναδρομολογεί προς αυτή, κάνοντας χρήση της καταχωρημένης CoA της.
4. Το MIPv6 επιτρέπει route optimization μέσω της απευθείας επικοινωνίας μεταξύ δύο κόμβων. Για να συμβεί αυτό, θα πρέπει ο κινητός κόμβος να αποστείλει προηγουμένως

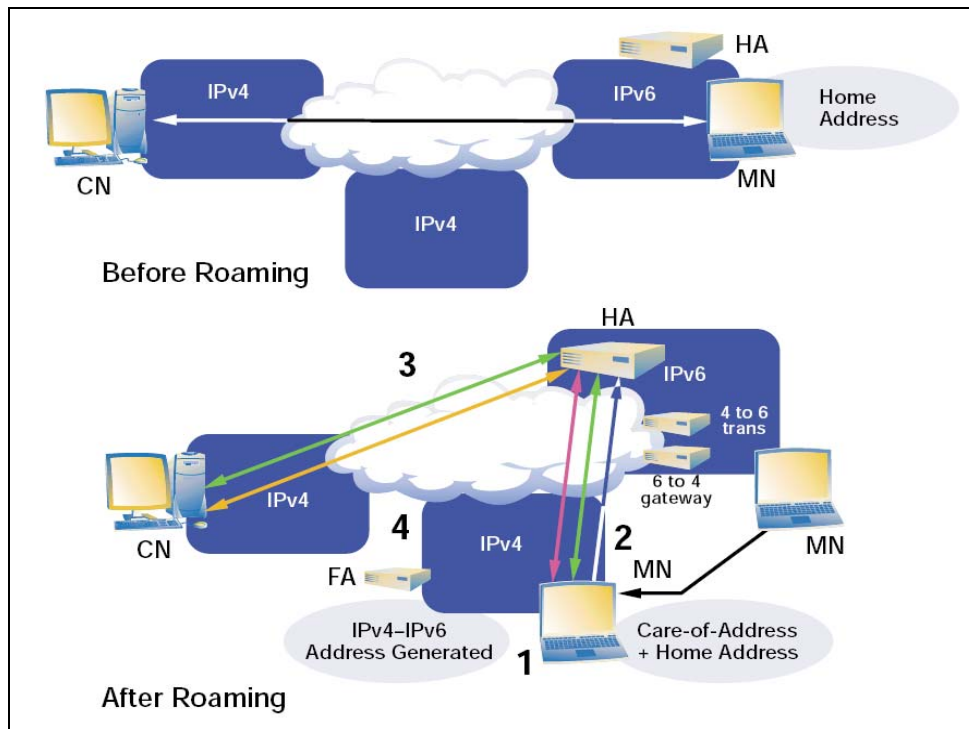
μήνυμα binding update προς οποιοδήποτε CN, ενημερώνοντάς τον με την τρέχουσα CoA του. Ο CN τότε μπορεί να προωθεί πακέτα απευθείας προς τον κινητό κόμβο.



Σχήμα 2-6: Mobile Node before and after roaming to IPv6 Network

2.6 MIPv6 Roaming between IPv4 Networks

1. Καθώς η κινητή συσκευή κινείται σε ένα foreign network, λαμβάνει μια νέα IPv4 IP διεύθυνση από ένα εξυπηρετητή DHCP στο νέο δίκτυο. Ο κινητός κόμβος τότε θα κάνει χρήση της διεύθυνσης αυτής ώστε να παράξει μία IPv4 mapped IPv6 CoA.
2. Χρησιμοποιώντας IPv4 tunneling, η συσκευή ενημερώνει τον home agent της για την νέα CoA της αποστέλλοντας binding update και ο HA απαντά με το κατάλληλο binding acknowledgement.
3. Από αυτό το σημείο και έπειτα ο home agent παρεμβάλλεται στην δρομολόγηση των πακέτων προς την home address της συσκευής και τα επαναδρομολογεί προς αυτή, κάνοντας χρήση της καταχωρημένης CoA της.
4. Για να επικοινωνήσει με μια κινητή συσκευή, ο IPv4 CN μεταφέρει τα πακέτα μέσω του home agent.



Σχήμα 2-7: Mobile Node before and after roaming to IPv4 Network

ΚΕΦΑΛΑΙΟ 3

ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΙΝΗΤΩΝ ΚΑΙ ΑΣΥΡΜΑΤΩΝ ΔΙΚΤΥΩΝ

3.1 Κυψελοειδή Δίκτυα Δεύτερης Γενιάς 2g

3.2 Κυψελοειδή Δίκτυα 2.5G (GPRS)

3.3 Κινητά Δίκτυα 3G

3.4 Ασύρματα Τοπικά Δίκτυα (WLANs) 802.11

3.1 Κυψελοειδή Δίκτυα Δεύτερης Γενιάς 2G

Κύριος στόχος του σχεδιασμού των συστημάτων δεύτερης γενιάς ήταν η μεγιστοποίηση της χωρητικότητας του συστήματος, που υπολογίζονταν σαν ο αριθμός των χρηστών σε κάθε περιοχή. Αυτό επιτεύχθηκε με δύο τρόπους, με την σμίκρυνση του μεγέθους της κυψέλης, επιτρέποντας περισσότερες κυψέλες ανά περιοχή, και με την καλύτερη συμπίεση της φωνής σε σχέση με αναλογικές κωδικοποιήσεις, πράγμα που επέτρεψε περισσότερες κλήσεις χρησιμοποιώντας το ίδιο bandwidth. Η τεχνολογία αυτή κάνει συχνή χρήση της ψηφιακής τεχνολογίας χρησιμοποιώντας ψηφιακούς vocoders, Forward Error Correction (FEC), ψηλού επιπέδου διαμόρφωση που αυξάνει την ποιότητα φωνής, ασφάλεια και αξιοπιστία κλήσης. Η τεχνολογία GSM αποτελεί μια πολύ σταθερή, διαδεδομένη και πιθανότατα την πιο δημοφιλή τεχνολογία για κινητή επικοινωνία. Το σημαντικότερο μειονέκτημα της τεχνολογίας αυτής είναι οι χαμηλές ταχύτητες μετάδοσης δεδομένων που υποστηρίζει, πράγμα που δεν ευνοεί τις εφαρμογές video. Υποστηρίζει μόνο υπηρεσίες χαμηλών ρυθμών μετάδοσης, μέχρι 9.6 Kbps, και την υπηρεσία Short Messages Services (SMS). Η υπηρεσία αυτή ήταν η πρώτη υποστηριζόμενη υπηρεσία δεδομένων για κινητές συσκευές.

3.2 Κυψελοειδή Δίκτυα 2.5G (GPRS)

Πρόκειται για μία ενδιάμεση γενιά ανάμεσα στις τεχνολογίες δικτύων 2G και 3G. Ουσιαστικά αναφέρεται σε 2G συστήματα που έχουν υλοποιήσει και μια packet switched περιοχή πέραν της circuit switched περιοχής.

Το General Packet Radio Service είναι μία τεχνολογία που έχει αναπτυχθεί από το Ινστιτούτο Ευρωπαϊκών Τηλεπικοινωνιακών Προτύπων. Τα GPRS, έχουν σχεδιαστεί για να παρέχουν ψηλό ρυθμό μετάδοσης δεδομένων στην υπηρεσία του δικτύου GSM. Πλεονεκτήματα από την εφαρμογή τους είναι η χρήση εφαρμογών video, ηλεκτρονικού ταχυδρομείου και άλλων. Ο ρυθμός μετάδοσης δεδομένων φτάνει τα 384Kbps. Επίσης, το GPRS μπορεί να χρησιμοποιηθεί για υπηρεσίες όπως Wireless Application Protocol (WAP), Short Message Service (SMS), Multimedia Messaging Service (MMS) και για επικοινωνία μέσω Διαδικτύου όπως e-mail και web browsing.

Η μεταφορά δεδομένων στο GPRS χρεώνεται με βάση τα δεδομένα που μεταφέρει ο χρήστης, ενώ η παραδοσιακή σύνδεση μέσω circuit switching χρεώνεται με βάση τον χρόνο σύνδεσης. Αντίθετα, με το circuit switching το GPRS είναι best- effort packet switched υπηρεσία όπου ένα καθορισμένο Quality of Service είναι εγγυημένο κατά τη διάρκεια της σύνδεσης.

3.3 Κινητά Δίκτυα 3G

Τα κινητά δίκτυα 3G επιτρέπουν την παροχή πιο εξελιγμένων υπηρεσιών, όπως video telephony, streaming video multimedia, broadband wireless data και web browsing με υψηλές ταχύτητες, σε ασύρματο περιβάλλον.

Σε αντίθεση με τα WLAN δίκτυα, που θα αναφερθούν στη συνέχεια, τα δίκτυα 3G είναι ευρείας περιοχής κυψελοειδή τηλεφωνικά δίκτυα, που αναπτύχθηκαν με κύριο στόχο να υποστηρίξουν υψηλών ταχυτήτων πρόσβαση στο Διαδίκτυο και video telephony. Από την άλλη τα WLANs είναι δίκτυα περιορισμένης εμβέλειας, με ψηλό bandwidth κυρίως ανεπτυγμένα για υπηρεσίες δεδομένων.

Ο ρυθμός μετάδοσης δεδομένων στα δίκτυα 3G μπορεί να φτάσει υπό προϋποθέσεις μέχρι και τα 2Mbps.

3.4 Ασύρματα Τοπικά Δίκτυα (WLANs) IEEE 802.11

Βασίζονται στα πρότυπα της IEEE και τελευταία έχουν γνωρίσει εξαιρετικής διάδοσης στον τομέα της ασύρματης σύνδεσης στο διαδίκτυο.

Χωρίζεται σε τρεις περαιτέρω τεχνολογίες 802.11a, 802.11b και 802.11g. Το 802.11b λειτουργεί στις συχνότητες 2.4 GHz και θεωρητικά ο ρυθμός μετάδοσης φτάνει μέχρι και τα 11Mbps. Παρόλα αυτά λειτουργεί και στα 1, 2 και 5 Mbps. Όσο αφορά το 802.11a, αυτό λειτουργεί στα 5GHz και το 802.11g στα 2.4GHz. Θεωρητικά, ο ρυθμός μετάδοσής τους φτάνει μέχρι τα 54Mbps, εντούτοις λειτουργούν και σε χαμηλότερα όρια όπως 6, 10, 12, 18, 36 και 48 Mbps.

ΚΕΦΑΛΑΙΟ 4

ΘΕΜΕΛΕΙΩΔΕΙΣ ΑΡΧΕΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΨΗΦΙΑΚΟΥ ΒΙΝΤΕΟ

4.1 Εισαγωγή

4.2 Θεμελιώδεις Αρχές Μείωσης Πλεονασμού

4.3 Φάσεις της Τεχνικής Συμπίεσης

4.4 Είδη Πλαισίων κατά την Κωδικοποίηση Βίντεο

4.5 Τεχνικές Κωδικοποίησης

4.1 Εισαγωγή

Η στατιστική ανάλυση ενός σήματος βίντεο υποδεικνύει ότι υπάρχει σημαντική οπτική συσχέτιση τόσο διαπλαισιακή (μεταξύ διαδοχικών frames), όσο και ενδοπλαισιακή (της οπτικής πληροφορίας που απεικονίζεται εντός ενός frame). Η συσχέτιση αυτή μπορεί να χαρακτηριστεί ως πλεονάζουσα πληροφορία και μπορεί επιτυχώς να αξιοποιηθεί για τη συμπίεση του όγκου των δεδομένων του σήματος. Επιπλέον, γνωρίζοντας τους περιορισμούς του ανθρώπινου οπτικού συστήματος, το οποίο παρουσιάζει υψηλή ευαισθησία μόνο στις χαμηλές χωρικές συχνότητες, μπορούμε να επιτύχουμε περαιτέρω συμπίεση των δεδομένων παραλείποντας συγκεκριμένες συχνότητες της πλεονάζουσας πληροφορίας, η αφαίρεση των οποίων θα προκαλέσει μικρής κλίμακας ποιοτικό υποβιβασμό.

Βάση των πιο πάνω παρατηρήσεων, οι μέθοδοι κωδικοποίησης βίντεο διαιρούνται σε δύο κατηγορίες, τις απωλεστικές και τις μη απωλεστικές. Σε ένα χωρίς απώλειες σύστημα συμπίεσης, ο στατιστικός πλεονασμός αφαιρείται με τέτοιο τρόπο ώστε να είναι δυνατή η πλήρης ανοικοδόμηση του αρχικού σήματος από πλευράς του δέκτη. Μέχρι και σήμερα, η

χρήση τέτοιων μεθόδων δεν προσφέρει υψηλό λόγο συμπίεσης και ως εκ τούτου δεν έχουν τύχει ευρείας εφαρμογής. Από την άλλη, στις απωλεστικές μεθόδους επιτυγχάνεται υψηλός λόγος συμπίεσης των δεδομένων, με κόστος βέβαια ότι το αποκωδικοποιημένο σήμα έχει υποβαθμιστεί ποιοτικά σε σχέση με το αρχικό. Κύριος στόχος κάθε απωλεστικής τεχνικής είναι εξισορρόπηση του υψηλού λόγου συμπίεσης, ελαχιστοποιώντας την αντίστοιχη ποιοτική υποβάθμιση.

Έχοντας αναφέρει όλα τα πιο πάνω, συμπεραίνουμε ότι η βασική αρχή πάνω στην οποία στηρίζονται όλες οι μέθοδοι ψηφιακής συμπίεσης βίντεο είναι το γεγονός ότι το σήμα εμπεριέχει ένα σημαντικό ποσοστό πλεονασμού. Με τον όρο αυτό εννοούμε την πληροφορία που μπορεί να συμπειστεί ή εν μέρη να παραλειφθεί, χωρίς να προκαλέσει σημαντική ποιοτική υποβάθμιση στο τελικό βίντεο. Ακολούθως παρουσιάζονται τα δύο είδη πλεονασμού.

Χωρικός Πλεονασμός

Η τιμή ενός εικονοστοιχείου (pixel) που εμπεριέχεται σε ένα frame του σήματος, μπορεί σε κάθε δεδομένη χρονική στιγμή να εκτιμηθεί με μεγάλη ακρίβεια από τις τιμές γειτονικών εικονοστοιχείων, στηριζόμενοι στο γεγονός ότι οι γειτονικές τιμές δειγμάτων μιας εικόνας σχετίζονται μεταξύ τους σε μεγάλο ποσοστό.

Χρονικός Πλεονασμός

Ένα σήμα βίντεο δομείται από διαδοχικά πλαίσια, τα οποία απεικονίζουν την εξελισσόμενη δράση ανά τακτά χρονικά διαστήματα εντός του ενός δευτερολέπτου. Συνεπώς, τα διαδοχικά καρέ παρουσιάζουν μεγάλες ομοιότητες, δημιουργώντας έντονες συσχετίσεις μεταξύ τους, οι οποίες χαρακτηρίζονται ως χρονικός πλεονασμός.



Σχήμα 4-1: Χωρικός και Χρονικός Πλεονασμός

4.2 Θεμελιώδεις Αρχές Μείωσης Πλεονασμού

Η πλειοψηφία των προτύπων κωδικοποίησης και συμπίεσης ψηφιακού βίντεο στηρίζονται στην αξιοποίηση του χρονικού και χωρικού πλεονασμού στοχεύοντας στη μείωσή του, γεγονός που θα έχει σαν αποτέλεσμα την συμπίεση του όγκου των δεδομένων. Οι τρεις θεμελιώδεις αρχές μείωσης του πλεονασμού κατά την κωδικοποίηση είναι οι πιο κάτω

4.2.1 Μείωση Χωρικού Πλεονασμού

Για να επιτευχθεί μείωση του πλεονασμού αυτού, μεταξύ των εικονοστοιχείων που δομούν ένα πλαίσιο, αξιοποιείται η ομοιότητα των τιμών των γειτονικών εικονοστοιχείων με τη χρήση κατάλληλων μετασχηματισμών.

4.2.2 Μείωση Χρονικού Πλεονασμού

Για την επίτευξη του χρονικού πλεονασμού μεταξύ διαδοχικών εικόνων ενός βίντεο, χρησιμοποιούνται εικόνες αναφοράς και αξιοποιείται η κωδικοποίηση μόνο των διαφορών μεταξύ των διαδοχικών εικόνων. Συνεπώς είναι εφικτό να προβλέψουμε ένα σημαντικό

τμήμα της εικόνας και να προσθέσουμε απλά τις επιμέρους διαφορές, χωρίς να χρειάζεται να κωδικοποιείται σε κάθε εικόνα ολόκληρη η πλεονάζουσα πληροφορία.

4.2.3 Εντροπική Κωδικοποίηση

Για να επιτευχθεί περαιτέρω συμπίεση των δεδομένων, πραγματοποιείται εντροπική συμπίεση με την εφαρμογή κωδικοποίησης μεταβλητού μήκους των συμβόλων της ροής δεδομένων.

4.3 Φάσεις της Τεχνικής Συμπίεσης

Σχετικά με την αξιοποίηση του χρονικού και χωρικού πλεονασμού, όλα τα πρότυπα κωδικοποίησης αξιοποιούν δύο διακριτές φάσεις στην τεχνική συμπίεσης, την ενδοπλαισιακή και την διαπλαισιακή.

4.3.1 Ενδοπλαισιακή Κωδικοποίηση (Intra-frame Coding)

Σε αυτό το είδος κωδικοποίησης, για την μείωση του χωρικού πλεονασμού, κάθε πλαίσιο αντιμετωπίζεται σαν αυτόνομη μονάδα και κωδικοποιείται ανεξάρτητα από τα υπόλοιπα πλαίσια αξιοποιώντας τον εσωτερικό του χωρικό πλεονασμό, οπότε το τελικό σήμα είναι μια σειρά από διακριτές ακίνητες εικόνες.

4.3.2 Διαπλαισιακή Κωδικοποίηση (Inter- frame Coding)

Στη Διαπλαισιακή Κωδικοποίηση για να επιτευχθεί μείωση του χρονικού πλεονασμού, λαμβάνονται υπόψη οι ομοιότητες των διαδοχικών πλαισίων (motion estimation) και κωδικοποιείται η διαφορά τους με χρήση της τεχνικής αντιστάθμισης της κίνησης (motion compensation). Έτσι, στο τελικό σήμα υπάρχει μία αλληλεξάρτηση μεταξύ των πλαισίων, αφού για να αποκωδικοποιηθεί και να ανασυγκροτηθεί κάποιο πλαίσιο, πιθανώς να πρέπει να ληφθούν πληροφορίες και από τα γειτονικά του (προηγούμενα ή επόμενα. Με αυτό τον τρόπο, επιτυγχάνεται υψηλός λόγος συμπίεσης, αφού κωδικοποιούνται μόνο οι διαφορές μεταξύ των διαδοχικών πλαισίων.

4.4 Είδη Πλαισίων κατά την Κωδικοποίηση Βίντεο

Η αξιοποίηση του χρονικού πλεονασμού κατά τη διαδικασία κωδικοποίησης στοχεύει στη μείωση της πλεονάζουσας πληροφορίας μεταξύ των διαδοχικών πλαισίων με την ανάπτυξη διαπλαισιακών συσχετίσεων. Για αυτό τον λόγο σε μια ροή δεδομένων κωδικοποιημένου βίντεο σύμφωνα με το πρότυπο MPEG, τα πλαίσια διακρίνονται στις ακόλουθες τρεις κατηγορίες.

4.4.1 I- Frames (Intra- frames)

Αυτό το είδος των πλαισίων κάνει χρήση του intra frame coding(ενδοπλαισιακής κωδικοποίησης). Η αποκωδικοποίηση τέτοιου είδους πλαισίου δεν απαιτεί την λήψη δεδομένων από κάποιο πλαίσιο αναφοράς, αφού πρόκειται για τα μόνα των οποίων η κωδικοποίηση είναι ασυσχέτιστη με τα γειτονικά πλαίσια και συνεπώς η αποκωδικοποίηση τους μπορεί να γίνει χωρίς αναφορά σε κάποιο άλλο. Είναι τα πλαίσια με το μεγαλύτερο μέγεθος σε μια ροή MPEG και αποτελούν σημεία αναφοράς για την κωδικοποίηση των μελλοντικών πλαισίων ώστε να επιτευχθεί η μείωση του χρονικού πλεονασμού. Λόγω της συσχέτισης αυτής, τα πλαίσια I επαναλαμβάνονται περιοδικά σε ένα ρεύμα ροής MPEG για να αποφευχθεί η διάδοση πιθανών σφαλμάτων που δημιουργεί η συσχετισμένη αποκωδικοποίηση των υπόλοιπων πλαισίων από I πλαίσια. Η συμπίεση βίντεο με την χρήση μόνο αυτού του τύπου εικόνων είναι πολύ γρήγορη, όμως δεν προσφέρει ιδιαίτερο λόγο συμπίεσης μεταξύ αρχικού και κωδικοποιημένου βίντεο. Το σύνολο των πλαισίων μεταξύ δύο διαδοχικών I εικόνων ονομάζεται Group of Pictures (GoP). Το μήκος του συνόλου αυτού ορίζεται κατά την κωδικοποίηση και ουσιαστικά αναφέρεται στην περιοδικότητα εμφάνισης των πλαισίων I, καθώς και στην αντιστάθμιση μιας αποδοτικής κωδικοποίησης και μιας λιγότερο ανθεκτικής σε λάθη μετάδοσης ροής MPEG. Αναφέρεται ότι τυπικά πλαίσια για το MPEG είναι 15- 18 εικόνες. Ως εκ τούτου τα πλαίσια I αποτελούν τον θεμέλιο λίθο στη μείωση του χρονικού πλεονασμού κατά την κωδικοποίηση ενός ψηφιακού βίντεο.

4.4.2 P- Frames (Predicted- Frames)

Τα πλαίσια τύπου P είναι αυτά των οποίων η κωδικοποίηση και αποκωδικοποίηση είναι συσχετισμένη και εξαρτημένη από ένα προηγούμενο I ή P πλαίσιο (το οποίο αντίστοιχα εξαρτάται από ένα προηγούμενό του P ή I πλαίσιο). Κάνοντας χρήση της εκτίμησης κίνησης, ο μηχανισμός κωδικοποίησης προβλέπει σε ένα P πλαίσιο τις σχετικές θέσεις των macroblocks ή blocks σε σχέση με το P ή I πλαίσιο αναφοράς. Συνεπώς δεν υπάρχει λόγος κωδικοποίησης ολόκληρης της εικόνας, αλλά μόνο της πληροφορίας που τη συσχετίζει με την εικόνα αναφοράς και ονομάζεται διάνυσμα κίνησης (motion vector). Επίσης, κωδικοποιείται και η διαφορά του εκτιμώμενου πλαισίου που δομείται με τις συσχετίσεις σε σχέση με το πραγματικό πλαίσιο, μέσω της τεχνικής αντιστάθμισης της κίνησης, ώστε κατά τη φάση της αποκωδικοποίησης να έχουμε πλήρη αναδόμηση του αρχικού πλαισίου. Τα πλαίσια P μπορούν με τη σειρά τους να αποτελέσουν πλαίσια αναφοράς για επόμενα πλαίσια, συμβάλλοντας με αυτό τον τρόπο είτε στην εισαγωγή είτε στη διάδοση σφαλμάτων εντός ενός GoP, εξαιτίας των δομημένων συσχετίσεων στις οποίες και βασίζεται η κωδικοποίησή τους.

Τα πλαίσια P είναι μικρότερα σε μέγεθος από τα I πλαίσια, γιατί δεν περιέχουν ολόκληρη την οπτική πληροφορία του κωδικοποιημένου πλαισίου, αξιοποιώντας τον χρονικό πλεονασμό.

Συνοψίζοντας, η διαδικασία της κωδικοποίησης ενός P πλαισίου έχει ως ακολούθως: Εντοπίζονται οι συσχετίσεις macroblocks/blocks στο πλαίσιο αναφοράς (που μπορεί να είναι το προηγούμενο P ή I, αναλόγως της θέσης του κωδικοποιημένου P εντός του GoP) και δημιουργία ενός συνδυασμού αυτών μέσω κατάλληλων διανυσμάτων κίνησης. Στη συνέχεια, πραγματοποιείται η αντιστάθμιση κίνησης με υπολογισμό και κωδικοποίηση των αντίστοιχων διαφορών μεταξύ του πραγματικού και του εκτιμώμενου πλαισίου. Συνεπώς, ένα πλαίσιο P δομείται από τα διανύσματα κίνησης, τα Intra- coded blocks που δεν συσχετίστηκαν με κάποιο προγενέστερο πλαίσιο P ή B και τη διαφορά του εκτιμώμενου πλαισίου από το πραγματικό.

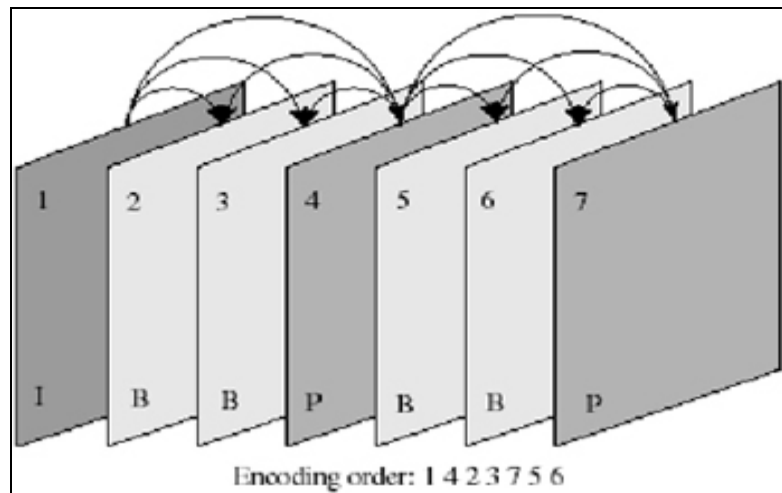
4.4.3 B- Frames (Bidirectional- Predicted Frames)

Τα πλαίσια τύπου B είναι πλαίσια που δημιουργούνται συσχετίζοντας σε επίπεδο block/macroblock το περιεχόμενο ενός προηγούμενου και ενός επόμενου πλαισίου P. Τα πλαίσια B δεν συντελούν στη διάδοση σφαλμάτων σε ένα GoP, καθώς δεν χρησιμοποιούνται ως εικόνες αναφοράς για επιπλέον μείωση του χρονικού πλεονασμού και η εκτίμησή τους περιορίζεται μόνο μεταξύ δυο διαδοχικών πλαισίων P. Συνεπώς, ο κύκλος ζωής των πλαισίων B περιορίζεται μόνο τοπικά και δεν επεκτείνεται μεταβιβάζοντας πληροφορίες σε άλλα πλαίσια. Η διαδικασία δόμησης τους είναι παρόμοια με αυτή των πλαισίων P, με βασική διαφορά ότι το εκτιμώμενο πλαίσιο B κατασκευάζεται λαμβάνοντας ως πλαίσια αναφοράς, τόσο το προηγούμενο, όσο και το επόμενο P πλαίσιο. Σε ιδιαίτερες περιπτώσεις, αν για παράδειγμα το B πλαίσιο βρίσκεται τελευταίο σε ένα GoP, είναι πιθανό για την εκτίμησή του να χρησιμοποιηθούν το προηγούμενο P πλαίσιο και το επόμενο πλαίσιο I. Επομένως, είναι αναγκαίο να αποκωδικοποιηθεί το επόμενο I-frame ή P- frame που ακολουθεί ένα B- frame, ώστε να είναι δυνατή η αποκωδικοποίηση και προβολή του B-frame. Αυτό καθιστά τα B- frames πολύπλοκα υπολογιστικά, αφού απαιτούν μεγάλα data buffers, και προκαλούν αυξημένη καθυστέρηση τόσο κατά την κωδικοποίηση όσο και κατά την αποκωδικοποίησή τους.

Πρόκειται για τα μικρότερα σε μέγεθος πλαίσια, λόγω της δια-κατευθυντικής εκτίμησης, και εκμεταλλεύονται στον μέγιστο βαθμό τον χρονικό πλεονασμό μεταξύ δύο P πλαισίων.

Αξίζει να σημειωθεί ότι η σειρά με την οποία κωδικοποιούνται οι εικόνες δεν είναι και η ίδια με την σειρά με την οποία εμφανίζονται. Συγκεκριμένα, για την εκτίμηση ενός B πλαισίου είναι αναγκαία η εκ των προτέρων κωδικοποίηση των αμέσως προηγούμενου και επόμενου P πλαισίου στα οποία αυτό αναφέρεται. Κατά την αναπαραγωγή, όμως, του βίντεο, το πλαίσιο B θα προηγείται του δεύτερου από τα δύο P frames που χρησιμοποιήθηκαν.

Η ακόλουθη εικόνα παρουσιάζει ένα τυπικό GoP και τις συσχετίσεις που δημιουργούνται μεταξύ των διαφόρων τύπων πλαισίων, ώστε να μειωθεί ο χρονικός πλεονασμός. Επίσης, παρουσιάζεται και η σειρά κωδικοποίησης των πλαισίων, που προφανώς, λοιπώς αναφέρθηκε πιο πάνω, διαφέρει από την σειρά αναπαραγωγής.



Σχήμα 4-2: Συσχετίσεις μεταξύ εικόνων σε ένα GoP και σειρά κωδικοποίησης

4.5 Τεχνικές Κωδικοποίησης

Για να επιτευχθεί η μείωση του χρονικού πλεονασμού μέσω των τριών ειδών πλαισίων που αναφέρθηκαν, χρησιμοποιούνται δύο τεχνικές κωδικοποίησης, η εκτίμηση κίνησης και η αντιστάθμιση κίνησης. Και οι δύο τεχνικές εφαρμόζονται σε επίπεδο macroblock ή block. Σε πιο σύγχρονα πρότυπα, όπως το H.264, χρησιμοποιούνται blocks μεταβλητού μεγέθους, το οποίο εξαρτάται από το περιεχόμενο, ώστε να επιτευχθεί η βέλτιστη απόδοση αυτών των τεχνικών.

4.5.1 Εκτίμηση Κίνησης

Κατά την εκτίμηση κίνησης, ο αλγόριθμος κωδικοποίησης αναζητεί στο πλαίσιο αναφοράς(ή στα 2 πλαίσια αναφοράς στην περίπτωση πλαισίου B), που μπορεί να είναι προγενέστερα ή μεταγενέστερα(ανάλογα και πάλι με το πλαίσιο που κωδικοποιείται), τη βέλτιστη περιοχή macroblock που ταιριάζει με το περιεχόμενο που κωδικοποιείται και που θα χρησιμοποιηθεί ως αναφορά για τη δόμησή του. Αυτό πραγματοποιείται μέσω της σύγκρισης συγκεκριμένων τετράγωνων(όχι απαραίτητα για το H.264) περιοχών μεταξύ του πλαισίου αναφοράς και του πλαισίου που κωδικοποιείται μέχρι να βρεθεί η βέλτιστη ταύτιση.

Ο αριθμός των αλγορίθμων που έχουν προταθεί για τη διαδικασία αυτή είναι πολύ μεγάλος. Όλοι είχαν σαν στόχο την αύξηση της ταχύτητας εκτέλεσης και ταυτόχρονα τη μείωση της πολυπλοκότητας. Σημειώνεται ότι η πλειοψηφία των αλγορίθμων εκτίμησης κίνησης, χρησιμοποιούν σαν κύρια μετρική για την εύρεση των ποσοτικών διαφορών μεταξύ διαδοχικών εικόνων το Άθροισμα της Απόλυτης Διαφοράς.

Για να καθοριστεί η περιοχή αναφοράς, γίνεται χρήση των διανυσμάτων κίνησης, μέσω των οποίων μπορεί να εξακριβωθεί επακριβώς η βέλτιστη περιοχή συσχέτισης στο πλαίσιο αναφοράς.

4.5.2 Αντιστάθμιση Κίνησης

Έχοντας εντοπίσει την περιοχή με τη βέλτιστη ταύτιση μέσω του αλγορίθμου εκτίμησης της κίνησης, τότε ακολουθεί η διαδικασία αντιστάθμισης κίνησης. Κατά τη διάρκεια αυτού του σταδίου, το πλαίσιο που βρίσκεται υπό κωδικοποίηση αναδομείται βάσει με τις συσχετίσεις που προέκυψαν από την εκτίμηση κίνησης, χρησιμοποιώντας διανύσματα κίνησης και το πλαίσιο αναφοράς. Με αυτό τον τρόπο δημιουργείται ένα πλαίσιο, που είναι αποτέλεσμα εκτίμησης, και το οποίο ακολούθως αφαιρείται από το αντίστοιχο πραγματικό πλαίσιο που αποφέρει σαν έξοδο ένα πλαίσιο διαφορών, το οποίο απεικονίζει την εντροπική τους διαφορά. Το παραγόμενο πλαίσιο διαφορών αποτελεί, ουσιαστικά, την απεικόνιση των διαφορών μεταξύ εκτιμώμενου και πραγματικού πλαισίου.



Σχήμα 4-3: Παράδειγμα πλαισίου εντροπικών διαφορών

Στην πιο πάνω εικόνα στο κέντρο, οι στατικές περιοχές απεικονίζουν περιοχές χωρίς διαφορές, οι σκοτεινές περιοχές αρνητικές διαφορές και οι φωτεινές περιοχές υποδηλώνουν θετικές διαφορές.

Μελετώντας τα πιο πάνω, συμπεραίνουμε ότι η αντιστάθμιση κίνησης χρησιμοποιείται συμπληρωματικά του μηχανισμού εκτίμησης κίνησης, διορθώνοντας και αντισταθμίζοντας, στο πλαίσιο που προκύπτει από την εκτίμηση, τις διαφορές που υπάρχουν σε σχέση με το πραγματικό πλαίσιο. Οι διαφορές αυτές αφού υπολογιστούν, ενσωματώνονται στη ροή και χρησιμοποιούνται κανονικά σε όλα τα επόμενα στάδια της κωδικοποίησης.

ΚΕΦΑΛΑΙΟ 5

VIDEO STREAMING ΚΑΙ ΠΡΟΤΥΠΑ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΨΗΦΙΑΚΟΥ ΒΙΝΤΕΟ

5.1 Video Streaming

5.2 Τρόποι Ανάκτησης Δεδομένων μετά από Απώλεια Πακέτου

5.3 Πρότυπα Συμπίεσης Ψηφιακού Βίντεο

5.4 Ομάδες Εργασίας στην Ανάπτυξη Προτύπων Ψηφιακού Βίντεο

5.5 Πρότυπα Κωδικοποίησης

5.1 Video Streaming

Τα τελευταία χρόνια οι εφαρμογές streaming ήχου και εικόνας, έχουν τύχει ευρείας διάδοσης η οποία τις καθιστά και μεταξύ των δημοφιλέστερων εφαρμογών. Επιπλέον αυτό έχει σαν αποτέλεσμα να είναι ένας σημαντικός καταναλωτής του διαθέσιμου εύρους ζώνης του Διαδικτύου. Όπως είναι λογικό, η ανάπτυξη αυτή έλαβε χώρα ταυτόχρονα με την ραγδαία αναβάθμιση της συμπίεσης δεδομένων και των δικτύων υψηλών ταχυτήτων, χαρακτηριστικά με τα οποία συνδέεται άμεσα η έννοια streaming πολυμέσων. Πέρα από τα πιο πάνω, έρχεται να προστεθεί και η διάδοση των φορητών συσκευών όπως φορητών υπολογιστών, υπολογιστών τσέπης (PDAs) και κινητών τηλεφώνων με δυνατότητα πρόσβασης στο Διαδίκτυο. Η ανάπτυξη της τεχνολογίας δικτύων σήμερα παρέχει την δυνατότητα για πρόσβαση υψηλών ταχυτήτων σε αυτές τις συσκευές μέσω ασύρματων δικτύων της οικογένειας WLAN και τεχνολογίας 3G.

Τα αρχεία πολυμέσων που πρόκειται να αποσταλούν σε κάποιο προορισμό στα πλαίσια του streaming είναι συνήθως αποθηκευμένα σε κάποιο εξυπηρετητή και η έναρξη της αποστολής γίνεται μετά από αίτηση κάποιου πελάτη προς αυτόν. Για τη μεταφορά είναι

δυνατή η χρήση τόσο του πρωτοκόλλου TCP, όσο και του UDP, εντούτοις πλέον είναι πιο διαδεδομένη η χρήση του πρώτου (αν και θεωρητικά συνιστάται η χρήση UDP για εφαρμογές πραγματικού χρόνου για τους λόγους που εξηγούνται σε κεφάλαιο που ακολουθεί) λόγω του ότι τα firewalls είναι ρυθμισμένα αρκετές φορές ώστε να αποκόπτουν ροές UDP και επιπλέον για να διασφαλίζεται η χωρίς σφάλματα και απώλειες λήψη του πολυμεσικού περιεχομένου ώστε να είναι δυνατή η αναπαραγωγή του σε μελλοντικό στάδιο από τοπική cache μνήμη. Τα αρχεία πολυμέσων είναι στην πλειοψηφία τους μεγάλα σε μέγεθος, πράγμα που καθιστά τη διάδοσή τους χρονικά μεγαλύτερη από την ανεκτή καθυστέρηση της αναπαραγωγής. Βάσει αυτού συμπεραίνουμε ότι το κατέβασμα ολόκληρου του αρχείου προτού έχουμε την δυνατότητα αναπαραγωγής του δεν αποτελεί αποδοτική λύση.

Έχοντας υπόψη το πιο πάνω γεγονός και το στοιχείο ότι τα streaming πολυμέσα αποτελούνται από πολλές ροές του video-audio, η αποδοτικότερη λύση είναι η αναπαραγωγή των ροών που έχουν ήδη παραληφθεί, ενώ παράλληλα γίνεται λήψη των υπόλοιπων ροών που μεταδίδονται. Με αυτό τον τρόπο είναι δυνατό οι χρήστες να μπορούν να ξεκινήσουν την αναπαραγωγή του video πριν τελειώσει εντελώς το κατέβασμά του.

Η ανάπτυξη όμως που αναφέραμε όσο αφορά τη διάδοση της εφαρμογής των streaming πολυμέσων, δημιουργεί την επιτακτική ανάγκη για διασφάλιση του επιπέδου της ποιότητας υπηρεσίας (QoS) που παρέχεται. Η φύση των εφαρμογών αυτών και το γεγονός ότι πρόκειται για εφαρμογές πραγματικού χρόνου, αποτελεί κύρια αιτία για τις υψηλές απαιτήσεις σε εύρος ζώνης που παρουσιάζουν. Επιπλέον, όμως, ιδιαιτερότητες παρουσιάζουν και όσο αφορά τις απώλειες πακέτων, την καθυστέρηση και την διακύμανση καθυστέρησης (jitter), στην οποία υπόκεινται τα πακέτα μίας διεργασίας. Το Jitter ορίζεται σαν η διαφορά ανάμεσα στις καθυστερήσεις που παρουσιάζουν διαδοχικά πακέτα που ανήκουν στην ίδια διεργασία.

Η εφαρμογή streaming πολυμέσων είναι ανεκτική στις απώλειες μερικώς πακέτων που ανήκουν στη συγκεκριμένη ροή. Ταυτόχρονα όμως, η ανεκτικότητα αυτή προκαλεί την αναγκαιότητα για ελαχιστοποίηση των λαθών που προκύπτουν για να υπάρχει η επιθυμητή απόδοση και αντιληπτή ποιότητα της εφαρμογής. Για τον λόγο αυτό χρησιμοποιείται ένας μηχανισμός ανίχνευσης λαθών ο οποίος επανακτά τα πακέτα στον πελάτη πριν την

προγραμματισμένη έναρξη της αναπαραγωγής του αρχείου. Μερικές τεχνικές εφαρμογής του πιο πάνω μηχανισμού είναι το Forward Error Correction (FEC), interleaving, και διαδοχικές επαναποστολές πακέτων.

5.2 Τρόποι Ανάκτησης Δεδομένων μετά από Απώλεια Πακέτου

5.2.1 Forward Error Correction (FEC)

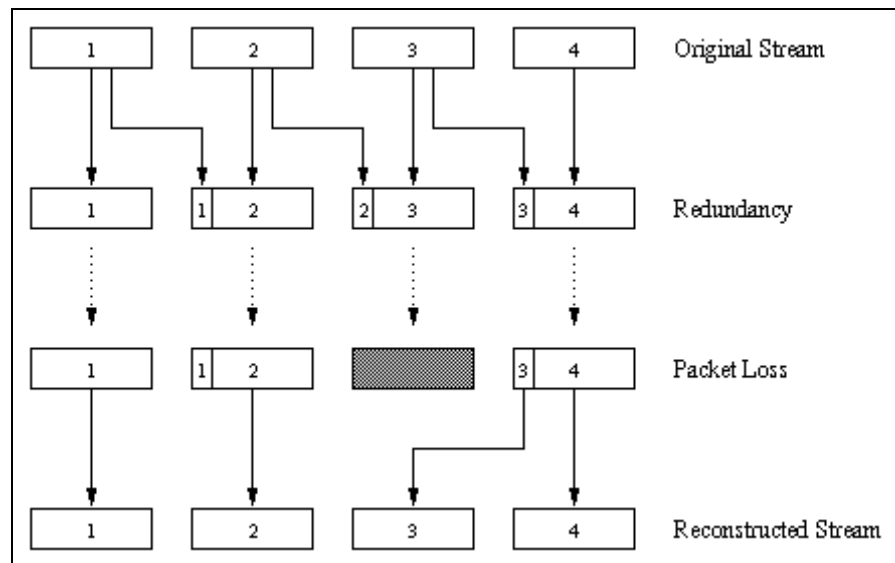
Πρόκειται για ένα μηχανισμό που βασίζεται στην αποστολή πλεονάζουσας πληροφορίας μαζί με την αρχική ροή δεδομένων. Αν και έχει σαν αποτέλεσμα την αύξηση του ρυθμού που απαιτείται για την αποστολή του αρχείου, εντούτοις, τα επιπλέον δεδομένα μπορούν να χρησιμοποιηθούν για την αναπαραγωγή σχετικών ή απολύτως όμοιων αντίγραφων των μερών που χάθηκαν σαν αποτέλεσμα απώλειας πακέτου.

Η πρώτη προσέγγιση απαιτεί όπως το επιπλέον κομμάτι δεδομένων προκύπτει μετά από την εφαρμογή αποκλειστικού OR (XOR) σε n κομμάτια της αρχικής ροής. Με αυτό τον τρόπο αν υπάρξει απώλεια οποιουδήποτε από τα $n+1$ πακέτα, τότε είναι δυνατή η πλήρης ανοικοδόμηση του χαμένου πακέτου. Σε περίπτωση απώλειας όμως δύο ή περισσότερων πακέτων δεν είναι δυνατή η ανάκτηση των χαμένων δεδομένων. Με αυτό τον τρόπο αντιλαμβανόμαστε ότι η επιλογή του μεγέθους n είναι αρκετά σημαντική αφού όσο μικρότερο το μέγεθος, τόσο μεγαλύτερη η δυνατότητα ανάκτησης των χαμένων πακέτων, αλλά από την άλλη ταυτόχρονα αυξάνεται και ο απαιτούμενος ρυθμός μετάδοσης της ροής, λόγω περισσότερων πλεονάζοντων δεδομένων. Συγκεκριμένα, ο λόγος αύξησης της τιμής αυτής είναι $1/n$. Επιπλέον, με την χρήση αυτής της απλής τακτικής, αυξάνεται και η καθυστέρηση αναπαραγωγής από πλευράς του παραλήπτη (playout delay), καθώς πρέπει να περιμένει την παραλαβή ολόκληρης της ομάδας από πακέτα προτού ξεκινήσει την αναπαραγωγή.

Η δεύτερη τεχνική που ακολουθείται στα πλαίσια του FEC είναι η αποστολή μιας επιπλέον ροής με το ίδιο περιεχόμενο η οποία όμως είναι χαμηλότερης ευκρίνειας, σαν τα πλεονάζοντα δεδομένα. Η ροή με το χαμηλό bitrate αναφέρεται σαν η πλεονάζουσα ροή. Ουσιαστικά, κάθε πακέτο είναι αποτέλεσμα συνένωσης των δεδομένων από την αρχική ροή και του προηγούμενου πακέτου δεδομένων από την πλεονάζουσα ροή. Με αυτό τον

τρόπο όποτε εμφανιστούν απώλειες σε μη διαδοχικά πακέτα ο παραλήπτης είναι σε θέση να αποκρύψει αυτή την απώλεια, αναπαράγοντας στην θέση των χαμένων δεδομένων το αντίστοιχο μέρος από την ροή χαμηλής ευκρίνειας που παραλαμβάνεται με το επόμενο πακέτο της κανονικής ροής. Φυσικά, όπως είναι λογικό κατά αυτά τα διαστήματα η ποιότητα των πολυμέσων θα είναι σαφώς χαμηλότερης ποιότητας, εντούτοις η μη ύπαρξη απώλειας πακέτων προσφέρει σε γενικές γραμμές αρκετά καλή αντιληπτή ποιότητα στον παραλήπτη. Αξίζει να σημειωθεί ότι με την προσέγγιση αυτή, τα πακέτα που είναι αναγκαίο να παραληφθούν πριν ξεκινήσει η αναπαραγωγή είναι μόλις 2, διατηρώντας χαμηλό το playout delay. Σε αυτή την περίπτωση η αύξηση στον απαιτούμενο ρυθμό διάδοσης των δεδομένων εξαρτάται από το bit rate στο οποίο είναι κωδικοποιημένη η πλεονάζουσα ροή.

Μία τροποποίηση η οποία μπορεί να εξασφαλίσει ανάκτηση χαμένων δεδομένων και σε περίπτωση απώλειας διαδοχικών πακέτων, είναι η σύμπτυξη σε κάθε πακέτο, περισσότερων από μόνο ένα πακέτο από την πλεονάζουσα ροή. Όπως είναι αναμενόμενο αυτό προσφέρει καλύτερη ποιότητα για πιο ανταγωνιστικά περιβάλλοντα που εξυπηρετούνται από δίκτυα best- effort, ωστόσο αυξάνουν τόσο το απαιτούμενο εύρος ζώνης, όσο και το playout delay.

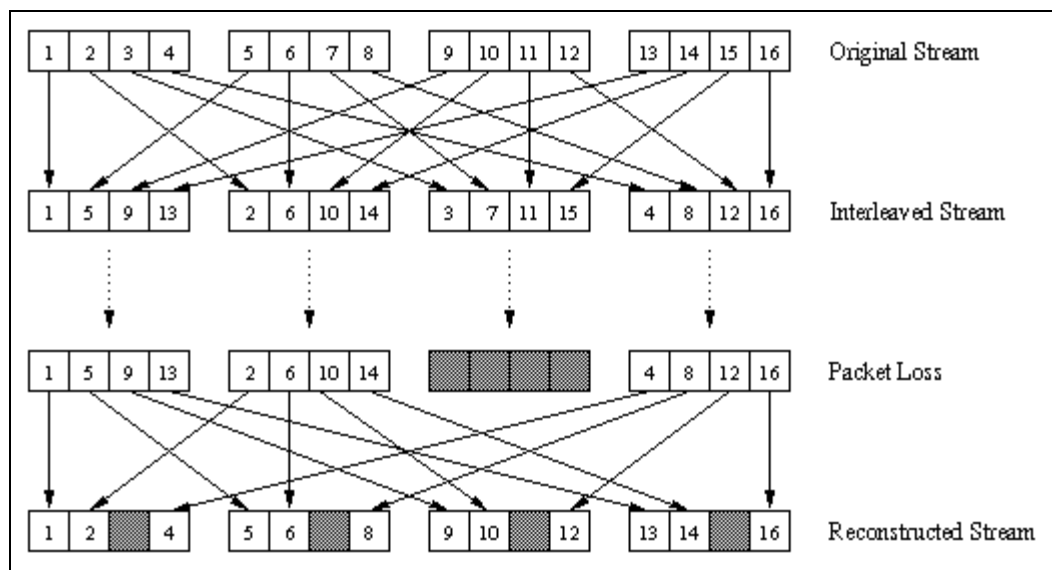


Σχήμα 5-1: Forward Error Correction με τη χρήση ροής χαμηλότερης ευκρίνειας

5.2.2 Ανάκτηση Δεδομένων μέσω Παρεμβολής (Interleaving)

Εναλλακτική λύση για την αποστολή πλεονάζοντων δεδομένων, μία εφαρμογή είναι σε θέση να αποστείλει παρεμβαλλόμενα δεδομένα. Ο αποστολέας επανατοποθετεί τις μονάδες δεδομένων που πρόκειται να αποσταλούν σε νέα ακολουθεία, έτσι ώστε αρχικά γειτονικά μονάδες απέχουν κατά μία συγκεκριμένη απόσταση στην νέα ροή. Προσφέρει τη δυνατότητα μετριασμού των συνεπειών από την απώλεια πακέτων. Αυτό επιτυγχάνεται καθώς, η απώλεια ενός πακέτου μπορεί να έχει σαν αποτέλεσμα, την ύπαρξη πολλαπλών μικρών κενών, στην ανοικοδομημένη ροή του παραλήπτη, σε αντίθεση με την ύπαρξη ενός μεγάλου κενού που θα λαμβάναμε χωρίς τη χρήση interleaving.

Με την παρεμβολή, αυξάνεται δραστικά η αντιληπτή ποιότητα των πολυμέσων που λαμβάνονται, και επιπλέον παρέχει χαμηλό επιπλέον κόστος όσο αφορά την απορρόφηση πόρων του δικτύου, αν και το κύριο μειονέκτημά της είναι η αύξηση της καθυστέρησης αφού απαιτείται η παραλαβή όλων των παρεμβαλλόμενων δεδομένων προτού εκκινήσει η αναπαραγωγή. Αυτό το γεγονός περιορίζει τη χρήση του σε streaming πολυμέσων και το καθιστά μη ελκυστικό για διαδραστικές εφαρμογές, όπως η τηλεδιάσκεψη. Το κύριο πλεονέκτημά του είναι η μη αύξηση του απαιτούμενου εύρους ζώνης για αποστολή των δεδομένων.



Σχήμα 5-23: Παράδειγμα Χρήσης Παρεμβολής για Αποστολή και Ανάκτηση Δεδομένων

5.3 Πρότυπα Συμπίεσης Ψηφιακού Βίντεο

5.3.1 Η ανάγκη για συμπίεση

Όπως είναι γνωστό το τυπικό αναλογικό σήμα PAL καταλαμβάνει εύρος ζώνης (bandwidth) ίσο με 8MHz. Για την ψηφιοποίηση του σήματος αυτού, χωρίς απώλεια πληροφορίας, και με βάση το θεώρημα Nyquist, πρέπει να πραγματοποιηθεί δειγματοληψία στη διπλάσια συχνότητα, δηλαδή στα 16MHz, που μεταφράζεται σε 16 εκατομμύρια δείγματα ανά δευτερόλεπτο. Για να λαμβάνουμε επαρκή ευκρίνεια (δηλαδή ικανοποιητικές αποχρώσεις του γκρι σε ασπρόμαυρο σήμα) πρέπει κάθε δείγμα να περιγράφεται από τουλάχιστον 8 bits, προσφέροντας έτσι $2^8=256$ διαφορετικά επίπεδα κωδικοποίησης. Αποτέλεσμα των πιο πάνω, είναι ο ψηφιακός ρυθμός μετάδοσης που αντιστοιχεί σε αυτά τα χαρακτηριστικά να είναι $16 \times 8 = 128$ Mbits/s.

Προφανώς ο ρυθμός αυτός είναι ιδιαίτερα υψηλός, πράγμα που καθιστά αδύνατη τη μετάδοση ασυμπίεστου ψηφιακού σήματος βίντεο ανάλυσης PAL. Το πρόβλημα του ρυθμού μετάδοσης γίνεται ακόμη εντονότερο αν θελήσουμε να έχουμε υψηλότερη ευκρίνεια (περισσότερες δηλαδή αποχρώσεις του γκρι σε ασπρόμαυρο σήμα), οπότε θα χρησιμοποιήσουμε μήκος λέξης των 16 ή 24 bits, ανάλογα με την επιθυμητή ποιότητα. Επιπλέον, και οι αντίστοιχοι ρυθμοί μετάδοσης που θα προκύψουν θα είναι υψηλότεροι. Είναι προφανές, λοιπόν, ότι καθιστάται αναγκαία η συμπίεση του βίντεο με στόχο την μείωση του απαιτούμενου ρυθμού μετάδοσης σε συνδυασμό με τη διατήρηση της αντιληπτής ποιότητας σε ικανοποιητικά για τον τελικό χρήστη επίπεδα.

Εκ του αποτελέσματος μπορεί να δηλωθεί ότι η συμπίεση βίντεο καθιστά δυνατή τόσο την μετάδοση ψηφιακού βίντεο τηλεοπτικής ανάλυσης (PAL ή NTSC) μέσω τηλεπικοινωνιακών δικτύων που δεν μπορούν να υποστηρίξουν τους υψηλούς ρυθμούς μετάδοσης του ασυμπίεστου βίντεο, όσο και την αποδοτικότερη αξιοποίηση των διαθέσιμων πόρων μετάδοσης, ουσιαστικά του εύρους ζώνης του δικτύου, και του αναγκαίου χώρου αποθήκευσης.

5.4 Ομάδες Εργασίας στην Ανάπτυξη Προτύπων Ψηφιακού Βίντεο

5.4.1 ITU-T VCEG

Η ομάδα εργασίας Video Coding Expert Group της ITU του τμήματος πρωτοτυποποίησης τηλεπικοινωνιακών θεμάτων (ITU-T) ασχολείται με την ανάπτυξη προτύπων ή συστάσεων για τις τηλεπικοινωνίες. Το τμήμα πρωτοτυποποίησης τηλεπικοινωνιακών θεμάτων συγκροτείται από υποομάδες εργασίες, όπου η κάθε υποομάδα έχει μια συγκεκριμένη θεματική περιοχή δράσης. Η υποομάδα 16 (SubGroup 16 ή SG 16) είναι υπεύθυνη για θέματα πολυμεσικών υπηρεσιών και εφαρμογών, όπως πολυμεσικά τερματικά, συστήματα, πρωτόκολλα και κωδικοποίηση σήματος. Η υποομάδα 16 υποδιαιρείται περαιτέρω σε μικρότερες εξειδικευμένες ομάδες εργασίας (working parties), εκ των οποίων η ομάδα 3 εξειδικεύεται με την κωδικοποίηση σήματος. Μεταξύ των ποικίλων ενοτήτων εργασίας αυτής της ομάδας, η ενότητα (question) 6 ασχολείται με τη ψηφιακή κωδικοποίηση βίντεο και ουσιαστικά εκείνα τα μέλη της που συμμετέχουν ενεργά στις συναντήσεις που πραγματοποιούνται στα πλαίσια της ενότητας αυτής, συνθέτουν την VCEG. Ως εκ τούτου, η επίσημη ονομασία της VCEG είναι ITU-T SG16 WP3 Q6.

Στόχος της είναι η προώθηση, διατύπωση και παραγωγή προτάσεων για μεθόδους κωδικοποίησης βίντεο, οι οποίες να είναι κατάλληλες για εφαρμογές συνδιάσκεψης και λοιπές εφαρμογές βίντεο (ροών βίντεο, τηλεοπτικής εκπομπής και ψηφιακού κινηματογράφου μεταξύ άλλων). Πέραν τούτου, η ITU-T SG16 WP3 Q6 εστιάζει και στην επέκταση των υπάρχοντων προτύπων κωδικοποίησης, ενώ δημιουργεί και τις κατάλληλες συνθήκες για νέες προτάσεις κωδικοποίησης.

5.4.2 ISO/MPEG

Η Moving Pictures Expert Group (MPEG) είναι μία ομάδα εργασίας του Διαθνή Οργανισμού Προτυποποίησης (International Organisation for Standardization - ISO) και της Διεθνής Ηλεκτροτεχνικής Επιτροπής (International Electrotechnical Committee – IEC). Επίσης η MPEG είναι το Working Group (WG) 11 της SubCommittee (SC) 29 του Joint

Technical Committee 1 και κατά συνέπεια η επίσημη ονομασία της είναι ISO/IEC JTC1/SC29/WG11.

Σκοπός της MPEG είναι η ανάπτυξη διεθνών προτύπων για συμπίεση, επεξεργασία και κωδικοποιημένη αναπαράσταση κινούμενων εικόνων, ήχου ή συνδυασμό και των δύο με σκοπό την κάλυψη ενός μεγάλου εύρους εφαρμογών.

Μεταξύ των κυριότερων στόχων της MPEG καθορίστηκαν και οι ακόλουθοι:

- Να δραστηριοποιείται εντός του ISO/IEC ως το υπεύθυνο σώμα για την πρόταση νέων προτύπων σχετικά με το αντικείμενο εργασίας.
- Να συνεργάζεται με παρόμοια σώματα προτυποποίησης που ασχολούνται με παρεμφερείς εφαρμογές.
- Να θέτει τις απαιτήσεις ώστε να είναι δυνατή η εφαρμογή των προτύπων της MPEG σε διαφορετικές εφαρμογές, όπως οι τηλεπικοινωνίες και η τηλεοπτική βιομηχανία.
- Να καθορίσει τις μεθόδους και διαδικασίες για την υποκειμενική εκτίμηση της αντιληπτής ποιότητας βίντεο, ήχου, κινούμενης εικόνας και συνδυασμό αυτών.
- Να αξιολογήσει τα ποικίλα χαρακτηριστικά των τεχνολογιών υλοποίησης των αλγορίθμων κωδικοποίησης ήχου, κινούμενης εικόνας και συνδυασμό αυτών.
- Να προτείνει πρότυπα για την κωδικοποιημένη αναπαράσταση κινούμενων εικόνων, ήχου και συνδυασμό αυτών.
- Να προτείνει πρότυπα για πρωτόκολλα που σχετίζονται με την κωδικοποιημένη αναπαράσταση κινούμενων εικόνων, ήχου ή συνδυασμό αυτών.

Για την επίτευξη των στόχων αυτών, η MPEG διαιρείται σε υποομάδες ειδικών εργασιών, όπου η κάθε μία εξειδικεύεται σε ένα συγκεκριμένο θέμα σχετικό με τις διαδικασίες προτυποποίησης. Οι ευθύνες των υποομάδων αυτών περιγράφονται στην συνέχεια.

Requirements: Εντοπισμός αναγκών και απαιτήσεων αγοράς για νέα πρότυπα

Systems: Μελέτη των προδιαγραφών των συστημάτων καταγραφής, αναπαραγωγής και επεξεργασίας ήχου, βίντεο και εικόνων, καθώς και των διαδικασιών μεταφοράς των δεδομένων αυτών μέσω τηλεπικοινωνιακών δικτύων.

Description: Μελέτη των δηλώσεων και περιγραφών των ψηφιακών πολυμεσικών αντικειμένων.

Video: Μελέτη της κωδικοποίησης της κινούμενης εικόνας.

Audio: Μελέτη της κωδικοποίησης ήχου.

Synthetic Natural Hybrid Coding: Μελέτη της κωδικοποίησης συνθετικού βίντεο και ήχου σε συνδυασμό με φυσικό βίντεο και ήχο.

Integration: Έλεγχος της συμβατότητας των υλοποιήσεων με το πρότυπο και ανάπτυξη του λογισμικού αναφοράς.

Test: Μελέτη των υποκειμενικών μεθόδων εκτίμησης της ποιότητας εικόνας βίντεο.

Implementation: Καθορισμός των προδιαγραφών ανάπτυξης και υλοποίησης σχετικών εφαρμογών.

Liaison: Ανάπτυξη δεσμών με άλλους σχετικούς φορείς και ομάδες εργασίας.

5.4.3 JVT

Η Joint Video Team (JVT) είναι μια ομάδα εργασίας που ιδρύθηκε το 2001 και αποτελείται από μέλη της ISO/IEC JTC1/SC29/WG11 (MPEG) και της ITU-T SG16 WP3 Q6 (VCEG). Η αιτία σύστασης της JVT ήταν η απαίτηση της MPEG για εργαλεία κωδικοποίησης βίντεο υψηλής απόδοσης. Βέλτιστος μηχανισμός κωδικοποίησης της εποχής ήταν ο MPEG-4 Part 2 Visual, που όμως ήταν βασισμένος σε μια αρκετά ξεπερασμένη τεχνολογία, που περιγραφόταν στο πρότυπο H.263 του 1995. Οι εξελίξεις που είχαν πραγματοποιηθεί στον χώρο της υπολογιστικής ισχύος και στον ερευνητικό τομέα της κωδικοποίησης κινούμενης εικόνας καθιστούσαν ξεκάθαρο το γεγονός ότι το έδαφος για ανάπτυξη ενός νέου προτύπου κωδικοποίησης ήταν γόνιμο. Το νέο πρότυπο θα αποτελούσε σημείο καμπής στην εξέλιξη της παρόδοσης των μεθόδων κωδικοποίησης βίντεο.

Μετά από την εκτίμηση των τότε γνωστών τεχνικών κωδικοποίησης κινούμενης εικόνας, η MPEG κατέληξε στο συμπέρασμα ότι ο υπό-ανάπτυξη αλγόριθμος κωδικοποίησης της VCEG H.26L ήταν ο καταλληλότερος για περαιτέρω βελτίωση και εξέλιξη, ώστε να επιτευχθεί ο στόχος για βέλτιστη κωδικοποίηση. Σαν αποτέλεσμα της επιλογής αυτής, αποφασίστηκε όπως μέλη της MPEG και της VCEG θα συστήσουν μια ομάδα δράσης που από κοινού θα διευθύνει τα τελικά στάδια εξέλιξης του H.26L. Η κοινή αυτή ομάδα δράσης ονομάστηκε JVT και αυτή η από κοινού εξέλιξη του H.26L οδήγησε τελικά στην

γέννηση του νεότερου προτύπου κωδικοποίησης κινούμενης εικόνας, με την ονομασία H.264, το 2003.

Αξίζει να σημειωθεί ότι ένα από τα πλέον προκαταρκτικά μελήματα των μελών της JVT ήταν η αναζήτηση και η εξεύρεση πιθανών υπαρκτών πατεντών ευρεσιτεχνίας σχετικών με τα βασικά προφίλ του υπό ανάπτυξη προτύπου. Στόχος ήταν η διασφάλιση της υλοποίησης του πλέον βασικού προφίλ του νέου προτύπου από οποιονδήποτε φορέα χωρίς να προκύπτουν ζητήματα πληρωμής για αδειοδότηση και πνευματικά δικαιώματα σε τρίτους φορείς που θα τα διατηρούν μέσω κάποιας σχετικής πατέντας ευρεσιτεχνίας. Η επιτυχία του νέου προτύπου θα κρινόταν σημαντικά από τέτοια ζητήματα ελεύθερης υλοποίησης, γεγονός που η JVT γνώριζε μέσα από την εμπειρία των μελών της και που ουσιαστικά φρόντισε να μη συμβεί.

5.5 Πρότυπα Κωδικοποίησης

5.5.1 ITU H.261

Αρχικά χρησιμοποιήθηκε σε παλαιότερα προϊόντα teleconferencing και videotelephony. Συνεπώς, ο αλγόριθμος κωδικοποίησης video πρέπει να είναι ικανός να λειτουργεί σε πραγματικό χρόνο με τη μικρότερη καθυστέρηση. Επιτύγχανε, εξάλλου, και αρκετά ικανοποιητική αντιληπτή ποιότητα. Είχε κατασκευαστεί από την ITU-T, και ήταν το πρώτο πρακτικό μοντέλο συμπίεσης video. Ουσιαστικά, όλοι οι επόμενοι σχεδιασμοί των video codecs βασίζονται σε αυτό.

Επειδή αρχικά σχεδιάστηκε για μεταφορά βίντεο πάνω από γραμμές ISDN, οι υποστηριζόμενοι ρυθμοί μετάδοσης της αρχικής έκδοσης ήταν ακέραια πολλαπλάσια των 64 kbps. Οι ρυθμοί bits για το H.261 κυμαίνονται από 64Kbps μέχρι και 2Mbps.

Συμπεριλαμβάνει καλά υλοποιημένες έννοιες όπως την YCbCr απεικόνιση χρώματος, το μοντέλο δειγματοληψίας 4:2:0, δείγμα ακριβείας 8-bit, 16x16 macroblocks, 8x8 block-wise tranformation και υποστηρίζει αναλύσεις CIF (352x288) και QCIF (176x144).

5.5.2 ITU H.263

Σχεδιάστηκε για να επιτρέπει την επικοινωνία σε χαμηλούς ρυθμούς δεδομένων και αρχικά χρησιμοποιήθηκε για videoconferencing, videotelephony και internet video.

Η σύσταση H.263 χρησιμοποιεί μία block motion-compensated DCT (Discrete Cosine Transformation) μέθοδο για την κωδικοποίηση του video. Η κωδικοποίηση κατά H.263 είναι πολύ πιο αποτελεσματική από ότι η κωδικοποίηση κατά H.261. Στην σύσταση H.263, η κωδικοποίηση πραγματοποιείται τεμαχίζοντας κάθε εικόνα σε macroblocks. Κάθε macroblock αποτελείται από 16x16 luminance blocks και 8x8 chrominance blocks. Κάθε macroblock μπορεί να κωδικοποιηθεί είτε ως intra (χωρίς συσχετίσεις ανάμεσα σε διαδοχικά frames) είτε ως inter (με συσχετίσεις ανάμεσα σε διαδοχικά frames). Η τεχνική DCT η οποία χρησιμοποιείται, παρέχει spatial redundancy και temporal redundancy. Η σύσταση H.263 στηρίζεται στην H.261 και παρέχει κάποιες επεκτάσεις για την υποστήριξη πιο αποτελεσματικής κωδικοποίησης. Ειδικά σε χαμηλά bit rates, μπορεί να παρέχει μια σημαντική βελτίωση στο bitrate που χρειάζεται για να φτάσουμε σε πειστικό επίπεδο ποιότητας εικόνας.

5.5.3 MPEG-1 Part 2

Το MPEG-1 στοχεύει στην κωδικοποίηση του video και του συνοδευόμενου ήχου σε ρυθμό bit περίπου 1.5Mbps. Το MPEG-1 μπορεί να χρησιμοποιηθεί σε υψηλότερους ή χαμηλότερους ρυθμούς δεδομένων από 1.5Mbps. Ανάμεσα στις προδιαγραφές του εξάλλου ήταν και η ανάγκη για συμπίεση εντός των ορίων ενός οπτικού δίσκου (700 MB) μίας μεγάλου μήκους ταινίας με ρυθμό bit μέχρι 1.5Mbps και χωρικής ανάλυσης CIF (352x288). Επίσης, η ποιότητα του συμπιεσμένου βίντεο που θα προέκυπτε θα έπρεπε να είναι εφάμιλλη του αναλογικού οικιακού προτύπου VHS.

Στα πλαίσια της μελέτης για το MPEG-1, έγινε αντιληπτό ότι εφόσον το συμπιεσμένο βίντεο προοριζόταν για μόνιμη αποθήκευση σε κάποιο οπτικό μέσο για μελλοντική προβολή, η καθυστέρηση που θα προέκυπτε κατά την κωδικοποίηση και συμπίεσή του δεν ήταν τόσο μεγάλης σημασίας όσο στο πρότυπο H.261 που στόχευε στην κωδικοποίηση και

αποστολή του κωδικοποιημένου σήματος σε πραγματικό χρόνο, αφού προοριζόταν για εφαρμογές πραγματικού χρόνου.

Η δυσκολία στο σχεδιασμό του αλγορίθμου MPEG ήταν το παρακάτω: από την μια μεριά να επιτύχει μια υψηλή αναλογία συμπίεσης, με χρήση μιας τεχνικής interframe κωδικοποίησης και από την άλλη υπήρχε η απαίτηση για δυνατότητα τυχαίας πρόσβασης στο video (κάτι που είναι δύσκολο να ικανοποιηθεί με την χρήση απλής κωδικοποίησης intraframe). Για να ικανοποιηθούν και οι δυο αυτές απαιτήσεις στο MPEG-1 video, μερικές εικόνες είναι κωδικοποιημένες intraframe και άλλες είναι κωδικοποιημένες interframe και intraframe. Το MPEG-1 καθορίζει τέσσερα είδη frames, εκ των οποίων τα Intra, Predicted και Bidirectionally Predicted frames ορίστηκαν και επεξηγήθηκαν με σαφήνεια στο προηγούμενο Κεφάλαιο. Το νέο είδος πλαισίου που χρησιμοποιείται είναι τα D frames και επεξηγείται στη συνέχεια.

- **D-frames:** Είναι εικόνες που χρησιμοποιούνται μόνο στο MPEG-1 και αποσύρθηκαν από ακόλουθα πρότυπα κωδικοποίησης. Είναι DC κωδικοποιημένες εικόνες που κωδικοποιούνται χωρίς αναφορά σε άλλες εικόνες. Από τους συντελεστές του DCT μόνο οι DC είναι υπάρχοντες (οι AC δεν χρησιμοποιούνται) και επομένως πρόκειται για εικόνες χαμηλής ποιότητας. Οι D- εικόνες χρησιμοποιούνται μόνο για κατάσταση εύρεσης γρήγορης-προώθησης.

Τα I frames καλούνται γενικά frames αναφοράς, ενώ τα P και τα B καλούνται γενικά frames predictive-coded. Η οργάνωση των τριών τύπων frames σε μια video ακολουθία είναι πολύ ευέλικτη. Η επιλογή αφήνεται στον αποκωδικοποιητή και εξαρτάται από τις απαιτήσεις της εφαρμογής. Τυπικά, ένα I-frame μπορεί να παρουσιάζεται κάθε μισό δευτερόλεπτο για να δώσει τυχαία γοργή πρόσβαση, με δυο B-frame εισερχόμενα ανάμεσα σε κάθε ζευγάρι των I ή P frames.

Κάθε εικόνα είναι διαιρεμένη σε macroblock των 16x16 pixels, που αποτελούν και την μικρότερη ανεξάρτητη μονάδα έγχρωμου βίντεο. Τα macroblocks στις I εικόνες είναι intracoded. Στις εικόνες P είναι intracoded ή predictive-coded βασισμένα στην προηγούμενη εικόνα I ή P, εξαρτώμενα από την ικανότητα κωδικοποίησης. Στις B εικόνες είναι intracoded, forward-predictive coded, backward-predictive coded ή bidirectionally-

predictive coded. Για τα macroblocks που είναι predictive-coded τα διανύσματα κίνησης βρίσκονται και τα predictive λάθη υπολογίζονται.

5.5.4 MPEG-2 Part 2

Το MPEG-2 στοχεύει στην κωδικοποίηση video σε τηλεοπτική studio-quality ποιότητα και σε κωδικοποίηση ήχου σε ποιότητα CD-quality. Το MPEG-2 απαιτεί 4 με 6 Mbps για την μετάδοση των δεδομένων που παράγει. Αναπτύχθηκε στοχεύοντας στη ψηφιακή τηλεόραση. Η χωρική ανάλυση της εικόνας ακολουθεί το τηλεοπτικό πρότυπο CCIR-601 (broadcast quality- ποιότητα εκπομπής), δηλαδή 704x480 pixels (NTSC) ή 704x576 pixels (PAL) για εικόνα πλεκτής σάρωσης (interlaced).

Το αντικείμενο του MPEG-2 είναι να βελτιώσει την ποιότητα μιας εικόνας, ενώ συγκρατεί τον ρυθμό bit κάτω από τα 10 Mbps. Το MPEG-2 είναι μια προέκταση του MPEG-1 το οποίο παρέχει υψηλή ποιότητα ήχο-οπτικής κωδικοποίησης. Έχει τέσσερα μέρη: MPEG-2 συστήματα, video, ήχο και συμμόρφωση. Αν και το MPEG-2 χρησιμοποιεί τις ίδιες ιδέες με το MPEG-1, είναι αρκετά πιο βελτιωμένο. Η συμπίεση MPEG-2 video χρησιμοποιεί την ίδια αρχή με την MPEG-1 συμπίεση με κάποιες επεκτάσεις και βελτιώσεις για να υποστηρίξει video υψηλής ποιότητας.

Το πρότυπο MPEG-2/ H.262 εισήγαγε την κλιμακωτή (scalable) κωδικοποίηση, όπου από μια μονή ροή MPEG-2 είναι δυνατό να εξαχθούν σήματα βίντεο διαφορετικής αντιληπτής ποιότητας (διαφορετική χρονική, χωρική, ποιοτική ανάλυση), ανάλογα με το ποσοστό της ροής που χρησιμοποιείται από τον αποκωδικοποιητή.

Υπάρχουν δυο κύριοι λόγοι: πρώτον, τα σημαντικά δεδομένα βίντεο τοποθετούνται σε υψηλή προτεραιότητα για την αποφυγή λάθους. Σε περίπτωση υψηλής συμφόρησης στο δίκτυο, τα δεδομένα χαμηλής προτεραιότητας απορρίπτονται πρώτα και τα υψηλής προτεραιότητας δεδομένα κρατούνται έτσι ώστε μια χαμηλότερης ποιότητας βίντεο να μπορεί να αναδομηθεί από αυτά τα δεδομένα. Αυτό αποκαλείται "ευχάριστος υποβιβασμός". Σε αντίθετη περίπτωση η αποστολή απλής ροής MPEG-2, θα ισοδυναμούσε με μη επιτυχή λήψη της υπηρεσίας, και κατά συνέπεια την καθολική απώλεια αυτής. Δεύτερον, το βαθμωτό βίντεο επιτρέπει στον αποκωδικοποιητή να αποκωδικοποιήσει επιλεκτικά μέρος ενός βίντεο. Η παραπάνω τεχνική είναι γνωστή και ως Layered Encoding.

Στο πρότυπο MPEG-2, ο αριθμός των B-frames μπορεί να μεταβληθεί ώστε να επιτύχουμε την επιθυμητή πολυπλοκότητα και κατά συνέπεια χρονική υστέρηση κατά την κωδικοποίηση. Η τεχνική αυτή μπορεί να βρει εφαρμογή σε τηλεπικοινωνιακά συστήματα, όπου απαιτείται σε πραγματικό χρόνο η ψηφιοποίηση και συμπίεση του οπτικοακουστικού συστήματος. Εξαιτίας αυτής της δυνατότητας προσαρμογής της πολυπλοκότητας του προτύπου MPEG-2, η ITU-T θιοθέτησε το πρότυπο MPEG-2 υπό το γενικότερο όνομα H.262 και για τις τηλεπικοινωνιακές εφαρμογές.

Οι εφαρμογές που συναντάμε το MPEG-2 είναι στην καλωδιακή τηλεόραση (cable TV), στη ψηφιακή δορυφορική τηλεόραση (Digital Video Broadcasting Satellite/ DVB-S) αλλά και στη αντίστοιχη επίγεια ψηφιακή τηλεόραση (Digital Video Broadcasting Terrestrial/ DVB-T). Επιπλέον, το MPEG-2 έχει γνωρίσει μεγάλη επιτυχία στην αποθήκευση και διανομή κινηματογραφικών ταινιών μέσω του οπτικού δίσκου DVD (Digital Versatile Disk).

5.5.5 MPEG-3

Μετά το πρότυπο MPEG-2, το οποίο προσέφερε κωδικοποίηση με ανάλυση ίση με το τηλεοπτικό σήμα και υψηλό ρυθμό κωδικοποίησης, οι ερευνητές εστίασαν τις προσπάθειες στην ανάπτυξη ενός προτύπου, το οποίο θα επέτρεπε την κωδικοποίηση βίντεο σήματος υψηλής ευκρίνειας. Οι προσπάθειες για την ανάπτυξη και καθορισμό του νέου προτύπου, του επονομαζόμενου MPEG-3, εν τέλει εγκαταλείφθηκαν αφού η εμπορική επιτυχία του MPEG-2 και η επέκταση των δυνατοτήτων αυτού, σύμφωνα με τις τροποποιήσεις του αρχικού προτύπου, να κωδικοποιεί βίντεο σε αναλύσεις υψηλής ευκρίνειας δεν άφησε κανένα ουσιαστικό περιθώριο για περαιτέρω ενασχόληση με το MPEG-3.

5.5.6 MPEG-4 Part 2

Το MPEG-4 είναι μια πρωτοβουλία μέσα στην όλη διαδικασία του MPEG με στόχο να βελτιώσει την κωδικοποίηση των δεδομένων παρέχοντας υψηλής ποιότητας κωδικοποιημένο βίντεο σε πολύ χαμηλό ρυθμό κωδικοποίησης, της τάξης των 64kbps και λιγότερο. Εκτός από αυτά, το MPEG-4 συμπεριλάμβανε και μια πληθώρα άλλων

εργαλείων για την κωδικοποίηση πολυμεσικών μορφών βίντεο. Το πρότυπο αυτό εισάγει για πρώτη φορά την έννοια της αντικειμενοστραφούς κωδικοποίησης βίντεο. Στην ουσία, ο όρος οπτικοακουστικά αντικείμενα χρησιμοποιείται για να περιγράψει τις διάφορες οντότητες που απαρτίζουν την εικόνα, οι οποίες μπορούν να διαχειριστούν αυτόνομα και ανεξάρτητα από τις υπόλοιπες, τόσο από τον κωδικοποιητή όσο και από τον αποκωδικοποιητή. Με τον όρο οντότητες εννοούμε φυσικά ή συνθετικά αντικείμενα και ήχους που όλα μαζί συνθέτουν το οπτικοακουστικό σήμα (π.χ. ήχος, αφήγηση, συνθετική αφήγηση, συνθετικός ήχος, βίντεο, εικόνες, δισδιάστατα γραφικά, τρισδιάστατα γραφικά, κείμενο, συνθετική απεικόνιση προσώπου και σώματος). Το MPEG-4 συνδυάζει όλα αυτά τα φυσικά ή συνθετικά αντικείμενα σύμφωνα με ένα περιγραφέα σκηνής και με χρήση κατάλληλου συγχρονισμού, ώστε να αναπαραστήσει σωστά την αρχική εικόνα που αναλύθηκε και εισήχθηκε για κωδικοποίηση.

Η χωρική ανάλυση της εικόνας μπορεί να είναι στα χαμηλά επίπεδα των 176x144 pixels (QCIF) σε σχετικά χαμηλούς ρυθμούς κωδικοποίησης που κυμαίνονται ανάμεσα στα 48 έως 128 kbps, κατάλληλα δηλαδή για μετάδοση σε δίκτυα με περιορισμένο εύρος ζώνης ανά συνδρομητή, όπως τα δίκτυα κινητών τηλεπικοινωνιών.

Συνεπώς, το πρότυπο MPEG-4 κάλυψε και μια πληθώρα νέων πολυμεσικών εφαρμογών, όπως η κωδικοποίηση βίντεο με τη δυνατότητα πολλαπλών σκηνών θέασης, συνθετικό βίντεο, συνδυασμό συνθετικού με πραγματικό βίντεο και διαδραστικό βίντεο. Οι ποικίλες αυτές μορφές ενσωματώθηκαν στο πρότυπο του MPEG-4 μέσω διαφόρων εργαλείων κωδικοποίησης, τα οποία είναι οργανωμένα σε προφίλ (profile). Το κάθε προφίλ προσδιορίζει και ορίζει ένα προτεινόμενο σύνολο εργαλείων κωδικοποίησης του προτύπου, που είναι κατάλληλα για συγκεκριμένες εφαρμογές. Η ταξινόμηση των προφίλ περιλαμβάνει:

- Απλά προφίλ που περιλαμβάνουν τα εργαλεία που απαιτούνται για την κωδικοποίηση φυσικών τετράγωνων frames.
- Αντικειμενοστραφή προφίλ για την κωδικοποίηση οπτικών αντικειμένων τυχαίου σήματος.
- Προφίλ ακίνητης εικόνας για την κωδικοποίηση ακίνητων εικόνων.

- Κλιμακωτά προφίλ για την κωδικοποίηση σε πολλαπλά χωρικά, χρονικά και ποιοτικά επίπεδα.
- Προφίλ στούντιο για την κωδικοποίηση υψηλής ποιότητας βίντεο για επαγγελματικές εφαρμογές.

Κάθε υλοποίηση MPEG-4 είναι συμβατή με ορισμένα προφίλ του προτύπου, που σημαίνει ότι μπορεί να κωδικοποιήσει/ αποκωδικοποιήσει μόνο περιεχόμενο MPEG-4, που είναι κωδικοποιημένο μόνο σύμφωνα με το συγκεκριμένο προφίλ. Πέραν τούτου, το κάθε προφίλ είναι υποδιαιρεμένο σε διαφορετικά επίπεδα (levels), τα οποία ορίζουν διάφορα επίπεδα πολυπλοκότητας τόσο σε χωρική και χρονική ανάλυση, όσο και σε ρυθμό κωδικοποίησης.

Αυτή την στιγμή το πρότυπο αποτελείται από 23 μέρη, εκ των οποίων τα μέρη 1-8 και 10 ήταν δημοσιευμένα στην αρχική έκδοση του προτύπου, ενώ τα υπόλοιπα ήταν ακόμη υπό ανάπτυξη και δεν είχαν καταλήξει στην αρχική τους μορφή. Το δεύτερο μέρος (part 2) στο οποίο αναφερόμαστε στην ενότητα αυτή, καταπιάνεται με την κωδικοποίηση φυσικών και συνθετικών οπτικών αντικειμένων.

Το μέρος αυτό αποτελείται από 21 διαφορετικά προφίλ, με σημαντικότερα το Simple και το Advanced Simple. Το πρώτο χρησιμοποιείται κυρίως σε περιπτώσεις στις οποίες χαμηλό bitrate και χαμηλό resolution καθίστανται απαραίτητα από από χαρακτηριστικά των εφαρμογών, όπως το εύρος ζώνης του δικτύου ή το μέγεθος της συσκευής. Το Advanced Simple Profile παρέχει μερικά επιπλέον χαρακτηριστικά, πέραν αυτών του Simple Profile. Συγκεκριμένα, υποστηρίζει κβαντοποίηση τύπου MPEG, συμπλεγμένο (interlaced) βίντεο, B- frames, αντιστάθμιση κίνησης $\frac{1}{4}$ pixel (Qpel), καθώς επίσης και καθολική αντιστάθμιση κίνησης (Global Motion Compensation- GMC). Η υποστήριξη κβαντοποίησης τύπου MPEG, συμπλεγμένου βίντεο και B- frames είναι σχεδιασμένες με παρόμοιο τρόπο, όπως παρουσιάζονται στο πρότυπο MPEG-2 Part 2. Από την άλλη η αντιστάθμιση κίνησης Qpel ήταν πρωτοποριακή και για αυτό τον λόγο υιοθετήθηκε και στο MPEG-4 Part 10. Τέλος, η καθολική αντιστάθμιση κίνησης δεν υποστηρίζεται σε όλες τις υλοποιήσεις, αν και το πρότυπο καθορίζει ότι οι αποκωδικοποιητές οφείλουν να το υποστηρίζουν.

Γνωστές εφαρμογές που υλοποιούν το πρότυπο είναι τα DivX, Xvid καθώς επίσης και το QuickTime.

Αν και το πρότυπο αυτό δεν πέτυχε σημαντική βελτίωση στον λόγο ποιότητας έναντι του ποσοστού συμπίεσης, εντούτοις πρέπει να αναγνωριστεί η σημαντική συνεισφορά του στην εξέλιξη και διάδοση των προτύπων κωδικοποίησης ψηφιακού βίντεο, κυρίως μέσω των κινητών τερματικών.

5.5.7 MPEG-4 Part 10/ Advanced Video Coding

5.5.7.1 Επισκόπηση

Το H.264 είναι ένα πρότυπο για την συμπίεση video, το οποίο είναι ισοδύναμο με το MPEG-4 Part 10 και το MPEG-4 AVC (Advanced Video Coding). Πρόκειται για το, μέχρι σήμερα, αποδοτικότερο και πιο σύγχρονο πρότυπο συμπίεσης που βασίζεται στην τεχνική της αντιστάθμισης κίνησης με βάση το μπλοκ (block-oriented motion compensation). Παρά το γεγονός ότι το H.264 αναφέρεται σαν επέκταση του MPEG-4, εντούτοις το MPEG-4 Visual και το Part 10 έχουν σημαντικές διαφορές, αφού επιδιώκουν την βελτίωση των προηγούμενων προτύπων με διαφορετικούς τρόπους.

Στόχος του MPEG-4 Visual είναι να επεκταθεί πέρα από την κωδικοποίηση ορθογώνιων κινούμενων εικόνων βίντεο και να παρέχει ένα ανοικτό, εύκαμπτο πλαίσιο για τις πολυμεσικές εφαρμογές. Αντίθετα, το H.264 εξακολουθεί να στοχεύει στη συμπίεση των ορθογώνιων εικόνων βίντεο, προσεγγίζοντας όμως το θέμα με αποδοτικότερο, ανθεκτικότερο σε λάθη και αποτελεσματικότερο τρόπο σε σχέση με προηγούμενα πρότυπα.

Ουσιαστικά σαν κύριο μέλημα της Joint Video Team ήταν η ανάπτυξη ενός προτύπου ικανό να παρέχει ικανοποιητική ποιότητα video σε σημαντικά χαμηλά bit rates παρά προηγούμενα πρότυπα, χωρίς να αυξάνει την πολυπλοκότητα στον σχεδιασμό σε βαθμό που να καθιστά δύσκολη ή και αποτρεπτική την υλοποίησή του. Επιπρόσθετος στόχος ήταν η παροχή αρκετής ελαστικότητας ώστε να μπορεί να εφαρμοστεί σε ένα μεγάλο εύρος από εφαρμογές, δίκτυα και συστήματα συμπεριλαμβανομένων διαφορετικών bit rates, ψηλής ή χαμηλής ευκρίνειας βίντεο, δυνατότητας αποθήκευσης σε DVD, ενθυλάκωσης σε πακέτα RTP και χρήσης σε πολυμεσικά συστήματα τηλεφωνίας της ITU-T.

Το πρότυπο H.264 αποτελείται από διάφορα προφίλ, τα οποία και θα περιγραφούν στη συνέχεια. Κάθε αποκωδικοποιητής είναι σε θέση να αποκωδικοποιήσει τουλάχιστον ένα

προφίλ, αλλά όχι αναγκαστικά όλα. Το προφίλ που είναι δυνατό να αποκωδικοποιηθεί καθορίζεται στις προδιαγραφές του.

Η προτυποποίηση της πρώτης έκδοσης του H.264/AVC ολοκληρώθηκε τον Μάιο του 2003. Αργότερα, η JVT εξέδωσε επεκτάσεις του αρχικού προτύπου οι οποίες έγιναν γνωστές σαν Fidelity Range Extensions (FRExt). Οι επεκτάσεις αυτές επιτρέπουν κωδικοποίηση βίντεο υψηλότερης ποιότητας μέσω της υποστήριξης αυξημένης ακρίβειας βάθους δειγματοληψίας σε bit και πληροφορίες χρώματος υψηλότερης ευκρίνειας, συμπεριλαμβανομένων δομών δειγματοληψίας που είναι γνωστές σαν YUV 4:2:2 και YUV 4:4:4. Άλλα χαρακτηριστικά περιλαμβάνουν προσαρμοστική εναλλαγή ανάμεσα σε 4x4 και 8x8 μετασχηματισμούς ακεραίων και αποτελεσματική κωδικοποίηση χωρίς απώλειες μεταξύ διαφορετικών εικόνων. Ο σχεδιασμός του FRExt ολοκληρώθηκε τον Ιούλιο του 2004 και η συγγραφή τον Σεπτέμβριο της ίδιας χρονιάς.

Συνεχίζοντας τις επεκτάσεις του προτύπου, σε αυτό έχουν προστεθεί πέντε νέα προφίλ με που απευθύνονται κυρίως σε εξειδικευμένες επαγγελματικές εφαρμογές. Οι προσθήκες περιλαμβάνουν υποστήριξη για επεκταμένο φάσμα χρώματος, καθορισμό επιπλέον δεικτών για λόγο εμφάνισης (aspect ratio) ενώ παράλληλα έκρινε αναγκαία την αναθεώρηση ενός από τα προφίλ του FRExt, καθώς η ανατροφοδότηση από την αγορά υπέδειξε ότι υπήρχε ανάγκη για διαφοροποίηση στον σχεδιασμό.

Η Κλιμακωτή Κωδικοποίηση Βίντεο (Scalable Video Coding- SCV), επιτρέπει την δημιουργία ροών από bits που περιέχουν υπο-ροές που υπακούουν στο H.264/AVC. Για χρονική κλιμάκωση, δηλαδή ύπαρξη υπο-ροής με μικρότερο ρυθμό χρονικής δειγματοληψίας από αυτό της αρχικής ροής, ολόκληρες μονάδες πρόσβασης αφαιρούνται κατά την εξαγωγή της υπο-ροής. Σε τέτοια περίπτωση οι εικόνες αναφοράς που χρησιμοποιούνται από άλλες εικόνες για πρόβλεψη στο bitstream, δημιουργούνται ανάλογα. Για χωρική κλιμάκωση και κλιμακώτητα ποιότητας, δηλαδή η ύπαρξη υποροής με χαμηλότερη χωρική ευκρίνεια ή ποιότητα από την αρχική ακολουθία από bits, αφαιρείται κατά την εξαγωγή της υποροής από την αρχική ροή, το NAL (Network Abstraction Layer). Σε αυτή την περίπτωση η πρόβλεψη μεταξύ διαφορετικών εικόνων, δηλαδή η πρόβλεψη του σήματος υψηλότερης χωρικής ευκρίνειας ή ποιότητας από δεδομένα χαμηλότερης ευκρίνειας ή ποιότητας, χρησιμοποιείται συνήθως για

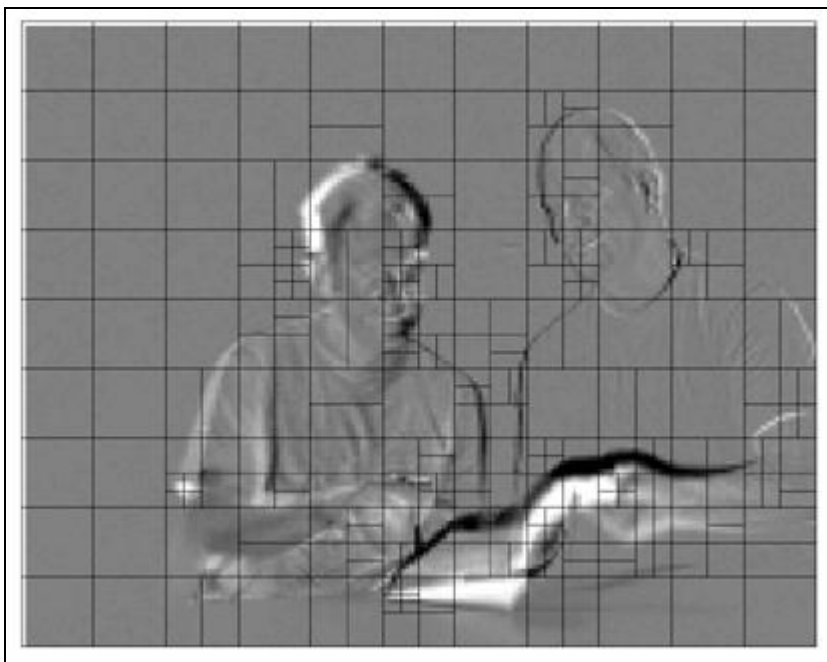
αποδοτικότερη κωδικοποίηση. Η ολοκλήρωση της επέκτασης SVC έγινε τον Νοέμβριο του 2007.

5.5.7.2 Κύρια Χαρακτηριστικά

Το H.264/MPEG-4 AVC περιλαμβάνει μία σειρά από νέα χαρακτηριστικά που επιτρέπουν την συμπίεση βίντεο πολύ πιο αποδοτικά από παλαιότερα πρότυπα, και προσφέρει ελαστικότητα στις εφαρμογές που το χρησιμοποιούν, σε μία μεγάλη ποικιλία από δικτυακές συνθήκες. Πιο συγκεκριμένα μερικά από τα κύρια χαρακτηριστικά του περιγράφονται στην συνέχεια.

Πρόβλεψη μεταξύ διαφορετικών εικόνων με την εμπλοκή πολλών εικόνων(inter-picture prediction) η οποία περιλαμβάνει αρκετές ιδιότητες. Αρχικά, χρησιμοποιούνται οι προηγούμενα κωδικοποιημένες εικόνες σαν εικόνες αναφοράς με πολύ πιο ελαστικό τρόπο σε σχέση με προηγούμενα πρότυπα, επιτρέποντας την ύπαρξη μέχρι και 16 πλαισίων αναφοράς. Αυτό έρχεται σε αντίθεση με προηγούμενα πρότυπα, στα οποία το όριο ήταν συνήθως ένα πλαίσιο, ή στην περίπτωση των συμβατικών B- frames, δύο. Αυτή η ιδιότητα επιτρέπει μικρές βελτιώσεις όσο αφορά το bit rate και την ποιότητα σε ορισμένες σκηνές. Επιπλέον ιδιότητα είναι η δυνατότητα χρήση μεταβλητού μπλοκ στην αντιστάθμιση κίνησης, με μεγέθη από 16x16 έως 4x4, επιτρέποντας την ακριβή τεμάχιση κινούμενων περιοχών. Τα υποστηριζόμενα μεγέθη μπλοκ πρόβλεψης όσο αφορά την φωτεινότητα είναι 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4, αρκετά από τα οποία δύναται να χρησιμοποιηθούν μαζί στο ίδιο macroblock. Τα chroma μπλοκς πρόβλεψης είναι αντιστοίχως μικρότερα ανάλογα με την δειγματοληψία που χρησιμοποιείται. Ακόμη, υπάρχει η δυνατότητα χρήσης πολλαπλών διανυσμάτων κίνησης ανά macroblock, με μέγιστο αριθμό τα 32 αν αναφερόμαστε σε B macroblocks που αποτελούνται από 16 κομμάτια 4x4. Μέσα από την ευχέρεια χρήσης οποιουδήποτε τύπου macroblock σε B-frames, είναι εφικτή η πιο αποδοτική κωδικοποίηση όταν χρησιμοποιούνται αυτού του είδους πλαίσια. Εξάλλου προσφέρεται ακρίβεια ενός τετάρτου του pixel για αντιστάθμιση κίνησης, επιτρέποντας ακριβή περιγραφή της μετατόπισης σε κινούμενες περιοχές. Η κίνηση στο ένα τέταρτο του pixel που είναι αποτέλεσμα γραμμικής παρεμβολής και γλυτώνει επιπλέον υπολογιστική ισχύ. Το πρότυπο H.264 επιτρέπει σταθμισμένη πρόβλεψη, επιτρέποντας σε ένα κωδικοποιητή να καθορίσει την χρήση μίας κλίμακας και

offset όταν εκτελεί αντιστάθμιση κίνηση, πράγμα που προσφέρει σημαντικά πλεονεκτήματα απόδοσης σε συγκεκριμένες περιπτώσεις, όπως για παράδειγμα βαθμιαίο σβήσιμο εικόνας (fade out) και βαθμιαία μετάβαση μεταξύ εικόνων.



Σχήμα 5-3: Παράδειγμα κωδικοποίησης H.264 με μεταβλητό μέγεθος μπλοκ

Πέραν των πιο πάνω το H.264 διαφέρει σε σχέση με το MPEG 2 και 4 Visual, και το H.263, καθώς προσφέρει χωρική πρόβλεψη από τα άκρα γειτονικών μπλοκς για intra κωδικοποίηση. Άλλο χαρακτηριστικό του προτύπου στο οποίο αναφερόμαστε είναι η κωδικοποίηση macroblock χωρίς απώλειες μέσω δύο διαφορετικών μηχανισμών, πρώτα με ένα τρόπο αναπαράστασης PCM macroblock χωρίς απώλειες, όπου τα δεδομένα από τη δειγματοληψία αναπαρίστανται απευθείας επιτρέποντας την τέλεια αναπαράσταση συγκεκριμένων περιοχών, και επιπλέον μέσω δεύτερου ενισχυμένου μηχανισμού που επιτρέπει την τέλεια και χωρίς απώλειες αναπαράσταση των κινούμενων περιοχών με την χρήση λιγότερων bits παρά ο προηγούμενος PCM τρόπος.

Επιπλέον, το πρότυπο H.264 ενσωματώνει στην αρχιτεκτονική του αποκωδικοποιητή ένα φίλτρο εξομάλυνσης των οπτικών τεχνουργημάτων, όπως το φαινόμενο τετραγώνισης. Μέσω αυτής της τεχνικής επιτυγχάνεται βελτιστοποίηση του αποκωδικοποιημένου οπτικού σήματος και της τελικής αντιληπτής ποιότητας.

Τα χαρακτηριστικά προσαρμοστικότητας και ανάκτησης της χαμένης πληροφορίας είναι αρκετά σημαντικά για το νέο πρότυπο. Αρχικά παρουσιάζουμε το Network Abstraction Layer(NAL), που επιτρέπει στην ίδια σύνταξη video να χρησιμοποιείται πάνω από διάφορες δικτυακές συνθήκες. Μία θεμελιώδης αρχή σχεδίασης του H.264, είναι η ικανότητα δημιουργίας πακέτων στα οποία δεν θα παρουσιάζεται η διπλότυπη επικεφαλίδα όπως το Header Extension Code του MPEG-4. Αυτό επιτεύχθηκε μέσα από τον διαχωρισμό πληροφοριών σχετικών με περισσότερα από ένα μέρη της ροής. Ο συνδυασμός των παραμέτρων ψηλότερου επιπέδου αποτελεί ένα σύνολο παραμέτρων. Στις προδιαγραφές του H.264 ορίζονται δύο τύπου συνόλων παραμέτρων, το Sequence Parameter Set(SPS) και το Picture Parameter Set(PPS). Μια ενεργή ακολουθία από σύνολα παραμέτρων παραμένει αναλλοίωτη κατά τη διάρκεια μίας κωδικοποιημένης ακολουθίας βίντεο και ένα ενεργό σύνολο από παραμέτρους εικόνας παραμένει αναλλοίωτο εντός της ίδιας κωδικοποιημένης εικόνας. Τα δύο σύνολα περιέχουν πληροφορίες όπως μέγεθος εικόνας και προαιρετικούς τρόπους κωδικοποίησης που χρησιμοποιήθηκαν. Άλλος τρόπος που κάνει το H.264 ανθεκτικό σε σφάλματα είναι η ελαστικότητα στην ταξινόμηση των macroblocks (Flexible macroblock ordering- FMO) και η αυθαίρετη ταξινόμηση των slices (arbitrary slice ordering- ASO), που αποτελούν τεχνικές για ανοικοδόμηση της σειράς με την οποία παρουσιάζονται στην οθόνη τα macroblocks. Ο μηχανισμός του διαχωρισμού δεδομένων ανάλογα με την σημαντικότητά τους, επιτρέπει την κατάταξή τους σε διαφορετικούς τύπους πακέτων δεδομένων, επιτρέποντας την εφαρμογή της άνισης προστασίας από σφάλματα. Επιπλέον, χρησιμοποιείται η τεχνική της αποστολής πλεονάζουσας πληροφορίας(συνήθως σε χαμηλότερη ευκρίνεια) που μπορεί να χρησιμοποιηθεί σε περίπτωση που μέρος της αρχικής πληροφορίας παραληφθεί με σφάλμα ή χαθεί κατά τη μεταφορά της. Τέλος, υλοποιείται η αρίθμηση των πλαισίων, ένα χαρακτηριστικό που επιτρέπει τη δημιουργία υπο-ακολουθιών και δίνει την ευχέρεια για εφαρμογή χρονικής κλιμάκωσης μέσα από την προαιρετική συμπερίληψη εικόνων μεταξύ άλλων εικόνων, και προσφέρει δυνατότητα για ανίχνευση και απόκρυψη των σφαλμάτων που προκύπτουν μέσα από απώλειες στο δίκτυο.

Αυτά είναι τα κύρια χαρακτηριστικά και τεχνικές που, μεταξύ άλλων, υποβοηθούν το H.264 να λειτουργεί σημαντικά καλύτερα από προηγούμενα πρότυπα, κάτω από ποικίλες περιστάσεις και δικτυακές συνθήκες. Αξίζει να σημειωθεί, ότι το H.264 μπορεί να αποδίδει

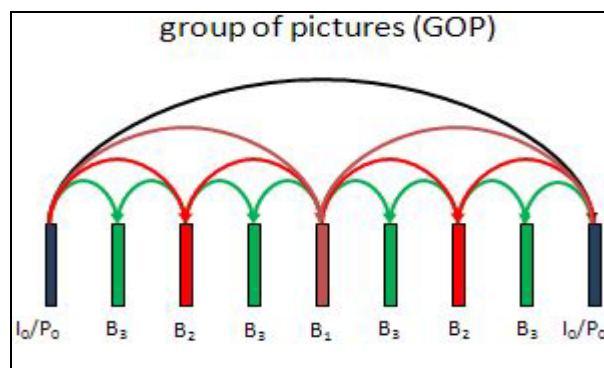
υπερβολικά καλύτερα από το MPEG- 2 βίντεο, επιτυγχάνοντας την ίδια ποιότητα ενώ απαιτεί μισό ή και λιγότερο bit rate από αυτό, ειδικά σε ψηλό bit rate και υψηλή ευκρίνεια.

5.5.7.2.1 Η Πρόβλεψη ως Τεχνική Συμπίεσης Βίντεο

Είναι μια διαδικασία με την οποία δημιουργείται ένα σύνολο προβλεπόμενων τιμών (συχνά βασιζόμενες εν μέρει σε κάποια ένδειξη που στάλθηκε από έναν κωδικοποιητή) που χρησιμοποιούνται για να προβλέψουν τις τιμές των εισερχόμενων δειγμάτων βίντεο, έτσι ώστε οι πραγματικές τιμές που χρειάζονται να προκύψουν μόνο από τις, ευκολότερα κωδικοποιούμενες, διαφορές από τις προβλεφθείσες τιμές. Αυτές οι διαφορές καλούνται κατάλοιπα (residual values).

Χρονική Κλιμάκωση (Temporal Scalability)

Στο H.264 κάθε εικόνα μπορεί να σημειωθεί σαν εικόνα αναφοράς και να χρησιμοποιηθεί για την πρόβλεψη μέσω αντιστάθμισης κίνησης για προηγούμενες εικόνες, ανεξάρτητα από τον τύπο κωδικοποίησης των slices. Η ιδιότητα αυτή επιτρέπει την κωδικοποίηση ακολουθιών εικόνων με αυθαίρετες χρονικές εξαρτήσεις.



Σχήμα 5-4: Ιεραρχική δομή για πρόβλεψη στο H.264

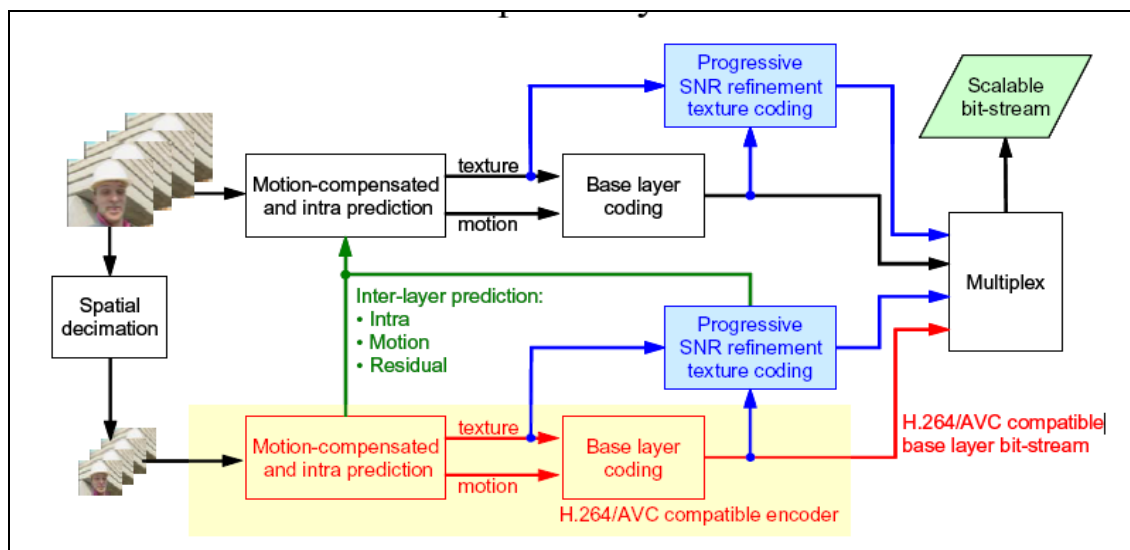
Ροή bit που να υποστηρίζει χρονική κλιμάκωση μπορεί να παραχθεί χρησιμοποιώντας την ιεραρχική δομή πρόβλεψης που φαίνεται πιο πάνω, χωρίς να απαιτούνται αλλαγές στο MPEG-4/AVC. Τα key pictures κωδικοποιούνται ανά τακτά χρονικά διαστήματα κάνοντας χρήση μόνο προηγούμενων key pictures σαν εικόνες αναφοράς. Οι υπόλοιπες εικόνες μεταξύ δύο key pictures κωδικοποιούνται ιεραρχικά όπως φαίνεται και στην πιο πάνω

εικόνα. Είναι προφανές ότι η ακολουθία των key pictures απεικονίζει την πιο απλή χρονική ευκρίνεια που υποστηρίζεται, η οποία μπορεί να βελτιώνεται με την προσθήκη των αμέσως επόμενων επιπέδων χρονικής πρόβλεψης.

Πέρα από τα πιο πάνω, η ιεραρχική δομή πρόβλεψης παρέχει ενισχυμένη αποδοτικότητα κωδικοποίησης σε σχέση με την κλασική μορφή κωδικοποίησης. Αξίζει επίσης να σημειωθεί ότι η καθυστέρηση της ιεραρχικής δομής μπορεί να περιοριστεί, μειώνοντας την πρόβλεψη μέσω αντιστάθμισης κίνησης από μελλοντικές εικόνες.

Χωρική Κλιμάκωση (Spatial Scalability)

Η χωρική κλιμάκωση επιτυγχάνεται μέσα από την προσέγγιση της πυραμίδας υπερδειγματοληψίας. Οι εικόνες που ανήκουν σε διαφορετικά χωρικά στρώματα κωδικοποιούνται ανεξάρτητα μαζί με συγκεκριμένες πληροφορίες κίνησης που απευθύνονται αποκλειστικά στο στρώμα αυτό.



Σχήμα 5-5: Δομή κωδικοποιητή με δύο χωρικά στρώματα

Όπως φαίνεται και στην πιο πάνω εικόνα όμως, για μεγαλύτερη απόδοση στην κωδικοποίηση των ενισχυτικών στρωμάτων, γίνεται χρήση επιπλέον μηχανισμών που επιτρέπουν την πρόβλεψη μεταξύ εικόνων που ανήκουν σε διαφορετικά στρώματα. Ο τρόπος υλοποίησης δίνει τη δυνατότητα στον κωδικοποιητή να επιλέγει κάθε φορά ένα από αυτούς τους μηχανισμούς πρόβλεψης, έτσι ώστε να εκμεταλλεύεται την καταλληλότερη

πληροφορία του βασικού στρώματος που θα αποφέρει τα πιο αποδοτικά ενισχυτικά στρώματα. Καθώς οι αρχές της πρόβλεψης μεταξύ εικόνων διαφορετικών στρωμάτων περιλαμβάνουν τεχνικές για παραμέτρους κίνησης και κατάλοιπης πρόβλεψης, οι δομές χρονικής πρόβλεψης των διαφόρων στρωμάτων θα πρέπει να είναι συγχρονισμένες για αποδοτικότερη πρόβλεψη μεταξύ διαφορετικών στρωμάτων.

5.5.7.2.2 Τεχνικές Πρόβλεψης μεταξύ Διαφορετικών Στρωμάτων

Στο σχεδιασμό του Scalable Video Coding περιλαμβάνονται οι τρεις τεχνικές πρόβλεψης μεταξύ διαφορετικών στρωμάτων που περιγράφονται στη συνέχεια.

Διαστρωματική Πρόβλεψη Κίνησης

Για να είναι δυνατή η εφαρμογή των δεδομένων κίνησης του στρώματος βάσης στην κωδικοποίηση των χωρικών ενισχυτικών στρωμάτων, έχουν υιοθετηθεί νέοι τύποι macroblocks στα ενισχυτικά χωρικά στρώματα. Ο διαχωρισμός των macroblocks γίνεται μετά από αύξηση του ρυθμού δειγματοληψίας του διαχωρισμού των 8x8 μπλοκς που είναι τοποθετημένα στις αντίστοιχες θέσεις στα στρώματα χαμηλότερης ευκρίνειας. Οι δείκτες των εικόνων αναφοράς αντιγράφονται από τις αντίστοιχες θέσεις των μπλοκ στο στρώμα βάσης, και τα αντίστοιχα διανύσματα κίνησης διπλασιάζονται. Επιπρόσθετα, ένα κλιμακωτό διάνυσμα κίνησης στρώματος χαμηλότερης ευκρίνειας μπορεί να χρησιμοποιηθεί για πρόβλεψη διανύσματος κίνησης των συμβατικών macroblocks.

Διαστρωματική Πρόβλεψη βάσει Κατάλοιπης Πληροφορίας

Η χρήση της πρόβλεψης αυτής γίνεται μόνο στην περίπτωση που αποστέλλεται η κατάλληλη σημαία(flag) για όλα τα δια-κωδικοποιημένα macroblocks. Σε τέτοια περίπτωση, γίνεται αύξηση του ρυθμού δειγματοληψίας του σήματος του στρώματος βάσης στα μπλοκ στις αντίστοιχες θέσεις και χρησιμοποιείται για πρόβλεψη για το κατάλοιπο σήμα του τρέχοντος macroblock, με αποτέλεσμα ουσιαστικά να κωδικοποιείται μόνο το αντίστοιχο σήμα διαφοράς.

Διαστρωματική Intra Πρόβλεψη

Ένας επιπλέον τύπος macroblock εισάγεται, στον οποίο το σήμα πρόβλεψης δημιουργείται μέσα από την αύξηση του ρυθμού κωδικοποίησης του ανοικοδομημένου σήματος που

βρίσκεται σε αντίστοιχη θέση σε κατώτερο στρώμα. Για να είναι δυνατή αυτή η κωδικοποίηση είναι απαραίτητη η πλήρης αποκωδικοποίηση περιλαμβανομένης και της πολύπλοκης υπολογιστικά λειτουργίας της αντιστάθμισης κίνησης. Ένας τρόπος παράκαμψης του προβλήματος αυτού, είναι με την διαστρωματική intra πρόβλεψη των μερών της εικόνας του κατώτερου στρώματος, η οποία είναι intra κωδικοποιημένη. Με αυτό τον περιορισμό, κάθε υποστηριζόμενο στρώμα μπορεί να αποκωδικοποιηθεί με ένα και μόνο επαναληπτικό βρόγχο αντιστάθμισης κίνησης.

Κλιμάκωση Ποιότητας

Στα πλαίσια του Scalable Video Coding, η κλιμάκωση όσο αφορά την ποιότητα της ροής που μεταδίδεται γίνεται μέσα από την αυξομείωση του Signal to Noise Ratio (SNR). Όσο μεγαλύτερος είναι ο λόγος του σήματος που εκπέμπεται προς τον θόρυβο που αλλοιώνει τα λαμβανόμενα δεδομένα, τόσο μεγαλύτερη θεωρείται η ποιότητα της ροής πολυμέσων που έχει μεταφερθεί από τον παραλήπτη προς τον αποστολέα.

Όσο αφορά αυτό το κριτήριο η κλιμάκωση των ροών βίντεο που προσφέρει το H.264 χωρίζονται σε Coarse-grain και fine-grain.

Coarse-Grain SNR Κλιμάκωση

Η κωδικοποίηση αυτή επιτυγχάνεται χρησιμοποιώντας τις ίδιες αρχές με την χωρική κλιμάκωση. Η μόνη διαφορά ανάμεσα στα δύο είναι η μη αναγκαιότητα για την αύξηση στον ρυθμο δειγματοληψίας (upsampling) στους διαστρωματικούς μηχανισμούς πρόβλεψης. Σημειώνεται ότι η περιορισμένη διαστρωματική πρόβλεψη που επιτρέπει αποκωδικοποίηση μέσω ενός και μόνο επαναληπτικού βρόγχου είναι ακόμα πιο σημαντική σε σχέση με την περίπτωση που χρησιμοποιείται για την χωρική κλιμακωτή κωδικοποίηση.

Fine-Grain SNR Κλιμάκωση

Για να είναι δυνατή η υποστήριξη αυτού τους είδους scalability, ήταν αναγκαία η εισαγωγή των προοδευτικά βελτιωμένων (Progressive Refinement- PR) slices. Κάθε ένα από αυτά αποτελεί μία βελτίωση του κατάλοιπου σήματος που αντιστοιχεί σε μια διχοτόμηση του μεγέθους κβαντοποίησης. Τα σήματα αυτά αναπαρίστανται σε μορφή ώστε να χρειάζεται

μόνο ένας αντίστροφος μετασχηματισμός για κάθε μετασχηματισμένο μπλοκ στην πλευρά του αποκωδικοποιητή.

Στον σχεδιασμό του SVC, χρησιμοποιείται η εικόνα αναφοράς με την ψηλότερη ποιότητα για την πρόβλεψη μέσω αντιστάθμισης κίνησης των υπόλοιπων εικόνων. Αυτό είναι αντίθετο με την τεχνική που πραγματεύεται το MPEG-4, και έχει σαν αποτέλεσμα την ενίσχυση της αποδοτικότητας χωρίς να αυξάνεται η πολυπλοκότητα όταν χρησιμοποιούνται ιεραρχικές δομές πρόβλεψης. Η αντιστάθμιση κίνησης για τις εικόνες αναφοράς γίνεται μόνο με χρήση των εικόνων αναφοράς στο στρώμα βάση και ως εκ τούτου μπορούν να χρησιμοποιηθούν σαν σημεία συγχρονισμού.

5.5.7.3 Προφίλ H.264

Το πρότυπο περιλαμβάνει τις ακόλουθες δυνατότητες, οι οποίες αναφέρονται σαν προφίλ και στοχεύουν σε συγκεκριμένες κατηγορίες εφαρμογών.

5.5.7.3.1 Baseline Profile (BP)

Το προφίλ αυτό προορίζεται για χαμηλού κόστους εφαρμογές, όπου υπάρχουν περιορισμένοι διαθέσιμοι υπολογιστικοί πόροι. Ήδη το προφίλ αυτό χρησιμοποιείται ευρέως σε εφαρμογές τηλεδιάσκεψης καθώς και σε κινητά τερματικά.

5.5.7.3.2 Main Profile (MP)

Αρχικά το προφίλ αυτό προοριζόταν για εφαρμογές εκπομπής τηλεοπτικών προγραμμάτων. Με την ευρύτερη τάση όμως για μετάβαση σε σήματα υψηλής ευκρίνειας, σταδιακά το προφίλ αυτό παραμερίστηκε και επικράτησε το μεταγενέστερο High Profile.

5.5.7.3.3 Extended Profile (XP)

Το προφίλ αυτό προορίζεται για εφαρμογές ροής βίντεο σε δίκτυα υπολογιστικών και κινητών τηλεπικοινωνιών, περιλαμβάνοντας υψηλή συμπίεση και ανοχή στα λάθη, καθώς και εναλλαγή των χαρακτηριστικών της ροής σε πραγματικό χρόνο.

5.5.7.3.4 High Profile (HiP)

Πρόκειται για το πρωταρχικό προφίλ για εφαρμογές εκπομπή ψηφιακού τηλεοπτικού σήματος υψηλής ευκρίνειας και αποθήκευσης αυτού σε οπτικά μέσα. Πρόκειται, εξάλλου, για το προφίλ που έχει υιοθετηθεί για το Blu-Ray Disc.

5.5.7.3.5 High 10 Profile (Hi10P)

Το προφίλ αυτό επεκτείνει το υπάρχον High Profile κάνοντας ένα βήμα πέρα από τις παρούσες δυνατότητες των καταναλωτικών προϊόντων. Η επέκταση αυτή προβλέπει την υποστήριξη μέχρι και 10 bits, αντί του συμβατικού μήκους των 8 bits, ανά δείγμα της αποκωδικοποιημένης εικόνας.

5.5.7.3.6 High 4:2:2 Profile (Hi422P)

Επεκτείνοντας επιπλέον το High 10 Profile, το προφίλ αυτό στοχεύει σε βίντεο πλεκτης σάρωσης, υποστηρίζοντας χρωματική δειγματοληψία 4:2:2 και μέχρι 10 bits ανά δείγμα της αποκωδικοποιημένης εικόνας.

5.5.7.3.7 High 4:4:4 Predictive Profile (Hi444PP)

Το προφίλ αυτό επεκτείνει ακόμη περισσότερο το Hi422P, προσθέτοντας υποστήριξη για χρωματική δειγματοληψία 4:4:4 και μέχρι 14 bits ανά δείγμα. Επιπλέον προβλέπει τη χρήση μη απωλεστικής κωδικοποίησης, καθώς και τη κωδικοποίηση του κάθε frame ξεχωριστά στα τρία χρωματικά πεδία.

Επιπροσθέτως, το πρότυπο προβλέπει και τέσσερα εξειδικευμένα προφίλ που περιλαμβάνουν μόνο κωδικοποίηση intra και ορίζονται ως υποσύνολα των προηγούμενων προφίλ. Τα εξειδικευμένα αυτά υποσύνολα προορίζονται για επαγγελματικές εφαρμογές και είναι τα ακόλουθα.

5.5.7.3.8 High 10 Intra Profile

Πρόκειται για το High 10 Profile με αποκλειστική χρήση Intra κωδικοποίησης.

5.5.7.3.9 High 4:2:2 Intra Profile

Πρόκειται για το High 4:2:2 Profile με αποκλειστική χρήση Intra κωδικοποίησης.

5.5.7.3.10 High 4:4:4 Intra Profile

Πρόκειται για το High 4:4:4 Profile με αποκλειστική χρήση Intra κωδικοποίησης.

5.5.7.3.11 CAVLC 4:4:4: Intra Profile

Πρόκειται για το High 4:4:4 Profile με αποκλειστική χρήση Intra κωδικοποίησης και μέθοδο CAVLC ως μέθοδο εντροπικής κωδικοποίησης(δηλαδή δεν υποστηρίζει CABAC).

Ως αποτέλεσμα της επέκτασης SVC, το πρότυπο περιλαμβάνει τρία επιπρόσθετα κλιμακωτά προφίλ, τα οποία ορίζονται σαν ένας συνδυασμός του H.264/AVC προφίλ για το στρώμα βάση και εργαλεία που επιτυγχάνουν την scalable επέκταση. Τα προφίλ τα οποία αναφέρονται πιο πάνω και συγκαταλέγονται στην επέκταση SVC, περιγράφονται συνοπτικά στη συνέχεια.

5.5.7.3.12 Scalable Baseline Profile

Κυρίως απευθύνεται σε εφαρμογές τηλεδιάσκεψης, ασύρματες συσκευές και εφαρμογές παρακολούθησης. Το προφίλ αυτό είναι επέκταση του Baseline Profile με ορισμένους περιορισμούς στους οποίους πρέπει να υπακούει το στρώμα βάση. Σχετικά με τα εργαλεία scalability, ένα υποσύνολό τους είναι ενεργοποιημένο.

5.5.7.3.13 Scalable High Profile

Βάση αυτού του προφίλ, το στρώμα βάση πρέπει να υπακούει στο προφίλ H.264/AVC High Profile. Κύριες εφαρμογές του είναι το streaming και το broadcast.

5.5.7.3.14 Scalable High Intra Profile

Πρόκειται για το Scalable High Profile αλλά με χρήση μόνο Intra κωδικοποίησης και στοχεύει κυρίως εφαρμογές παραγωγής.

ΚΕΦΑΛΑΙΟ 6

ΒΑΣΙΚΑ ΠΡΩΤΟΚΟΛΛΑ ΔΙΑΔΙΚΤΥΟΥ ΓΙΑ ΤΗ ΜΕΤΑΔΟΣΗ ΠΟΛΥΜΕΣΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ

6.1 Επίπεδα και Αρχιτεκτονική του Διαδικτύου

6.2 Επίπεδο Εφαρμογών

6.3 Επίπεδο Μεταφοράς

6.4 Επίπεδο Δικτύου

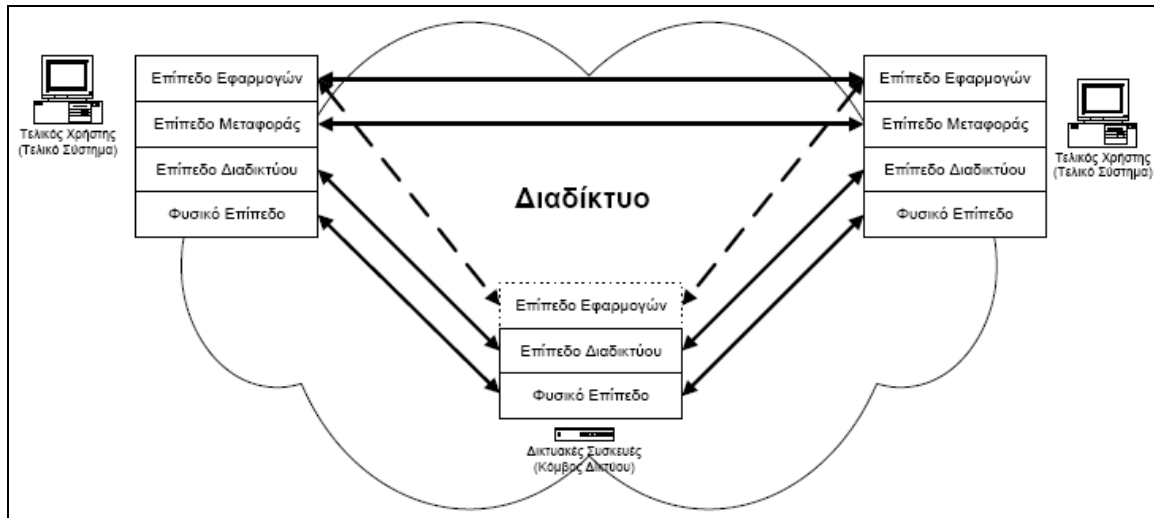
6.1 Επίπεδα και Αρχιτεκτονική του Διαδικτύου

Σε αντίθεση με τα κλασικά τηλεπικοινωνιακά δίκτυα των προηγούμενων δεκαετιών τα οποία χρησιμοποιούσαν τεχνικές μεταγωγής κυκλώματος (circuit switching), το Διαδίκτυο χρησιμοποιεί την τεχνική της μεταγωγής πακέτου (packet switching). Με τη χρήση της μεταγωγής κυκλώματος, κάθε ροή δεδομένων λαμβάνει συγκεκριμένο εύρος ζώνης το οποίο εξομοιώνει ένα δεσμευμένο φυσικό κύκλωμα ανάμεσα στον αποστολέα και τον παραλήπτη για αυτή την ροή δεδομένων. Αντίθετα με τη χρήση της μεταγωγής πακέτων τα πακέτα πολλών ροών δεδομένων πολυπλέκονται και ανταγωνίζονται για την εκμετάλλευση των διαθέσιμων δικτυακών πόρων. Επιπλέον, με την χρήση της μεταγωγής πακέτου, πακέτα που ανήκουν στην ίδια ροή δεδομένων μπορεί να ακολουθήσουν διαφορετικά μονοπάτια στο δίκτυο κατά την πορεία τους από αποστολέα προς παραλήπτη.

Στην αρχιτεκτονική του Διαδικτύου γίνεται διαχωρισμός ανάμεσα στα τελικά συστήματα και στους κόμβους του δικτύου. Το μοντέλο TCP/IP, όπως και άλλες σουίτες πρωτοκόλλων, μπορεί να παρατηρηθεί σαν ένα σύνολο επιπέδων, κάθε ένα εκ των οποίων είναι υπεύθυνο για την επίλυση προβλημάτων σχετικών με την μεταφορά των δεδομένων από ένα κόμβο σε άλλο. Συγκεκριμένα, το μοντέλο αυτό αποτελείται από 4 επίπεδα, το επίπεδο Εφαρμογών, επίπεδο Μεταφοράς, επίπεδο Διαδικτύου και Φυσικό επίπεδο (από το

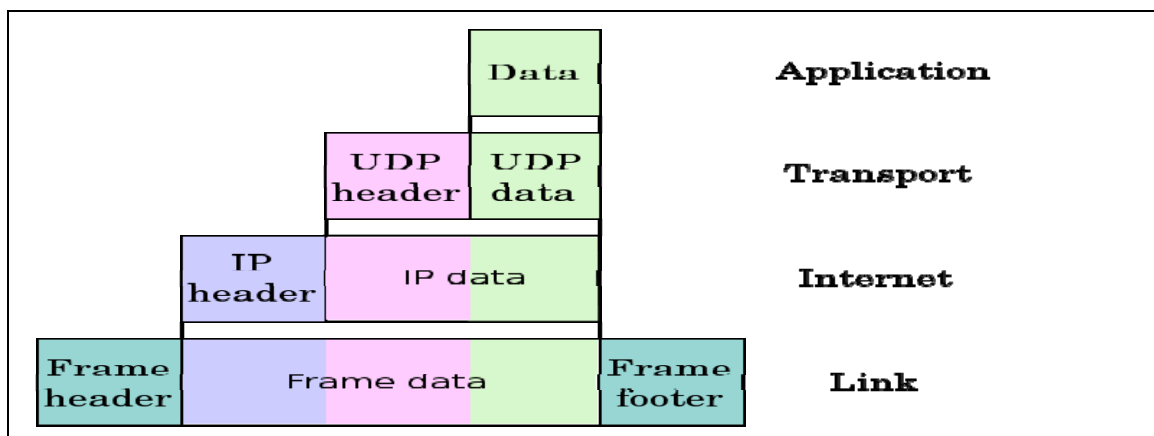
ψηλότερο στο χαμηλότερο επίπεδο). Κάθε επίπεδο ενισχύει την αφαιρετικότητα, και προσφέρει μια καλά καθορισμένη υπηρεσία στα ψηλότερα επίπεδα. Όσο πιο ψηλά βρίσκεται ένα επ΄εδο, τόσο πιο κοντά βρίσκεται στον χρήστη και βασίζεται στα χαμηλότερα επίπεδα να μετατρέψουν τα δεδομένα σε μορφές που να μπορούν να σταλούν μέσω του Δικτύου.

Σε ένα σενάριο χρήσης του Διαδικτύου (βλέπε Σχήμα 6.1), μια ροή δεδομένων μεταδίδεται από ένα τελικό σύστημα σε ένα άλλο διασχίζοντας ένα ή περισσότερους κόμβους του δικτύου. Σε γενικές γραμμές στα τελικά συστήματα εκτελούνται οι εφαρμογές του τελικού χρήστη οι οποίες παράγουν δεδομένα. Τα δεδομένα των εφαρμογών του τελικού χρήστη χωρίζονται σε πακέτα δεδομένων τα οποία μεταδίδονται στην διεύθυνση προορισμού τους χρησιμοποιώντας τη λειτουργικότητα που παρέχει το επίπεδο μεταφοράς του Διαδικτύου. Το επίπεδο του Διαδικτύου είναι υπεύθυνο για τη δρομολόγηση των πακέτων δεδομένων μέσα στο δίκτυο ώστε αυτά να φτάσουν στον τελικό προορισμό τους. Στους κόμβους του δικτύου τα εισερχόμενα πακέτα μπορεί να αποθηκεύονται τοπικά για κάποιο χρονικό διάστημα σε μνήμες προσωρινής αποθήκευσης (buffers), μπορεί να τυγχάνουν επεξεργασίας, μπορεί να προωθούνται σε κάποιον άλλο κόμβο του δικτύου ή να προωθούνται στο τελικό σύστημα που είναι ο προορισμός τους. Θα πρέπει να τονίσουμε πως δεν είναι απαραίτητο όλοι οι κόμβοι του δικτύου να υλοποιούν τις λειτουργικότητες όλων των επιπέδων του Διαδικτύου, για παράδειγμα μπορεί να υπάρχουν κόμβοι οι οποίοι απλά αντιγράφουν τα πακέτα τα οποία λαμβάνουν στην θύρα εισόδου τους, στην θύρα εξόδου τους στο φυσικό επίπεδο χωρίς να χρησιμοποιούν κάποια λειτουργία του επιπέδου Διαδικτύου.



Σχήμα 6-1: Αρχιτεκτονική και Επίπεδα Διαδικτύου στο Μοντέλο TCP/IP

Το TCP/IP χρησιμοποιεί ενθυλάκωση (encapsulation) για να προσφέρει αφαιρετικότητα τόσο δεδομένων, όσο και υπηρεσιών. Ουσιαστικά, μια εφαρμογή χρησιμοποιεί διάφορα πρωτόκολλα για να μεταφέρει τα δεδομένα της σε χαμηλότερα επίπεδα, σε κάθε ένα από τα οποία, τα δεδομένα αυτά ενθυλακώνονται περαιτέρω (Σχήμα 9) σε μηνύματα του εκάστοτε πρωτοκόλλου που χρησιμοποιείται σε κάθε επίπεδο. Από τα πιο πάνω, είναι προφανής η ανάγκη ύπαρξης των διαφόρων πρωτοκόλλων του Διαδικτύου σε όλα τα επίπεδα, αφού υποβοηθούν την λειτουργία του καθορίζοντας συγκεκριμένους τρόπους μεταφοράς, λήψης, επεξεργασίας και προβολής των δεδομένων στο Διαδίκτυο.



Σχήμα 6-2: Ενθυλάκωση Μηνυμάτων Διαδικτύου στα διάφορα επίπεδα

Στις ακόλουθες ενότητες θα αναφερθούν και θα επεξηγηθούν διάφορα πρωτόκολλα τα οποία σχετίζονται με την μεταφορά πολυμεσικού περιεχομένου μέσω του Διαδικτύου.

6.2 Επίπεδο Εφαρμογών

Το επίπεδο εφαρμογών βρίσκεται στην κορυφή της ιεραρχίας στη σουίτα TCP/IP και προσφέρει στον χρήστη τη δυνατότητα προσπέλασης στις πληροφορίες του Διαδικτύου, μέσω κάποιας εφαρμογής. Η επικοινωνία ανάμεσα σε εφαρμογές γίνεται από άκρο σε άκρο πράγμα που σημαίνει ότι μια εφαρμογή μεταδίδει τα δεδομένα της σε μια άλλη εφαρμογή χωρίς να απαιτείται η άμεση αλληλεπίδραση των εφαρμογών με τους κόμβους του δικτύου. Στις περισσότερες περιπτώσεις οι κόμβοι του δικτύου προωθούν και δρομολογούν τα δεδομένα στο Διαδίκτυο. Επιπλέον οι εφαρμογές μπορεί να χρησιμοποιούν πρωτόκολλα στο επίπεδο εφαρμογών τα οποία υλοποιούν συγκεκριμένες λειτουργίες και παρέχουν πληροφορίες για τα δεδομένα ή επιπλέον δυνατότητες για τη διαχείριση των δεδομένων. Αποτελεί την κύρια διασύνδεση του χρήστη με την εφαρμογή, και συνεπώς με το δίκτυο.

Πρωτόκολλα Επιπέδου Εφαρμογών

6.2.1 Πρωτόκολλο Session Initiation Protocol (SIP)

Χαρακτηριστικά

Το Πρωτόκολλο Εκκίνησης Συνόδου (Session Initiation Protocol- SIP) είναι ένα πρωτόκολλο σηματοδότησης του επιπέδου εφαρμογών για την αρχικοποίηση, την τροποποίηση και τον τερματισμό μιας διαδραστικής συνεδρίας μεταξύ χρηστών, η οποία συμπεριλαμβάνει στοιχεία πολυμέσων όπως βίντεο, φωνή, άμεση αποστολή και λήψη μηνυμάτων, τα διαδικτυακά παιχνίδια και η εικονική πραγματικότητα. Οι τροποποιήσεις που μπορούν να γίνουν σε μια τέτοια επικοινωνία είναι η αλλαγή διευθύνσεων ή ports, η πρόσκληση και άλλων συμμετεχόντων καθώς επίσης και η πρόσθεση ή αφαίρεση ροών πολυμέσων. Σε μια τέτοια συνεδρία είναι δυνατό να λαμβάνουν μέρος δύο ή περισσότερα μέρη, και η επικοινωνία μπορεί να γίνεται μέσω unicast ή multicast. Πιο συνηθισμένα χρησιμοποιείται για συστήματα VoIP (Voice over IP), αλλά τελευταία η χρήση του επεκτάθηκε στο ευρύ φάσμα των πολυμεσικών εφαρμογών που αναφέρθηκαν πιο πάνω. Το

SIP αναπτύσσεται από την Ομάδα Εργασίας SIP, η οποία εργάζεται στα πλαίσια του Internet Engineering Task Force (IETF).

Το πρωτόκολλο αυτό είναι σχεδιασμένο, ώστε να λειτουργεί ανεξάρτητα από το πρωτόκολλο του επιπέδου μεταφοράς, δηλαδή μπορεί να χρησιμοποιηθεί τόσο με το TCP, όσο και με το UDP, πράγμα που εξασφαλίζει μικρότερη καθυστέρηση στην εγκατάσταση της επικοινωνίας μεταξύ των χρηστών.. Εξίσου σημαντικό χαρακτηριστικό του, είναι το γεγονός ότι βασίζεται σε κείμενο (text- based), δηλαδή τα μηνύματα του παρουσιάζονται σε μορφή που επιτρέπουν την ανάγνωση και ανάλυσή τους από τον άνθρωπο.

Η δομή που ακολουθείται, είναι παρόμοια με αυτή του IP, καθιστώντας απλούστερες τις λειτουργίες που πρέπει να εκτελεστούν στο δίκτυο. Ουσιαστικά, ο πυρήνας του δικτύου είναι υπεύθυνος μόνο για τις ανταλλαγές των μηνυμάτων, γεγονός που περιορίζει την δέσμευση των πόρων (όπως μνήμη και υπολογιστική ισχύ), και παράλληλα προσφέρει υψηλή αξιοπιστία λόγω της κλιμακωτής μορφής.

Στοιχεία Δικτύου SIP

Οι συσκευές των τελικών χρηστών αποτελούν τους SIP User Agents (UAs), που χρησιμοποιούνται για την έναρξη και διαχείριση μιας SIP συνόδου. Αποτελούνται από δύο κύρια συστατικά, τον User Agent Client (UAC) που αποστέλλει μηνύματα και απαντά μέσω SIP αποκρίσεων και τον User Agent Server (UAS) που απαντά στις SIP αιτήσεις που αποστέλλονται από τον άλλο συνομιλητή. Οι User Agents μπορούν να επικοινωνήσουν και σε μορφή σημείου προς σημείο.

Το SIP εξάλλου καθορίζει και τους τύπους των αναγκαίων εξυπηρετητών δικτύου. Αν και δύο τερματικά που χρησιμοποιούν το πρωτόκολλο SIP μπορούν να επικοινωνήσουν και χωρίς καμία ενδιάμεση υποδομή, η προσέγγιση αυτή δεν είναι πρακτική στην περίπτωση που η υπηρεσία χρησιμοποιείται δημόσια. Οι εξυπηρετητές που καθορίζονται στο πρωτόκολλο είναι:

Proxy Server: Πρόκειται για μια ενδιάμεση οντότητα που μπορεί να λειτουργήσει τόσο σαν εξυπηρετητής, όσο και σαν client, με σκοπό να κάνει αιτήσεις εκ μέρους άλλων clients. Κυρίως χρησιμοποιείται για την δρομολόγηση, δηλαδή ο ρόλος του είναι να εξασφαλίσει ότι μια αίτηση έχει αποσταλεί σε ένα χρήστη που βρίσκεται πιο κοντά στον χρήστη- παραλήπτη. Ακόμη, οι εξυπηρετητές Proxy, είναι χρήσιμοι για την διασφάλιση της

σωστής εφαρμογής της πολιτικής που πρέπει να ακολουθείται, όπως για παράδειγμα έλεγχος αν ένας χρήστης δικαιούται να κάνει μια κλήση. Ανάμεσα στις λειτουργίες του, είναι επίσης να μεταφράζει και αν κριθεί αναγκαίο να αντιγράφει συγκεκριμένα μέρη του μηνύματος αίτησης πριν το προωθήσει.

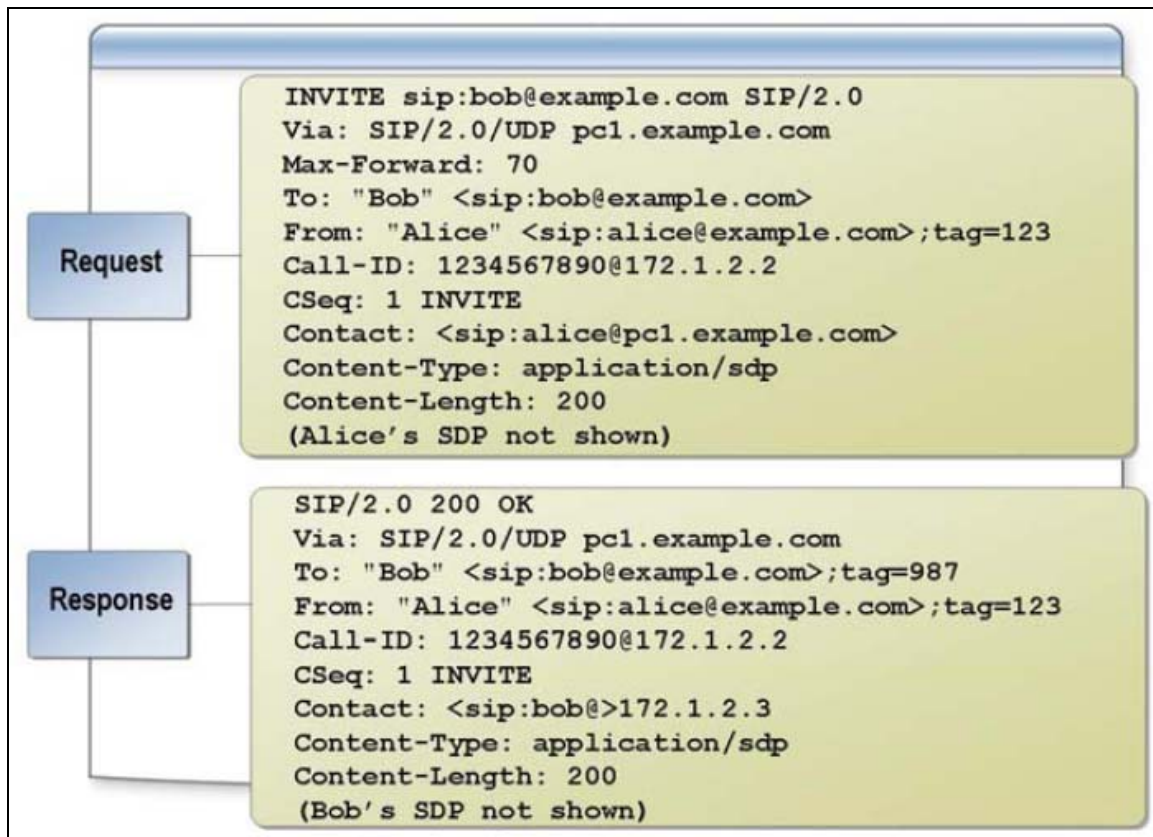
Registrar: Είναι εξυπηρετητής που λαμβάνει αιτήσεις REGISTER (εγγραφής) και αποθηκεύει πληροφορίες, που λαμβάνει μέσω αυτών, στον εξυπηρετητή της τοποθεσίας για το πεδίο (domain) που χειρίζεται.

Redirect Server: Είναι UASs που παράγουν αποκρίσεις της μορφής 3xx για τις αιτήσεις που λαμβάνει, κατευθύνοντας τον χρήστη να επικοινωνήσει με ένα εναλλακτικό σύνολο από URIs (Universal Resource Indicator). Επιτρέπει με αυτό τον τρόπο σε ένα Proxy Server να κατευθύνει τις προσκλήσεις για εκκίνηση συνόδου SIP και σε εξωτερικά domains.

Στο πρωτόκολλο, τονίζεται ότι η διάκριση αυτή ανάμεσα στα προαναφερθέντα διαφορετικά είδη εξυπηρετητών είναι λογική, παρά φυσική.

Σημαντικότερα Πεδία της Επικεφαλίδας ενός SIP Μηνύματος

Αν και η επικεφαλίδα ενός μηνύματος που αποστέλλεται στα πλαίσια μίας συνόδου SIP αποτελείται από πολλά πεδία, εντούτοις θα αναφερθούμε μόνο στα σημαντικότερα από αυτά. Ορισμένα πεδία είναι δυνατό να εμφανιστούν τόσο σε μηνύματα αιτήσεων, όσο και σε απαντήσεις και ονομάζονται γενικά πεδία επικεφαλίδας, σε αντίθεση με άλλα που μπορεί να εμφανιστούν είτε μόνο σε αιτήσεις, είτε μόνο σε αποκρίσεις.



Σχήμα 6-3: Παράδειγμα Μηνυμάτων SIP

Via: υπογραφή του User Agent Client και του εξυπηρετητή proxy, ώστε να δρομολογηθούν οι αποκρίσεις πίσω μέσω του ίδιου μονοπατιού.

From: η SIP διεύθυνση του αποστολέα του αιτήματος.

Contact: Εμφανίζεται στα αιτήματα INVITE / OPTIONS / ACK / REGISTER και στις απαντήσεις αυτών. Εμφανίζει την διεύθυνση των ενδιάμεσων παραληπτών, μέχρι τον τελικό αποδέκτη.

Priority: Καταδεικνύει το πόσο επείγει το συγκεκριμένο αίτημα

Content- Type: Ο τύπος του φορτίου που περιέχει το μήνυμα.

Content- Length: Είναι το μέγεθος του κυρίως σώματος ενός μηνύματος σε bytes.

CSeq: Αποτελείται από ένα αριθμό ακολουθίας και μία μέθοδο αιτήματος. Χρησιμοποιείται για τη σωστή αρίθμηση των συναλλαγών στα πλαίσια ενός διαλόγου, παρέχει τρόπο για μοναδική αναγνώριση των συναλλαγών και για την διαφοροποίηση ανάμεσα στις νέες αιτήσεις και τις αιτήσεις επαναποστολής (retransmission).

Μέθοδοι SIP

Το αρχικό πρωτόκολλο SIP, καθορίζει έξι μεθόδους SIP, που χρησιμοποιούνται για τους διάφορους τύπους αιτήσεων. Οι μέθοδοι αυτές αναφέρονται και περιγράφονται στη συνέχεια.

INVITE	Χρησιμοποιείται για την εκκίνηση μίας συνόδου, περιλαμβάνει πληροφορίες για τον χρήστη που καλεί και τον χρήστη που καλείται και επίσης για τον τύπο των δεδομένων που θα ανταλλαχθούν.
ACK	Αποστέλλεται από τον client που αποστέλλει και το INVITE. Χρησιμοποιείται για να επιβεβαιώσει ότι η σύνοδος έχει εκκινήσει. Ακολούθως είναι δυνατή η ανταλλαγή δεδομένων.
BYE	Τερματίζει μία σύνοδο και μπορεί να το αποστείλει οποιοσδήποτε χρήστης.
CANCEL	Μπορεί να χρησιμοποιηθεί για τον τερματισμό μιας αίτησης που εκκρεμεί. Αν όμως, η σύνοδος έχει ξεκινήσει, είναι αναγκαία η μέθοδος BYE, για να επιτευχθεί ο τερματισμός της.
OPTIONS	Ζητά πληροφορίες σχετικά με τις δυνατότητες του εξυπηρετητή ή άλλως συσκευών.
REGISTER	Χρησιμοποιείται από ένα client για να κάνει login και να εγγράψει την τρέχουσα διεύθυνσή του στον SIP εξυπηρετητή REGISTRAR.

Αποκρίσεις SIP

Ένα μήνυμα απόκρισης SIP αποτελείται από ένα τριψήφιο κωδικό που αντιστοιχεί στο αποτέλεσμα της αίτησης που προηγήθηκε. Επιπλέον, περιλαμβάνει και μία φραστική περιγραφή του αποτελέσματος, η οποία βοηθά στην αναγνώρισή του από τον άνθρωπο, σε αντίθεση με τον κωδικό που μεταφράζεται από λογισμικό. Οι τιμές των κωδικών έχουν πεδίο από 100 έως 699, με το πρώτο ψηφίο να καθορίζει την κλάση της απόκρισης.

Πιο κάτω αναγράφονται και περιγράφονται οι κλάσεις αποκρίσεων του πρωτοκόλλου SIP.

1xx: Provisional	Η αίτηση έχει παραληφθεί και εξακολουθεί να τυγχάνει επεξεργασίας. Για παράδειγμα 180 RINGING, το τηλέφωνο του παραλήπτη κουδουνίζει.
2xx: Success	Η ενέργεια λήφθηκε επιτυχώς, έγινε κατανοητή και αποδεκτή. Η κλάση αυτή αποτελείται μόνο από το 200 OK και το 202 ACCEPTED.
3xx: Redirection	Επιπλέον ενέργειες πρέπει να γίνουν, ώστε να ολοκληρωθεί η αίτηση. Για παράδειγμα 302 MOVED TEMPORARILY.
4xx: Client Error	Φανερώνει την ύπαρξη συντακτικού λάθους στο αίτημα, ή δεν μπορεί να ικανοποιηθεί σε αυτό τον εξυπηρετητή. Παράδειγμα της κλάσης είναι το 401 UNAUTHORIZED, αν ο client είναι υποχρεωμένος να παρέχει και στοιχεία για να δικαιούται πρόσβαση στον server.
5xx: Server Error	Ο εξυπηρετητής απέτυχε να φέρει εις πέρας μία έγκυρη αίτηση. Για παράδειγμα ο κωδικός 504 TIMEOUT, αν δεν επιτεύχθηκε η έναρξη συνόδου εντός των χρονικών πλαισίων που τέθηκαν λόγω προβλήματος στον εξυπηρετητή.
6xx: Global Failure	Η αίτηση δεν μπορεί να διεκπεραιωθεί από οποιοδήποτε εξυπηρετητή. Πρόκειται για μια νέα κλάση αποκρίσεων για το πρωτόκολλο SIP.

Παράδειγμα Τυπικής Συνόδου SIP



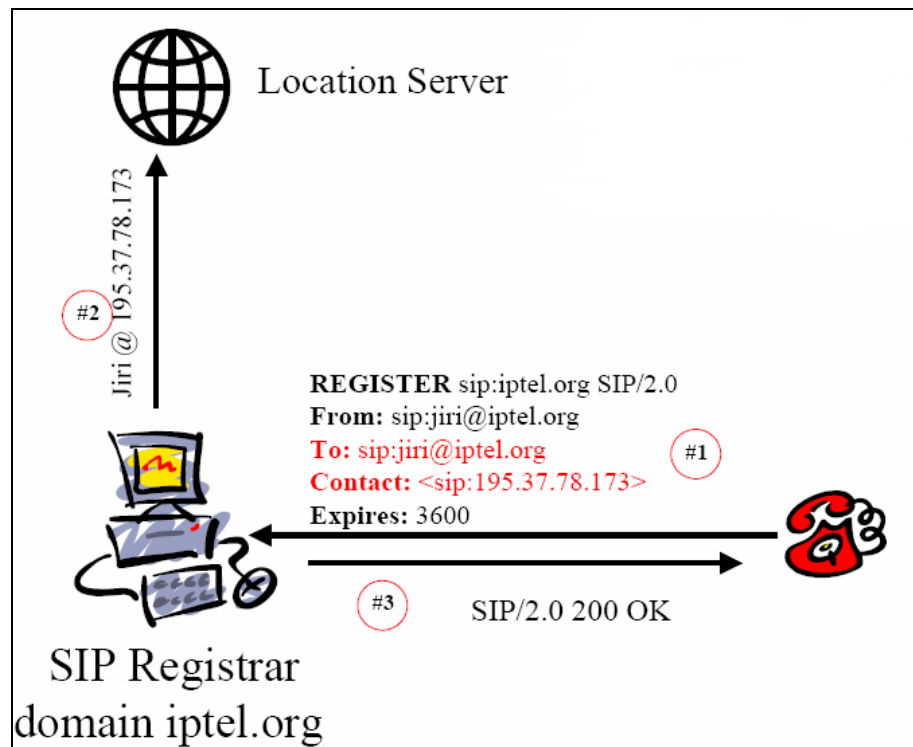
Σχήμα 6-4: Τυπική Σύνοδος SIP

Η διαδικασία ξεκινά με την αποστολή ενός μηνύματος INVITE από τον χρήστη που καλεί προς τον καλούμενο. Κατα την διάρκεια αναμονής της απάντησης, ο καλών είναι δυνατό να λαμβάνει διάφορα μηνύματα, όπως για παράδειγμα ότι ο καλούμενος έχει ειδοποιηθεί και το τηλέφωνό του χτυπά. Όταν ο καλούμενος αποδεκτεί την κλήση, αποστέλλεται στον καλών μήνυμα OK. Στη συνέχεια αυτός αποστέλλει το μήνυμα επιβεβαίωση έναρξης της επικοινωνίας ACK και μπορεί να ξεκινήσει η ανταλλαγή των δεδομένων από τα συμμετέχοντα μέρη. Όταν ένας από τους χρήστες τερματίσει την επικοινωνία, παράγεται ένα μήνυμα BYE και αποστέλλεται στον άλλο, ο οποίος με τη σειρά του επιβεβαιώνει τον τερματισμό της συνόδου με μήνυμα OK.

Παράδειγμα Διαδικασίας Εγγραφής Χρήστη

Σε μια τυπική σύνοδο SIP, ο χρήστης που κάνει την κλήση δεν γνωρίζει εξ αρχής την ακριβή διεύθυνση του παραλήπτη. Αντιθέτως, αυτό επιτυγχάνεται μέσω των proxy εξυπηρετητών. Ουσιαστικά, αυτό που συμβαίνει είναι η εγγραφή της τρέχουσας τοποθεσίας κάθε χρήστη σε ένα REGISTRAR. Αυτό γίνεται μέσω της αποστολής μηνύματος Register από την εφαρμογή, ενημερώνοντας τον εξυπηρετητή για την παρούσα τοποθεσία της. Ο Registrar αποθηκεύει αυτό το δέσιμο (ανάμεσα στον χρήστη και την

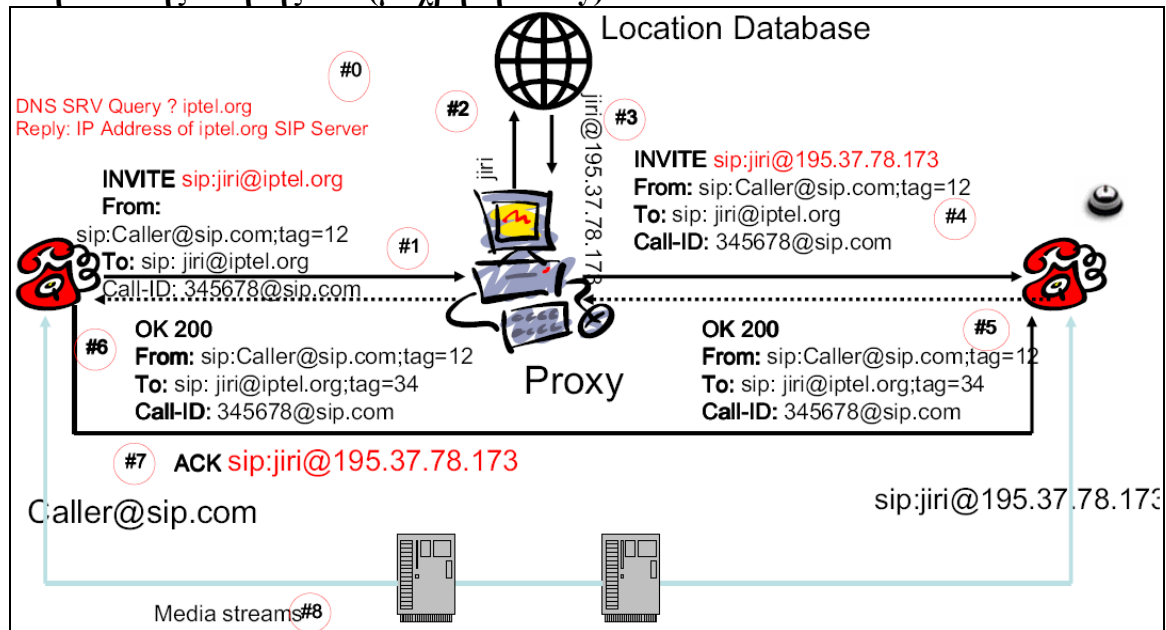
τρέχουσα διεύθυνσή του) σε ένα location server που χρησιμοποιείται και από άλλα proxies, ούτως ώστε να είναι δυνατή η ανίχνευση του χρήστη.



Σχήμα 6-5: Διαδικασία Εγγραφής Χρήστη (REGISTER)

Ο τελικός χρήστης με την είσοδό του, ενημερώνει τον Registrar μέσω μιας εγγραφής που περιέχει το νέο IP που αντιστοιχεί στην τρέχουσα τοποθεσία του, όπως αυτό φαίνεται στο πεδίο Contact. Με την σειρά του ο εξυπηρετητής αυτός ενημερώνει τον location server, ώστε να είναι δυνατή η επικοινωνία με τον χρήστη αυτό και μέσω άλλων εξωτερικών proxies. Το πεδίο Expires δηλώνει την διάρκεια για την οποία ισχύει η εγγραφή αυτή. Στο πιο πάνω παράδειγμα δηλαδή, η διεύθυνση που αναγράφεται στο πεδίο To θα παραμείνει δεμένη με την τρέχουσα IP διεύθυνση που φαίνεται στο πεδίο Contact για μία ώρα.

Ροή Τυπικής Κλήσης SIP (με χρήση Proxy)



Σχήμα 6-6: Ροή Τυπικής Κλήσης SIP (χρήση Proxy)

Το INVITE μήνυμα του αποστολέα, καταφθάνει στον proxy server, που εκτελεί ερώτημα στην location βάση δεδομένων, ώστε μέσω της διεύθυνσης του παραλήπτη, να μπορέσει να λάβει σαν απάντηση την τρέχουσα IP διεύθυνσή του. Ακολουθώντας, και πάλι ο proxy server, προωθεί το μήνυμα INVITE στον παραλήπτη στην τρέχουσα τοποθεσία του. Ο παραλήπτης απαντά με OK, που μέσω του proxy, παραδίδεται στο μέρος της συνόδου που την ξεκίνησε. Ο αρχικός αποστολέας, αποστέλλει στη συνέχεια μήνυμα επιβεβαίωσης έναρξης της συνομιλίας ACK στον παραλήπτη, αυτή όμως την φορά απευθείας, αφού πλέον γνωρίζει την IP διεύθυνση που αντιστοιχεί σε αυτόν. Έπειτα από το σημείο αυτό, η ανταλλαγή των ροών δεδομένων γίνεται μεταξύ των συμμετεχόντων στην σύνοδο απευθείας, χωρίς την χρήση του proxy server.

6.2.2 Πρωτόκολλο Real Time Streaming Protocol (RTSP)

Χαρακτηριστικά

Το Πρωτόκολλο Ροής Πραγματικού Χρόνου είναι ένα πρωτόκολλο του επιπέδου εφαρμογών το οποίο παρέχει μηχανισμούς για την υποστήριξη streaming πολυμέσων σε εφαρμογές πολλών σημείων, χρησιμοποιώντας τεχνολογίες μετάδοσης unicast και multicast. Ουσιαστικά ελέγχει την παράδοση των ροών μέσω, πάνω από unicast ή multicast δίκτυα με χαρακτηριστικά πραγματικού χρόνου. Η παρουσίαση μιας πολυμεσικής ροής δεδομένων αναγνωρίζεται μέσω ενός RTSP URL. Παρέχει στον χρήστη μηχανισμούς που επιτρέπουν την εκτέλεση εντολών τύπου τηλεχειριστηρίου προς τον εξυπηρετητή που μεταδίδει τα streaming μέσα, όπως πάγωμα, προώθηση προς τα εμπρός, μεταφορά προς τα πίσω και επιλογή συγκεκριμένης θέσης. Η μετάδοση των δεδομένων δεν γίνεται από το RTSP, αλλά από πρωτόκολλα του επιπέδου μεταφοράς από τα οποία και είναι ανεξάρτητο. Όσο αφορά τη σύνταξη και τη λειτουργία του είναι παρόμοιο με το HTTP 1.1, με τη διαφορά ότι το RTSP έχει καταστάσεις. Επιπλέον διαφορά είναι το γεγονός ότι σε αυτό το πρωτόκολλο είναι δυνατή η πραγματοποίηση αιτήσεων και από τους εξυπηρετητές και από τους παραλήπτες.

Σημαντικό χαρακτηριστικό του, είναι ότι έχει την δυνατότητα να υποστηρίξει πολλαπλούς εξυπηρετητές. Δηλαδή, ο παραλήπτης μπορεί να δέχεται μια ροή δεδομένων μέσω πολλαπλών συνόδων με πολλούς εξυπηρετητές ταυτόχρονα. Επιπλέον, ο έλεγχος ροής στη μετάδοση των δεδομένων καθοδηγείται από τον παραλήπτη.

Το πρωτόκολλο αυτό χρησιμοποιείται τόσο για την μετάδοση πραγματικού χρόνου αποθηκευμένων πολυμέσων, όσο και για την μετάδοση πραγματικού χρόνου ζωντανών πολυμέσων. Πέραν αυτών όμως, βρίσκει χρήση και στην μετάδοση μη πραγματικού χρόνου αποθηκευμένων πολυμέσων.

Παρουσίαση και Περιγραφή Παρουσίασης

Όπως αναφέρθηκε, οι παρουσιάσεις πολυμέσων αναγνωρίζονται από τις rtsp διευθύνσεις τους. Το hostname καθορίζει τον εξυπηρετητή στον οποίο βρίσκονται τα μέσα, ενώ το port δηλώνει σε ποιο port πρέπει να αποστέλλονται οι διάφορες αιτήσεις.

Παρουσίαση (Presentation) είναι μια ολοκληρωμένη ομάδα από ροές δεδομένων που συνιστά μια αυτοτελή οντότητα πολυμεσικής πληροφορίας, για παράδειγμα μια κινηματογραφική ταινία.

Η Περιγραφή Παρουσίασης (Presentation Description) περιέχει πληροφορία, για μία ή περισσότερες ροές δεδομένων πολυμέσων μέσα σε μια παρουσίαση, που περιλαμβάνει τους τρόπους κωδικοποίησης, τις διευθύνσεις στο δίκτυο και δεδομένα σχετικά με το περιεχόμενο.

Μηνύματα RTSP

Υπάρχουν δύο είδη μηνυμάτων στο RTSP. Οι αιτήσεις (request) και οι αποκρίσεις (response). Είναι και οι δύο τύποι text based κωδικοποιημένοι σε UTF-8 μορφή. Η πρώτη γραμμή κάθε μηνύματος καθορίζει τον τύπο του μηνύματος και καθορίζει συγκεκριμένα το είδος της αίτησης ή αποκρίσης για την οποία πρόκειται. Ακολούθως εμφανίζεται η επικεφαλίδα που παρέχει επιπλέον πληροφορίες, αναγκαίες για την ορθή διερμηνεία του μηνύματος. Τέλος, είναι δυνατό τα μηνύματα να περιέχουν και ένα κυρίως μέρος.

Οι αιτήσεις αποτελούνται ουσιαστικά από μεθόδους, οι οποίες περιγράφουν το τι ο client ζητά από τον εξυπηρετητή σχετικά με μια δεδομένη ροή και η απόκριση είναι η κατάσταση με στην οποία βρίσκεται η αίτηση.

Μέθοδοι RTSP

Οι μέθοδοι του πρωτοκόλλου RTSP που χρησιμοποιούνται για τους διάφορους τύπους αιτήσεων αναφέρονται και περιγράφονται συνοπτικά στην συνέχεια.

OPTIONS

Ο παραλήπτης ή ο εξυπηρετητής λέει στην άλλη πλευρά τις παραμέτρους που μπορεί να δεχτεί.

SETUP

Ο παραλήπτης ζητάει από τον εξυπηρετητή να δεσμεύσει

	πόρους για ροή δεδομένων, και να ξεκινήσει μια RTSP σύνοδο.
ANNOUNCE	Όταν στέλνεται από τον παραλήπτη στον εξυπηρετητή, δηλώνει την περιγραφή μιας παρουσίασης ενός πολυμεσικού αντικειμένου. Όταν στέλνεται από τον εξυπηρετητή στον παραλήπτη, ενημερώνει την περιγραφή σε πραγματικό χρόνο.
DESCRIBE	Ο παραλήπτης διαβάζει την περιγραφή μιας παρουσίασης ενός πολυμεσικού αντικειμένου.
PLAY	Ο παραλήπτης ζητάει από τον εξυπηρετητή να ξεκινήσει να στέλνει δεδομένα.
RECORD	Ο παραλήπτης ξεκινάει να αποθηκεύει ένα κομμάτι των πολυμεσικών δεδομένων σύμφωνα με την περιγραφή παρουσίασης.
REDIRECT	Ο εξυπηρετητής ενημερώνει τους παραλήπτες ότι πρέπει να συνδεθούν με ένα εξυπηρετητή σε μία άλλη τοποθεσία.
PAUSE	Ο παραλήπτης προσωρινά σταματάει την ροή δεδομένων χωρίς να ελευθερώνει τους δεσμευμένους πόρους γι' αυτή τη ροή δεδομένων στον εξυπηρετητή. Διατηρεί την κατάσταση.
GET_PARAMETER	Διαβάζει την τιμή μιας παραμέτρου για την παρουσίαση μίας ροής δεδομένων.
SET_PARAMETER	Θέτει την τιμή μιας παραμέτρου για την παρουσίαση μίας ροής δεδομένων.
TEARDOWN	Ο παραλήπτης ζητάει από τον εξυπηρετητή να σταματήσει την αποστολή μίας ροής δεδομένων και να ελευθερώσει τους δεσμευμένους πόρους γι' αυτή τη ροή δεδομένων. Αφαίρεση της κατάστασης.

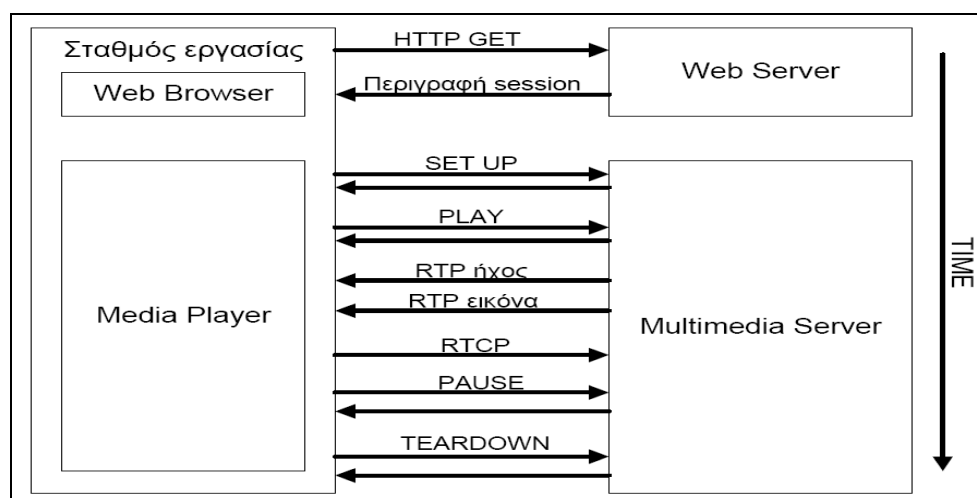
Αποκρίσεις RTSP

Ένα μήνυμα απόκρισης RTSP αποτελείται από ένα τριψήφιο κωδικό που αντιστοιχεί στο αποτέλεσμα της αίτησης που προηγήθηκε. Επιπλέον, περιλαμβάνει και μία φραστική περιγραφή του αποτελέσματος. Οι τιμές των κωδικών έχουν πεδίο από 200 έως 599, με το πρώτο ψηφίο να καθορίζει την κλάση της απόκρισης.

Πιο κάτω αναγράφονται και περιγράφονται οι κλάσεις αποκρίσεων του πρωτοκόλλου RSTP.

2xx: Success	Η ενέργεια λήφθηκε επιτυχώς, έγινε κατανοητή και αποδεκτή.
3xx: Redirection	Επιπλέον ενέργειες πρέπει να γίνουν, ώστε να ολοκληρωθεί η αίτηση.
4xx: Client Error	Φανερώνει την ύπαρξη συντακτικού λάθους στο αίτημα, ή δεν μπορεί να ικανοποιηθεί σε αυτό τον εξυπηρετητή.
5xx: Server Error	Ο εξυπηρετητής απέτυχε να φέρει εις πέρας μία έγκυρη αίτηση.

Παράδειγμα Τυπικής Συνόδου RTSP



Σχήμα 6-7: Τυπική Σύνοδος RTSP

Αρχικά ο φυλλομετρητής θα εκδόσει και θα αποστείλει στον εξυπηρετητή ένα μήνυμα GET, ώστε να ανοίξει η επιθυμητή σελίδα. Στη συνέχεια, ο web server θα απαντήσει με ένα μήνυμα το οποίο θα περιέχει την περιγραφή της παρουσίασης που υπάρχει σε αυτή την σελίδα. Ακολούθως, ο web browser θα φορτώσει τον κατάλληλο media player που είναι υπεύθυνος για αναπαραγωγή αυτών των ροών, ο οποίος και θα επικοινωνήσει με τον εξυπηρετητή στον οποίο είναι αποθηκευμένα τα πολυμέσα. Αρχικά θα αποστείλει μήνυμα με την μέθοδο SETUP, ώστε να δεσμευτούν πόροι και να εκκινήσει η σύννοδος μεταξύ των δύο επικοινωνούντων μερών. Αν η απάντηση στην αίτηση αυτή είναι καταφατική, ο παραλήπτης των ροών θα αποστείλει στον εξυπηρετητή αίτηση PLAY, ώστε να αρχίσει η λήψη και αναπαραγωγή των ροών πολυμέσων. Όπως είναι γνωστό το RTSP δεν είναι υπεύθυνο και για τη μεταφορά των πολυμέσων. Αντίθετα, η μεταφορά ήχου και εικόνας είναι αρμοδιότητα του πρωτοκόλλου RTP. Τα πακέτα ελέγχου που πιθανό να πρέπει να αποσταλούν από τον client προς τον εξυπηρετητή, μεταφέρονται με τη χρήση του πρωτοκόλλου, RTCP, ώστε να στέλνονται ανεξάρτητα των πακέτων που αφορούν τις ροές των πολυμέσων. Οι πληροφορίες που λαμβάνονται από αυτού του είδους την επικοινωνία αφορούν τον αριθμό των πακέτων που έχουν παραλειφθεί, την απώλεια στο δίκτυο και άλλα στατιστικά, τα οποία είναι δυνατό να χρησιμοποιηθούν ώστε να μεταβληθεί ανάλογα η ροή, είτε για βελτίωση της ποιότητας, είτε για μείωση των απωλειών. Κατά τη διάρκεια της συνόδου, είναι δυνατή η αποστολή διαφόρων μηνυμάτων από τον παραλήπτη προς τον server, όπως για παράδειγμα PAUSE ή RECORD. Σε κάθε περίπτωση για κάθε αίτηση, ο εξυπηρετητής απαντά με ένα μήνυμα που να δηλώνει την κατάσταση στην οποία η αίτηση αυτή βρίσκεται. Μία αίτηση είναι δυνατό να επεξεργάζεται ακόμη, ή να έχει ήδη επεξεργαστεί και το μήνυμα να είναι επιτυχία, ή κάποιο μήνυμα σφάλματος. Στην περίπτωση της μεθόδου, PAUSE αξίζει να αναφερθεί ότι η κατάσταση διατηρείται, μπορεί δηλαδή να επανεκκινήσει από το ίδιο σημείο στο οποίο σταμάτησε η αναπαραγωγή, ενώ παράλληλα οι πόροι που δεσμεύτηκαν για την ροή αυτή παραμένουν δεσμευμένοι. Για τη διακοπή της συνόδου, απαιτείται η αποστολή ενός μηνύματος που να περιέχει την μέθοδο TEARDOWN, ώστε να αποδεσμευτούν και οι πόροι που είχαν δεσμευτεί για τη μεταφορά της ροής πολυμέσων και συνεπώς η μεταφορά αυτή να σταματήσει. Προφανώς, με την εντολή αυτή, αφαιρείται και η οποιαδήποτε κατάσταση σχετικά με το πρωτόκολλο.

6.2.3 Πρωτόκολλο Real- Time Transport Protocol (RTP)

Εισαγωγή

Το πρωτόκολλο RTP παρέχει ένα πρότυπο για τον τύπο πακέτων που χρησιμοποιούνται για την μεταφορά δεδομένων ήχου ή και εικόνας στο Διαδίκτυο. Για την μεταφορά των δεδομένων αυτών, τα πακέτα στα οποία θα ενθυλακώνονται θα έπρεπε να περιέχουν πεδία που να εξυπηρετούν τέτοιες εφαρμογές, όπως αριθμό ακολουθίας και timestamp. Αναπτύχθηκε από την Audio- Video Transport Working Group του IETF, με την πρώτη έκδοση να χρονολογείται το 1996. Από τότε χρησιμοποιείται ευρέως σε συστήματα επικοινωνίας και διασκέδασης, όπως video streaming, τηλεφωνία, τηλεδιασκέψεις και συστήματα push to talk. Μεταφέρει ροές δεδομένων των πρωτοκόλλων σηματοδότησης, όπως H.323 και SIP, πράγμα που καθιστά το RTP τεχνικά θεμελιώδες για εφαρμογές όπως το VoIP.

Βασικά Χαρακτηριστικά

Τα πακέτα RTP συνήθως μεταφέρονται μέσω του πρωτοκόλλου UDP. Ο αποστολέας ενθυλακώνει τα δεδομένα πολυμέσων σε ένα RTP πακέτο, στη συνέχεια αυτό με τη σειρά του ενθυλακώνεται σε ένα UDP segment και ακολούθως μεταφέρεται στο IP. Η αντίστροφη διαδικασία ακολουθείται στον παραλήπτη, με τα τελικά δεδομένα να μεταφέρονται στον media player για αναπαραγωγή.

Αν δυο εφαρμογές χρησιμοποιούν RTP, παρά κάποιο ιδιωτικό πρωτόκολλο για μεταφορά δεδομένων πολυμέσων, τότε είναι πολύ πιο πιθανά εφικτή η δυνατότητα συνεργασίας και η λειτουργικότητα με μία άλλη εφαρμογή που επίσης χρησιμοποιεί RTP.

Πρέπει να σημειωθεί ότι το RTP δεν παρέχει οποιεσδήποτε εγγυήσεις σχετικά με την Ποιότητα Υπηρεσίας που παρέχεται, ούτε για την έγκαιρη παράδοση των πακέτων. Δεν εγγυάται καν την παράδοση των πακέτων, ούτε την σε σειρά μεταφορά τους από αποστολέα σε παραλήπτη. Αντίθετα, η ενθυλάκωση που συμβαίνει στα πλαίσια του RTP είναι ορατή μόνο από τα τερματικά, ενώ ούτε οι δρομολογητές είναι σε θέση να ξεχωρίσουν ανάμεσα σε IP πακέτα που μεταφέρουν RTP και κανονικά IP πακέτα.

Επιπλέον χαρακτηριστικό του RTP είναι η δυνατότητα που παρέχει σε κάθε πηγή, να της ανατίθεται η δική της ανεξάρτητη RTP ροή πακέτων. Για παράδειγμα, για μία τηλεδιάσκεψη, είναι πιθανό να δημιουργηθούν δύο ροές ήχου (μία προς κάθε κατεύθυνση)

και δύο ροές εικόνας (και πάλι μία προς κάθε κατεύθυνση). Εντούτοις, πολλές σύγχρονες τεχνικές κωδικοποίησης δένουν ήχο και εικόνα κατά την κωδικοποίηση και έτσι είναι αναγκαία η δημιουργία μίας μόνο RTP ροής προς κάθε κατεύθυνση.

Το πρωτόκολλο αυτό μπορεί να χρησιμοποιηθεί και πέραν του unicast, σε περιβάλλοντα multicast είτε με ένα και μόνο αποστολέα, είτε με πολλούς αποστολείς. Για μία σύνοδο multicast many-to-many, όλοι οι αποστολείς χρησιμοποιούν την ίδια διεύθυνση multicast group για αποστολή των πακέτων RTP.

Επικεφαλίδα Πακέτων RTP

Η επικεφαλίδα ενός πακέτου RTP αποτελείται από πολλά πεδία, με τέσσερα από αυτά να είναι τα πιο σημαντικά. Αυτά είναι ο τύπος payload που μεταφέρει το τρέχον πακέτο, ο αριθμός ακολουθίας, το timestamp και το πεδίο source identifier (αναγνωριστικό πηγής).

Το πεδίο τύπος payload αποτελείται από 7 bits. Ανάλογα με το αν μεταφέρεται ήχος ή εικόνα, το πεδίο αυτό καθορίζει τον τύπο κωδικοποίησης που χρησιμοποιείται. Εξάλλου, αν ο αποστολέας αποφασίσει εν μέσω της μεταφοράς να αλλάξει κωδικοποίηση μπορεί να ενημερώσει τον παραλήπτη για αυτή την αλλαγή, μέσω του πεδίου αυτού. Αυτό είναι πιθανό να συμβεί αν ο αποστολέας επιθυμεί να βελτιώσει την ποιότητα των πολυμέσων που αποστέλλονται, ή αν μέσω καποιου μηχανισμού αντιληφθεί ότι το διαθέσιμο εύρος ζώνης έχει μειωθεί και ως εκ τούτου πρέπει να μειωθεί και το bitrate της ροής που αποστέλλεται.

Το πεδίο αριθμός ακολουθίας έχει μέγεθος 16 bits. Ο αριθμός αυτός αυξάνεται κατά ένα κάθε φορά που γίνεται μία νέα αποστολή RTP πακέτου, και μπορεί να χρησιμοποιηθεί από τον παραλήπτη για να ανιχνεύσει χαμένα πακέτα, ή να διορθώσει την σειρά των πακέτων. Για παράδειγμα, λαμβάνοντας δύο διαδοχικά πακέτα που διαφέρουν μεταξύ τους κατά δύο αριθμούς ακολουθίας, ο παραλήπτης αντιλαμβάνεται την απώλεια του ενός πακέτου και προσπαθεί να επικαλύψει το σφάλμα αυτό.

Το timestamp αποτελείται από 32 bits. Αντιπροσωπεύει την στιγμή που έγινε η δειγματοληψία του πρώτου byte που περιέχεται στο τρέχον πακέτο. Μπορεί να χρησιμοποιηθεί για την διόρθωση της διακύμανσης καθυστέρησης και να προσφέρει συγχρονισμένη αναπαραγωγή στον παραλήπτη. Ο χρόνος του timestamp αυξάνεται κατά

ένα κάθε περίοδο δειγματοληψίας, και συνεχίζει να αυξάνεται ακόμα και όταν η πηγή είναι ανενεργή.

Το πεδίο SSRC (Synchronization Source Identifier) αποτελείται από 32 bits. Χρησιμοποιείται για την αναγνώριση των RTP ροών. Συνήθως σε κάθε σύνοδο RTP, ανατίθεται διαφορετικός αριθμός SSRC σε κάθε πηγή. Δεν είναι το ίδιο με την IP διεύθυνση, αλλά ένας τυχαίος αριθμός. Ωστόσο, και πάλι η πιθανότητα επιλογής του ίδιου SSRC από δύο πηγές είναι πολύ μικρή, αν συμβεί, όμως, οι δύο πηγές επιλέγουν νέους αριθμούς SSRC.

6.2.4 Πρωτόκολλο RTP Control Protocol (RTCP)

Εισαγωγή

Το πρωτόκολλο αυτό έχει οριστεί ώστε να χρησιμοποιείται σε συνδυασμό με το πρωτόκολλο RTP. Ουσιαστικά, το RTP είναι το πρωτόκολλο για μεταφορά των δεδομένων, ενώ το RTCP το πρωτόκολλο για τη μεταφορά των πακέτων ελέγχου. Τα πακέτα RTP και RTCP χρησιμοποιούν διαφορετικούς αριθμούς port, ώστε να μην προκαλείται παρεμβολή, συγκεκριμένα για το RTCP χρησιμοποιείται πάντα ο επόμενος αριθμός port που χρησιμοποιείται από την αντίστοιχη ροή RTP.

Τα πακέτα RTCP δεν χρησιμοποιούνται για ενθουλάκωση δεδομένων ήχου ή εικόνας, αλλά αποστέλλονται περιοδικά και περιέχουν αναφορές είτε του αποστολέα είτε του παραλήπτη με στατιστικά που είναι χρήσιμα για την εφαρμογή. Αυτά τα στατιστικά περιλαμβάνουν αριθμό πακέτων που έχουν σταλεί, αριθμό χαμένων πακέτων και διακύμανση καθυστέρησης. Δεν καθορίζεται, βάση του πρωτοκόλλου, πώς οι εφαρμογές πρέπει να χειρίζονται τις αναφορές αυτές, ως εκ τούτου η ανατροφοδότηση αυτή είναι στα χέρια του προγραμματιστή να την διαχειριστεί όπως αυτός επιθυμεί. Για παράδειγμα, ο αποστολέας μπορεί να χρησιμοποιήσει τα στατιστικά για να ρυθμίσει τον ρυθμό μετάδοσης δεδομένων, ενώ ο παραλήπτης για διαγνωστικούς σκοπούς, αν δηλαδή προβλήματα που παρατηρούνται είναι τοπικά ή μεγαλύτερης κλίμακας.

Τύποι Πακέτων RTCP

Receiver Reports

Για κάθε ρεύμα RTP για το οποίο ο παραλήπτης λαμβάνει στα πλαίσια μίας συνόδου, δημιουργεί μία αναφορά παραλήπτη. Ο αποστολέας συγκεντρώνει αυτές του τις αναφορές του σε ένα πακέτο RTCP. Ακολουθώντας αυτό αποστέλλεται στον αποστολέα, ή σε περίπτωση multicast στο δέντρο που χρησιμοποιείται.

Τα πιο σημαντικά από τα πεδία που παρατηρούνται σε ένα πακέτο RTCP είναι αυτά που περιγράφονται στη συνέχεια.

Το SSRC, είναι το αναγνωριστικό της ροής RTP για την οποία έχει εκδοθεί η συγκεκριμένη αναφορά.

Ο λόγος των χαμένων πακέτων εντός μιας ροής RTP. Κάθε παραλήπτης υπολογίζει τον αριθμό των πακέτων που χάθηκαν διά τον αριθμό των πακέτων που αποστάληκαν ως μέρος της ροής. Αν ένα αποστολέας παραλάβει αναφορές παραλήπτη που καταδεικνύουν ότι λαμβάνεται μόνο ένα μικρό ποσοστό των πακέτων που αποστέλλονται, μπορεί να μεταβεί σε χαμηλότερο ρυθμό κωδικοποίησης των πολυμέσων, με στόχο να ξεπεραστεί η συμφόρηση στο δίκτυο και να βελτιωθεί ο ρυθμός λήψης δεδομένων.

Ο τελευταίος αριθμός ακολουθίας πακέτου RTP, στην ροή από RTP πακέτα.

Η διακύμανση καθυστέρησης, που πρόκειται για ένα υπολογισμό της διαφοράς στον χρόνο παραλαβής μεταξύ διαδοχικών πακέτων στη ροή RTP.

Sender Reports

Για κάθε RTP ροή που μεταδίδει ένας αποστολέας, δημιουργεί και μεταδίδει παράλληλα και αναφορές αποστολέα RTCP. Τα πακέτα αυτά αποτελούνται από πληροφορίες σχετικά με την RTP ροή. Μερικά από αυτά τα στοιχεία είναι τα ακόλουθα.

Το SSRC αναγνωριστικό της ροής για την οποία δημιουργείται το πακέτο.

Το timestamp και ο πραγματικός χρόνος του πιο πρόσφατα δημιουργημένου πακέτου RTP στη ροή.

Ο αριθμός των πακέτων που έχουν αποσταλεί.

Ο αριθμός από bytes που έχουν αποσταλεί στα πλαίσια της ροής.

Τα sender reports έχουν άλλη μια ιδιότητα, που βασίζεται στα πεδία των πακέτων τους. Μπορούν να χρησιμοποιηθούν για τον συγχρονισμό των διαφόρων ροών πολυμέσων, εντός της ίδιας συνόδου RTP. Για παράδειγμα, σε περίπτωση ύπαρξης διαφορετικών ροών για ήχο και εικόνα σε μια τηλεδιάσκεψη, τα timestamps στα RTP πακέτα ήχου και εικόνας απλά αναφέρονται στην χρονική στιγμή δειγματοληψίας των δεδομένων. Κάθε sender report, όμως, περιέχει και την πραγματική ώρα και το timestamp του τελευταία δημιουργημένου πακέτου στη ροή, πράγμα που επιτρέπει τη σύνδεση μεταξύ των δύο τιμών. Αυτό δίνει την ευχέρεια στους παραλήπτες να συγχρονίσουν την αναπαραγωγή ήχου και εικόνας, με βάση τις πιο πάνω τιμές στις αναφορές αποστολέα.

RTCP και Εύρος Ζώνης

Παρατηρώντας τον τρόπο λειτουργίας του πρωτοκόλλου RTCP, μπορεί κάποιος εύκολα να καταλάβει ότι το μέγεθος του εύρους ζώνης που καταναλώνεται από την αποστολή αναφορών, αποστολών και παραληπτών, αυξάνεται ανάλογα με τον αριθμό των οντοτήτων αυτών. Ανάλογα με την περιοδικότητα αποστολής των μνημάτων αυτών, είναι πιθανό σε κάποιο σημείο ο όγκος δεδομένων που αποστέλλονται στα πλαίσια του RTCP να ξεπεράσει τον όγκο δεδομένων που αποστέλλονται μέσω RTP, καθώς με τη χρήση multicast ο αριθμός των παραληπτών δεν επηρεάζει τον όγκο RTP δεδομένων που αποστέλλονται.

Για να επιλυθεί το πρόβλημα αυτό, το RTCP μεταβάλλει τον ρυθμό που ένας συμμετέχοντας αποστέλλει πακέτα RTCP στο multicast δέντρο, συναρτήσει του αριθμού των συμμετεχόντων στην επικοινωνία. Ο αριθμός αυτός μπορεί να ανακαλυφθεί από κάθε συμμετέχοντα, αφού όλοι στέλνουν πακέτα σε όλους τους άλλους.

Η προσπάθεια του πρωτοκόλλου είναι να περιορίσει το bandwidth που χρησιμοποιεί στο πέντε τοις εκατόν του εύρους ζώνης που αφιερώνεται στη σύνοδο. Από αυτό το ποσοστό, το 25 τοις εκατόν προσφέρεται στους αποστολείς, και το 75 τοις εκατόν στους παραλήπτες. Ένας συμμετέχοντας μπορεί δυναμικά να υπολογίσει την περίοδο μετάδοσης RTCP πακέτων υπολογίζοντας το μέσο μέγεθος RTCP πακέτου (καθόλη τη διάρκεια της συνόδου) και διαιρώντας το με το διαθέσιμο για αυτόν εύρος ζώνης. Κλείνοντας, παρουσιάζονται οι εξισώσεις υπολογισμού της περιόδου μετάδοσης RTCP πακέτων T, τόσο για παραλήπτες, όσο και για αποστολείς.

Για αποστολές,

$$T = \frac{\text{αριθμός αποστολών} * \text{μέσο μέγεθος RTCP πακέτου}}{0.25 * 0.05 * \text{Εύρος Ζώνης Συνόδου}}$$

Για παραλήπτες,

$$T = \frac{\text{αριθμός παραληπτών} * \text{μέσο μέγεθος RTCP πακέτου}}{0.75 * 0.05 * \text{Εύρος Ζώνης Συνόδου}}$$

6.3 Επίπεδο Μεταφοράς

Το επίπεδο μεταφοράς του Διαδικτύου είναι υπεύθυνο για την από άκρο σε άκρο μετάδοση των πακέτων δεδομένων στο τελικό σύστημα και πιο συγκεκριμένα στην διεργασία του τελικού συστήματος στη οποία απευθύνονται τα πακέτα δεδομένων. Η διαδικασία αυτή είναι ανεξάρτητη του δικτύου στο οποίο λαμβάνει χώρα. Το επίπεδο μεταφοράς υποστηρίζει την υπηρεσία αυτή, της μεταφοράς των μηνυμάτων και παράδοσης τους στην κατάλληλη εφαρμογή, μέσω της χρήσης των ports. Τα δύο βασικά πρωτόκολλα του επιπέδου αυτού είναι το Transmission Control Protocol (TCP) και το User Datagram Protocol (UDP). Βασικό στοιχείο το οποίο διαφοροποιεί τα δύο αυτά πρωτόκολλα είναι η έννοια της σύνδεσης μεταξύ των επικοινωνούντων άκρων, με το πρώτο να την υποστηρίζει, ενώ το δεύτερο όχι. Συνεπώς, το connection oriented TCP προσφέρει αξιόπιστη μεταφορά των δεδομένων στην σειρά με την οποία μεταδόθηκαν, σε αντίθεση με το connectionless UDP που δεν παρέχει εγγυήσεις όσο αφορά την παράδοση των πακέτων δεδομένων.

6.3.1 Πρωτόκολλα Επιπέδου Μεταφοράς

Κοινή Υπηρεσία: Πολύπλεξη- Αποπολύπλεξη

Όπως αναφέρθηκε πιο πάνω και θα μελετηθεί λεπτομερώς στη συνέχεια, τα δύο πρωτόκολλα του επιπέδου μεταφοράς, UDP και TCP διαφέρουν μεταξύ τους όσο αφορά τις υπηρεσίες που προσφέρουν, πράγμα που προκύπτει από τον τρόπο λειτουργίας τους. Εντούτοις μία υπηρεσία είναι κοινή και προσφέρεται και από τα δύο. Αυτή είναι η πολύπλεξη και αποπολύπλεξη.

Το επίπεδο δικτύου είναι υπεύθυνο για την μεταφορά των δεδομένων από ένα host σε άλλο. Το επίπεδο μεταφοράς επεκτείνει την λειτουργία αυτή, μεταφέροντας τα segments που λαμβάνει από το πιο κάτω επίπεδο, παραδίδοντας τα δεδομένα από μία διεργασία που τρέχει σε ένα host, σε μια διεργασία που τρέχει σε ένα άλλο. Αυτό επιτυγχάνεται μέσω των διαδικασιών πολύπλεξης στον αποστολέα και αποπολύπλεξης στην παραλήπτη.

Στα πλαίσια της πολύπλεξης, το επίπεδο μεταφοράς συλλέγει τα δεδομένα από τις διάφορες θύρες υποδοχής(sockets) που αντιστοιχούν στις εφαρμογές, τα ενθυλακώνει σε segments προσθέτοντας τις αναγκαίες επιπλέον πληροφορίες επικεφαλίδας και τα προωθεί στο επίπεδο δικτύου. Από την άλλη, στη μεριά του παραλήπτη εκτελείται η αποπολύπλεξη, δηλαδή το επίπεδο μεταφοράς λαμβάνει τα πακέτα από το επίπεδο δικτύου, εξετάζει τα πεδία της επικεφαλίδας και προσδιορίζει σε ποια θύρα υποδοχής πρέπει να αποσταλούν, και τα προωθεί σε αυτή. Μέσω της θύρας υποδοχής, τα δεδομένα καταφθάνουν στην διεργασία που συνδέεται με αυτή.

Αν και η υπηρεσία αυτή υποστηρίζεται και από τα δύο πρωτόκολλα, ωστόσο ο τρόπος λειτουργίας και πάλι διαφέρει. Συγκεκριμένα, στο TCP ένα socket είναι πλήρως καθορισμένο από μία τετράδα πληροφοριών, IP διεύθυνση αποστολέα και παραλήπτη και αριθμό θύρας υποδοχής σε αποστολέα και παραλήπτη. Αυτό είναι αναγκαία λόγω της υποχρεωτικής αποστολής επιβεβαιώσεων στο TCP, για κάθε πακέτο που καταφθάνει στον παραλήπτη. Από την άλλη, στο UDP είναι αναγκαία μόνο η IP διεύθυνση παραλήπτη και ο αριθμός θύρας υποδοχής σε αυτόν. Αποτέλεσμα της πιο πάνω διαφοράς είναι segments που αποστέλλονται από διαφορετικό αποστολέα ή και αριθμό θύρας υποδοχής στον ίδιο παραλήπτη και θύρα υποδοχής θα προωθηθούν στην ίδια διεργασία μέσω του ίδιου socket. Αντίθετα, στο TCP, δύο segments που καταφθάνουν από διαφορετικό αποστολέα ή και θύρα υποδοχής, θα προωθηθούν σε δύο διαφορετικές διεργασίες, μέσω δύο διαφορετικών sockets.

6.3.1 Πρωτόκολλο Transmission Control Protocol (TCP)

Το πρωτόκολλο TCP αποτελεί ένα από τα δύο κύρια πρωτόκολλα της σουίτας TCP/IP και παρέχει μια αξιόπιστη υπηρεσία βασισμένη σε συνδέσεις η οποία εγγυάται την αξιόπιστη μετάδοση των πακέτων δεδομένων στην σειρά με την οποία μεταδόθηκαν, ανάμεσα σε δύο

τερματικά συστήματα. Πρόκειται για πολύ σημαντικό πρωτόκολλο, αφού αποτελεί ένα ενδιάμεσο επίπεδο επικοινωνίας χωρίς λάθη ανάμεσα στις εφαρμογές και στο πρωτόκολλο IP. Πέρα από την αξιοπιστία την οποία προσφέρει στην παράδοση των πακέτων δεδομένων, χρησιμοποιεί και κατάλληλους μηχανισμούς που επιτρέπουν την αποφυγή συμφόρησης στο δίκτυο και πρόληψης προβλημάτων που αυτή θα δημιουργούσε. Επιπλέον λαμβάνει υπόψη και τον ρυθμό με τον οποίο ο παραλήπτης είναι σε θέση να λαμβάνει τα πακέτα που του αποστέλλονται.

Τρόπος και Χαρακτηριστικά Λειτουργίας

Εδραίωση και Τερματισμός Σύνδεσης

Όπως ήδη αναφέρθηκε βασικό στοιχείο της λειτουργίας του TCP, αποτελεί η έννοια της σύνδεσης ανάμεσα στις δύο επικοινωνούντες οντότητες. Η έναρξη της επικοινωνίας αυτής επιτυγχάνεται με την τριμερή χειραψία (3 way handshake). Η διαδικασία αυτή ξεκινά με την αποστολή ενός μηνύματος, στο οποίο το SYN (synchronize) bit είναι ενεργοποιημένο. Επίσης, το πεδίο ακολουθίας στην TCP επικεφαλίδα, ορίζεται στον αρχικό αριθμό του. Από την άλλη, ο εξυπηρετητής είναι δυνατό να απαντήσει με δύο τρόπους. Είτε με πακέτο SYN και ACK (acknowledge) στο οποίο ο αριθμός ακολουθίας είναι κατά ένα μεγαλύτερος από αυτό που έλαβε από τον client, για να αποδεκτεί τη σύνδεση, είτε με πακέτο SYN/RST ώστε να ενημερώσει τον client ότι αρνείται τη σύνδεση και να τερματιστεί η διαδικασία.

Για τερματισμό της σύνδεσης αυτής απαιτείται η διαδικασία ενός four way handshake, με την κάθε πλευρά να τερματίζει ξεχωριστά. Αρχικά, πρέπει η πλευρά που επιθυμεί να κλείσει τη σύνδεση αποστέλλει ένα μήνυμα με το FIN (finish) πεδίο της επικεφαλίδας ενεργοποιημένο. Το πακέτο αυτό επιβεβαιώνεται από την άλλη πλευρά με ένα ACK, και στην συνέχεια αποστέλλει επίσης ένα πακέτο FIN. Τα βήματα ολοκληρώνονται με την πλευρά που ξεκίνησε τη διαδικασία να επιβεβαιώνει τη λήψη του FIN με ένα ACK.

Αξιόπιστη Μετάδοση

Προκειμένου να επιτύχει την αξιόπιστη μετάδοση των πακέτων δεδομένων ο αποστολέας της TCP κίνησης απαιτεί από τον παραλήπτη να επιβεβαιώνει την ορθή λήψη των πακέτων που του μεταδίδει. Για τον σκοπό αυτό ο αποστολέας της TCP κίνησης διατηρεί ένα

αντίγραφο κάθε πακέτου το οποίο μεταδίδει μέχρι αυτό να επιβεβαιωθεί από τον παραλήπτη. Εάν ο αποστολέας της TCP κίνησης δεν λάβει επιβεβαίωση για ένα πακέτο δεδομένων μετά από χρόνο t_{out} από τότε που το μετάδωσε, το ξαναμεταδίδει. Η τιμή του t_{out} πρέπει να επιλεγεί με προσοχή ώστε ο αποστολέας να αντιλαμβάνεται έγκαιρα τις απώλειες των πακέτων αλλά και να μην ξαναμεταδίδει πακέτα τα οποία ακόμη βρίσκονται καθοδόν προς τον παραλήπτη τους.

Ο αλγόριθμος που ακολουθεί, χρησιμοποιείται για τον υπολογισμό του t_{out} , ο οποίος βασίζεται σε υπολογισμούς του χρόνου καθυστέρησης μετάδοσης μετά επιστροφής ενός πακέτου. Ο χρόνος αυτός έχει το όνομα RTT (Round Trip Time) και αντιστοιχεί στον χρόνο που απαιτείται για την αποστολή ενός πακέτου δεδομένων από τον αποστολέα στον παραλήπτη και στη συνέχεια πίσω στον αποστολέα.

$$t_{diff} = t_{measured} - t_{RTT}$$

$$t_{RTT} = t_{RTT} + \delta * t_{diff}$$

$$\sigma_n = \sigma_{n-1} + \rho(|t_{diff}| - \sigma_{n-1})$$

$$t_{out} = t_{RTT} + n * \sigma_n$$

Στον πιο πάνω αλγόριθμο $t_{measured}$ είναι ο χρόνος RTT ο οποίος μετρίεται στο δίκτυο, t_{RTT} είναι ο φιλτραρισμένος χρόνος RTT, δηλαδή η φιλτραρισμένη μέση διακύμανση και t_{out} είναι ο χρόνος λήξης για την αναμετάδοση ενός πακέτου δεδομένων. Ο χρόνος μετρίεται βάσει του χρόνου που απαιτείται από την αποστολή ενός πακέτου από τον αποστολέα μέχρι αυτός να λάβει την επιβεβαίωση από τον παραλήπτη. Η παράμετρος δ έχει τιμές από 0 έως 1 και καθορίζει πόσο ευαίσθητος θα είναι ο υπολογισμός του t_{RTT} στις μετρήσεις οι οποίες λαμβάνονται από το δίκτυο και συνήθως έχει την τιμή 0.125. Η παράμετρος ρ καθορίζει πόσο γρήγορα τα νέα δείγματα επηρεάζουν την μέση διακύμανση δηλαδή και συνήθως έχει την τιμή 0.25. Η παράμετρος n καθορίζει το πόσο η μέση διακύμανση επηρεάζει την τιμή του χρόνου t_{out} . Αρχικά είχε προταθεί η τιμή 2 για την παράμετρο n , όμως στην έκδοση Reno του TCP προτείνεται η τιμή 4. Επιπλέον του χρόνου λήξης

επαναμετάδοσης του, η ανίχνευση των πακέτων που απωλέσθηκαν μπορεί να γίνει και με την χρήση του σχήματος γρήγορης επαναμετάδοσης (fast retransmission scheme). Σύμφωνα με το σχήμα γρήγορης επαναμετάδοσης, το πακέτο θεωρείται ότι απωλέσθηκε εάν ο αποστολέας λάβει περισσότερες από μια (συνήθως τρεις) επιβεβαιώσεις για πακέτα μετά το n-1 πακέτο. Σε αυτή την περίπτωση το πακέτο μεταδίδεται ξανά.

Έλεγχος Ροής και Συμφόρησης

Το TCP παρέχει ένα ιδιαίτερα αξιόπιστο μηχανισμό για τον έλεγχο ροής και συμφόρησης στον οποίο οφείλεται και η σημαντική χρήση και εξάπλωση του πρωτοκόλλου TCP. Ο μηχανισμός αυτός βασίζεται στην τεχνική του κυλιόμενου παραθύρου για την μετάδοση των πακέτων δεδομένων με την χρήση του οποίου αποφεύγονται φαινόμενα συμφόρησης τα οποία οδηγούν σε απώλειες πακέτων δεδομένων. Ο μηχανισμός αυτό λειτουργεί ως εξής: Ο αποστολέας της TCP κίνησης διατηρεί μια μνήμη προσωρινής αποθήκευσης (buffer) η οποία ονομάζεται «παράθυρο μετάδοσης» (transmission window), στην οποία εισάγει ένα αντίγραφο από κάθε πακέτο δεδομένων το οποίο μεταδίδει. Όσο υπάρχει διαθέσιμος χώρος στην μνήμη προσωρινής αποθήκευσης ο αποστολέας μεταδίδει πακέτα στο δίκτυο και σε περίπτωση που η μνήμη προσωρινής αποθήκευσης γεμίσει, ο αποστολέας «παγώνει» την μετάδοση πακέτων. Ο παραλήπτης των TCP πακέτων επιβεβαιώνει την λήψη των πακέτων δεδομένων και στο πακέτο επιβεβαίωσης το οποίο στέλνει περιλαμβάνει την ελεύθερη χωρητικότητα της δικής του μνήμης προσωρινής αποθήκευσης και τον αριθμό του τελευταίου πακέτου το οποίο έλαβε από τον αποστολέα. Ο αποστολέας όταν λαμβάνει ένα πακέτο επιβεβαίωσης από τον παραλήπτη, διαγράφει από την μνήμη προσωρινής αποθήκευσης του τα πακέτα δεδομένων που επιβεβαιώνει ο παραλήπτης και έτσι δημιουργείται ελεύθερος χώρος στην μνήμη προσωρινής αποθήκευσης για την μετάδοση επιπλέον πακέτων δεδομένων. Η πρώτη έκδοση του TCP λάμβανε υπόψη της μόνο τα θέματα της αξιοπιστίας και του ελέγχου ροής και προσπαθούσε να μην μεταδίδει περισσότερα πακέτα δεδομένων από αυτά τα οποία μπορεί να διαχειριστεί ο παραλήπτης. Η τακτική αυτή έχει ικανοποιητικά αποτελέσματα όταν ο περιοριστικός παράγοντας στην μετάδοση των δεδομένων είναι ο παραλήπτης και αντίθετα δεν λειτουργεί ικανοποιητικά όταν περιοριστικός παράγοντας είναι το δίκτυο. Σε αυτή την

περίπτωση, η παραπάνω τακτική μπορεί να οδηγήσει στο φαινόμενο του «congestion collapse». Σύμφωνα με το φαινόμενο αυτό, ενώ υπάρχει υψηλή κίνηση στο δίκτυο το ποσοστό της «ωφέλιμης» κίνησης είναι μικρό και το μεγαλύτερο μέρος της κίνησης οφείλεται στην επαναμετάδοση πακέτων που είτε έχουν παραδοθεί στον παραλήπτη (αλλά ο αποστολέας δεν έχει λάβει ακόμη την επιβεβαίωση από τον παραλήπτη) είτε βρίσκονται στο δίκτυο ακόμη.

Additive Increase, Multiplicative Decrease

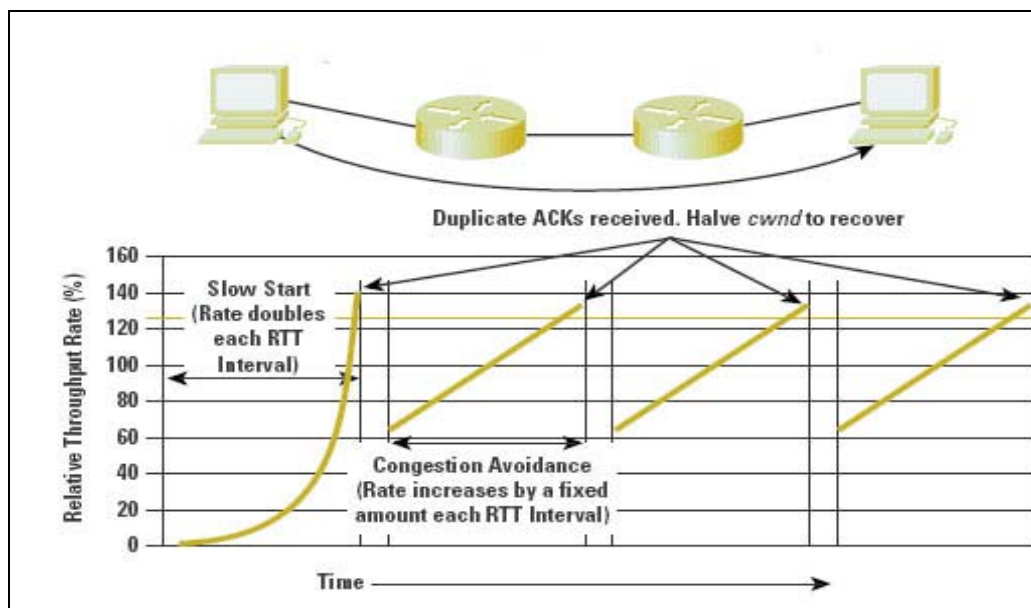
Η βασική ιδέα πίσω από τον έλεγχο συμφόρησης του TCP είναι η μείωση του ρυθμού μετάδοσης δεδομένων από μέρους του αποστολέα (μέσω της ελάττωσης του μεγέθους του παραθύρου συμφόρησης), σε περίπτωση ύπαρξης απώλειας πακέτου. Καθώς, οι TCP συνδέσεις που παρνοούν μέσω δρομολογητών που παρουσιάζουν συμφόρηση, πιθανό να αντιμετωπίζουν και απώλειες πακέτων, είναι επόμενο με τη μείωση των ρυθμών που η κάθε μία αποστέλλει δεδομένα στο μονοπάτι αυτό, να εξασθαινίσει και το φαινόμενο της συμφόρησης. Για την επιλογή του πόσο θα πρέπει όμως να μειωθεί το παράθυρο συμφόρησης, το πρωτόκολλο TCP ακολουθεί την προσέγγιση του multiplicative decrease, δηλαδή μετά από ανίχνευση απώλειας, το παράθυρο μειώνεται στο μισό της τιμής που είχε τη στιγμή που παρουσιάστηκε η απώλεια. Αυτή η μείωση μπορεί να συνεχίσει να εκτελείται μετά από διαδοχικές απώλειες πακέτων, μέχρι το παράθυρο να έχει το μέγεθος ενός MSS(maximum segment size).

Σε περίπτωση που γίνει αντιληπτό ότι δεν υφίσταται συμφόρηση στο μονοπάτι που χρησιμοποιείται για μία σύνδεση TCP, τότε είναι λογικό να υπάρχει διαθέσιμο επιπλέον εύρος ζώνης, που δύναται να χρησιμοποιηθεί. Ο τρόπος που αυτό αντιμετωπίζεται από το TCP είναι με την αύξηση κατά ένα MSS του παραθύρου συμφόρησης για κάθε RTT, δεδομένου ότι λαμβάνει πακέτα επιβεβαίωσης για τα δεδομένα που αποστέλλει.

Η φάση αυτή του ελέγχου συμφόρησης του TCP, δηλαδή η γραμμική αύξηση, ονομάζεται «αποφυγή συμφόρησης» (congestion avoidance).

Η τιμή του παραθύρου συμφόρησης περνά από πολλές διακυμάνσεις κατά τη διάρκεια μιας μεγάλης διάρκειας TCP σύνδεσης, αφού επανηλειμμένα μειώνεται στο μισό και

ακολουθώς αυξάνεται γραμμικά, προκαλώντας τη χαρακτηριστική γραφική παράσταση του παραθύρου συμφόρησης, όπως φαίνεται στην πιο κάτω εικόνα.



Σχήμα 6-8: Έλεγχος Συμφόρησης μέσω Additive Increase, Multiplicative Decrease

Αργό ξεκίνημα (slow start)

Όταν εδραιώνεται μια TCP σύνδεση, το παράθυρο συμφόρησης έχει μέγεθος ένα και κάθε φορά που λαμβάνεται μια επιβεβαίωση από τον παραλήπτη, το παράθυρο συμφόρησης αυξάνεται κατά το αριθμό των πακέτων που επιβεβαιώνονται. Συνεπώς, η τιμή του παραθύρου συμφόρησης ουσιαστικά διπλασιάζεται κάθε RTT. Αυτό συμβαίνει καθώς το διαθέσιμο εύρος ζώνης είναι λογικό να είναι κατά πολύ μεγαλύτερο από MSS/RTT , ο οποίος είναι και ο αρχικός ρυθμός μετάδοσης δεδομένων. Στη συνέχεια ο αποστολέας μπορεί να μεταδίδει τόσα πακέτα δεδομένων όσο η μικρότερη τιμή από το παράθυρο συμφόρησης και το μέγεθος παραθύρου που περιέχεται στο πακέτο επιβεβαίωση. Η διαδικασία αυτή συνεχίζεται μέχρι να ανιχνευτεί απώλεια κάποιου πακέτου.

Τρόπος Αντίδρασης σε Περίπτωση Timeout

Ο τρόπος με τον οποίο ο μηχανισμός ελέγχου συμφόρησης του TCP αντιμετωπίζει τις απώλειες πακέτων διαφέρει ανάμεσα στην περίπτωση που η απώλεια είναι συνέπεια timeout και στην περίπτωση που η απώλεια είναι αποτέλεσμα τριών διαδοχικών επιβεβαιώσεων για το ίδιο πακέτο. Στην πρώτη περίπτωση, η αντίδραση δεν είναι η ίδια με αυτή που αναφέρθηκε σε προηγούμενες παραγράφους και η οποία αφορούσε απώλειες που προέκυπταν μέσω της παραλαβής τριών διαδοχικών πακέτων επιβεβαίωσης.

Μετά από timeout, ο αποστολέας μπαίνει στην φάση «αργό ξεκίνημα», δηλαδή θέτει το παράθυρο συμφόρησής του σε 1 MSS και συνεχίζει να αυξάνει το μέγεθός του εκθετικά. Αυτό συμβαίνει μέχρι το μέγεθος του παραθύρου να φτάσει το μισό της τιμής που είχε πριν συμβεί η απώλεια πακέτου. Από αυτό το σημείο και έπειτα, η αύξηση ξεκινά και πάλι να είναι γραμμική, όπως στην περίπτωση των τριών διαδοχικών επιβεβαιώσεων.

Εκδόσεις TCP

Από την δημιουργία του Διαδικτύου διάφορες εκδόσεις του TCP έχουν προταθεί, στην συνέχεια αναφέρουμε τις πιο σημαντικές εκδόσεις του TCP.

Tahoe – TCP

Το Tahoe - TCP υλοποιήθηκε το 1988 στο 4.3 BSD λειτουργικό. Το Tahoe – TCP χρησιμοποιεί τους αλγορίθμους «αργό ξεκίνημα» και «αποφυγή συμφόρησης» όπως επίσης και αλγόριθμο «γρήγορης επαναμετάδοσης» (fast retransmission).

Reno – TCP

Το Reno - TCP υλοποιήθηκε στο 1990 και αποτελεί την πιο ευρέως χρησιμοποιούμενη έκδοση του TCP. Το Reno - TCP χρησιμοποιεί τους αλγορίθμους «αργό ξεκίνημα», «αποφυγή συμφόρησης», «γρήγορης επαναμετάδοσης» και επιπλέον χρησιμοποιεί ένα μηχανισμό «γρήγορης επαναφοράς» (fast recovery) σύμφωνα με τον οποίο ο αποστολέας της TCP κίνησης μεταδίδει ξανά ένα χαμένο πακέτο δεδομένων αφού λάβει τρεις διπλές επιβεβαιώσεις πακέτων.

New - Reno TCP

Η έκδοση Tahoe - TCP μειώνει το παράθυρο μετάδοσης στο ένα πακέτο μετά από μια απώλεια πακέτου, συμπεριφορά η οποία είναι αρκετά συντηρητική και μπορεί να οδηγήσει σε μικρή αξιοποίηση των πόρων του δικτύου. Αντίθετα το Reno - TCP μπορεί να οδηγήσει σε μικρή αξιοποίηση των πόρων του δικτύου όταν υπάρχουν πολλαπλές απώλειες πακέτων στον χρόνο RTT.

SACK – TCP

Στο SACK - TCP (Selective ACKnowledgment TCP) γίνεται επιλεκτική επιβεβαίωση πακέτων από τον παραλήπτη. Ο παραλήπτης συμπεριλαμβάνει σε κάθε πακέτο επιβεβαίωσης έναν αριθμό από επιβεβαιώσεις οι οποίες επιβεβαιώνουν μη συνεχόμενες ομάδες πακέτων δεδομένων οι οποίες έχουν ληφθεί σωστά. Αυτό επιτρέπει στον παραλήπτη να αντιλαμβάνεται έγκαιρα τις απώλειες δεδομένων και να αποφεύγει περιττές καθυστερήσεις με αποτέλεσμα την καλύτερη απόδοση.

Vegas – TCP

Το Vegas - TCP χρησιμοποιεί μια βελτιωμένη μέθοδο για να υπολογίζει το μέγεθος των παραθύρων η οποία βασίζεται στην απόκλιση των μετρήσεων του χρόνου RTT.

ECN – TCP

Έχουν προταθεί διάφορες επεκτάσεις στο TCP που εκμεταλλεύονται τις πληροφορίες για πρόωρη ανίχνευση της συμφόρησης (ECN - Early Congestion Notification) που δύναται να παρέχει το δίκτυο. Σε αυτή την περίπτωση οι δρομολογητές του δικτύου μαρκάρουν ένα συγκεκριμένο bit στις επικεφαλίδες των πακέτων δεδομένων όταν αρχίζουν να υπερφορτώνονται. Το TCP λαμβάνοντας αυτήν την πληροφορία μπορεί να μειώσει το ρυθμό μετάδοσης πριν αρχίσουν να απορρίπτονται πακέτα δεδομένων και με αυτόν τον τρόπο πετυχαίνει καλύτερη απόδοση. Το βασικότερο μειονέκτημα αυτών των επεκτάσεων είναι ότι απαιτούνται αλλαγές στην υποδομή του δικτύου και όχι μόνο στα τελικά συστήματα όπως συμβαίνει στις άλλες επεκτάσεις του TCP.

6.3.2 Το Πρωτόκολλο User Datagram Protocol (UDP)

Πρόκειται για το έτερο πλέον διαδεδομένο πρωτόκολλο του επιπέδου μεταφοράς στο Διαδίκτυο. Σε αντίθεση με το TCP που εξετάστηκε λεπτομερώς προηγουμένως, το UDP δεν προσφέρει καμία εγγύηση όσο αφορά την αξιόπιστη και σε σειρά μεταφορά των πακέτων δεδομένων(που ονομάζονται datagrams) από τον αποστολέα στον παραλήπτη. Αντιθέτως, οι μοναδικές υπηρεσίες που προσφέρονται στα πλαίσια του πρωτοκόλλου αυτού είναι η πολύπλεξη και αποπολύπλεξη, με τον τρόπο λειτουργίας που επεξηγήθηκε πιο πριν. Στο UDP εξάλλου δεν χρησιμοποιείται η έννοια της σύνδεσης μεταξύ των επικοινωνούντων οντοτήτων και ο τρόπος λειτουργίας του είναι πολύ απλούστερος από το TCP.

Το γεγονός ότι το TCP απαιτεί από τον παραλήπτη να επιβεβαιώνει στον αποστολέα την λήψη κάθε πακέτου δεδομένων, μπορεί να δημιουργήσει το φαινόμενο της «πλημμυράς πληροφοριών ανάδρασης» (feedback implosion) στην περίπτωση μιας σύνδεσης ενός σημείου προς πολλά σημεία (point to multipoint). Κατά συνέπεια η multicast μετάδοση δεδομένων υλοποιείται αποτελεσματικότερα με τη χρήση του UDP στο Διαδίκτυο. Επιπλέον το TCP μπορεί να διακόψει την μετάδοση των δεδομένων όσο περιμένει ένα πακέτο επιβεβαίωσης ή μπορεί να μειώσει τον ρυθμό μετάδοσης δεδομένων δραστικά όταν αντιληφθεί απώλεια πακέτων. Αυτή τη συμπεριφορά είναι αποδεκτή για εφαρμογές που η εγγυημένη παράδοση δεδομένων είναι απαραίτητη όμως δεν είναι κατάλληλη για εφαρμογές που χαρακτηρίζονται σαν time- sensitive και κυρίως εφαρμογές πραγματικού χρόνου όπως οι εφαρμογές πολυμέσων οι οποίες λαμβάνουν πληροφορία την οποία παρουσιάζουν στον χρήστη. Πολλές φορές σε αυτές τις εφαρμογές η έγκαιρη μετάδοση είναι πιο σημαντική από την 100% ορθή μετάδοση των δεδομένων. Για τους παραπάνω λόγους οι εφαρμογές πολυμέσων στο Διαδίκτυο (πχ τηλεφωνία με χρήση του πρωτοκόλλου IP) στηρίζονται στην μετάδοση δεδομένων με την χρήση του πρωτοκόλλου UDP. Το γεγονός ότι το πρωτόκολλο UDP δεν υποστηρίζει έλεγχο ροής και αξιόπιστη μετάδοση δεδομένων, έχει ως αποτέλεσμα τα παραπάνω χαρακτηριστικά να υλοποιούνται στο επίπεδο των εφαρμογών, όταν αυτό κρίνεται αναγκαίο.

Αιτίες Επιλογής UDP για Εφαρμογές Ευαίσθητες σε Καθυστέρηση

- Καλύτερος έλεγχος σχετικά με ποια δεδομένα αποστέλλονται και πότε: Με τη χρήση UDP, όταν μια διεργασία προωθήσει δεδομένα στο επίπεδο μεταφοράς, το UDP θα ενθυλακώσει τα δεδομένα αυτά σε ένα UDP segment και αμέσως θα τα προωθήσει στο επίπεδο δικτύου. Αντίθετα, κάνοντας χρήση του TCP, είναι πιθανό λόγω του ελέγχου συμφόρησης, η διαδικασία αυτή να υποστεί καθυστέρηση αν ένας από τους συνδέσμους στο μονοπάτι που θα ακολουθηθεί αντιμετωπίζει συμφόρηση. Επίσης, με το TCP, ένα πακέτο θα αποστέλλεται συνεχώς μέχρι να ληφθεί το πακέτο επιβεβαίωσης για αυτό, πράγμα που δεν είναι απαραίτητο για εφαρμογές που είναι ανεχτικές στις απώλειες, όπως εφαρμογές πολυμεσικού περιεχομένου πραγματικού χρόνου. Εξάλλου, τέτοιες εφαρμογές είναι συχνά πλέον σε θέση να χρησιμοποιούν το UDP για μεταφορά δεδομένων, κάνοντας παράλληλα χρήση μηχανισμών που επιτρέπουν την διόρθωση των σφαλμάτων που προκύπτουν από τυχόν απώλειες πακέτων, όπως για παράδειγμα το Forward Error Correction σε εφαρμογές Video.
- Μη αναγκαία εγκαθίδρυση σύνδεσης: Σε αντίθεση με το TCP, όπου εμφανίζεται η έννοια της σύνδεσης και της τριμερούς χειραψίας, στο UDP αυτό δεν είναι αναγκαίο, και συνεπώς δεν προσθέτει επιπλέον καθυστέρηση στη μεταφορά των δεδομένων. Φυσικά, αυτό είναι αναγκαίο για εφαρμογές που δεν είναι ανεχτικές στις απώλειες, όπως το ηλεκτρονικό ταχυδρομίο, εντούτοις η εγκαθίδρυση σύνδεσης αποτελεί ένα μεγάλο μέρος της καθυστέρησης που παρατηρείται κατά τη χρήση της εφαρμογής αυτής.
- Μη Ύπαρξη της Έννοιας της Σύνδεσης: Το TCP διατηρεί κατάσταση σύνδεσης στα τερματικά μεταξύ των οποίων μεταφέρονται τα δεδομένα. Αυτό έχει σαν αποτέλεσμα την εμφάνιση πολλών παραμέτρων όπως αριθμών ακολουθίας και επιβεβαίωσης και μνήμες προσωρινής αποθήκευσης για αποστολή και παραλαβή. Τα χαρακτηριστικά αυτά είναι αναγκαία για την εξασφάλιση της αξιόπιστης μεταφοράς δεδομένων που προσφέρει το πρωτόκολλο TCP και ασφαλώς δεν εμφανίζονται στο UDP. Ως εκ τούτου είναι επόμενο αρκετοί εξυπηρετητές που είναι αφοσιωμένοι σε μια εφαρμογή να είναι σε θέση να υποστηρίξουν ταυτόχρονα περισσότερους ενεργούς πελάτες που κάνουν χρήση UDP, παρά TCP.
- Μικρό Μέγεθος Επικεφαλίδας Πακέτου: Σε αντίθεση με τα 20 bytes της επικεφαλίδας του TCP πακέτου, το UDP έχει μέγεθος επικεφαλίδας μόλις 8 bytes.

6.4 Επίπεδο Δικτύου

Το επίπεδο του Δικτύου είναι υπεύθυνο για τη δρομολόγηση των πακέτων δεδομένων στον τελικό τους προορισμό. Κατά τη διάρκεια μετάδοσης των πακέτων δεδομένων δε χρησιμοποιείται κάποια σύνδεση και είναι πιθανόν πακέτα δεδομένων της ίδιας ροής δεδομένων να ακολουθήσουν διαφορετικές διαδρομές για να φτάσουν στον προορισμό τους. Το πρωτόκολλο του επιπέδου Δικτύου το οποίο χρησιμοποιείται στον Διαδίκτυο είναι το IP (Internet Protocol).

Θα πρέπει να τονίσουμε ότι το επίπεδο Δικτύου δεν παρέχει καμία εγγύηση για την παράδοση των πακέτων δεδομένων, κατά συνέπεια τα πακέτα δεδομένων μπορεί να φτάσουν καθυστερημένα στον παραλήπτη τους, να φτάσουν σε διαφορετική σειρά από την σειρά την οποία μεταδόθηκαν ή να μην φτάσουν καθόλου (δηλαδή να χαθούν στο δίκτυο). Η απώλεια των πακέτων μπορεί να οφείλεται είτε σε λάθη κατά τη μετάδοσή τους στο φυσικό μέσο είτε λόγω υπερχείλισης της μνήμης προσωρινής αποθήκευσης (buffer) των δικτυακών συσκευών. Στα σημερινά δίκτυα η απώλεια πακέτων οφείλεται κυρίως στην υπερχείλιση της μνήμης προσωρινής αποθήκευσης των δικτυακών συσκευών και λιγότερο στα λάθη που παρουσιάζονται στο φυσικό μέσο. Η σημαντικότερη υπηρεσία την οποία παρέχουν οι δικτυακές συσκευές οι οποίες λειτουργούν στο επίπεδο του Δικτύου (δηλαδή οι δρομολογητές του Διαδικτύου) είναι η δρομολόγηση των πακέτων δεδομένων από τον αποστολέα στον παραλήπτη. Λόγω του γεγονότος ότι τα πακέτα δεδομένων σε μια εισερχόμενη γραμμή ενός δρομολογητή μπορεί να λαμβάνονται με διαφορετικό ρυθμό από τον ρυθμό τον οποίο η δικτυακή συσκευή τα μεταδίδει σε μια εξερχόμενη γραμμή, πολλές φορές τα εισερχόμενα πακέτα δεδομένων χρειάζεται να αποθηκευτούν στην μνήμη προσωρινής αποθήκευσης πριν προωθηθούν σε μία εξερχόμενη γραμμή. Σημαντικός παράγοντας στη λειτουργία της δικτυακής συσκευής είναι ο τρόπος με τον οποίο η δικτυακή συσκευή διαχειρίζεται τα πακέτα δεδομένων τα οποία βρίσκονται αποθηκευμένα στη μνήμη προσωρινής αποθήκευσης. Πολλές τεχνικές έχουν προταθεί για αυτό τον σκοπό (γνωστές ως αλγόριθμοι δρομολόγησης) και στη συνέχεια παρουσιάζονται οι πιο διαδεδομένες τεχνικές.

FIFO (First In First Out)

Σε αυτήν την τεχνική (η οποία είναι μια από τις πιο απλές τεχνικές αλλά και από τις πλέον χρησιμοποιημένες στους δρομολογητές του Διαδικτύου) η μνήμη προσωρινής αποθήκευσης αποτελείται από μία FIFO ουρά στην οποία τα πακέτα δεδομένων προωθούνται στις εξερχόμενες γραμμές με την σειρά με την οποία έφτασαν στις εισερχόμενες γραμμές του δρομολογητή. Σε περίπτωση που η FIFO ουρά φτάσει στο μέγιστό μέγεθος της τα πακέτα τα οποία συνεχίζουν να έρχονται στις εισερχόμενες γραμμές απορρίπτονται, μέχρι να υπάρχει διαθέσιμος χώρος στην FIFO ουρά. Αυτή η τεχνική μπορεί να υλοποιηθεί σχετικά εύκολα αλλά πάσχει από προβλήματα συγχρονισμού και άδικου διαχωρισμού του εύρους ζώνης ανάμεσα σε ροές δεδομένων οι οποίες ανταγωνίζονται για το διαθέσιμο εύρος ζώνης. Το πρόβλημα συγχρονισμού των ροών δεδομένων είναι ιδιαίτερα έντονο όταν TCP ροές δεδομένων διασχίζουν δρομολογητές οι οποίοι είναι υπερφορτωμένοι και απορρίπτουν συχνά πακέτα. Λόγω της συμφόρησης η οποία παρουσιάζεται όλες οι TCP συνδέσεις χρησιμοποιούν τον μηχανισμό αποτροπής της συμφόρησης τον οποίο υλοποιούν και μειώνουν τον ρυθμό μετάδοσης δεδομένων περίπου την ίδια χρονική στιγμή με αποτέλεσμα οι δρομολογητές να μην είναι πλέον υπερφορτωμένοι. Στην συνέχεια οι TCP συνδέσεις αυξάνουν τον ρυθμό μετάδοσης δεδομένων μέχρι οι δρομολογητές να υπερφορτωθούν ξανά όποτε οι TCP συνδέσεις ξανά μειώνουν τον ρυθμό μετάδοσης δεδομένων. Το φαινόμενο αυτό επαναλαμβάνεται συνεχώς και οδηγεί σε μη πλήρη αξιοποίηση των υπάρχοντων δικτυακών πόρων. Επιπλέον, η FIFO ουρά μπορεί να οδηγήσει και σε προβλήματα δικαιοσύνης λόγω του ότι η πιθανότητα να απορριφθεί ένα πακέτο δεδομένων είναι η ίδια για όλες τις ροές δεδομένων οι οποίες συναγωνίζονται για το διαθέσιμο εύρος ζώνης. Το γεγονός αυτό έχει ως αποτέλεσμα μια άπληστη πηγή δεδομένων (η οποία δεν εφαρμόζει κάποιο μηχανισμό αποφυγής της συμφόρησης και μεταδίδει δεδομένα με υψηλό ρυθμό μετάδοσης με αποτέλεσμα να χρησιμοποιεί μεγάλο μέρος του διαθέσιμου εύρους ζώνης) να έχει το ίδιο ποσοστό απώλειας των πακέτων δεδομένων της με μια πηγή δεδομένων η οποία καταναλώνει πολύ μικρότερο μέρος του διαθέσιμου εύρους ζώνης. Αυτό δεν είναι δίκαιο λόγω του γεγονότος ότι παρόλο που το πρόβλημα της συμφόρησης οφείλεται κυρίως στην πηγή με τον υψηλό ρυθμό μετάδοσης δεδομένων αυτή η πηγή συνεχίζει να λαμβάνει μεγάλο μέρος του διαθέσιμου εύρους ζώνης.

RED (Random Early Detection)

Η τεχνική RED αποτελείται από μια ουρά FIFO στην οποία έχουν ορισθεί δύο όρια. Όταν το μέγεθος της ουράς ξεπεράσει το πρώτο όριο, η ουρά απορρίπτει τα εισερχόμενα πακέτα δεδομένων με κάποια ορισμένη πιθανότητα. Όταν το μέγεθος της ουράς ξεπεράσει και το δεύτερο όριο, η ουρά απορρίπτει όλα τα εισερχόμενα πακέτα δεδομένων. Με αυτή την τεχνική το μήκος της ουράς διατηρείται μικρό και ταυτόχρονα αποφεύγεται το φαινόμενο του συγχρονισμού. Η τεχνική RED παρουσιάζει τα προβλήματα δικαιοσύνης τα οποία παρουσιάζει και η ουρά FIFO. Η τεχνική RED υποστηρίζεται από τους περισσότερους σύγχρονους δρομολογητές αν και δεν χρησιμοποιείται σε πολλούς εγκατεστημένους δρομολογητές του Διαδικτύου.

Δίκαιη Διαχείριση Ουρών (Fair Queuing Mechanism)

Οι περισσότερες από αυτές τις τεχνικές πετυχαίνουν ομοιόμορφη κατανομή του εύρους ζώνης και απομονώνουν τις άπληστες πηγές δεδομένων. Κάποιες άλλες τεχνικές δεσμεύουν συγκεκριμένο εύρος ζώνης για κάθε ροή δεδομένων και εγγυώνται, όσο αυτό μπορεί να ισχύσει στο Διαδίκτυο, συγκεκριμένα όρια τόσο στο εύρος ζώνης, όσο και στην καθυστέρηση.

Οι παραπάνω τεχνικές είναι σε γενικές γραμμές πολύπλοκες στην χρήση τους (ειδικά στην περίπτωση που οι δρομολογητές δεν διαχειρίζονται από τον ίδιο οργανισμό όπως στην περίπτωση του Διαδικτύου) και στις μέρες μας βρίσκουν εφαρμογή κυρίως στα εσωτερικά δίκτυα οργανισμών και λιγότερο στους κεντρικούς δρομολογητές του Διαδικτύου.

ΚΕΦΑΛΑΙΟ 7

ΣΥΣΤΗΜΑ ADIVIS (ADAPTIVE INTERNET VIDEO STREAMING)

7.1 Εισαγωγή

7.2 Αλγόριθμος ADIVIS

7.3 Κωδικοποίηση σε Στρώματα (Layered Encoding)

7.4 Μηχανισμός Ανατροφοδότησης (Feedback Mechanism)

7.5 Μηχανισμός Συγκεχυμένης Απόφασης (Fuzzy Decision Mechanism)

7.1 Εισαγωγή

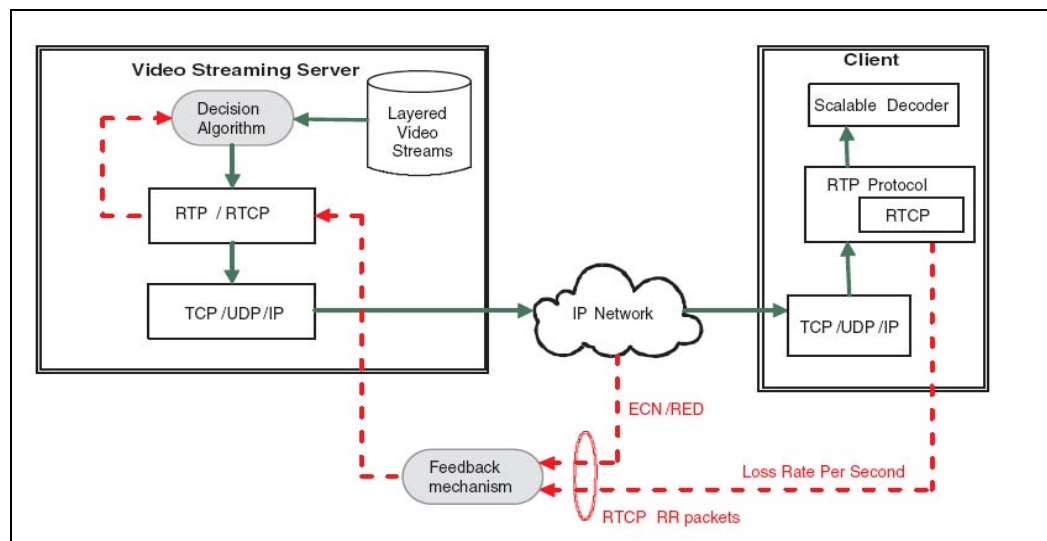
Η μεταφορά δεδομένων πάνω από ασύρματα δίκτυα είναι μια αναπτυσσόμενη και πολλά υποσχόμενη τεχνολογία. Πέραν τούτου, πολλές εφαρμογές απαιτούν τη μεταφορά του βίντεο σε πραγματικό χρόνο, εντούτοις η απροσδιοριστία που χαρακτηρίζει τα ασύρματα και κινητά δίκτυα από πλευράς εύρους ζώνης, καθυστέρησης και χαμένων πακέτων έχει μεγάλο αντίκτυπο στην μετάδοση των ροών βίντεο. Σε συνδυασμό με τα πιο πάνω, η αλματώδης αύξηση των ετερογενών συσκευών με δυνατότητα προβολής βίντεο και πρόσβασης σε ασύρματα δίκτυα, καθιστά ακόμα πιο αναγκαία την εξεύρεση αποδοτικών και αποτελεσματικών τεχνικών για adaptive συμπιεσμένες ροές βίντεο, ώστε να ικανοποιούνται καλύτερα οι περιορισμοί, οι απαιτήσεις και οι ικανότητες των ασύρματων και κινητών δικτύων, εφαρμογών, υπηρεσιών και τελικών χρηστών.

Η προσέγγιση του ADIVIS προς το πρόβλημα αυτό συνδιάζει τεχνικές προσαρμογής περιεχομένου (Content Adaptation Techniques- CATs) και τεχνικές προσαρμογής δικτύου (Network Adaptation Techniques- NATs) με τελικό στόχο να προσαρμόσει το bit stream στις μεταβαλλόμενες παραμέτρους του δικτύου. Τα NATs είναι υπεύθυνο για την end to

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

7.2 Αλγόριθμος ADIVIS

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



Σχήμα 7.1: Σύστημα Βασισμένο στο ADIVIS

Όπως είναι φανερό από την πιο πάνω διάταξη, οι δύο μηχανισμοί, ανατροφοδότησης και συγκεχυμένης απόφασης, στοχεύουν στην προσαρμογή του στρωματοποιημένου περιεχομένου στο διαθέσιμο εύρος ζώνης. Με διακεκομμένες γραμμές παρατηρούμε την ροή των πακέτων ελέγχου, ενώ με τις συμπαγείς γραμμές φαίνεται η κίνηση των πακέτων δεδομένων βίντεο.

Ο αλγόριθμος ανατροφοδότησης συλλέγει πληροφορίες σχετικά με την ποιότητα υπηρεσίας, τόσο από τον πυρήνα του δικτύου όσο και από τον παραλήπτη, οι οποίες θα χρησιμοποιηθούν για την εκτίμηση του διαθέσιμου εύρους ζώνης. Ο αλγόριθμος απόφασης που είναι υλοποιημένος στην πλευρά του αποστολέα, επεξεργάζεται τις πληροφορίες αυτές και αποφασίζει για τον βέλτιστο αριθμό στρωμάτων που θα αποσταλούν. Ο ρόλος του στοιχείων ανατροφοδότησης και προσαρμογής είναι να συνδέσουν τις απαιτήσεις ποιότητας της εφαρμογής βίντεο, με το δίκτυο πάνω από το οποίο εκτελείται. Η προσαρμογή στην κατάσταση του δικτύου, οφείλει να ακολουθείται και από προσαρμογή στο περιεχόμενο, που επιτυγχάνεται μέσα από κωδικοποίηση στρωματοποιημένου βίντεο.

7.3 Κωδικοποίηση σε Στρώματα (Layered Encoding)

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

7.4 Μηχανισμός Ανατροφοδότησης (Feedback Mechanism)

Κάθε παραλήπτης αποστέλλει στατιστικές παραλαβής χρησιμοποιώντας πακέτα RTCP. Συγκεκριμένα, τα πακέτα που στέλνονται από τους συμμετέχοντες οι οποίοι δεν είναι ενεργοί αποστολείς, είναι Receiver Reports- RR και περιέχουν στατιστικές παραλαβής. Το ποσοστό απώλειας πακέτων μέσα σε ένα χρονικό διάστημα δίνεται από τον αριθμό των αναμενόμενων πακέτων διαιρεμένο με τον αριθμό των χαμένων πακέτων μέσα σε αυτό το χρονικό διάστημα. Ο ρυθμός απώλειας ανά δευτερόλεπτο (LRPS), προκύπτει διαιρώντας το ποσοστό απώλειας, με τη διαφορά στους χρόνους(timestamps) μεταξύ των RRs. Η διαφορά ανάμεσα σε δύο διαδοχικές τιμές LRPS, μπορεί να χρησιμοποιηθεί για την ανίχνευση αυξητικής ή μειωτικής τάσης σχετικά με το ποσοστό απώλειας πακέτων.

Σαν ένα δεύτερο συστατικό του μηχανισμού αυτού, έρχεται να προστεθεί η έννοια της ρητής ειδοποίησης σχετικά με τη συμφόρηση στο δίκτυο (Explicit Congestion Notification- ECN). Σύμφωνα με την τεχνική αυτή, τα στοιχεία του δικτύου, όπως για παράδειγμα οι δρομολογητές, μπορούν να ειδοποιήσουν ρητά τον αποστολέα αναφορικά με την τρέχουσα κατάσταση της συμφόρησης εντός του πυρήνα του δικτύου, κάτι που μπορεί να χρησιμοποιηθεί αποδοτικά για την αξιολόγηση του διαθέσιμου εύρους ζώνης. Οι ενημερώσεις ECN, μπορούν επιπλέον να εφαρμοστούν ώστε να ειδοποιηθούν έγκαιρα οι τερματικοί κόμβοι και να προληφθούν μη αναγκαίες απώλειες πακέτων.

7.5 Μηχανισμός Συγκεχυμένης Απόφασης (Fuzzy Decision Mechanism)

Σαν εισαγωγή αναφέρεται ότι η συγκεχυμένη λογική χρησιμοποιείται για τον σχεδιασμό χειριστιρίων της ανατροφοδότησης σε περιπτώσεις όπου η θεωρητική προσέγγιση για απόλυτο έλεγχο δεν μπορεί να εφαρμοστεί πρακτικά, λόγω της μη δυνατότητας εξαγωγής κατάλληλου αυστηρού αναλυτικού μοντέλου.

Ο αλγόριθμος που μελετούμε, διαξάγεται στην πλαυρά του αποστολέα, ο οποίος αρχικά επεξεργάζεται την πληροφορία ανατροφοδότησης και αποφασίζει τον βέλτιστο αριθμό στρωμάτων που θα αποσταλούν, χρησιμοποιώντας, όπως προαναφέρθηκε. Συγκεχυμένη λογική εξόδο. Βασίζεται σε δύο γλωσσικές μεταβλητές εισόδου και μία μεταβλητή εξόδου. Όλες οι ποσότητες περιλαμβάνουν την διακριτή σταθερά κT όπου T ορίζεται σαν η περίοδος απόφασης.

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

$$D_{LRPS}(kT) = LRPS(kT) - LRPS(kT - T)$$

Οι τιμές της παραμέτρου LRPS κυμαίνονται από 0 έως 1 και ως εκ τούτου το κάτω και άνω φράγμα του $D_{LRPS}(kT)$ είναι -1 και 1 αντίστοιχα.

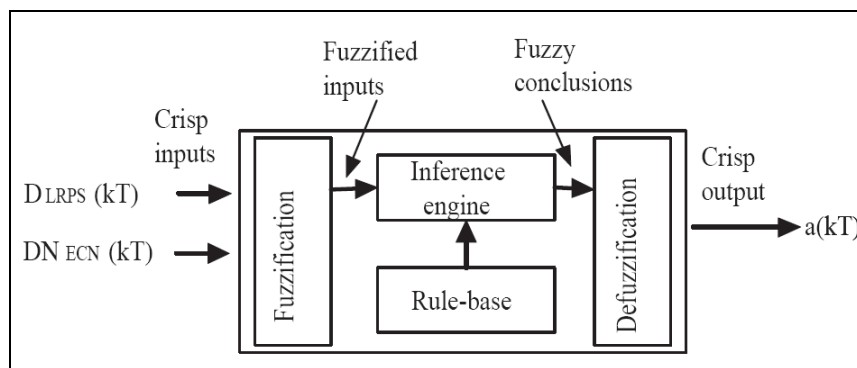
Ως δεύτερη γλωσσική μεταβλητή χρησιμοποιείται ο αριθμός των πακέτων που έχουν ενεργοποιημένο το ECN bit εντός μιας περιόδου, σαν σαφής ένδειξη της συμφόρησης. Ο παραλήπτης περιοδικά υπολογίζει τον αριθμό αυτό που αντιπροσωπεύεται από την μεταβλητή $N_{ECN}(KT)$ και τον αποστέλλει στα πλαίσια των RR πακέτων που εκπέμπει προς τον αποστολέα. Ο τελευταίος, στη συνέχεια, εξάγει την τιμή αυτή από τα RRs και υπολογίζει μία σταθμισμένη παράμετρο, το $N_{ECNsc}(kT)$, που κυμαίνεται από -1 έως +1 και αναπαριστά το ποσοστό των πακέτων που είχαν ενεργοποιημένο το ECN bit εντός μίας περιόδου. Για τον υπολογισμό της σταθμισμένης παραμέτρου χρησιμοποιείται ο ακόλουθος τύπος:

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

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

$$DN_{ECNsc}(kT) = N_{ECNsc}(kT) - N_{ECNsc}(kT - T)$$

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



Σχήμα 7.2: Σύστημα Ελέγχου Συγκεκριμένης Λογικής

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

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

Ο τύπος αυτός είναι πολύ σημαντικός στα πλαίσια της μελέτης αυτής, καθώς μπορεί να χρησιμοποιηθεί σε συνδυασμό με το εκτελέσιμο BitStreamExtractorStatic του λογισμικού JSVM, που μελετήσαμε σε προηγούμενη ενότητα, για την εξαγωγή στρωμάτων βίντεο με bit rate ίσο με το διαθέσιμο εύρος ζώνης.

Ο αλγόριθμος του ADIVIS, πρέπει να αποφασίσει ποια στρώματα θα αποσταλούν ανάλογα με το διαθέσιμο εύρος ζώνης του δικτύου, βασισμένος σε μία μη δυναμική προσέγγιση επιλογής στρωμάτων. Ο εξυπηρετητής θα έχει αποθηκευμένο ένα κατάλληλο αριθμό από

στρώματα, σε κάθε ένα εκ των οποίων αντιστοιχεί διαφορετικός ρυθμός μετάδοσης. Για την αποφυγή βεβιασμένων ενεργειών που θα απαιτούν άμεση αντιστροφή στην προηγούμενη κατάσταση, δεν πρέπει να υπάρχει άμεση μετάβαση σε ανώτερο στρώμα κάθε φορά που το διαθέσιμο εύρος ζώνης ξεπερνά το κατώφλι ενός συγκεκριμένου ρυθμού μετάδοσης, που αντιστοιχεί σε ψηλότερο στρώμα. Αντίθετα, το σύστημα θα πρέπει να μπαίνει σε μια φάση χρονικής υστέρησης, ώστε να αποφευχθούν τακτικές μεταβάσεις από ένα στρώμα σε άλλο.

Αντίθετα, στην περίπτωση μετάβασης σε χαμηλότερο στρώμα, οι ενέργειες είναι άμεσες, καθώς προσπαθούμε να ανανήψουμε από συμφόρηση όσο το δυνατό συντομότερα. Η χρονική υστέρηση είναι ίση με το χρονικό διάστημα μεταξύ της παραλαβής δύο διαδοχικών RR πακέτων. Αν το διαθέσιμο εύρος ζώνης υπερβεί το κατώφλι ενός συγκεκριμένου ρυθμού μετάδοσης που ανταποκρίνεται σε στρώμα ανώτερου επιπέδου, ενεργοποιείται η μεταβλητή της χρονικής υστέρησης. Όταν ένα νέο πακέτο RR παραληφθεί, αν το διαθέσιμο εύρος ζώνης παραμένει στα ίδια επίπεδα, τότε έχουμε μετάβαση σε ψηλότερο στρώμα. Οι πιο πάνω συνθήκες συνοψίζονται στον ακόλουθο ψευδοκώδικα που περιγράφει τη διαδικασία μετάβασης ανάμεσα σε στρώματα.

```

For all layers j up to MAX LAYER
If avail_bw <= BitRate(j) and hysteresis=false {
    If layer(j) = current_layer
        break;
    hysteresis = true;
    selected_layer = layer(j);
}
Else If avail_bw <= BitRate(j) {
    hysteresis = false;
    If selected_layer < layer(j) {
        current_layer = selected_layer;
        break;
    }
}
else { current_layer = layer(j);

```

```

break;
}
}

```

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

$a(kT)$		$DN_{ECN_{sc}}(kT)$							
		NVB	NB	NS	Z	PS	PB	PVB	
$D_{LRPS}(kT)$	NVB	H	H	B	B	Z	S	VS	
	NB	H	VB	Z	Z	Z	S	VS	
	NS	B	Z	B	Z	Z	S	VS	
	Z	B	Z	Z	B	Z	S	VS	
	PS	Z	Z	Z	Z	S	S	VS	
	PB	Z	Z	Z	Z	S	S	VS	
	PVB	S	S	S	S	VS	VS	VS	

Πίνακας 7.3: Γλωσσικοί Κανόνες

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

ΚΕΦΑΛΑΙΟ 8

JSVM (JOINT SCALABLE VIDEO MODE)

8.1 Εισαγωγή

8.2 Βιβλιοθήκες και Εκτελέσιμα

8.3 Χρησιμότητα

8.4 Κωδικοποιητής H264AVCEncoderLibTestStatic

8.5 Εξαγωγή Bitstream BitStreamExtractorStatic

8.6 Μετατροπή SVC Bitstream σε AVC AvcRewriterStatic

8.7 Μέτρηση Ποιότητας PSNRStatic

8.1 Εισαγωγή

Το JSVM είναι το λογισμικό αναφοράς σχετικά με το project Scalable Video Coding από την Joint Video Team. Καθώς το έργο SVC είναι ακόμη υπό ανάπτυξη, το ίδιο συμβαίνει και με το λογισμικό JSVM, με αποτέλεσμα να αλλάζει αρκετά συχνά, με τις αλλαγές να έχουν την μορφή διορθώσεων ή αναπτύξεων.

Το JSVM είναι γραμμένο στη γλώσσα προγραμματισμού C++, διατίθεται δωρεάν υπό την μορφή πηγαίου κώδικα και η εκάστοτε πιο πρόσφατη έκδοσή του μπορεί να ανακτηθεί από τον CVS (Concurrent Versioning System) εξυπηρετητή του Τεχνικού Πανεπιστημίου του Άαχεν. Επιπλέον, μαζί με τον κώδικα είναι δυνατή και η ανάκτηση βίντεο σε YUV μορφή τα οποία μπορούν να χρησιμοποιηθούν για δοκιμές του. Τα βίντεο αυτά προσφέρονται σε χωρική ευκρίνεια QCIF (176x144), CIF (352x288) και 4CIF (704x576) και σε framerates 7.5, 15, 30 και 60 πλαισίων ανά δευτερόλεπτο.

Σημαντικό χαρακτηριστικό του λογισμικού JSVM είναι η μεταφερσιμότητά του, πράγμα που προσφέρει την δυνατότητα να τρέχει τόσο σε λειτουργικό σύστημα Windows, όσο και σε Linux.

Μετά την μεταγλώττιση του κώδικα του λογισμικού δημιουργούνται οι καταλόγοι που περιέχουν τις διάφορες αναγκαίες βιβλιοθήκες, τα εκτελέσιμα αρχεία που εκτελούν διάφορες λειτουργίες, όπως κωδικοποίηση σε scalable video, downsampling, εξαγωγής επιλεγμένων στρωμάτων από ένα στρωματοποιημένο βίντεο και μετατροπής μίας ακολουθίας από SVC σε AVC, μεταξύ άλλων. Τα πιο σημαντικά εκτελέσιμα αρχεία, τα οποία και θα χρησιμοποιηθούν στα πλαίσια της μελέτης αυτής θα παρουσιαστούν και θα εξηγηθούν με περισσότερη λεπτομέρεια στη συνέχεια.

Για περαιτέρω λεπτομέρειες σχετικά με τον τρόπο ανάκτησης του πηγαίου κώδικα, τον τρόπο μεταγλώττισης και τις πιθανές παραμέτρους της γραμμής εντολής για κάθε ένα από τα πολλά εκτελέσιμα αρχεία που παρέχουν πλειάδα δυνατοτήτων όσο αφορά την εξαγωγή και επεξεργασία scalable βίντεο, ο αναγνώστης προτρέπεται να διαβάσει το εγχειρίδιο χρήσης του εργαλείου JSVM, στο οποίο μπορούν να βρεθούν και πολλά παραδείγματα εκτέλεσης.

8.2 Βιβλιοθήκες και Εκτελέσιμα

Στον ακόλουθο πίνακα, από το εγχειρίδιο χρήσης του JSVM, συνοψίζονται οι βιβλιοθήκες που περιέχονται και είναι αναγκαίες για την εκτέλεσή του, μαζί με μία περιγραφή όσο αφορά την λειτουργία των κλάσεων που τις αποτελούν.

Βιβλιοθήκη	Περιγραφή
H264AVCCommonLibStatic	common lib Παρέχει κλάσεις που χρησιμοποιούνται τόσο από τον κωδικοποιητή, όσο και από τον αποκωδικοποιητή, όπως για παράδειγμα δομές δεδομένων για τα macroblocks και προσωρινές μνήμες για πρόσβαση σε δεδομένα εικόνων.
H264AVCEncoderLibStatic	encoder lib Παρέχει κλάσεις αποκλειστικά για τον κωδικοποιητή, όπως για παράδειγμα κλάσεις για εκτίμηση κίνησης και κωδικοποίηση εντροπίας.

H264AVCDecoderLibStatic	decoder lib Εδώ παρέχονται κλάσεις που χρησιμοποιούνται μόνο από τον αποκωδικοποιητή, όπως κλάσεις για αποκωδικοποίηση εντροπίας και ανάγνωση-ανάλυση του bit-stream.
AvcRewriterLibStatic	SVC to AVC rewriter lib Η βιβλιοθήκη αυτή αποτελείται από τα ίδια αρχεία πηγαίου κώδικα με την H264AVCDecoderLibStatic, αλλά όταν μεταγλωττίζεται με την παράμετρο SHARP_AVC_REWRITE_OUTPUT, παρέχει κλάσεις που χρησιμοποιούνται μόνο από το εκτελέσιμο AvcRewriter.
H264AVCVideoIoLibStatic	io lib Χρησιμοποιείται για την ανάγνωση και εγγραφή μονάδων NAL στο byte-stream, όπως επίσης και για ανάγνωση και γραφή δεδομένων raw video.

Πίνακας 8.1: Βιβλιοθήκες JSVM και περιγραφή τους

Ο επόμενος πίνακας συνοψίζει την γκάμα των εκτελέσιμων αρχείων που παρέχονται στα πλαίσια του JSVM, μαζί με περιγραφή της λειτουργικότητάς τους, καθώς και παραπομπή στον αντίστοιχο τομέα του εγχειριδίου χρήσης (Έκδοση 9.17), όπου παρουσιάζεται πλήρως ο τρόπος εκτέλεσης μαζί με ανάλογα παραδείγματα.

Εκτελέσιμο Αρχείο	Περιγραφή
DownConvertStatic	resampler Μπορεί να χρησιμοποιηθεί για χωρική ή χρονική επαναδειγματοληψία (upsampling ή downsampling) των ακολουθιών βίντεο. Τομέας 2.1.
H264AVCEncoderLibTestStatic	AVC/SVC encoder Χρησιμοποιώντας τον κωδικοποιητή, δίνεται η δυνατότητα παραγωγής bit-streams μονού στρώματος(AVC) ή scalable(SVC). Τομέας 2.2.
H264AVCDecoderLibTestStatic	SVC decoder Μέσω του εκτελέσιμου αυτού είναι δυνατή η αποκωδικοποίηση AVC ή SVC bit-streams, και η δημιουργία ακολουθιών raw βίντεο. Τομέας 2.3.
AvcRewriter	SVC to AVC rewriter Μπορεί να χρησιμοποιηθεί για την μετατροπή των SVC bit-streams σε AVC bit-streams, μόνο αν η

	μεταβλητή <code>avc_rewrite_flag</code> χρησιμοποιείται στα ενισχυτικά στρώματα. Τομέας 2.4.
BitStreamExtractorStatic	bit-stream extractor Ο bit-stream extractor χρησιμοποιείται για την εξαγωγή υπο-ροών με χαμηλότερη χωρική ή χρονική ευκρίνεια, ή και bit rate από ένα scalable bit-stream. Τομέας 2.5.
QualityLevelAssignerStatic	quality level assigner Δεδομένου ενός scalable bit-stream, μπορεί να δημιουργήσει ένα νέο bit stream με επιπλέον στρώματα ποιότητας. Πέραν του παράγοντα της ποιότητας τα δύο bit streams είναι τα ίδια. Τομέας 2.6.
MCTFPreProcessor	MCTF pre-processor Ο προεπεξεργαστής MCTF μπορεί να χρησιμοποιηθεί για το προ-φιλτράρισμα ακολουθιών από εικόνες. Τομέας 2.7..
PSNRStatic	PSNR tool Μέσω του εργαλείου PSNR είναι δυνατή η μέτρηση του PSNR(Peak Signal to Noise Ratio) ανάμεσα σε δύο ακολουθίες raw βίντεο. Επιπλέον είναι σε θέση να μετρήσει το bitrate ενός bit stream που δίνεται σαν είσοδος. Τομέας 2.7.
FixedQPEncoderStatic	fixed QP encoder Επιτρέπει τον έλεγχο του κωδικοποιητή και μέσω αυτού μπορούμε να καθορίσουμε το bit rate του bit-stream που θέλουμε να εξάγουμε. Τομέας 2.9.
SIPAnalyser	SIP Analyser Το εργαλείο SIPAnalyser χρησιμοποιείται για την απόφαση σχετικά με την επιλεκτική διαστρωματική πρόβλεψη. Τομέας 2.10.
YUVCompareStatic	tool for comparing YUV sequences Χρήσιμο για την εύρεση ασυνέπειας συσχέτισης μεταξύ κωδικοποιητή και αποκωδικοποιητή. Εκτελώντας το αρχείο αυτό χωρίς παραμέτρους, παρέχονται πληροφορίες σχετικά με τη χρήση του.

Πίνακας 8.2: Εκτελέσιμα αρχεία JSVM και περιγραφή τους

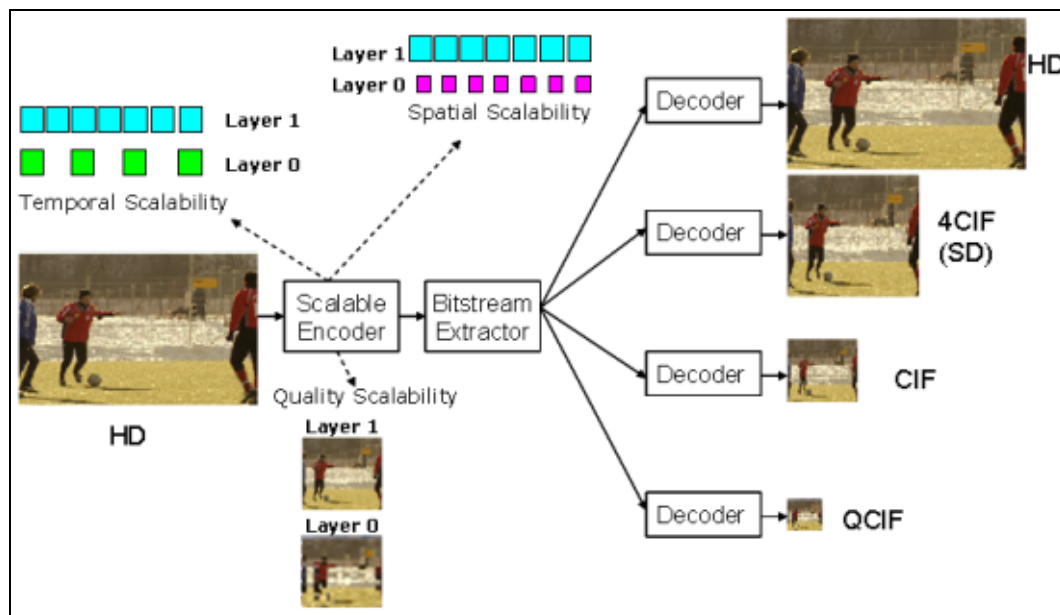
8.3 Χρησιμότητα

Σκοπός του JSVM είναι να προσφέρει μία εύχρηστη και φιλική προς τον χρήστη διαπροσωπεία για διεκπεραίωση των λειτουργιών που προσφέρονται μέσα από το πρότυπο H.264/SVC. Με τη χρήση του εργαλείου αυτού είναι δυνατή η κωδικοποίηση μίας ακολουθίας βίντεο σε στρωματοποιημένο βίντεο, η εξαγωγή bit-stream στο οποίο να περιέχονται συγκεκριμένα στρώματα ή και bit stream το οποίο να απαιτεί συγκεκριμένο εύρος ζώνης, ανάλογα με το επιλεγμένο bit rate, για να μεταδοθεί πάνω από κάποιο δίκτυο, καθώς επίσης προσφέρει και πλειάδα άλλων εξειδικευμένων επιλογών.

Με τη χρήση του κωδικοποιητή είναι δυνατή η δημιουργία πολλαπλών διαφορετικών υποροών από ένα συγκεκριμένο βίντεο, κάθε μία από αυτές με διαφορετικό bit rate, καθώς επωφελείται του χρονικού και χωρικού πλεονασμού. Οι παραγόμενες υποροές είναι δυνατό να διαφέρουν στον αριθμό πλαισίων ανά δευτερόλεπτο, στην χωρική ευκρίνεια ή και στην τελική ποιότητα.

Όσο αφορά τον χρονικό πλεονασμό, το JSVM δομεί τις εξαρτήσεις κατά την αντιστάθμιση κίνησης με τέτοιο τρόπο, ώστε ολόκληρες εικόνες να θεωρούνται περιττές για την αποκωδικοποίηση της ακολουθίας. Σχετικά με τον χωρικό πλεονασμό, τα δεδομένα και αποκωδικοποιημένα δείγματα χαμηλότερων ευκρινειών, είναι δυνατό να χρησιμοποιηθούν κατά την αποκωδικοποίηση εικόνων που αντιστοιχούν σε υψηλότερη ευκρίνεια. Το ίδιο συμβαίνει και όσο αφορά την ποιότητα(ή SNR), όπου εικόνες με χαμηλότερη ποιότητα, είναι δυνατό να χρησιμοποιηθούν για την αποκωδικοποίηση εικόνων σε στρώματα με υψηλότερη ποιότητα.

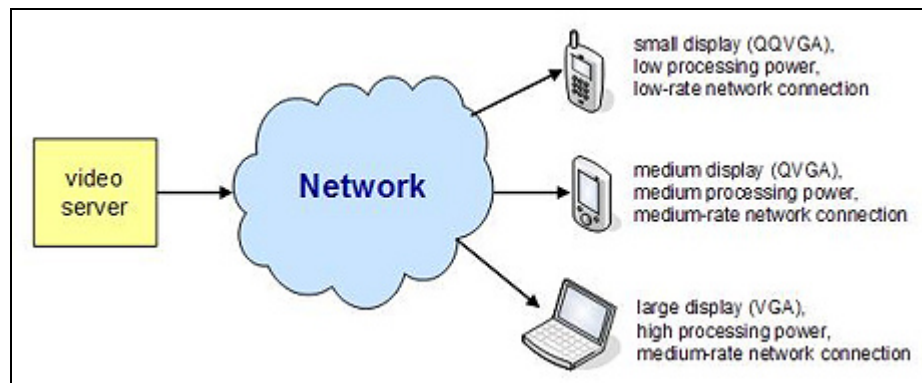
Τα πιο πάνω δίνουν την δυνατότητα αποθήκευσης ενός μόνο βίντεο, και ανάλογα με τον τελικό χρήστη που πρόκειται να το λάβει, ή ανάλογα με τις συνθήκες που επικρατούν την εκάστοτε στιγμή στο δίκτυο πάνω από το οποίο θα γίνει η μεταφορά του, να επιλεγθεί ο κατάλληλος αριθμός στρωμάτων, ώστε η αντιληπτή ποιότητα από τον χρήστη να είναι η υψηλότερη δυνατή. Εφαρμογές για τις οποίες θεωρείται ιδανική η πιο πάνω λύση αποτελούν η τηλεδιάσκεψη, το video streaming και τα κλειστά κυκλώματα παρακολούθησης μεταξύ άλλων.



Σχήμα 8.4: Παράδειγμα Κωδικοποίησης Scalable Video

Στην πιο πάνω εικόνα παρουσιάζονται και οι τρεις τύποι Scalability που προσφέρονται από το SVC μέσω του JSVM. Συγκεκριμένα, παρατηρούμε ότι δίνοντας σαν είσοδο στον κωδικοποιητή μία εικόνα σε υψηλή ευκρίνεια, είναι δυνατή η εξαγωγή ακολουθίας bit stream που να ανταποκρίνεται σε βίντεο χαμηλότερης ποιότητα, αριθμού πλαισίων ανά δευτερόλεπτο και χωρικής ευκρίνειας(διαστάσεων). Διαγραμματικά, παρουσιάζεται και η έννοια του στρώματος, αφού βλέπουμε ότι στρώματα με χαμηλότερο αριθμό στρώματος παρουσιάζουν είτε χαμηλότερο framerate, είτε χαμηλότερη ευκρίνεια σε σχέση με στρώματα με μεγαλύτερο αριθμό. Ουσιαστικά, στο χρονικό scalability βλέπουμε ότι ο αριθμός των εικόνων που κωδικοποιούνται είναι μικρότερος, ενώ στο χωρικό μειώνονται οι διαστάσεις του στρώματος με αριθμό 0.

Έχοντας στην διάθεσή μας αυτές τις επιλογές, είναι δυνατός ο καθορισμός της μορφής του βίντεο που θα επιλέξουμε να μεταφέρουμε μέσω δικτύου λαμβάνοντας υπόψη την ετερογένεια που παρουσιάζεται ανάμεσα στις συσκευές των παραληπτών, την υπολογιστική τους ισχύ και τις δυνατότητες του δικτύου. Ένα παράδειγμα αυτού του σεναρίου παρουσιάζεται πιο κάτω, όπου και οι τρεις παράγοντες χρησιμοποιούνται για την αποστολή βίντεο σε τρεις διαφορετικούς παραλήπτες, το οποίο είναι αποθηκευμένο στην πλευρά του αποστολέα σε διαφορετικές μορφές που θα χρησιμοποιηθούν για την στρωματοποίηση και την εξαγωγή των απαραίτητων στρωμάτων.



Σχήμα 8.5: Αξιοποίηση Στρωματοποιημένου Βίντεο για Επίλυση Ετερογένειας

Σε αντίθεση με πιο πάνω, οι παράγοντες οι οποίοι θα μας επηρεάσουν, στα πλαίσια της μελέτης αυτής, στην επιλογή των κατάλληλων στρωμάτων που θα αποσταλούν είναι η συμφόρηση στο δίκτυο, καθώς και η ποιότητα την οποία αντιλαμβάνεται ο χρήστης, μέσω της περιοδικής καταμέτρησης του αριθμού των χαμένων πακέτων.

8.4 Κωδικοποιητής H264AVCEncoderLibTestStatic

Για την κωδικοποίηση μίας ακολουθίας βίντεο σε στρωματοποιημένο SVC βίντεο χρησιμοποιείται το εκτελέσιμο H264AVCEncoderLibTestStatic που προσφέρεται από το λογισμικό JSVM. Μπορεί επίσης να χρησιμοποιηθεί για την παραγωγή AVC βίντεο.

Η επιλογή των παραμέτρων με τις οποίες εκτελείται ο κωδικοποιητής επηρεάζει σε μεγάλο βαθμό την αποδοτικότητα της κωδικοποίησης και είναι δυνατό να αυξήσει αρκετά τον απαιτούμενο για τη λειτουργία αυτή χρόνο.

Το αρχείο H264AVCEncoderLibTestStatic μπορεί να τρέξει με δύο τρόπους κωδικοποίησης, είτε για την παραγωγή βίντεο ενός στρώματος, είτε για παραγωγή scalable βίντεο. Ωστόσο, είναι σε θέση να παράγει bit stream ενός στρώματος, ακόμα και όταν τρέξει υπό τη μορφή κωδικοποίησης scalable βίντεο. Η ρύθμιση για μονό στρώμα παρέχει περισσότερη ευελιξία σε σχέση με τα πολλαπλά στρώματα, εντούτοις δεν είναι δυνατή η παραγωγή scalable βιντεοροής.

Για την εκτέλεση του κωδικοποιητή με οποιοδήποτε από τους δύο τρόπους, απαιτείται η δημιουργία τουλάχιστον ενός αρχείου ρυθμίσεων, αν και στην περίπτωση που επιθυμούμε να παράγουμε scalable video, επιπλέον αρχεία ρυθμίσεων πρέπει να καθοριστούν, τα οποία

αναφέρονται στα ενισχυτικά στρώματα. Ο αριθμός των παραμέτρων που μπορούν να χρησιμοποιηθούν είναι μεγάλος, όμως οι σημαντικότερες από αυτές στο κύριο αρχείο ρυθμίσεων είναι το όνομα του αρχείου εισόδου, οι διαστάσεις του, ο αριθμός πλαισίων ανά δευτερόλεπτο στο αρχείο εισόδου και πόσα αυτά επιθυμούμε να κωδικοποιηθούν, το μέγεθος του Group of Pictures(σύνολο εικόνων μεταξύ δύο διαδοχικών πλαισίων I), το όνομα του αρχείου στο οποίο θα αποθηκευτεί το bit stream που θα παραχθεί, ο αριθμός των επιπλέον στρωμάτων και τα ονόματα των αρχείων ρυθμίσεων για κάθε ένα από αυτά τα στρώματα. Κάθε στρώμα αντιπροσωπεύει είτε ένα χωρικό στρώμα ή ένα στρώμα ποιότητας fine grain SNR scalability. Τον τρόπο με τον οποίο θα τρέξει ο κωδικοποιητής, δηλαδή για δημιουργία μονού στρώματος ή scalable bit stream, καθορίζει η παράμετρος AVCMODE. Στα αρχεία ρυθμίσεων των στρωμάτων είναι απαραίτητη η δήλωση των διαστάσεων του αρχείου εισόδου, το όνομά του, ο αριθμός των πλαισίων του ανά δευτερόλεπτο και ο αριθμός των πλαισίων ανά δευτερόλεπτο που θα περιέχονται στην κωδικοποιημένη ακολουθία, ο τύπος της διαστρωματικής πρόβλεψης που θα χρησιμοποιηθεί και η παράμετρος κβαντοποίησης που χρησιμοποιείται για την εκτίμηση κίνησης και τον καθορισμό των key pictures.

Κάθε παράμετρος πρέπει να είναι γραμμένη σε μία γραμμή του κάθε αρχείου ρυθμίσεων. Για κάθε παράμετρο υπάρχει εξ ορισμού τιμή, η οποία και χρησιμοποιείται σε περίπτωση που δεν αναφέρεται ρητά νέα τιμή στα αρχεία ρυθμίσεων. Ως εκ τούτου δεν είναι αναγκαίος ο καθορισμός τιμής για όλες τις παραμέτρους που αναφέρονται στο εγχειρίδιο χρήσης του JSVM.

Με το πέρας της διεκπεραίωσης της κωδικοποίησης εμφανίζεται στην οθόνη η ακόλουθη έξοδος, όπου παρουσιάζονται πληροφορίες σχετικά με κάθε ζεύγος χωρικής- χρονικής ευκρίνειας. Η πιο σημαντική παράμετρος που παρατηρείται είναι το bitrate, το οποίο και καθορίζει το αναγκαίο εύρος ζώνης για αποστολή του βίντεο μέσω δικτύου.

>H264AVCEncoderLibTestStatic -pf main.cfg						
SUMMARY:						
		bitrate	Min-bitr	Y-PSNR	U-PSNR	V-PSNR
		-----	-----	-----	-----	-----
176x144 @	1.8750	93.4125	47.0745	38.5063	42.4337	43.8098
176x144 @	3.7500	127.9137	64.0311	37.1204	42.0545	43.4157
176x144 @	7.5000	165.0995	81.1974	35.9945	41.7574	43.1639
176x144 @	15.0000	210.6160	101.4816	35.1407	41.6518	43.0275

352x288 @ 1.8750	404.2635	188.1210	38.7535	42.7444	44.3954
352x288 @ 3.7500	547.1179	252.8147	37.1742	42.1826	44.0011
352x288 @ 7.5000	703.0626	317.7947	35.9920	41.8283	43.7475
352x288 @ 15.0000	895.7664	396.9712	35.1249	41.6106	43.5677
352x288 @ 30.0000	1065.7728	475.8656	34.4091	41.3834	43.3742

Πίνακας 8.1: Έξοδος Εκτέλεσης H264EncoderLibTestStatic

Στο πιο πάνω παράδειγμα, ο κωδικοποιητής έτρεξε με ρύθμιση για scalable βίντεο. Παρατηρούμε ότι όσο αυξάνεται η χωρική και χρονική ευκρίνεια, αναλόγως αυξάνεται και το bit rate του bit stream που παράγεται. Τα βίντεο που πέρασαν σαν είσοδος ήταν η ακολουθία BUS σε δύο μορφές. Η πρώτη με διαστάσεις 176x144 και 15 πλαίσια ανά δευτερόλεπτο, και η δεύτερη με διαστάσεις 352x288 και 30 πλαίσια ανά δευτερόλεπτο.

Για να δούμε τα στρώματα που παράχθηκαν και είναι δυνατό να εξαχούν σαν υποροές, πρέπει να εκτελέσουμε την εντολή BitStreamExtractorStatic και να δώσουμε σαν παράμετρο, μόνο το όνομα του αρχείου που έδωσε σαν έξοδο το εκτελέσιμο H264EncoderLibTestStatic. Παράδειγμα αυτής της εκτέλεσης παρουσιάζεται στη συνέχεια, με είσοδο το αρχείο που παρήχθηκε στο προηγούμενο παράδειγμα.

```
>BitStreamExtractorStatic test.264
```

Contained Layers:
=====

Layer	Resolution	Framerate	Bitrate	MinBitrate	DTQ
0	176x144	1.8750	47.10	47.10	(0,0,0)
1	176x144	3.7500	64.00	64.00	(0,1,0)
2	176x144	7.5000	81.20	81.20	(0,2,0)
3	176x144	15.0000	101.50	101.50	(0,3,0)
4	176x144	1.8750	93.40		(0,0,1)
5	176x144	3.7500	127.90		(0,1,1)
6	176x144	7.5000	165.10		(0,2,1)
7	176x144	15.0000	210.60		(0,3,1)
8	352x288	1.8750	234.50	188.20	(1,0,0)
9	352x288	3.7500	316.70	252.80	(1,1,0)
10	352x288	7.5000	401.70	317.80	(1,2,0)
11	352x288	15.0000	506.10	397.00	(1,3,0)
12	352x288	30.0000	585.00	475.90	(1,4,0)
13	352x288	1.8750	404.30		(1,0,1)
14	352x288	3.7500	547.10		(1,1,1)
15	352x288	7.5000	703.10		(1,2,1)
16	352x288	15.0000	895.80		(1,3,1)
17	352x288	30.0000	1065.80		(1,4,1)

Πίνακας 8.2: Έξοδος Εκτέλεσης BitStreamExtractorStatic

Εδώ παρουσιάζονται τα στρώματα που μπορούμε να εξαγάγουμε, μαζί με τον αριθμό στρώματος. Η τριάδα DTQ στην τελευταία στήλη παρουσιάζει το επίπεδο εξάρτησης της

υποροής, το επίπεδο χρονικής ευκρίνειας και το επίπεδο ποιότητας. Ο τελευταίος παράγοντας είναι και αυτός που προσφέρει περισσότερα αποτελέσματα σε αυτή την έξοδο, παρά στην προηγούμενη, όπου δεν υπήρχε αναφορά σε επίπεδα ποιότητας του bit stream.

8.5 Εξαγωγή Bitstream BitStreamExtractorStatic

Σκοπός του εργαλείου αυτού είναι να επιτρέψει την εξαγωγή υποροών από μία ροή bit σε μορφή AVC ή SVC. Οι εξαγόμενες υποροές έχουν μειωμένη χρονική ή και χωρική ευκρίνεια σε σύγκριση με την αρχική ροή, καθώς και πιθανώς χαμηλότερο επίπεδο ποιότητας, ανάλογα με την κωδικοποίηση που προηγήθηκε. Σε κάθε περίπτωση, μειώνοντας κάποιο από τους πιο πάνω παράγοντες, αυτόματα μειώνεται και το bitrate της υποροής.

Οι επιλογές που προσφέρει όσο αφορά τα κριτήρια που μπορούν να τεθούν για εξαγωγή υποροών είναι πολλά. Τα σημαντικότερα από αυτά όμως είναι ο καθορισμός του επιπέδου στρώματος που θέλουμε να εξαχθεί, με το οποίο ταυτόχρονα εξάγονται και όλα από όσα εξαρτάται, η εξαγωγή όλων των στρωμάτων με συγκεκριμένη χωρική ή χρονική ευκρίνεια ή εξαγωγή βάσει επιπέδου ποιότητας. Μία από τις σημαντικότερες όμως δυνατότητες που προσφέρει είναι η δήλωση των διαστάσεων, του αριθμού πλαισίου ανά δευτερόλεπτο και του bit rate του bitstream που επιθυμούμε να εξάγουμε. Σε τέτοια περίπτωση ο BitStreamExtractorStatic θα ψάξει για τέτοιο στρώμα και σε περίπτωση που υπάρχει κάποιο με μεγαλύτερο bit rate σε σχέση με αυτό που εμείς δηλώσαμε σαν παράμετρο, θα το μειώσει και θα το εξάγει.

Πέρα από αυτή τη χρήση, βέβαια, το BitStreamExtractorStatic χρησιμοποιείται περνώντας απλά σαν παράμετρο το όνομα ενός bit stream μπορεί να μας παρουσιάσει πληροφορίες σχετικά με αυτό και όλα τα πιθανά στρώματα που το αποτελούν, όπως είδαμε και στην αμέσως προηγούμενη παράγραφο.

Τρέχοντας το εκτελέσιμο αυτό με επιλογή `-sl <αριθμός στρώματος>` εξάγεται το στρώμα με ταυτότητα `<αριθμός στρώματος>` και όλα τα χαμηλότερα στρώματα από τα οποία αυτό εξαρτάται. Με την επιλογή `-b <target bit rate>` εξάγεται το στρώμα, πιθανά τροποποιημένο, το οποίο έχει bit rate ίσο με `<target bit rate>`. Τέλος, σημαντική για την μελέτη αυτή είναι η ακόλουθη εκτέλεση: `BitStreamExtractorStatic -e AxB@C:D`.

Χρησιμοποιώντας αυτή την επιλογή μπορούμε να εξάγουμε ένα στρώμα, πιθανά και πάλι τροποποιημένο, με χωρική ευκρίνεια $A \times B$, αριθμό πλαισίων ανά δευτερόλεπτο ίσο με C , και το οποίο να έχει bit rate ίσο με D . Η εντολή αυτή είναι ιδιαίτερα σημαντική, όταν δεδομένων των προτιμήσεων του χρήστη, βάση της συσκευής στην οποία επιθυμεί να παραλάβει το βίντεο, ξέρουμε την χωρική ευκρίνεια του βίντεο που θα αποστείλουμε και ακολούθως λαμβάνοντας μετρήσεις από το δίκτυο, μπορούμε να καθορίσουμε και το bit rate του και να το εξάγουμε από το bit stream χωρίς επιπλέον καθυστέρηση.

8.6 Μετατροπή SVC Bitstream σε AVC AvcRewriterStatic

Το εκτελέσιμο αυτό παρέχει την δυνατότητα μετατροπής ενός bit stream που έχει κωδικοποιηθεί σε SVC, σε AVC bit stream. Για να επιτευχθεί αυτό πρέπει η παράμετρος `avc_rewrite_flag` να χρησιμοποιείται στο ψηλότερο ενισχυτικό στρώμα, ενώ πρέπει παράλληλα να είναι ίση με μηδέν στο χαμηλότερο στρώμα. Δηλαδή, πρέπει να χρησιμοποιηθεί η παράμετρος αυτή κατάλληλα στα αρχεία ρυθμίσεων που θα συνταχθούν για την παραγωγή ενός SVC bit stream.

Εξάλλου, μετά από διάφορες δοκιμές της λειτουργίας αυτής, ανακαλύφθηκε ότι δεν είναι επιτρεπτή η χρήση της λειτουργίας αυτής, όταν η παράμετρος αυτή χρησιμοποιηθεί σε αρχεία ρυθμίσεων που αξιοποιούν το χωρικό scalability. Είναι εφικτό δηλαδή να χρησιμοποιηθεί μόνο σε στρώματα τα οποία έχουν την ίδια χωρική ευκρίνεια.

Με τη χρήση της λειτουργίας αυτής, καθίσταται πλεονάζουσα η διαδικασία μετατροπής του κωδικοποιημένου stream σε άλλη μορφή κωδικοποίησης (transcoding), μέσω άλλης εφαρμογής, πράγμα που θα πρόσθετε επιπλέον καθυστέρηση στην περίπτωση εφαρμογών πραγματικού χρόνου, εάν ο αξιολογητής μέσω του οποίου μεταδίδεται το βίντεο δεν υποστηρίζει μετάδοση ροών SVC.

8.7 Μέτρηση Ποιότητας PSNRStatic

Πρόκειται για ένα σημαντικό εργαλείο στα πλαίσια του λογισμικού JSVM. Χρησιμοποιείται για την μέτρηση του PSNR (Peak Signal to Noise Ratio) μεταξύ μίας αρχικής ακολουθίας και μιας αποκωδικοποιημένης ακολουθίας, οι οποίες δίνονται σαν

παραμέτροι σε αυτό. Σε αυτή την περίπτωση, το σήμα (signal) είναι τα δεδομένα της αρχικής ακολουθίας, ενώ ο θόρυβος είναι τα σφάλματα που προκύπτουν μετά από τη συμπίεσή της. Χρησιμοποιείται σαν μια προσέγγιση της ανθρώπινης αντίληψης σχετικά με την ανοικοδομημένη ακολουθία. Στα πλαίσια του JSVM, εξάλλου, μπορεί να χρησιμοποιηθεί και για μέτρηση του bit rate μίας ακολουθίας.

Για την αρχική του χρήση, την σύγκριση μεταξύ δύο ακολουθιών, η χωρική τους ευκρίνεια θα πρέπει να είναι η ίδια, ενώ για την χρονική ευκρίνεια είναι δυνατή η ρητή αναφορά μεταβλητής downsampling, αν η ανοικοδομημένη ακολουθία έτυχε δειγματοληψίας σε χαμηλότερο ρυθμό. Εξ ορισμού, το PSNR κάθε πλαισίου εμφανίζεται στον περιγραφέα αρχείου stdout, ενώ το μέσο PSNR όλων των πλαισίων, γράφεται στον περιγραφέα αρχείου stderr.

Το bit rate μίας ακολουθίας μπορεί επίσης να εμφανιστεί σαν έξοδος όταν οριστεί σαν είσοδος ένα αρχείο bit stream και ένας λόγος πλαισίου.

```
PSNRStatic 176 144 org.yuv rec.yuv 0 0 str.svc 15 2>PSNR.txt
type PSNR.txt
128,00 32,23 38,79 39.02
```

Πίνακας 8.3: Παράδειγμα Εκτέλεσης Εργαλείου PSNRStatic

Στο πιο πάνω παράδειγμα δίνονται σαν παραμέτροι οι διαστάσεις των δύο ακολουθιών και τα ονόματά τους, πρώτα της αρχικής και ακολούθως της ανοικοδομημένης. Στη συνέχεια αναφέρεται ότι δεν έγινε χρονικό downsampling και ότι όλα τα πλαίσια πρέπει να ληφθούν υπόψη στον υπολογισμό. Με αυτό τον τρόπο υπολογίζεται το PSNR των δύο ακολουθιών και το bit rate του str.svc το οποίο έχει λόγο πλαισίων 15 πλαίσια ανά δευτερόλεπτο. Στο αρχείο PSNR.txt καταγράφεται το bit rate, το Y-PSNR που αντιστοιχεί στο στοιχείο luminance, το U-PSNR που αντιστοιχεί στο chrominance στοιχείο U και τέλος το V-PSNR που αντιστοιχεί στο chrominance στοιχείο V.

ΚΕΦΑΛΑΙΟ 9

ΣΥΣΤΗΜΑ ΜΕΤΑΔΟΣΗΣ ΠΡΟΣΑΡΜΟΣΤΙΚΟΥ ΒΙΝΤΕΟ ΓΙΑ ΧΡΗΣΗ ΣΕ ΚΙΝΗΤΑ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ

9.1 Προτεινόμενη Λύση

9.2 Προτεινόμενος Αλγόριθμος

9.3 Σχηματική Αναπαράσταση Προτεινόμενης Λύσης

9.4 Ακολουθίες Video που Χρησιμοποιήθηκαν

9.5 Εξαγωγή Αποτελεσμάτων

9.6 Παρατηρήσεις

9.1 Προτεινόμενη Λύση

Έπειτα από μελέτη πολλών προτύπων σχετικά με τα ασύρματα δίκτυα, τον τρόπο λειτουργίας τους και τις προκλήσεις που τα συνοδεύουν, πρωτοκόλλων σχετικά με την μεταφορά πολυμέσων μέσω του Διαδικτύου και ενδελεχή έρευνα σχετικά με το ψηφιακό βίντεο, τις τεχνικές κωδικοποίησης και συμπίεσης, καθώς επίσης και ενημέρωση γύρω από τα πρότυπα κωδικοποίησης ψηφιακού βίντεο και συγκεκριμένα του H.264, κατέληξα στην λύση που θεωρώ καταλληλότερη για επίτευξη του στόχου της έρευνας αυτής. Συγκεκριμένα, για την ανάπτυξη ενός συστήματος το οποίο να είναι σε θέση να προσαρμόζει την ροή ψηφιακού βίντεο που αποστέλλεται μέσω ασύρματου δικτύου, επιλέχθηκε η υλοποίηση του συστήματος ADIVIS και η δημιουργία διαπροσωπείας μεταξύ αυτού και του λογισμικού αναφοράς του προτύπου H.264, του JSVM. Και τα δύο αυτά συστατικά στοιχεία και οι δυνατότητές τους, παρουσιάζονται λεπτομερώς στα αμέσως προηγούμενα κεφάλαια.

Αρχικά, σημαντικό ρόλο στην επιλογή του JSVM, έπαιξε το γεγονός ότι παρέχει όλες τις απαραίτητες λειτουργίες όσο αφορά τη δημιουργία στρωματοποιημένου βίντεο και την

εξαγωγή της κατάλληλης υποροής που περιέχει τον ιδανικό αριθμό στρωμάτων ανάλογα με διάφορες παραμέτρους που εμείς επιλέγουμε. Επιπλέον, πρόκειται για λογισμικό το οποίο διατίθεται δωρεάν και συνεχώς αναβαθμίζεται, με αποτέλεσμα να είναι αρκετά πιθανή η προς τα πίσω συμβατότητα των μελλοντικών εκδόσεων με την τρέχουσα έκδοση που θα χρησιμοποιηθεί, ενώ παράλληλα θα προστίθενται επιπλέον λειτουργίες. Πέραν των πιο πάνω, τρίτος παράγοντας που λειτούργησε θετικά στην επιλογή του λογισμικού αυτού είναι η ευκολία χρήσης του, καθώς αν και χρησιμοποιεί στυλ αλληλεπίδρασης γραμμής εντολών, εντούτοις τόσο κατά την εκτέλεση, όσο και στο εγχειρίδιο χρήσης, οι οδηγίες είναι αρκετά κατανοητές για αναγνώστη με γνώσεις όσο αφορά την επέκταση Scalable Video Coding του προτύπου H.264.

Προχωρώντας σε πιο τεχνικές λεπτομέρειες, ο μεγάλος χρόνος που απαιτείται κατά την κωδικοποίηση των αρχείων που δίνουμε σαν είσοδο σε μορφή scalable bit stream δεν θεωρείται σημαντικό μειονέκτημα, ούτε και αποτελεί αποτρεπτικό παράγοντα για επιλογή του JSVM, αφού με βάση τις προδιαγραφές του ADIVIS, το βίντεο βρίσκεται ήδη κωδικοποιημένο σε στρωματοποιημένη μορφή στην πλευρά του αποστολέα, και ουσιαστικά αυτό που απαιτείται να λαμβάνει χώρα σε πραγματικό χρόνο είναι η μετάβαση σε διαφορετικό στρώμα ανάλογα με την ανατροφοδότηση που παρέχουν δίκτυο και παραλήπτης. Η λειτουργία αυτή, δηλαδή της εξαγωγής των απαραίτητων στρωμάτων είναι αρκετά γρήγορη, ενώ μέσω των παραμέτρων που δίνουμε στο εκτελέσιμο πρόγραμμα που υλοποιεί τη λειτουργία αυτή, μπορούμε να επιλέξουμε εξαγωγή στρωμάτων μόνο στηριζόμενοι στο bitrate που θέλουμε να έχει η εξαγόμενη υποροή, ή να διατηρούμε σταθερά κατά τη διάρκεια ολόκληρου του streaming τα χαρακτηριστικά χωρικής ευκρίνειας και frame rate μεταβάλλοντας μόνο το bitrate, ανάλογα με επιλογές του χρήστη που καθορίζονται από τη στιγμή της αίτησης.

Μία επιπλέον ελκυστική δυνατότητα του JSVM, είναι η ευχέρεια που μας παρέχει για επανεγγραφής μίας ροής SVC σε AVC ροή, αφαιρώντας χαρακτηριστικά από την πρώτη, δεδομένου ότι οι απαραίτητες ρυθμίσεις χρησιμοποιήθηκαν κατά τη διαδικασία κωδικοποίησης του SVC bitstream. Η μορφή αυτή του bit stream καθιστά περιττή την ύπαρξη scalable αποκωδικοποιητή από πλευράς του παραλήπτη, ο οποίος με τη σειρά του μπορεί να αναπαράξει το video που λαμβάνει, μετά από αποκωδικοποίηση, με ένα συμβατό media player. Εξάλλου, η λύση αυτή είναι πολύ πιο αποδοτική από την εναλλακτική, που

θα ήταν το transcoding, δηλαδή η κωδικοποίηση του κωδικοποιημένου SVC bitstream σε κάποια άλλη μορφή, μέσω άλλης εφαρμογής, συμβάλλοντας στη διατήρηση της χαμηλής καθυστέρησης κατά την διαδικασία μεταβολής ανάμεσα στα bit streams που μεταδίδονται. Ακολουθώντας, αναλύεται το δεύτερο μέρος της προτεινόμενης λύσης, που είναι το σύστημα ADIVIS. Θεωρείται αναγκαίο για την επίτευξη του στόχου μας, καθώς η αστάθεια που συναντάται στα ασύρματα δίκτυα από πλευράς εύρους ζώνης, καθυστέρησης, απωλειών και jitter έχει τεράστιο αρνητικό αντίκτυπο στην μετάδοση των video streams. Λαμβάνοντας υπόψη και την αλματώδη αύξηση των ασύρματων συσκευών που χρησιμοποιούν εφαρμογές video streaming μέσω διαδικτύου, είναι φυσικό επακόλουθο η μεγάλη ανάγκη για εξεύρεση αποδοτικών και αποτελεσματικών τεχνικών για adaptive μετάδοση συμπιεσμένων βιντεοροών, ώστε να ικανοποιούνται σε μεγαλύτερο βαθμό οι περιορισμοί, οι ικανότητες και οι απαιτήσεις των κινητών και ασύρματων δικτύων, ως επίσης και των εφαρμογών, υπηρεσιών και γενικότερα των χρηστών.

Επαναλαμβάνεται, ότι σύμφωνα με τον σχεδιασμό του ADIVIS, τα πακέτα δεδομένων αποστέλλονται στον παραλήπτη μέσω RTP, συνεπώς είναι λογική και η ύπαρξη RTCP επικοινωνίας μεταξύ των δύο οντοτήτων.

Ο feedback μηχανισμός του ADIVIS προϋποθέτει τη λήψη ανατροφοδότησης τόσο από τον ίδιο τον παραλήπτη, όσο και από τον πυρήνα του δικτύου. Όσο αφορά την πληροφόρηση από τον παραλήπτη, αυτή προέρχεται μέσω των RTCP Receiver Reports. Η μετρική που χρησιμοποιείται από τα πακέτα αυτά είναι ο αριθμός των χαμένων πακέτων μεταξύ της στιγμής αυτής και της λήψης του προηγούμενου RR. Ο χρόνος αυτός ονομάζεται περίοδος απόφασης. Ο παραλήπτης επεξεργάζεται την πληροφορία αυτή, όπως περιγράφεται στο Κεφάλαιο 7, ώστε να εξαχθεί η πρώτη παράμετρος που δίνεται σαν είσοδος στον αλγόριθμο συγκεχυμένης απόφασης. Περνώντας τώρα στο δεύτερο κομμάτι πληροφόρησης που λαμβάνεται από τον πυρήνα του δικτύου, αυτή προέρχεται μέσα από την Ρητή Ειδοποίηση Συμφόρησης και απαιτεί την ύπαρξη της κατάλληλης υποδομής στο δίκτυο, ώστε να είναι δυνατή. Συγκεκριμένα, στοιχεία του δικτύου, όπως για παράδειγμα οι δρομολογητές, αποστέλλουν πακέτα προς τον αποστολέα με το ECN bit ενεργοποιημένο. Ο αριθμός των πακέτων αυτών, εντός μιας περιόδου απόφασης, τυγχάνει επεξεργασίας, ώστε να εξαχθεί η δεύτερη τιμή εισόδου του αλγορίθμου συγκεχυμένης απόφασης.

Ο αλγόριθμος αυτός, δεδομένων των τιμών των παραμέτρων που δίνονται σαν είσοδος, δίνει σαν έξοδο μία τιμή, που ουσιαστικά καθορίζει την αύξηση, μείωση ή διατήρηση του τρέχοντος bandwidth στο δίκτυο για μετάδοση του βίντεο. Ακολούθως, το εύρος ζώνης αυτό χρησιμοποιείται για τον καθορισμό του αριθμού ατρωμάτων από τα οποία θα αποτελείται το νέο bitstream που θα μεταδίδεται. Ο αλγόριθμος αυτό είναι επαναληπτικός, μέχρι την ολοκλήρωση του streaming ολόκληρου του βίντεο.

Για την προγραμματιστική υλοποίηση του αλγορίθμου ADIVIS έχει χρησιμοποιηθεί η γλώσσα προγραμματισμού C# σε περιβάλλον Microsoft Visual Studio 2008. Τα εργαλεία αυτά χρησιμοποιήθηκαν και για την υλοποίηση της διαπροσωπείας μεταξύ ADIVIS και JSVM, ενώ παράλληλα αναπτύχθηκε γραφικό περιβάλλον για τις κυριότερες από τις λειτουργίες του λογισμικού JSVM

9.2 Προτεινόμενος Αλγόριθμος

Με βάση την υπάρχουσα υποδομή, και σε αντίθεση με τον σχεδιασμό του ADIVIS, δεν είναι δυνατή η λήψη ανατροφοδότησης από τον πυρήνα του δικτύου, ως εκ τούτου δεν είναι εφικτό να έχουμε πραγματικές τιμές για την μεταβλητή DNECN που χρησιμοποιείται. Για να γίνει αυτό είναι αναγκαία η υλοποίηση του συστήματος και η χρήση του σε πραγματικές συνθήκες πάνω από ένα δίκτυο με δυνατότητα αποστολής ECN ενημερώσεων προς τον αποστολέα μέσω των ενδιάμεσων δρομολογητών του δικτύου, ώστε να αποτραπούν περιττές απώλειες πακέτων. Για αυτό τον λόγο κατά την υλοποίηση θα προσομοιωθεί η αποστολή των πακέτων αυτών μέσω της λήψης τυχαίων τιμών για την παράμετρο, οι οποίες πάντα θα ανταποκρίνονται σε τιμές οι οποίες είναι έγκυρες για την γλωσσική μεταβλητή DNECNsc(kT), δηλαδή από -1 μέχρι +1, σύμφωνα με τις προδιαγραφές του ADIVIS. Σημειώνεται ότι οι τιμές αυτές δεν ανατίθενται απευθείας στη παράμετρο αυτή, αλλά στη μεταβλητή NECN(kT), με τις πράξεις που απαιτούνται για υπολογισμό της γλωσσικής μεταβλητής να εκτελούνται σύμφωνα με τον σχεδιασμό του ADIVIS. Με αυτό τον τρόπο καθιστάται ευκολότερη η μετάβαση του προγράμματος σε πραγματικό σύστημα, καθώς η μόνη αλλαγή που θα απαιτείται είναι η αλλαγή της πηγής αποστολής των ενημερώσεων ECN.

Επιπλέον, η δυνατότητα λήψης πακέτων RTCP, προϋποθέτει την ύπαρξη μίας συνόδου RTP για αποστολή του βίντεο μέσω αποστολέα και παραλήπτη. Σε αυτή την περίπτωση όμως υλοποιείται μόνο το πρόγραμμα που χρησιμοποιείται από την πλευρά του αποστολέα του βίντεο και συνεπώς δεν είναι εφικτή η δημιουργία τέτοιας συνόδου επικοινωνίας. Προφανώς, επίσης, δεν είναι δυνατό να λαμβάνονται πραγματικά RTCP πακέτα, αφού δεν δημιουργείται σύνδεση μεταξύ των δύο οντοτήτων της αρχιτεκτονικής πελάτη-εξυπηρετητή, για να είναι δυνατή η αποστολή πάνω από RTP του βίντεο που επιλέγεται. Ως εκ τούτου, είναι αναγκαίο και πάλι με κάποια τεχνική να προσομοιώνεται η μεταβλητή LRSP(κΤ), και ακολούθως μέσω της εκτέλεσης των απαραίτητων πράξεων, όπως αυτές καθορίζονται στις προδιαγραφές του ADIVIS, να εξάγεται η γλωσσική παράμετρος DLRPS(κΤ). Και πάλι, ο τρόπος με τον οποίο λαμβάνεται η τιμή αυτή είναι η εκτέλεση των πράξεων πάνω στην μεταβλητή LRSP(κΤ), στην οποία και ανατίθενται τυχαίες τιμές. Όπως και για την προηγούμενη παράμετρο, οι τιμές δεν ανατίθενται απευθείας στην γλωσσική παράμετρο, με στόχο να παραμείνει αυτή η διαφορά σε σχέση με τον αρχικό σχεδιασμό, και όταν αναπτυχθεί το πλήρες σύστημα να είναι αναγκαία μόνο η ανάθεση της τιμής που θα λαμβάνεται μέσω των RTCP πακέτων στη μεταβλητή LRSP(κΤ), σε κάθε περίοδο.

Η περίοδος απόφασης είναι το χρονικό διάστημα μεταξύ της λήψης δύο διαδοχικών Receiver Reports στα πλαίσια μίας RTCP συνόδου. Μιας και όπως αναφέρθηκε πιο πάνω κάτι τέτοιο δεν υπάρχει στο παρόν στάδιο, θα πρέπει να ανατεθεί μία τιμή στην μεταβλητή αυτή ώστε ανά τακτά χρονικά διαστήματα, να ελέγχεται το διαθέσιμο εύρος ζώνης και ανάλογα να μεταβάλλεται το bitrate και συνεπώς ο αριθμός των στρωμάτων του βίντεο που αποστέλλεται. Έπειτα από έρευνα σε υπάρχοντα άρθρα σχετικά με τον αλγόριθμο του ADIVIS, αποφασίστηκε όπως η τιμή της μεταβλητής αυτής να είναι 0.5 δευτερόλεπτα. Αυτό κρίθηκε αναγκαίο καθώς με αυτό τον τρόπο το σύστημα ανταποκρίνεται γρήγορα στις μεταβολές που παρουσιάζονται στο δίκτυο. Λαμβάνοντας υπόψη συχνά τις αλλαγές στις γλωσσικές μεταβλητές, αντιλαμβανόμαστε νωρίς τις μεταβολές στις δικτυακές συνθήκες και επίσης αποτρέπονται πιθανές απώλειες πακέτων σε περίπτωση ύπαρξης συμφόρησης. Από την άλλη δεν προτιμήθηκε κάποια μικρότερη τιμή γιατί πιθανό να οδηγούσε σε αχρείαστες εναλλασσόμενες μεταβάσεις από ένα αριθμό στρωμάτων σε ένα άλλο, λόγω μιας στιγμιαίας μεταβολής στις γλωσσικές μεταβλητές, Επιπλέον, μειώνοντας

την τιμή περισσότερο σημαίνει ότι τα Receiver Reports αποστέλλονται πολύ συχνά με αποτέλεσμα να καταναλώνουν μεγάλο μέρος του διαθέσιμου bandwidth, κάτι για το οποίο εξάλλου λαμβάνεται πρόνοια και κατά τον σχεδιασμό του πρωτοκόλλου RTCP.

Σαν αρχικό εύρος ζώνης το οποίο θα χρησιμοποιηθεί στο πρόγραμμα θα είναι το 85% του bandwidth το οποίο ο χρήστης ορίζει στο πρόγραμμα. Αυτό αντικατοπτρίζει σε ένα βαθμό τα πραγματικά δικτυακά δεδομένα, αφού λόγω απωλειών, συμφόρησης και άλλων παραγόντων ποτέ δεν είναι εφικτή η αξιοποίηση για κατέβασμα αρχείων το 100% του διαθέσιμου link capacity. Εξάλλου, θεωρείται ότι αυτός ο ρυθμός μετάδοσης δεδομένων δεν είναι ποτέ δυνατό να ξεπεραστεί ακόμη και όταν το αποτέλεσμα του αλγορίθμου συγκεχυμένης λογικής του ADIVIS, θεωρεί ότι υπάρχει επιπλέον διαθέσιμο bandwidth. Σε τέτοια περίπτωση, το bandwidth διατηρείται σταθερό στην τιμή αυτή. Μια επιπλέον υπόθεση στο πρόγραμμα είναι ότι δεν καταναλώνεται καθόλου από το διαθέσιμο εύρος ζώνης από οποιαδήποτε άλλη εφαρμογή και είναι όλο αφιερωμένο στο video streaming.

Επαναληπτικά θα αποστέλλονται μέρη του βίντεο που επιλέχθηκε σύμφωνα με το διαθέσιμο bandwidth. Η καθυστέρηση η οποία θα προκύπτει από την αποστολή κάθε πακέτου είναι ίση με τον μέγεθος του πακέτου διαιρεμένο με το στιγμιαίο διαθέσιμο bandwidth. Ανά χρονικά διαστήματα ίσα με T , θα εκτελείται ο αλγόριθμος συγκεχυμένης απόφασης του ADIVIS, ο οποίος θα καθορίζει και το νέο διαθέσιμο bandwidth. Ακολούθως θα γίνεται μία νέα εξαγωγή βιντεοροής από το αρχείο εισόδου, η οποία και θα αποστέλλεται. Η εύρεση του σημείου από το οποίο θα συνεχίσει η αποστολή πακέτων του νέου αρχείου, ώστε να επιτευχθεί συγχρονισμός με το σημείο όπου σταμάτησε το προηγούμενο αρχείο δεν είναι εφικτή με τις υπάρχουσες τεχνικές, αφού δεν υπάρχει τρόπος τυχαίας προσπέλασης μέσα στο bit stream. Συνεπώς, κάποιος άλλος τρόπος συνέχισης του stream πρέπει να χρησιμοποιηθεί. Κάθε φορά που ξεκινά η αποστολή νέας ροής ξεκινά να αυξάνεται ένας μετρητής των δεδομένων που έχουν αποσταλεί μέχρι τη στιγμή που πρέπει να γίνει μια νέα μετάβαση σε άλλη βιντεοροή. Πριν την έναρξη αποστολής της ακόλουθης ροής, υπολογίζεται ο λόγος των δεδομένων που έχουν αποσταλεί από το προηγούμενο βίντεο σε σχέση με το συνολικό μέγεθός του. Ο λόγος αυτός πολλαπλασιάζεται με το συνολικό μέγεθος του νέου βίντεο και η αποστολή των δεδομένων ξεκινά από αυτό το σημείο. Φυσικά, η τεχνική αυτή δεν μπορεί να θεωρηθεί απόλυτα σωστή, καθώς δεν είναι εξασφαλισμένη η αναλογία όγκου δεδομένων μιας βιντεοροής με τον χρόνο στον οποίο

αυτά αντιστοιχούν, εντούτοις για πειραματικούς σκοπούς μπορεί να χρησιμοποιηθεί για την εκτέλεση του streaming.

Όσο αφορά τις μεταβάσεις αυτές εκτελούνται σύμφωνα με το ADIVIS. Δηλαδή σε περίπτωση αντίληψης ότι το διαθέσιμο bandwidth έχει μειωθεί τότε έχουμε άμεση μετάβαση προς τα κάτω, δηλαδή λαμβάνονται δραστικά μέτρα. Αντίθετα σε περίπτωση αύξησης του διαθέσιμου εύρους ζώνης, το σύστημα μπαίνει σε φάση χρονικής υστέρησης και εάν το bandwidth διατηρηθεί στα ίδια επίπεδα μέχρι τον επόμενο έλεγχο (επόμενη περίοδος απόφασης) τότε εκτελείται μετάβαση σε βίντεο περισσότερων στρωμάτων. Κάθε φορά η παράμετρος που δίνεται στο JSVM bit stream extractor είναι το διαθέσιμο bandwidth, έτσι ώστε το πρόγραμμα αυτό να δώσει σαν έξοδο το βίντεο του οποίου το bit rate είναι το αμέσως μικρότερο αν όχι ίσο με αυτό που εμείς μπορούμε να αποστείλουμε τη δεδομένη χρονική στιγμή.

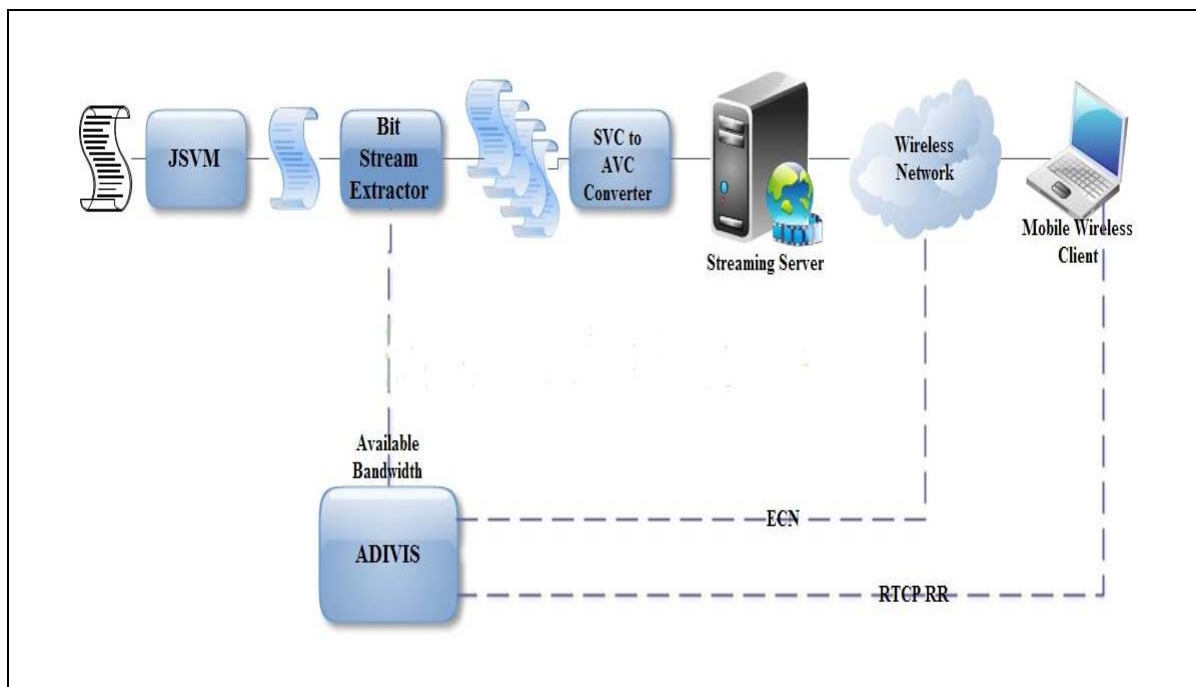
Στον πίνακα 8.1 της εργασίας αυτής παρουσιάζεται σχηματικά ο τρόπος χειρισμού των τιμών του αλγορίθμου συγκεχυμένης λογικής του ADIVIS, ώστε να εξαχθεί η παράμετρος εξόδου $a(kT)$. Στον πίνακα αυτό δεν εμφανίζονται οι ακριβείς τιμές των παραμέτρων αλλά λεκτικές κωδικοποιήσεις, τόσο για την είσοδο, όσο και για την έξοδο. Στον πίνακα που ακολουθεί παρουσιάζονται οι τιμές που καθορίστηκε να λαμβάνει η τιμή εξόδου, ανάλογα με τις τιμές των δύο παραμέτρων εισόδου, στο πρόγραμμα.

$a(kT)$		$DN_{ECNsc}(kT)$						
		[-1, -0.85]	[-0.84, -0.35]	[-0.1, -0.34]	(-0.1, 0.1)	[0.1, 0.34]	[0.35, 0.84]	[0.85, 1]
$D_{LRPS}(kT)$	[-1, -0.85]	1.5	1.5	1.15	1.15	0	0.75	0.5
	[-0.84, -0.35]	1.5	1.35	0	0	0	0.75	0.5
	[-0.1, -0.34]	1.15	0	1.15	0	0	0.75	0.5
	(-0.1, 0.1)	1.15	0	0	1.15	0	0.75	0.5
	[0.1, 0.34]	0	0	0	0	0.75	0.75	0.5
	[0.35, 0.84]	0	0	0	0	0.75	0.75	0.5
	[0.85, 1]	0.75	0.75	0.75	0.75	0.5	0.5	0.5

Πίνακας 9.1: Μετάφραση λεκτικών μεταβλητών Πίνακα 7.1 σε αριθμητικές τιμές

Έπειτα από την εξαγωγή της κατάλληλης υποροής σε κάθε περίοδο απόφασης το εξαγόμενο SVC bit stream μετατρέπεται σε AVC bit stream μέσω του εργαλείου AVCRewrite του JSVM. Αυτό αφαιρεί διάφορα χαρακτηριστικά του scalable bit stream αλλά προσφέρει την δυνατότητα μεταφοράς του βίντεο από περισσότρους εξυπηρετητές, αφού δεν υποστηρίζουν όλοι scalable video και επιπλέον δεν είναι πλέον αναγκαία η ύπαρξη scalable decoder από πλευράς του παραλήπτη, όπου η ροή μπορεί να φτάσει απευθείας στον media player για αναπαραγωγή.

9.3 Σχηματική Αναπαράσταση Προτεινόμενης Λύσης



Εικόνα 9.1: Σχηματική Αναπαράσταση Προτεινόμενης Λύσης

Η πιο πάνω εικόνα παρουσιάζει διαγραμματικά την προτεινόμενη λύση. Φυσικά, η αποστολή της AVC ροής προς τον πελάτη δεν υλοποιείται και επιπλέον, η ανατροφοδότηση από τις δύο πηγές, πελάτη και δίκτυο, δεν υλοποιούνται με τον πιο πάνω τρόπο για λόγους που αναφέρθηκαν προηγουμένως.

Όπως φαίνεται, ένα αρχείο βίντεο δίνεται σαν είσοδος στον κωδικοποιητή του JSVM, παράγοντας στην έξοδο ένα στρωματοποιημένο αρχείο βίντεο. Δεδομένου ενός αρχικού εύρους ζώνης εξάγεται η υποροή με τα κατάλληλα στρώματα και ακολούθως μετατρέπεται σε AVC ροή. Αφού ξεκινήσει η αποστολή της, περιοδικά λαμβάνονται πληροφορίες

ανατροφοδότησης από το δίκτυο και τον πελάτη, οι οποίες φτάνουν σαν είσοδος στο σύστημα ADIVIS. Με μηχανισμούς που περιγράφηκαν στην προηγούμενη ενότητα, αποφασίζεται το νέο διαθέσιμο bandwidth, που ουσιαστικά καθορίζει και το bitrate της νέας υποροής που θα εξαχθεί. Η διαδικασία αυτή είναι επαναληπτική και τερματίζεται όταν ολοκληρωθεί το streaming ολόκληρου του βίντεο.

9.4 Ακολουθίες Video που Χρησιμοποιήθηκαν

Για τον έλεγχο του συστήματος ήταν αναγκαία η κωδικοποίηση σε στρωματοποιημένο video κάποια ακολουθίας. Το video που επιλέχθηκε να χρησιμοποιηθεί ήταν η ακολουθία Foreman, αφού πρόκειται για μία εναλλασσόμενη ακολουθία όσο αφορά την κίνηση και την αλλαγή του φόντου.

Για να είναι δυνατή η παραγωγή στρωματοποιημένου βίντεο, ήταν υποχρεωτική η χρήση τεσσάρων διαφορετικών μορφών της ακολουθίας αυτής, οι οποίες διέφεραν στο resolution και στο framerate.

Ακολουθία	Χωρική Ανάλυση	Πλαίσια ανά Δευτερόλεπτο	Μέγεθος Αρχείου (KB)
176x144_7.5_FOREMAN.yuv	176x144	7.5	2785
176x144_15_FOREMAN.yuv	176x144	15	5569
352x288_15_FOREMAN.yuv	352x288	15	22275
352x288_30_FOREMAN.yuv	352x288	30	44550

Πίνακας 9.2: Ακολουθίες που χρησιμοποιήθηκαν και χαρακτηριστικά τους

Παρατηρούμε ότι κανένα ζεύγος χαρακτηριστικών χωρικής ανάλυσης και framerate δεν είναι το ίδιο. Επιπλέον, το πρώτο video χρησιμοποιείται για την εξαγωγή του base layer, δηλαδή του στρώματος 0 με το χαμηλότερο bitrate. Επίσης, στο αρχείο ρυθμίσεων για το εξαγόμενο SVC bitstream καθορίστηκε ότι ο μέγιστος αριθμός από πλαίσια ανά δευτερόλεπτο σε οποιοδήποτε στρώμα, θα ήταν 30. Οι χωρικές αναλύσεις οι οποίες θα έχουν οι διαθέσιμες υποροές της εξόδου του κωδικοποιητή θα είναι αυτές των πιο πάνω ακολουθιών.

Το μέγεθος αρχείου εμφανίζεται για να δείξει την αναλογία του όγκου δεδομένων που απαιτούνται για την αποθήκευση ενός βίντεο, σε σχέση με την πολυπλοκότητα των χαρακτηριστικών του. Εξάλλου, αυτός είναι και ένα από τους λόγους που επιλέχθηκε ο τρόπος καθορισμού του σημείου έναρξης streaming κάθε νέου video να είναι αυτός που περιγράφεται στην πρώτη ενότητα του κεφαλαίου αυτού.

Το αποτέλεσμα κωδικοποίησης των πιο πάνω βίντεο, δηλαδή τα στρώματα τα οποία είναι δυνατό να εξαχθούν, παρουσιάζονται στην επόμενη εικόνα, μετά από εκτέλεση του BitStreamExtractor..

Contained Layers:
=====

Layer	Resolution	Framerate	Bitrate	MinBitrate	DTQ
0	176x144	1.8750	27.20	27.20	(0,0,0)
1	176x144	3.7500	42.10	42.10	(0,1,0)
2	176x144	7.5000	59.00	59.00	(0,2,0)
3	176x144	1.8750	32.50	32.50	(1,0,0)
4	176x144	3.7500	48.40	48.40	(1,1,0)
5	176x144	7.5000	66.60	66.60	(1,2,0)
6	176x144	15.0000	83.60	83.60	(1,3,0)
7	352x288	1.8750	109.70	109.70	(2,0,0)
8	352x288	3.7500	158.10	158.10	(2,1,0)
9	352x288	7.5000	211.80	211.80	(2,2,0)
10	352x288	15.0000	270.70	270.70	(2,3,0)
11	352x288	1.8750	158.40	158.40	(3,0,0)
12	352x288	3.7500	217.30	217.30	(3,1,0)
13	352x288	7.5000	283.00	283.00	(3,2,0)
14	352x288	15.0000	358.50	358.50	(3,3,0)
15	352x288	30.0000	419.80	419.80	(3,4,0)

Εικόνα 9-2:Στρώματα που προκύπτουν μετά από την Κωδικοποίηση

Όπως προτρέπει και η σχεδίαση του ADIVIS, το στρώμα βάσης είναι κωδικοποιημένο σε αρκετά χαμηλό bitrate και resolution, ώστε να μπορεί να εξυπηρετεί πολλές συσκευές και να ικανοποιεί χρήστες με χαμηλή σύνδεση στο Διαδίκτυο. Ακολουθώντας, σε κάθε στρώμα, ανάλογα με τον συνδυασμό των χαρακτηριστικών ποιότητας, χωρικής ανάλυσης και πλαισίων ανά δευτερόλεπτο, αυξάνεται και το bitrate τους.

9.5 Εξαγωγή Αποτελεσμάτων

Σύμφωνα με την υπάρχουσα τοπολογία και τρόπο λειτουργίας του προγράμματος δεν είναι δυνατή η λήψη αποτελεσμάτων που να είναι ουσιώδη και να μπορούν να αναλυθούν με σαφήνεια, καθώς με τη χρήση τυχαίων αριθμών για τις δύο γλωσσικές μεταβλητές που δεν είναι δυνατό να λαμβάνουμε όπως θα γινόταν κάτω από πραγματικές δικτυακές συνθήκες, δεν έχουμε τη δυνατότητα να γνωρίζουμε την ρεαλιστική συμπεριφορά του προγράμματος βάσει της μεταβολής τους.

Μία εναλλακτική λύση θα ήταν η χρήση κάποιου αρχείου, στο οποίο να μπορούσαμε χειρωνακτικά να εισάγουμε συγκεκριμένες τιμές, ώστε να γνωρίζουμε εκ των προτέρων ποια θα ήταν η φυσιολογική συμπεριφορά του προγράμματος, υπό αυτές τις συνθήκες. Εντούτοις, μία τέτοια λύση δεν θα ήταν πάλι σε θέση να μας δώσει ακριβή αποτέλεσμα που να ανταποκρίνονται απόλυτα στην πραγματικότητα, καθώς τα ασύρματα δίκτυα είναι εκ φύσεως απροσδιόριστα και παρουσιάζουν πολύ απότομες μεταβολές σε διάφορα στοιχεία, όπως το διαθέσιμο εύρος ζώνης, αφού εξαρτώνται από πολλούς παράγοντες και είναι εκτεθειμένα στις διάφορες παρεμβολές από άλλα δίκτυα ή συσκευές.

Πέραν των πιο πάνω, όμως, έπειτα από πολλές εκτελέσεις του προγράμματος και αξιολόγηση του log file, που δημιουργείται με το πέρας κάθε εκτέλεσης, παρατηρούμε ότι το πρόγραμμα εμφανίζει την αναμενόμενη συμπεριφορά και εκτελεί τις ορθές μεταβάσεις μεταξύ ακολουθιών που περιέχουν διαφορετικά στρώματα ανάλογα με το bandwidth. Συγκεκριμένα, στο αρχείο αυτό εμφανίζεται ο χρόνος λήψης κάθε μέτρησης, που αντιστοιχεί σε κάθε περίοδο απόφασης, οι τιμές για τις δύο γλωσσικές μεταβλητές, κατά πόσο έχουμε μπει σε φάση χρονικής υστέρησης, η τιμή εξόδου του αλγορίθμου συγκεκριμένης λογικής, το τρέχον διαθέσιμο εύρος ζώνης και αν πρέπει να γίνει τώρα νέα μετάβαση. Στην τελευταία περίπτωση τυπώνεται επίσης και η εντολή που εκτελείται για εξαγωγή νέας υποροής από το αρχικό bit stream.

Μέσα από αυτά τα αποτελέσματα, παρατηρούμε ότι όσο μεγαλύτερο είναι το αρχικό διαθέσιμο bandwidth, τόσο μεγαλύτερα είναι και τα bitrates τα οποία ζητούμε από τον bit stream extractor να έχουν τα εξαγόμενα βίντεο. Επίσης, έχοντας ένα ψηλό αρχικό bandwidth, επακόλουθες μειώσεις του, εξακολουθούν να μειώνουν το bitrate σταδιακά, διατηρώντας το εντούτοις σε ψηλότερα επίπεδα παρά με ένα χαμηλότερο αρχικό.

Μεταβάσεις προς ανώτερα στρώματα σπάνια εμφανίζονται, καθώς όπως αναφέρθηκε, οι γλωσσικές μεταβλητές λαμβάνονται με τυχαίο τρόπο. Ως εκ τούτου, και βάση των πιθανοτήτων που προσφέρονται από τον πίνακα που μεταφράζει τις τιμές αυτές στην τιμή εξόδου, είναι αρκετά δύσκολη η διατήρηση της τιμής $a(kT)$ σε τιμές μεγαλύτερες του 1 για δύο συνεχόμενες περιόδους απόφασης.

Άλλη παρατήρηση που γίνεται, είναι ότι η καθυστέρηση στην ολοκλήρωση του streaming εξαρτάται από δύο παράγοντες. Ο πρώτος είναι το εκάστοτε στιγμιαίο bandwidth, ενώ ο δεύτερος, ο αριθμός των μεταβάσεων που πρέπει να εκτελεστούν. Συγκεκριμένα, όσο περισσότερες είναι οι μεταβάσεις, τόσο περισσότερος είναι ο απαιτούμενος χρόνος για να ολοκληρωθεί η εκτέλεση. Αυτό είναι συνέπεια δύο λόγων, αρχικά με τις μεταβάσεις που συμβαίνουν, αν είναι προς τα κάτω, μειώνεται και το διαθέσιμο bandwidth και προφανώς αυξάνεται η καθυστέρηση. Ο δεύτερος λόγος είναι το overhead που προκαλούν οι διάφοροι υπολογισμοί και λειτουργίες, όπως εξαγωγή νέας βιντεοροής, σε περίπτωση που κριθεί αναγκαία η εκτέλεση νέας μετάβασης.

Σχετικά με τον τρόπο ελέγχου των γλωσσικών μεταβλητών του αλγόριθμου απόφασης, κρίθηκε σοφότερο και οικονομικότερο από άποψη καθυστέρησης στο σύστημα λόγω περιττών ελέγχων και υπολογισμών (overhead), να υλοποιηθεί με τρόπο διαφορετικό παρά τα πολλαπλά if statements που συνδυάζουν στις συνθήκες και τις δύο μεταβλητές. Συγκεκριμένα, δημιουργείται εξ αρχής ο πίνακας που περιέχει τις τιμές που μπορεί να πάρει η μεταβλητή εξόδου $a(kT)$. Ακολούθως, όποτε πρέπει να ληφθεί απόφαση, κάθε παράμετρος από τις δύο, ελέγχεται ξεχωριστά, με την μία να καθορίζει την γραμμή και την δεύτερη την στήλη μέσα στον Πίνακα 9.1. Με αυτό τον τρόπο βρίσκουμε τη θέση του πίνακα που αντιστοιχεί στην τρέχουσα τιμή εξόδου πολύ πιο γρήγορα παρά με διαδοχικούς και πολλαπλούς ελέγχους που συνδυάζουν σύγκριση τιμών και για τις δύο παραμέτρους.

Με βάση τα όσα έχουν γραφεί πιο πάνω σχετικά με την μεγάλη και συνήθως μεταβαλλόμενη καθυστέρηση είναι λογικό να αναμένουμε ότι από την πλευρά του παραλήπτη, αυτά θα μεταφραστούν σε καθυστέρηση και jitter, δύο παράγοντες που θα μειώσουν την ποιότητα της υπηρεσίας που του παρέχεται. Συνεπώς, αντιλαμβανόμαστε ότι τόσο η συμφόρηση, όσο και οι απώλειες που προκύπτουν στο δίκτυο και για τις οποίες μας ενημερώνουν τα Receiver Reports, αποτελούν σημαντικούς ρυθμιστικούς παράγοντες της ποιότητας της υπηρεσίας που απολαμβάνει ο τελικός χρήσης. Ωστόσο, με αύξηση του

bandwidth, η ποιότητα αυτή δεν επηρεάζεται αρνητικά, καθώς ναι μεν τα πακέτα θα μεταδίδονται με μικρότερη καθυστέρηση και εισάγεται jitter, ωστόσο με τη χρήση ενός ικανοποιητικά μεγάλου playout buffer από πλευράς παραλήπτη, καθίσταται δυνατή η προσωρινή αποθήκευσή τους, μέχρι να φτάσει η στιγμή αναπαραγωγής τους.

9.6 Αποτελέσματα Προγράμματος

Έχοντας αναφέρει πιο πάνω αρκετά χαρακτηριστικά του υλοποιημένου προγράμματος, είναι στιγμή να παρουσιαστούν και τα πραγματικά αποτελέσματα, μέσω του log file που παράγεται έπειτα από την ολοκλήρωση της εκτέλεσης. Ακολουθώντας στην εικόνα 9-3 παρουσιάζονται τυπικά τα αποτελέσματα από μία εκτέλεση του προγράμματος, κατά την οποία παρουσιάζονται μεταβάσεις προς τα πάνω αλλά και προς τα κάτω, ενώ υπάρχουν επίσης και οι περιπτώσεις που το διαθέσιμο εύρος ζώνης παραμένει αναλλοίωτο.

```
Time(in ms)= 0 DLRPS= 0 DNECN= 0 Packets Sent= 0
Current BW= 56000 Previous Hysteresis= False Current Hysteresis= False New
Extraction= False
foreman_final.264 Streamed.264 47.6 AVC_Streamed.avc

Time(in ms)= 500 DLRPS= 0.833577382766445 DNECN= 0.162740792689259 Packets
Sent= 4
Current BW= 42000 Defuzzied Output= 0.75 Previous Hysteresis= False Current
Hysteresis= False New Extraction= True

foreman_final.264 Streamed.264 35.7 AVC_Streamed.avc
Time(in ms)= 1000 DLRPS= -0.350013439706533 DNECN= 0.694568292095591 Packets
Sent= 1
Current BW= 31500 Defuzzied Output= 0.75 Previous Hysteresis= False Current
Hysteresis= False New Extraction= True

foreman_final.264 Streamed.264 26.775 AVC_Streamed.avc
Time(in ms)= 1500 DLRPS= -0.184932887640285 DNECN= -0.560215377975356
Packets Sent= 2
Current BW= 31500 Defuzzied Output= 0 Previous Hysteresis= False Current Hysteresis=
False New Extraction= False

Time(in ms)= 2000 DLRPS= 0.121143298745688 DNECN= 0.674745890160439 Packets
Sent= 1
Current BW= 23625 Defuzzied Output= 0.75 Previous Hysteresis= False Current
Hysteresis= False New Extraction= True
```

```

foreman_final.264 Streamed.264 20.08125 AVC_Streamed.avc
Time(in ms)= 2500 DLRPS= -0.136744752124299 DNECN= -0.517689496985492
Packets Sent= 1
Current BW= 23625 Defuzzied Output= 0 Previous Hysteresis= False Current Hysteresis=
False New Extraction= False

Time(in ms)= 3000 DLRPS= 0.304577785685928 DNECN= -0.517689496985492 Packets
Sent= 0
Current BW= 23625 Defuzzied Output= 0 Previous Hysteresis= False Current Hysteresis=
False New Extraction= False

Time(in ms)= 3500 DLRPS= 0.368091758046342 DNECN= 0.516329655198534 Packets
Sent= 1
Current BW= 17718.75 Defuzzied Output= 0.75 Previous Hysteresis= False Current
Hysteresis= False New Extraction= True

foreman_final.264 Streamed.264 15.0609375 AVC_Streamed.avc
Time(in ms)= 4000 DLRPS= -0.806933961253117 DNECN= 0.0176731804468079
Packets Sent= 1
Current BW= 17718.75 Defuzzied Output= 0 Previous Hysteresis= False Current
Hysteresis= False New Extraction= False

Time(in ms)= 4500 DLRPS= 0.707354472348166 DNECN= -0.401067228708913 Packets
Sent= 0
Current BW= 17718.75 Defuzzied Output= 0 Previous Hysteresis= False Current
Hysteresis= False New Extraction= False

Time(in ms)= 5000 DLRPS= -0.167792282145374 DNECN= -0.857646021460018
Packets Sent= 1
Current BW= 20376.5625 Defuzzied Output= 1.15 Previous Hysteresis= False Current
Hysteresis= True New Extraction= False

Time(in ms)= 5500 DLRPS= -0.203890408949875 DNECN= -0.857646021460018
Packets Sent= 0
Current BW= 23433.046875 Defuzzied Output= 1.15 Previous Hysteresis= True Current
Hysteresis= True New Extraction= True

foreman_final.264 Streamed.264 23433.046875 AVC_Streamed.avc
Time(in ms)= 6000 DLRPS= -0.350780842989115 DNECN= 0.573882602888105 Packets
Sent= 1
Current BW= 17574.785156 Defuzzied Output= 0.75 Previous Hysteresis= True Current
Hysteresis= False New Extraction= True

foreman_final.264 Streamed.264 17574.785156 AVC_Streamed.avc

```

Εικόνα 9-3: Παράδειγμα αποτελέσματος εκτέλεσης ADIVIS

Όπως έχει ήδη περιγραφεί αρκετές φορές προγουμένως, οι τιμές των γλωσσικών μεταβλητών DLRPS και DNECN καθορίζουν την τιμή της μεταβλητής εξόδου. Ακολουθώντας, ελέγχεται κατά πόσο αυτή είναι μεγαλύτερη του μηδέν, μικρότερη ή ίση με τον αριθμό αυτό. Αυτό το χαρακτηριστικό της καθορίζει κατά πόσο θα πρέπει να εκτελεστεί μετάβαση ή όχι, ή ακόμα αν το σύστημα πρέπει να εισέλθει σε φάση χρονικής υστέρησης. Επιπλέον, οι τιμές των γλωσσικών παραμέτρων καθορίζουν και ποσοτικά τη μεταβολή πάνω στο τρέχον διαθέσιμο εύρος ζώνης, σύμφωνα με τον Πίνακα 9.1. Σε περίπτωση που κριθεί αναγκαία η μετάβαση, ανεξάρτητα από το αν το bandwidth έχει μειωθεί ή όχι, γίνεται TRUE η τιμή της παραμέτρου που στο log file φαίνεται σαν New Extraction. Το γεγονός ότι δεν μας απασχολεί αν πρόκειται για αύξηση ή μείωση στρωμάτων οφαίλεται στο γεγονός ότι το JSVM λαμβάνει σαν παράμετρο το bit rate που επιθυμούμε να έχει η εξαγόμενη υποροή.

Συνεχίζοντας τώρα με την ανάλυση του αποτελέσματος που παρουσιάζεται, παρατηρούμε ότι αρχικά (Time= 0), όλες οι παράμετροι είναι μηδέν, αφού είναι η στιγμή εκκίνησης του streaming. Το μόνο δεδομένο που υπάρχει είναι το εύρος ζώνης που δόθηκε σαν είσοδος από τον χρήστη και είναι 56000bps. Στον χρόνο αυτό, η τιμή της μεταβλητής που ορίζει αν πρέπει να υπάρξει νέα μετάβαση είναι False, ωστόσο αυτό είναι αποτέλεσμα της χρήσης των τιμών των υπόλοιπων παραμέτρων κατά την προσομοίωση, για αυτό και εκτελείται εξαγωγή νέας υποροής από το αρχείο εισόδου. Συγκεκριμένα, η γραμμή: foreman_final.264 Streamed.264 47.6 AVC_Streamed.avc, περιέχει τις 4 παραμέτρους εισόδου του batch file που καλεί σε ακολουθία τον BitStreamExtractor για να εξάγει τα απαραίτητα στρώματα και ακολούθως το AVCRewriteTool για μετατροπή του SVC αρχείου σε AVC. Πρώτα δίνεται το αρχείο που περιέχει τα στρώματα βίντεο που προέκυψαν από την κωδικοποίηση, ακολούθως το όνομα του αρχείου στο οποίο πρέπει να αποθηκευτεί η εξαγόμενη υποροή, τρίτη παράμετρος είναι το bit rate που επιθυμούμε να έχει και τέλος δίνουμε το όνομα του αρχείου στο οποίο θέλουμε να αποθηκευτεί τελικά το AVC stream που θα χρησιμοποιηθεί για το streaming.

Τα αποτελέσματα γράφονται στο αρχείο αυτό περιοδικά, κάθε μισό δευτερόλεπτο, όσο ορίστηκε να είναι η περίοδος απόφασης (ο χρόνος μεταξύ της λήψης δύο διαδοχικών Receiver Reports). Η τιμή της παραμέτρου Current BW στο αρχείο αυτό είναι το

αποτέλεσμα του πολλαπλασιασμού του προηγούμενου διαθέσιμου εύρους ζώνης με την τιμή Defuzzied Output, που αναφέρεται στην τιμή εξόδου του αλγορίθμου απόφασης. Σε περίπτωση που η τιμή εξόδου είναι μηδέν, βέβαια, δεν εκτελείται η πράξη αυτή, αφού κάτι τέτοιο θα μηδένιζε το διαθέσιμο εύρος ζώνης, αντίθετα διατηρείται σταθερό όπως και πριν. Κάτι που είναι εμφανές μελετώντας το αποτέλεσμα, είναι η άμεση μετάβαση σε χαμηλότερα στρώματα όταν μειώνεται το διαθέσιμο εύρος ζώνης. Συγκεκριμένα, κάτι τέτοιο συμβαίνει δύο φορές, αμέσως μόλις εντοπιστεί ότι η τιμή της μεταβλητής $a(kT)$ είναι μικρότερη του 0. Αντίθετα, σε περίπτωση που το διαθέσιμο εύρος ζώνης είναι μεγαλύτερο σε σχέση με την προηγούμενη περίοδο απόφασης, το σύστημα μπαίνει σε φάση χρονικής υστέρησης, και η μετάβαση προς τα πάνω πραγματοποιείται αφότου ελεγχθεί για δεύτερη διαδοχική περίοδο απόφασης το διαθέσιμο bandwidth και βρεθεί να κυμαίνεται στα ίδια επίπεδα. Αυτό συμβαίνει για να αποφεύγονται οι συχνες μεταβάσεις από ψηλότερο στρώμα σε χαμηλότερο σε μικρό χρονικό διάστημα, ενώ η άμεση μείωση των στρωμάτων σε αντίθετη περίπτωση προσπαθεί να προλάβει τις απώλειες πακέτων, αφού το bit rate του stream που μεταδίδεται είναι μεγαλύτερο από το bandwidth που μπορεί να αφιερωθεί στην εφαρμογή αυτή.

Λόγω της πιο πάνω συμπεριφοράς, παρατηρούμε ότι αν και το αρχικό bitrate του βίντεο που αποστέλλεται ορίζεται να είναι 47.6Kbps (και όχι 56, καθώς μειώνουμε το διαθέσιμο bandwidth στο 85% του link capacity), εντούτοις οι διαδοχικές μεταβολές του bandwidth, έχουν σαν αποτέλεσμα το τελευταίο bit stream που εξάγεται να έχει bit rate μόλις κάτω από 18kbps.

Time(in ms)= 0 DLRPS= 0 DNECN= 0 Packets Sent= 0 Current BW= 56000 Previous Hysteresis= False Current Hysteresis= False New Extraction= False foreman_final.264 Streamed.264 176 144 15 47.6 AVC_Streamed.avc
Time(in ms)= 500 DLRPS= 0.692323786063271 DNECN= 0.438603942719569 Packets Sent= 2 Current BW= 42000 Defuzzied Output= 0.75 Previous Hysteresis= False Current Hysteresis= False New Extraction= True foreman_final.264 Streamed.264 176 144 15 35.7 AVC_Streamed.avc

Time(in ms)= 1000 DLRPS= -0.0586357279953713 DNECN= -0.237968369963564
Packets Sent= 2
Current BW= 48300 Defuzzied Output= 1.15 Previous Hysteresis= False Current
Hysteresis= True New Extraction= False

Time(in ms)= 1500 DLRPS= -0.357830376530918 DNECN= 0.224112945945986 Packets
Sent= 2
Current BW= 48300 Defuzzied Output= 0 Previous Hysteresis= True Current Hysteresis=
False New Extraction= True

foreman_final.264 Streamed.264 176 144 15 41.055 AVC_Streamed.avc
Time(in ms)= 2000 DLRPS= 0.713978219178495 DNECN= -0.0653061797215166
Packets Sent= 1
Current BW= 48300 Defuzzied Output= 0 Previous Hysteresis= False Current Hysteresis=
False New Extraction= False

Time(in ms)= 2500 DLRPS= -0.779474393827596 DNECN= -0.00750004500499929
Packets Sent= 2
Current BW= 48300 Defuzzied Output= 0 Previous Hysteresis= False Current Hysteresis=
False New Extraction= False

Time(in ms)= 3000 DLRPS= -0.00555220293139674 DNECN= 0.0132290781537206
Packets Sent= 2
Current BW= 48300 Defuzzied Output= 0 Previous Hysteresis= False Current Hysteresis=
False New Extraction= False

Time(in ms)= 3500 DLRPS= 0.211317277611847 DNECN= 0.344964258067759 Packets
Sent= 1
Current BW= 36225 Defuzzied Output= 0.75 Previous Hysteresis= False Current
Hysteresis= False New Extraction= True

foreman_final.264 Streamed.264 176 144 15 30.79125 AVC_Streamed.avc

Εικόνα 9-4: Παράδειγμα Εκτέλεσης ADIVIS με Σταθερές Επιλογές Χρήστη

Στην Εικόνα 9-4, παρουσιάζεται το αποτέλεσμα εκτέλεσης του ADIVIS, στην περίπτωση που εκτελείται με την επιλογή η χωρική ανάλυση και ο λόγος πλαισίων ανά δευτερόλεπτο να παραμείνει σταθερός καθόλη τη διάρκεια του streaming και ουσιαστικά το η διαφορά ανάμεσα στις υποροές που εξάγονται να είναι αποτέλεσμα του SNR scalability. Και πάλι

είναι φανερή η διαφορά της συμπεριφοράς του αλγορίθμου στις περιπτώσεις που υπάρχει μετάβαση προς τα πάνω και στις περιπτώσεις που τα στρώματα πρέπει να μειωθούν. Σημαντικό χαρακτηριστικό της εκτέλεσης αυτής όμως είναι η σειρά των γεγονότων που οδήγησαν σε αύξηση του bit rate. Αρχικά, η τιμή εξόδου παίρνει τιμή 1.15 και συνεπώς γίνεται εισαγωγή σε φάση υστέρησης. Στον επόμενο έλεγχο, η τιμή αυτή είναι 0, δηλαδή δεν υπάρχει μεταβολή, πράγμα που συνεπάγεται διαθέσιμο εύρος ζώνης ψηλότερο από τη στιγμή που έγινε έλεγχος και το σύστημα μπήκε σε φάση χρονικής υστέρησης, ως εκ τούτου κρίνεται εφικτή η αύξηση των στρωμάτων του βίντεο που θα εξαχθεί στη συνέχεια. Επιπλέον, παρατήρηση που αφορά συγκεκριμένα το παράδειγμα αυτό είναι η μη μεταβολή του bandwidth για την πλειοψηφία των περιόδων, αφού η τιμή εξόδου του αλγορίθμου συγκεκριμενής απόφασης είναι 0. Αυτός είναι και ο κύριος λόγος που το τελικό bandwidth ναί μεν είναι χαμηλότερο από το αρχικό, εντούτοις είναι ψηλότερο από το τελικό bandwidth του προηγούμενου παραδείγματος, όπου εκτελέστηκαν περισσότερες μεταβολές του bandwidth προς τα κάτω.

Εξάλλου, το γεγονός ότι επιλέχθηκε η διατήρηση των χαρακτηριστικών του resolution και του framerate καθόλη τη διάρκεια του streaming, προκαλεί την προσθήκη τριών νέων παραμέτρων που δίνονται σαν είσοδος στο batch file που εκτελεί τα αρχεία του JSVM για εξαγωγή υποροής και μετατροπής της σε AVC υποροή. Αυτά είναι το μήκος εικόνας, πλάτος εικόνας και αριθμός πλαισίων ανά δευτερόλεπτο, όπως φαίνονται στην γραμμή foreman_final.264 Streamed.264 176 144 15 30.79125 AVC_Streamed.avc

9.7 Παρατηρήσεις

Μέχρι τη στιγμή αυτή, έστω και αν ο αλγόριθμος έχει υλοποιηθεί προγραμματιστικά, κάτω από τις πιο πάνω συνθήκες, δεν μπορούμε να εξάγουμε ακριβή και ασφαλή συμπεράσματα ως προς τον σχεδιασμό του και την αποτελεσματικότητά του. Τα αποτελέσματα που είναι δυνατό να εξεταστούν, όπως δίνονται μέσα από το log file, ναί μεν ξεφεύγουν από το μέχρι σήμερα απλά θεωρητικό επίπεδο στο οποίο κινούνταν όλες οι έρευνες γύρω από τον αλγόριθμο και τη λειτουργία του ADIVIS και αποδεικνύουν ότι είναι δυνατή η μετάβαση σε βίντεο που αποτελείται από διαφορετικό αριθμό στρωμάτων, ανάλογα με το διαθέσιμο εύρος ζώνης, ωστόσο λόγω της τυχαίας ανάθεσης τιμών στις δύο μεταβλητές του

αλγόριθμου απόφασης, δεν είμαστε σε θέση να συσχετίσουμε τα αποτελέσματα και τις μεταβάσεις που προκύπτουν με κάποιο φυσικό χαρακτηριστικό του ασύρματου δικτύου στο οποίο υλοποιείται.

Συγκεκριμένα, οι απώλειες λαμβάνουν τυχαία τιμή και ως εκ τούτου δεν υπάρχει κάποια σύνδεση με το εύρος ζώνης που αφιερώνεται στον χρήστη που εκτελεί την εφαρμογή του video streaming. Επιπλέον, η καθυστέρηση ορίζεται ρητά από εμάς χρησιμοποιώντας το διαθέσιμο bandwidth και το μέγεθος κάθε πακέτου RTP, οπότε δεν μπορούν να εξαχθούν απόλυτα συμπεράσματα όσο αφορά την ποιότητα υπηρεσίας που προσφέρεται, αφού διακύμανση καθυστέρησης θα παρατηρείται μόνο κατά τις στιγμές που υπάρχει μετάβαση από ένα bitstream σε άλλο.

ΚΕΦΑΛΑΙΟ 10

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

10.1 Συμπεράσματα

10.2 Μελλοντική Εργασία

10.1 Συμπεράσματα

Όπως αναφέρθηκε και προηγουμένως, τα συμπεράσματα που μπορούν να εξαχθούν βάσει της μελέτης αυτής είναι κυρίως βασισμένα στη θεωρία, καθώς πολλές από τις παραμέτρους που χρησιμοποιούνται στην υλοποίηση δεν αντικατοπτρίζουν πραγματικές τιμές που λαμβάνονται από το δίκτυο ή από τον παραλήπτη του βίντεο, όπως περιγράφεται στις προδιαγραφές του συστήματος ADIVIS. Εντούτοις, παρατηρούμε ότι το σύστημα που αναπτύχθηκε είναι σε θέση να αντιλαμβάνεται τις αλλαγές στις δικτυακές συνθήκες ελέγχοντας τις απώλειες πακέτων και την συμφόρηση στο δίκτυο ανά τακτά χρονικά διαστήματα και εκτελεί τις ανάλογες και αναγκαίες μετατροπές στο bit stream που μεταδίδεται. Πέραν αυτών, ο τρόπος επιλογής του σημείου από το οποίο πρέπει να ξεκινήσει να μεταδίδεται η νέα ροή αποτελεί ερωτηματικό, αφού ο τρόπος με τον οποίο υλοποιείται στο πρόγραμμα που έχω αναπτύξει δεν εγγυάται τον ακριβή συγχρονισμό των δύο ροών.

Εντούτοις, ακόμα και κάτω από αυτές τις προϋποθέσεις είναι προφανές ότι η χρήση adaptive εφαρμογών και ειδικά στον τομέα των πολυμέσων μπορεί να επιφέρει θετικά αποτελέσματα, όσο αφορά την αξιοποίηση των διαθέσιμων πόρων του δικτύου και ταυτόχρονα να μειώσει τις απώλειες πακέτων που επιφέρουν, όπως είναι λογικό, μείωση στην ποιότητα της υπηρεσίας που παρέχεται στον παραλήπτη. Αν αναλογιστούμε και την αστάθεια που παρουσιάζουν τα ασύρματα δίκτυα με το συνεχές μεταβαλλόμενο διαθέσιμο bandwidth και το αρκετά ψηλό ποσοστό αναφορικά με τις απώλειες πακέτων, τότε η σημασία του adaptive video streaming λαμβάνει ακόμη μεγαλύτερες διαστάσεις.

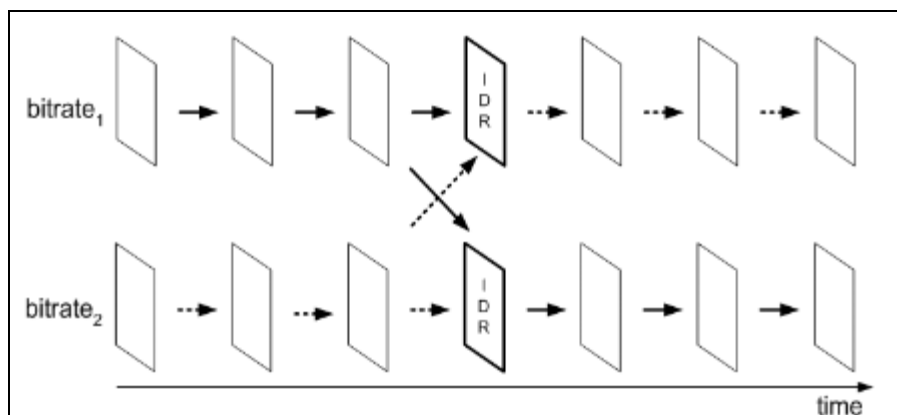
Μελετώντας το log file που παράγεται και ουσιαστικά αποτελεί περιγραφή της κατάστασης του δικτύου κατά τη διάρκεια του streaming παρατηρούμε ότι σπάνια έχουμε μεταβολές προς περισσότερα στρώματα στο video που αποστέλλεται. Αυτό ίσως είναι ένδειξη ότι θα πρέπει να αναθεωρηθεί ο αλγόριθμος συγκεχυμένης λογικής, ώστε να περιλαμβάνει περισσότερες τιμές ελέγχου, που να παρέχουν με μεγαλύτερη πιθανότητα την αύξηση των στρωμάτων, ακόμα και όταν η αύξηση του διαθέσιμου εύρους ζώνης είναι μικρότερη από τις τιμές που αναφέρονται στον αρχικό σχεδιασμό.

Επίσης, ένα άλλο συμπέρασμα που εξάγεται από το πρόγραμμα και επικυρώνει την θεωρία είναι το γεγονός ότι η απόδοση της εφαρμογής video streaming επηρεάζεται αρνητικά όταν ο ρυθμός κωδικοποίησης, το bitrate, της υποροής βίντεο που αποστέλλεται είναι πολύ κοντά ή ίση με το διαθέσιμο εύρος ζώνης που είναι διαθέσιμο. Σε τέτοια περίπτωση είναι πολύ πιθανό στην αμέσως επόμενη περίοδο απόφασης να χρειαστεί μετάβαση προς χαμηλότερο στρώμα. Στις περιπτώσεις των ασύρματων δικτύων και ειδικά σε περιπτώσεις στις οποίες η λήψη που έχει ο παραλήπτης δεν είναι αρκετά ικανοποιητική, αποδεικνύεται η σημασία κωδικοποίησης του στρώματος βάσης, του στρώματος δηλαδή με το χαμηλότερο bitrate, σε πολύ χαμηλό ρυθμό κωδικοποίησης, ώστε να μπορεί τουλάχιστον να αποσταλεί ακόμα και κάτω από τέτοιες συνθήκες.

Ερωτηματικά προκαλεί η αξιοπιστία της μεταβλητής που καταδεικνύει τον λόγο των χαμένων πακέτων ανά περίοδο απόφασης, ως ένδειξης για ύπαρξη μεγάλης συμφόρησης στο δίκτυο μέσω του οποίου αποστέλλεται η υποροή. Συγκεκριμένα, όπως είναι γνωστό, στα ασύρματα δίκτυα, είναι πολύ πιθανό η απώλεια πακέτων να οφείλεται σε σφάλματα εντός του ίδιου του πακέτου (bits μπορεί εύκολα να μεταβληθούν λόγω παρεμβολών, με αποτέλεσμα λανθασμένο checksum και drop σε ενδιάμεσο router). Συνεπώς, ακόμα και ένα κανάλι το οποίο δεν παρουσιάζει συμφόρηση, μπορεί να θεωρηθεί λανθασμένα πως δεν είναι σε θέση να ικανοποιήσει την μεταφορά του βίντεο.

Τέλος, αρκετά σημαντικό συμπέρασμα είναι η αδυναμία τυχαίας προσπέλασης του SVC bit stream, χωρίς την χρήση και γνώση της θέσης των IDR frames εντός αυτού. Πρόκειται για πλαίσια που χρησιμοποιούνται για συγχρονισμό εντός μιας ακολουθίας, αφού σε αντίθεση με τα I frames που περιέχουν αναφορές μεταξύ τους, μετά από ένα πλαίσιο IDR κανένα πλαίσιο δεν κάνει αναφορά προς τα πίσω. Αυτός είναι και ο τρόπος με τον οποίο θα πρέπει

να επιτευχθεί ο συγχρονισμός ανάμεσα στις υποροές που αποστέλλονται, όταν αντιληφθούμε την ανάγκη για μετάβαση σε διαφορετικό στρώμα.



Εικόνα 10-1:Χρήση των IDR Frames σε ένα bitstream

Όπως φαίνεται και στην πιο πάνω εικόνα, για να είναι δυνατή η μετάβαση από μία ροή που αποστέλλεται πάνω από ένα δίκτυο σε μια άλλη, είναι απαραίτητη η ανίχνευση του επόμενου IDR frame, ώστε να εμφανίζονται εικόνες που δεν χρειάζονται προηγούμενες για να αποκωδικοποιηθούν, πράγμα που θα ήταν αδύνατο, αφού μέσω της αλλαγής στρώματος αλλάζει τουλάχιστον ένα από τα χαρακτηριστικά ευκρίνειας, λόγου πλαισίων ανά δευτερόλεπτο και ποιότητας.

10.2 Μελλοντική Εργασία

Σε αυτό το σημείο θα αναφερθούν τα σημεία που θεωρείται ορθό να υλοποιηθούν στα πλαίσια μελλοντικών εργασιών ώστε το σύστημα να είναι ολοκληρωμένο και σε θέση να επιτύχει τους στόχους του σχεδιασμού του ADIVIS, καθώς επίσης και μερικές βελτιώσεις, οι οποίες έχουν σαν στόχο την αύξηση της απόδοσής του, από άποψη χρόνου που χρειάζεται για να επιτευχθεί η εναλλαγή μεταξύ video που αποστέλλονται.

Αρχικά θα πρέπει να αναπτυχθεί ένα πιο ρεαλιστικό σύστημα, το οποίο να δέχεται σύνδεση από κάποιο πελάτη και να υπάρχει πραγματική μεταφορά πακέτων στα πλαίσια των πρωτοκόλλων RTP και RTCP, ώστε να καταστήσει δυνατή την λήψη των στοιχείων από τα πακέτα αυτά, παρά να ανατίθενται τυχαίες τιμές στις μεταβλητές. Εξάλλου, είναι αναγκαία για την εκπλήρωση των προδιαγραφών του συστήματος ADIVIS, η ενσωμάτωση

λειτουργίας streaming server σε αυτό, ο οποίος θα αποστέλλει τα πακέτα RTP που ενθυλακώνουν τα δεδομένα video, στον παραλήπτη.

Εξάλλου, το εργαλείο AVCRewrite του JSVM δεν είναι επαρκώς τεκμηριωμένο και η λειτουργία του απαιτεί αρκετές ώρες πειραμάτων, μέχρι να επιτευχθεί κωδικοποίηση SVC, που να υποστηρίζει τέτοιου είδους μετατροπή. Επομένως, είναι αναγκαία είτε η αναζήτηση επιπλέον πηγών αναφορικά με αυτό το εργαλείο, ή η αναμονή για νέα εκδόση τόσο του εργαλείου αυτού με περισσότερες δυνατότητες, ή ανανεωμένου εγχειριδίου χρήσης του JSVM.

Τέλος, καλό θα ήταν να αναφερθεί και ένα στοιχείο το οποίο θα προσθέσει επιπλέον πλεονεκτήματα στο adaptive σύστημα που θα αναπτυχθεί Συγκεκριμένα, μετά από αρκετή έρευνα και πολλές ώρες σύνταξης του κώδικα του προγράμματος που αναπτύχθηκε, θεωρώ πως είναι δυνατή η μείωση του χρόνου που χρειάζεται ώστε να ελεγχθούν οι γλωσσικές μεταβλητές και να επιτευχθεί η μετάβαση από ένα στρώμα σε ένα άλλο, μέσω της χρήσης πολλών threads. Συγκεκριμένα, μπορεί να χρησιμοποιηθεί ένα thread για τον έλεγχο κάθε μίας από τις γλωσσικές μεταβλητές. Επιπλέον, threads μπορούν να αξιοποιηθούν και για άλλες λειτουργίες, όπως ανάγνωση παραμέτρων από κάποιο αρχείο και γραφή σε κάποιο άλλο. Βέβαια, κάτι τέτοιο κάνει την υλοποίηση ακόμη πιο πολύπλοκη απ'ότι ήδη είναι αφού τίθενται πλέον και θέματα συγχρονισμού.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] P. Antoniou, A. Pitsillides, V. Vassiliou, “Adaptive Feedback Algorithm for Internet Video Streaming based on Fuzzy Rate Control”, in *Proceedings of the IEEE 18th International Symposium on Volume*, September 2007
- [2] P. Antoniou, V. Vassiliou, A. Pitsillides, “Delivering Adaptive Scalable Video over the Wireless Internet”, in 1st ERCIM Workshop on eMobility, Coimbra, Portugal, May, 2007, pp. 23-34
- [3] P. Antoniou, V. Vassiliou, A. Pitsillides, “ADIVIS: A Novel Adaptive Algorithm for Video Streaming over the Internet”, in *Proceedings to The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2007
- [4] A. C. Bovik, “Handbook of image and video processing”, Second Edition, Elsevier Science & Technology Books, 2005, pp.643-877
- [5] S. F. Chang, A. Vetro, “Video Adaptation: Concepts, Technologies, and Open Issues”, *Proceedings of the IEEE*, Vol. 93, No. 1, January 2005
- [6] “The Multimedia Control Protocol RTCP”, class notes for COMP 249 Advanced Distributed Systems, Department of Computer Science, University of North Carolina at Chapel Hill, 1999
- [7] J. F. Kurose and K. W. Ross, *Computer Networking: A top down approach*, 4th Edition, New York: Addison Wesley, 2008
- [8] J. Kuthan, D. Sisalem, *SIP: More Than You Ever Wanted To Know About*, Tekelec, March 2007
- [9] JVT, *JSVM Software Manual*, Version JSVM 9.17, March 24th, 2009
- [10] Microsoft Corporation, *Module 8: Overview of SIP*
- [11] M. Neophytou, K. Stavrou, V. Vassiliou, A. Pitsillides, “The Importance of Adaptive Applications in Mobile Wireless Networks”, in *Proceedings of the*

International Conference on Software in Telecommunications and Computer Networks, October 2006

- [12] P. Schmitz, G. Weaver, “MIPv6: New Capabilities for Seamless Roaming Among Wired, Wireless, and Cellular Networks”, *Intel Developer Update Magazine*, September 2002
- [13] T. Sikora, “Trends and Perspectives in Image and Video Coding”, in *Proceedings of the IEEE*, Vol. 93, No. 1, January 2005
- [14] T. Stockhammer, M. M. Hannuksela, T. Wiegand , “H.264/AVC in Wireless Environments”, *IEEE Transactions on circuits and systems for video technology*, Vol. 13, No. 7, July 2003
- [15] T. Stockhammer, T. Wiegand, “H.264/AVC for Wireless Applications”, presented at the 8th International Workshop on Mobile Multimedia Communications, Germany, October 2003
- [16] V. Vassiliou, P. Antoniou, I. Giannakou, A. Pitsillides. “Requirements for the Transmission of Streaming Video in Mobile Wireless Networks”, in *International Conference on Artificial Neural Networks*, Athens, Greece, September 2006, pp. 528- 537
- [17] S. Wenger, “H.264/AVC Over IP”, *IEEE Transactions on circuits and systems for video technology*, Vol. 13, No. 7, July 2003
- [18] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, A. Luthra, “Overview of the H.264/AVC Video Coding Standard”, *IEEE Transactions on circuits and systems for video technology*, Vol. 13, No. 7, July 2003
- [19] J. R. Ohm, “Advances in Scalable Video Coding”, *Proceedings of the IEEE*, Vol. 93, No. 1, January 2005
- [20] Χ.Γ. Κουμαράς, “Μέθοδος Πρόβλεψης της Αντιληπτής Ποιότητας Ψηφιακού Βίντεο Συναρτήσεως του Ρυθμού Κωδικοποίησης και της Δυναμικής του Περιεχομένου”, Διδακτορική Διατριβή, Σεπτέμβριος 2007
- [21] Λ. Μιχαήλ, “Χρήση Multicast για Μετάδοση Πολυμεσικού Περιεχομένου σε Κινητά Ipv6 Δίκτυα”, Ατομική Διπλωματική Εργασία, 2007
- [22] Χ. Μπούρας, “Πρωτόκολλα Πραγματικού Χρόνου και Μετάδοση Πολυμέσων”, Πανεπιστήμιο Πατρών, Πολυτεχνική Σχολή

- [23] Ι. Συκοπετρίτης, “Βελτιστοποίηση της ποιότητας των βιντεοροών συναρτήσει ανάδρασης και μετάδοση τους σε στρώματα”, Διατριβή Μάστερ, Μάιος 2006
- [24] Σ. Χριστοδούλου, Χρήση Multicast για Μετάδοση Πολυμεσικού Περιεχομένου σε Κινητά Ipv6 Δίκτυα”, Ατομική Διπλωματική Εργασία, 2008

ΠΑΡΑΡΤΗΜΑ Α

ΒΑΣΙΚΟΣ ΚΩΔΙΚΑΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΕ ΣΤΗΝ ΥΛΟΠΟΙΗΣΗ

Φόρμα 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();
        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;
        int packetsSentInPeriod = 0;
        double NECNSC = 0;
        bool hysteresis = false, prevHysteresis=false;
        double currentBW = 0, previousBW = 0, toCheckBW = 0;
        double progressRatio = 0;
        int InitialBW=0;
        bool extractNew = false;
        TextWriter tw;

        public ADIVISProg()
        {
```

```

        InitializeComponent();
    }

    private void ADIVISProg_Load(object sender, EventArgs e)
    {

    }

    private void writeStreamInfo(object sender,
TestProcessCaller.DataReceivedEventArgs e)
    {
        this.txtResults.AppendText(e.Text + Environment.NewLine);
        Application.DoEvents();
    }

    private void processCompletedOrCanceled(object sender, EventArgs
e)
    {
        this.Cursor = Cursors.Default;
        this.txtResults.AppendText("Ended" + Environment.NewLine);
        Application.DoEvents();
    }

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

```

```

        column = 0;

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

    }

    public void OnTimer(Object source, ElapsedEventArgs e)
    {
        timerClock.Stop();
        string line;
        prevHysteresis = hysteresis;
        currentLRPS = RandomClass.NextDouble();
        DLRPS = currentLRPS - previousLRPS;

        //Calculate scaled NECN and DNECN only if any packets were
sent during this period
        if (packetsSentInPeriod > 0)
        {
            currentNECN = RandomClass.NextDouble();
            NECNSC = currentNECN / packetsSentInPeriod;
            DNECN = NECNSC - previousNECN;
        }

        //Increment time
        this.clockTime+=500;
        //Write parameters to log file
        line = "Time(in ms)= " + this.clockTime.ToString() + " DLRPS="
+ DLRPS.ToString() + " DNECN= " + DNECN.ToString() + " Packets Sent= "
+ packetsSentInPeriod.ToString();
        tw.WriteLine(line);

        calculateDefuzzied(DLRPS, DNECN);

        //After calling function to calculate defuzzifiedA determine
what happens with BW
        if (packetsSentInPeriod > 0)
        {
            if (defuzzifiedA > 1.0 && hysteresis == false) //Bandwidth
increases, enter hysteresis phase
            {
                hysteresis = true;
                toCheckBW = currentBW;
                previousBW = currentBW;
                currentBW = defuzzifiedA * currentBW;
                if (currentBW > InitialBW)
                    currentBW = InitialBW;
                extractNew = false;
            }
            else if (defuzzifiedA < 1.0) //Bandwidth decreases

```

```

        {
            previousBW = currentBW;
            //if defuzzified output == 0 do not multiply since
bandwidth becomes 0
            if (defuzzifiedA != 0.0)
                currentBW = currentBW * defuzzifiedA;
            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;
            }
            if (currentBW < toCheckBW)
                hysteresis = false;

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

            extractNew = true; //flag to signal main to proceed
to new extraction
            hysteresis = false;
            if (currentBW > InitialBW)
                currentBW = InitialBW;
        }
    }
    else
    {
        extractNew = false;

    }

    line = "Current BW= " + currentBW.ToString() + " Defuzzified
Output= " + defuzzifiedA.ToString() + "Previous Hysteresis= " +
prevHysteresis.ToString() + " Current Hysteresis=
"+hysteresis.ToString()+ " New Extraction= " + extractNew.ToString();
    tw.WriteLine(line);
    tw.WriteLine("\n");
    Application.DoEvents();
    //Change variables accordingly
    previousNECN = NECNSC;
    previousLRPS = currentLRPS;

```

```

        packetsSentInPeriod = 0;
        timerClock.Start();
    }

    //Main function of current form
    private void btnMeasure_Click(object sender, EventArgs e)
    {
        DialogResult result;
        //Timer initialization, enabled when streaming begins
        timerClock.Elapsed += new ElapsedEventHandler(OnTimer);
        timerClock.Interval = 500;
        timerClock.Enabled = false;
        tw = new StreamWriter("ADIVIS.log");
        const int RtpPacketSize = 1456; //Bytes of payload in Rtp
        Packet if IP MTU=1500 bytes
        long filesize;
        string command = "";
        string toTW = "";

        btnStream.Enabled = false;
        btnMainMenu.Enabled = false;

        //Check for correctness of given input data
        if (txtFile.Text == "")
        {
            MessageBox.Show("Input File is Required!!!!");
            btnStream.Enabled = true;
            btnMainMenu.Enabled = true;
        }
        else if ((txtBW.Text == "") || (!((int.TryParse(txtBW.Text,
        out InitialBW))))))
        {
            MessageBox.Show("Link BandWidth must be an Integer!!!!");
            btnStream.Enabled = true;
            btnMainMenu.Enabled = true;
        }
        else
        {
            //Extract Initial Sequence

            if (chbSet.Checked == true)
            {
                command = txtFile.Text + " " + "Streamed.264" + " " +
                txtWidth.Text + " " + txtHeight.Text + " " + txtFPS.Text + " " +
                ((double)(InitialBW * 0.85) / 1000) + " " + "AVC_Streamed.avc"; ;
            }
            else
            {
                command = txtFile.Text + " " + "Streamed.264" + " " +
                ((double)(InitialBW * 0.85) / 1000) + " " + "AVC_Streamed.avc";
            }
            TestProcessCaller.ProcessCaller processCaller = new
            TestProcessCaller.ProcessCaller(this);
            //processCaller.FileName = @"..\..\hello.bat";
            if (chbSet.Checked == true)

```

```

        {
            processCaller.FileName = @"extractor_AVC.bat";
        }
        else
        {
            processCaller.FileName = @"extractor_OnlyBR_AVC.bat";
        }
        processCaller.Arguments = command;
        processCaller.StdErrReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
        processCaller.StdoutReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
        processCaller.Completed += new
EventHandler(processCompletedOrCanceled);
        processCaller.Cancelled += new
EventHandler(processCompletedOrCanceled);
        this.txtResults.Text = "Extracting Initial Sequence.
Please stand by.." + Environment.NewLine;

        processCaller.Start();

        Application.DoEvents();
        //Get Size of input file
        System.IO.FileInfo fi = new
System.IO.FileInfo("AVC_Streamed.avc");
        filesize = fi.Length;
        long constFileLength= filesize;
        string timeToSleepStr = (RtpPacketSize /
(double)((double)(InitialBW * 0.85) / 8)).ToString("F3");
        string[] ret = timeToSleepStr.Split('.');
        long timeToSleep = 0;
        long.TryParse(ret[1], out timeToSleep);
        int DataSent=0;

        //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() + " Packets
Sent= " + packetsSentInPeriod.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();
        Application.DoEvents();

```

```

//Beginning of the streaming
while (filesize > 0)
{
    //Begin timer to check every period(T=0.5 seconds)
    timerClock.Enabled = true;
    Application.DoEvents();
    filesize -= RtpPacketSize;
    DataSent+=RtpPacketSize;
    packetsSentInPeriod++;
    System.Threading.Thread.Sleep((int) timeToSleep);

    //Check if a transition to new layer must be made
    //Keep user preferences for framerate and spatial
resolution
    if (extractNew == true)
    {
        if (chbSet.Checked == true)
        {
            command = txtFile.Text + " " + "Streamed.264"
+ " " + txtWidth.Text + " " + txtHeight.Text + " " + txtFPS.Text + " " +
((double)(currentBW * 0.85) / 1000)+" "+"AVC_Streamed.avc";
            processCaller.FileName = @"extractor_AVC.bat";
        }
        //Decide for new layer only based on current
Bandwidth
        else if (chbSet.Checked == false)
        {
            command = txtFile.Text + " " + "Streamed.264"
+ " " + ((double)(currentBW * 0.85) / 1000)+" "+"AVC_Streamed.avc";
            processCaller.FileName =
@"extractor_OnlyBR_AVC.bat";
        }
        System.Console.Out.WriteLine(command);
        processCaller.Arguments = command;
        processCaller.StderrReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
        processCaller.StdoutReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
        processCaller.Completed += new
EventHandler(processCompletedOrCanceled);
        processCaller.Cancelled += new
EventHandler(processCompletedOrCanceled);
        this.txtResults.Text = "Performing Transition to
new layer. Please stand by.." + Environment.NewLine;
        tw.WriteLine(command);
        Application.DoEvents();
        processCaller.Start();

        //Get new filesize
        try
        {
            System.IO.FileInfo file = new
System.IO.FileInfo("AVC_Streamed.avc");
            filesize = file.Length;

```

```

        }
        catch (FileNotFoundException)
        {
            this.txtResults.AppendText("Could not Create
AVC Stream from the given SVC stream. Terminating" + Environment.NewLine);
            MessageBox.Show("Error Creating AVC Stream
from SVC Stream!!\nTerminating Program Execution!!");
            this.Close();
        }
        //Calculate ratio of sent data until now
        progressRatio = (double)((double)DataSent /
(double)constFileLength);
        //Assumption of sent data based on ration from
previously streamed file
        DataSent =Convert.ToInt32(progressRatio *
constFileLength);
        //Store filesize of new bitstream
        constFileLength = filesize;
        //Get size in bytes of data left to stream
        filesize -= DataSent;

        timeToSleepStr = (RtpPacketSize /
(double)((double)(currentBW * 0.85) / 8)).ToString("F3");
        ret = timeToSleepStr.Split('.');
        long.TryParse(ret[1], out timeToSleep);

        extractNew = false;
    }
    //Used for Progress Bar
    int prbValue=((int) ((double)((double)DataSent /
(double)constFileLength)*100));
    if(prbValue>100)
        prbValue=100;
    prbProgress.Value=prbValue;

}
//Stop Timer after the completion of the streaming
timerClock.Enabled = false;
clockTime = 0;
btnStream.Enabled = true;
btnMainMenu.Enabled = true;

tw.Close();
result=MessageBox.Show("Log File Created.\nOpen Now????",
"Log File Created", MessageBoxButtons.YesNo);
if (result == DialogResult.Yes)
    System.Diagnostics.Process.Start("ADIVIS.log");
    }
}

private void btnMainMenu_Click(object sender, EventArgs e)
{
    this.Close();
}

```

```

        private void btnBrowse_Click(object sender, EventArgs e)
        {
            InputFileDialog.Filter = "SVC files (*.svc)|*.svc|264 Files (*.264)|*.264";

            if (InputFileDialog.ShowDialog() == DialogResult.OK)
            {
                string[] fileArray = InputFileDialog.FileName.Split('\\');

                txtFile.Text = fileArray[fileArray.Length - 1];
            }
        }

        private void chbSet_CheckedChanged(object sender, EventArgs e)
        {
            if (chbSet.Checked == true && (txtFPS.Text == "" || txtHeight.Text == "" || txtWidth.Text == ""))
            {
                MessageBox.Show("All three Attributes are Required!!!!");
                chbSet.Checked=false;
            }
        }
    }
}

```

Κλάση CommonFunctions

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;
using System.IO;
using System.Data;

namespace ADIVIS.Common
{
    public class CommonFunctions
    {
        public static string[] ProcessDir(string sourceDir)
        {
            // Process the list of files found in the directory.
            string[] fileEntries = Directory.GetFiles(sourceDir);
            return fileEntries;
        }

        public static string ExecuteCommandSync(object command)
        {
            System.Diagnostics.ProcessStartInfo procStartInfo = new
            System.Diagnostics.ProcessStartInfo("cmd", "/c " + command);
            procStartInfo.RedirectStandardOutput = true;
            procStartInfo.UseShellExecute = false;

```

```

        //procStartInfo.CreateNoWindow = true;
        System.Diagnostics.Process proc = new
System.Diagnostics.Process();
        proc.StartInfo = procStartInfo;
        proc.Start();

        string result = proc.StandardOutput.ReadToEnd();
        return result;
    }

    public static void LoadFilesInDropDownLists(string Dir, ArrayList
FileType, System.Windows.Forms.ComboBox ComboBox)
    {
        string[] fileEntries = Directory.GetFiles(Dir);
        for (int i = 0; i < FileType.Count; i++)
        {
            foreach (string fileName in fileEntries)
            {
                if (Dir != null)
                {
                    if (fileName.IndexOf(FileType[i].ToString()) > 0)
                        ComboBox.Items.Add(fileName.Remove(0,
Dir.Length + 1));
                }
                else
                    ComboBox.Items.Add(fileName.Remove(0, Dir.Length
+ 1));
            }
        }
    }

    public static ArrayList FindSupportedFiled(string FilePath,
string XMLTable)
    {
        //string myXMLfile = @"../../Data/SupportedFileTypes.xml";
        ArrayList FileTypesArray = new ArrayList();
        string myXMLfile = FilePath;
        DataSet ds = new DataSet();
        DataTable dt = new DataTable();
        // Create new FileStream with which to read the schema.
        System.IO.FileStream fsReadXml = new System.IO.FileStream
(myXMLfile, System.IO.FileMode.Open);
        ds.ReadXml(fsReadXml);
        dt = ds.Tables[XMLTable];
        if (dt != null)
            foreach (DataRow row in dt.Rows)
                FileTypesArray.Add(row.ItemArray[0].ToString());
        return FileTypesArray;
    }

    public static void readCommonParameters()
    {
        DataSet ds = new DataSet();
        DataTable dt = new DataTable();
        // Create new FileStream with which to read the schema.

```

```

        System.IO.FileStream fsReadXml = new System.IO.FileStream
            (Common.Parameters.PARAMETERS_XML,
System.IO.FileMode.Open);
        ds.ReadXml(fsReadXml);
        dt = ds.Tables[0];
        Common.Parameters.DELAY_BETWEEN_FRAMES =
Int32.Parse(dt.Rows[0]["Value"].ToString());
        Common.Parameters.NUMBER_OF_FRAMES_TO_DECODE =
Int32.Parse(dt.Rows[1]["Value"].ToString());
        fsReadXml.Close();
    }

    public static ArrayList FindFavoriteIpsArraylist()
    {
        ArrayList favorites = new ArrayList();
        DataSet ds = new DataSet();
        DataTable dt = new DataTable();
        // Create new FileStream with which to read the schema.
        System.IO.FileStream fsReadXml = new System.IO.FileStream
            (Common.Parameters.FAVORITE_IPS_XML,
System.IO.FileMode.Open);
        ds.ReadXml(fsReadXml);
        fsReadXml.Close();
        dt = ds.Tables[0];
        if (dt != null)
            foreach (DataRow row in dt.Rows)
                favorites.Add(row.ItemArray[1].ToString());
        return favorites;
    }
}
}

```

Φόρμα BitStreamExtractor

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace ADIVIS
{
    public partial class BitStreamExtractor : Form
    {
        public BitStreamExtractor()
        {
            InitializeComponent();
        }
    }
}

```

```

    }

    private void label11_Click(object sender, EventArgs e)
    {
    }

    private void button2_Click(object sender, EventArgs e)
    {
        InputFileDialog.Filter = "SVC files (*.svc)|*.svc|264 Files (*.264)|*.264";

        if (InputFileDialog.ShowDialog() == DialogResult.OK)
        {
            string[] fileArray = InputFileDialog.FileName.Split('\\');

            txtVideoInput.Text = fileArray[fileArray.Length - 1];
        }
    }

    private void writeStreamInfo(object sender,
TestProcessCaller.DataReceivedEventArgs e)
    {
        this.txtResults.AppendText(e.Text + Environment.NewLine);
    }
    private void processCompletedOrCanceled(object sender, EventArgs
e)
    {
        this.Cursor = Cursors.Default;
        this.txtResults.AppendText("Ended" + Environment.NewLine);
        btnExtract.Enabled = true;
        btnMainMenu.Enabled = true;
    }

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

    private void btnExtract_Click(object sender, EventArgs e)
    {
        string command="";
        btnExtract.Enabled = false;
        btnMainMenu.Enabled = false;
        if (txtVideoInput.Text=="")
        {
            MessageBox.Show("Input File is Required!!");
            btnExtract.Enabled = true;
            btnMainMenu.Enabled = true;
        }
    }

    if(txtWidth.Text!=""
&&txtHeight.Text!=""&&txtBR.Text!=""&&txtFR.Text!="")
    {
        if (txtOutputFile.Text != "")
        {

```

```

        command = txtVideoInput.Text + " " + txtOutputFile.Text +
" " + txtWidth.Text + " " + txtHeight.Text + " " + txtFR.Text + " " +
txtBR.Text;
    }
    else
    {
        txtOutputFile.Text = "defaultOutputFile.264";
        command = txtVideoInput.Text + " " + txtOutputFile.Text+"
" + txtWidth.Text + " " + txtHeight.Text + " " + txtFR.Text + " " +
txtBR.Text;
    }

    TestProcessCaller.ProcessCaller processCaller = new
TestProcessCaller.ProcessCaller(this);
    //processCaller.FileName = @"..\..\hello.bat";
    processCaller.FileName = @"extractor.bat";
    processCaller.Arguments = command;
    processCaller.StdErrReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
    processCaller.StdOutReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
    processCaller.Completed += new
EventHandler(processCompletedOrCanceled);
    processCaller.Cancelled += new
EventHandler(processCompletedOrCanceled);
    // processCaller.Failed += no event handler for this one,
yet.

    this.txtResults.Text = "Started function. Please stand
by.." + Environment.NewLine;

    // the following function starts a process and returns
immediately,
    // thus allowing the form to stay responsive.
    processCaller.Start();

    //txtResults.Text =
Common.CommonFunctions.ExecuteCommandSync(command);
    //txtResults.Visible = true;
}
else if (txtWidth.Text == "" && txtHeight.Text == "" &&
txtBR.Text != "" && txtFR.Text == "")
{
    if (txtOutputFile.Text != "")
    {
        command = txtVideoInput.Text + " " + txtOutputFile.Text +
" " + txtBR.Text;
    }
    else
    {
        txtOutputFile.Text = "defaultOutputFile.264";
        command = txtVideoInput.Text + " "+ txtOutputFile.Text +
" "+ txtBR.Text;
    }
    TestProcessCaller.ProcessCaller processCaller = new
TestProcessCaller.ProcessCaller(this);
    //processCaller.FileName = @"..\..\hello.bat";

```

```

        processCaller.FileName = @"extractor_onlyBR.bat";
        processCaller.Arguments = command;
        processCaller.StdErrReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
        processCaller.StdOutReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
        processCaller.Completed += new
EventHandler(processCompletedOrCanceled);
        processCaller.Cancelled += new
EventHandler(processCompletedOrCanceled);
        // processCaller.Failed += no event handler for this one, yet.

        this.txtResults.Text = "Started function. Please stand by.."
+ Environment.NewLine;

        // the following function starts a process and returns
immediately,
        // thus allowing the form to stay responsive.
        processCaller.Start();

        //txtResults.Text =
Common.CommonFunctions.ExecuteCommandSync(command);
        //txtResults.Visible = true;
    }
    else if (txtWidth.Text == "" && txtHeight.Text == "" &&
txtBR.Text == "" && txtFR.Text == "" && txtOutputFile.Text == "")
    {
        command = txtVideoInput.Text;
        TestProcessCaller.ProcessCaller processCaller = new
TestProcessCaller.ProcessCaller(this);
        //processCaller.FileName = @"..\..\hello.bat";
        processCaller.FileName = @"extractor_OnlyLayers.bat";
        processCaller.Arguments = command;
        processCaller.StdErrReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
        processCaller.StdOutReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
        processCaller.Completed += new
EventHandler(processCompletedOrCanceled);
        processCaller.Cancelled += new
EventHandler(processCompletedOrCanceled);
        // processCaller.Failed += no event handler for this one, yet.

        this.txtResults.Text = "Started function. Please stand by.."
+ Environment.NewLine;

        // the following function starts a process and returns
immediately,
        // thus allowing the form to stay responsive.
        processCaller.Start();

        //txtResults.Text =
Common.CommonFunctions.ExecuteCommandSync(command);
        //txtResults.Visible = true;
    }
}

```

```

        private void btnMainMenu_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

```

Κλάση ProcessCaller

```

using System;
using System.Diagnostics;
using System.IO;
using System.Text;
using System.Threading;
using System.Windows.Forms;
using System.Collections;
using System.ComponentModel;

namespace TestProcessCaller
{
    /// <summary>
    /// Delegate used by the events StdOutReceived and
    /// StdErrReceived...
    /// </summary>
    public delegate void DataReceivedHandler(object sender,
        DataReceivedEventArgs e);

    /// <summary>
    /// Event Args for above delegate
    /// </summary>
    public class DataReceivedEventArgs : EventArgs
    {
        /// <summary>
        /// The text that was received
        /// </summary>
        public string Text;
        /// <summary>
        /// Constructor
        /// </summary>
        /// <param name="text">The text that was received for this event
to be triggered.</param>
        public DataReceivedEventArgs(string text)
        {
            Text = text;
        }
    }

    public class ProcessCaller : AsyncOperation
    {
        /// <summary>
        /// The command to run (should be made into a property)

```

```

    /// </summary>
    public string FileName;
    /// <summary>
    /// The Arguments for the cmd (should be made into a property)
    /// </summary>
    public string Arguments;

    /// <summary>
    /// The WorkingDirectory (should be made into a property)
    /// </summary>
    public string WorkingDirectory;

    /// <summary>
    /// Fired for every line of stdout received.
    /// </summary>
    public event DataReceivedHandler StdOutReceived;

    /// <summary>
    /// Fired for every line of stderr received.
    /// </summary>
    public event DataReceivedHandler StdErrReceived;

    /// <summary>
    /// Amount of time to sleep on threads while waiting
    /// for the process to finish.
    /// </summary>
    public int SleepTime = 500;

    /// <summary>
    /// The process used to run your task
    /// </summary>
    private Process process;

    /// <summary>
    /// Initialises a ProcessCaller with an association to the
    /// supplied ISynchronizeInvoke. All events raised from this
    /// object will be delivered via this target. (This might be a
    /// Control object, so events would be delivered to that
Control's
    /// UI thread.)
    /// </summary>
    /// <param name="isi">An object implementing the
    /// ISynchronizeInvoke interface. All events will be delivered
    /// through this target, ensuring that they are delivered to the
    /// correct thread.</param>
    public ProcessCaller(ISynchronizeInvoke isi)
        : base(isi)
    {
    }

    /// <summary>
    /// Launch a process, but do not return until the process has
exited.
    /// That way we can kill the process if a cancel is requested.
    /// </summary>

```

```

protected override void DoWork()
{
    StartProcess();

    // Wait for the process to end, or cancel it
    while (! process.HasExited)
    {
        Thread.Sleep(SleepTime); // sleep
        if (CancelRequested)
        {
            // Not a very nice way to end a process,
            // but effective.
            process.Kill();
            AcknowledgeCancel();
        }
    }
}

/// <summary>
/// This method is generally called by DoWork()
/// which is called by the base class Start()
/// </summary>
protected virtual void StartProcess()
{
    // Start a new process for the cmd
    process = new Process();
    process.StartInfo.UseShellExecute = false;
    process.StartInfo.RedirectStandardOutput = true;
    process.StartInfo.RedirectStandardError = true;
    process.StartInfo.CreateNoWindow = true;
    process.StartInfo.FileName = FileName;
    process.StartInfo.Arguments = Arguments;
    process.StartInfo.WorkingDirectory = WorkingDirectory;
    process.Start();

    // Invoke stdout and stderr readers - each
    // has its own thread to guarantee that they aren't
    // blocked by, or cause a block to, the actual
    // process running (or the gui).
    new MethodInvoker(ReadStdOut).BeginInvoke(null, null);
    new MethodInvoker(ReadStdErr).BeginInvoke(null, null);
}

/// <summary>
/// Handles reading of stdout and firing an event for
/// every line read
/// </summary>
protected virtual void ReadStdOut()
{
    string str;
    while ((str = process.StandardOutput.ReadLine()) != null)
    {
        FireAsync(StdOutReceived, this, new
DataReceivedEventArgs(str));
    }
}

```

```

        Application.DoEvents();
    }
}

/// <summary>
/// Handles reading of stderr
/// </summary>
protected virtual void ReadStdErr()
{
    string str;
    while ((str = process.StandardError.ReadLine()) != null)
    {
        FireAsync(StdErrReceived, this, new
DataReceivedEventArgs(str));
        Application.DoEvents();
    }
}
}
}

```

Φόρμα Encoder

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace ADIVIS
{
    public partial class Encoder : Form
    {
        public Encoder()
        {
            InitializeComponent();

            private void button2_Click(object sender, EventArgs e)
            {
                InputFileDialog.Filter = "Configuration files (*.cfg)|*.cfg";
                if (InputFileDialog.ShowDialog() == DialogResult.OK)
                {
                    string[] fileArray = InputFileDialog.FileName.Split('\\');

                    textBox3.Text = fileArray[fileArray.Length - 1];
                }
            }

            private void writeStreamInfo(object sender,
TestProcessCaller.DataReceivedEventArgs e)
            {
                this.txtResults.AppendText(e.Text + Environment.NewLine);
            }
        }
    }
}

```

```

e) private void processCompletedOrCanceled(object sender, EventArgs
{
    this.Cursor = Cursors.Default;
    this.txtResults.AppendText("Ended" + Environment.NewLine);
    button1.Enabled = true;
    btnMainMenu.Enabled = true;
}
private void button1_Click(object sender, EventArgs e)
{
    button1.Enabled = false;
    btnMainMenu.Enabled = false;
    string command = "";

    if (textBox3.Text == "")
    {
        MessageBox.Show("Input and output files are required!!");
        button1.Enabled = true;
        btnMainMenu.Enabled = true;
    }
    else
    {
        command = " -pf " + textBox3.Text;
        TestProcessCaller.ProcessCaller processCaller = new
TestProcessCaller.ProcessCaller(this);
        //processCaller.FileName = @"..\..\hello.bat";
        processCaller.FileName =
@"H264AVCEncoderLibTestStatic.exe";
        processCaller.Arguments = command;
        processCaller.StdErrReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
        processCaller.StdOutReceived += new
TestProcessCaller.DataReceivedHandler(writeStreamInfo);
        processCaller.Completed += new
EventHandler(processCompletedOrCanceled);
        processCaller.Cancelled += new
EventHandler(processCompletedOrCanceled);
        // processCaller.Failed += no event handler for this one,
yet.

        this.txtResults.Text = "Started function. Please stand
by.." + Environment.NewLine;

        // the following function starts a process and returns
immediately,
        // thus allowing the form to stay responsive.
        processCaller.Start();

        //txtResults.Text =
Common.CommonFunctions.ExecuteCommandSync(command);
        //txtResults.Visible = true;
    }
}

private void button3_Click(object sender, EventArgs e)
{

```

```

        ConfigurationFiles cf = new ConfigurationFiles();
        cf.Show();
    }

    private void txtResults_TextChanged(object sender, EventArgs e)
    {

    }

    private void button4_Click(object sender, EventArgs e)
    {
        BitStreamExtractor bse = new BitStreamExtractor();
        bse.Show();
    }

    private void Encoder_Load(object sender, EventArgs e)
    {

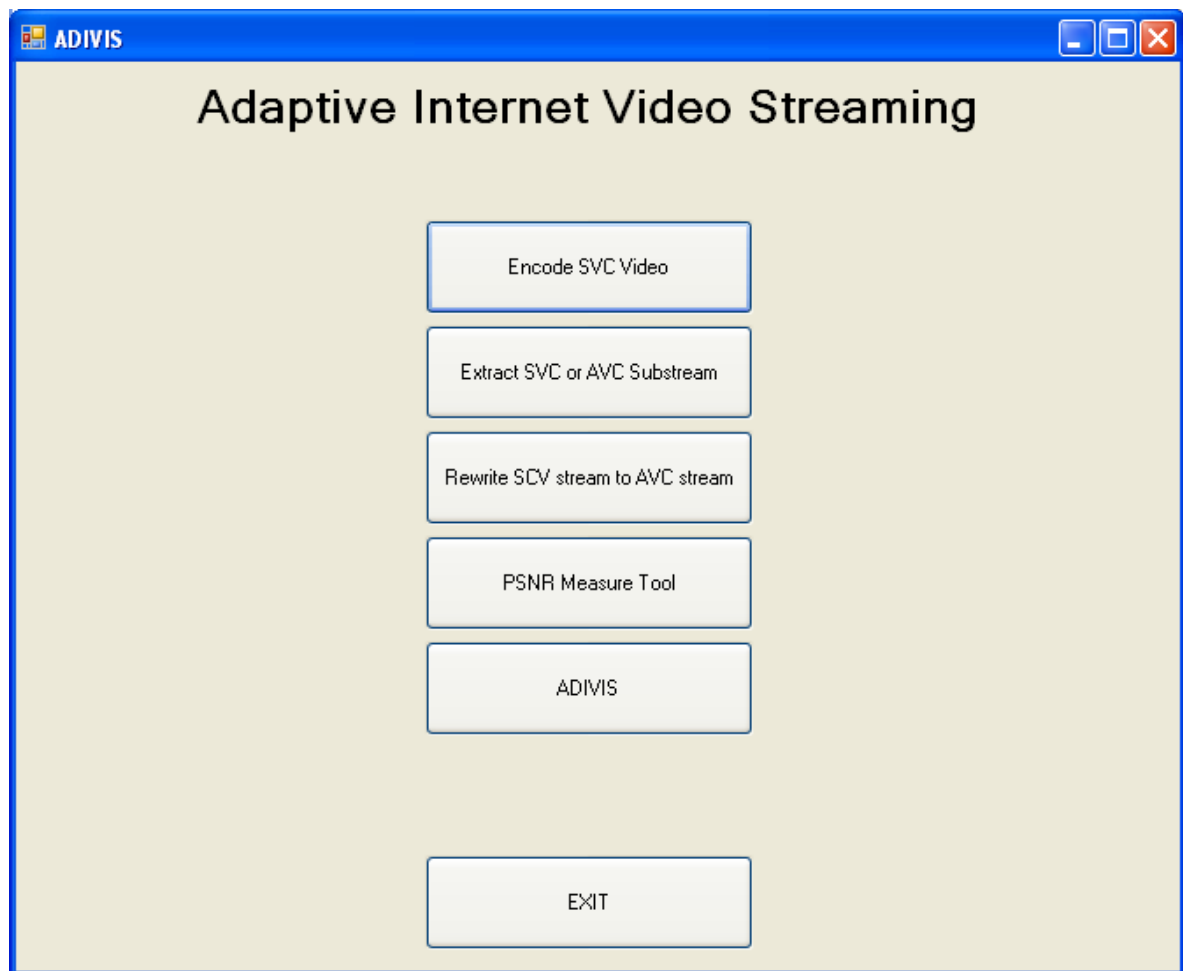
    }

    private void btnMainMenu_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

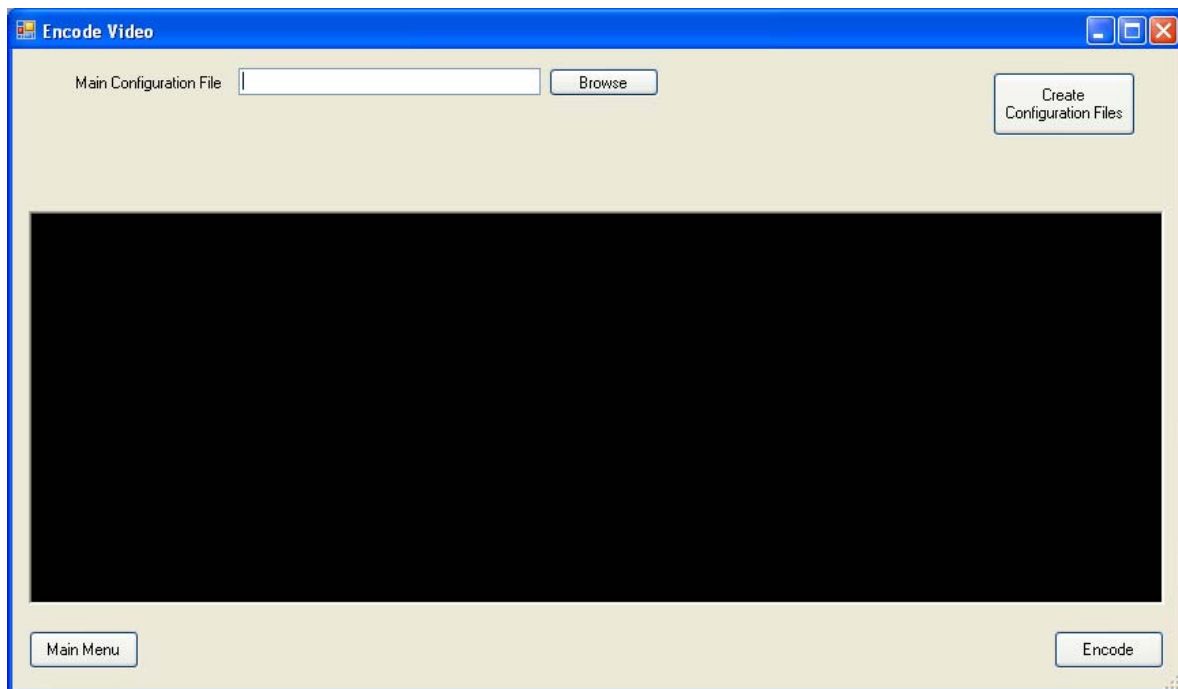
```

ΠΑΡΑΡΤΗΜΑ Β

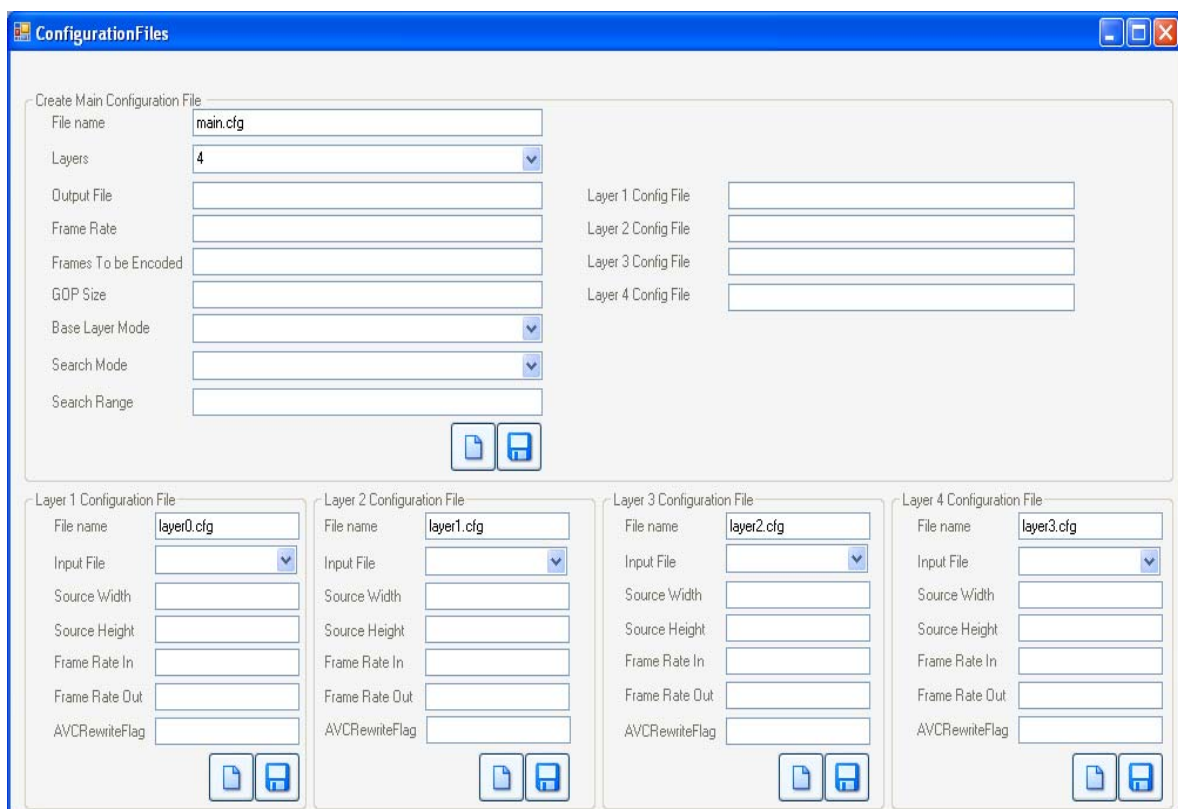
ΒΑΣΙΚΕΣ ΟΘΟΝΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ



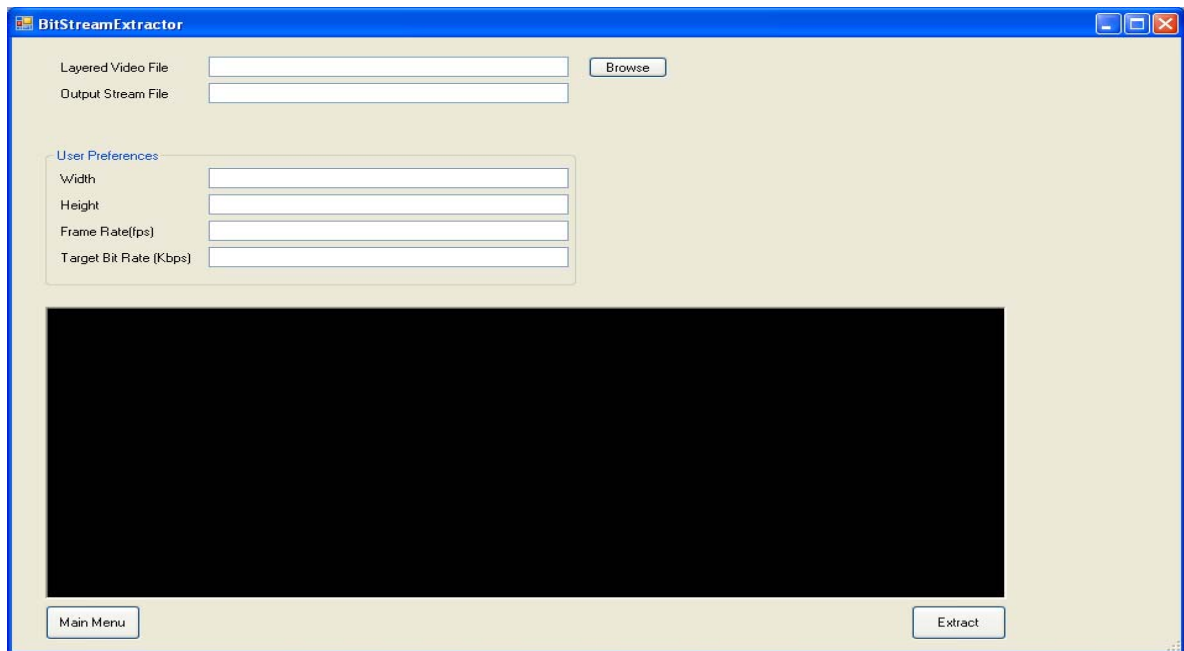
Σχήμα Β-1: Κεντρικό Μενού ADIVIS



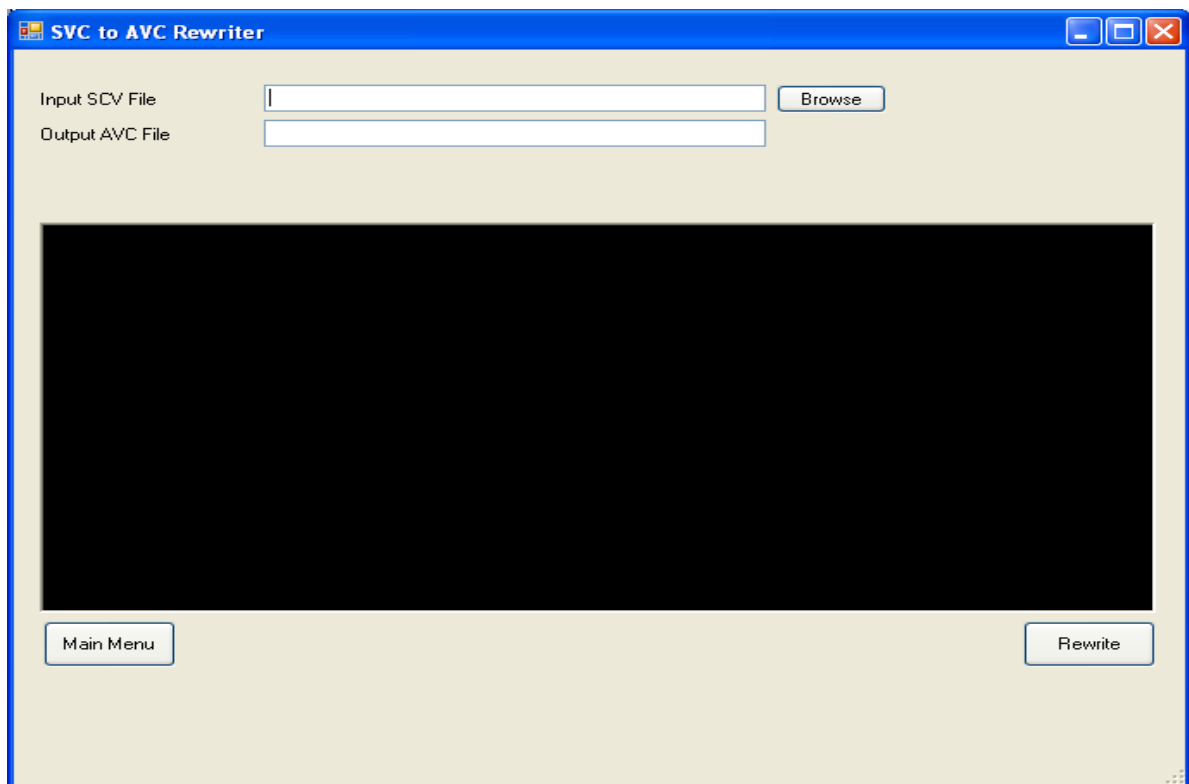
Σχήμα Β-2: Οθόνη Κωδικοποίησης Βίντεο



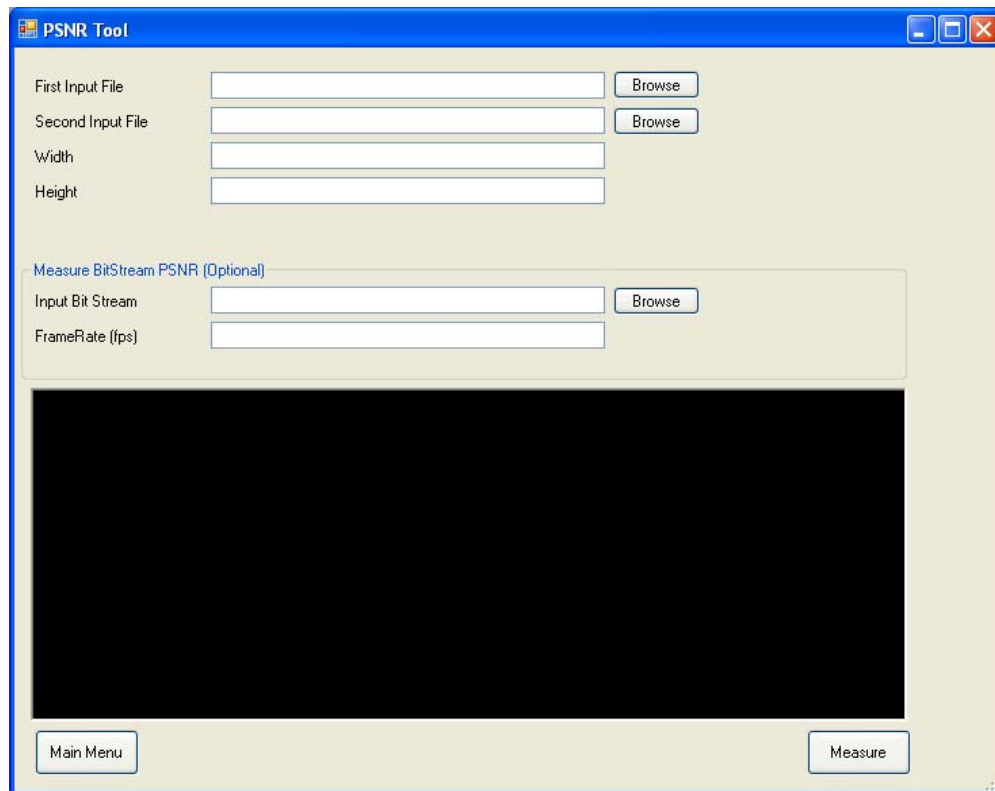
Σχήμα Β-3: Οθόνη Δημιουργίας Αρχείων Ρυθμίσεων



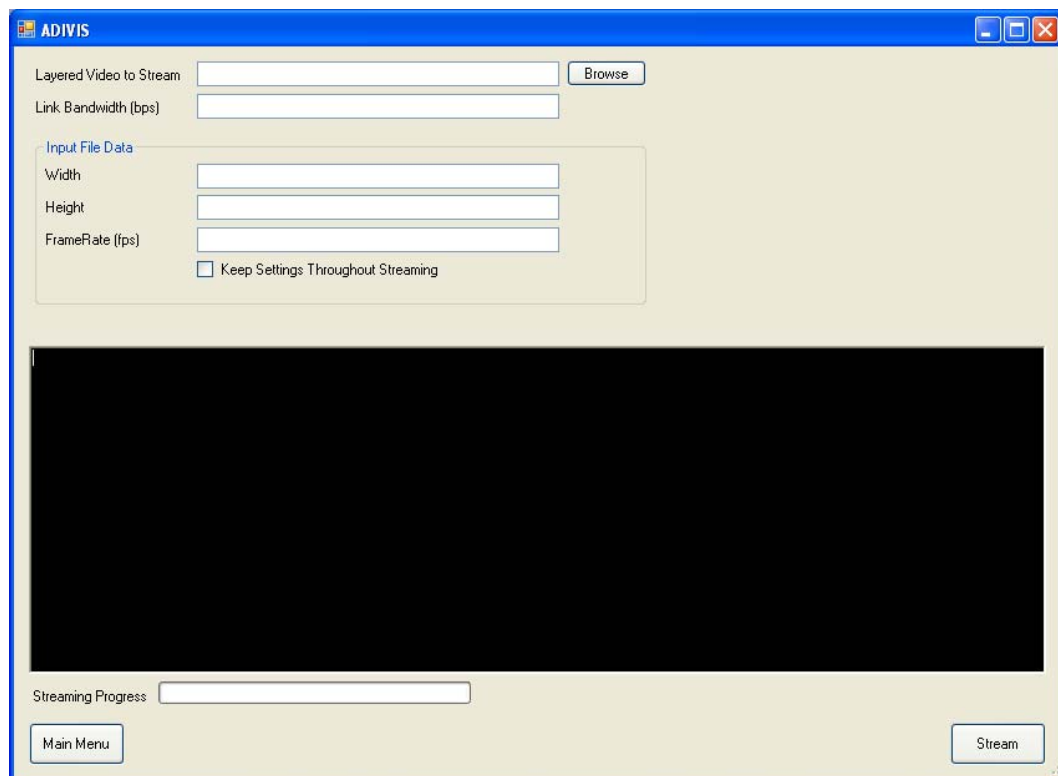
Σχήμα Β-4: Οθόνη Εξαγωγής Υποροών μέσω Bit Stream Extractor



Σχήμα Β-5: Οθόνη Μετατροπής SVC Stream σε AVC Stream



Σχήμα Β-6: Οθόνη Μέτρησης PSNR

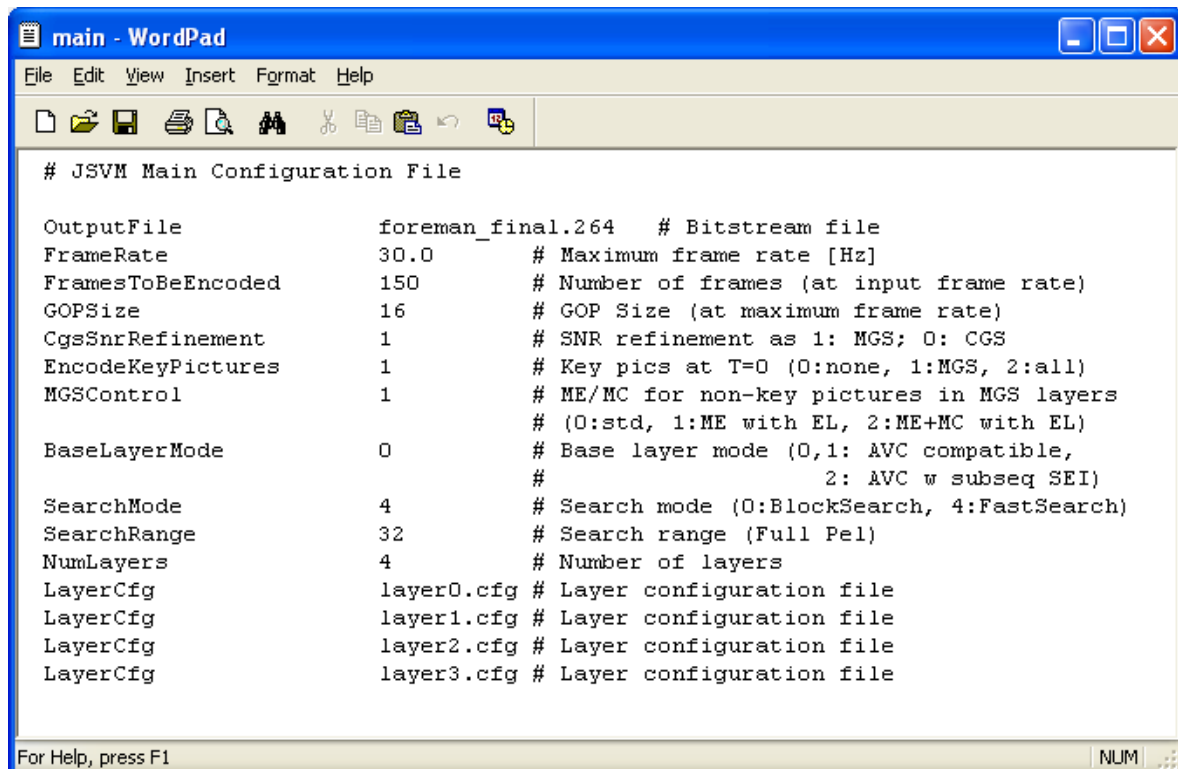


Σχήμα Β-7: Οθόνη Εκτέλεσης ADIVIS

ΠΑΡΑΡΤΗΜΑ Γ

ΑΡΧΕΙΑ ΡΥΘΜΙΣΕΩΝ ΓΙΑ ΚΩΔΙΚΟΠΟΙΗΣΗ SCALABLE VIDEO

Γ.1 Κύριο Αρχείο Ρυθμίσεων

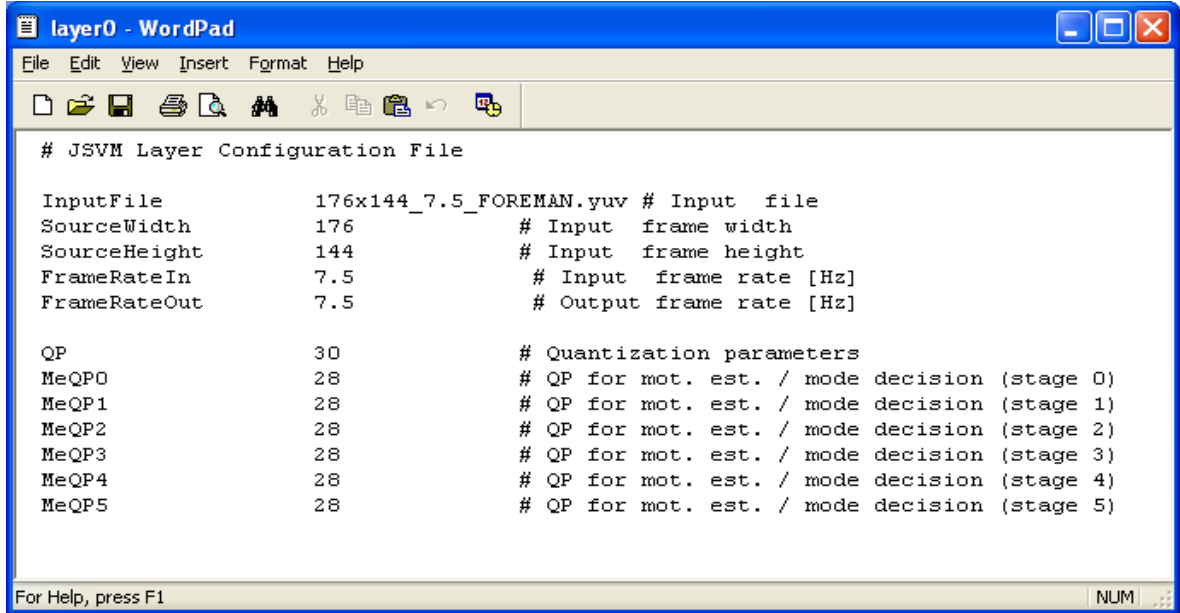


```
# JSVM Main Configuration File

OutputFile          foreman_final.264  # Bitstream file
FrameRate           30.0               # Maximum frame rate [Hz]
FramesToBeEncoded   150                # Number of frames (at input frame rate)
GOPSize             16                 # GOP Size (at maximum frame rate)
CgsSnrRefinement     1                 # SNR refinement as 1: MGS; 0: CGS
EncodeKeyPictures    1                 # Key pics at T=0 (0:none, 1:MGS, 2:all)
MGSControl          1                 # ME/MC for non-key pictures in MGS layers
                                # (0:std, 1:ME with EL, 2:ME+MC with EL)
BaseLayerMode        0                 # Base layer mode (0,1: AVC compatible,
                                #                               2: AVC w subseq SEI)
SearchMode           4                 # Search mode (0:BlockSearch, 4:FastSearch)
SearchRange          32                # Search range (Full Pel)
NumLayers            4                 # Number of layers
LayerCfg             layer0.cfg        # Layer configuration file
LayerCfg             layer1.cfg        # Layer configuration file
LayerCfg             layer2.cfg        # Layer configuration file
LayerCfg             layer3.cfg        # Layer configuration file
```

Γ.2 Αρχεία Ρυθμίσεων Στρωμάτων

Γ.2.1 Στρώμα 0

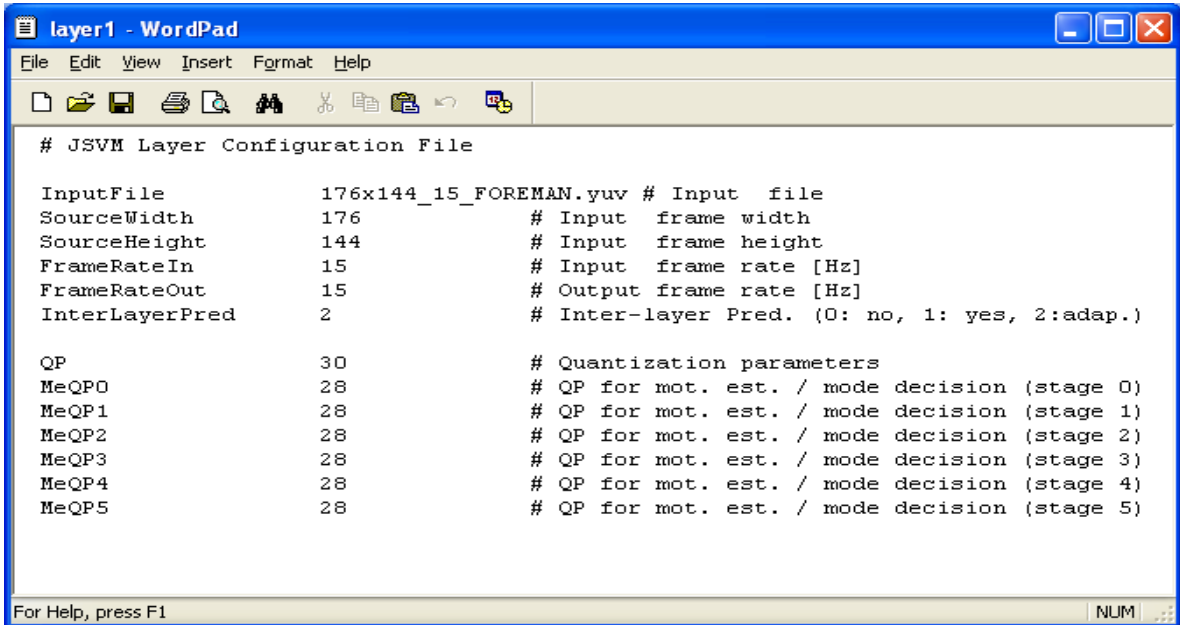


```
# JSVM Layer Configuration File

InputFile          176x144_7.5_FOREMAN.yuv # Input  file
SourceWidth        176                    # Input  frame width
SourceHeight       144                    # Input  frame height
FrameRateIn        7.5                    # Input  frame rate [Hz]
FrameRateOut       7.5                    # Output frame rate [Hz]

QP                 30                    # Quantization parameters
MeQP0              28                    # QP for mot. est. / mode decision (stage 0)
MeQP1              28                    # QP for mot. est. / mode decision (stage 1)
MeQP2              28                    # QP for mot. est. / mode decision (stage 2)
MeQP3              28                    # QP for mot. est. / mode decision (stage 3)
MeQP4              28                    # QP for mot. est. / mode decision (stage 4)
MeQP5              28                    # QP for mot. est. / mode decision (stage 5)
```

Γ.2.2 Στρώμα 1

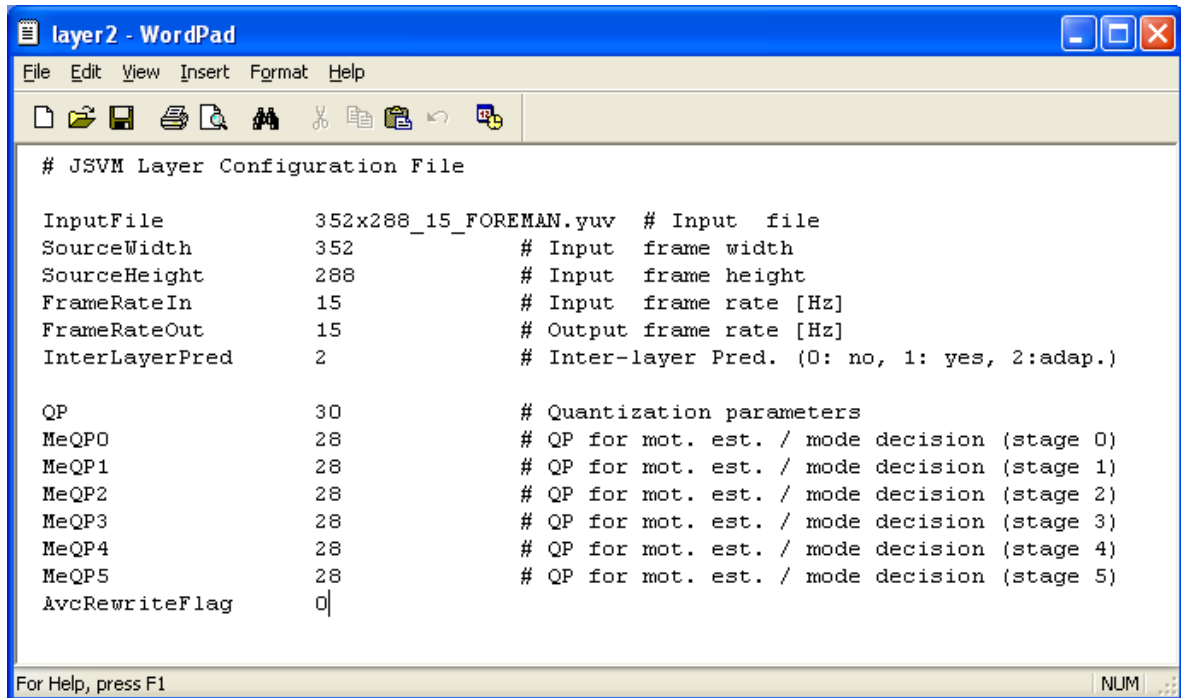


```
# JSVM Layer Configuration File

InputFile          176x144_15_FOREMAN.yuv # Input  file
SourceWidth        176                    # Input  frame width
SourceHeight       144                    # Input  frame height
FrameRateIn        15                    # Input  frame rate [Hz]
FrameRateOut       15                    # Output frame rate [Hz]
InterLayerPred      2                    # Inter-layer Pred. (0: no, 1: yes, 2: adap.)

QP                 30                    # Quantization parameters
MeQP0              28                    # QP for mot. est. / mode decision (stage 0)
MeQP1              28                    # QP for mot. est. / mode decision (stage 1)
MeQP2              28                    # QP for mot. est. / mode decision (stage 2)
MeQP3              28                    # QP for mot. est. / mode decision (stage 3)
MeQP4              28                    # QP for mot. est. / mode decision (stage 4)
MeQP5              28                    # QP for mot. est. / mode decision (stage 5)
```

Γ.2.3 Στρώμα 2

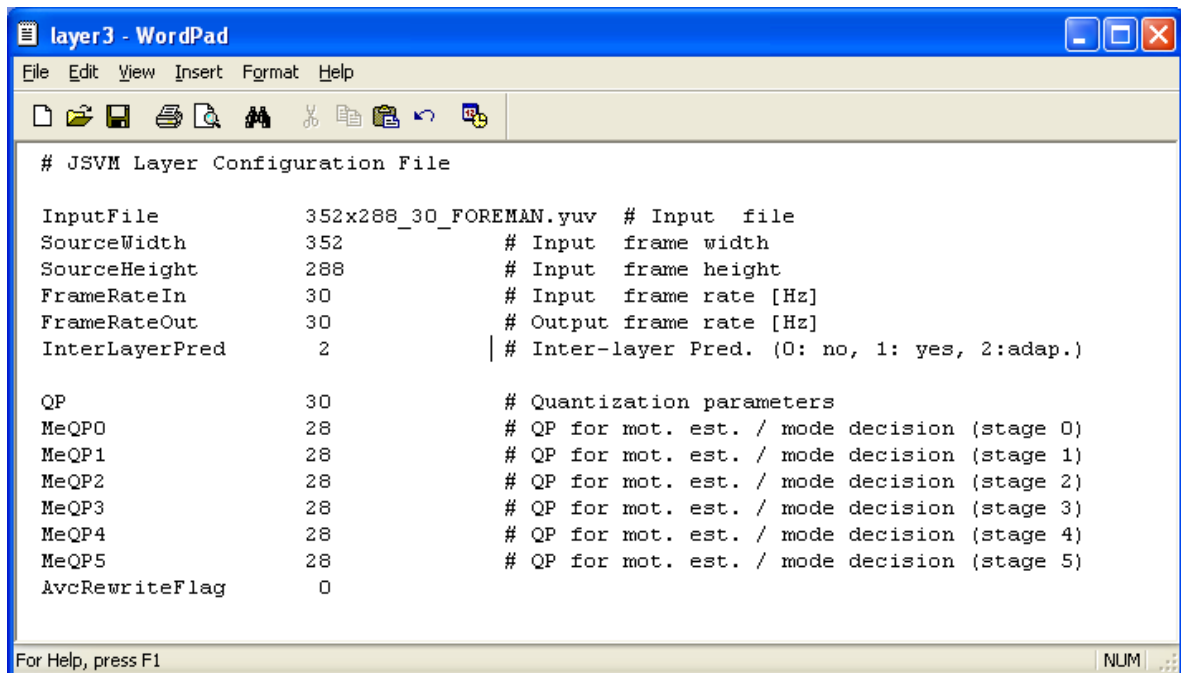


```
# JSVM Layer Configuration File

InputFile          352x288_15_FOREMAN.yuv # Input  file
SourceWidth        352                    # Input  frame width
SourceHeight       288                    # Input  frame height
FrameRateIn        15                    # Input  frame rate [Hz]
FrameRateOut       15                    # Output frame rate [Hz]
InterLayerPred     2                     # Inter-layer Pred. (0: no, 1: yes, 2:adap.)

QP                 30                    # Quantization parameters
MeQP0              28                    # QP for mot. est. / mode decision (stage 0)
MeQP1              28                    # QP for mot. est. / mode decision (stage 1)
MeQP2              28                    # QP for mot. est. / mode decision (stage 2)
MeQP3              28                    # QP for mot. est. / mode decision (stage 3)
MeQP4              28                    # QP for mot. est. / mode decision (stage 4)
MeQP5              28                    # QP for mot. est. / mode decision (stage 5)
AvcRewriteFlag     0
```

Γ.2.4 Στρώμα 3



```
# JSVM Layer Configuration File

InputFile          352x288_30_FOREMAN.yuv # Input  file
SourceWidth        352                    # Input  frame width
SourceHeight       288                    # Input  frame height
FrameRateIn        30                    # Input  frame rate [Hz]
FrameRateOut       30                    # Output frame rate [Hz]
InterLayerPred     2                     # Inter-layer Pred. (0: no, 1: yes, 2:adap.)

QP                 30                    # Quantization parameters
MeQP0              28                    # QP for mot. est. / mode decision (stage 0)
MeQP1              28                    # QP for mot. est. / mode decision (stage 1)
MeQP2              28                    # QP for mot. est. / mode decision (stage 2)
MeQP3              28                    # QP for mot. est. / mode decision (stage 3)
MeQP4              28                    # QP for mot. est. / mode decision (stage 4)
MeQP5              28                    # QP for mot. est. / mode decision (stage 5)
AvcRewriteFlag     0
```