ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μέθοδοι διαχείρισης μνήμης σε διακομιστές μεσολάβησης για πλήρη και μερική παρακολούθηση βίντεο σε πραγματικό χρόνο

ΘΕΟΔΩΡΟΥ ΜΑΡΙΑ

Α.Μ: 2008030055

Εξεταστική επιτροπή: Καθ. Μ.Πατεράκης (Επιβλέπων)

Επ.Καθ Α.Μπλέτσας

Επ.Καθ Π.Κουτσάκης

Χανιά, 2013

Ευχαριστίες

Αρχικά θέλω να ευχαριστήσω την οικογένεια μου, που στηρίζει πάντα τις επιλογές μου και μου δίνει δύναμη να προχωρώ. Ευχαριστώ θερμά τον επιβλέποντα της διπλωματικής εργασίας Καθηγητή κ. Μιχάλη Πατεράκη. Με τις συμβουλές, το ενδιαφέρον, τις γνώσεις και την πολύτιμη καθοδήγησή του κατάφερα να υλοποιήσω επιτυχώς τη διπλωματική εργασία. Επίσης θέλω να ευχαριστήσω τα μέλη της εξεταστικής επιτροπής κ. Μπλέτσα και κ. Κουτσάκη. Τέλος θα ήθελα να ευχαριστήσω όλους τους φίλους μου για τα υπέροχα πέντε χρόνια σπουδών.

Περίληψη

Η εκτεταμένη ανάπτυξη των δικτύων και η εύκολη πρόσβαση σε αυτά έχει οδηγήσει στη συμφόρηση και στην καθυστέρηση που υπόκεινται διάφορες εφαρμογές που απαιτούν την έγκαιρη μετάδοση δεδομένων. Μερικοί χρήστες παρακολουθούν online βίντεο με ταινίες, διαλέξεις, ποδοσφαιρικούς αγώνες κτλ. Οι συγκεκριμένες αιτήσεις απαιτούν τη μετάδοση δεδομένων από το δίκτυο χωρίς καθυστέρηση. Αν υπάρχει χρονική καθυστέρηση στη μεταφορά εικόνας και ήχου ο χρήσης θα παρακολουθεί το βίντεο με παύσεις στην εικόνα, το οποίο είναι αρκετά ενοχλητικό. Για να μπορούν να εξυπηρετηθούν οι χρήστες οι οποίοι απαιτούν μετάδοση δεδομένων σε πραγματικό χρόνο έχουν ενσωματωθεί στο δίκτυο οι διακομιστές μεσολάβησης (proxy servers). Η παρούσα διπλωματική εργασία προτείνει μεθόδους διαχείρισης της διαθέσιμης μνήμης του proxy server για μετάδοση βίντεο σε πραγματικό χρόνο.

Στο πρώτο μέρος της διπλωματικής εργασίας προτείνονται μέθοδοι διαχείρισης μνήμης υπό την προϋπόθεση ότι οι χρήστες παρακολουθούν σειριακά ολόκληρο το βίντεο. Για κάθε βίντεο ορίζεται μια μεταβλητή η οποία μας βοηθά να εκτιμήσουμε την δημοτικότητα των βίντεο και με βάση τη τιμή της συγκεκριμένης μεταβλητής παίρνονται οι αποφάσεις για εισαγωγή ή αφαίρεση κάποιου βίντεο από τη μνήμη του διακομιστή μεσολάβησης. Οι αποφάσεις αυτές στοχεύουν στην διατήρηση των δημοφιλών βίντεο στη μνήμη. Σε κάθε αίτηση οι αλγόριθμοι που προτείνονται προσπαθούν να αποθηκεύουν στη μνήμη του proxy server ολόκληρο το περιεχόμενο των πιο δημοφιλών βίντεο. Σε σχέση με προηγούμενη μελέτη υπό τις ίδιες συνθήκες και σύστημα, η μέθοδος που προτείνουμε πετυχαίνει σημαντική βελτίωση σε ότι αφορά τις παρακάτω μετρικές απόδοσης του συστήματος: (1) Byte - Hit Ratio, (2) καθυστερήσεις κατά το ξεκίνημα των βίντεο και (3) overhead του συστήματος.

Στο δεύτερο μέρος της διπλωματικής εργασίας δίνεται μια ελευθερία κινήσεων στους χρήστες, η παύση. Δηλαδή ο χρήστης μπορεί να παρακολουθήσει ολόκληρο το βίντεο σειριακά ή να παρακολουθήσει ένα αρχικό τμήμα του βίντεο. Προτείνονται δύο νέες μέθοδοι διαχείρισης μνήμης ώστε να μπορούν να εξυπηρετηθούν αρκετές αιτήσεις βίντεο και παράλληλα εξετάζεται και η καλύτερη μέθοδος από το πρώτο μέρος της διπλωματικής εργασίας. Οι τρείς συγκεκριμένες μέθοδοι συγκρίνονται στη συνέχεια με μια μέθοδο που προτάθηκε για μερική παρακολούθηση των βίντεο, από προηγούμενη έρευνα της βιβλιογραφίας. Υλοποιούνται τρία σενάρια για μερική παρακολούθηση των βίντεο και μέσα από αυτά φαίνεται η αποτελεσματικότητα της κάθε μεθόδου. Από τα αποτελέσματα των προσομοιώσεων φαίνεται η βελτίωση στις παρακάτω μετρικές απόδοσης του συστήματος: (1) Byte-Hit Ratio, (2) καθυστερήσεις κατά το ξεκίνημα των βίντεο και (3) overhead του συστήματος. Επιπλέον από το διαφορετικό τρόπο διαχείρισης της

μνήμης από τις τέσσερις μεθόδους που χρησιμοποιούνται και μέσα από τα αποτελέσματα των σεναρίων που εξετάζονται προκύπτουν σημαντικά συμπεράσματα σχετικά με την αποτελεσματικότητα των μεθόδων που υλοποιούν τμηματοποίηση και των μεθόδων που δεν υλοποιούν κάποια τμηματοποίηση στα βίντεο. Καταλήγουμε στο συμπέρασμα ότι η τμηματοποίηση των βίντεο είναι αποδοτική όταν το μεγαλύτερο ποσοστό των χρηστών παρακολουθεί μερικώς το βίντεο. Η πλήρης αποθήκευση των βίντεο είναι αποδοτική όταν το μεγαλύτερο ποσοστό των χρηστών παρακολουθεί ολόκληρο το βίντεο, ακόμα και αν το υπόλοιπο ποσοστό των χρηστών που παρακολουθεί μερικώς το βίντεο σταματά πολύ νωρίς.

Περιεχόμενα

Κεφάλαιο 1 : Εισαγωγή στη μετάδοση δεδομένων πραγματικού χρόνου με τη χρήση διακομιστή μεσολάβησης 1

1.1 Εισαγωγή στην αποθήκευση βίντεο σε διακομιστή μεσολάβησης 1

1.1.1 Μετάδοση δεδομένων σε πραγματικό χρόνο και τα προβλήματα που προκύπτουν 2

1.1.2 Πρωτόκολλα για μετάδοση δεδομένων σε πραγματικό χρόνο [13], [14] 3

1.1.2.1 Resource ReSerVation Protocol (RSVP) 4

1.1.2.2 Real Time Transport Protocol (RTP) 4

1.1.2.3 Real Time Control Protocol (RTCP) 6

1.1.2.4 Real Time Streaming Protocol (RTSP) 6

1.2 Προηγούμενη έρευνα για αποθήκευση βίντεο σε διακομιστή μεσολάβησης 7

1.2.1 Segment Caching 7

1.2.2 Hotspot Caching 9

1.2.3 Prefix caching 9

1.2.4 Prefetching methods 9

1.3 Οργάνωση της εργασίας 12

Κεφάλαιο 2: Μέθοδοι αποθήκευσης και αντικατάστασης βίντεο σε διακομιστή μεσολάβησης και υπολογισμός απόδοσης σε πλήρη παρακολούθηση των βίντεο 14

2.1 Εισαγωγή 14

2.2 Περιγραφή Συστήματος 14

2.2.1 Τοπολογία Συστήματος 15

2.2.2 Μέθοδοι αποθήκευσης και αντικατάστασης 16

2.2.3 Στρατηγικές διαχείρισης της μνήμης 16

2.2.3.1 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη A 17

2.2.3.2 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη B 18

2.2.3.2.1 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη B για τη μεθοδολογία “full video with prefix” 18

2.2.3.2.2 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη B για τη μεθοδολογία “full video with prefix and with limitation” 19

2.2.3.3 Στρατηγική εισαγωγής και αντικατάστασης στηριζόμενη σε ιδέες από το [2] 20

2.3 Αξιολόγηση Απόδοσης 21

2.3.1 Μετρικές απόδοσης 21

2.3.2 Μοντέλο Προσομοίωσης 22

2.3.3 Αποτελέσματα Προσομοίωσης 24

2.3.3.1 Η επίδραση διαφορετικών μεγεθών μνήμης. 25

2.3.3.2 Η επίδραση διαφορετικού αριθμού συνολικών βίντεο στο σύστημα 28

2.3.3.3 Η επίδραση διαφορετικού μέσου μεγέθους βίντεο 30

2.3.3.4 Η επίδραση διαφορετικού διαθέσιμου ποσοστού μνήμης για τα αρχικά τμήματα των βίντεο 32

2.3.3.5 Η επίδραση διαφορετικών τιμών της παραμέτρου skew της κατανομής δημοτικότητας των βίντεο 34

2.3.3.6 Η επίδραση διαφορετικών μέγιστων μετατοπίσεων κατά την αλλαγή δημοτικότητας των βίντεο 37

2.4 Συμπεράσματα από τα αποτελέσματα των προσομοιώσεων 39

Κεφάλαιο 3: Μέθοδοι αποθήκευσης και αντικατάστασης βίντεο σε διακομιστή μεσολάβησης και υπολογισμός απόδοσης σε μερική παρακολούθηση των βίντεο 42

3.1 Εισαγωγή 42

3.2 Περιγραφή Συστήματος 43

3.2.1 Μέθοδοι αποθήκευσης και αντικατάστασης 43

3.2.2 Μέθοδοι διαχείρισης μνήμης και στρατηγικές εισαγωγής και αντικατάστασης 43

3.2.2.1 Lazy Segmentation 43

3.2.2.1.1 Επιθετική στρατηγική εισαγωγής (Aggressive Admission Policy) της “Lazy Segmentation” 44

3.2.2.1.2 Στρατηγική τμηματοποίησης βίντεο «θύματος» της Lazy Segmentation 45

3.2.2.1.3 Επαναληπτική στρατηγική αντικατάστασης δύο φάσεων (Two-Phase Iterative Replacement Policy) της Lazy Segmentation 45
3.2.2.2 Aggressive and Fixed Segmentation 46
3.2.2.2.1 Στρατηγική εισαγωγής και αντικατάστασης στη Μνήμη A 47
3.2.2.2.2 Στρατηγική εισαγωγής και αντικατάστασης στη Μνήμη B 47
3.2.2.2.2.1 Στρατηγική εισαγωγής στη Μνήμη B 47
3.2.2.2.2.2 Στρατηγική αντικατάστασης στη Μνήμη B 47
3.2.2.3 Aggressive and guarded method 49
3.2.2.3.1 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη A 51
3.2.2.3.2 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη B 51
3.2.2.3.2.1 Στρατηγική εισαγωγής στη μνήμη B 51
3.2.2.3.2.2 Στρατηγική αντικατάστασης στη μνήμη B 52
3.2.2.3.3 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη C 52
3.2.2.3.3.1 Στρατηγική εισαγωγής στη μνήμη C 53
3.2.2.3.3.2 Στρατηγική αντικατάστασης στη μνήμη C 53
3.2.2.4 “Full video with prefix and with limitation” 53
3.3 Αξιολόγηση Απόδοσης 54
3.3.1 Μετρικές απόδοσης 54
3.3.2 Μοντέλο Προσομοίωσης 54
3.3.2.1 Σενάρια Προσομοίωσης 55
3.3.2.1.1 Σενάριο 1 55
3.3.2.1.2 Σενάριο 2 56
3.3.2.1.3 Σενάριο 3 56
3.3.3 Αποτελέσματα Προσομοίωσης 56
3.3.3.1 Αποτελέσματα των προσομοιώσεων στο σενάριο 1 57
3.3.3.1.1 Η επίδραση διαφορετικών μεγεθών μνήμης. 58
3.3.3.1.2 Η επίδραση διαφορετικών τιμών της παραμέτρου skew της κατανομής δημοτικότητας των βίντεο 60

3.3.3.2 Αποτελέσματα των προσομοιώσεων στο σενάριο 2 62
3.3.3.2.1 Η επίδραση διαφορετικών μεγεθών μνήμης. 62
3.3.3.2.2 Η επίδραση διαφορετικών τιμών της παραμέτρου skew της κατανομής δημοτικότητας των βίντεο 64
3.3.3.3 Αποτελέσματα των προσομοιώσεων στο σενάριο 3 66
3.3.3.3.1 Η επίδραση διαφορετικών μεγεθών μνήμης. 66
2.3.3.3.2 Η επίδραση διαφορετικών τιμών της παραμέτρου skew της κατανομής δημοτικότητας των βίντεο 68
3.4 Πολυπλοκότητα αλγορίθμων 70
3.5 Συμπεράσματα από τα αποτελέσματα των προσομοιώσεων 70
Κεφάλαιο 4 : Επίλογος 73
4.1 Συνεισφορά διπλωματικής εργασίας 73
4.2 Ιδέες μελλοντικών επεκτάσεων 74
Βιβλιογραφία 76

Κατάλογος Σχημάτων

Σχ.1.1: Παράδειγμα εκθετικής τμηματοποίησης 8

Σχ.1.2: Prefetching Architecture 10

Σχ.1.3: Partial Aggressive Prefetching Pattern 11

Σχ.1.4: The operation of server-assisted prefetching 12

Σχ.2.1: Αρχιτεκτονική συστήματος για μετάδοση δεδομένων σε πραγματικό χρόνο 15

Σχ.2.2: Byte-Hit Ratio σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 27

Σχ.2.3: Καθυστερήσεις στην αρχή του βίντεο σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 28

Σχ.2.4: Αριθμός συνολικών αντικαταστάσεων σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 28

Σχ.2.5: Byte-Hit Ratio σαν συνάρτηση με το συνολικό αριθμό βίντεο στο σύστημα 29

Σχ.2.6: Καθυστερήσεις στην αρχή του βίντεο σαν συνάρτηση με το συνολικό αριθμό βίντεο στο σύστημα 30

Σχ.2.7: Συνολικός αριθμός αντικαταστάσεων σαν συνάρτηση με το συνολικό αριθμό βίντεο στο σύστημα 30

Σχ.2.8: Byte-Hit Ratio σαν συνάρτηση διαφορετικής διάρκειας βίντεο 31

Σχ.2.9: Καθυστερήσεις στο ξεκίνημα των βίντεο σαν συνάρτηση διαφορετικής διάρκειας βίντεο 32

Σχ.2.10: Συνολικός αριθμός αντικαταστάσεων σαν συνάρτηση διαφορετικής διάρκειας βίντεο 32

Σχ.2.11: Byte-Hit Ratio σαν συνάρτηση με το διαθέσιμο ποσοστό της μνήμης A 33

Σχ.2.12: Καθυστερήσεις κατά το ξεκίνημα των βίντεο σαν συνάρτηση με το διαθέσιμο ποσοστό της μνήμης A 34

Σχ.2.13: Συνολικές αντικαταστάσεις σαν συνάρτηση με το διαθέσιμο ποσοστό της μνήμης A 34

Σχ.2.14: Byte-Hit Ratio σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 36

Σχ.2.15: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 36

Σχ.2.16: Αριθμός συνολικών αντικαταστάσεων σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 36

Σχ.2.17: Byte-Hit Ratio σαν συνάρτηση με τη μέγιστη μετατόπιση στη δημοτικότητα των βίντεο 38

Σχ.2.18: Καθυστερήσεις κατά το ξεκίνημα των βίντεο σαν συνάρτηση με τη μέγιστη μετατόπιση στη δημοτικότητα των βίντεο 38

Σχ.2.19: Αριθμός συνολικών αντικαταστάσεων σαν συνάρτηση με τη μέγιστη μετατόπιση δημοτικότητας των βίντεο 39

Σχ.3.1: Byte-Hit Ratio σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 59

Σχ.3.2: Καθυστερήσεις κατά το ξεκίνημα των βίντεο σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 60

Σχ.3.3: Συνολικές αντικαταστάσεις σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 60

Σχ.3.4 Byte-Hit Ratio σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 61

Σχ.3.5: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 61

Σχ.3.6: Συνολικές αντικαταστάσεις σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 62

Σχ.3.7: Byte-Hit Ratio σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 63

Σχ.3.8: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 63

Σχ.3.9: Συνολικές αντικαταστάσεις σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 64

Σχ.3.10: Byte-Hit Ratio σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 65

Σχ.3.11: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 65

Σχ.3.12: Συνολικές αντικαταστάσεις σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 65

Σχ.3.13: Byte-Hit Ratio σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 67

Σχ.3.14: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 67

Σχ.3.15: Συνολικές αντικαταστάσεις σαν συνάρτηση της συνολικής χωρητικότητας μνήμης 68

Σχ.3.16: Byte-Hit Ratio σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 69

Σχ.3.17: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 69

Σχ.3.18: Συνολικές αντικαταστάσεις σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο 69

Κατάλογος Πινάκων

Πίνακας 2.1: Παράμετροι του συστήματος και οι προκαθορισμένες τιμές τους 24

Πίνακας 3.1: Παράδειγμα τμηματοποίησης βίντεο (α) μικρής διάρκειας 252 μπλοκ (β) μεσαίας διάρκειας 1050 μπλοκ και (γ) μεγάλης διάρκειας 3555 μπλοκ 50

Πίνακας 3.2: Παράμετροι του συστήματος και οι προκαθορισμένες τιμές τους 55

Κεφάλαιο 1 : Εισαγωγή στη μετάδοση δεδομένων πραγματικού χρόνου με τη χρήση διακομιστή μεσολάβησης

1.1 Εισαγωγή στην αποθήκευση βίντεο σε διακομιστή μεσολάβησης

Τα δίκτυα υπολογιστών σχεδιάστηκαν για να συνδέουν υπολογιστές που βρίσκονται σε διαφορετικές τοποθεσίες έτσι ώστε να μπορούν να επικοινωνούν και να ανταλλάσσουν δεδομένα. Η εκτεταμένη ανάπτυξη των δικτύων υψηλής ταχύτητας και η εύκολη πρόσβαση σε αυτά έχει οδηγήσει στη συμφόρηση του δικτύου και στην καθυστέρηση που υπόκεινται διάφορες εφαρμογές οι οποίες απαιτούν την έγκαιρη μετάδοση δεδομένων. Ο χρήστης μπορεί πλέον εύκολα να συνδεθεί στο διαδίκτυο όπου και αν βρίσκεται μέσω laptop, smart phone, i-phone ή ακόμα και απλού κινητού κ.α. Στις πρώιμες εποχές των δικτύων το μεγαλύτερο μέρος των δεδομένων που διακινούνταν ήταν μορφής κειμένου. Σήμερα, με την ανάπτυξη των πολυμέσων και των δικτυακών τεχνολογιών, τα πολυμεσικά δεδομένα είναι αυτά που κυριαρχούν και καταναλώνουν τη μερίδα του λέοντος όσον αφορά το διαθέσιμο bandwidth. Μεγάλο μερίδιο χρηστών όταν συνδέεται στο διαδίκτυο θέλει να παρακολουθήσει σε πραγματικό χρόνο διαλέξεις, ταινίες, τηλεόραση κ.α. Με τη ραγδαία αύξηση των χρηστών που μπορούν να έχουν εύκολα πρόσβαση στο διαδίκτυο, μπορεί η μετάδοση δεδομένων σε πραγματικό χρόνο να γίνει προβληματική, εξαιτίας της μεγάλης καθυστέρησης και της αδυναμίας του συστήματος να εξυπηρετήσει όλους τους χρήστες παρέχοντάς τους τη μετάδοση που ζητούν με συνέχεια και χωρίς καθυστερήσεις. Σημαντικό ρόλο στη δυσκολία που έχει το σύστημα για να εξυπηρετήσει άμεσα όλες τις αιτήσεις είναι η μεγάλη διάρκεια που έχουν συνήθως τα βίντεο, οι απαιτήσεις μεγάλου εύρου ζώνης για την υλοποίηση της μετάδοσης δεδομένων σε πραγματικό χρόνο και η ελευθερία των κινήσεων στα βίντεο που έχουν οι χρήστες. Οι παράγοντες αυτοί περιορίζουν τη μετάδοση δεδομένων σε πραγματικό χρόνο και αναλύονται στην επόμενη ενότητα στο 1.1.1.

Τα τελευταία χρόνια η μετάδοση δεδομένων βίντεο σε πραγματικό χρόνο αποτελεί αντικείμενο αρκετών ερευνών. Έχουν προταθεί διάφορες τεχνικές για τη μείωση των προβλημάτων που προκύπτουν από τη μεγάλη αύξηση χρηστών και τις επιπλέον προσδοκίες που έχουν από κάποιες ιστοσελίδες όπως για παράδειγμα με live streaming αναπτύσσοντας και ενσωματώνοντας στο δίκτυο τους διακομιστές μεσολάβησης (proxy servers). Στόχος είναι να αποθηκεύονται στους διακομιστές μεσολάβησης, οι οποίοι τοποθετούνται κοντά στους χρήστες, τα πιο δημοφιλή βίντεο έτσι ώστε να μπορεί το σύστημα να εξυπηρετεί χωρίς καθυστέρηση το μεγαλύτερο ποσοστό των

χρηστών. Επιπλέον εκτός από την ικανοποίηση των χρηστών αποσκοπούν και στη μείωση κίνησης του δικτύου.

1.1.1 Μετάδοση δεδομένων σε πραγματικό χρόνο και τα προβλήματα που προκύπτουν

Για να υπάρχει αξιόπιστη μετάδοση δεδομένων σε πραγματικό χρόνο και για να είναι ικανοποιημένος ο χρήστης πρέπει να ικανοποιούνται οι παρακάτω στόχοι που αφορούν μετρικές απόδοσης, οι οποίοι αποτελούν τα κριτήρια απόδοσης για τις μεθόδους διαχείρισης μνήμης οι οποίες παρουσιάζονται στα Κεφάλαια 2 και 3:

- Υψηλό Byte Hit Ratio.

- Χαμηλή καθυστέρηση κατά το ξεκίνημα του βίντεο, όταν ο χρήστης κάνει μια αίτηση.

- Μείωση στην κίνηση του δικτύου.

Υπάρχουν αρκετοί παράγοντες οι οποίοι περιορίζουν τη μετάδοση δεδομένων σε πραγματικό χρόνο [15]. Ένας σημαντικός παράγοντας είναι η ελευθερία των κινήσεων που δίνεται στους χρήστες όταν παρακολουθούν το βίντεο. Από έρευνες [4], [7]- [11], [20] προκύπτει ότι οι περισσότεροι χρήστες δεν παρακολουθούν το βίντεο σειριακά από την αρχή μέχρι το τέλος. Ειδικότερα προκύπτει ότι οι χρήστες παρακολουθούν ένα μέρος του βίντεο ή χρησιμοποιούν κινήσεις προς τα εμπρός και προς τα πίσω (forward and backward). Επειδή οι χρήστες τις περισσότερες φορές δε γνωρίζουν το περιεχόμενο του βίντεο που ζητούν οι κινήσεις αυτές είναι «τυφλές» [10]. Το αποτέλεσμα είναι ότι περιπλέκεται ο τρόπος που πρέπει το σύστημα να διαχειρίζεται τη μνήμη ώστε αυτή να αξιοποιείται στο έπακρο.

Ένας δεύτερος παράγοντας που προκαλεί προβλήματα στη μετάδοση δεδομένων πραγματικού χρόνου είναι το μεγάλο μέγεθος που έχουν συνήθως τα βίντεο. Για παράδειγμα, ένα MPEG-1 βίντεο έχει περίπου 675 Mbytes, 1-ώρα ταινία με κωδικοποίηση MPEG4, μπορεί να χρειάζεται περισσότερο από 1 GB χωρητικότητα. Αν το βίντεο αποθηκευτεί ολόκληρο στη μνήμη του proxy server, αυτό θα έχει σαν αποτέλεσμα να εξαντληθεί νωρίς η μνήμη του διακομιστή μεσολάβησης, ενώ αν γίνει αποθήκευση μερικών τμημάτων του βίντεο τότε η μνήμη του διακομιστή μεσολάβησης θα εξαντληθεί πιο αργά. Σε κάθε περίπτωση όμως σημασία έχουν οι κινήσεις του χρήστη. Για παράδειγμα αν το βίντεο είναι ολόκληρο αποθηκευμένο μπορεί να εξυπηρετήσει μια αίτηση στην οποία ο χρήστης παρακολουθεί ολόκληρο το περιεχόμενο. Όμως εάν το συγκεκριμένο βίντεο ήταν μερικώς αποθηκευμένο, δεν θα μπορούσε να υλοποιηθεί με επιτυχία η μετάδοση δεδομένων πραγματικού χρόνου, γιατί θα προέκυπταν καθυστερήσεις κατά τη μεταφορά της εικόνας και του ήχου. Μπορεί αντίστοιχα να υπάρχει το αντίθετο σενάριο στο

οποίο ο χρήστης θέλει να παρακολουθήσει ένα πολύ μικρό τμήμα του βίντεο. Στην περίπτωση στην οποία το βίντεο είναι ολόκληρο αποθηκευμένο, το σύστημα μπορεί να εξυπηρετήσει την αίτηση σπαταλώντας όμως αρκετή μνήμη του διακομιστή μεσολάβησης. Στην περίπτωση στην οποία το βίντεο είναι μερικώς αποθηκευμένο, μπορεί να εξυπηρετηθεί η αίτηση χωρίς να γίνεται κατάχρηση της μνήμης.

Τέλος ένας σημαντικός παράγοντας που περιορίζει τη μετάδοση δεδομένων σε πραγματικό χρόνο είναι η εκτεταμένη χρήση σημαντικού ποσοστού εύρου ζώνης (bandwidth) του δικτύου που απαιτείται για να γίνει η μετάδοση. Ως εύρος ζώνης ορίζουμε το μέγιστο ρυθμό μεταφοράς δεδομένων ο οποίος μπορεί να εγκαθιδρυθεί μεταξύ δύο σημείων. Το δίκτυο εξυπηρετεί ταυτόχρονα πολλούς χρήστες το οποίο επιβαρύνει ακόμα περισσότερο την κατάσταση, εφόσον θα αυξάνεται το απαιτούμενο εύρος ζώνης.

1.1.2 Πρωτόκολλα για μετάδοση δεδομένων σε πραγματικό χρόνο [13], [14]

Τα τελευταία χρόνια υπήρξε μια αλματώδης ανάπτυξη στις τεχνολογίες και στις υποδομές του Internet. Αυτό είχε ως άμεσο αποτέλεσμα τη δημιουργία όλο και περισσοτέρων υπηρεσιών που προσφέρουν στο χρήστη μεταφορά δεδομένων σε πραγματικό χρόνο. Είναι σημαντικό τα δεδομένα ήχου και βίντεο να αναπαράγονται συνεχώς και χωρίς διακοπές. Αν τα δεδομένα δεν φτάσουν στον παραλήπτη τη στιγμή που πρέπει και υποστούν καθυστέρηση που θα ξεπερνά ένα συγκεκριμένο όριο, τότε η αναπαραγωγή θα παγώσει, γεγονός αρκετά ενοχλητικό για τον χρήστη.

Το Integrated Services working group στο IETF (Internet Engineering Task Force), ανέπτυξε ένα εξελιγμένο μοντέλο υπηρεσιών Internet που καλείται Integrated Services και περιλαμβάνει best-effort service και real-time service. Το Resource ReSerVation Protocol (RSVP), μαζί με το Real Time Transport Protocol (RTP), το Real Time Control Protocol (RTCP) και το Real Time Streaming Protocol (RTSP), παρέχουν τη βάση για υπηρεσίες πραγματικού χρόνου. Το κυρίαρχο πρωτόκολλο αυτή την στιγμή για την μεταφορά streaming πολυμέσων είναι το RTP (Realtime Transport Protocol) τα πακέτα του οποίου ενσωματώνονται συνήθως σε UDP datagrams για να φτάσουν στον προορισμό τους και το Real Time Streaming Protocol (RTSP).

Στην συνέχεια της παραγράφου 1.1.2 και για λόγους πληρότητας παραθέτουμε συνοπτική περιγραφή των παραπάνω. Δεδομένου ότι τα πρωτόκολλα αυτά δεν σχετίζονται άμεσα με τους αλγόριθμους που αναπτύσονται και εξετάζονται στην διπλωματική εργασία ο αναγνώστης μπορεί να παραλείψει το υπόλοιπο της παραγράφου 1.1.2.

1.1.2.1 Resource ReSerVation Protocol (RSVP)

Το πρωτόκολλο RSVP λειτουργεί πάνω από το IP και σχεδιάστηκε για να λειτουργεί με unicast και multicast routing πρωτόκολλα. Επειδή το RSVP προορίζεται για μεγάλα multicast περιβάλλοντα, έχει σχεδιαστεί κατά τέτοιο τρόπο, που οι παραλήπτες είναι υπεύθυνοι να κάνουν την κράτηση δικτυακών πόρων (reservation). Οι παραλήπτες προσπαθούν να κάνουν το reservation, ανάλογα με τις παραμέτρους QoS που ζητά ο αποστολέας. Αυτές οι παράμετροι είναι σε μορφή traffic specification και path characteristics. Εφαρμογές που τρέχουν σε IP συστήματα μπορούν να χρησιμοποιήσουν το RSVP για να υποδηλώσουν σε άλλα σημεία την φύση (bandwidth κλπ.) των πακέτων που θέλουν να αποκτήσουν.

Το RSVP αποτελείται από αποστολείς (senders), παραλήπτες (receivers), σταθμούς (hosts) και δρομολογητές (routers) που βρίσκονται μεταξύ των πρώτων δύο. Σε αυτό το πρωτόκολλο, ο αποστολέας του μηνύματος επιτρέπει στον παραλήπτη να γνωρίζει ότι θέλει να μεταδώσει ένα μήνυμα με κάποιες παραμέτρους QoS. Έτσι ο αποστολέας μεταδίδει ένα path message, κατά μήκος του μονοπατιού (route), όπου θα σταλεί το μήνυμα στον παραλήπτη, ο οποίος μπορεί να είναι ένας μοναδικός (unicast) ή και πολλαπλοί (multicast) προορισμοί. Όταν ο παραλήπτης λάβει το μήνυμα, είναι υπεύθυνος για να κάνει την προκράτηση του μονοπατιού στέλνοντας πίσω στον αποστολέα ένα μήνυμα. Αν το μήνυμα που έρχεται πίσω στον αποστολέα είναι θετικό, τότε ξέρει ότι μπορεί να μεταδώσει τα μηνύματα που θέλει στον συγκεκριμένο σταθμό. Ο σκοπός αυτού του πρώτου μηνύματος (path message) είναι διπλός. Πρώτο για να βρεθεί το κατάλληλο μονοπάτι μεταξύ αποστολέα και παραλήπτη και δεύτερο για να συλλέξει ο αποστολέας τις απαραίτητες QoS πληροφορίες τις οποίες μπορεί να προσφέρει ο κάθε router στο μονοπάτι που δημιουργείται.

Το RSVP είναι ένα πρωτόκολλο που καθιερώθηκε από τον IETF για να επιτρέπει σε μία εφαρμογή να δεσμεύει το εύρος ζώνης του δικτύου δυναμικά. Το RSVP δεν είναι πρωτόκολλο δρομολόγησης. Σχεδιάστηκε να λειτουργεί με παρόντα και μελλοντικά unicast και multicast πρωτόκολλα δρομολόγησης.

1.1.2.2 Real Time Transport Protocol (RTP)

Το RTP χρησιμοποιείται εκτενώς σε εφαρμογές που μεταδίδουν δεδομένα σε πραγματικό χρόνο όπως είναι ο ήχος, το βίντεο και η προσομοίωση δεδομένων πάνω σε multicast μετάδοση. Το RTP συνήθως χρησιμοποιείται σε συνδυασμό με το RTCP που θα αναλυθεί παρακάτω. Ενώ το RTP μεταφέρει τις ροές των δεδομένων, το RTCP χρησιμοποιείται για να παρακολουθεί και να καταγράφει στατιστικά στοιχεία για τη μετάδοση και την ποιότητα των υπηρεσιών καθώς και για την αποστολή δεδομένων ελέγχου.

Το RTP δεν καταγράφει τη δέσμευση των πόρων και δεν εγγυάται ποιότητα υπηρεσίας - QoS (Quality of Service) για τις υπηρεσίες πραγματικού χρόνου. Δεν παρέχει καμία εγγύηση για την επιτυχημένη αποστολή των δεδομένων, αλλά με την αρίθμηση των πακέτων που χρησιμοποιεί μας επιτρέπει να γνωρίζουμε ποια πακέτα έχουν χαθεί.

Το TCP δεν χρησιμοποιείται συνήθως από το RTP καθώς εισάγει καθυστέρηση λόγω της εγκαθίδρυσης των συνδέσεων και της διόρθωσης των λαθών. Έτσι οι περισσότερες υλοποιήσεις του RTP «τρέχουν» πάνω από το πολύ πιο απλό UDP. Το UDP επιλέγεται σαν το πρωτόκολλο μεταφοράς του RTP για 2 λόγους:

1. Το RTP είναι κυρίως σχεδιασμένο για multicast κάτι το οποίο δεν συμβαδίζει με το connection-oriented TCP.
2. Για δεδομένα πραγματικού χρόνου, η αξιοπιστία δεν είναι τόσο σημαντική όσο η έγκαιρη μετάδοση. Η αξιόπιστη μετάδοση, η οποία επιτυγχάνεται μέσω της επαναμετάδοσης των χαμένων πακέτων, μπορεί να μην είναι επιθυμητή, αφού μπορεί να προκαλέσει υπερφόρτωση του δικτύου και προβλήματα στη συνεχή μετάδοση των δεδομένων.

Ένας αποστολέας RTP είναι υπεύθυνος για σύλληψη δεδομένων ήχου και βίντεο και για την μετατροπή τους σε frames χρησιμοποιώντας έναν κατάλληλο κωδικοποιητή. Αυτά τα frames στη συνέχεια στέλνονται σαν RTP πακέτα. Ο αποστολέας μπορεί περιστασιακά να εκτελεί ενέργειες διόρθωσης λαθών και ελέγχου της συμφόρησης. Ο αποστολέας μπορεί να πραγματοποιήσει αλλαγές στον ρυθμό της μετάδοσης ανάλογα με τις αναφορές που λαμβάνει από το RTCP. Ο παραλήπτης RTP συλλέγει τα πακέτα από το δίκτυο, χειρίζεται τις όποιες απώλειες πακέτων, πραγματοποιεί τον συγχρονισμό των ροών, τις αποκωδικοποιεί και παρουσιάζει το πολυμεσικό περιεχόμενο στο χρήστη.

Οι πολυμεσικές εφαρμογές χαρακτηρίζονται από αυστηρούς χρονικούς περιορισμούς στη μετάδοση των δεδομένων, κάτι που δεν συμβαδίζει με τη λογική λειτουργίας του Internet. Το RTP παρέχει κάποιους μηχανισμούς που λαμβάνουν υπόψη τα θέματα αυτά. Τέτοιοι μηχανισμοί είναι το timestamp και το sequence numbering. Το timestamp είναι σημαντική πληροφορία για τις εφαρμογές πραγματικού χρόνου. Ο αποστολέας βάζει σε κάθε πακέτο ένα timestamp, το οποίο χρησιμοποιεί ο παραλήπτης για να βρει τη χρονική στιγμή που πρέπει να παρουσιάσει τα δεδομένα στον χρήστη. Δηλαδή, το timestamp παρέχει σήματα χρονισμού ώστε να είναι δυνατό στους παραλήπτες να ανακατασκευάσουν τα αρχικά δεδομένα όπως αυτά μεταδόθηκαν από τον αποστολέα. Το timestamp χρησιμοποιείται επίσης για το συγχρονισμό διαφορετικών streams, όπως βίντεο και ήχου. Το RTP δεν είναι υπεύθυνο για το συγχρονισμό αυτό, τον οποίο πραγματοποιούν οι εφαρμογές. Το UDP, το οποίο συνήθως χρησιμοποιείται για τη μετάδοση των RTP/RTCP πακέτων, δεν παραδίδει τα πακέτα με τη σειρά με την οποία στάλθηκαν γι’ αυτό τα πακέτα αριθμούνται τη στιγμή που στέλνονται, έτσι ώστε να μπορεί ο παραλήπτης να τα

τοποθετήσει στη σωστή σειρά. Οι αριθμοί αυτοί χρησιμοποιούνται επίσης για να ανιχνεύονται απώλειες στη μετάδοση των πακέτων.

1.1.2.3 Real Time Control Protocol (RTCP)

Το RTCP είναι ένα πρωτόκολλο ελέγχου που σχεδιάστηκε για να συνεργάζεται με το RTP. Παρέχει στατιστικά στοιχεία και πληροφορίες ελέγχου για μια ροή δεδομένων RTP. Συνοδεύει το RTP στη μεταφορά αλλά το ίδιο δεν μεταφέρει δεδομένα πολυμέσων. Η κύρια λειτουργία του είναι να παρέχει στον αποστολέα RTP ανατροφοδότηση σχετικά με την ποιότητα των υπηρεσιών (QoS) στέλνοντας στατιστικά στοιχεία που αφορούν τη μεταφορά και τη λήψη των πακέτων RTP. Συγκεντρώνει δεδομένα όπως ο αριθμός των πακέτων που έχουν μεταδοθεί, ο αριθμός των χαμένων πακέτων, το jitter και το round-trip delay.

Το RTCP περιλαμβάνει διάφορους τύπους πακέτων όπως sender report, receiver report, source description και το bye.

- (Sender Report) SR: Το πρώτο RTCP πακέτο στο συνδυασμένο πακέτο πρέπει να είναι ένα πακέτο αναφοράς για να διευκολύνει την επικύρωση της επικεφαλίδας (header) του πακέτου. Αυτό αληθεύει ακόμα και αν δεν στέλνονται ή δεν λαμβάνονται καθόλου δεδομένα ή ακόμα και αν το μοναδικό RTCP πακέτο στο σύνθετο πακέτο είναι ένα BYE πακέτο, οπότε στέλνεται ένα άδειο RR.

- Receiver Report (RR): Τα πακέτα RR στέλνονται από τους παθητικούς συμμετέχοντες, αυτούς δηλαδή που δεν στέλνουν RTP πακέτα. Η αναφορά ενημερώνει τους RTP Senders για την ποιότητα των υπηρεσιών.

- (Source DEScription) SDES: Ένα μήνυμα SDES χρησιμοποιείται για να στείλει τα αναγνωριστικά των συμμετεχόντων στους υπόλοιπους συμμετέχοντες.

- (End of participation) BYE: Μια πηγή στέλνει ένα μήνυμα BYE για να κλείσει μια ροή δεδομένων. Χρησιμοποιείται από έναν συμμετέχοντα για να ανακοινώσει ότι εγκαταλείπει τη συνεδρία.

1.1.2.4 Real Time Streaming Protocol (RTSP)

Το RTSP είναι ένα πρωτόκολλο παρουσίασης πολυμέσων που βασίζεται στο μοντέλο client-server και επιτρέπει την ελεγχόμενη διανομή ροών πολυμεσικών δεδομένων μέσα από IP δίκτυα. Στόχος του RTSP είναι να παρέχει για βίντεο και ήχο τις ίδιες υπηρεσίες που παρέχει για text και γραφικά το HTTP. Στο RTSP, κάθε παρουσίαση ενός media stream αναγνωρίζεται μέσω ενός RTSP URL. Μερικές από τις δυνατότητες που παρέχει φαίνονται πιο κάτω:

- Μπορεί να ανακτήσει πληροφορία από ένα media server: Ο client μπορεί να ζητήσει μια περιγραφή της επιθυμητής πληροφορίας μέσω του HTTP. Αν η μετάδοση είναι multicast η περιγραφή αυτή περιέχει και τις διευθύνσεις και τις θύρες (ports) που θα χρησιμοποιηθούν για τη συνεχή ροή της πληροφορίας. Αν η μετάδοση είναι unicast ο client είναι αυτός που θα παρέχει τη διεύθυνση του για λόγους ασφάλειας.

- Μπορεί να προσκαλέσει ένα media server σε μια τηλεδιάσκεψη: Ο media server μπορεί να προσκληθεί σε μια συνεδρία ώστε να αναπαράγει πολυμεσικό περιεχόμενο.

- Μπορεί να προσθέσει πληροφορία σε μια ήδη υπάρχουσα media παρουσίαση: αυτό το χαρακτηριστικό είναι χρήσιμο στις περιπτώσεις των "ζωντανών" παρουσιάσεων οπότε και ο server θα έχει τη δυνατότητα να γνωστοποιεί στον client τη νέα πληροφορία που είναι διαθέσιμη.

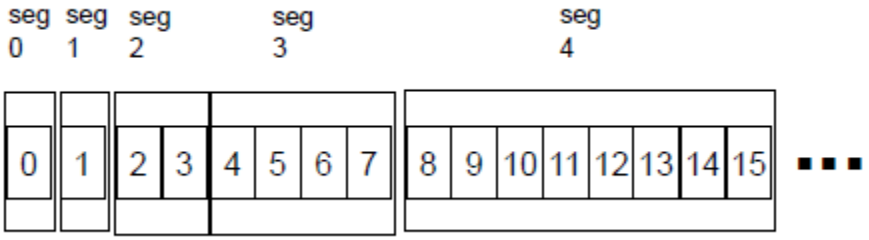
1.2 Προηγούμενη έρευνα για αποθήκευση βίντεο σε διακομιστή μεσολάβησης

Αρκετές εργασίες εστιάζουν το ενδιαφέρον τους στη χρήση των proxy servers για αποθήκευση των πιο δημοφιλών βίντεο, είτε αποθηκεύοντας κάποια τμήματά τους είτε αποθηκεύοντας ολόκληρα τα βίντεο, για βελτίωση της ποιότητας των μεταδόσεων δεδομένων σε πραγματικό χρόνο. Στην συνέχεια παρουσιάζονται σημαντικές τεχνικές από διάφορες μεθόδους αποθήκευσης των βίντεο στους proxy servers.

1.2.1 Segment Caching

Στα [1] και [5] προτάθηκε η τμηματοποίηση βίντεο στην οποία το μέγεθος του κάθε τμήματος θα αυξάνεται εκθετικά όπως φαίνεται στο Σχ.1.1 (Pyramid Segmentation). Η ιδέα αυτή στηρίζεται στην παρατήρηση ότι τα αρχικά τμήματα των βίντεο είναι πιο πιθανόν να

προσπελαστούν από τους χρήστες σε αντίθεση με τα τελευταία τμήματα. Έτσι όταν απαιτείται αφαίρεση κάποιου βίντεο θύματος θα μπορεί να αφαιρείται ένα μεγάλο τελευταίο τμήμα του βίντεο χωρίς αυτό να επηρεάσει την δυνατότητα υποστήριξης μελλοντικών αιτήσεων για το ίδιο βίντεο. Επίσης στις ίδιες εργασίες προτάθηκε και η τμηματοποίηση του βίντεο με βάση ένα σταθερό αριθμό από blocks (Fixed Segmentation).



Σχ.1.1: Παράδειγμα εκθετικής τμηματοποίησης. [1]

Με βάση τις μεθόδους αποθήκευσης και αντικατάστασης δίνονται τα caching values στα διάφορα τμήματα. Το caching value καθορίζει την εκτίμηση δημοτικότητας των τμημάτων κάθε βίντεο και εξαρτάται από δύο μεταβλητές: τη συχνότητα ζήτησης του βίντεο και την απόσταση του τμήματος από την αρχή, και ορίστηκε να είναι ίσο με $\frac{1}{(Tc-Tr)*i}$, όπου Tc ο τρέχον χρόνος, Tr ο χρόνος τελευταίας αναφοράς για το βίντεο και i ο αριθμός τμήματος του βίντεο. Όπως φαίνεται για το ίδιο βίντεο το πρώτο τμήμα έχει μεγαλύτερο caching value σε σχέση με το τελευταίο τμήμα.

Επιπλέον έχει προταθεί η ιδέα να αφιερώνεται ένα μέρος της συνολικής μνήμης για την αποθήκευση του προθέματος κάθε βίντεο (prefix) και το υπόλοιπο μέρος της μνήμης να προορίζεται για τα υπόλοιπα τμήματα των βίντεο (suffix). Το πρόθεμα αντιστοιχεί σε ένα σταθερό αριθμό Kmin από τα αρχικά blocks του βίντεο και υπολογίζεται λαμβάνοντας υπόψη την καθυστέρηση που θα υπάρχει στο σύστημα αν τα blocks αυτά δε βρίσκονται αποθηκευμένα στον proxy server την στιγμή που έρχεται η αίτηση για το συγκεκριμένο βίντεο. Η ιδέα αυτή προσδίδει στο σύστημα λιγότερες καθυστερήσεις κατά το ξεκίνημα των βίντεο.

Η ιδέα είναι να αποθηκεύονται στη μνήμη μόνο τα δημοφιλή βίντεο. Βασικός παράγοντας είναι η απόσταση του τμήματος από τη αρχή του βίντεο (segment number). Κάθε τμήμα με αριθμό τμήματος μικρότερο από το όριο Kmin θα είναι πάντα επιλέξιμο για αποθήκευση. Αν ο αριθμός αυτός είναι μεγαλύτερος από Kmin τότε το τμήμα θα είναι επιλέξιμο μόνο αν το caching value που έχει είναι μεγαλύτερο από το caching value κάποιων άλλων τμημάτων που είναι ήδη αποθηκευμένα στη μνήμη και τα οποία έχουν επίσης αριθμό τμήματος μεγαλύτερο από Kmin.

1.2.2 Hotspot Caching

Στο [16] προτάθηκε η ιδέα να μαρκάρονται σημαντικές θέσεις του βίντεο τα λεγόμενα hotspots τα οποία θα αποθηκεύονται στον proxy server ώστε να μπορούν να αντιπροσωπεύουν σημαντικές λεπτομέρειες του βίντεο. Με αυτό δίνεται η δυνατότητα στους χρήστες να κάνουν προεπισκόπηση στο βίντεο ώστε να αποφασίσουν αν τους ενδιαφέρει τελικά πριν το ζητήσουν ολόκληρο. Τα hotspots πρέπει να αποτελούν τουλάχιστο μια συμπληρωμένη σκηνή του βίντεο, ώστε να έχει νόημα η εικόνα που παρουσιάζεται στους χρήστες. Με τη χρήση των hotspots επιτυγχάνεται μείωση στις καθυστερήσεις που συμβαίνουν από την τυχαία αναπαραγωγή βίντεο όταν οι χρήστες χρησιμοποιούν κινήσεις προς τα εμπρός ή προς τα πίσω. Ακόμα τα hotspots έχουν πολύ μικρότερες απαιτήσεις αποθήκευσης από την αποθήκευση ολόκληρου του βίντεο στον διακομιστή μεσολάβησης, έτσι με τη χρήση τους γίνεται πιο αποδοτική η αξιοποίηση της μνήμης. Τέλος η χρήση των hotspots επιφέρει μείωση στον αριθμό αιτήσεων που σταματάνε τα βίντεο στα αρχικά frames και έτσι πετυχαίνει καλύτερη εξυπηρέτηση των περισσότερων αιτήσεων.

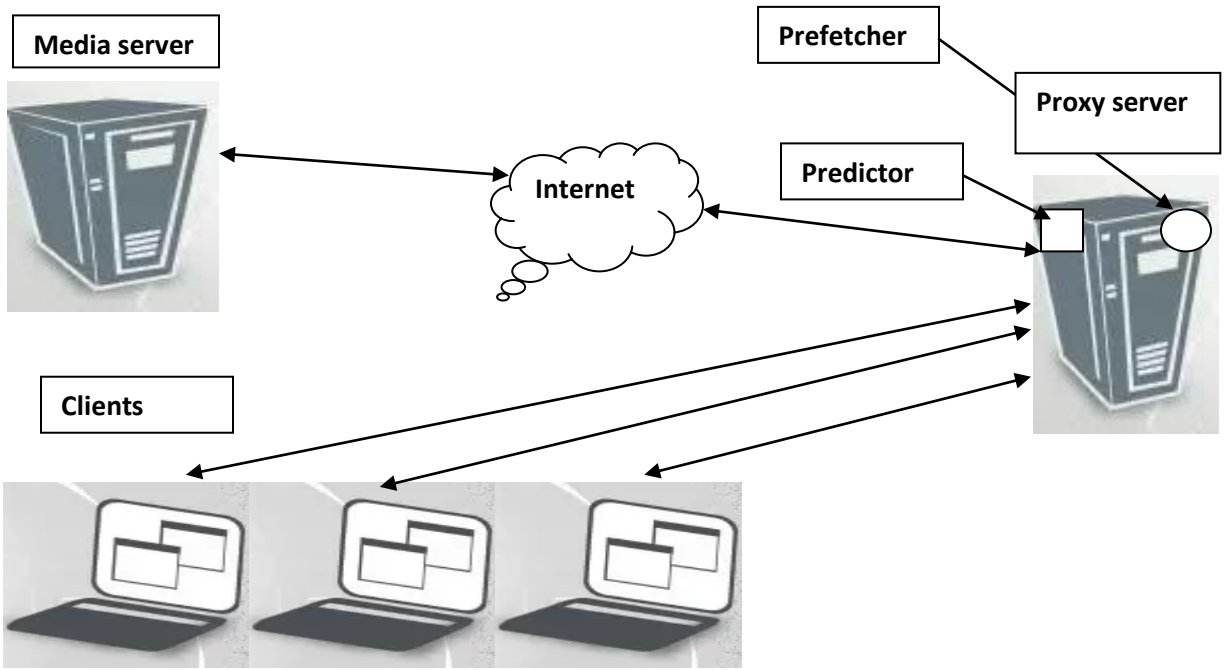
1.2.3 Prefix caching

Στο [17] προτάθηκε η ιδέα να αποθηκεύονται τα αρχικά τμήματα των βίντεο, τα λεγόμενα *prefixes*, στον proxy server έτσι ώστε να μειωθούν οι καθυστερήσεις κατά το ξεκίνημα των βίντεο. Το μέγεθος του *prefix* καθορίζεται με βάση την καθυστέρηση που θα υπάρχει κατά το ξεκίνημα του βίντεο μέχρι να ολοκληρωθεί η μεταφορά του τμήματος αυτού, αν δεν ήταν ήδη αποθηκευμένο. Ορίζετε ως max{dmax-s ,0}, όπου dmax είναι η μέγιστη καθυστέρηση από τον server στον proxy server (σε αριθμό από frames) και s ορίζεται η καθυστέρηση αναπαραγωγής στον πελάτη (σε αριθμό από frames). Όταν έρθει αίτηση για κάποιο βίντεο το σύστημα θα ξεκινήσει τη μεταφορά του *prefix*, αν είναι ήδη αποθηκευμένο στη μνήμη του proxy server και θα ζητήσει να προσκομιστεί το υπόλοιπο βίντεο (*suffix*) από τον server.

1.2.4 Prefetching methods

Έχουν προταθεί διάφορες τεχνικές οι οποίες ως στόχο έχουν την υποστήριξη μετάδοσης δεδομένων σε πραγματικό χρόνο. Αυτές οι τεχνικές προσπαθούν να προβλέψουν το επόμενο τμήμα του βίντεο το οποίο θα ζητηθεί από τον χρήστη, ώστε να έχει γίνει ήδη η μεταφορά από τον media server στον proxy server πριν ακόμα ζητηθεί το συγκεκριμένο τμήμα.

Το μοντέλο το οποίο προτείνεται στο [18] αποτελείται από δύο κύρια μέρη, τον predictor engine και τον prefetching engine. Ο predictor προβλέπει την επόμενη πρόσβαση του χρήστη χρησιμοποιώντας τις πληροφορίες από προηγούμενες προσβάσεις. Επίσης κρατά τις προσβάσεις που γίνονται από τον media server και υιοθετεί τις τρέχουσες αλλαγές στο αρχείο πρόσβασης, οι οποίες θα είναι διαθέσιμες στον proxy server. Ο prefetcher προσπαθεί να μεταφέρει κάποια τμήματα από το βίντεο πριν αυτά ζητηθούν. Η αρχιτεκτονική που προτείνεται φαίνεται στο Σχ.1.2.



Σχ.1.2: Prefetching Architecture. [18]

Ο proxy server φροντίζει την πρόβλεψη και την προσκόμιση των τμημάτων κάθε βίντεο, με τη βοήθεια του prefetcher και του predictor. Κανονικά ο proxy server αποθηκεύει το βίντεο που έχει ήδη μεταδώσει σε ένα πελάτη και θα μεταδοθεί αργότερα αν κάποιος άλλος πελάτης ζητήσει το ίδιο βίντεο. Στην αρχιτεκτονική που προτείνεται, κάποιο τμήμα της συνολικής μνήμης του proxy server προορίζεται για την αποθήκευση των τμημάτων πριν ακόμα ζητηθούν (prefetched media chunks), τα οποία προβλέφθηκαν ότι θα ζητηθούν αργότερα. Ο predictor ενημερώνεται συνέχεια από τον media server και ενημερώνει την πρόβλεψή του για το μέλλον, η οποία μπορεί να χρησιμοποιηθεί από τον proxy server.

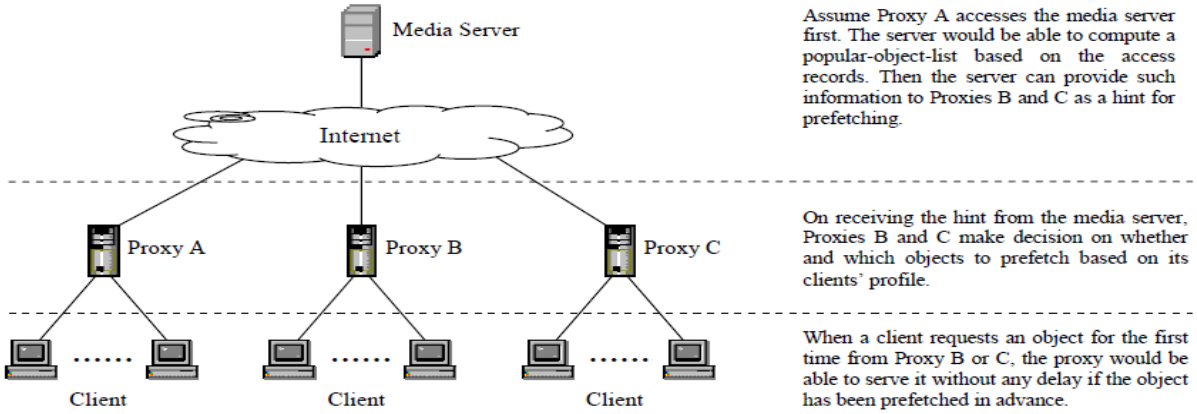
Ο prefetcher παίρνει την ενημέρωση συνεχούς ροής που προβλέφθηκε από τον predictor και ζητά από την μνήμη να επαληθεύσει τα τμήματα τα οποία είναι ήδη αποθηκευμένα. Μετά την ενημέρωση ο prefetcher προσπαθεί να προσκομίσει από πριν τα προβλεπόμενα τμήματα στη μνήμη του proxy server, εφόσον γίνει μεταφορά από τον media server. Ο prefetcher δεν εισάγει ολόκληρο το περιεχόμενο στη μνήμη, αντί αυτού φέρνει το 10% του βίντεο από την αρχή, μέση και τέλος.

Τα τμήματα τα οποία έχουν προσκομιστεί πριν ακόμα ζητηθούν θα μπορούν να εξυπηρετήσουν τον πελάτη όταν κάνει αίτηση για το συγκεκριμένο βίντεο. Κατά τη διάρκεια που προβάλλονται τα αρχικά τμήματα στο πελάτη, ο proxy server είναι έτοιμος να προσκομίσει και τα υπόλοιπα τμήματα τα οποία δεν έχουν αποθηκευτεί στη μνήμη του. Η μορφή ενός βίντεο που προσκομίστηκε μερικώς φαίνεται στο Σχ.1.3. Τα συμπαγή τμήματα έχουν προσκομιστεί και τα μη συμπαγή δεν έχουν ακόμα προσκομιστεί. Επιλέχτηκε να αποθηκεύεται από πριν το 10% του βίντεο γιατί οι περισσότερες αιτήσεις παρακολουθούν το αρχικό τμήμα των βίντεο και μπορεί να μεταβούν στη μέση ή στο τέλος του ανάλογα με το ενδιαφέρον του χρήστη. Έτσι το σύστημα μπορεί να εξυπηρετήσει τυχαίες προσπελάσεις του βίντεο. Ακόμα έχει πετύχει μείωση στο απαιτούμενο εύρος ζώνης γιατί δεν αποθηκεύει απευθείας ολόκληρο το περιεχόμενο του βίντεο αλλά κάποια τμήματά του.

Σχ.1.3: Partial Aggressive Prefetching Pattern. [18]

Στο [19] προτείνεται μια Aggressive prefetching μέθοδος, η οποία προσπαθεί να έχει τα πιο δημοφιλή τμήματα των βίντεο στη μνήμη του proxy server πριν ακόμα ζητηθούν, έτσι ώστε να μπορεί να υποστηρίξει χωρίς καθυστέρηση τη μετάδοση δεδομένων σε πραγματικό χρόνο, σε περίπτωση που έρθει κάποια αίτηση για αυτά. Στο Σχ.1.4 φαίνεται η αρχιτεκτονική που προτείνεται. Ο media server μπορεί να γνωρίζει τα δημοφιλή βίντεο με βάση το αρχείο καταχώρησης που έχει για κάθε βίντεο. Σε κάθε αίτηση του proxy server προς τον media server θα στέλλεται η λίστα με τα πιο δημοφιλή βίντεο στον proxy server. Ακολούθως ο proxy server θα μπορεί να ελέγχει το περιεχόμενο της λίστας και τα προφίλ των πελατών και να αποφασίζει αν πρέπει να αποθηκεύσει στη μνήμη του κάποια τμήματα των βίντεο πριν αυτά ζητηθούν. Κάθε βίντεο στο προτεινόμενο σύστημα, μπορεί να προσκομίζει μόνο τα αρχικά τμήμα του, γιατί λόγω του μεγάλου μεγέθους που έχουν τα βίντεο θα εξαντλούσαν τη μνήμη του proxy server αν

εισάγονταν ολόκληρα. Η λειτουργία prefetching πρέπει να έχει χαμηλότερη προτεραιότητα από την κανονική αίτηση και πρέπει να εκτελείται όταν το φόρτος εργασίας δεν είναι μεγάλο.



Σχ.1.4: The operation of server-assisted prefetching. [19]

1.3 Οργάνωση της εργασίας

Στο Κεφάλαιο 2 παρουσιάζετε το πρώτο μέρος της εργασίας, το οποίο βασίζεται στην υπόθεση ότι ο χρήστης θα παρακολουθήσει σειριακά ολόκληρο το βίντεο από την αρχή μέχρι το τέλος. Αρχικά στο 2.1 εισάγουμε τις βασικές ιδέες για αποθήκευση στον διακομιστή μεσολάβησης και τον τρόπο που θα χειριστούμε τη μνήμη του. Στο 2.2 παρουσιάζουμε το μοντέλο του συστήματος και τις μεθόδους για αποθήκευση στον διακομιστή μεσολάβησης. Στο 2.3 γίνεται παρουσίαση των μετρικών απόδοσης που θεωρούνται σημαντικές για το σύστημα. Ακολούθως περιγράφεται το μοντέλο προσομοίωσης και δίνονται τα αποτελέσματα για διάφορες αλλαγές στις παραμέτρους του συστήματος. Τέλος στην ενότητα 2.4 δίνονται τα συμπεράσματα που έχουν προκύψει από τα αποτελέσματα των προσομοιώσεων του Κεφαλαίου.

Στο Κεφάλαιο 3 παρουσιάζετε το δεύτερο μέρος της δουλειάς, το οποίο βασίζεται στην υπόθεση ότι ο χρήστης μπορεί να παρακολουθήσει το βίντεο είτε ολόκληρο είτε μερικώς. Στην ενότητα 3.1 γίνεται εισαγωγή στη μερική παρακολούθηση των βίντεο. Στην ενότητα 3.2 παρουσιάζουμε τους προτεινόμενους αλγορίθμους και το τρόπο διαχείρισης της μνήμης από κάθε αλγόριθμο με σκοπό να μπορεί το σύστημα να υποστηρίξει μερική παρακολούθηση των βίντεο. Στην ενότητα 3.3 παρουσιάζονται οι μετρικές αποδόσεις, ακολούθως περιγράφεται το μοντέλο

προσομοίωσης και τα διάφορα σενάρια που υλοποιήθηκαν και στο τέλος δίνονται τα αποτελέσματα των προσομοιώσεων του Κεφαλαίου. Στην ενότητα 3.4 σχολιάζεται η πολυπλοκότητα των αλγορίθμων που παρουσιάστηκαν στο Κεφάλαιο αυτό ως προς την υλοποίηση και την ποσότητα πληροφορίας που χρειάζεται να διατηρούν. Τέλος στην ενότητα 3.5 δίνονται τα συμπεράσματα που έχουν προκύψει από τα αποτελέσματα των προσομοιώσεων του Κεφαλαίου.

Το Κεφάλαιο 4 παρουσιάζει τον επίλογο της διπλωματικής εργασίας. Στην ενότητα 4.1 παρουσιάζεται η συνεισφορά της διπλωματικής εργασίας και στην ενότητα 4.2 δίνονται μερικές ιδέες για μελλοντικές επεκτάσεις.

Κεφάλαιο 2: Μέθοδοι αποθήκευσης και αντικατάστασης βίντεο σε διακομιστή μεσολάβησης και υπολογισμός απόδοσης σε πλήρη παρακολούθηση των βίντεο

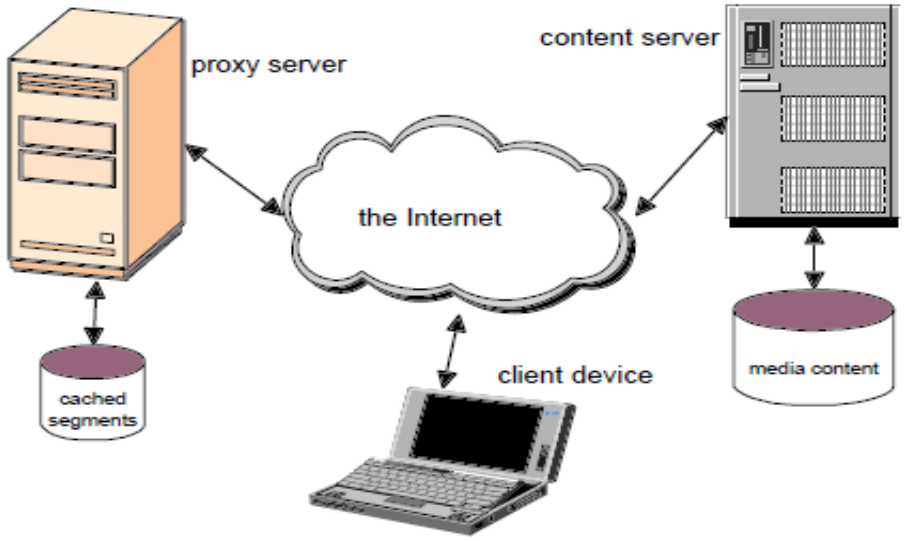
2.1 Εισαγωγή

Στο Κεφάλαιο αυτό εισάγουμε διάφορες αλγοριθμικές ιδέες για αποθήκευση των βίντεο στον διακομιστή μεσολάβησης, έτσι ώστε να έχουμε αποδοτική αξιοποίηση της μνήμης. Στην μελέτη μας έχει γίνει η υπόθεση ότι κάθε χρήστης παρακολουθεί το βίντεο το οποίο ζήτησε από την αρχή μέχρι το τέλος σειριακά, χωρίς παύσεις ή άλλες κινήσεις από τη μεριά του χρήστη. Στη μελέτη μας η μνήμη του διακομιστή μεσολάβησης χωρίζεται σε δύο τμήματα. Το πρώτο τμήμα ορίζεται ως μνήμη A και μπορεί να αποθηκεύει ένα μικρό τμήμα του βίντεο το οποίο θα αντιστοιχεί στο πρόθεμά του. Το δεύτερο τμήμα ορίζεται ως μνήμη B και μπορεί να αποθηκεύει ένα μεγάλο ενιαίο τμήμα του βίντεο που θα αντιστοιχεί σε ολόκληρο το μέγεθός του, εκτός από το πρόθεμα το οποίο διαχειρίζεται μόνο από τη μνήμη A. Μια δημοφιλής τεχνική είναι να αποδίδεται σε κάθε βίντεο μια μεταβλητή η οποία θα καθορίζει την εκτίμηση δημοτικότητας του κάθε βίντεο. Στη μελέτη μας η συγκεκριμένη μεταβλητή θα αναφέρεται ως caching value και θα είναι μοναδική για κάθε βίντεο. Για τον υπολογισμό του caching value λαμβάνονται υπόψη σημαντικές παράμεροι όπως είναι ο αριθμός των συνολικών αιτήσεων για το βίντεο, ο αριθμός των μπλοκ που έχει ήδη αποθηκεύσει το βίντεο στη μνήμη του διακομιστή μεσολάβησης, καθώς επίσης και ο χρόνος που έχει παρέλθει μεταξύ της τελευταίας προσπέλασης μέχρι την τρέχουσα προσπέλαση για το βίντεο. Με βάση τα αποτελέσματα των προσομοιώσεων έχουμε πετύχει σημαντική βελτίωση σε ότι αφορά τις παρακάτω μετρικές απόδοσης: (1) Byte-Hit Ratio, (2) καθυστερήσεις κατά το ξεκίνημα των βίντεο και (3) overhead του συστήματος.

2.2 Περιγραφή Συστήματος

2.2.1 Τοπολογία Συστήματος

Η αρχιτεκτονική του συστήματος για μετάδοση δεδομένων σε πραγματικό χρόνο φαίνεται στο Σχ.2.1. Ο διακομιστής μεσολάβησης (proxy server) τοποθετείται κοντά στους πελάτες. Όταν το αρχείο, στη δική μας περίπτωση το βίντεο, βρίσκεται ήδη αποθηκευμένο στη μνήμη του proxy server, ολόκληρο ή μερικώς, τότε το σύστημα είναι έτοιμο να αρχίσει τη μετάδοση. Η καθυστέρηση που εισάγεται στη μεταφορά του περιεχομένου του βίντεο από τον proxy server στον πελάτη στην περίπτωση στην οποία το βίντεο έχει ολόκληρο το περιεχόμενό του αποθηκευμένο είναι αμελητέα. Αν όμως το βίντεο που ζήτησε ο χρήστης δε βρίσκεται στη μνήμη του proxy server (ολόκληρο ή μερικώς), τότε πρέπει να γίνει μεταφορά του περιεχομένου που δε βρέθηκε, από τον content server στον proxy server. Η μεταφορά σε αυτή την περίπτωση εισάγει σημαντικές καθυστερήσεις στο σύστημα με σοβαρές επιπτώσεις στους χρήστες και στην κίνηση του δικτύου. Όταν οι καθυστερήσεις αυτές επαναλαμβάνονται κατά τη διάρκεια στην οποία ο χρήστης παρακολουθεί κάποιο βίντεο, τότε το αποτέλεσμα θα είναι να μην υπάρχει συνεχής μετάδοση πληροφορίας και ο χρήστης να παρακολουθεί το βίντεο με διακοπές.



Σχ.2.1: Αρχιτεκτονική συστήματος για μετάδοση δεδομένων σε πραγματικό χρόνο. [1]

2.2.2 Μέθοδοι αποθήκευσης και αντικατάστασης

Στις επόμενες ενότητες εισάγουμε τις στρατηγικές εισαγωγής και αντικατάστασης που προτείνουμε για τη διαχείριση της μνήμης του proxy server ώστε το σύστημα να είναι σε θέση να εξυπηρετήσει πλήρη παρακολούθηση των βίντεο (complete video view). Η πρώτη μέθοδος που προτείνεται θα αναφέρεται στη συνέχεια ως "*full video with prefix*" και η δεύτερη ως "*full video with prefix and with limitation*". Ακολούθως προσαρμόζουμε στο μοντέλο μας τις στρατηγικές εισαγωγής και αντικατάστασης οι οποίες προτάθηκαν στο [2] ώστε να συγκρίνουμε τις διαφορετικές στρατηγικές. Η τελευταία μέθοδος ονομάζεται "*full video with prefix in LRLFU*".

Η μέθοδος διαχείρισης μνήμης "*full video with prefix*" στηρίζεται σε ιδέες από τα [1], [2], [3] και [5]. Το «επιθετικό» μέρος του αλγορίθμου βασίζεται σε ιδέες από το [3] και ο τρόπος διαχωρισμού της μνήμης του proxy sever στηρίζεται σε ιδέες από τα [1], [2] και [5]. Η μέθοδος "*full video with prefix and with limitation*"είναι μια παραλλαγμένη μορφή της μεθόδου "*full video with prefix*", ώστε να περιορίσουμε κάποιες λανθασμένες αποφάσεις στις οποίες οδηγείται η τελευταία εξαιτίας της «επιθετικής» στρατηγικής που εφαρμόζει.

2.2.3 Στρατηγικές διαχείρισης της μνήμης

Για όλες τις μεθόδους που παρουσιάζονται στο συγκεκριμένο Κεφάλαιο η μνήμη του proxy server θα χωριστεί σε δύο τμήματα, στηριζόμενοι στην ιδέα που προτάθηκε στα [1], [2] και [5]. Μέσα από τις συγκεκριμένες έρευνες αποδείχθηκε ότι ο συγκεκριμένος τρόπος διαχείρισης της μνήμης είναι αποδοτικός, γιατί μπορεί να βελτιώσει τις καθυστερήσεις που υπόκειται ένα χρήστης κατά το ξεκίνημα του βίντεο. Το πρώτο μέρος της μνήμης του proxy server ονομάζεται μνήμη Α και το δεύτερο μέρος μνήμη Β. Στη μελέτη μας η μνήμη Α αντιστοιχεί στο 10% της συνολικής αρχικής μνήμης και η μνήμη Β αντιστοιχεί στο εναπομείναν 90%. Ορίζουμε ότι στη μνήμη Α θα αποθηκεύονται τμήματα από σταθερό αριθμό από blocks τα οποία αντιστοιχούν στο πρόθεμα (prefix) των βίντεο. Το μέγεθος του prefix καθορίζεται με βάση την καθυστέρηση που θα υπάρχει στην αρχή του βίντεο αν το συγκεκριμένο τμήμα δεν βρίσκεται ήδη στη μνήμη του proxy server όταν έρθει αίτηση για αυτό και χρειάζεται να γίνει η μεταφορά του από τον content server στον proxy server. Το πρόθεμα ορίζεται ίσο με Bp blocks και αντιστοιχεί στα αρχικά Bp blocks του βίντεο. Ακολούθως ορίζουμε ότι στη μνήμη Β θα αποθηκεύεται το υπόλοιπο μέρος του βίντεο, δηλαδή το συνολικό μέγεθός του εκτός από το πρόθεμα. Αυτό το τμήμα του βίντεο ορίζεται ως κατάληξη (suffix) και θεωρείται ότι αποτελεί ένα μεγάλο και ενιαίο τμήμα του βίντεο. Στην επόμενη ενότητα περιγράφονται οι διαφορετικές στρατηγικές εισαγωγής και αντικατάστασης για τη μνήμη Α και τη μνήμη Β. Χρησιμοποιούνται διαφορετικές στρατηγικές για την κάθε μνήμη γιατί

θέλουμε να χειριζόμαστε διαφορετικά το αρχικό τμήμα (prefix) που αποθηκεύεται στη μνήμη Α από το υπόλοιπο τμήμα (suffix) το οποίο αποθηκεύεται στη μνήμη Β.

Όπως αναφέραμε στην εισαγωγή χρησιμοποιείται μια μεταβλητή η οποία αποκαλείται caching value, ώστε να καθορίζεται η εκτίμηση δημοτικότητας κάθε βίντεο. Στη μελέτη μας το caching value ορίζεται ίσο με $\frac{n*ns}{Tc-Tr}$, όπου n ο αριθμός αιτήσεων του βίντεο, ns ο αριθμός των blocks που έχει ήδη αποθηκευμένα (στη μνήμη Α και Β), Tc ο τρέχον χρόνος και Tr ο χρόνος τελευταίας αναφοράς για το βίντεο αυτό. Η συγκεκριμένη μεταβλητή είναι μοναδική για κάθε βίντεο και ανανεώνεται σε κάθε καινούργια αίτηση η οποία αναφέρεται σε αυτό. Το caching value κάθε βίντεο θα χρησιμοποιείται από όλες τις προτεινόμενες μεθόδους του Κεφαλαίου κατά την εφαρμογή της στρατηγικής αντικατάστασης της μνήμης Α ή της μνήμης Β.

Ο τύπος για το συγκεκριμένο caching value έχει επιλεγεί μετά από διάφορες δοκιμές. Θεωρούμε ότι όσο πιο συχνά ζητείται κάποιο βίντεο, δηλαδή δεν μεσολαβεί μεγάλο χρονικό διάστημα μεταξύ του τρέχοντος χρόνου και του χρόνου προηγούμενης αίτησης για το συγκεκριμένο βίντεο (Tc-Tr), τόσο πιο μεγάλη θα πρέπει να είναι η τιμή του caching value. Επίσης όσες περισσότερες αιτήσεις έρχονται για κάποιο βίντεο και όσο πιο μεγάλο είναι το αποθηκευμένο περιεχόμενό του (n*ns) τόσο θα πρέπει να αυξάνεται η τιμή του caching value. Στη μελέτη μας έχουμε επίσης δοκιμάσει το caching value όπως ορίζεται στο [2], το οποίο ορίστηκε ίσο με $\frac{n}{Tc-Tr}$, χωρίς να πάρουμε καλύτερα αποτελέσματα.

2.2.3.1 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη Α

Η στρατηγική αυτή ισχύει και για τις δύο μεθόδους διαχείρισης μνήμης που προτείνουμε στο παρόν Κεφάλαιο, οι οποίες όπως ήδη αναφέρθηκε ονομάστηκαν "*full video with prefix*" και "*full video with prefix and with limitation*". Όταν έρθει αίτηση για κάποιο βίντεο και το prefix δεν είναι ήδη αποθηκευμένο στη μνήμη Α τότε ξεκινά η στρατηγική εισαγωγής και αντικατάστασης. Αν στη μνήμη Α υπάρχει επαρκής χώρος, τότε το prefix αποθηκεύεται χωρίς να απαιτείται κάποιος έλεγχος. Αν όμως η μνήμη Α δεν έχει τον απαραίτητο διαθέσιμο χώρο, τότε ξεκινά η στρατηγική αντικατάστασης για να αποφασίσει το σύστημα αν μπορεί να υλοποιηθεί η αποθήκευση του προθέματος του καινούργιου βίντεο ή όχι. Στην εφαρμογή της στρατηγικής αντικατάστασης, το σύστημα υπολογίζει τα caching values όλων των βίντεο που είναι αποθηκευμένα στη μνήμη Α και τα οποία δεν παρακολουθούνται (not currently played) από κάποιο χρήστη, ώστε να βρεθεί το υποψήφιο βίντεο θύμα με το μικρότερο caching value. Ακολούθως, εξετάζεται αν το caching value του υποψήφιου βίντεο θύματος είναι μικρότερο από το caching value του βίντεο που έκανε την αίτηση. Αν ισχύει τότε αφαιρείται το πρόθεμα του βίντεο θύματος και αποθηκεύεται το πρόθεμα του καινούργιου βίντεο. Κάθε φορά που αφαιρείται το prefix, από το βίντεο θύμα με το μικρότερο

caching value, θα εξετάζεται αν υπάρχει το suffix τμήμα αποθηκευμένο στη μνήμη Β. Αν δεν υπάρχει το suffix τμήμα αποθηκευμένο στη μνήμη Β, τότε σημαίνει πως το βίντεο θύμα έχει αφαιρεθεί εξ' ολοκλήρου από τη μνήμη και θα διαγράφεται όλη η πληροφορία για το συγκεκριμένο βίντεο. Αν όμως υπάρχει το suffix τμήμα στη μνήμη Β, τότε το caching value εξακολουθεί να διατηρείται και να ανανεώνεται σε κάθε αίτηση.

2.2.3.2 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη Β

Η μνήμη Β όπως αναφέρθηκε χειρίζεται το suffix τμήμα κάθε βίντεο. Το μέγεθος του suffix αντιστοιχεί στο συνολικό μέγεθος του συγκεκριμένου βίντεο εκτός από το πρόθεμά του και θεωρείται ως ένα μεγάλο ενιαίο τμήμα του βίντεο. Στις επόμενες δύο ενότητες περιγράφονται οι στρατηγικές της μνήμης Β για τις μεθόδους "*full video with prefix*" και "*full video with prefix and with limitation*".

2.2.3.2.1 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη Β για τη μεθοδολογία "full video with prefix"

Υιοθετούμε την ιδέα από το [3] και θεωρούμε ότι κάθε βίντεο που κάνει αίτηση είναι πάντα υποψήφιο για να αποθηκεύσει το suffix τμήμα του στη μνήμη Β, αν δεν ήταν ήδη αποθηκευμένο. Αυτό θα ισχύει πάντα ακόμα και αν το prefix τμήμα του δεν ήταν ήδη αποθηκευμένο στη μνήμη Α όταν ήρθε η αίτηση του βίντεο. Όταν η μνήμη Β διαθέτει τον απαραίτητο χώρο, τότε αποθηκεύεται το suffix του βίντεο που έκανε την αίτηση. Αν όμως δεν υπάρχει επαρκής χώρος μνήμης, εφόσον ο proxy server γνωρίζει το μέγεθος του suffix του συγκεκριμένου βίντεο, αποδεσμεύεται ο απαραίτητος χώρος ώστε να μπορεί να υλοποιηθεί η αποθήκευση του suffix τμήματος. Για την αποδέσμευση του απαραίτητου χώρου, αρχικά το σύστημα υπολογίζει τα caching values όλων των βίντεο στη μνήμη Β τα οποία δεν παρακολουθούνται από κάποιο χρήστη και στη συνέχεια επιλέγεται ως βίντεο θύμα το βίντεο με το μικρότερο caching value. Ακολούθως, αφαιρείται το suffix τμήμα από το βίντεο θύμα (χωρίς να γίνει σύγκριση αν το βίντεο που έκανε την αίτηση είναι πιο δημοφιλές από το βίντεο θύμα) και η διαδικασία επαναλαμβάνεται μέχρι να αποδεσμευτεί αρκετός χώρος για να αποθηκευτεί το suffix τμήμα του βίντεο που έκανε την αίτηση. Κάθε φορά που θα αφαιρείται το suffix τμήμα από το βίντεο θύμα θα εξετάζεται αν υπάρχει το prefix τμήμα του αποθηκευμένο στη μνήμη Α. Αν δεν υπάρχει το πρόθεμα στη μνήμη Α τότε το βίντεο έχει αφαιρεθεί εξ' ολοκλήρου από τη μνήμη του proxy server και θα διαγράφεται όλη η πληροφορία που είχαμε για το συγκεκριμένο βίντεο.

2.2.3.2.2 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη Β για τη μεθοδολογία "full video with prefix and with limitation"

Η στρατηγική ονομάστηκε "*full video with prefix and with limitation*" γιατί χρησιμοποιεί παρόμοιες στρατηγικές με την πιο πάνω μεθοδολογία εισάγοντας μόνο ένα μικρό περιορισμό. Θέλουμε να περιορίσουμε κάποιες λανθασμένες αποφάσεις στις οποίες οδηγείται η μέθοδος "*full video with prefix*", εξαιτίας της «επιθετικής» στρατηγικής στη μνήμη Β. Η μέθοδος "*full video with prefix and with limitation*" στοχεύει στην αποθήκευση των δημοφιλών βίντεο στη μνήμη Β και στην αποφυγή των περιττών αντικαταστάσεων.

Συνεχίζουμε να θεωρούμε ότι το βίντεο που έκανε την αίτηση πρέπει να αποθηκευτεί στη μνήμη Β ακόμη και αν το prefix του δε βρέθηκε στη μνήμη Α. Στην περίπτωση στην οποία η μνήμη Β έχει επαρκή χώρο, αποθηκεύεται το suffix μέρος του βίντεο χωρίς να απαιτείται κάποιος έλεγχος. Στη περίπτωση που δεν υπάρχει επαρκής χώρος στη μνήμη Β, το σύστημα δεν θα αφαιρεί πάντα το βίντεο θύμα για να αποδεσμευτεί ο απαραίτητος χώρος για την αποθήκευση του βίντεο που έκανε την αίτηση. Αντιθέτως με την πιο πάνω μέθοδο θα υπάρχει περιορισμός για την αφαίρεση κάποιου βίντεο θύματος. Αρχικά θα ελέγχονται όλα τα βίντεο στη μνήμη Β τα οποία δεν είναι currently played και θα επιλέγεται ως υποψήφιο θύμα το βίντεο με το μικρότερο caching value. Στη συνέχεια θα εξετάζεται **αν το υποψήφιο βίντεο θύμα έχει χρόνο τελευταίας αναφοράς (Tr) μικρότερο από το χρόνο τελευταίας αναφοράς του βίντεο που έκανε την αίτηση.** Αν ισχύει ο περιορισμός αυτός, τότε θα αφαιρείται το suffix μέρος του βίντεο θύματος και η διαδικασία θα επαναλαμβάνεται εάν δεν αποδεσμεύτηκε ήδη ο απαραίτητος χώρος. Αν μετά από τις απαραίτητες επαναλήψεις αποδεσμευτεί αρκετός διαθέσιμος χώρος τότε αποθηκεύεται το suffix μέρος του βίντεο που έκανε την αίτηση. Αν όμως το σύστημα δεν καταφέρει να αποδεσμεύσει τον απαραίτητο χώρο, δεν μπορεί να αποθηκευτεί το suffix τμήμα του βίντεο που έκανε την αίτηση. Κάθε φορά που θα αφαιρείται το suffix τμήμα από το βίντεο θύμα θα εξετάζεται αν υπάρχει το prefix αποθηκευμένο στη μνήμη Α. Αν δεν είναι αποθηκευμένο το πρόθεμα στη μνήμη Α, τότε θα μηδενίζονται όλες οι παράμετροι για το βίντεο αυτό και θα διαγράφεται όλη η πληροφορία που είχαμε.

Σε αντίθεση με το [2], όταν ψάχνουμε για βίντεο θύμα στη μνήμη Β, μας ενδιαφέρει να βρούμε το βίντεο με το μικρότερο caching value το οποίο δεν παρακολουθείται από κάποιο χρήστη, χωρίς να μας ενδιαφέρει αν το βίντεο θύμα που θα επιλέξουμε έχει μικρότερο caching value σε σχέση με το βίντεο που έκανε την αίτηση. Η δημοτικότητα των βίντεο ακολουθεί φθίνουσα κατανομή όπως θα εξηγηθεί αργότερα και για αυτό θεωρούμε πιο αποδοτικό να αποφεύγεται ο συγκεκριμένος έλεγχος που χρησιμοποιείται στο [2]. Όταν το βίντεο που αφαιρέσαμε με την πάροδο του χρόνου γίνει ξανά δημοφιλές και έρχονται αιτήσεις για αυτό, του δίνεται η ευκαιρία να αποθηκευτεί ξανά στη μνήμη (Α και Β) και να αποκτήσει ένα υψηλό caching value ώστε να παραμείνει σε αυτή.

Αξίζει να τονίσουμε στο σημείο αυτό ότι δεν γίνεται κάποια τμηματοποίηση του βίντεο. Η μνήμη Α διαχειρίζεται το πρόθεμα σαν ένα τμήμα σταθερού μεγέθους και ίσο με Bp blocks το οποίο αποθηκεύεται στη μνήμη Α κάθε φορά που έρχεται καινούργια αίτηση (αν δεν ήταν ήδη αποθηκευμένο). Αν απαιτείται διαθέσιμη μνήμη τότε θα αφαιρείται από τη μνήμη Α τμήμα ίσο με Bp blocks από κάποιο άλλο βίντεο. Επίσης η μνήμη Β διαχειρίζεται το suffix ως ένα μεγάλο τμήμα ίσο με το μέγεθος του συγκεκριμένου βίντεο αφαιρώντας το πρόθεμά του. Αυτό το τμήμα θα αποθηκεύεται και θα αφαιρείται από τη μνήμη Β θεωρώντας ότι αποτελεί ένα μεγάλο ενιαίο τμήμα του βίντεο.

2.2.3.3 Στρατηγική εισαγωγής και αντικατάστασης στηριζόμενη σε ιδέες από το [2]

Για να μπορούμε στη συνέχεια να συγκρίνουμε τις προτεινόμενες μεθόδους με αυτή που προτείνεται στο [2] προσαρμόζουμε στο μοντέλο μας τις στρατηγικές εισαγωγής και αντικατάστασης που προτάθηκαν στο [2]. Δηλαδή η μνήμη Α εξακολουθεί να αποθηκεύει το πρόθεμα του βίντεο και η μνήμη Β να αποθηκεύει το υπόλοιπο περιεχόμενο του βίντεο θεωρώντας ότι αποτελεί ενιαίο τμήμα του βίντεο, με βάση τις προτεινόμενες στρατηγικές του [2]. Η μέθοδος που παρουσιάζεται πιο κάτω με τη χρήση των στρατηγικών εισαγωγής και αντικατάστασης του [2] ονομάζεται "*full video with prefix in LRLFU*". Ακολουθεί μια σύντομη αναφορά για τις στρατηγικές που προτάθηκαν στο [2] οι οποίες εφαρμόζονται στη μέθοδο "*full video with prefix in LRLFU*".

Στη συγκεκριμένη παράγραφο περιγράφεται η στρατηγική εισαγωγής και αντικατάστασης στη μνήμη Α. Σε κάθε καινούργια αίτηση το βίντεο είναι πάντα υποψήφιο για να αποθηκεύσει το πρόθεμά του στη μνήμη Α, αν δεν ήταν ήδη αποθηκευμένο. Αν η μνήμη Α έχει τον απαραίτητο χώρο η αποθήκευση υλοποιείται χωρίς έλεγχο. Σε αντίθετη περίπτωση, το σύστημα θα υπολογίζει τα caching values όλων των βίντεο που βρίσκονται στη μνήμη Α και τα οποία δεν παρακολουθούνται από κάποιο χρήστη. Στη συνέχεια το βίντεο με το μικρότερο caching value επιλέγεται ως θύμα και αφαιρείται το πρόθεμά του από τη μνήμη Α για να αποθηκευτεί το πρόθεμα του βίντεο που έκανε την αίτηση. Ως βίντεο θύμα δηλαδή επιλέγεται το βίντεο με το μικρότερο caching value χωρίς να γίνεται σύγκριση αν είναι λιγότερο δημοφιλές από το βίντεο που έκανε την αίτηση, όπως γίνεται στις μεθόδους *"full video with prefix"* και *"full video with prefix and with limitation"*. Επιπλέον, κάθε φορά που η μέθοδος *"full video with prefix in LRLFU"* αφαιρεί το πρόθεμα από το βίντεο θύμα, ελέγχει αν το συγκεκριμένο βίντεο θύμα έχει και το suffix τμήμα του αποθηκευμένο στη μνήμη Β. Στην περίπτωση στην οποία το suffix τμήμα είναι αποθηκευμένο στη μνήμη Β τότε αφαιρείται και διαγράφεται όλη η διαθέσιμη πληροφορία για το συγκεκριμένο βίντεο.

Επιπρόσθετα η μνήμη Β δεν εισάγει το suffix τμήμα του βίντεο κάθε φορά που έρχεται αίτηση και δεν είναι ήδη αποθηκευμένο σε αυτήν. Όταν έρθει αίτηση για κάποιο βίντεο και έχει ήδη αποθηκευμένο το prefix τμήμα του, τότε θα εξετάζει αν πρέπει να αποθηκεύσει και το suffix στη μνήμη Β. Αν η μνήμη Β έχει διαθέσιμο χώρο, τότε η αποθήκευση γίνεται χωρίς κανένα έλεγχο. Αν όμως δεν υπάρχει αρκετός διαθέσιμος χώρος η απόφαση για αποθήκευση ή όχι του suffix του νέου βίντεο στη μνήμη Β, θα λαμβάνεται εξετάζοντας κάποιους περιορισμούς. Αρχικά το σύστημα υπολογίζει τα caching values από όλα τα βίντεο που βρίσκονται στη μνήμη Β και τα οποία δεν παρακολουθούνται από κάποιο χρήστη και θα επιλέγεται ως υποψήφιο θύμα το βίντεο με το μικρότερο caching value. Στη συνέχεια για να μπορεί να αφαιρεθεί το suffix του υποψήφιου βίντεο θύματος πρέπει η τιμή του caching value που έχει να είναι μικρότερη από την αντίστοιχη τιμή caching value του βίντεο που έκανε την αίτηση. Αν μπορούν να βρεθούν βίντεο θύματα με μικρότερο caching value από το caching value του βίντεο που έκανε την αίτηση και επιπλέον αν ο χώρος που θα είναι διαθέσιμος μετά την αφαίρεσή τους είναι αρκετός για την αποθήκευση του suffix τμήματος του καινούργιου βίντεο, τότε επιτρέπεται η αφαίρεση των απαραίτητων suffixes από τα διάφορα βίντεο θύματα και αποθηκεύεται το suffix μέρος του βίντεο που έκανε την αίτηση.

Η στρατηγική ορίζεται ως Least Recently Least Frequently Used (LRLFU) γιατί η μνήμη Α αποφασίζει ποια βίντεο είναι τα δημοφιλή και η μνήμη Β αποφασίζει ποία βίντεο μπορούν να αποθηκεύσουν ολόκληρο το περιεχόμενό τους. Το caching value υποτίθεται όπως και στη δική μας περίπτωση ίσο με $\frac{n*ns}{Tc-Tr}$.

2.3 Αξιολόγηση Απόδοσης

2.3.1 Μετρικές απόδοσης

Για να υπάρχει αποδοτική αξιοποίηση της μνήμης του proxy server πρέπει το σύστημα να αποθηκεύει τα πιο περιζήτητα βίντεο στη μνήμη και να μπορεί να προσαρμόζεται εύκολα και γρήγορα σε τυχόν αλλαγές στη δημοτικότητα των βίντεο με απώτερο σκοπό να μειωθεί ο όγκος των δεδομένων που μεταφέρονται από τον content server στον proxy server. Μια σημαντική μετρική απόδοσης για το σύστημά μας είναι το Byte-Hit Ratio **(BHR)** το οποίο ορίζεται ως ο συνολικός αριθμός από blocks που βρίσκονται στον proxy server όταν έρθει μια αίτηση για κάποιο βίντεο, σε σχέση με το συνολικό αριθμό από blocks που ζητήθηκαν (δηλαδή το συνολικό μέγεθος του βίντεο, γιατί στην περίπτωσή μας έχουμε υποθέσει ότι ο χρήστης παρακολουθεί ολόκληρο το βίντεο). Η μετρική αυτή μπορεί να δώσει μια συνολική εικόνα για την κίνηση του δικτύου, γιατί όσο πιο μεγάλη είναι η τιμή της τόσο λιγότερα blocks θα απαιτούνται να μεταφερθούν από τον

content server στον proxy server. Επιπλέον όσο μεγαλύτερο είναι το BHR τόσο λιγότερο εύρος ζώνης απαιτείται, γιατί θα χρειάζεται μεταφορά λιγότερων blocks από τον content server στον proxy server.

Μια δεύτερη μετρική απόδοσης που εξετάζουμε είναι οι καθυστερήσεις στις οποίες υπόκειται ένας χρήστης στην αρχή των βίντεο **(delayed starts)**. Η μετρική αυτή ορίζεται ως το ποσοστό των συνολικών αιτήσεων που έχουν υποστεί καθυστέρηση κατά το ξεκίνημα του βίντεο σε σχέση με τις συνολικές αιτήσεις που εξυπηρετήθηκαν από το σύστημα. Αν έρθει αίτηση για κάποιο βίντεο και δεν έχει ήδη αποθηκευμένο το πρόθεμά του στη μνήμη A τότε υφίσταται καθυστέρηση κατά το ξεκίνημα του βίντεο. Αν η μετρική αυτή έχει μεγάλη τιμή, σημαίνει ότι το σύστημα τείνει να αποθηκεύει τα λάθος βίντεο στη μνήμη A και όχι τα πιο δημοφιλή. Οι καθυστερήσεις αυτές είναι ενοχλητικές για το χρήστη και υπάρχει μεγάλη πιθανότητα να τον αποτρέψουν να συνεχίσει να περιμένει για να δει το βίντεο.

Η τελευταία μετρική απόδοσης που εξετάζουμε είναι ο αριθμός των συνολικών αντικαταστάσεων **(number of replacements)** που γίνονται κατά τη διάρκεια της προσομοίωσης. Αν η μετρική αυτή έχει μικρή τιμή σημαίνει ότι είναι μικρό το κόστος σε ότι αφορά τη διαχείριση της μνήμης και οι στρατηγικές που εφαρμόζονται είναι αποτελεσματικές. Θεωρούμε ότι η αφαίρεση ενός προθέματος από τη μνήμη A και η αφαίρεση ενός suffix από τη μνήμη B αντιστοιχούν σε μια αντικατάσταση αντίστοιχα.

2.3.2 Μοντέλο Προσομοίωσης

Στην επόμενη ενότητα παρουσιάζονται τα αποτελέσματα της προσομοίωσης των μεθόδων *"full video with prefix"*, *"full video with prefix and with limitation"* και *"full video with prefix in LRLFU"* με βάση το μοντέλο προσομοίωσης το οποίο περιγράφεται στην παρούσα ενότητα. Για να μπορεί να γίνει σύγκριση με τη καλύτερη μέθοδο από το [2] χρησιμοποιούμε το ίδιο μοντέλο προσομοίωσης.

Έχουμε υποθέσει ότι οι αφίξεις αιτήσεων για τα διάφορα βίντεο μοντελοποιούνται με κατανομή Poisson [20] και άρα οι χρόνοι μεταξύ διαδοχικών αφίξεων είναι εκθετικά κατανεμημένοι με μέση τιμή $\frac{1}{\lambda}$. Η παράμετρος $\frac{1}{\lambda}$ θεωρείται ίση με 60 sec. Στο σύστημά μας υποτίθεται ότι υπάρχουν V διαφορετικά βίντεο τα οποία αρχικά είναι αποθηκευμένα στη μνήμη του content server. Το μέγεθος των βίντεο καθορίζεται με βάση μια ομοιόμορφη (uniform) κατανομή μεταξύ 0.5M και 1.5M μπλοκ. Το M ορίζεται ίσο με 2.000 και αντιστοιχεί στο μέσο μέγεθος βίντεο. Ο χρόνος που αντιστοιχεί σε ένα block υποτίθεται ίσος με 1.8 δευτερόλεπτα, το οποίο σημαίνει ότι η διάρκεια των βίντεο κυμαίνεται μεταξύ 30 και 90 λεπτών. Το πρόθεμα όπως

ήδη αναφέρθηκε αντιστοιχεί σε Bp blocks. Το Bp θεωρείται ίσο με 32 το οποίο δηλώνει ότι η συνολική διάρκεια του προθέματος είναι 57.6 δευτερόλεπτα.

Η δημοτικότητα των βίντεο όπως προκύπτει από προηγούμενες έρευνες [4], [12], [20] μπορεί να προσεγγιστεί με μια Zipf-like κατανομή. Η κατανομή αυτή ακολουθεί ένα προβλέψιμο πρότυπο. Το πρότυπο ανακαλύφθηκε από τον George Zipf και ονομάζεται νόμος του Zipf. Ο νόμος αυτός ορίζει ότι, αν έχουμε V αντικείμενα τα οποία μπορούμε να διατάξουμε σύμφωνα με τη δημοτικότητά τους, έτσι ώστε x1>x2>…>xV, τότε στη γενική περίπτωση η πιθανότητα εμφάνισης του κ-πιο δημοφιλούς αντικειμένου ισούται με, [6]:

$$f(k,s,v) = \frac{\frac{1}{k^{1-s}}}{\sum_{n=1}^{v} \frac{1}{n^{1-s}}}$$

Από την εξίσωση παρατηρούμε ότι όσο πιο μικρός είναι ο εκθέτης s (skew), τόσο πιο σκεβρωμένη είναι η κατανομή. Ενώ όσο το s πλησιάζει το 1, τόσο περισσότερο η κατανομή πλησιάζει την ομοιόμορφη (για s=1, ταυτίζεται με την ομοιόμορφη κατανομή). Όλα τα αποτελέσματα που παρουσιάζονται στο Κεφάλαιο αυτό αντιστοιχούν σε s=0.2. Επίσης έχουμε υποθέσει, όπως και στο [2], ότι η δημοτικότητα των βίντεο αλλάζει κάθε R αιτήσεις. Όταν συμβεί αυτό, υπολογίζεται μια δεύτερη Zipf-like κατανομή με τις ίδιες παραμέτρους s και V. Για να συσχετίσουμε τις δύο κατανομές χρησιμοποιούμε μια σταθερά k η οποία παίρνει ακέραιες τιμές μεταξύ 1 και V. Το πρώτο πιο δημοφιλές βίντεο στην πρώτη Zipf-like κατανομή θα αντιστοιχεί στο r1 πιο δημοφιλές βίντεο στη δεύτερη Zipf-like κατανομή, το r1 επιλέγεται ομοιόμορφα μεταξύ 1 και k. Ακολούθως το δεύτερο πιο δημοφιλές βίντεο στην πρώτη Zipf-like κατανομή θα αντιστοιχεί στο r2 πιο δημοφιλές βίντεο στη δεύτερη Zipf-like κατανομή, το r2 επιλέγεται ομοιόμορφα μεταξύ 1 και min(V, k+1), εκτός από την τιμή r1 η οποία δεν επιτρέπεται. Αυτό επαναλαμβάνεται μέχρι να ολοκληρωθεί η αλλαγή δημοτικότητας και των V βίντεο.

Η τιμή k αντιστοιχεί στη μέγιστη μετατόπιση στη δημοτικότητα που μπορεί να συμβεί σε ένα βίντεο όταν γίνεται η αντιστοιχία από την πρώτη στη δεύτερη Zipf-like κατανομή. Για k=1 έχουμε τέλεια συσχέτιση και για k=V δεν υπάρχει συσχέτιση. Στην προσομοίωσή μας θεωρούμε k=10 και R=200. Έχουμε δηλαδή υιοθετήσει το σενάριο στο οποίο η δημοτικότητα των βίντεο αλλάζει περίπου κάθε 3 ώρες.

Η συνολική χωρητικότητα της μνήμης του proxy server έχει οριστεί ίση με 400.000 blocks, το οποίο αντιστοιχεί κατά μέσο όρο στο 10% του συνολικού όγκου που καταλαμβάνουν όλα τα βίντεο. Η μνήμη A θα έχει διαθέσιμη μνήμη το 10% της συνολικής χωρητικότητας, δηλαδή 40.000 blocks και το υπόλοιπο 90% θα είναι διαθέσιμο για τη μνήμη B. Όλες οι τιμές των παραμέτρων φαίνονται στον Πίνακα 2.1.

Παράμετροι	**Περιγραφή και προκαθορισμένες τιμές (default values)**
Bp	Μέγεθος προθέματος (32 μπλοκ)
1/λ	Μέση τιμή μεταξύ διαδοχικών αφίξεων αιτήσεων για τα διάφορα βίντεο (60 sec)
V	Αριθμός διακριτών βίντεο στο σύστημα (2.000)
s	Παράμετρος skew στη Zipf-like κατανομή (0.2)
k	Μέγιστη απόσταση μετατόπισης δημοτικότητας βίντεο (10)
R	Αριθμός αιτήσεων για αλλαγή της θέσης δημοτικότητας των βίντεο (200)
C	Συνολική χωρητικότητα μνήμης (400.000 μπλοκ)
pr	Ποσοστό συνολικής μνήμης το οποίο είναι διαθέσιμο για αποθήκευση των αρχικών τμημάτων των βίντεο (μνήμη A) (10%)
Runs	Αριθμός αιτήσεων στην προσομοίωσή μας (100.000 αιτήσεις)

Πίνακας 2.1: Παράμετροι του συστήματος και οι προκαθορισμένες τιμές τους.

2.3.3 Αποτελέσματα Προσομοίωσης

Στην ενότητα αυτή γίνεται μια σύντομη αναφορά στα σημαντικότερα αποτελέσματα που θα παρουσιαστούν στις επόμενες ενότητες, για τις μεθόδους "*full video with prefix*", "*full video with prefix and with limitation*" και "*full video with prefix in LRLFU*" με τα αντίστοιχα αποτελέσματα της μεθόδου από το [2]. Παρατηρούμε ότι στην μετρική BHR και οι τρεις μέθοδοι που υλοποιούν πλήρη αποθήκευση του βίντεο (full video caching) υπερτερούν της μεθόδου στο [2]. Επίσης παρατηρούμε ότι οι καθυστερήσεις στην αρχή του βίντεο μειώνονται όταν γίνει χρήση των δικών μας προτεινόμενων μεθόδων "*full video with prefix*" και "*full video with prefix and with limitation*", ενώ στην περίπτωση της "*full video with prefix in LRLFU*" παραμένουν σχεδόν ίδιες με τη μέθοδο στο [2] εφόσον η στρατηγική αντικατάστασης και αποθήκευσης στη μνήμη A δεν διαφέρει. Οι μικρές διαφορές που παρατηρούνται μεταξύ της μεθόδου "*full video with prefix in LRLFU*" και της μεθόδου στο [2] στην μετρική που αφορά τις καθυστερήσεις στην αρχή των βίντεο οφείλονται

στον αριθμό των blocks που ελέγχουμε για να πάρουμε την απόφαση αν χρησιμοποιείται το συγκεκριμένο βίντεο από το χρήστη, όταν ψάχνουμε για βίντεο θύμα στη μνήμη A. Κατά τη διάρκεια της στρατηγικής αντικατάστασης στη μνήμη A όταν αφαιρούμε το πρόθεμα και υπάρχει το suffix τμήμα στη μνήμη B θα ελέγξουμε περισσότερα blocks στη μέθοδο "*full video with prefix in LRLFU*" σε σχέση με τη μέθοδο στο [2] γιατί το βίντεο εισάγεται ολόκληρο στη μνήμη B και όχι σταδιακά όπως γίνεται στη μέθοδο στο [2] (άρα σε κάποιες περιπτώσεις βρίσκουμε διαφορετικά βίντεο που δεν παρακολουθούνται από κάποιο χρήστη). Ακόμα οι μικρές διαφορές που παρατηρούνται μπορεί να οφείλονται και στο διαφορετικό caching value που χρησιμοποιείται για την επιλογή θυμάτων όταν απαιτείται διαθέσιμη μνήμη. Τέλος παρατηρούμε ότι με τη μέθοδο "*full video with prefix*" αυξάνεται ο αριθμός των συνολικών αντικαταστάσεων σε σχέση με τη μέθοδο στο [2], ενώ η κατάσταση αντιστρέφεται αν γίνει χρήση των πιο επιφυλακτικών μεθόδων "*full video with prefix and with limitation*" και "*full video with prefix in LRLFU*".

Στη συνέχεια παρουσιάζονται τα αποτελέσματα μελέτης της επίδρασης διαφορετικών μεγεθών μνήμης, διαφορετικού αριθμού συνολικών βίντεο, διαφορετικού μέσου μεγέθους βίντεο, διαφορετικού ποσοστού μνήμης για την αποθήκευση των αρχικών μπλοκ των βίντεο, διαφορετικών τιμών της παραμέτρου s της κατανομής δημοτικότητας των βίντεο και διαφορετικών μέγιστων μετατοπίσεων κατά την αλλαγή δημοτικότητας των βίντεο, στις μετρικές απόδοσης του συστήματος.

2.3.3.1 Η επίδραση διαφορετικών μεγεθών μνήμης.

Στην παρούσα ενότητα παρουσιαζόταν τα αποτελέσματα και η αποδοτικότητα των μεθόδων "*full video with prefix*", "*full video with prefix and with limitation*" και "*full video with prefix in LRLFU*" για διάφορες τιμές συνολικού μεγέθους μνήμης, για τις μετρικές απόδοσης που μελετάμε. Από τα αποτελέσματα που ακολουθούν στα Σχ.2.2-Σχ.2.4 παρατηρούμε ότι και οι τρεις μέθοδοι, που αναφέρθηκαν πιο πάνω, είναι καλύτερες σε σχέση με τη μέθοδο στο [2] στην μετρική BHR. Στην μετρική καθυστερήσεις κατά το ξεκίνημα των βίντεο μόνο οι μέθοδοι "*full video with prefix*" και "*full video with prefix and with limitation*" εισάγουν βελτίωση (μεταξύ τους παρουσιάζουν παρόμοια αποτελέσματα). Τέλος στην μετρική που εξετάζει τον συνολικό αριθμό αντικαταστάσεων που συμβαίνουν στην προσομοίωση μόνο οι μέθοδοι "*full video with prefix and with limitation*" και "*full video with prefix in LRLFU*" επιτυγχάνουν καλύτερα αποτελέσματα από τη μέθοδο στο [2].

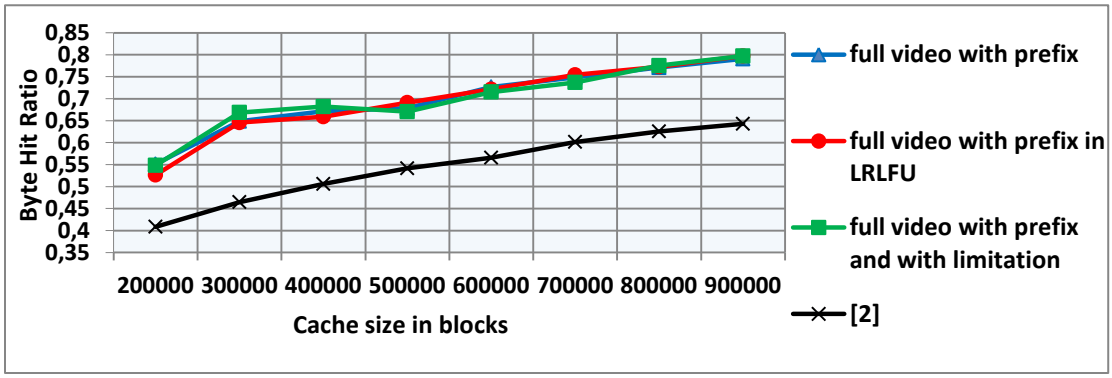
Πιο συγκεκριμένα σε σχέση με τα αποτελέσματα της μεθόδου στο [2], στις μεθόδους "*full video with prefix*" και "*full video with prefix and with limitation*" η επιτυγχανόμενη βελτίωση στο BHR είναι κατά μέσο όρο **29%** και στη μέθοδο "*full video with prefix in LRLFU*" η επιτυγχανόμενη

βελτίωση είναι κατά μέσο όρο **28.5%**. Η μεγάλη βελτίωση που παρατηρείται στις συγκεκριμένες μεθόδους οφείλεται στη στρατηγική της μνήμης Β η οποία επιτρέπει στο βίντεο να αποθηκευτεί γρήγορα στη μνήμη του proxy server, σε αντίθεση με τη στρατηγική της μεθόδου από το [2], η οποία σε κάθε αίτηση προσπαθεί να αποθηκεύσει ένα επιπλέον τμήμα του βίντεο (το μέγεθος του τμήματος ορίστηκε να είναι ένας σταθερός αριθμός από μπλοκ). Στις δικές μας μεθόδους δεν υλοποιείται κάποια τμηματοποίηση στα βίντεο, αντιθέτως θεωρείται το suffix τμήμα κάθε βίντεο σαν ένα μεγάλο ενιαίο τμήμα και έτσι οι συγκεκριμένες μέθοδοι υλοποιούν πλήρη αποθήκευση των βίντεο (full video caching). Όταν το σύστημα φτάσει σε μόνιμη κατάσταση οι συγκεκριμένες μέθοδοι θα έχουν λιγότερα αποθηκευμένα βίντεο στο σύστημα σε σχέση με τη μέθοδο στο [2], αλλά θα μπορούν να εξυπηρετήσουν μεγαλύτερη διάρκεια. Επειδή η μέθοδος στο [2] υλοποιεί τμηματοποίηση στα βίντεο, το σύστημα θα αποθηκεύει περισσότερα βίντεο σε σχέση με τις άλλες τρείς μεθόδους, έχοντας όμως λίγα τμήματα από το καθένα και μόνο τα δημοφιλή βίντεο θα μπορούν να εισάγουν ολόκληρο το περιεχόμενό τους. Όταν έρθει αίτηση για κάποιο βίντεο η μέθοδος στο [2] πιθανόν να μπορεί να εξυπηρετήσει ένα μέρος του συνολικού περιεχομένου και όχι ολόκληρο το περιεχόμενο. Εφόσον έχει γίνει η υπόθεση ότι ο χρήστης παρακολουθεί ολόκληρο το βίντεο όπως φαίνεται οι εισαγωγές και εξαγωγές μεγάλων τμημάτων στη μνήμη Β είναι αποδοτικές.

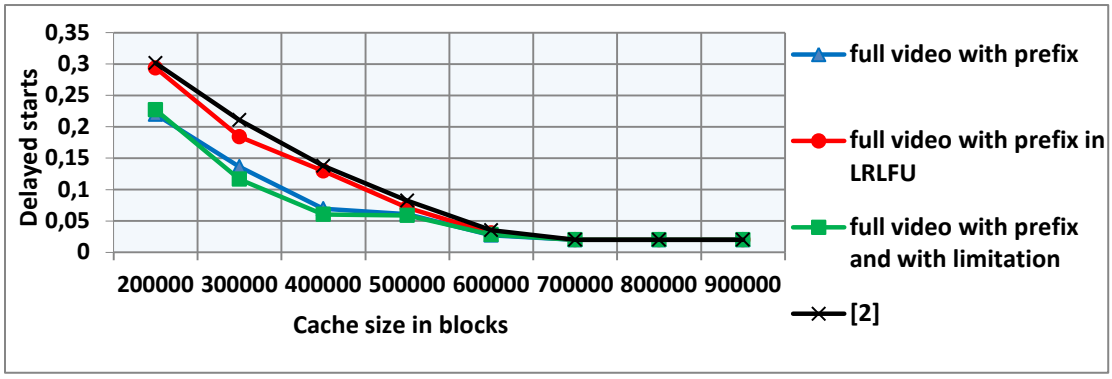
Ακολούθως μελετήσαμε τα αποτελέσματα στην μετρική που αφορά τις καθυστερήσεις κατά το ξεκίνημα των βίντεο όπως φαίνονται στο Σχ.2.3. Σε σχέση με τα αποτελέσματα της μεθόδου στο [2], στη μέθοδο "*full video with prefix*" η επιτυγχανόμενη βελτίωση είναι κατά μέσο όρο **22.5%** και στη μέθοδο "*full video with prefix and with limitation*" η επιτυγχανόμενη βελτίωση είναι κατά μέσο όρο **24.8%**. Αυτό οφείλεται στη διαφορετική στρατηγική που χρησιμοποιείται στη μνήμη Α από τις μεθόδους "*full video with prefix*" και "*full video with prefix and with limitation*". Στην περίπτωση των δύο συγκεκριμένων μεθόδων που αναφέρθηκαν πιο πάνω, όταν η μνήμη Α δεν έχει επαρκή χώρο και απαιτείται ένα βίντεο θύμα θα εξετάζονται όλα τα βίντεο στη μνήμη Α τα οποία δεν είναι currently played λαμβάνοντας υπόψη μόνο το μέγεθος του προθέματος. Δηλαδή ως υποψήφια θύματα θα είναι τα βίντεο στα οποία τα πρώτα 32 blocks δεν παρακολουθούνται από κάποιο χρήστη, χωρίς να μας ενδιαφέρει αν το suffix στη μνήμη Β (αν υπάρχει) παρακολουθείται από κάποιο χρήστη εφόσον δεν θα το αφαιρέσουμε. Όπως ορίστηκε η στρατηγική στη μνήμη Α στο 2.2.3.1, όταν αφαιρείται το πρόθεμα κάποιου βίντεο δεν αφαιρείται και το suffix (αν υπάρχει) από τη μνήμη Β. Επιπλέον κάθε φορά που απαιτείται η αφαίρεση ενός προθέματος από τη μνήμη Α, εξετάζεται αν το βίντεο θύμα είναι λιγότερο δημοφιλές από το βίντεο που έκανε την αίτηση. Όλες οι προαναφερθείσες στρατηγικές έρχονται σε αντίθεση με τη στρατηγική στη μνήμη Α που χρησιμοποιείται στη μέθοδο "*full video with prefix in LRLFU*" και στη μέθοδο στο [2] οι οποίες κάθε φορά που αφαιρούν το πρόθεμα από το βίντεο θύμα αφαιρούν και το suffix μέρος (αν υπάρχει) από τη μνήμη Β. Έτσι όταν η στρατηγική αντικατάστασης στη μνήμη Α ψάχνει για βίντεο θύμα, θα πρέπει να ελέγχει τα βίντεο που δεν παρακολουθούνται από κάποιο χρήστη, βασισμένη στο συνολικό αριθμό των blocks (prefix και suffix) που είναι αποθηκευμένα στη

μνήμη A και στη μνήμη B. Έτσι πολλές φορές αποτρέπεται η αφαίρεση κάποιου μη δημοφιλούς βίντεο γιατί έχει την προστασία «currently played». Στη μέθοδο "*full video with prefix in LRLFU*" τα αποτελέσματα είναι σχεδόν ίδια με τη μέθοδο στο [2] με μικρές διαφορές όπως εξηγήθηκε στην ενότητα 2.3.3.

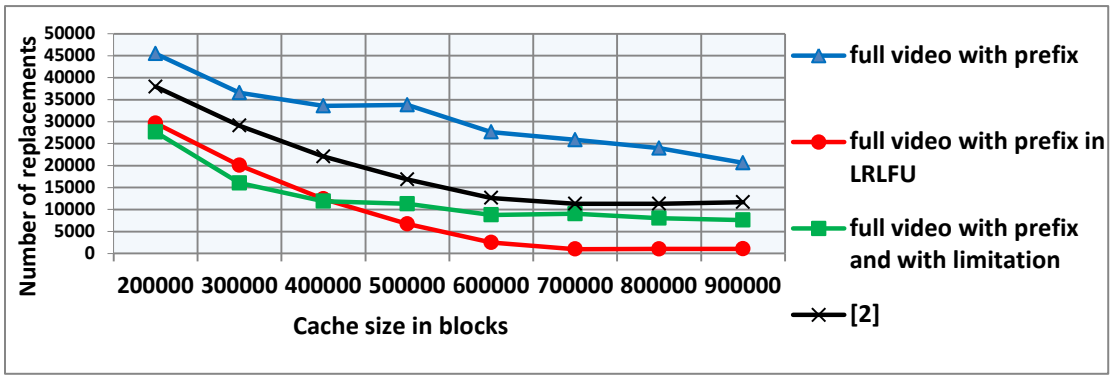
Στο Σχ.2.4 φαίνεται ότι οι μέθοδοι "*full video with prefix and with limitation*" και "*full video with prefix in LRLFU*" έχουν πετύχει το μικρότερο αριθμό συνολικών αντικαταστάσεων. Οι δύο συγκεκριμένες μέθοδοι εξετάζουν αν το βίντεο πληροί κάποιες προϋποθέσεις για να αποθηκευτεί στη μνήμη B, σε αντίθεση με τη μέθοδο "*full video with prefix*" η οποία με την «επιθετική» στρατηγική της αποθηκεύει πάντα το suffix του βίντεο επιφέροντας έτσι στο σύστημα το μεγαλύτερο αριθμό συνολικών αντικαταστάσεων. Επιπρόσθετα, η βελτίωση των μεθόδων "*full video with prefix and with limitation*" και "*full video with prefix in LRLFU*", σε σχέση με τη μέθοδο στο [2], στη συγκεκριμένη μετρική απόδοσης οφείλεται στη χρήση μεγαλύτερων τμημάτων τα οποία αναφέρονται ως suffixes. Στις μεθόδους που υλοποιούν full video caching το suffix θεωρείται ως ενιαίο τμήμα και κοστίζει μια αντικατάσταση κάθε φορά που αφαιρείται. Στη μέθοδο στο [2], το suffix τμήμα είχε διαιρεθεί σε αρκετά τμήματα των 960 blocks. Άρα η απελευθέρωση της απαραίτητης μνήμης, από τη μέθοδο στο [2], απαιτεί την αφαίρεση περισσότερων τμημάτων. Δηλαδή σε όλες τις γραφικές που παρουσιάζονται στο Κεφάλαιο αυτό, όσον αφορά την μετρική συνολικές αντικαταστάσεις που συμβαίνουν στην προσομοίωση, τα αποτελέσματα της μεθόδου στο [2] αντιστοιχούν σε συνολικές αντικαταστάσεις τμημάτων (prefix και τμήματα του suffix), ενώ τα αποτελέσματα στις δικές μας μεθόδους αντιστοιχούν σε συνολικές αντικαταστάσεις (prefix και suffix). Σε σχέση με τη μέθοδο στο [2], η μέθοδος "*full video with prefix*" προσδίδει αύξηση στις αντικαταστάσεις κατά μέσο όρο **66.7%** και οι μέθοδοι "*full video with prefix and with limitation*" και "*full video with prefix in LRLFU*" επιτυγχάνουν κατά μέσο όρο βελτίωση ίση με **33.2%** και **63.5%** αντίστοιχα.



Σχ.2.2: Byte-Hit Ratio σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.



Σχ.2.3: Καθυστερήσεις στην αρχή του βίντεο σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.



Σχ.2.4: Αριθμός συνολικών αντικαταστάσεων σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.

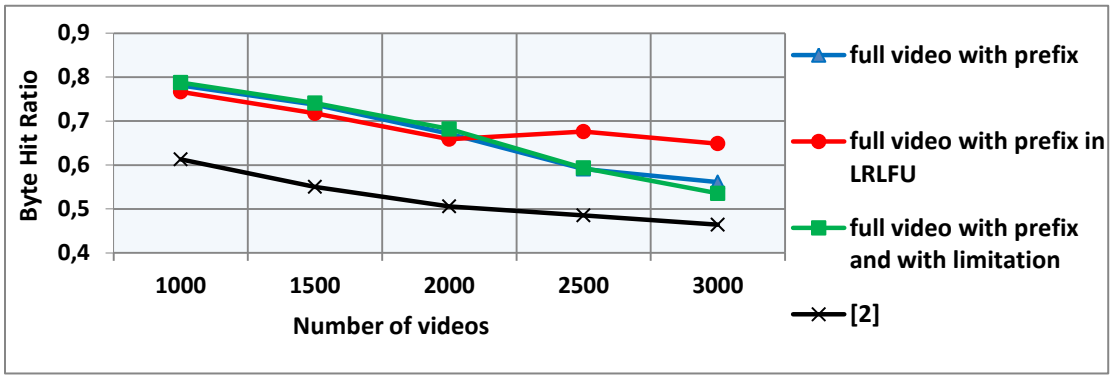
2.3.3.2 Η επίδραση διαφορετικού αριθμού συνολικών βίντεο στο σύστημα

Στην ενότητα αυτή μελετάμε την επίδραση του διαφορετικού αριθμού συνολικών βίντεο στο σύστημα, στις μετρικές απόδοσης BHR, καθυστερήσεις κατά το ξεκίνημα των βίντεο και συνολικές αντικαταστάσεις που συμβαίνουν κατά τη διάρκεια της προσομοίωσης. Τα αποτελέσματα δείχνουν ότι η μέθοδος **"***full video with prefix in LRLFU*" πετυχαίνει το καλύτερο BHR όταν τα βίντεο στο σύστημα είναι περισσότερα από αυτά που υποθέσαμε αρχικά στο Πίνακα 2.1, ενώ όταν τα βίντεο είναι λιγότερα η πιο αποδοτική μέθοδος είναι η "*full video with prefix and with limitation*". Όπως φαίνεται οι μέθοδοι "*full video with prefix*" και "*full video with prefix and with limitation*" εξακολουθούν να πετυχαίνουν τις μικρότερες καθυστερήσεις στην αρχή των βίντεο. Τέλος στην μετρική που αφορά τον συνολικό αριθμό αντικαταστάσεων, η πιο αποδοτική μέθοδος είναι η *"full video with prefix in LRLFU"*.

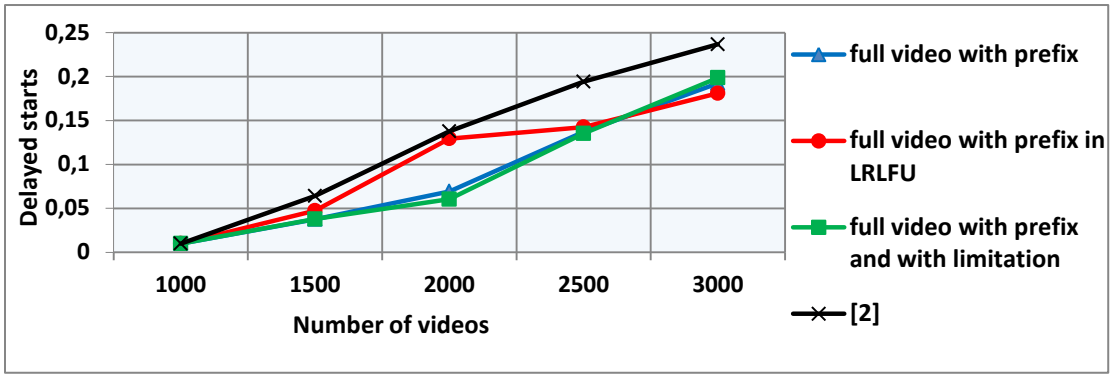
Όπως φαίνεται από το Σχ.2.5 όσο αυξάνεται ο αριθμός των βίντεο στο σύστημα μειώνεται το BHR. Αυτό είναι αναμενόμενο γιατί όσο αυξάνεται ο αριθμός των συνολικών βίντεο, οι στρατηγικές αντικατάστασης που εφαρμόζονται έχουν να επιλέξουν τα δημοφιλή και μη δημοφιλή βίντεο μεταξύ περισσότερων βίντεο. Ακόμη με την αύξηση των βίντεο στο σύστημα θα υπάρχουν και περισσότερες αιτήσεις που θα αναφέρονται σε μη δημοφιλή βίντεο τα οποία πιθανόν να μην βρίσκονται στη μνήμη του proxy server όταν έρθει η αίτηση για αυτά. Συνολικά η επιτυγχανόμενη βελτίωση είναι κατά μέσο όρο **27.8%**, **27.3%** και **32.4%** για τις μεθόδους "*full video with prefix*", "*full video with prefix and with limitation*" και **"***full video with prefix in LRLFU*" αντίστοιχα, σε σχέση με τη μέθοδο στο [2].

Όπως φαίνεται από το Σχ.2.6 αυξάνονται οι καθυστερήσεις κατά το ξεκίνημα των βίντεο με την αύξηση του αριθμού των βίντεο στο σύστημα. Σε σχέση με τα αποτελέσματα της μεθόδου στο [2], οι μέθοδοι "*full video with prefix*" και "*full video with prefix and with limitation*" βελτιώνουν τις καθυστερήσεις που γίνονται στην αρχή των βίντεο κατά μέσο όρο **30.5%** και η μέθοδος **"***full video with prefix in LRLFU*" βελτιώνει στις καθυστερήσεις κατά μέσο όρο **17.8%**.

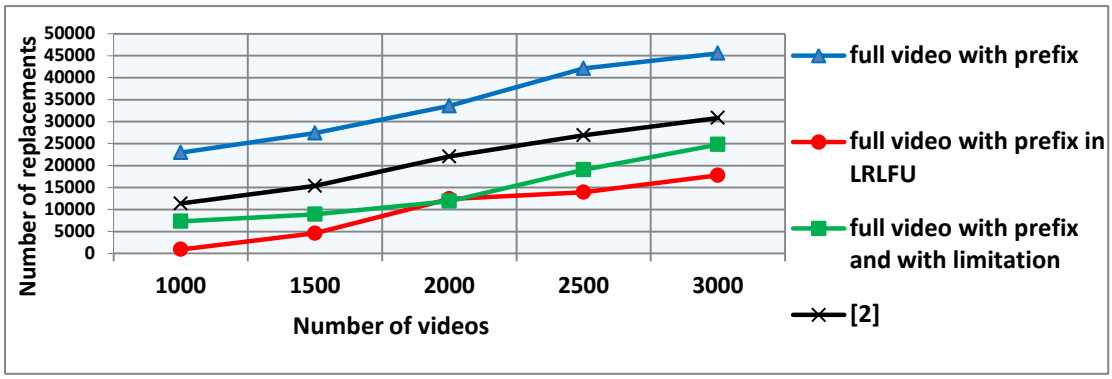
Τέλος εξετάζοντας τα αποτελέσματα όσον αφορά την μετρική συνολικές αντικαταστάσεις που γίνονται στο σύστημα, υπολογίζεται ότι σε σχέση με τα αποτελέσματα της μεθόδου στο [2], η μέθοδος "*full video with prefix*" εισάγει αύξηση στο συνολικό αριθμό αντικαταστάσεων κατά μέσο όρο **67.4%.** Στη μέθοδο "*full video with prefix and with limitation*" η επιτυγχανόμενη βελτίωση είναι κατά μέσο όρο **34.4%** και στη μέθοδο **"***full video with prefix in LRLFU*" η επιτυγχανόμενη βελτίωση είναι κατά μέσο όρο **59.2%.**



Σχ.2.5: Byte-Hit Ratio σαν συνάρτηση με το συνολικό αριθμό βίντεο στο σύστημα.



Σχ.2.6: Καθυστερήσεις στην αρχή του βίντεο σαν συνάρτηση με το συνολικό αριθμό βίντεο στο σύστημα.



Σχ.2.7: Συνολικός αριθμός αντικαταστάσεων σαν συνάρτηση με το συνολικό αριθμό βίντεο στο σύστημα.

2.3.3.3 Η επίδραση διαφορετικού μέσου μεγέθους βίντεο

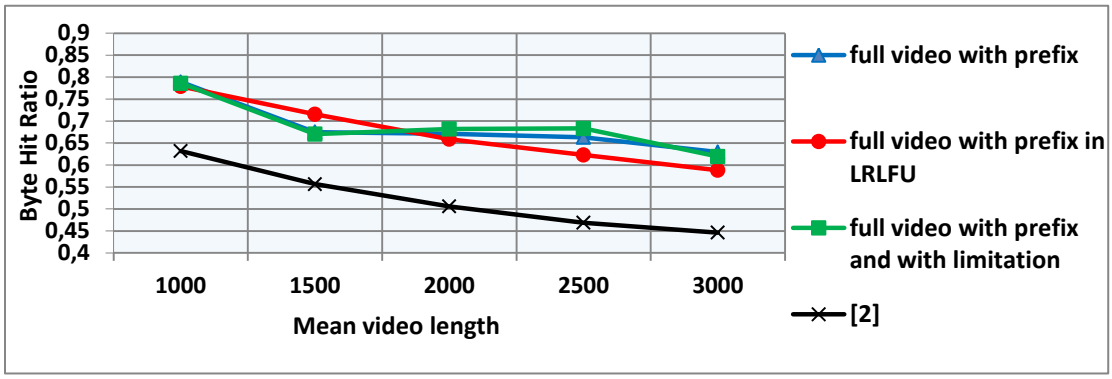
Έχουμε προσομοιώσει το σενάριο στο οποίο αλλάζει το μέσο μέγεθος βίντεο M, δηλαδή αλλάζει η διάρκεια στην οποία κυμαίνονται τα βίντεο. Στο συγκεκριμένο σενάριο όπως φαίνεται από τα Σχ.2.8-Σχ.2.10 η πιο αποδοτική μέθοδος είναι η "*full video with prefix and with limitation*" γιατί πετυχαίνει το υψηλότερο ποσοστό BHR, καθώς επίσης και τις μικρότερες καθυστερήσεις που υπόκεινται οι χρήστες κατά το ξεκίνημα των βίντεο. Οι διαφορές στο συνολικό αριθμό αντικαταστάσεων που παρατηρούνται σε σχέση με τη μέθοδο **"***full video with prefix in LRLFU*" δεν είναι σημαντικές.

Συγκρίνοντας τη μέθοδο "*full video with prefix*" σε σχέση με τα αποτελέσματα της μεθόδου στο [2] παρατηρούμε ότι το BHR αυξάνεται κατά μέσο όρο **32.7%**. Για M=1.000 η αύξηση είναι

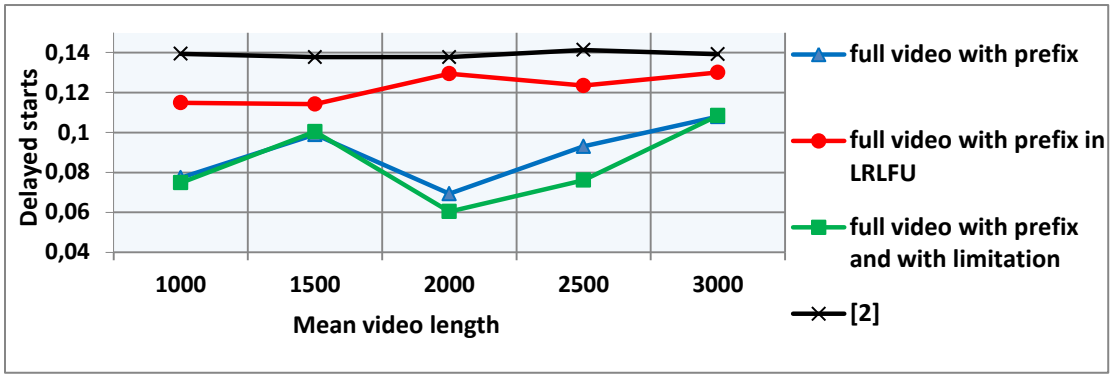
κατά 23.8%, για M=1.500 κατά 21.8%, για M=2.000 κατά 34%, για M=2.500 κατά 43.4% και για M=3.000 κατά 40.9%. Η μέθοδος "*full video with prefix and with limitation*" προσδίδει βελτίωση στο BHR κατά μέσο όρο **33.6%** και η μέθοδος **"***full video with prefix in LRLFU*" κατά μέσο όρο **29.5%**. Όπως περιμέναμε το BHR μειώνεται καθώς αυξάνεται η διάρκεια των βίντεο εφόσον θα αποθηκεύονται λιγότερα βίντεο στη μνήμη B.

Στο Σχ.2.9 παρουσιάζεται η αποτελεσματικότητα των μεθόδων στη μετρική καθυστερήσεις κατά το ξεκίνημα των βίντεο. Σε σχέση με τα αποτελέσματα της μεθόδου στο [2], η επιτυγχανόμενη βελτίωση στη μέθοδο "*full video with prefix*" είναι κατά μέσο όρο **37.9%**, στη μέθοδο "*full video with prefix and with limitation*" η επιτυγχανόμενη βελτίωση είναι κατά μέσο όρο **39.2** και στη μέθοδο **"***full video with prefix in LRLFU*" η επιτυγχανόμενη βελτίωση είναι κατά μέσο όρο **10.5%** . Οι καθυστερήσεις δεν επηρεάζονται πολύ από τη διάρκεια των βίντεο εφόσον το μέγεθος του prefix το οποίο καθορίζει αν έχουμε καθυστέρηση στην αρχή του βίντεο παραμένει το ίδιο, όπως επίσης και η χωρητικότητα μνήμης που αφιερώνεται για τη μνήμη A.

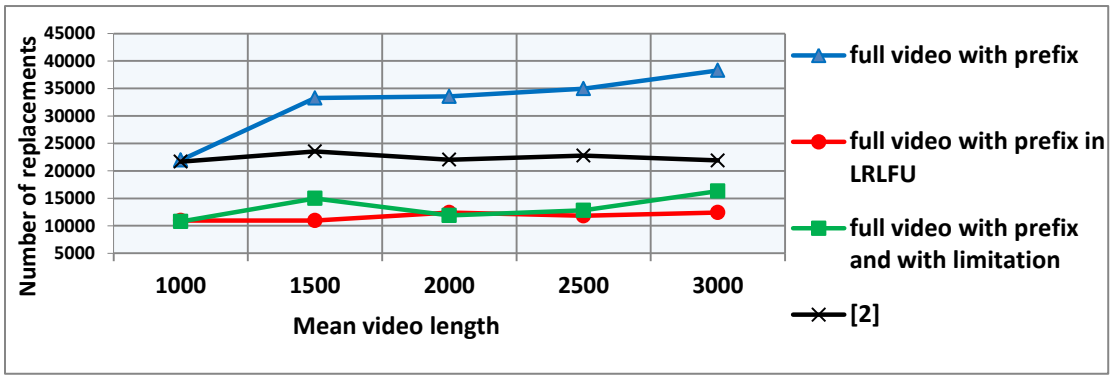
Σε σχέση με τα αποτελέσματα της μεθόδου στο [2], υπολογίζεται ότι η μέθοδος "*full video with prefix*" είναι υποδεέστερη στο συνολικό αριθμό αντικαταστάσεων κατά μέσο όρο **44.7%**. Οι μέθοδοι "*full video with prefix and with limitation*" και **"***full video with prefix in LRLFU*" προσδίδουν βελτίωση στο overhead του συστήματος κατά μέσο όρο **40.3%** και **47.5%** αντίστοιχα (ΣΧ.2.10). Όταν αυξάνεται η διάρκεια των βίντεο ως επακόλουθο είναι να αποθηκεύονται λιγότερα βίντεο στη μνήμη B, άρα θα έχουμε αρκετές αιτήσεις στις οποίες δε θα βρίσκεται στη μνήμη το βίντεο που έκανε την αίτηση και θα χρειαστεί να βρούμε θύματα ώστε να αποθηκευτεί στη μνήμη.



Σχ.2.8: Byte-Hit Ratio σαν συνάρτηση διαφορετικής διάρκειας βίντεο.



Σχ.2.9: Καθυστερήσεις στο ξεκίνημα των βίντεο σαν συνάρτηση διαφορετικής διάρκειας βίντεο.



Σχ.2.10: Συνολικός αριθμός αντικαταστάσεων σαν συνάρτηση διαφορετικής διάρκειας βίντεο.

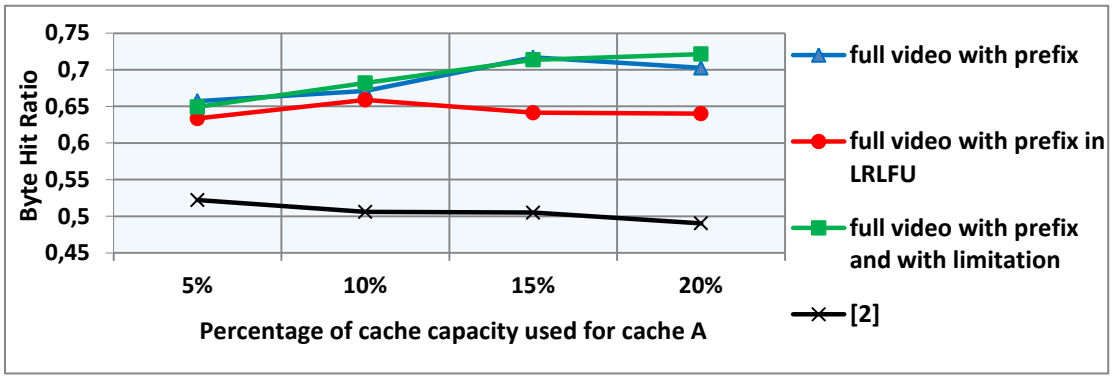
2.3.3.4 Η επίδραση διαφορετικού διαθέσιμου ποσοστού μνήμης για τα αρχικά τμήματα των βίντεο

Το ποσοστό της μνήμης που διαθέτει το σύστημα για τη μνήμη Α είναι θεμελιώδες για τις καθυστερήσεις που θα υπάρξουν κατά το ξεκίνημα των βίντεο. Επιπρόσθετα καθορίζει και το BHR γιατί όσο περισσότερο μέρος της συνολικής μνήμης αφιερώνεται για τη μνήμη Α, τόσο λιγότερο μέρος διατίθεται για τη μνήμη Β και έτσι μπορούν να αποθηκεύονται λιγότερα suffixes από τα διάφορα βίντεο. Τα αποτελέσματα δείξανε για άλλη μια φορά ότι η μέθοδος *"full video with prefix and with limitation"* πετυχαίνει τις μικρότερες καθυστερήσεις κατά το ξεκίνημα των βίντεο και το μεγαλύτερο BHR. Παρατηρούνται μικρές διαφορές στο συνολικό αριθμό αντικαταστάσεων σε σχέση με τη μέθοδο **"***full video with prefix in LRLFU*" και εφόσον η συγκεκριμένη μετρική απόδοσης δεν είναι τόσο σημαντική όσο το BHR και οι καθυστερήσεις κατά το ξεκίνημα των βίντεο, εξακολουθεί η μέθοδος *"full video with prefix and with limitation"* να είναι η πιο αποδοτική.

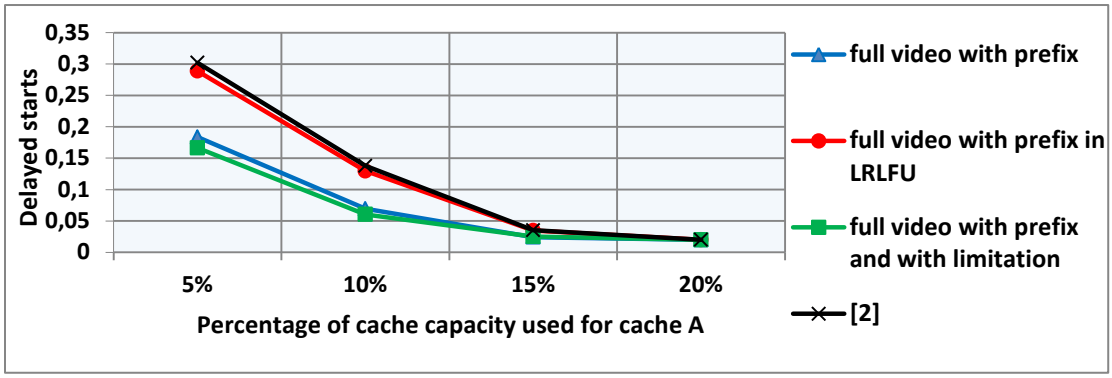
Εξετάζοντας τα αποτελέσματα όσον αφορά την μετρική απόδοσης BHR, υπολογίζεται ότι η επιτυγχανόμενη βελτίωση στη μέθοδο "*full video with prefix*" είναι κατά μέσο όρο **35.9%** και η επιτυγχανόμενη βελτίωση στις μεθόδους "*full video with prefix and with limitation*" και **"***full video with prefix in LRLFU*" είναι κατά μέσο όρο **37%** και **27.4%** αντίστοιχα, συγκρίνοντας με τα αποτελέσματα της μεθόδου στο [2]. Όπως φαίνεται στο Σχ.2.11 οι δικές μας μέθοδοι ("*full video with prefix*" και "*full video with prefix and with limitation*") είναι αρκετά αποτελεσματικές όταν η μνήμη B είναι περιορισμένη.

Σε σχέση με το [2], η μέθοδος "*full video with prefix*" προσδίδει βελτίωση στις καθυστερήσεις που συμβαίνουν κατά το ξεκίνημα των βίντεο κατά μέσο όρο **31.7%**, η μέθοδος "*full video with prefix and with limitation*" προσδίδει βελτίωση κατά μέσο όρο **33.4%** και η μέθοδος **"***full video with prefix in LRLFU*" προσδίδει βελτίωση μόλις κατά μέσο όρο **3.5%** (Σχ.2.12).

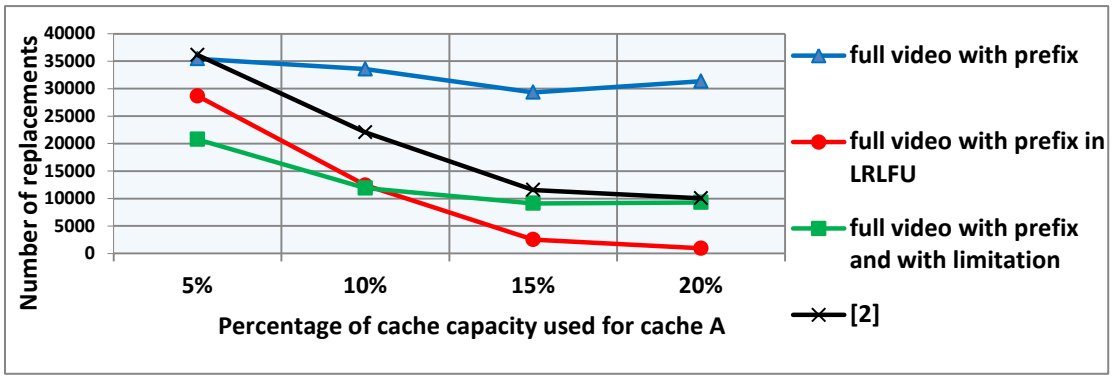
Εξετάζοντας την μετρική που αφορά τις συνολικές αντικαταστάσεις, όπως φαίνεται από το Σχ.2.13 η μέθοδος "*full video with prefix*" επιφέρει το μεγαλύτερο συνολικό κόστος. Η συγκεκριμένη μέθοδος για διαθέσιμο ποσοστό χρήσης της μνήμης A ίσο με 5% βελτιώνει τις συνολικές αντικαταστάσεις κατά 2%, σε σχέση με τη μέθοδο στο [2], αλλά για διαθέσιμο ποσοστό για τη μνήμη A ίσο με 20% έχουμε τριπλάσιο αριθμό συνολικών αντικαταστάσεων. Στη μέθοδο "*full video with prefix with limitation*" η επιτυγχανόμενη βελτίωση είναι κατά μέσο όρο **29.3%** και στη μέθοδο **"***full video with prefix in LRLFU*" η επιτυγχανόμενη βελτίωση είναι κατά μέσο όρο **58.3%**. Όπως φαίνεται για περιορισμένη χωρητικότητα της μνήμης A γίνονται αρκετές αφαιρέσεις εφόσον η μνήμη A δε μπορεί να αποθηκεύει αρκετά προθέματα.



Σχ.2.11: Byte-Hit Ratio σαν συνάρτηση με το διαθέσιμο ποσοστό της μνήμης A.



Σχ.2.12: Καθυστερήσεις κατά το ξεκίνημα των βίντεο σαν συνάρτηση με το διαθέσιμο ποσοστό της μνήμης Α.



Σχ.2.13: Συνολικές αντικαταστάσεις σαν συνάρτηση με το διαθέσιμο ποσοστό της μνήμης Α.

2.3.3.5 Η επίδραση διαφορετικών τιμών της παραμέτρου skew της κατανομής δημοτικότητας των βίντεο

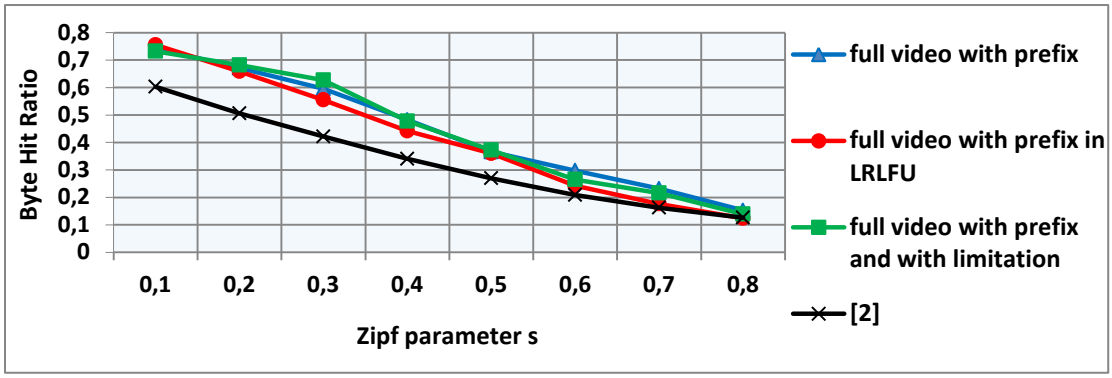
Τα αποτελέσματα που ακολουθούν στις γραφικές Σχ.2.14, Σχ.2.15 και Σχ.2.16 δείχνουν την επίδραση σε κάθε μέθοδο όταν αλλάζει η παράμετρος skew στην Zipf-like κατανομή. Όσο η παράμετρος s πλησιάζει το 0, τόσο πιο εκθετική είναι η Zipf-like κατανομή και τόσο πιο εύκολο είναι να ξεχωρίζουν τα δημοφιλή βίντεο γιατί οι πιθανότητες εμφάνισής τους είναι αρκετά διαφορετικές. Αντίστοιχα όσο η παράμετρος s πλησιάζει το 1, πλησιάζει την ομοιόμορφη κατανομή και οι πιθανότητες εμφάνισης κάθε βίντεο είναι πιο κοντινές. Το σενάριο αυτό είναι πολύ σημαντικό, καθώς μπορεί να μας δώσει μια αξιόπιστη εικόνα για τη συμπεριφορά των μεθόδων μας σε ένα περιβάλλον στο οποίο οι χρήστες έχουν παρόμοιες προτιμήσεις και σε ένα εντελώς διαφορετικό περιβάλλον στο οποίο οι προτιμήσεις είναι αρκετά διαφορετικές. Όπως φαίνεται από τα αποτελέσματα οι μέθοδοι "*full video with prefix*" και "*full video with prefix and with limitation*"

είναι οι πιο αποτελεσματικές στη μετρική BHR με την πρώτη μέθοδο να είναι καλύτερη με μικρή διαφορά. Οι ίδιες μέθοδοι μπορούν επίσης να δώσουν και τις μικρότερες καθυστερήσεις που συμβαίνουν κατά το ξεκίνημα των βίντεο με μικρές διαφορές μεταξύ τους. Εφόσον η μέθοδος "*full video with prefix and with limitation*" πετυχαίνει και τις λιγότερες αντικαταστάσεις τμημάτων συμπεραίνουμε ότι εξακολουθεί να είναι η πιο αποδοτική μέθοδος και σε αυτό το σενάριο.

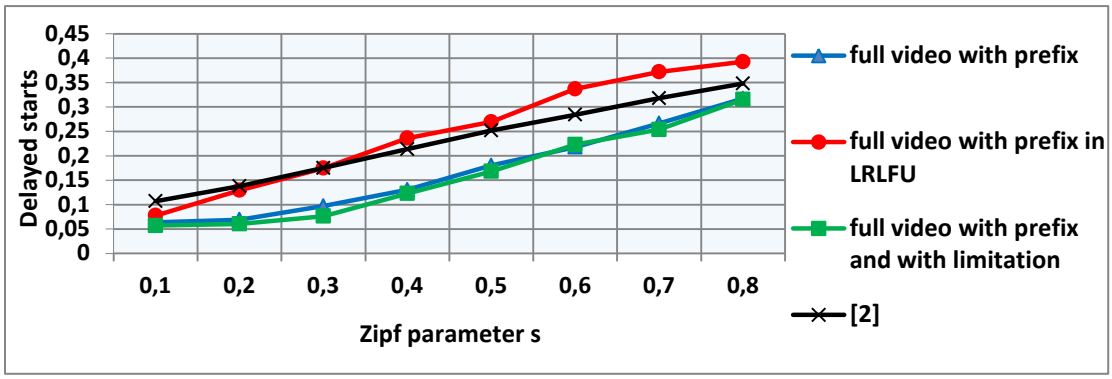
Όπως φαίνεται από το Σχ.2.14 οι μέθοδοι "*full video with prefix*" και "*full video with prefix and with limitation*" υπερτερούν της μεθόδου **"***full video with prefix in LRLFU*" όταν οι προτιμήσεις των χρηστών είναι αρκετά διαφορετικές. Η μέθοδος "*full video with prefix*" με την «επιθετική» στρατηγική που εφαρμόζει στη μνήμη Β μπορεί να ανταποκρίνεται γρήγορα και να αποθηκεύει τα δημοφιλή βίντεο στη μνήμη Β. Επίσης η μέθοδος "*full video with prefix and with limitation*" έχει πιο «χαλαρό» έλεγχο για να αποφασίσει αν το βίντεο πρέπει να αποθηκευτεί στη μνήμη Β σε σχέση με την **"***full video with prefix in LRLFU*", όταν δεν υπάρχει η απαραίτητη διαθέσιμη μνήμη. Αυτό όπως φαίνεται δίνει προβάδισμα στη μέθοδο ώστε να ανταποκρίνεται πιο γρήγορα στις διαφορετικές προτιμήσεις χρηστών. Σε σχέση με τη μέθοδο στο [2], η μέθοδος "*full video with prefix*" βελτίωσε το BHR κατά μέσο όρο **35.1%** και οι μέθοδοι "*full video with prefix and with limitation*" και **"***full video with prefix in LRLFU*" εισάγουν βελτίωση κατά μέσο όρο **31.2%** και **21.4%** αντίστοιχα.

Εξετάζοντας την μετρική που αφορά το ποσοστό των χρηστών που είχαν υποστεί καθυστέρηση κατά το ξεκίνημα των βίντεο παρατηρούμε ότι η μέθοδος "*full video with prefix and with limitation*" έχει πετύχει τα καλύτερα αποτελέσματα. Συγκεκριμένα σε σχέση με τη μέθοδο στο [2], η επιτυγχανόμενη βελτίωση στη μέθοδο "*full video with prefix*" είναι κατά μέσο όρο **32.1%**, η επιτυγχανόμενη βελτίωση στη μέθοδο "*full video with prefix and with limitation*" είναι κατά μέσο όρο **36.3%** και η μέθοδος "*full video with prefix in LRLFU"* είναι υποδεέστερη κατά μέσο όρο **3.4%**. Αν συγκρίνουμε τις δικές μας μεθόδους με τη μέθοδο "*full video with prefix in LRLFU"* παρατηρούμε ότι επιφέρουν πολύ μικρότερες καθυστερήσεις όπως φαίνεται στο Σχ.2.15. Πιο συγκεκριμένα η μέθοδος "*full video with prefix*" είναι κατά μέσο όρο **35.4%** καλύτερη από την μέθοδο "*full video with prefix in LRLFU"* και η μέθοδος "*full video with prefix and with limitation*" είναι κατά μέσο όρο **38.7%** καλύτερη.

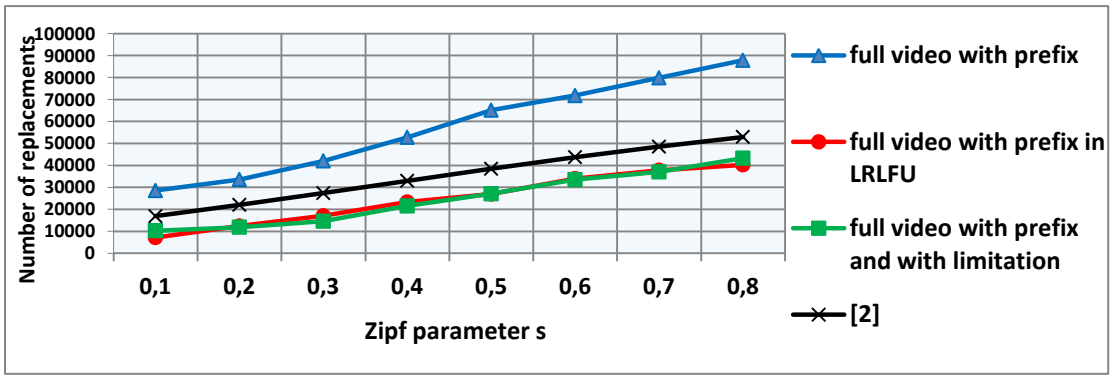
Ακολούθως στο Σχ.2.16 απεικονίζονται τα αποτελέσματά στην μετρική που αφορά το συνολικό αριθμό αντικαταστάσεων που γίνονται στο σύστημα. Σε σχέση με τη μέθοδο στο [2], η μέθοδος "*full video with prefix*" είναι υποδεέστερη κατά μέσο όρο **62.2%.** Στις μεθόδους "*full video with prefix and with limitation*" και "*full video with prefix in LRLFU"* η επιτυγχανόμενη βελτίωση είναι κατά μέσο όρο **32.6%** και **33.3%**, αντίστοιχα. Παρατηρούμε ότι η μέθοδος "*full video with prefix*" έχει συχνά διπλάσιο αριθμό συνολικών αντικαταστάσεων σε σχέση με την μέθοδο "*full video with prefix in LRLFU"*. Με τη μέθοδο "*full video with prefix and with limitation*" περιορίσαμε τον αριθμό συνολικών αντικαταστάσεων έχοντας παρόμοια αποτελέσματα με την μέθοδο "*full video with prefix in LRLFU"*.



Σχ.2.14: Byte-Hit Ratio σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.



Σχ.2.15: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.



Σχ.2.16: Αριθμός συνολικών αντικαταστάσεων σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.

2.3.3.6 Η επίδραση διαφορετικών μέγιστων μετατοπίσεων κατά την αλλαγή δημοτικότητας των βίντεο

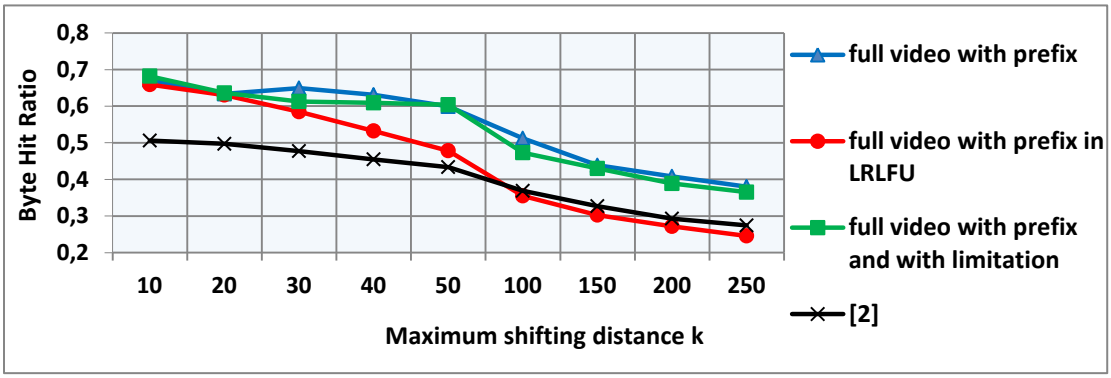
Το τελευταίο σενάριο στην προσομοίωσή μας ελέγχει τη συμπεριφορά των μεθόδων σε σχέση με το ρυθμό αλλαγής στη δημοτικότητα των βίντεο. Στο σενάριο μας έχουμε υποθέσει ότι η δημοτικότητα των βίντεο αλλάζει περίπου κάθε 3 ώρες. Το κατά πόσο η αλλαγή αυτή θα είναι απότομη, δηλαδή τα βίντεο που ήτανε μέχρι τώρα πολύ δημοφιλή γίνουν μη δημοφιλή και αντίστοιχα τα μη δημοφιλή βίντεο ανήκουν πλέον στα δημοφιλή, καθορίζεται από την τιμή της μεταβλητής k στο Πίνακα 2.1. Όπως ήδη αναφέρθηκε κάθε 3 ώρες υπολογίζεται μια νέα Zipf-like κατανομή και γίνεται συσχέτιση στη δημοτικότητα των βίντεο από την πρώτη κατανομή στη δεύτερη, ανάλογα με τις επιτρεπτές τιμές που ορίζει η μέγιστη μετατόπιση δημοτικότητας, δηλαδή η παράμετρος k. Τα αποτελέσματα δείχνουν ότι η μέθοδος "*full video with prefix*" πετυχαίνει το υψηλότερο BHR. Η μέθοδος "*full video with prefix and with limitation*" πετυχαίνει παρόμοιο BHR με τη μέθοδο "*full video with prefix*", πετυχαίνοντας επιπρόσθετα τις μικρότερες καθυστερήσεις και το μικρότερο αριθμό συνολικών αντικαταστάσεων στο σύστημα.

Πιο συγκεκριμένα εξετάζοντας την μετρική BHR σε σχέση με τη μέθοδο στο [2], υπολογίζεται ότι η μέθοδος "*full video with prefix*" προσδίδει βελτίωση κατά μέσο όρο **36.7%**, η μέθοδος "*full video with prefix and with limitation*" προσδίδει βελτίωση κατά μέσο όρο **32.5%** και η μέθοδος "*full video with prefix in LRLFU"* προσδίδει βελτίωση μόνο κατά **9.1%**. Όπως φαίνεται από το Σχ.2.17 οι δικές μας μέθοδοι ("*full video with prefix*" και "*full video with prefix and with limitation*") υπερτερούν της μεθόδου *"full video with prefix in LRLFU"* στη συγκεκριμένη μετρική απόδοσης, καθώς μπορούν να ανταποκρίνονται πιο γρήγορα στις απότομες αλλαγές που συμβαίνουν στο σύστημα εξαιτίας της πιο «χαλαρής» στρατηγικής αντικατάστασης που εφαρμόζεται στη μνήμη Β.

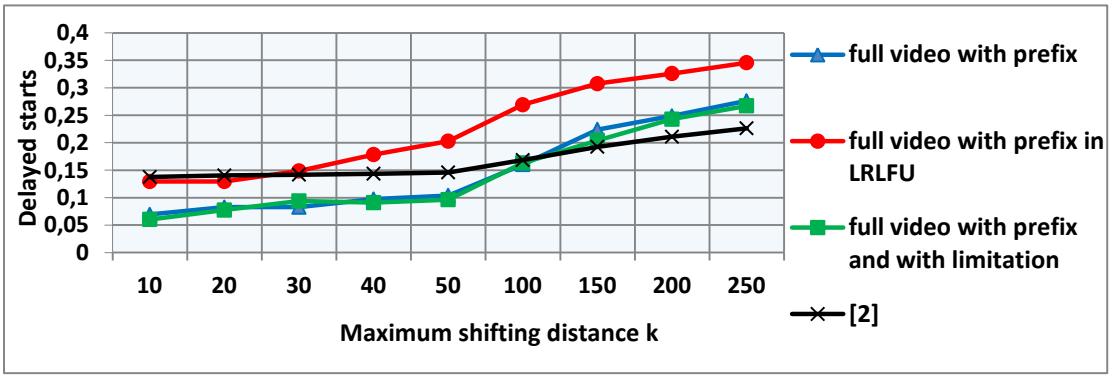
Συγκρίνοντας τα αποτελέσματα στην μετρική που αφορά τις καθυστερήσεις κατά το ξεκίνημα των βίντεο, υπολογίζεται ότι η επιτυγχανόμενη βελτίωση στη μέθοδο *"full video with prefix*" είναι κατά μέσο όρο **16.6%** σε σχέση με τη μέθοδο στο [2]. Όπως φαίνεται στο Σχ.2.18, όσο η μέγιστη μετατόπιση δημοτικότητας είναι μικρότερη από 100 η μέθοδός μας εισάγει βελτίωση, όμως για μεγαλύτερη τιμή στη μέγιστη επιτρεπτή μετατόπιση δημοτικότητας είναι χειρότερες οι καθυστερήσεις σε σχέση με τη μέθοδο στο [2]. Αυτό οφείλεται στη διαφορετική στρατηγική αντικατάστασης στη μνήμη Α. Η μέθοδος στο [2], όταν ψάχνει τα βίντεο που δεν παρακολουθούνται από κάποιο χρήστη, εξετάζει όλο το αποθηκευμένο περιεχόμενο των βίντεο (prefix και suffix). Αυτό μπορεί να προστατεύσει τα βίντεο και που είναι currently played. Τα συγκεκριμένα βίντεο αν ανήκουν στα μη δημοφιλή, μετά την ολοκλήρωση της απότομης αλλαγής στη μετατόπιση δημοτικότητας των βίντεο, θα ανήκουν πλέον στα δημοφιλή και έτσι το σύστημα μπορεί να αντιληφθεί τα νέα δεδομένα. Σε σχέση με τη μέθοδο στο [2], η επιτυγχανόμενη

βελτίωση στη μέθοδο "*full video with prefix and with limitation*" είναι κατά μέσο όρο **19.2%** και σε σχέση με τη μέθοδο "*full video with prefix in LRLFU*" είναι καλύτερη κατά μέσο όρο **39%**.

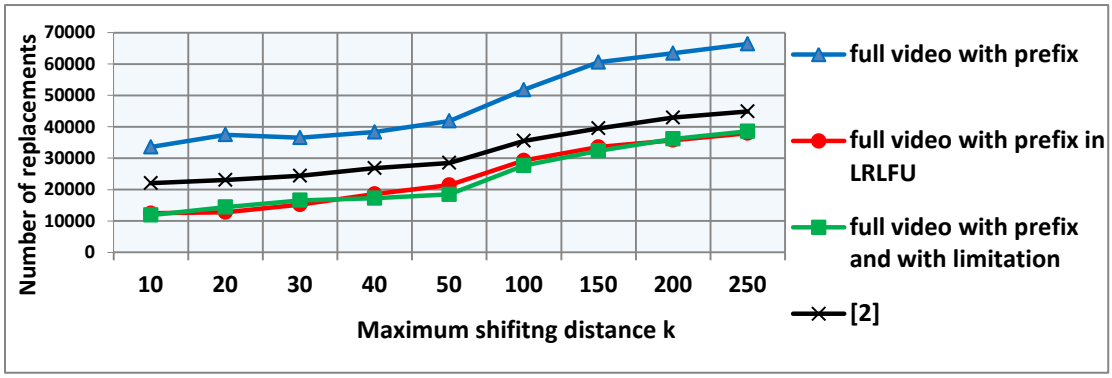
Στο Σχ.2.19 εξετάζεται η μετρική που αφορά το συνολικό αριθμό αντικαταστάσεων, σε σχέση με τη μέγιστη μετατόπιση δημοτικότητας των βίντεο. Σε σχέση με τη μέθοδο στο [2], υπολογίζεται ότι η μέθοδος "*full video with prefix*" είναι υποδεέστερη κατά μέσο όρο **49.8%** ,η μέθοδος "*full video with prefix and with limitation*" προσδίδει μείωση κατά μέσο όρο **28.5%** και η μέθοδος "*full video with prefix in LRLFU*" προσδίδει μείωση κατά μέσο όρο ίση με **27.3%**.



Σχ.2.17: Byte-Hit Ratio σαν συνάρτηση με τη μέγιστη μετατόπιση στη δημοτικότητα των βίντεο.



Σχ.2.18: Καθυστερήσεις κατά το ξεκίνημα των βίντεο σαν συνάρτηση με τη μέγιστη μετατόπιση στη δημοτικότητα των βίντεο.



Σχ.2.19: Αριθμός συνολικών αντικαταστάσεων σαν συνάρτηση με τη μέγιστη μετατόπιση δημοτικότητας των βίντεο.

2.4 Συμπεράσματα από τα αποτελέσματα των προσομοιώσεων

Στην ενότητα αυτή παρουσιάζονται τα σημαντικότερα συμπεράσματα που προκύπτουν από τις προσομοιώσεις του Κεφαλαίου. Από τα αποτελέσματα των προσομοιώσεων μπορούμε να έχουμε μια συνολική εικόνα για τις τρείς μετρικές απόδοσης. Επιπλέον, μέσα από τα αποτελέσματα φαίνεται η συμπεριφορά κάθε μεθόδου σε διαφορετικές συνθήκες και η αποτελεσματικότητά της στα διάφορα σενάρια.

Καταλήγουμε στο συμπέρασμα ότι η πλήρης αποθήκευση των βίντεο (full video caching) είναι πιο αποδοτική και μπορεί να δώσει μεγάλη αύξηση στη μετρική BHR στην περίπτωση που οι χρήστες παρακολουθούν ολόκληρο το βίντεο. Αντιθέτως, η μερική αποθήκευση των βίντεο (partial video caching), που υλοποιείται από τη μέθοδο στο [2], δίνει υποδεέστερα αποτελέσματα στη συγκεκριμένη μετρική απόδοσης, γιατί ένα δημοφιλές βίντεο εισάγει ολόκληρο το περιεχόμενό του μετά από αρκετές αιτήσεις. Οι δικές μας μέθοδοι δεν υλοποιούν κάποια τμηματοποίηση στο βίντεο, σε αντίθεση με τη μέθοδο στο [2], στην οποία υλοποιείται τμηματοποίηση στα βίντεο. Το αποτέλεσμα αυτής της διαφοροποίησης είναι ότι στις δικές μας μεθόδους, όταν το σύστημα φτάσει σε μόνιμη κατάσταση, θα υπάρχουν λιγότερα αποθηκευμένα βίντεο στη μνήμη του διακομιστή μεσολάβησης σε σχέση με τη μέθοδο στο [2], αλλά τα συγκεκριμένα βίντεο θα έχουν αποθηκευμένο ολόκληρο το περιεχόμενό τους. Στη μέθοδο στο [2] όταν το σύστημα φτάσει σε μόνιμη κατάσταση, το αποτελέσματα θα είναι ότι θα υπάρχουν περισσότερα βίντεο σε σχέση με τις μεθόδους μας στη μνήμη του διακομιστή μεσολάβησης, όμως τα περισσότερα από αυτά θα είναι μερικώς αποθηκευμένα. Έτσι η μέθοδος στο [2] σε αρκετές αιτήσεις, που απαιτούν παρακολούθηση ολόκληρου του βίντεο, αδυνατεί να εξυπηρετήσει ολόκληρο το περιεχόμενο της αίτησης που απαιτείται, αλλά τις περισσότερες φορές μπορεί να εξυπηρετήσει μόνο ένα μέρος του

βίντεο. Η συγκεκριμένη διαφοροποίηση που εξηγήθηκε πιο πάνω δικαιολογεί το υψηλό BHR που πετυχαίνεται από τις μεθόδους μας που υλοποιούν πλήρη αποθήκευση των βίντεο.

Οι μέθοδοι "*full video with prefix*" και "*full video with prefix and with limitation*" φαίνεται ότι βρίσκονται σε πλεονεκτική θέση σε σχέση με τη μέθοδο **"***full video with prefix in LRLFU*" και τη μέθοδο στο [2] στην μετρική που αφορά τις καθυστερήσεις κατά το ξεκίνημα των βίντεο. Από όλες τις περιπτώσεις που μελετήθηκαν γίνεται φανερό ότι οι μειωμένες καθυστερήσεις κατά το ξεκίνημα των βίντεο μπορούν να επιτευχθούν με την πλήρη ανεξαρτησία του prefix και suffix τμήματος κάθε βίντεο. Η ανεξαρτησία οφείλεται στη στρατηγική αντικατάστασης της μνήμης Α που εφαρμόζεται στις μεθόδους "*full video with prefix*" και "*full video with prefix and with limitation*", η οποία όταν χρειαστεί να αφαιρέσει το prefix του βίντεο θύματος δεν αφαιρεί και το suffix τμήμα (αν υπάρχει) από τη μνήμη Β, σε αντίθεση με τη μέθοδο **"***full video with prefix in LRLFU*" και τη μέθοδο στο [2] οι οποίες αφαιρούν και το suffix τμήμα (αν υπάρχει) από τη μνήμη Β. Οι μέθοδοι "*full video with prefix*" και "*full video with prefix and with limitation*" επιτρέπουν στη μνήμη Α και στη μνήμη Β να αποφασίσουν αν πρέπει να μηδενιστούν οι μεταβλητές κάποιου μη δημοφιλούς βίντεο. Η μνήμη Α θα διαγράψει όλη την πληροφορία εφόσον αφαιρεί το πρόθεμα κάποιου βίντεο και δεν υπάρχει το suffix τμήμα αποθηκευμένο στη μνήμη Β. Επιτρέπεται και στη μνήμη Β να διαγράψει όλη την πληροφορία εφόσον αφαιρέσει το suffix κάποιου βίντεο και δεν υπάρχει το πρόθεμά του αποθηκευμένο στη μνήμη Α. Αυτός ο βαθμός ελευθερίας που αποδίδεται διαφοροποιεί τις μεθόδους "*full video with prefix*" και "*full video with prefix and with limitation*", σε σχέση με τη μέθοδο **"***full video with prefix in LRLFU*" και τη μέθοδο στο [2], στον αριθμό των blocks που ελέγχει η μνήμη Α για να δει αν το βίντεο χρησιμοποιείται από κάποιο χρήστη, σε περίπτωση που προσπαθεί να βρει κάποιο βίντεο θύμα. Στις δικές μας μεθόδους η μνήμη Α ελέγχει μόνο το πρόθεμα των βίντεο (32 blocks) που είναι αποθηκευμένα σε αυτή για να βρει αυτά που δεν παρακολουθούνται από κάποιο χρήστη και να αφαιρέσει αν χρειάζεται χώρος αυτό με το μικρότερο caching value. Στην περίπτωση όμως της μεθόδου **"***full video with prefix in LRLFU*" και της μεθόδου από το [2] αν το βίντεο θύμα έχει και το suffix αποθηκευμένο, τότε η μνήμη Α θα πρέπει να ελέγξει περισσότερα blocks γιατί όταν αφαιρεί το πρόθεμα κάποιου βίντεο αφαιρεί και το suffix τμήμα του. Έτσι κάποιες φορές αποτρέπεται η αφαίρεση κάποιου μη δημοφιλούς βίντεο επειδή έχει την «προστασία» ότι είναι currently played. Επιπλέον στις μεθόδους "*full video with prefix*" και "*full video with prefix and with limitation*" πριν αφαιρεθεί το πρόθεμα από το βίντεο θύμα εξετάζεται αν το βίντεο θύμα είναι λιγότερο δημοφιλές από το βίντεο που έκανε την αίτηση.

Παρατηρούμε ότι όσο πιο αυστηρός είναι ο έλεγχος για εισαγωγή κάποιου suffix στη μνήμη Β, τόσο μικρότερος θα είναι ο αριθμός των συνολικών αντικαταστάσεων, αλλά το σύστημα θα αργεί να ανταποκρίνεται γρήγορα σε απότομες αλλαγές στη δημοτικότητα των βίντεο ή σε διαφορετικές προτιμήσεις χρηστών και έτσι θα έχει μικρότερο ποσοστό BHR. Όταν δεν υπάρχει επαρκής χώρος στη μνήμη Β, η μέθοδος "*full video with prefix in LRLFU"* για να αφαιρέσει κάποιο βίντεο θύμα πρέπει να έχει μικρότερο caching value σε σχέση με το caching value του βίντεο που

έκανε την αίτηση. Σε αυτή την περίπτωση η μέθοδος *"full video with prefix and with limitation"* θα αφαιρέσει κάποιο βίντεο θύμα, αν έχει μικρότερο χρόνο τελευταίας αναφοράς σε σχέση με το βίντεο που έκανε την αίτηση. Αυτός ο έλεγχος είναι πιο χαλαρός και έτσι το σύστημα προσαρμόζεται πιο γρήγορα στα νέα δεδομένα.

Τέλος οι μέθοδοι *"full video with prefix"* και *"full video with prefix and with limitation"* μπορούν να δώσουν μια συνολική εικόνα για τη σχέση των μετρικών απόδοσης BHR, delayed starts και number of replacements. Το υψηλό BHR και οι μειωμένες καθυστερήσεις μπορούν να επιτευχθούν χωρίς να χρειαστεί να «πληρώσουμε» τις αυξημένες αντικαταστάσεις.

Κεφάλαιο 3: Μέθοδοι αποθήκευσης και αντικατάστασης βίντεο σε διακομιστή μεσολάβησης και υπολογισμός απόδοσης σε μερική παρακολούθηση των βίντεο

3.1 Εισαγωγή

Στο Κεφάλαιο αυτό παρουσιάζουμε τις αλγοριθμικές ιδέες που προτείνουμε για αποθήκευση των βίντεο στον διακομιστή μεσολάβησης με σκοπό να υποστηρίξουμε αποδοτικά την μερική παρακολούθηση των βίντεο. Όπως αναφέρθηκε στην εισαγωγή του Κεφαλαίου 1 το μεγαλύτερο ποσοστών των χρηστών δεν παρακολουθεί το βίντεο σειριακά από την αρχή μέχρι το τέλος. Αντί αυτού οι χρήστες χρησιμοποιούν όλους τους βαθμούς ελευθερίας κινήσεων (forward, backward, pause) που τους δίνονται από τα προγράμματα αναπαραγωγής βίντεο. Στο Κεφάλαιο αυτό μελετάμε την απόδοση αλγορίθμων, δίνοντας την ευχέρεια στους χρήστες να μπορούν να παρακολουθήσουν ολόκληρο το βίντεο σειριακά ή να παρακολουθήσουν μόνο ένα αρχικό τμήμα του. Δηλαδή δίνεται μια ελευθερία κινήσεων στους χρήστες, η παύση. Σύμφωνα το [21], η παύση είναι η πιο συχνή κίνηση χρηστών που χρησιμοποιείται στα διάφορα προγράμματα αναπαραγωγής βίντεο. Στο Κεφάλαιο αυτό υλοποιήθηκαν τρία διαφορετικά σενάρια για μερική παρακολούθηση των βίντεο, μέσα από τα οποία φαίνεται η απόδοση κάθε εξεταζόμενου αλγορίθμου σε διαφορετικές συνθήκες και μέσα από τα σενάρια αυτά προκύπτουν ενδιαφέροντα συμπεράσματα τα οποία παρουσιάζονται στο τέλος του Κεφαλαίου. Το πρώτο σενάριο βασίζεται σε ποσοστά παρακολούθησης των βίντεο (πλήρη ή μερική παρακολούθηση) στηριζόμενα στο [8]. Θεωρείται το πιο σημαντικό σενάριο καθώς αρκετές έρευνες [4], [11], [12], συμφωνούν ότι οι περισσότεροι χρήστες που παρακολουθούν μερικώς το βίντεο σταματάνε νωρίς (δηλαδή στα αρχικά frames). Το δεύτερο σενάριο αναφέρεται σε μια μέση κατάσταση και στηρίζεται σε ποσοστά παρακολούθησης των βίντεο από το [3]. Τέλος το τρίτο σενάριο το οποίο στηρίζεται σε ποσοστά παρακολούθησης των βίντεο από το [4], θεωρείται μια ακραία περίπτωση για μερική παρακολούθηση των βίντεο (partial video viewing) η οποία πιθανόν να ισχύει σε ένα περιβάλλον στο οποίο οι χρήστες παρακολουθούν online διαλέξεις. Με βάση τα αποτελέσματα των προσομοιώσεων συμπεραίνουμε ότι έχουμε πετύχει σημαντική βελτίωση σε ότι αφορά τις παρακάτω μετρικές απόδοσης: (1) Byte-Hit Ratio, (2) καθυστερήσεις κατά το ξεκίνημα των βίντεο και (3) overhead του συστήματος, σε σύγκριση με τον αλγόριθμο που προτάθηκε στο [3].

3.2 Περιγραφή Συστήματος

Η τοπολογία του συστήματος υποτίθεται ίδια με το Κεφάλαιο 2. Η περιγραφή δίνεται στο 2.2.1.

3.2.1 Μέθοδοι αποθήκευσης και αντικατάστασης

Στις επόμενες ενότητες περιγράφεται η λειτουργία τριών αλγορίθμων. Αρχικά παρουσιάζεται ο αλγόριθμος από το [3], *"Lazy Segmentation"*, ο οποίος στη συνέχεια συγκρίνεται με δικούς μας προτεινόμενους αλγορίθμους.

Προτείνεται και υλοποιείται αλγόριθμος τμηματοποίησης των βίντεο τον οποίο ονομάζουμε, *"Aggressive and Fixed Segmentation".* Ο αλγόριθμος αυτός χωρίζει την συνολική μνήμη του proxy server σε δύο τμήματα, στη μνήμη Α και στη μνήμη Β. Στη μελέτη μας η μνήμη Α προορίζεται για τα προθέματα των βίντεο και η μνήμη Β προορίζεται για το υπόλοιπο τμήμα των βίντεο, το οποίο τμηματοποιείται σε λίγα τμήματα (segments).

Επιπλέον προτείνεται και υλοποιείται αλγόριθμος διαχείρισης μνήμης που ονομάζεται, *"Aggressive and guarded method"* σύμφωνα με τον οποίο η συνολική μνήμη χωρίζεται σε τρία τμήματα, τα οποία θα αναφέρονται ως μνήμες Α, Β και C. Στη μελέτη μας η μνήμη Α αποθηκεύει τα προθέματα των βίντεο και οι μνήμες Β και C το υπόλοιπο περιεχόμενο των βίντεο.

Τέλος χρησιμοποιείται ο καλύτερος αλγόριθμος από το Κεφάλαιο 2, *"Full video with prefix and with limitation"*, για να ελέγξουμε την αποδοτικότητά του και να συγκρίνουμε με τους παραπάνω τρείς αλγορίθμους στα διάφορα σενάρια για μερική παρακολούθηση των βίντεο.

3.2.2 Μέθοδοι διαχείρισης μνήμης και στρατηγικές εισαγωγής και αντικατάστασης

3.2.2.1 Lazy Segmentation

Κάθε βίντεο το οποίο προσπελαύνεται για 1^{η} φορά αποθηκεύεται ολόκληρο στη μνήμη του proxy server (admission policy-3.2.2.1.1). Το βίντεο παραμένει στη μνήμη μέχρι να επιλεγεί σαν «θύμα» και να αφαιρεθεί (replacement policy-3.2.2.1.3). Εφόσον επιλεγεί ότι αυτό είναι το βίντεο το οποίο θα αφαιρεθεί, χωρίζεται με βάση τη στρατηγική τμηματοποίησης του βίντεο «θύματος»

(3.2.2.1.2) και κάποια από τα τμήματά του αφαιρούνται κατά τη 1η φάση του αλγορίθμου και κάποια κατά τη 2η φάση του αλγορίθμου (η επαναληπτική στρατηγική αντικατάστασης δύο φάσεων περιγράφεται στην ενότητα 3.2.2.1.3).

Για κάθε βίντεο που αποθηκεύεται στη μνήμη του proxy server δημιουργείται ένα αρχείο καταχώρησης (access log) κάποιων σημαντικών παραμέτρων έτσι ώστε να μπορούμε να γνωρίζουμε τη δημοτικότητα του κάθε βίντεο. Οι παράμετροι αυτοί είναι ο χρόνος πρώτης προσπέλασης του βίντεο (T1), ο χρόνος τελευταίας αναφοράς του βίντεο (Tr), ο οποίος είναι ίσος με το T1 όταν το βίντεο προσπελαύνεται για πρώτη φορά, ο τρέχον χρόνος (Tc), το άθροισμα της συνολικής διάρκειας παρακολούθησης του βίντεο από όλες τις αιτήσεις (Lsum), ο αριθμός συνολικών αφίξεων του βίντεο στο σύστημα (n), το μέγεθος του βασικού τμήματος (Lb) και ο αριθμός συνολικών τμημάτων που έχει αποθηκεύσει το βίντεο στη μνήμη του proxy server (ns). Οι μεταβλητές Tr, n, ns ανανεώνονται σε κάθε νέα άφιξη αίτησης για το βίντεο. Η μεταβλητή Lsum ανανεώνεται εφόσον ολοκληρωθεί η παρακολούθηση του βίντεο και η μεταβλητή Lb υπολογίζεται κατά τη διάρκεια τμηματοποίησης του βίντεο. Από το πιο πάνω αρχείο καταχώρησης παραμέτρων μπορεί να υπολογιστεί η συχνότητα άφιξης αιτήσεων για το βίντεο η οποία ορίζεται ως F=$\frac{n}{Tr-T1}$ καθώς επίσης και η μέση διάρκεια παρακολούθησης η οποία ορίζεται ως Lavg=$\frac{Lsum}{n}$.

3.2.2.1.1 Επιθετική στρατηγική εισαγωγής (Aggressive Admission Policy) της "Lazy Segmentation"

Με βάση τη στρατηγική εισαγωγής ο αλγόριθμος αποφασίζει για κάθε βίντεο που προσπελαύνεται σε ποιά από τις τρείς παρακάτω περιπτώσεις ανήκει.

- Αν δεν υπάρχει αρχείο καταχώρησης για το συγκεκριμένο βίντεο, σημαίνει ότι το βίντεο προσπελαύνεται για 1η φορά και έτσι μπορεί να αποθηκεύσει ολόκληρο το περιεχόμενό του στη μνήμη του proxy server. Εφόσον ο proxy server γνωρίζει το μέγεθος του βίντεο που πρέπει να αποθηκευτεί, με βάση τον adaptive replacement algorithm (3.2.2.1.3) δεσμεύεται η απαιτούμενη μνήμη (αν δεν ήταν ήδη διαθέσιμη). Ακολούθως, το βίντεο αποθηκεύεται ολόκληρο ανεξάρτητα από τη διάρκεια παρακολούθησης του χρήστη. Τέλος, δημιουργείται το αρχείο καταγραφής παραμέτρων για το συγκεκριμένο βίντεο στο οποίο θα αποθηκεύονται οι αλλαγές στις διάφορες παραμέτρους σε κάθε καινούργια αίτησή του.

- Αν υπάρχει αρχείο καταχώρησης παραμέτρων για το συγκεκριμένο βίντεο και το βίντεο είναι ολόκληρο αποθηκευμένο στην μνήμη του proxy server, τότε απλά ενημερώνεται το αρχείο καταχώρησης παραμέτρων.

- Αν υπάρχει αρχείο καταχώρησης παραμέτρων για το συγκεκριμένο βίντεο και το βίντεο βρίσκεται εν μέρη αποθηκευμένο στη μνήμη του proxy server, τότε ο αλγόριθμος εξετάζει αν μπορεί να αποθηκευτεί το επόμενο (ns+1) τμήμα του βίντεο. Αν ισχύει ότι Lavg>= $\frac{1}{a}$*(ns+1)*Lb τότε η replacement policy (3.2.2.1.3) θα προσπαθήσει να απελευθερώσει την απαραίτητη μνήμη για την αποθήκευση του επόμενου τμήματος αν δεν ήταν ήδη διαθέσιμη. Σε αντίθετη περίπτωση που δεν ισχύει ο πιο πάνω περιορισμός, απορρίπτεται η προσπάθεια για αποθήκευση του επόμενου τμήματος. Η παράμετρος α καθορίζεται κατά τη διάρκεια τμηματοποίησης του βίντεο.

3.2.2.1.2 Στρατηγική τμηματοποίησης βίντεο «θύματος» της Lazy Segmentation

Αν δεν υπάρχει διαθέσιμη ελεύθερη μνήμη και απαιτείται να ελευθερωθεί, τότε η replacement policy (3.2.2.1.3) υπολογίζει τα caching values όλων των βίντεο που είναι ήδη αποθηκευμένα στη μνήμη του proxy server. Το βίντεο με τη μικρότερη τιμή επιλέγεται ως «θύμα». Αν το βίντεο «θύμα» είναι ολόκληρο αποθηκευμένο τότε τμηματοποιείται με βάση την μέση διάρκεια πρόσβασης *L*avg για τον τρέχοντα χρόνο. Ορίζεται το βασικό τμήμα Lb=Lavg και ακολούθως το βίντεο τμηματοποιείται ομοιόμορφα με βάση το Lb (εκτός από το τελευταίο τμήμα). Στη συνέχεια τα πρώτα α τμήματα παραμένουν στη μνήμη και τα υπόλοιπα αφαιρούνται με βάση τις φάσεις της στρατηγικής αντικατάστασης που περιγράφονται στην πιο κάτω ενότητα 3.2.2.1.3. Τέλος ανανεώνεται ο αριθμός των αποθηκευμένων τμημάτων και το αρχείο καταγραφής παραμέτρων για το βίντεο θύμα. Αν μια μεταγενέστερη αίτηση απαιτεί περισσότερα τμήματα από τα ήδη αποθηκευμένα τότε θα προσκομιστεί από τον content server πληροφορία συνολικού μεγέθους Lb.

3.2.2.1.3 Επαναληπτική στρατηγική αντικατάστασης δύο φάσεων (Two-Phase Iterative Replacement Policy) της Lazy Segmentation

Χρησιμοποιείται η replacement policy για να επιλεγούν τα βίντεο «θύματα» που θα αφαιρεθούν από τη μνήμη του proxy server όταν απαιτείται αποδέσμευση μνήμης. Αρχικά υπολογίζονται κάποιες σημαντικοί παράμετροι (μέσος αριθμός πρόσβασης, μέση διάρκεια πρόσβασης, μέγεθος αποθηκευμένης πληροφορίας, πιθανότητα μελλοντικής ζήτησης), που θα βοηθήσουν στη επιλογή του βίντεο «θύματος». Μετά υπολογίζονται τα caching values όλων των βίντεο που βρίσκονται αποθηκευμένα στη μνήμη του proxy server και τα οποία δεν παρακολουθούνται από κάποιο χρήστη. Ακολούθως το βίντεο με τη μικρότερη τιμή επιλέγεται ως

το βίντεο «θύμα». Τμήματα αυτού του βίντεο αφαιρούνται με βάση τις δύο φάσεις του αλγορίθμου που περιγράφονται πιο κάτω. Το caching value κάθε βίντεο ορίζεται ίσο με: $\frac{\frac{Lsum}{Tr-T1}*MIN\left\{1,\frac{\frac{Tr-T1}{n}}{Tc-Tr}\right\}}{ns*Lb}$

Στη 1η φάση του αλγορίθμου: Αν το αρχείο καταγραφής του βίντεο υποδηλώνει ότι το συγκεκριμένο βίντεο είναι ολόκληρο αποθηκευμένο, τότε το βίντεο τμηματοποιείται με βάση την ενότητα 3.2.2.1.2. Τα πρώτα α (εδώ έχουμε υποθέσει α=2 όπως και στο [3]) τμήματα παραμένουν στη μνήμη και τα υπόλοιπα αφαιρούνται αμέσως μετά την ολοκλήρωση της τμηματοποίησης.

Στη 2η φάση του αλγορίθμου: Αν το βίντεο είναι μερικώς αποθηκευμένο τότε αφαιρείται το τελευταίο τμήμα του.

Το caching value κάθε βίντεο θύματος ανανεώνεται στο τέλος κάθε αντικατάστασης και η διαδικασία επαναλαμβάνεται μέχρι να αποδεσμευτεί η απαραίτητη μνήμη. Αξίζει να σημειωθεί ότι αν το βίντεο θύμα αντικατασταθεί πλήρως, το σύστημα εξακολουθεί να κρατά το αρχείο καταχώρησής του. Αν δεν το έκανε, τότε το βίντεο θα αποθηκευόταν ξανά ολόκληρο σε μια μεταγενέστερη αίτηση το οποίο θα ήταν αναποτελεσματικό για το σύστημα. Το σύστημα επιτρέπει στο βίντεο το οποίο αντικαταστήθηκε πλήρως να αποθηκευτεί ξανά ολόκληρο στη μνήμη, αν οι συνολικές αιτήσεις του είναι μεγαλύτερες ή ίσες από το μέσο αριθμό αιτήσεων από όλα τα βίντεο που βρίσκονται στη μνήμη του proxy server.

3.2.2.2 Aggressive and Fixed Segmentation

Ακολουθεί η περιγραφή του προτεινόμενου αλγόριθμου τμηματοποίησης *"Aggressive and Fixed Segmentation",* στον οποίο κάθε βίντεο τμηματοποιείται ομοιόμορφα με βάση ένα σταθερό μέγεθος το οποίο ορίστηκε ίσο με 500 μπλοκ (διάρκεια playback ίση με 15 λεπτά). Στη μελέτη μας η μνήμη του proxy server χωρίζεται σε δύο μέρη, τη μνήμη Α και τη μνήμη Β. Η μνήμη Α προορίζεται για τα αρχικά τμήματα των βίντεο τα οποία θα αναφέρονται ως προθέματα (prefixes) και το μέγεθός της αντιστοιχεί στο 10% της συνολικής χωρητικότητας της μνήμης. Η μνήμη Β προορίζεται για τα υπόλοιπα τμήματα των βίντεο και το μέγεθός της αντιστοιχεί στο 90% της συνολικής χωρητικότητας της μνήμης. Σε αυτό το Κεφάλαιο ο χρήστης πιθανόν να παρακολουθήσει μερικώς το βίντεο και ως επακόλουθο η αργή εισαγωγή τμημάτων στη μνήμη Β μπορεί να είναι αποδοτική για το σύστημα. Στις επόμενες ενότητες 3.2.2.2.1 και 3.2.2.2.2 περιγράφονται οι στρατηγικές εισαγωγής και αντικατάστασης για τις μνήμες Α και Β αντίστοιχα, που εφαρμόζονται στον προτεινόμενο αλγόριθμο διαχείρισης μνήμης *"Aggressive and Fixed Segmentation".*

3.2.2.2.1 Στρατηγική εισαγωγής και αντικατάστασης στη Μνήμη Α

Λειτουργεί με ίδιες στρατηγικές όπως και στο Κεφάλαιο 2. Η περιγραφή δίνεται στο 2.2.3.1.

3.2.2.2.2 Στρατηγική εισαγωγής και αντικατάστασης στη Μνήμη Β

Η στρατηγική στη συγκεκριμένη μνήμη στηρίζεται σε ιδέες από τη μέθοδο τμηματοποίησης από το [3], την *"Lazy Segmentation"*, με πιο επιφυλακτική προσέγγιση. Στη μέθοδο *"Lazy Segmentation"* σε κάθε νέα αίτηση το βίντεο είναι πάντα επιλέξιμο να αποθηκεύσει ολόκληρο το περιεχόμενό του από την πρώτη αίτηση. Στη δική μας περίπτωση κάθε καινούργιο βίντεο είναι πάντα επιλέξιμο να αποθηκεύσει μόνο το πρώτο τμήμα του στη μνήμη Β. Για τα υπόλοιπα τμήματα του βίντεο η αποθήκευση δεν υλοποιείται πάντα, αντί αυτού υπάρχουν περιορισμοί. Το μέγεθος του τμήματος υπολογίζεται με βάση το μέγεθος του suffix (ολόκληρο το βίντεο εκτός από το prefix) τμηματοποιημένο σε τμήματα των 500 blocks (εκτός από το τελευταίο τμήμα). Ως επακόλουθο τα μικρά βίντεο που έχουν μέγεθος μικρότερο ή ίσο από 500 blocks μπορούν να αποθηκεύσουν ολόκληρο το περιεχόμενό τους από την πρώτη αίτηση.

3.2.2.2.2.1 Στρατηγική εισαγωγής στη Μνήμη Β

Εάν το βίντεο δεν είναι ολόκληρο αποθηκευμένο, τότε το σύστημα εξετάζει αν μπορεί να αποθηκεύσει το επόμενο (ns+1) τμήμα του, όπου ns είναι ο αριθμός των τμημάτων που έχει αποθηκεύσει το βίντεο στη μνήμη Β. Εάν η μνήμη Β διαθέτει τον απαραίτητο χώρο, η αποθήκευση γίνεται απευθείας χωρίς κάποιο έλεγχο για όλα τα τμήματα. Διαφορετικά όταν δεν υπάρχει επαρκής χώρος μνήμης υλοποιείται η στρατηγική αντικατάστασης στην ενότητα 3.2.2.2.2.2 για να εξεταστεί η περαιτέρω αποθήκευση του βίντεο.

3.2.2.2.2.2 Στρατηγική αντικατάστασης στη Μνήμη Β

Η στρατηγική αντικατάστασης υλοποιείται όταν απαιτείται διαθέσιμη μνήμη και δεν υπάρχει. Χρησιμοποιείται διαφορετική στρατηγική όταν πρόκειται να αποθηκευτεί το πρώτο τμήμα του βίντεο από αυτή που χρησιμοποιείται για τα υπόλοιπα τμήματα.

Όταν πρόκειται για την αποθήκευση του πρώτου τμήματος εφαρμόζεται η πιο κάτω «επιθετική» στρατηγική για να αφαιρεθούν τα απαραίτητα βίντεο θύματα. Αρχικά, το σύστημα υπολογίζει τα caching values όλων των βίντεο στη μνήμη B τα οποία δεν παρακολουθούνται από κάποιο χρήστη και το βίντεο με τη μικρότερη τιμή επιλέγεται ως το βίντεο θύμα. Στη συνέχεια αφαιρείται το τελευταίο αποθηκευμένο τμήμα του βίντεο θύματος και η διαδικασία επαναλαμβάνεται μέχρι να αποδεσμευτεί ο απαραίτητος χώρος για να αποθηκευτεί το πρώτο τμήμα του καινούργιου βίντεο. Επειδή η παρακολούθηση των βίντεο γίνεται σειριακά από την αρχή, τα τελευταία τμήματα είναι λιγότερο πιθανά να προσπελαστούν, έτσι σε κάθε βίντεο θύμα πάντα θα γίνεται αφαίρεση του τελευταίου τμήματος που έχει αποθηκευτεί στη μνήμη B. Το caching value είναι μοναδικό για κάθε βίντεο και υποτίθεται ίδιο με αυτό στο Κεφάλαιο 2 όπου ορίστηκε ίσο με $\frac{n*ns}{Tc-Tr}$, όπου n είναι ο αριθμός συνολικών αιτήσεων του βίντεο, ns ο αριθμός των blocks που έχει αποθηκεύσει στη μνήμη (A και B), Tc ο τρέχον χρόνος και Tr ο χρόνος τελευταίας αναφοράς για το συγκεκριμένο βίντεο.

Όταν πρόκειται για την αποθήκευση των υπόλοιπων τμημάτων (δεν πρόκειται δηλαδή για το πρώτο τμήμα), ακολουθείται διαφορετική διαδικασία ώστε να αποφασίσει το σύστημα αν θα επιτρέψει την αφαίρεση θυμάτων για την αποθήκευση του επόμενου τμήματος ή όχι από το βίντεο που έκανε την αίτηση. Όταν βρεθεί το βίντεο θύμα με το μικρότερο caching value τότε συγκρίνεται επιπλέον αν ο χρόνος τελευταίας αναφοράς (Tr) του υποψήφιου βίντεο θύματος είναι μικρότερος από τον χρόνο τελευταίας αναφοράς του βίντεο που έκανε την αίτηση. Αν ισχύει ότι το υποψήφιο βίντεο θύμα έχει μικρότερο χρόνο τελευταίας αναφοράς, θα ελέγχεται επιπρόσθετα αν το βίντεο που έκανε την αίτηση έχει συνολικό αριθμό αιτήσεων μεγαλύτερο ή ίσο από το μέσο αριθμό αιτήσεων από όλα τα βίντεο στη μνήμη (A και B). Στην περίπτωση που ισχύουν και οι δύο πιο πάνω προϋποθέσεις, θα αφαιρείται το τελευταίο αποθηκευμένο τμήμα από το βίντεο θύμα και η διαδικασία θα επαναλαμβάνεται μέχρι να αποδεσμευτεί η απαραίτητη μνήμη. Η αφαίρεση θυμάτων θα επιτρέπεται μόνο αν μπορούν να βρεθούν αρκετά υποψήφια βίντεο θύματα που ικανοποιούν τον περιορισμό, αν μετά την αφαίρεσή τους θα αποδεσμευτεί η απαιτούμενη μνήμη και αν επιπλέον το βίντεο που έκανε την αίτηση έχει αριθμό αφίξεων (n) μεγαλύτερο ή ίσο από τον μέσο αριθμό αφίξεων όλων των βίντεο στη μνήμη. Αν κάποιος από τους παραπάνω περιορισμούς δεν ικανοποιείται, τότε δεν υλοποιείται καμία αφαίρεση τμήματος από τα βίντεο θύματα. Οι έλεγχοι αυτοί αποσκοπούν να αποθηκεύονται στη μνήμη B του proxy server αρκετά τμήματα μόνο από τα πιο δημοφιλή βίντεο.

Σε κάθε περίπτωση, κάθε φορά που αφαιρείται ολόκληρο το περιεχόμενο του βίντεο θύματος από τη μνήμη B, τότε το σύστημα θα εξετάζει αν υπάρχει το πρόθεμα του βίντεο στη μνήμη A. Αν δεν υπάρχει, τότε το βίντεο έχει αφαιρεθεί εξ' ολοκλήρου και θα διαγράφεται όλη η διαθέσιμη για αυτό πληροφορία.

3.2.2.3 Aggressive and guarded method

Σε αυτή την ενότητα περιγράφεται ο αλγόριθμος της επιθετικής και συγχρόνως επιφυλακτικής μεθόδου που προτείνουμε, *"Aggressive and guarded method"*. Στον αλγόριθμο αυτό η συνολική διαθέσιμη μνήμη του proxy server χωρίζεται σε τρία τμήματα και το κάθε τμήμα λειτουργεί σύμφωνα με τις δικές του στρατηγικές εισαγωγής και αντικατάστασης. Η μνήμη A αποτελεί το αρχικό τμήμα της συνολικής μνήμης και προορίζεται για την αποθήκευση των πρώτων Bp μπλοκ του βίντεο που έκανε την αίτηση. Τα μπλοκ αυτά αντιστοιχούν στο πρόθεμα του βίντεο και τα διαχειριζόμαστε ως ένα μικρό και ενιαίο τμήμα. Ως δεύτερο τμήμα της συνολικής μνήμης ορίζεται η μνήμη B και ως τρίτο τμήμα η μνήμη C, οι οποίες προορίζονται για την αποθήκευση του υπόλοιπου τμήματος του βίντεο. Η κύρια διαφοροποίηση μεταξύ της μνήμης B και C, είναι ότι στη μνήμη B εισάγονται τμήματα των βίντεο με αργό και προσεκτικό ρυθμό σε αντίθεση με τη μνήμη C η οποία υλοποιεί πιο επιθετική στρατηγική και μπορεί να αποθηκεύει ολόκληρο το βίντεο (εκτός από το πρόθεμα το οποίο αποθηκεύεται στη μνήμη A) σε μια αίτηση. Δηλαδή η μνήμη B λειτουργεί με μεθόδους τμηματοποίησης των βίντεο και η μνήμη C λειτουργεί παρόμοια με τη μνήμη B στο Κεφάλαιο 2 χωρίς να γίνεται κάποια τμηματοποίηση, αλλά χειρίζεται το suffix τμήμα του βίντεο σαν ένα μεγάλο ενιαίο τμήμα.

Κατά την αρχικοποίηση του συστήματος κάθε βίντεο που υπάρχει στο σύστημα τμηματοποιείται ομοιόμορφα σε 20 τμήματα ίσου μεγέθους και το τελευταίο τμήμα θα έχει μέγεθος ίσο με το μέγεθος του βίντεο αφαιρώντας το μέγεθος των 20 τμημάτων και το μέγεθος του προθέματος. Εφόσον το σύστημά μας αποτελείται από βίντεο μικρής, μεσαίας και μεγάλης διάρκειας είναι πιο αποτελεσματικό να επιλέγεται το μέγεθος κάθε τμήματος ανάλογα με το συνολικό μέγεθος κάθε βίντεο και όχι από ένα σταθερό αριθμό. Στον Πίνακα 3.1 παρουσιάζεται πως υλοποιείται η τμηματοποίηση για ένα βίντεο (α) μικρής, (β) μεσαίας και (γ) μεγάλης διάρκειας. Τα τμήματα αυτά μπορούν να χρησιμοποιηθούν μόνο από τη μνήμη B γιατί όπως ήδη αναφέρθηκε, αυτή είναι η μοναδική μνήμη που χρησιμοποιεί την τμηματοποίηση (segmentation) στα βίντεο. Η επιλογή να χωρίζονται τα βίντεο σε 20 ή 21 τμήματα επιλέχθηκε μετά από δοκιμές. Για τμήματα λιγότερα από 20 τα αποτελέσματα ήταν χειρότερα και για τμήματα περισσότερα από 20 ήταν παρόμοια.

Size in blocks	**Prefix**	**1-st Segment**	**…**								**20-st Segment**	**Last segment**
(α) 252	32	11	11	11	11	11	11	11	11	11	11	0
(β) 1050	32	50	50	50	50	50	50	50	50	50	50	18
(γ) 3555	32	176	176	176	176	176	176	176	176	176	176	3

Πίνακας 3.1: Παράδειγμα τμηματοποίησης βίντεο (α) μικρής διάρκειας 252 μπλοκ (β) μεσαίας διάρκειας 1050 μπλοκ και (γ) μεγάλης διάρκειας 3555 μπλοκ

Αρχικά σε κάθε καινούργια αίτηση το βίντεο προσπαθεί να αποθηκεύσει το πρόθεμά του στη μνήμη A (αν δεν ήταν ήδη αποθηκευμένο) και ακολούθως εξετάζεται ποιά από τις 4 πιο κάτω περιπτώσεις ισχύει, ώστε να παρθεί η απόφαση αν πρέπει να αποθηκευτεί το βίντεο στη μνήμη B ή στη μνήμη C ή αν είναι ήδη αποθηκευμένο σε κάποια από τις δύο μνήμες. Οι περιπτώσεις που διακρίνονται φαίνονται πιο κάτω:

- Το βίντεο έχει αποθηκεύσει ολόκληρο το περιεχόμενό του (χωρίς να λαμβάνεται υπόψη το πρόθεμα το οποίο διαχειρίζεται η μνήμη A) στη μνήμη C και ανανεώνει τις μεταβλητές του. Δεν επιτρέπεται να αποθηκεύσει κάποιο τμήμα του στη μνήμη B.

- Το βίντεο έχει αποθηκεύσει ολόκληρο το περιεχόμενό του (χωρίς να λαμβάνεται υπόψη το πρόθεμα το οποίο διαχειρίζεται η μνήμη A) στη μνήμη B και ανανεώνει τις μεταβλητές του. Δεν επιτρέπεται να αποθηκεύσει κάποιο τμήμα του στη μνήμη C.

- Το βίντεο είναι μερικώς αποθηκευμένο στη μνήμη B, χωρίς να έχει εισάγει ολόκληρο το περιεχόμενό του και θα προσπαθήσει να αποθηκεύσει το επόμενο τμήμα του στη μνήμη B.

- Το βίντεο δε βρίσκεται στη μνήμη B ή στη μνήμη C. Στην περίπτωση στην οποία η μνήμη C έχει τον απαραίτητο διαθέσιμο χώρο τότε αποθηκεύεται σε αυτή ολόκληρο το περιεχόμενο του suffix χωρίς να απαιτείται κάποιος έλεγχος. Αντιθέτως, αν δεν υπάρχει επαρκής χώρος στη μνήμη C, τότε θα εξετάζεται αν το μέγεθος τμήματος του βίντεο είναι μεγαλύτερο ή ίσο από 100 blocks. Αν ο συγκεκριμένος έλεγχος ισχύει, τότε το βίντεο θα δοκιμάσει να αποθηκεύσει ολόκληρο το suffix τμήμα του στη μνήμη C και αν αποτύχει θα δοκιμάσει να αποθηκεύσει το πρώτο τμήμα του στη μνήμη B. Αλλιώς αν το μέγεθος τμήματος είναι μικρότερο από 100 blocks, θα δοκιμάσει αρχικά στη μνήμη B και αν αποτύχει θα δοκιμάσει στη μνήμη C. Κατά συνέπεια, όταν γεμίσει η μνήμη C επιτρέπεται μόνο σε βίντεο μεγάλης διάρκειας να δοκιμάσουν να αποθηκευτούν σε αυτή. Η επιλογή των 100 blocks έχει γίνει

μετά από δοκιμές. Ως καλύτερο βρέθηκε να επιτρέπουμε σε μεγάλα βίντεο, διάρκειας μεγαλύτερης από 60 λεπτά, να αποθηκεύονται στη μνήμη C όταν αυτή γεμίσει.

Όπως φαίνεται από τις τέσσερις πιο πάνω περιπτώσεις η λειτουργία της μνήμης B δεν παρεμβάλει στην λειτουργία της μνήμης C, και το αντίστροφο. Αν ο αλγόριθμος ορίσει ότι το βίντεο θα αποθηκευτεί στη μνήμη B, θα προσπαθεί σε μελλοντικές αιτήσεις να εισάγει ολόκληρο το περιεχόμενό του στη συγκεκριμένη μνήμη. Μόνο αν μετά από την εφαρμογή του αλγόριθμου αντικατάστασης στη μνήμη B αφαιρεθούν όλα τα τμήματά του από τη συγκεκριμένη μνήμη, δίνεται η ευκαιρία στο συγκεκριμένο βίντεο στην επόμενη αίτησή του να αποθηκευτεί στη μνήμη B ή στη C, ανάλογα με την τρέχουσα κατάσταση που υπάρχει στο σύστημα. Αντίστοιχα αν κάποιο βίντεο αποθηκευτεί στη μνήμη C, δεν παρεμβάλει στην λειτουργία της μνήμης B. Οι στρατηγικές εισαγωγής και αντικατάστασης που χρησιμοποιούνται από τις μνήμες A, B και C, στην “Aggressive and guarded method”, περιγράφονται στις επόμενες ενότητες.

3.2.2.3.1 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη A

Λειτουργεί όμοια με τη μνήμη A στο Κεφάλαιο 2. Η περιγραφή έγινε στην ενότητα 2.2.3.1. Η μόνη διαφορά είναι στον έλεγχο που γίνεται για να βρεθεί αν το βίντεο θύμα έχει αφαιρεθεί εξ’ ολοκλήρου από τη μνήμη. Σε αυτή την περίπτωση όταν αφαιρείται το πρόθεμα από το βίντεο θύμα, θα ελέγχεται αν υπάρχουν blocks αποθηκευμένα στις μνήμες B ή C. Αν το βίντεο δε βρίσκεται σε κάποια από τις δύο μνήμες, τότε σημαίνει πως έχει αφαιρεθεί εξ’ ολοκλήρου και διαγράφεται όλη η πληροφορία που είχαμε για το συγκεκριμένο βίντεο. Αν όμως το βίντεο θύμα έχει αποθηκεύσει κάποιο από το περιεχόμενό του στη μνήμη B ή στη μνήμη C, τότε εξακολουθεί να διατηρείται το caching value και να ανανεώνεται σε κάθε νέα αίτηση που αναφέρεται σε αυτό.

3.2.2.3.2 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη B

Η μνήμη B αντιστοιχεί σε ποσοστό 95% της συνολικής αρχικής μνήμης εφόσον αφαιρεθεί η μνήμη που είναι διαθέσιμη για τη μνήμη A **(**Cache B=(Total Cache -(Total Cache*0.1))*0.95**)**. Σε κάθε αίτηση το βίντεο μπορεί να αποθηκεύει μόνο ένα επιπλέον τμήμα του, το μέγεθος του οποίου έχει οριστεί κατά την αρχικοποίηση, όπως εξηγήθηκε προηγουμένως στην ενότητα 3.2.2.3.

3.2.2.3.2.1 Στρατηγική εισαγωγής στη μνήμη B

Όπως αναφέρθηκε ένα βίντεο είναι πάντα επιλέξιμο να αποθηκεύσει το πρώτο τμήμα του στη μνήμη Β επιλέγοντας κάποια βίντεο θύματα αν απαιτούνται με βάση τη στρατηγική αντικατάστασης η οποία περιγράφεται στην επόμενη ενότητα. Όταν το βίντεο προσπαθεί να αποθηκεύσει κάποιο τμήμα του στη μνήμη Β και δεν πρόκειται για το πρώτο τμήμα του, τότε η στρατηγική εισαγωγής διαφοροποιείται. Στην περίπτωση που υπάρχει επαρκής χώρος μνήμης, η αποθήκευση υλοποιείται χωρίς έλεγχο (για όλα τα τμήματα) όμως αν δεν υπάρχει επαρκής μνήμη, θα εφαρμόζεται η στρατηγική αντικατάστασης στην ενότητα 3.2.2.3.2.2 ώστε να εξεταστεί αν το βίντεο πληροί τις προδιαγραφές για περαιτέρω αποθήκευση. Όταν το βίντεο βρίσκεται ολόκληρο αποθηκευμένο στη μνήμη Β τότε απλά θα ανανεώνονται οι μεταβλητές του.

3.2.2.3.2.2 Στρατηγική αντικατάστασης στη μνήμη Β

Όταν απαιτείται αποδέσμευση μνήμης η στρατηγική αντικατάστασης προσπαθεί να βρει τα βίντεο θύματα ώστε να αποδεσμευτεί μνήμη ικανή για την αποθήκευση του επόμενου τμήματος του βίντεο που έκανε την αίτηση. Για την επιλογή του βίντεο θύματος αρχικά θα εξετάζονται όλα τα βίντεο στη μνήμη Β τα οποία δεν παρακολουθούνται από κάποιο χρήστη και τα οποία έχουν ήδη αποθηκεύσει στη μνήμη Β περισσότερα ή ίσα από 3 τμήματα. Ο επιπλέον έλεγχος για τον αριθμό τμημάτων που έχει το βίντεο στη μνήμη Β αποσκοπεί στην επιλογή θυμάτων με αρκετά τμήματα στη μνήμη, ώστε μετά την αφαίρεση κάποιου τμήματος να αποτρέπεται η ολική διαγραφή του βίντεο θύματος. Δεδομένου ότι το μεγαλύτερο ποσοστό των χρηστών δεν παρακολουθεί ολόκληρο το βίντεο αλλά μόνο ένα αρχικό μέρος του, η στρατηγική στη μνήμη Β έχει σκοπό να κρατά τα αρχικά τμήματα των βίντεο για όσο μεγαλύτερο χρονικό διάστημα αυτό είναι εφικτό. Εφόσον βρεθούν υποψήφια βίντεο θύματα που πληρούν τις πιο πάνω προϋποθέσεις, ως θύμα θα επιλέγεται αυτό με το μικρότερο caching value το οποίο για κάθε βίντεο είναι ίσο με $\frac{n*ns}{Tc-Tr}$. Εάν δε μπορεί να βρεθεί κάποιο βίντεο θύμα με περισσότερα ή ίσα από 3 τμήματα στη μνήμη Β και το οποίο δεν παρακολουθείται από κάποιο χρήστη, τότε επιτρέπουμε να επιλέγονται θύματα με ένα ή δύο τμήματα στη μνήμη Β. Αντίστοιχα από τα συγκεκριμένα υποψήφια θύματα, ως θύμα θα είναι το βίντεο με το μικρότερο caching value. Ακολούθως εφαρμόζεται η ίδια στρατηγική με αυτή στην ενότητα 3.2.2.2.2.2.

3.2.2.3.3 Στρατηγική εισαγωγής και αντικατάστασης στη μνήμη C

Η συγκεκριμένη μνήμη έχει μέγεθος ίσο με την συνολική μνήμη του proxy server αφαιρώντας το μέγεθος το οποίο δεσμεύτηκε για τη μνήμη Α και για τη μνήμη Β. Στη μνήμη C δεν

υλοποιείται κάποια τμηματοποίηση των βίντεο. Η λειτουργία της μνήμη C βασίζεται σε ιδέες από το Κεφάλαιο 2 και αποτελεί το «επιθετικό» κομμάτι του αλγορίθμου. Η μόνη περίπτωση για να δοθεί σε κάποιο βίντεο η ευκαιρία να αποθηκευτεί στη μνήμη C είναι να μην έχει αποθηκευτεί ακόμα στη μνήμη B ή C και η μνήμη C να έχει διαθέσιμο χώρο για την αποθήκευσή του ή το μέγεθος του τμήματός του να είναι μεγαλύτερο από 100 blocks οπότε ο αλγόριθμος διαχείρισης θα προσπαθήσει να βρει βίντεο θύματα για να απελευθερωθεί η απαραίτητη μνήμη.

3.2.2.3.3.1 Στρατηγική εισαγωγής στη μνήμη C

Το βίντεο που έκανε την αίτηση μπορεί να εισάγει σε μια αίτηση ολόκληρο το περιεχόμενο του suffix τμήματός του. Όταν η μνήμη C έχει διαθέσιμο χώρο τότε η αποθήκευση του suffix τμήματος γίνεται απευθείας. Όταν δεν υπάρχει διαθέσιμος χώρος τότε με βάση τη στρατηγική αντικατάστασης θα εξεταστεί αν το συγκεκριμένο βίντεο πληροί τις προδιαγραφές για να αποθηκευτεί στη μνήμη C ή αν πρέπει να απορριφθεί η αποθήκευσή του στην μνήμη C και να του δοθεί η ευκαιρία να αποθηκευτεί στη μνήμη B.

3.2.2.3.3.2 Στρατηγική αντικατάστασης στη μνήμη C

Όταν δεν υπάρχει διαθέσιμη μνήμη και απαιτείται τότε ξεκινά η στρατηγική αντικατάστασης για να εξεταστεί αν το βίντεο μπορεί να αποθηκευτεί στη μνήμη C ή αν πρέπει να απορριφθεί και να αποθηκεύσει το πρώτο τμήμα του στη μνήμη B. Για να παρθεί η συγκεκριμένη απόφαση αρχικά εξετάζονται όλα τα βίντεο στη μνήμη C τα οποία δεν παρακολουθούνται από κάποιο χρήστη και ως θύμα επιλέγεται το βίντεο με το μικρότερο caching value, το οποίο για κάθε βίντεο είναι ίσο με $\frac{n*ns}{Tc-Tr}$. Αν μπορούν να βρεθούν αρκετά βίντεο θύματα με caching value μικρότερο από το caching value του βίντεο που έκανε την αίτηση και εφόσον η μνήμη που θα αποδεσμευτεί μετά την αφαίρεσή τους είναι αρκετή για να αποθηκευτεί το suffix του βίντεο που έκανε την αίτηση, τότε ξεκινά η αφαίρεση των απαραίτητων θυμάτων και αποθηκεύεται το suffix του συγκεκριμένου βίντεο. Σε αντίθετη περίπτωση, απορρίπτεται η αίτηση του βίντεο για αποθήκευση στη μνήμη C και δίνεται η ευκαιρία στο συγκεκριμένο βίντεο να αποθηκεύσει το πρώτο τμήμα του στη μνήμη B.

3.2.2.4 “Full video with prefix and with limitation”

Η μέθοδος έχει περιγραφεί στο Κεφάλαιο 2 στις ενότητες 2.2.3.1 και 2.2.3.2.2.

3.3 Αξιολόγηση Απόδοσης

3.3.1 Μετρικές απόδοσης

Χρησιμοποιούμε τις ίδιες με μετρικές απόδοσης με αυτές στο Κεφάλαιο 2. Οι λεπτομέρειες δίνονται στην ενότητα 2.3.1. Αξίζει να αναφέρουμε πώς υπολογίζεται το κόστος αντικαταστάσεων σε κάθε μέθοδο. Στη μέθοδο *"Lazy Segmentation"* η αφαίρεση ενός τμήματος αντιστοιχεί σε μια αντικατάσταση. Στις μεθόδους *"Aggressive and Fixed Segmentation"* και *"Aggressive and guarded method"*, η αφαίρεση του προθέματος από τη μνήμη A και η αφαίρεση ενός τμήματος από τη μνήμη B αντιστοιχούν σε μια αντικατάσταση, αντίστοιχα. Τέλος στη *"Aggressive and guarded method"*, η αφαίρεση του suffix από τη μνήμη C αντιστοιχεί σε μια αντικατάσταση.

3.3.2 Μοντέλο Προσομοίωσης

Στις προσομοιώσεις μας υλοποιήθηκαν τρία διαφορετικά σενάρια για να δείξουμε την αποτελεσματικότητα των προτεινόμενων μεθόδων που παρουσιάστηκαν για μερική παρακολούθηση των βίντεο. Έχουμε υποθέσει ότι οι αφίξεις αιτήσεων για τα διάφορα βίντεο μοντελοποιούνται με κατανομή Poisson και έτσι οι χρόνοι μεταξύ διαδοχικών αφίξεων είναι εκθετικά κατανεμημένοι με μέση τιμή $\frac{1}{\lambda} = 15\ sec$. Στο σύστημά μας υποτίθεται ότι υπάρχουν V διαφορετικά βίντεο τα οποία αρχικά είναι αποθηκευμένα στον content server. Το μέγεθος των βίντεο καθορίζεται με βάση το σενάριο της προσομοίωσης. Τα σενάρια περιγράφονται στην επόμενη ενότητα. Η δημοτικότητα των βίντεο όπως αναφέραμε και στο Κεφάλαιο 2 προσεγγίζεται με μια Zipf-like κατανομή. Όλα τα αποτελέσματα που παρουσιάζονται στο Κεφάλαιο αυτό αντιστοιχούν σε παράμετρο skew της Zipf-like κατανομής s=0.53. Σε αυτό το Κεφάλαιο η κατανομή δημοτικότητας των βίντεο πλησιάζει πιο πολύ την ομοιόμορφη κατανομή και ως αποτέλεσμα η πιθανότητα μελλοντικής αίτησης για κάποιο βίντεο είναι πιο δύσκολο να προβλεφθεί. Η συγκεκριμένη τιμή που επιλέξαμε της Zipf-like κατανομής, για την παράμετρο s δεν είναι τυχαία. Στο [12] προκύπτει ότι δημοτικότητα των βίντεο μπορεί να προσεγγιστεί καλά με μια Zipf-like κατανομή με παράμετρο s=0.53, όταν η προσομοίωση γίνεται για μικρό χρονικό διάστημα. Για μεγαλύτερο χρονικό διάστημα (μεγαλύτερο από ένα μήνα), η δημοτικότητα των βίντεο δεν προσεγγίζεται καλά από μια Zipf-like κατανομή, [8]. Στις προσομοιώσεις μας οι συνολικές αφίξεις

στο σύστημα υποτίθενται 100.000, το οποίο αντιστοιχεί κατά μέσο όρο περίπου σε 17 μέρες λειτουργίας του συστήματος.

Η χωρητικότητα της μνήμης έχει οριστεί να είναι ίση με 500.000 blocks, το οποίο αντιστοιχεί περίπου στο 10% του συνολικού όγκου που καταλαμβάνουν όλα τα βίντεο. Όλες οι τιμές των παραμέτρων φαίνονται στον Πίνακα 3.2.

Παράμετροι	**Περιγραφή και προκαθορισμένες τιμές (default values)**
Bp	Μέγεθος προθέματος (32 μπλοκ)
1/λ	Μέση τιμή μεταξύ διαδοχικών αφίξεων αιτήσεων για τα διάφορα βίντεο (15 sec)
V	Αριθμός διακριτών βίντεο στο σύστημα (2.000)
s	Παράμετρος skew στη Zipf-like κατανομή (0.53)
tb	Διάρκεια ενός block (1.8 sec)
C	Συνολική χωρητικότητα μνήμης (500.000 μπλοκ)
pr	Ποσοστό συνολικής μνήμης το οποίο είναι διαθέσιμο για αποθήκευση των αρχικών τμημάτων των βίντεο (μνήμη A) (10%)
Runs	Αριθμός αιτήσεων στην προσομοίωσή μας (100.000 αιτήσεις)

Πίνακας 3.2: Παράμετροι του συστήματος και οι προκαθορισμένες τιμές τους.

3.3.2.1 Σενάρια Προσομοίωσης

Στις προσομοιώσεις μας υλοποιούνται τρία διαφορετικά σενάρια μέσα από τα οποία φαίνεται η αποτελεσματικότητα των τεσσάρων μεθόδων που περιγράφηκαν στην ενότητα 3.2. Στις επόμενες ενότητες ακολουθεί η περιγραφή των σεναρίων και τα διάφορα ποσοστά των χρηστών που επιλέγουν πλήρη ή μερική παρακολούθηση των βίντεο.

3.3.2.1.1 Σενάριο 1

Στο πρώτο σενάριο το μέγεθος των βίντεο επιλέγεται με βάση μια ομοιόμορφη κατανομή μεταξύ 1.000-4.000 blocks. Κάθε block διαρκεί 1.8 sec, κατά συνέπεια η διάρκεια των βίντεο κυμαίνεται μεταξύ 30-120 λεπτά. Στα [8] και [9] έχει μελετηθεί η συμπεριφορά των χρηστών σε πραγματικά συστήματα και τα αποτελέσματα που προκύπτουν εφαρμόζονται στο συγκεκριμένο

σενάριο. Έχει βρεθεί ότι ανεξάρτητα από το μέγεθος του βίντεο, το 78% των αιτήσεων σταματάνε την παρακολούθηση του βίντεο πριν τα 10 λεπτά, το 11% των χρηστών παρακολουθούν μεταξύ 10-30 λεπτών και μόλις το 11% παρακολουθεί ολόκληρο το βίντεο. Παρόμοια ποσοστά έχουν βρεθεί στο [11]. Επίσης στα [4] και [12] έχει βρεθεί ότι από τις αιτήσεις που θα παρακολουθήσουν μερικώς το βίντεο, οι περισσότερες σταματάνε στα αρχικά frames του βίντεο. Ως ελάχιστο χρόνο παρακολούθησης έχουμε θεωρήσει τα 2 λεπτά.

3.3.2.1.2 Σενάριο 2

Το μέγεθος των βίντεο επιλέγεται με βάση μια ομοιόμορφη κατανομή μεταξύ 250-4.000 blocks. Κάθε block διαρκεί 1.8 sec, κατά συνέπεια η διάρκεια των βίντεο κυμαίνεται μεταξύ 7.5-120 λεπτά. Όπως και στο [3], έγινε η υπόθεση ότι το 80% των αιτήσεων παρακολουθούν ακριβώς το 20% του βίντεο και το 20% των αιτήσεων παρακολουθούν ολόκληρο το βίντεο.

3.3.2.1.3 Σενάριο 3

Το μέγεθος των βίντεο επιλέγεται με βάση μια ομοιόμορφη κατανομή μεταξύ 250-4.000 blocks. Κάθε block διαρκεί 1.8 sec, κατά συνέπεια η διάρκεια των βίντεο κυμαίνεται μεταξύ 7.5-120 λεπτά. Το σενάριο αυτό στηρίζεται σε μετρήσεις που αναφέρονται στο [4] όπου μελετήθηκε η συμπεριφορά των χρηστών σε ένα πανεπιστήμιο. Τα βίντεο που ήταν αποθηκευμένα στον server αντιστοιχούν σε διαλέξεις από διάφορα μαθήματα. Βρέθηκε ότι το 55% των αιτήσεων παρακολουθεί ολόκληρο το βίντεο και το 45% παρακολουθεί μερικώς το βίντεο. Επιπρόσθετα εμείς υποθέσαμε ότι οι περισσότερες αιτήσεις (80%) που παρακολουθούν μερικώς το βίντεο σταματάνε στο 5% του συνολικού μεγέθους του βίντεο και οι υπόλοιπες (20%) σταματάνε στο 20% του συνολικού μεγέθους του βίντεο.

3.3.3 Αποτελέσματα Προσομοίωσης

Στις επόμενες ενότητες συγκρίνουμε τα αποτελέσματα των μεθόδων *“Aggressive and Fixed Segmentation”*, *“Aggressive and guarded method”* και *“Full video with prefix and with limitation”* με τη μέθοδο τμηματοποίησης από το [3] την *“Lazy Segmentation”*. Από τα αποτελέσματα των προσομοιώσεων παρατηρούμε ότι η *“Lazy Segmentation”* δεν είναι αποτελεσματική όταν το μέγεθος των βίντεο είναι μεγάλο και κατανέμεται σύμφωνα με ομοιόμορφη κατανομή. Πετυχαίνει

πάντα το μικρότερο BHR, τις μεγαλύτερες καθυστερήσεις κατά το ξεκίνημα των βίντεο καθώς επίσης και το μεγαλύτερο αριθμό συνολικών αντικαταστάσεων που συμβαίνουν στην προσομοίωση.

Στην συνέχεια παρουσιάζονται τα αποτελέσματα μελέτης της επίδρασης διαφορετικών μεγεθών μνήμης και διαφορετικών τιμών της παραμέτρου s της κατανομής δημοτικότητας των βίντεο, στις μετρικές απόδοσης του συστήματος και για τα τρία διαφορετικά σενάρια.

3.3.3.1 Αποτελέσματα των προσομοιώσεων στο σενάριο 1

Στην ενότητα αυτή υλοποιείται η προσομοίωση του σεναρίου 1 ώστε να μπορεί να γίνει υπολογισμός και σύγκριση των αποτελεσμάτων κάθε μεθόδου σε μερική παρακολούθηση των βίντεο. Στο συγκεκριμένο σενάριο έχουμε υποθέσει ότι περισσότερες αιτήσεις σταματάνε να βλέπουν τα βίντεο στα αρχικά frames τους. Όπως ήτανε αναμενόμενο οι μέθοδοι *"Full video with prefix and with limitation"* και *"Lazy Segmentation"* οι οποίες υλοποιούν πλήρη αποθήκευση του βίντεο (full video caching) δεν είναι αποδοτικές στην μετρική BHR. Στις συγκεκριμένες μεθόδους γίνεται κατάχρηση της μνήμης η οποία δεν αξιοποιείται σωστά ώστε να μπορούν να εξυπηρετούνται περισσότερες αιτήσεις. Η μέθοδος *"Lazy Segmentation"* στην περίπτωση που υπάρχει αρκετή διαθέσιμη συνολική μνήμη, τείνει να έχει και μερικώς και πλήρως αποθηκευμένα βίντεο, επειδή υλοποιεί τμηματοποίηση στα βίντεο κατά τη διάρκεια της στρατηγικής αντικατάστασης. Όμως για περιορισμένη μνήμη, όπως συμβαίνει και στα σενάρια μας, η συγκεκριμένη μέθοδος τείνει να υλοποιεί πλήρη αποθήκευση των βίντεο. Αυτό συμβαίνει εξαιτίας των περιορισμένων υποψήφιων θυμάτων που υπάρχουν κατά τη διάρκεια της στρατηγικής αντικατάστασης, επειδή τα περισσότερα βίντεο που εξετάζονται είναι currently played και ως αποτέλεσμα συχνά ως υποψήφια θύματα επιλέγονται αυτά που είναι μερικώς αποθηκευμένα, και έτσι δεν δίνεται η ευκαιρία να τμηματοποιηθούν αρκετά βίντεο.

Οι μέθοδοι *"Aggressive and Fixed Segmentation"* και *"Aggressive and guarded method"* είναι οι πιο αποδοτικές στην μετρική BHR γιατί υλοποιούν μερική αποθήκευση του βίντεο (partial video caching), αποθηκεύοντας έτσι από περισσότερα βίντεο κάποια αρχικά τμήματα και μόνο για τα πολύ δημοφιλή βίντεο αποθηκεύεται ολόκληρο το περιεχόμενό τους στη μνήμη B του proxy server. Αυτό συμβαίνει επειδή υπάρχει περιορισμός για την αποθήκευση των τμημάτων σε περίπτωση που δεν υπάρχει επαρκής χώρος στην μνήμη και επειδή οι συγκεκριμένες μέθοδοι υλοποιούν τμηματοποίηση στα βίντεο.

Στην μετρική καθυστερήσεις κατά το ξεκίνημα των βίντεο, οι μέθοδοι που προτείναμε παρουσιάζουν τα καλύτερα αποτελέσματα τα οποία δεν διαφέρουν πολύ μεταξύ τους. Όπως

αναφέρθηκε οι συγκεκριμένες μέθοδοι αφιέρωσαν το 10% της συνολικής μνήμης του proxy server για την αποθήκευση των video prefixes.

Στην μετρική που αφορά το συνολικό αριθμό αντικαταστάσεων, οι μέθοδοι *"Aggressive and guarded method"* και *"Full video with prefix and with limitation"* πετυχαίνουν τις μικρότερες τιμές. Ακολουθεί η μέθοδος *"Aggressive and Fixed Segmentation*", ενώ η χειρότερη και σε αυτή την μετρική απόδοσης είναι η μέθοδος *"Lazy Segmentation"*.

3.3.3.1.1 Η επίδραση διαφορετικών μεγεθών μνήμης.

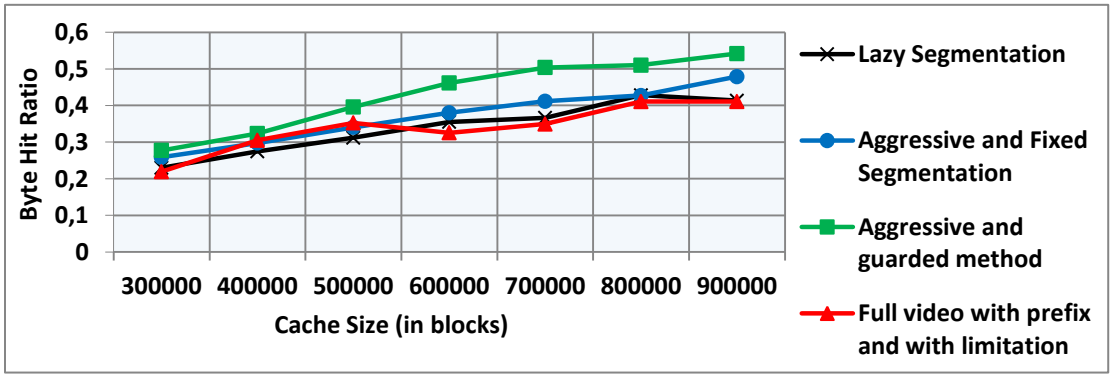
Σε αυτή την ενότητα εξετάζεται η αποτελεσματικότητα των μεθόδων στις μετρικές απόδοσης με τη χρήση διαφορετικών μεγεθών μνήμης. Από τα αποτελέσματα στα τρία σχήματα που ακολουθούν, Σχ.3.1-Σχ.3.3, φαίνεται ότι η πιο αποδοτική μέθοδος σε αυτό το σενάριο είναι η *"Aggressive and guarded method"*. Η μέθοδος αυτή πετυχαίνει το υψηλότερο BHR, οι καθυστερήσεις κατά το ξεκίνημα των βίντεο είναι μικρές χωρίς να υπάρχουν μεγάλες διαφορές από αυτές της μεθόδου *"Full video with prefix and with limitation"*, και τέλος πετυχαίνει μικρό αριθμό συνολικών αντικαταστάσεων.

Στο Σχ.3.1 φαίνεται η υπεροχή της *"Aggressive and guarded method"* στη μετρική BHR, η οποία οφείλεται στη στρατηγική της μνήμης B καθώς και στο τρόπο τμηματοποίησης του βίντεο. Η συγκεκριμένη μέθοδος τμηματοποιεί το βίντεο με βάση το συνολικό μέγεθός του και όχι με βάση κάποιο σταθερό αριθμό από blocks όπως γίνεται στη μέθοδο *"Aggressive and Fixed Segmentation"*. Αυτό όπως φαίνεται είναι αποδοτικό γιατί σε κάθε νέα αίτηση το βίντεο προσπαθεί να εισάγει στη μνήμη B ένα επιπλέον τμήμα το οποίο είναι προσαρμοσμένο στη συνολική του διάρκεια. Επίσης η στρατηγική αντικατάστασης στη μνήμη B αποσκοπεί στην αφαίρεση τμημάτων από το βίντεο θύμα με αρκετά τμήματα αποθηκευμένα στη μνήμη B. Η αφαίρεση κάποιων τμημάτων από το συγκεκριμένο βίντεο πιθανόν να μην επηρεάζει μελλοντικές αιτήσεις που θα αναφέρονται σε αυτό, εφόσον για μια αίτηση η πιθανότητα παρακολούθησης ολόκληρου του βίντεο είναι μικρή. Έτσι με βάση τους πιο πάνω περιορισμούς που αναφέρθηκαν και εξαιτίας της χρήσης μικρότερων και περισσότερων τμημάτων σε σχέση με την *"Aggressive and Fixed Segmentation"*, η *"Aggressive and guarded method"* μπορεί να αποθηκεύει περισσότερα βίντεο στη μνήμη B έχοντας από το κάθε ένα αποθηκευμένα κάποια αρχικά τμήματα. Επίσης η συγκεκριμένη μέθοδος μπορεί να εξυπηρετήσει ένα μικρό ποσοστό αιτήσεων που παρακολουθεί ολόκληρο το βίντεο, εξαιτίας της ύπαρξης μνήμης C στην οποία αποθηκεύονται εξ' ολοκλήρου λίγα βίντεο. Αυτό έχει σαν αποτέλεσμα η συγκεκριμένη μέθοδος να είναι αρκετά αποδοτική στη μετρική BHR εφόσον πετυχαίνει να εξυπηρετήσει τις περισσότερες αιτήσεις που ζητάνε μόνο τα αρχικά frames των

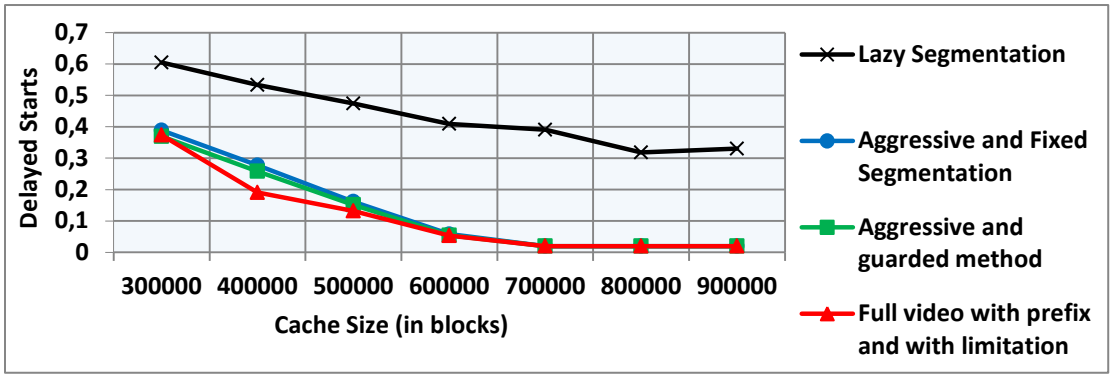
βίντεο και παράλληλα να εξυπηρετήσει τις περισσότερες από τις λίγες αιτήσεις που παρακολουθούν ολόκληρο το περιεχόμενο του βίντεο.

Στην μετρική που αφορά τις καθυστερήσεις κατά το ξεκίνημα των βίντεο όπως φαίνεται στο Σχ.3.2 οι δικές μας μέθοδοι παρουσιάζουν παρόμοια αποτελέσματα, εξαιτίας της όμοιας στρατηγικής εισαγωγής και αντικατάστασης που χρησιμοποιείται στη μνήμη Α. Σε κάποιες περιπτώσεις η μέθοδος *"Full video with prefix and with limitation"* πετυχαίνει λίγο καλύτερα αποτελέσματα γιατί η αφαίρεση του suffix του βίντεο θύματος από τη μνήμη Β γίνεται πιο γρήγορα, δίνοντας την ευκαιρία στη μέθοδο να μειώσει το caching value του συγκεκριμένου βίντεο, εφόσον η παράμετρος ns παίρνει μικρότερη τιμή. Έτσι είναι πιθανόν το συγκεκριμένο βίντεο να είναι το επόμενο θύμα που θα αφαιρεθεί από τη μνήμη Α όταν απαιτείται εκεί διαθέσιμος χώρος. Τέλος, παρατηρούμε ότι η μέθοδος *"Lazy Segmentation"* πετυχαίνει τις μεγαλύτερες καθυστερήσεις κατά το ξεκίνημα των βίντεο γιατί η συγκεκριμένη μέθοδος δεν αφιερώνει κάποιο τμήμα της συνολικής μνήμης του proxy server για την αποθήκευση των προθεμάτων των βίντεο.

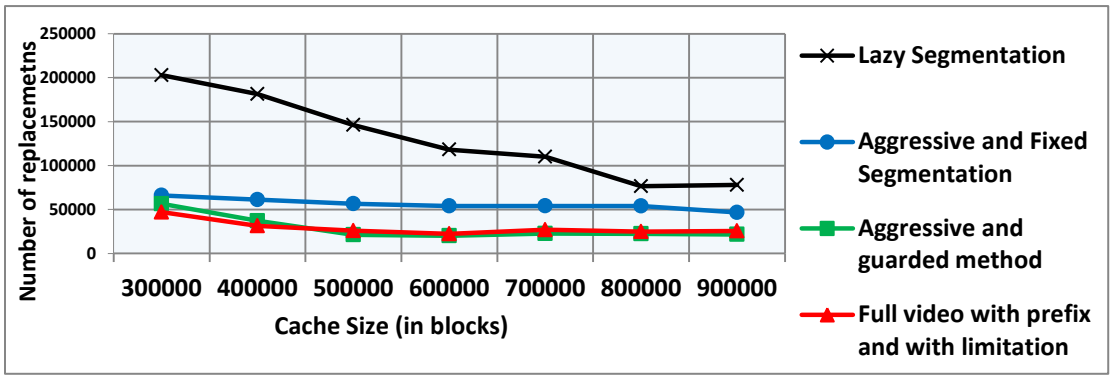
Στην μετρική που αφορά τον συνολικό αριθμό αντικαταστάσεων όπως φαίνεται στο Σχ.3.3, η μέθοδος *"Lazy Segmentation"* πετυχαίνει τα χειρότερα αποτελέσματα γιατί σε κάθε καινούργια αίτηση το βίντεο είναι πάντα επιλέξιμο να αποθηκευτεί στη μνήμη του proxy server, αν δεν ήταν ήδη αποθηκευμένο σε αυτή. Αυτή η επιθετική στρατηγική επιφέρει αυξημένες αντικαταστάσεις, γιατί απαιτούνται συχνά αφαιρέσεις θυμάτων για να αποδεσμευτεί η απαραίτητη μνήμη. Οι δικές μας μέθοδοι σε κάθε αίτηση προσπαθούν να αποθηκεύσουν ένα μέρος του συνολικού περιεχομένου του βίντεο μετά από ελέγχους. Αυτό οδηγεί τις μεθόδους μας σε καλύτερες αποφάσεις και κατά συνέπεια σε μειωμένο αριθμό συνολικών αντικαταστάσεων.



Σχ.3.1: Byte-Hit Ratio σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.



Σχ.3.2: Καθυστερήσεις κατά το ξεκίνημα των βίντεο σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.

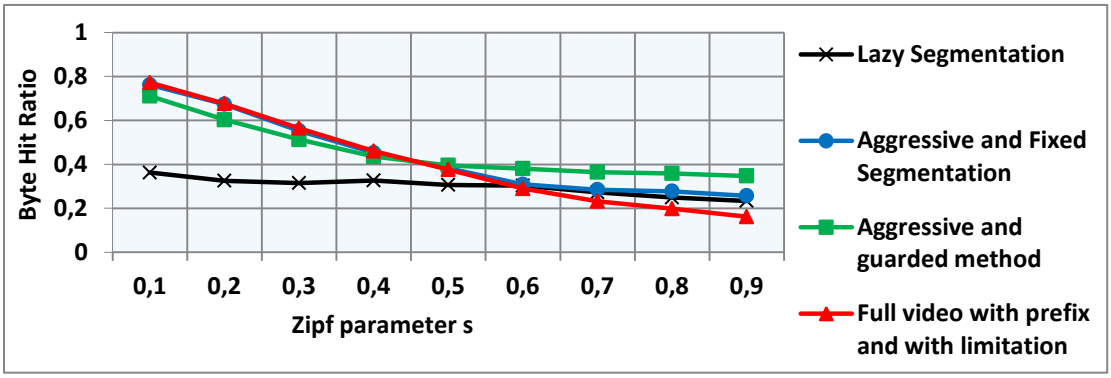


Σχ.3.3: Συνολικές αντικαταστάσεις σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.

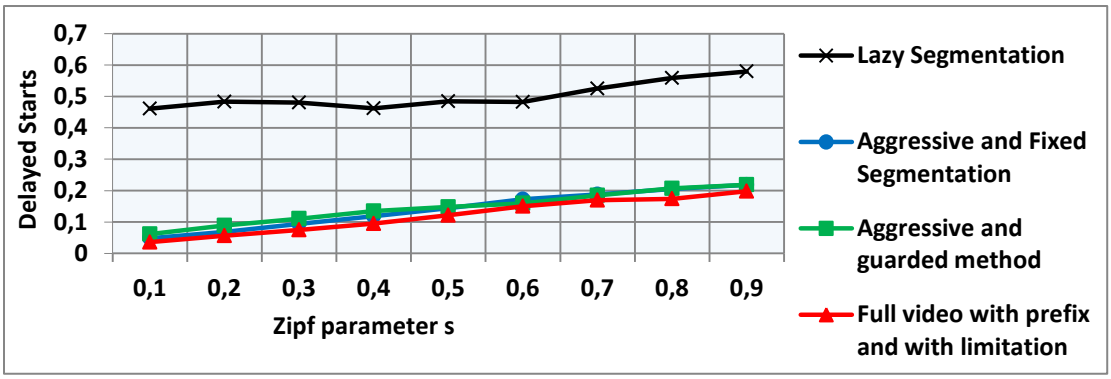
3.3.3.1.2 Η επίδραση διαφορετικών τιμών της παραμέτρου skew της κατανομής δημοτικότητας των βίντεο

Στην προσομοίωση αυτή μελετήσαμε την απόδοση των αλγορίθμων για διαφορετικές τιμές της παραμέτρου s της Zipf-like κατανομής. Όταν η παράμετρος s είναι μικρότερη ή ίση από 0.4 η πιο αποδοτική μέθοδος είναι η *"Full video with prefix and with limitation"*. Όμως για μεγαλύτερες τιμές της παραμέτρου s τα αποτελέσματα δείχνουν ότι η μέθοδος *"Full video with prefix and with limitation"* γίνεται η χειρότερη στην μετρική BHR και η *"Aggressive and guarded method"* δίνει τα καλύτερα αποτελέσματα όσον αφορά στις μετρικές απόδοσης BHR και συνολικός αριθμός αντικαταστάσεων. Στην μετρική που αφορά τις καθυστερήσεις κατά το ξεκίνημα των βίντεο δεν παρατηρούνται μεγάλες διαφορές μεταξύ των δύο μεθόδων.

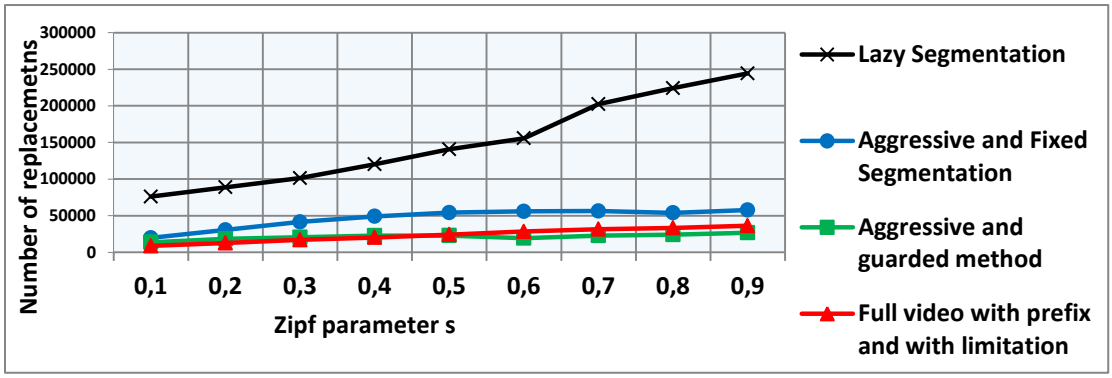
Όπως ήδη αναφέραμε, στο [12] προκύπτει ότι η δημοτικότητα των βίντεο μπορεί να προσεγγιστεί καλά με μια Zipf-like κατανομή με παράμετρο s=0.53. Η *"Aggressive and guarded method"* υπερτερεί των άλλων μεθόδων στη μετρική BHR όταν η Zipf-like κατανομή πλησιάζει την ομοιόμορφη κατανομή. Με τις στρατηγικές της η μέθοδος αυτή τείνει να έχει περισσότερα μερικώς αποθηκευμένα βίντεο στη μνήμη B, όταν το σύστημα φτάνει σε μόνιμη κατάσταση και έτσι μπορεί να εξυπηρετήσει περισσότερες αιτήσεις σε σχέση με τις άλλες μεθόδους. Από τα αποτελέσματα για τρείς μετρικές απόδοσης στα Σχ.3.4-Σχ.3.6, φαίνεται ότι η *"Aggressive and guarded method"* είναι η πιο σταθερή, γιατί όταν η παράμετρος s είναι μεταξύ 0.5 και 0.9 εξακολουθεί να επιτυγχάνει σχεδόν σταθερό BHR. Στις άλλες δύο μετρικές απόδοσης τα αποτελέσματα δεν διαφέρουν σημαντικά σε σχέση με τη μέθοδο *"Full video with prefix and with limitation"*.



Σχ.3.4: Byte-Hit Ratio σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.



Σχ.3.5: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.



Σχ.3.6: Συνολικές αντικαταστάσεις σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.

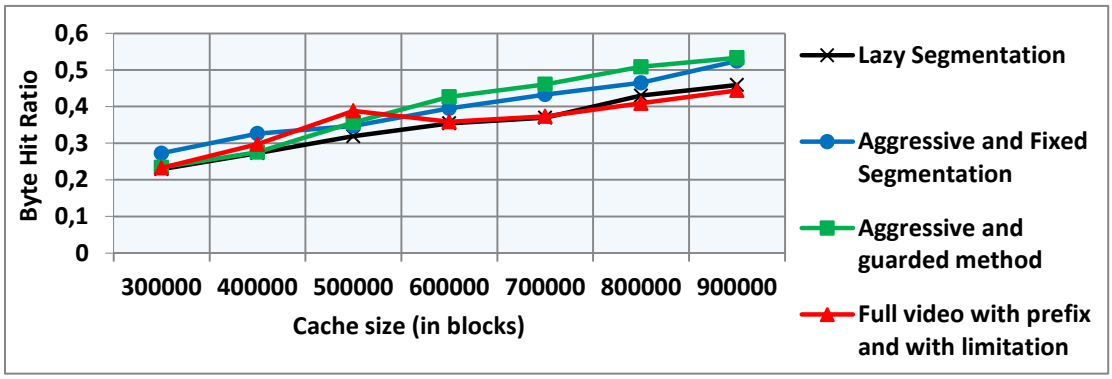
3.3.3.2 Αποτελέσματα των προσομοιώσεων στο σενάριο 2

Το σενάριο αυτό αντιπροσωπεύει μια μέση κατάσταση όσον αφορά στα ποσοστά παρακολούθησης των βίντεο, καθώς οι χρήστες οι οποίοι παρακολουθούν μερικώς το βίντεο αναμένεται να παρακολουθήσουν κατά μέσο όρο 12 λεπτά. Οι μέθοδοι *"Aggressive and Fixed Segmentation"* και *"Aggressive and guarded method"* είναι οι πιο αποδοτικές όσον αφορά την μετρική BHR γιατί στο μεγαλύτερο μέρος της συνολικής μνήμης αποθηκεύονται τα suffix των βίντεο με μεθόδους τμηματοποίησης. Με αυτόν τον τρόπο αποθηκεύονται τα αρχικά τμήματα από αρκετά βίντεο. Στην μετρική καθυστερήσεις κατά το ξεκίνημα των βίντεο, οι δικές μας μέθοδοι παρουσιάζουν τα καλύτερα αποτελέσματα τα οποία δεν διαφέρουν πολύ μεταξύ τους. Στην μετρική που αφορά το συνολικό αριθμό αντικαταστάσεων, οι μέθοδοι *"Aggressive and guarded method"* και *"Full video with prefix and with limitation"* πετυχαίνουν τις μικρότερες τιμές.

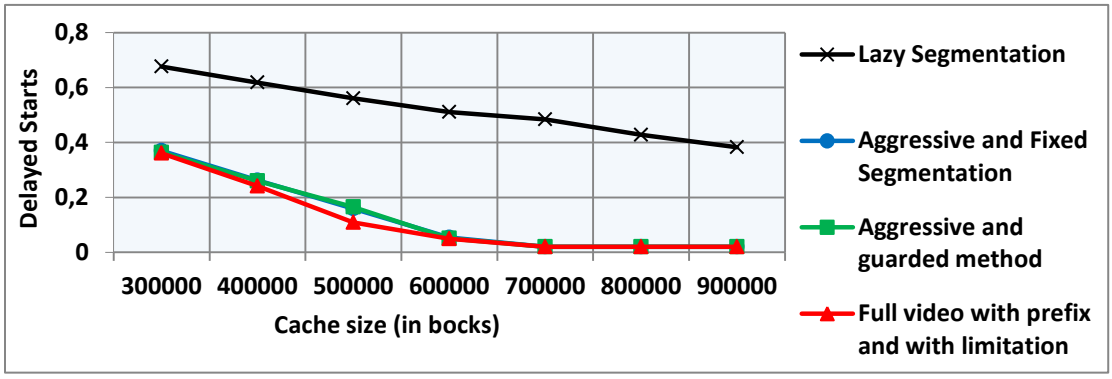
3.3.3.2.1 Η επίδραση διαφορετικών μεγεθών μνήμης.

Από τα αποτελέσματα που ακολουθούν στα Σχ.3.7-Σχ.3.9 παρατηρούμε ότι η *"Lazy Segmentation"* δίνει τα χειρότερα αποτελέσματα και για τις τρείς μετρικές απόδοσης του συστήματος και σε αυτό το σενάριο. Η *"Aggressive and guarded method"* δίνει τα καλύτερα αποτελέσματα, παρόλο που για μικρή μνήμη έχει μικρότερο BHR σε σχέση με την *"Aggressive and Fixed Segmentation"*, όταν όμως η μνήμη αυξηθεί τα αποτελέσματα αντιστρέφονται. Η *"Aggressive and guarded method"* επιτυγχάνει επίσης το μικρότερο αριθμό συνολικών αντικαταστάσεων.

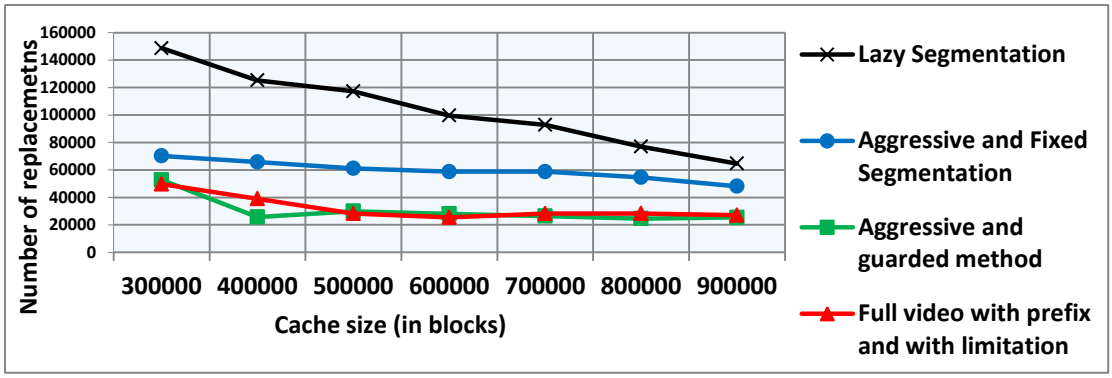
Από τις υποθέσεις του συγκεκριμένου σεναρίου μόνο το 20% των αιτήσεων παρακολουθούν ολόκληρο το βίντεο. Οι μέθοδοι *"Lazy Segmentation"* και *"Full video with prefix and with limitation"*, κάνουν κατάχρηση στη μνήμη με την εισαγωγή μεγάλου περιεχομένου των βίντεο. Παρόλο που οι συγκεκριμένες μέθοδοι επιτυγχάνουν παρόμοιες τιμές στη μετρική BHR, σε αυτό το σενάριο, στις άλλες μετρικές απόδοσης η μέθοδος *"Full video with prefix and with limitation"* είναι πολύ καλύτερη από την μέθοδο *"Lazy Segmentation"* γιατί ελέγχει αν το βίντεο πληροί τις προδιαγραφές για να αποθηκεύσει το suffix τμήμα του στη μνήμη B, επιφέροντας μικρότερο αριθμό συνολικών αντικαταστάσεων. Επίσης η μέθοδος *"Full video with prefix and with limitation"* αφιερώνει ένα τμήμα της συνολικής μνήμης του proxy server για την αποθήκευση των prefixes των βίντεο, και έτσι πετυχαίνει μικρές καθυστερήσεις κατά το ξεκίνημα των βίντεο.



Σχ.3.7: Byte-Hit Ratio σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.



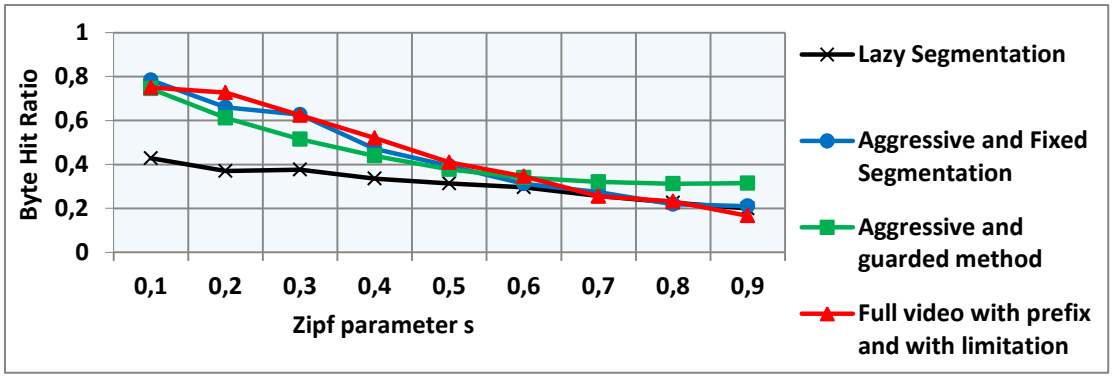
Σχ.3.8: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.



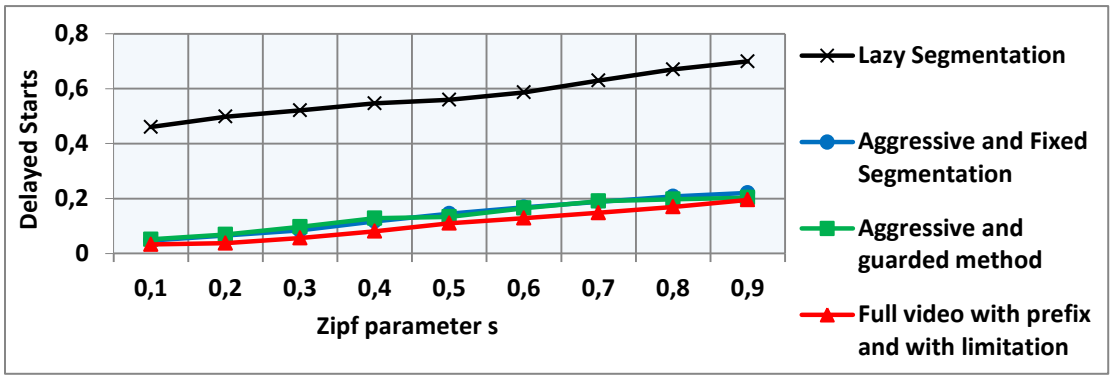
Σχ.3.9: Συνολικές αντικαταστάσεις σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.

3.3.3.2.2 Η επίδραση διαφορετικών τιμών της παραμέτρου skew της κατανομής δημοτικότητας των βίντεο

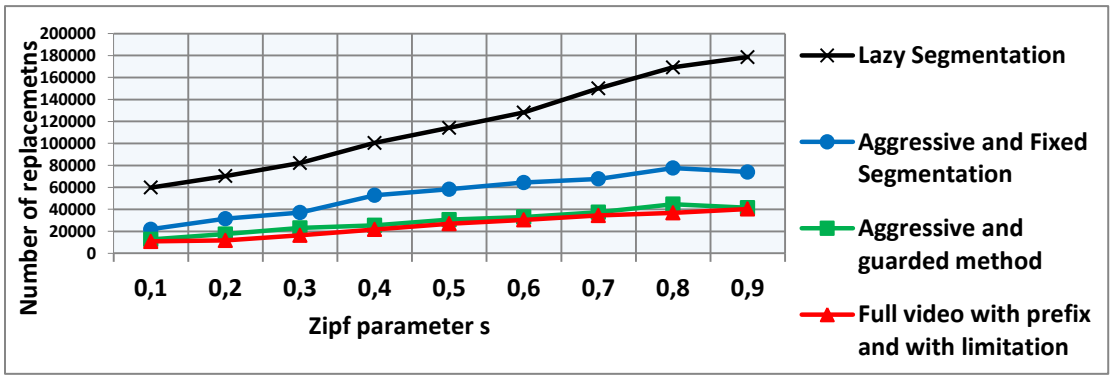
Στην προσομοίωση αυτή μελετήσαμε την απόδοση των αλγορίθμων για διαφορετικές τιμές της παραμέτρου s της Zipf-like κατανομής. Όταν η παράμετρος s είναι μικρότερη από 0.5 η πιο αποδοτική μέθοδος για όλες τις μετρικές απόδοσης είναι η *"Full video with prefix and with limitation"*. Οι πιθανότητες ζήτησης των βίντεο διαφέρουν αρκετά και έτσι είναι εύκολο να ξεχωρίσουν τα δημοφιλή βίντεο. Όταν τα δημοφιλή βίντεο αποθηκευτούν ολόκληρα στη μνήμη τότε μπορούν να εξυπηρετήσουν τις περισσότερες αφικνούμενες αιτήσεις καθώς αυτές θα αναφέρονται με μεγάλη πιθανότητα στα συγκεκριμένα βίντεο. Όταν η παράμετρος s είναι μεγαλύτερη από 0.5 η πιο αποδοτική μέθοδος στη μετρική BHR είναι η *"Aggressive and guarded method"* η οποία με τις στρατηγικές της μπορεί να αποθηκεύει περισσότερα μερικώς αποθηκευμένα βίντεο και έτσι να εξυπηρετεί τις περισσότερες αιτήσεις. Όσον αφορά στις άλλες δύο μετρικές απόδοσης τα αποτελέσματα δεν διαφέρουν σημαντικά σε σχέση με την μέθοδο *"Full video with prefix and with limitation"*.



Σχ.3.10: Byte-Hit Ratio σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.



Σχ.3.11: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.



Σχ.3.12: Συνολικές αντικαταστάσεις σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.

3.3.3.3 Αποτελέσματα των προσομοιώσεων στο σενάριο 3

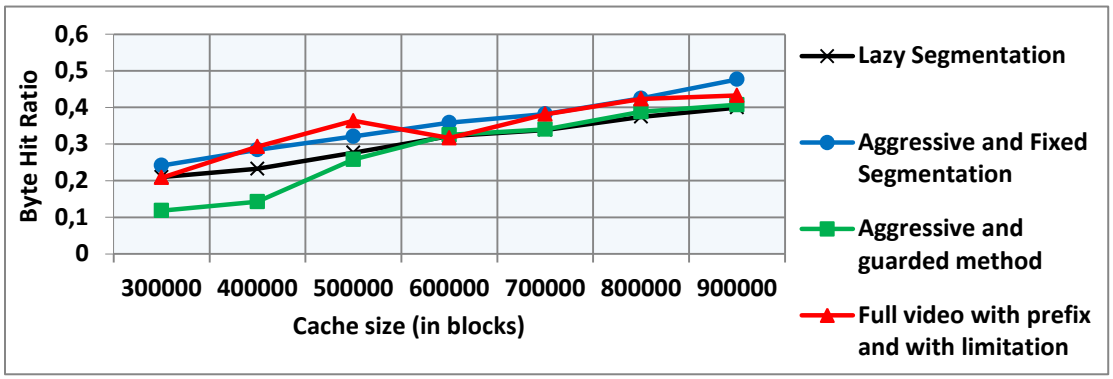
Το σενάριο αυτό διαφέρει αρκετά από τα δύο προηγούμενα καθώς τώρα το μεγαλύτερο ποσοστό των χρηστών παρακολουθεί ολόκληρο το βίντεο, ενώ το υπόλοιπο ποσοστό των χρηστών παρακολουθεί μόνο ένα πολύ μικρό μέρος του βίντεο το οποίο μπορεί να αντιστοιχεί σε μόλις μερικά δευτερόλεπτα.

Σε αυτό το σενάριο η πιο αποδοτική μέθοδος είναι η *"Full video with prefix and with limitation"* η οποία επιτυγχάνει το υψηλότερο BHR. Αυτό οφείλεται στην πλήρη αποθήκευση του βίντεο (full video caching) που υλοποιεί. Επιπλέον η συγκεκριμένη μέθοδος ελέγχει αν το βίντεο πληροί τις προδιαγραφές για αποθήκευση στη μνήμη B και έτσι αποθηκεύονται σε αυτή μόνο τα δημοφιλή βίντεο. Όταν το σύστημα φτάσει σε μόνιμη κατάσταση, στη συγκεκριμένη μέθοδο θα υπάρχουν λιγότερα βίντεο αποθηκευμένα στη μνήμη B του proxy server σε σχέση με τις μεθόδους που υλοποιούν τμηματοποίηση στα βίντεο, όμως τα συγκεκριμένα βίντεο θα έχουν αποθηκευμένο ολόκληρο το περιεχόμενό τους και έτσι θα εξυπηρετούνται οι περισσότερες αιτήσεις οι οποίες παρακολουθούν ολόκληρο το περιεχόμενο των βίντεο. Στη μετρική καθυστερήσεις κατά το ξεκίνημα των βίντεο, οι δικές μας μέθοδοι παρουσιάζουν τα καλύτερα αποτελέσματα επειδή αφιερώνουν ένα μέρος της συνολικής μνήμης για την αποθήκευση των video prefixes. Στη μετρική που αφορά τον συνολικό αριθμό αντικαταστάσεων, οι μέθοδοι *"Full video with prefix and with limitation"* και *"Aggressive and guarded method"* επιτυγχάνουν τα καλύτερα αποτελέσματα.

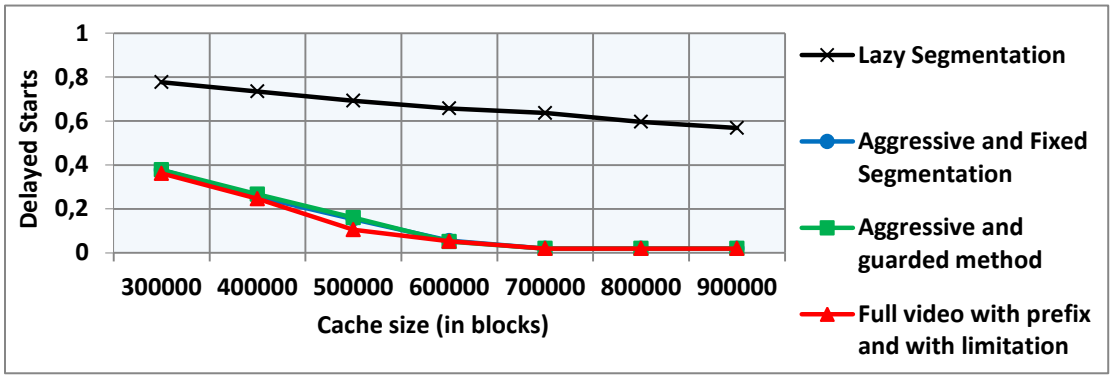
3.3.3.3.1 Η επίδραση διαφορετικών μεγεθών μνήμης.

Στο Σχ.3.13 φαίνονται τα αποτελέσματα των μεθόδων που προτάθηκαν, όσον αφορά τη μετρική BHR σαν συνάρτηση των διαφορετικών μεγεθών συνολικής μνήμης. Η συνολική μνήμη μεγέθους 300.000 blocks αντιστοιχεί περίπου στο 7% του συνολικού όγκου που καταλαμβάνουν όλα τα βίντεο ενώ συνολική μνήμη μεγέθους 900.000 blocks αντιστοιχεί περίπου στο 20% του συνολικού όγκου. Οι μέθοδοι *"Aggressive and Fixed Segmentation"* και *"Full video with prefix and with limitation"* είναι οι πιο αποδοτικές στη συγκεκριμένη μετρική απόδοσης. Σε αντίθεση με την *"Aggressive and guarded method",* οι παραπάνω μέθοδοι εισάγουν το suffix μέρος των βίντεο πιο γρήγορα. Η μέθοδος *"Aggressive and Fixed Segmentation"* επειδή χρησιμοποιεί μεγαλύτερα τμήματα σε σχέση με την *"Aggressive and guarded method"* αποθηκεύει κάποια τμήματα από λιγότερα βίντεο, αλλά όταν έρθει αίτηση για κάποιο από αυτά θα μπορεί να εξυπηρετήσει μεγαλύτερη διάρκεια σε σχέση με την *"Aggressive and guarded method"*. Η μέθοδος "*Lazy Segmentation*" έχει μέτρια αποτελέσματα σε αυτή την μετρική απόδοσης. Ο λόγος οφείλεται στην επιθετική στρατηγική που εφαρμόζει κατά την οποία δεν ελέγχει αν πρέπει να αποθηκεύσει

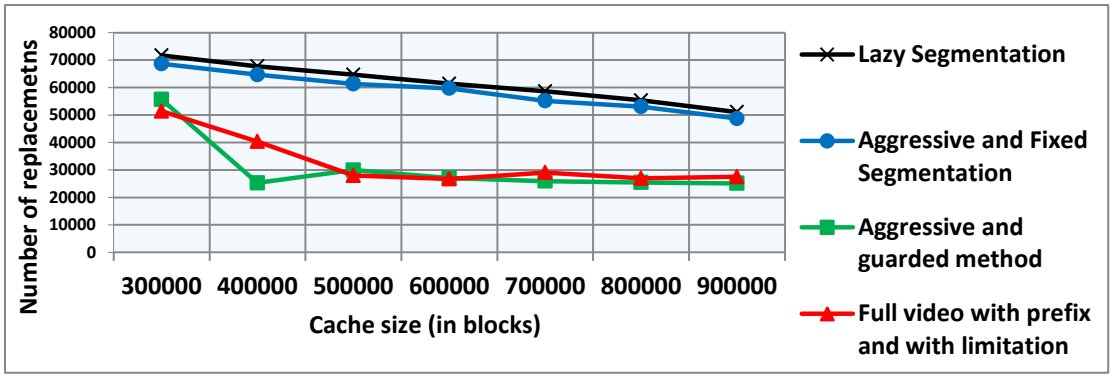
κάποιο βίντεο που έρχεται πρώτη φορά στο σύστημα, αλλά το αποθηκεύει πάντα. Όμως όταν η μνήμη είναι περιορισμένη ο αλγόριθμος αυτός οδηγείται σε λάθος αποφάσεις και δεν κάνει αποτελεσματική χρησιμοποίηση της μνήμης.



Σχ.3.13: Byte-Hit Ratio σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.



Σχ.3.14: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.



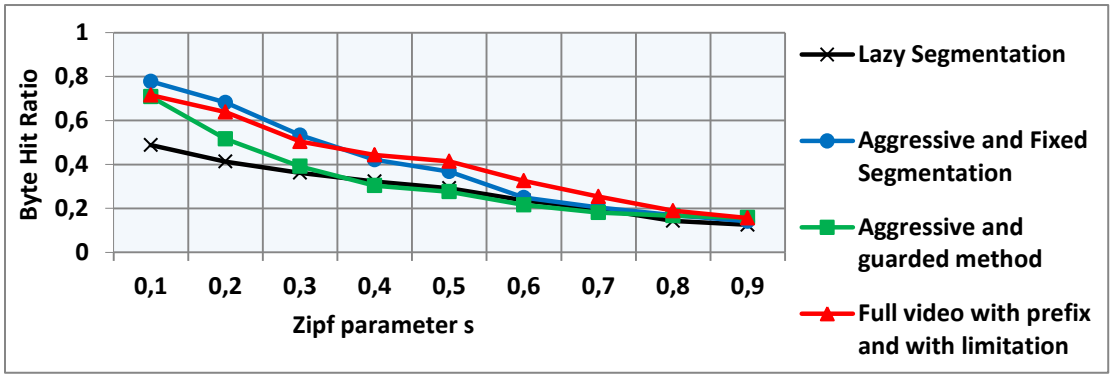
Σχ.3.15: Συνολικές αντικαταστάσεις σαν συνάρτηση της συνολικής χωρητικότητας μνήμης.

2.3.3.3.2 Η επίδραση διαφορετικών τιμών της παραμέτρου skew της κατανομής δημοτικότητας των βίντεο

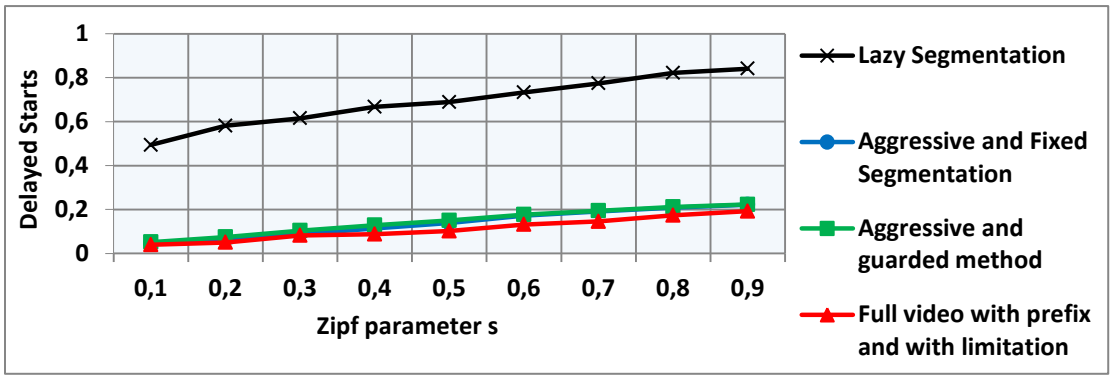
Όπως φαίνεται από τα αποτελέσματα στα Σχ.3.16- Σχ.3.18, εξακολουθεί σε αυτό το σενάριο να πετυχαίνει τα καλύτερα αποτελέσματα και για τις τρείς μετρικές απόδοσης του συστήματος η μέθοδος *"Full video with prefix and with limitation"*. Η χειρότερη μέθοδος για όλες τις μετρικές απόδοσης είναι για άλλη μια φορά η *"Lazy Segmentation"*.

Στο συγκεκριμένο σενάριο η *"Aggressive and guarded method"* δεν είναι αποτελεσματική στη μετρική BHR επειδή σε σχέση με τις άλλες μεθόδους, αποθηκεύει αρχικά τμήματα περισσότερων βίντεο στο σύστημα. Όμως οι περισσότερες αιτήσεις σε αυτό το σενάριο παρακολουθούν ολόκληρο το αντίστοιχο βίντεο, ενώ η συγκεκριμένη μέθοδος μπορεί να εξυπηρετήσει καλύτερα αιτήσεις που παρακολουθούν μερικώς τα βίντεο.

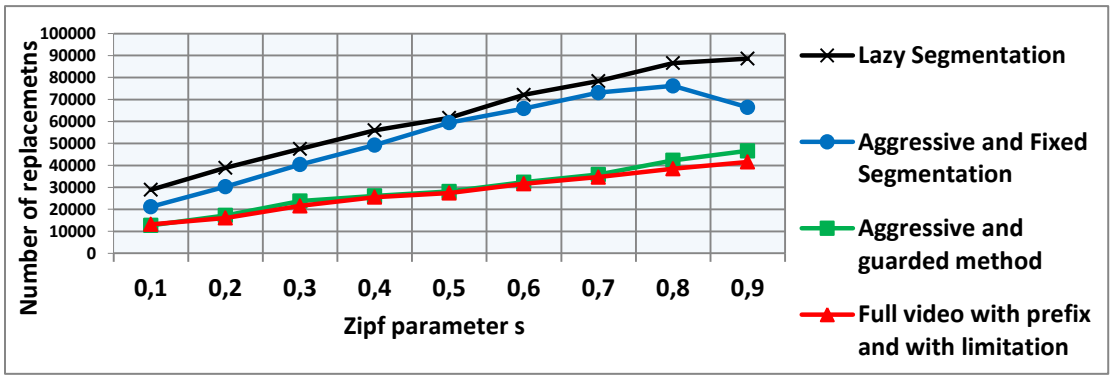
Η *"Aggressive and guarded method"* μπορεί να λειτουργήσει με πλήρη αποθήκευση των βίντεο (full video caching) αν δεν αφιερωθεί καθόλου μνήμη του proxy server για τη μνήμη B και αφιερωθεί ολόκληρη η μνήμη (εκτός από αυτήν που αφιερώνεται στην μνήμη A) στη μνήμη C. Μπορεί επίσης να λειτουργήσει υλοποιώντας στο μεγαλύτερο μέρος της συνολικής μνήμης πλήρη αποθήκευση των βίντεο και σε ένα μικρότερο μέρος μερική αποθήκευση των βίντεο (partial video caching), αν αφιερωθεί μεγαλύτερο ποσοστό συνολικής μνήμης στην μνήμη C και λιγότερο στην μνήμη B.



Σχ.3.16: Byte-Hit Ratio σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.



Σχ.3.17: Καθυστερήσεις στην αρχή των βίντεο σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.



Σχ.3.18: Συνολικές αντικαταστάσεις σαν συνάρτηση της παραμέτρου s της κατανομής δημοτικότητας των βίντεο.

3.4 Πολυπλοκότητα αλγορίθμων

Αξίζει να σχολιάσουμε την πολυπλοκότητα υλοποίησης των αλγορίθμων ειδικότερα όσον αφορά την ποσότητα πληροφορίας που χρειάζεται να διατηρεί και να ανανεώνει ο κάθε αλγόριθμος για να λειτουργήσει. Οι μέθοδοι *"Aggressive and guarded method"*, *"Aggressive and Fixed Segmentation"* και *"Full video with prefix and with limitation"* για να μπορούν να υπολογίζουν το caching value κάθε βίντεο πρέπει να κρατάνε την ίδια πληροφορία καθώς χρησιμοποιούν τον ίδιο τύπο. Η μέθοδος *"Lazy Segmentation"* για τον υπολογισμό του caching value χρειάζεται να αποθηκεύει περισσότερη πληροφορία. Παρόλο που χρειάζεται περισσότερη πληροφορία για κάθε βίντεο, παρουσιάζει τα χειρότερα αποτελέσματα γιατί με την στρατηγική της δεν κάνει αποδοτική αξιοποίηση της μνήμης. Όταν η μνήμη είναι περιορισμένη τότε η συγκεκριμένη μέθοδος τείνει να έχει αποθηκευμένα ολόκληρα βίντεο. Για να μπορεί να αποθηκεύει αρκετά βίντεο μερικώς, πρέπει να υπάρχει περισσότερη διαθέσιμη μνήμη, ώστε κατά την επιλογή θυμάτων να έχει να επιλέξει μεταξύ περισσότερων βίντεο και έτσι να υλοποιήσει τμηματοποίηση σε αρκετά βίντεο θύματα.

Οι μέθοδοι *"Aggressive and guarded method"*, *"Aggressive and Fixed Segmentation"* και *"Lazy Segmentation"*, οι οποίες υλοποιούν τμηματοποίηση στα βίντεο πρέπει να κρατάνε επιπλέον πληροφορία για το συνολικό αριθμό τμημάτων του κάθε βίντεο, για το μέγεθος του κάθε τμήματος, καθώς και τον αριθμό των τμημάτων που έχει ήδη αποθηκεύσει κάποιο βίντεο στη μνήμη ώστε να γνωρίζει το σύστημα το επόμενο τμήμα προς αποθήκευση ή το τμήμα προς αφαίρεση.

3.5 Συμπεράσματα από τα αποτελέσματα των προσομοιώσεων

Από τα τρία σενάρια που εξετάστηκαν για μερική παρακολούθηση των βίντεο, είτε σε μεγάλο είτε σε μικρότερο ποσοστό και από τις διαφορετικές στρατηγικές των τεσσάρων μεθόδων, μπορούμε να έχουμε μια συνολική εικόνα για τον τρόπο τμηματοποίησης του βίντεο και τον τρόπο διαχείρισης της συνολικής μνήμης του proxy server ώστε να αξιοποιείται αποδοτικά.

Όταν το μεγαλύτερο ποσοστό των χρηστών παρακολουθεί ένα μικρό αρχικό μέρος του βίντεο, τότε είναι πιο αποδοτικό να αποθηκεύονται στη μνήμη του proxy server αρκετά βίντεο με λίγα τμήματα από το καθένα. Αυτό υλοποιείται από την *"Aggressive and guarded method"*, η οποία τμηματοποιεί τα βίντεο σε αρκετά τμήματα ανάλογα με τη διάρκειά τους και επιπλέον στην επιλογή θυμάτων ελέγχει πρώτα τα βίντεο που έχουν αποθηκεύσει αρκετά τμήματα στη μνήμη του proxy server έτσι ώστε μετά την αφαίρεσή τους να μην επηρεαστούν πολλές από τις μελλοντικές

αιτήσεις για τα συγκεκριμένα βίντεο, όταν οι περισσότεροι χρήστες παρακολουθούν μερικώς τα βίντεο, η *"Aggressive and guarded method"* η οποία τμηματοποιεί το βίντεο με βάση τη διάρκειά τους είναι πιο αποδοτική από την *"Aggressive and Fixed Segmentation"*, η οποία τμηματοποιεί το βίντεο με βάση ένα σταθερό αριθμό από blocks. Οι δύο συγκεκριμένες μέθοδοι έχουν τους ίδιους περιορισμούς για εισαγωγή κάποιου τμήματος στη μνήμη B του proxy server, στην περίπτωση που δεν υπάρχει επαρκής χώρος. Η διαφορά τους όπως ήδη αναφέρθηκε έγκειται στο μέγεθος του τμήματος το οποίο αποθηκεύεται και στην προτεραιότητα στην αφαίρεση των βίντεο που έχουν αποθηκευμένα αρκετά τμήματα. Η *"Aggressive and Fixed Segmentation"* επειδή τμηματοποιεί κάθε βίντεο σε μεγαλύτερα τμήματα αποθηκεύει λιγότερα βίντεο σε σχέση με την *"Aggressive and guarded method"* αλλά από τα συγκεκριμένα βίντεο είναι αποθηκευμένη μεγαλύτερη διάρκεια. Έτσι, η *"Aggressive and guarded method"* είναι αποδοτικότερη στην περίπτωση που οι περισσότερες αιτήσεις απαιτούν μερική παρακολούθηση των βίντεο και η *"Aggressive and Fixed Segmentation"* είναι αποδοτικότερη όταν οι περισσότερες αιτήσεις απαιτούν πλήρη παρακολούθηση των βίντεο.

Όταν το μεγαλύτερο ποσοστό των χρηστών παρακολουθεί ολόκληρο το βίντεο, τότε εξακολουθεί να είναι αποδοτική η πλήρης αποθήκευση των βίντεο στη μνήμη, ακόμα και αν το υπόλοιπο ποσοστό των χρηστών παρακολουθεί ένα μικρό μέρους του βίντεο το οποίο μπορεί να αντιστοιχεί ακόμα και σε μερικά δευτερόλεπτα.

Τα σημαντικότερα συμπεράσματα που προκύπτουν από την μελέτη στο Κεφάλαιο 3 φαίνονται αναλυτικά παρακάτω:

1. Η μερική αποθήκευση των βίντεο (partial video caching) είναι αποδοτική όταν το μεγαλύτερο ποσοστό των χρηστών παρακολουθεί μερικώς τα βίντεο.

2. Όταν οι περισσότεροι χρήστες παρακολουθούν μερικώς τα βίντεο είναι προτιμότερο να τμηματοποιείται το βίντεο με βάση το μέγεθός του και όχι με βάση κάποιο σταθερό αριθμό από blocks. Όταν το βίντεο τμηματοποιείται σε αρκετά τμήματα και επιπλέον κατά την αντικατάσταση εξετάζονται πρώτα ως βίντεο θύματα τα βίντεο με αρκετά τμήματα αποθηκευμένα στη μνήμη, τότε στη μόνιμη κατάσταση του συστήματος θα υπάρχουν περισσότερα μερικώς αποθηκευμένα βίντεο και έτσι μπορούν να εξυπηρετηθούν αρκετές αιτήσεις οι οποίες σταματάνε στα αρχικά frames.

3. Η μερική αποθήκευση των βίντεο είναι προτιμότερη όταν οι περισσότεροι χρήστες παρακολουθούν μερικώς τα βίντεο και επιπλέον οι προτιμήσεις των χρηστών είναι αρκετά διαφορικές. Όπως φαίνεται από τα σενάρια 1 και 2, η *"Aggressive and guarded method"* υπερτερεί των άλλων μεθόδων σε αυτές τις συνθήκες γιατί με τα περισσότερα μερικώς αποθηκευμένα βίντεο που πετυχαίνει, μπορεί να εξυπηρετήσει το μεγαλύτερο ποσοστό των χρηστών ειδικά όταν οι χρήστες έχουν διαφορετικές προτιμήσεις.

4. Η πλήρη αποθήκευση των βίντεο (full video caching) είναι αποδοτική όταν το μεγαλύτερο ποσοστό των χρηστών παρακολουθεί ολόκληρο το βίντεο. Αυτό ισχύει και όταν οι προτιμήσεις των χρηστών είναι αρκετά διαφορετικές. Αυτό το συμπέρασμα προκύπτει και από το Κεφάλαιο 2.

5. Η απόδοση κάθε αλγορίθμου δεν εξαρτάται μόνο από την μέθοδο διαχείρισης της μνήμης που αυτός υλοποιεί, αλλά πρωταρχικό ρόλο παίζει το αν οι χρήστες επιλέγουν να παρακολουθήσουν ολόκληρο το βίντεο ή μόνο κάποιο αρχικό μέρος του.

6. Ο “Lazy Segmentation” αλγόριθμος δεν είναι αποδοτικός όταν το μέγεθος των βίντεο είναι μεγάλο και κατανέμεται σύμφωνα με την ομοιόμορφη κατανομή.

Κεφάλαιο 4 : Επίλογος

Στο Κεφάλαιο αυτό αναλύεται η συμβολή της διπλωματικής εργασίας και προτείνονται ιδέες για μελλοντικές επεκτάσεις.

4.1 Συνεισφορά διπλωματικής εργασίας

Στο πρώτο μέρος της διπλωματικής εργασίας (Κεφάλαιο 2), μέσα από τα αποτελέσματα των προσομοιώσεων φαίνεται ότι οι στρατηγικές εισαγωγής και αντικατάστασης που προτάθηκαν έχουν επιφέρει σημαντική βελτίωση στις μετρικές απόδοσης που μελετήσαμε σε σχέση με προηγούμενη έρευνα η οποία έγινε στο [2].

Αρχικά δείξαμε ότι το υψηλό BHR μπορεί να επιτευχθεί εισάγοντας λίγα βίντεο στη μνήμη του proxy server τα οποία θα ανήκουν στα δημοφιλή και θα αποθηκεύουν ολόκληρο το περιεχόμενό τους. Αυτό είναι προτιμότερο στην περίπτωση στην οποία ο χρήστης θα παρακολουθήσει σειριακά ολόκληρο το βίντεο. Αν αποθηκεύαμε αρκετά βίντεο εισάγοντας από το κάθε βίντεο λίγα τμήματα, όπως γίνεται στη μέθοδο στο [2], τότε η μετρική απόδοσης BHR δεν θα είχε τόσο καλά αποτελέσματα, γιατί συχνά δεν θα μπορούσε να εξυπηρετηθεί ολόκληρο το βίντεο μιας αίτησης από τον proxy server.

Ακολούθως μέσα από τις προσομοιώσεις φαίνεται ότι μειωμένες καθυστερήσεις κατά το ξεκίνημα των βίντεο μπορούν να επιτευχθούν με την πλήρη ανεξαρτησία στην διαχείριση του prefix και του suffix τμήματος της μνήμης. Οι μέθοδοι *"Full video with prefix"* και *"Full video with prefix and with limitation"* οι οποίες εφαρμόζουν αυτή την ανεξαρτησία, παρουσιάζουν πάντα καλύτερα αποτελέσματα στη μετρική καθυστερήσεις κατά το ξεκίνημα των βίντεο, σε σχέση με τις μεθόδους *"Full video with prefix in LRLFU"* και τη μέθοδο στο [2].

Συγκρίνοντας τις μεθόδους *"Full video with prefix"* και *"Full video with prefix and with limitation"* ως προς τη στρατηγική εισαγωγής στη μνήμη B, η οποία είναι η μόνη διαφορά τους, συμπεραίνουμε ότι είναι πιο αποδοτικό να ελέγχουμε αν ένα βίντεο πληροί κάποιες προϋποθέσεις για να αποθηκευτεί στη μνήμη του proxy server. Αν δεν υπάρχει κάποιος έλεγχος τότε το σύστημα οδηγείται σε αυξημένο overhead. Ο έλεγχος αυτός πρέπει να επιλεγεί προσεχτικά γιατί αν είναι αρκετά «αυστηρός», όπως γίνεται στη *"Full video with prefix in LRLFU",* τότε το σύστημα αργεί να προσαρμοστεί αν οι προτιμήσεις των χρηστών αλλάζουν απότομα ή είναι αρκετά διαφορετικές.

Στο δεύτερο μέρος της διπλωματικής εργασίας (Κεφάλαιο 3), από τα αποτελέσματα των προσομοιώσεων φαίνεται ότι οι στρατηγικές εισαγωγής και αντικατάστασης που προτάθηκαν έχουν επιφέρει σημαντική βελτίωση στις μετρικές απόδοσης που μελετήσαμε, σε σχέση με προηγούμενη έρευνα η οποία έγινε στο [3].

Μέσα από τις προσομοιώσεις φαίνεται ότι οι κινήσεις των χρηστών επηρεάζουν την αποδοτικότητα των μεθόδων. Μια μέθοδος μπορεί να παρουσιάζει καλά αποτελέσματα σε ένα σενάριο και σε ένα αρκετά διαφορετικό σενάριο η κατάσταση να αντιστραφεί.

Αρχικά δείξαμε ότι στην περίπτωση που το μεγαλύτερο ποσοστό των χρηστών παρακολουθεί μερικώς τα βίντεο, είναι προτιμότερο να έχουμε στη μνήμη του proxy server αρκετά μερικώς αποθηκευμένα βίντεο ώστε να μπορούν να εξυπηρετηθούν περισσότερες αιτήσεις. Το μέγεθος του τμήματος είναι πιο αποδοτικό να επιλέγεται με βάση τη διάρκεια του βίντεο και όχι με βάση ένα σταθερό αριθμό από blocks. Επιπλέον κατά την εφαρμογή της στρατηγικής αντικατάστασης είναι προτιμότερο να αφαιρούνται πρώτα κάποια τμήματα από τα βίντεο που έχουν αρκετά τμήματα στη μνήμη του proxy server, ώστε να έχουμε όσο περισσότερα μερικώς αποθηκευμένα βίντεο και να εξυπηρετούνται αρκετές αιτήσεις, οι οποίες παρακολουθούν ένα αρχικό τμήμα του βίντεο.

Όταν το μεγαλύτερο ποσοστό των χρηστών παρακολουθεί ολόκληρο το βίντεο τότε οι μέθοδοι οι οποίες δεν υλοποιούν κάποια τμηματοποίηση στο βίντεο αλλά εισάγουν ολόκληρο το περιεχόμενο σε μια αίτηση είναι αποδοτικές, ακόμα και αν οι προτιμήσεις των χρηστών είναι αρκετά διαφορετικές.

Τέλος μέσα από τις προσομοιώσεις φαίνεται ότι η "*Lazy Segmentation*" δεν είναι αποδοτική όταν το μέγεθος του βίντεο είναι μεγάλο και ομοιόμορφα κατανεμημένο. Όταν η μνήμη του proxy server είναι περιορισμένη και αποθηκεύονται μεγάλα βίντεο σε αυτήν, τότε η συγκεκριμένη μέθοδος τείνει να αποθηκεύει λίγα βίντεο ολόκληρα το οποίο δεν είναι αποδοτικό σε μερική παρακολούθηση των βίντεο. Όταν η μνήμη του proxy server είναι περιορισμένη είναι πιο αποδοτικό να ελέγχουμε αν το βίντεο πληροί κάποιες προϋποθέσεις πριν αποθηκευτεί, αντί να χειριζόμαστε τη μνήμη με επιθετικές στρατηγικές. Οι επιθετικές στρατηγικές είναι αποδοτικές αν εφαρμόζονται για την αποθήκευση ενός τμήματος από κάθε βίντεο, όπως γίνεται στις "*Aggressive and guarded method*" και "*Aggressive and Fixed Segmentation*".

4.2 Ιδέες μελλοντικών επεκτάσεων

Ενδιαφέρον θα παρουσίαζε η υλοποίηση αλγορίθμων που μπορούν να υποστηρίζουν περισσότερες κινήσεις των χρηστών όπως forward και rewind. Μπορεί να εξεταστεί αν οι μέθοδοι που εισάγουν ολόκληρο το περιεχόμενο των βίντεο, μπορούν να εξυπηρετήσουν μεγάλο μερίδιο χρηστών οι οποίοι θα χρησιμοποιούν όλους τους βαθμούς ελευθερίας κινήσεων. Είναι λογικό ότι ένα βίντεο που είναι ολόκληρο αποθηκευμένο μπορεί να εξυπηρετήσει όλες τις κινήσεις του χρήστη. Όμως οι συγκεκριμένες μέθοδοι μπορεί να κάνουν κατάχρηση της μνήμης και να αδυνατούν να υποστηρίξουν την εξυπηρέτηση αρκετών αιτήσεων που χρησιμοποιούν κινήσεις προς τα μπρός ή προς τα πίσω.

Θα μπορούσε επίσης οι μέθοδοι που υλοποιούν τμηματοποίηση στο βίντεο να προσπαθούν να εισάγουν το επόμενο τμήμα στηριζόμενες σε κάποιο μοντέλο πρόβλεψης και όχι σε κάθε καινούργια αίτηση για το βίντεο, όπως γίνεται σε όλες τις μεθόδους μας. Η μνήμη θα μπορούσε να χωριστεί σε δύο τμήματα. Το πρώτο τμήμα θα αφιερώνεται για την αποθήκευση των προθεμάτων και το δεύτερο τμήμα για την αποθήκευση των τμημάτων που προβλέπεται ότι θα ζητηθούν. Μια εναλλακτική πρόταση είναι ο χωρισμός της μνήμης σε τρία τμήματα. Το πρώτο τμήμα θα αφιερώνεται για την αποθήκευση των προθεμάτων, το δεύτερο τμήμα για την αποθήκευση των τμημάτων που προβλέπεται ότι θα ζητηθούν και το τρίτο τμήμα για την αποθήκευση τμημάτων που ήδη ζητήθηκαν και ανήκουν στα δημοφιλή. Ένα τέτοιο μοντέλο θα μπορούσε να εξεταστεί ως προς την αποδοτικότητά του να υποστηρίζει τυχαίες προσπελάσεις στο βίντεο.

Βιβλιογραφία

[1] K.-L. Wu, P. S. Yu, and J. L. Wolf, “Segment-based proxy caching of multimedia streams”, in Proc. of the 2001 International WWW Conference, Hong Kong, pages 36—44, 2001.

[2] A.Satsiou, M.Paterakis, “Impact of Frequency-Based Cache Management Policies on the Performance of Segment Based Video Caching Proxies”, in Proc. of the 2004 IFIP Networking Conference, Athens, Greece, pages 1120—1131, 2004.

[3] Songqing Chen, Bo Shen, Susie Wee and Xiaodong Zhang “Adaptive and Lazy Segmentation Based Proxy Caching for Streaming Media Delivery”, *NOSSDAV’03,* Monterey, California, USA, pages 22-31, June 1–3, 2003.

[4] Soam Acharya, Brian Smit, Peter Parnes “Characterizing User Access to Videos on the World Wide Web”. In Proc. SPIE/ACM MMCN, San Jose, USA, Jan. 2000.

[5] K.-L. Wu, P. S. Yu, and J. L. Wolf, “Segmentation of Multimedia Streams for proxy caching” , IEEE Trans. on Multimedia, 6(5), pages 770—780, Oct. 2004.

[6] http://en.wikipedia.org/wiki/Zipf's_law

[7] Wei Tu, Eckehard Steinbach, Muhammad Muhammad, and Xiaoling Li, “Proxy Caching for Video-on-Demand Using Flexible Starting Point Selection”, in IEEE Transactions on Multimedia, Vol. 11, No. 4, pages 716-729, June 2009.

[8] Ludmila Cherkasova and Minaxi Gupta, “Analysis of Enterprise Media Server Workloads: Access Patterns, Locality, Content Evolution, and Rates of Change”, in IEEE/ACM Transactions on Networking, Vol. 12, No. 5, pages 781-793, October 2004.

[9] Ludmila Cherkasova and Minaxi Gupta, “Characterizing Locality, Evolution, and Life Span of Accesses in Enterprise Media Server Workloads”*, NOSSDAV’02,* Miami, Florida, USA, May 2002.

[10] Changxi Zheng, Guobin Shen and Shipeng Li, “Distributed Prefetching Scheme for Random Seek Support in Peer-to-Peer Streaming Applications”, P2PMMS’05, Singapore, November 2005.

[11] S. Chen, H. Wang, X. Zhang, B. Shen, and S. Wee, “Segment-based proxy caching for internet streaming media delivery,” IEEE Multimedia, vol. 12, no. 3, pages 59–67, July 2005.

[12] M. Chesire, A.Wolman, G. M. Voelker, and H. M. Levy, “Measurement and analysis of a streaming media workload”, in Proc.3rd USENIX Symp. Internet Technologies and Systems, San Francisco, CA, pages 1–12, Mar.2001.

[13] Παντελής Σταμπουλή, “Δικτυακή Προσομοίωση με OMNeT++ Φιλτράρισμα Καθυστερημένων πακέτων RTP”, Πτυχιακή εργασία, Πανεπιστήμιο Μακεδονίας, pages 15-22, 2010.

[14] Γκάμας Απόστολος, “Υπηρεσίες και Πρωτόκολλα Πραγματικού Χρόνου”, Διπλωματική εργασία στα πλαίσια του μεταπτυχιακού διπλώματος , Πανεπιστήμιο Πατρών, pages 51-78, 2000.

[15] J. Liu, J. Xu, “Proxy caching for Media Streaming over the Internet”, IEEE Communications, pages 88-94, August 2004.

[16] H. Fahmi, M. Latif, S. Sedigh-Ali, A. Ghafoor, P. Liu, and L. Hsu, “Proxy servers for scalable interactive video support, IEEE Computer, pages 54-60, Sep. 2001.

[17] S. Sen, J. Rexford, and D. Towsley, “Proxy prefix caching for multimedia streams”, in Proc. IEEE INFOCOM Conf., pages 1310-1319, 1999.

[18] S. P. Ponnusamy, E. Karthikeyan, “An Observed Study on Improved Caching by Adaptive and Partial Aggressive Prefetching”, in International Journal Computer Science and Application, pages 190-194, vol 1, 2010.

[19] Junli Yuan, Qibin Sun, Susanto Rahardja, “A More Aggressive Prefetching Scheme for Streaming Media Delivery over the Internet”, in Proc. Visual Communications and Image Processing, vol. 6508, pp. 650818-1–8, Feb. 2007.

[20] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, “Understanding User Behavior in Large-Scale Video-on-demand Systems”, In EuroSys’06, pages 333-344,2006.

[21] C. Costa, I. Cunha, A. Borges, C. Ramos, M. Rocha, J. Almeida, and B. Ribeiro-Neto, “Analyzing client interactivity in streaming media”, In Proceedings of the 13th International World Wide Web Conference (WWW), New York, USA, pages 534-543,May 2004.