ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

&

ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Διπλωματική Εργασία

"Προσωρινή αποθήκευση σε κρυφή μνήμη (cache) για παροχή κατ' απαίτηση υπηρεσιών IPTV"

ΑΛΕΞΑΝΔΡΟΣ ΜΑΡΚΟΠΟΥΛΟΣ

Εξεταστική επιτροπή: Καθ. Μιχαήλ Πατεράκης (Επιβλέπων)

Καθ. Αθανάσιος Λιάβας

Αν. Καθ. Άγγελος Μπλέτσας

Χανιά, Κρήτη

2016

Περίληψη

Η προσωρινή αποθήκευση αντικειμένων πολυμέσων κοντά στους χρήστες σε ένα σύστημα παροχής υπηρεσιών κατ' απαίτηση σε δίκτυο IPTV μειώνει το φορτίο του δικτύου και βελτιώνει την καθυστέρηση παράδοσης των αντικειμένων. Η μερική προσωρινή αποθήκευση τμημάτων αντικειμένου video είναι αποτελεσματική λόγω του περιορισμένου χώρου της κρυφής μνήμης-cache και του γεγονότος ότι ορισμένα τμήματα του αντικειμένου έχουν διαφορετική δημοτικότητα από άλλα τμήματα του. Σε αυτή την εργασία προτείνουμε μια στρατηγική μερικής προσωρινής αποθήκευσης, η οποία λαμβάνει υπόψη της αλλαγές στη δημοτικότητα των αντικειμένων με την πάροδο του χρόνου για τον υπολογισμό της χρησιμότητας των διαφόρων τμημάτων των video. Χρησιμοποιούμε διαμερισμό της κρυφής μνήμης-cache για να αποφύγουμε την έξοδο από αυτήν δημοφιλών αντικειμένων που παροδικά δε ζητούνται συχνά, εξαιτίας μη δημοφιλών αντικειμένων που παροδικά ζητούνται με μεγαλύτερη συχνότητα. Τα αποτελέσματα της μελέτης μας μέσω προσομοίωσης δείχνουν ότι το προτεινόμενο σύστημα προσωρινής αποθήκευσης βελτιώνει το Byte Hit Ratio και μειώνει το κλάσμα των αιτήσεων που εξυπηρετούνται με καθυστέρηση (Delayed Starts) συγκριτικά με τις τεχνικές προσωρινής αποθήκευσης Least Recently Used (LRU) και Least Recently Least Frequently Used (LRFLU) στις παρακάτω περιπτώσεις που εξετάστηκαν: i) με στατικό εξυπηρετητή παροχής αντικειμένων video (ο εξυπηρετητής περιέχει ένα σταθερό αριθμό αντικειμένων video, των οποίων οι δημοτικότητες δε μεταβάλλονται με το χρόνο), ii) με δυναμικό εξυπηρετητή παροχής αντικειμένων video στην διάρκεια λειτουργίας του εξυπηρετητή (έχουμε εισαγωγή νέων αντικειμένων video σε αυτόν με παράλληλη απομάκρυνση παλιότερων αντικειμένων) και iii) με επίδραση του ρυθμού γήρανσης στις δημοτικότητες των αντικειμένων video (η δημοτικότητα των αποθηκευμένων αντικειμένων video στον εξυπηρετητή αλλάζει με την πάροδο του χρόνου).

Ευχαριστίες

Με τη Διπλωματική Εργασία ολοκληρώνονται οι σπουδές μου στη σχολή Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών του Πολυτεχνείου Κρήτης. Θα ήθελα, λοιπόν, με την αφορμή αυτή, να ευχαριστήσω όλους εκείνους που στάθηκαν δίπλα μου σε ολόκληρη τη φοιτητική μου πορεία. Καταρχάς, θα ήθελα να ευχαριστήσω την οικογένειά μου, που μου έδωσε τη δυνατότητα να σπουδάσω, και που πάντα με στήριζε και με στηρίζει στις επιλογές μου. Θα ήθελα ακόμη να ευχαριστήσω όλους τους συμφοιτητές και φίλους μου, καθώς, χωρίς τη συνεργασία, την αλληλοϋποστήριξη και την ανταλλαγή ιδεών και εμπειριών, θα ήταν αδύνατη η περάτωση μιας τόσο δύσκολης Σχολής. Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου εργασίας κ. Μιχάλη Πατεράκη που μου έδωσε την δυνατότητα να ασχοληθώ με κάτι που μου αρέσει και με ενδιαφέρει, καθώς επίσης και για τις πολύτιμες συμβουλές και το ενδιαφέρον του. Ακόμη, θα ήθελα να ευχαριστήσω τα μέλη της εξεταστικής μου επιτροπής κ. Αθανάσιο Λιάβα και κ. Άγγελο Μπλέτσα για το χρόνο που διέθεσαν να διαβάσουν και να εξετάσουν την διπλωματική μου εργασία.

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Κεφάλαιο 1

1.1 Εισαγωγή

Το Διαδίκτυο έχει αναδειχθεί σε κύριο μέσο για τηλεοπτικές και ραδιοφωνικές εκπομπές, ταινίες και ανταλλαγή video για προσωπική και επαγγελματική χρήση. Η εμφάνιση ιστοσελίδων όπως το Netflix και το Hulu, οι οποίες προσφέρουν μετάδοση τηλεοπτικών προγραμμάτων, έχει κάνει το διαδίκτυο μια σημαντική πηγή για την ψηφιακή ψυχαγωγία και ενημέρωση. Η αυξανόμενη χρήση και η δημοτικότητα παρόχων συνεχούς ροής πολυμέσων μεταξύ των χρηστών είναι στενά συνδεδεμένη με την ανάπτυξη των ευρυζωνικών (broadband) δικτύων. Τα ευρυζωνικά δίκτυα χαρακτηρίζονται από μεγάλες χωρητικότητες και ταχεία μετάδοση πληροφορίας. Διάφορες υπηρεσίες παροχής πολυμέσων κατόπιν αιτήματος έχουν προταθεί για ευρυζωνικά δίκτυα [14]. Ωστόσο, ένα δίκτυο που χρησιμοποιεί το διαδικτυακό πρωτόκολλο IP (Internet Protocol) είναι κατάλληλο για υποστήριξη εξατομικευμένων υπηρεσιών, επειδή επιστρέφει ένα φυσικό κανάλι, που μεταφέρει εύκολα τις αιτήσεις των χρηστών στο κέντρο ελέγχου της υπηρεσίας, και επειδή μπορεί άμεσα να διαχειριστεί τον χρήστη, αφού έχει στη διάθεση του τη διεύθυνση IP του [15].

Τις προηγούμενες δεκαετίες, το τηλεοπτικό σήμα μεταδιδόταν μέσω επίγειων συστημάτων («ερτζιανά κύματα»), τηλεοπτικών δορυφόρων ή καλωδίου. Επιχειρήσεις κάθε είδους, ανεξαρτήτως κλάδου δραστηριοποίησης, χρησιμοποίησαν τις δυνατότητες του νέου μέσου, για να προβάλλουν τις υπηρεσίες και τα προϊόντα τους. Τα τελευταία χρόνια, με την εξάπλωση του Διαδίκτυου και κυρίως με την επίτευξη πολύ υψηλών ταχυτήτων μεταγωγής δεδομένων, την αύξηση των χρηστών που συνδέονται στο Διαδίκτυο και τη συνεχή μείωση του σχετικού κόστους, αναπτύχθηκε η τεχνολογία IPTV. Με πιο τεχνικούς όρους, η διαδικτυακή τηλεόραση περιγράφεται ως ένα σύστημα κατά το οποίο ψηφιακό τηλεοπτικό σήμα εκπέμπεται σε χρήστες του Διαδικτύου με τη βοήθεια του πρωτοκόλλου IP και μιας ευρυζωνικής σύνδεσης. Η υπηρεσία αυτή, συχνά, παρέχεται σε συνδυασμό με video κατ' απαίτηση (video on demand) και μπορεί να περιλαμβάνει ταυτόχρονα και άλλες διαδικτυακές υπηρεσίες, όπως για παράδειγμα το λεγόμενο triple play (internet, τηλέφωνο και βίντεο). Το τηλεοπτικό σήμα που φέρει το περιεχόμενο είναι, συνήθως, κωδικοποιημένο σε μορφή MPEG2 και διανέμεται μέσω IP Multicast, μέθοδος με την οποία η πληροφορία μπορεί

να αποσταλεί ταυτόχρονα σε πολλούς αποδέκτες/υπολογιστές με το πρότυπο H.264. Αυτό που χαρακτηρίζει το IPTV είναι ότι παρέχει ένα ευρύ σύνολο από ετερογενείς υπηρεσίες, οι οποίες διαφοροποιούνται σημαντικά από την παραδοσιακή ψηφιακή τηλεόραση, κυρίως σε ότι αφορά τις υπηρεσίες «on demand» και τις διαδραστικές υπηρεσίες (interactive).

Οι υπηρεσίες αυτές διαχωρίζονται σε:

- Broadcast υπηρεσίες: Οι «broadcast» υπηρεσίες υλοποιούν την ταυτόχρονη μετάδοση περιεχομένου στο σύνολα των συνδρομητών με πλεονέκτημα την μικρότερη επιβάρυνση για το δίκτυο. Είναι αυτές που βρίσκονται πιο κοντά στο παραδοσιακό μοντέλο του broadcast TV.
- On demand υπηρεσίες: Αντίθετα με τις broadcast υπηρεσίες, οι «on demand» υπηρεσίες βασίζονται στη ξεχωριστή μετάδοση περιεχομένου προς τον χρήστη, μόνο αφού ο ίδιος εκφράσει ενδιαφέρον για τη συγκεκριμένη υπηρεσία.

Γενικά, το IPTV φαίνεται ότι γρήγορα γίνεται ένα δημοφιλές μέσο για την μεταφορά υπηρεσιών ψηφιακής τηλεόρασης στους συνδρομητές. Όμως, λόγω της φύσης του IPTV, απαιτεί ένα γρήγορα δίκτυο για την μεταφορά των δεδομένων του με ικανοποιητική ταχύτητα προς τους συνδρομητές. Ο κύριος σκοπός αυτού του δικτύου είναι να μεταφέρει τα bits των δεδομένων μεταξύ του κέντρου δεδομένων, του παρόχου υπηρεσίας και του IPTV set-top box του συνδρομητή και πρέπει να γίνεται με τέτοιο τρόπο, ώστε να μην επηρεάζεται η ποιότητα του video που εκπέμπεται στο συνδρομητή. Σε ένα IPTV δίκτυο, η παροχή υπηρεσιών κατά απαίτηση δημιουργεί την ανάγκη μεταφοράς μιας μεγάλης ποσότητας δεδομένων από τα κεντρικά γραφεία της εκάστοτε υπηρεσίας (Video Head Office-VHO) στους συνδρομητές, καταναλώνοντας ένα σημαντικό μέρος των πόρων του δικτύου. Η μετάδοση πολυμέσων συνεχούς ροής απαιτεί ένα υψηλό και σταθερό ρυθμό μετάδοσης. Όμως, λόγω του μεγάλου μεγέθους τους, η μετάδοση πολυμέσων συνεχούς ροής δεσμεύει ένα μεγάλος μέρος του διαθέσιμου φάσματος (bandwidth) και αυτό είναι ένα από τα σημαντικότερα προβλήματα που αντιμετωπίζει σήμερα το Διαδίκτυο. Σημαντικό ρόλο στην ανάπτυξη αυτών των προβλημάτων παίζει και η αύξηση των χρηστών που έχουν πρόσβαση στο Διαδίκτυο. Στην εργασία αυτή θα ασχοληθούμε με «on demand» υπηρεσίες, που επιτρέπουν στους χρήστες να επιλέξουν μέσω ενός διαδραστικού συστήματος,

ανάμεσα σε μια πληθώρα από αποθηκευμένα videos και να τα παρακολουθήσουν μέσω του δικτύου.

Σε τέτοιες εφαρμογές, η προσωρινή αποθήκευση αντικειμένων πολυμέσων σε σημεία κοντά στους χρήστες μειώνει σημαντικά την κίνηση στο δίκτυο και την καθυστέρηση προβολής των videos στους χρήστες, ως εκ τούτου μειώνει το κόστος παροχής υπηρεσιών κατ' απαίτηση. Σε μια τυπική αρχιτεκτονική IPTV, τα αντικείμενα video είναι αποθηκευμένα σε κρυφές μνήμες, που είναι τοποθετημένες κοντά στους συνδρομητές, είτε σε πολυπλέκτες/αποπολυπλέκτες (DSLAM) ψηφιακών συνδρομητικών γραμμών DSL, είτε σε κεντρικά γραφεία (COS), είτε σε ενδιάμεσα γραφεία (IOS)[10]. Οι κρυφές μνήμες αναλαμβάνουν να παίξουν το ρόλο του ενδιάμεσου ανάμεσα στον χρήστη και στους κεντρικούς υπολογιστές (servers), από τον οποίο ο χρήστης ζητά να δει κάποια αντικείμενα video [12]. Έτσι προκειμένου να διεκπεραιώσει το αίτημα του χρήστη, η κρυφή μνήμη-cache, λειτουργώντας ως διεργασία αντιπρόσωπος, αρχικά ελέγχει αν διαθέτει την πληροφορία που ζητείται. Αν την διαθέτει την στέλνει κατευθείαν στον χρήστη, χωρίς να χρειάζεται ο τελευταίος να περιμένει να λάβει την πληροφορία από τους κεντρικούς υπολογιστές. Αν η ζητούμενη πληροφορία δεν είναι διαθέσιμη, τότε η μνήμη cache αναλαμβάνει να την «κατεβάσει» και ακολούθως, την στέλνει στον χρήστη, ενώ ταυτόχρονα την αποθηκεύει προσωρινά, για τη περίπτωση που ζητηθεί ξανά στο εγγύς μέλλον.

Οι παραδοσιακές στρατηγικές προσωρινής αποθήκευσης πληροφορίας σε κρυφές μνήμες (caches) έχουν αποδειχθεί αναποτελεσματικές για την περίπτωση αντικειμένων video, κυρίως λόγω του μεγάλου μεγέθους των videos και των διαφορετικών προτύπων πρόσβασης των αντικείμενων [5]. Λόγω του μεγάλου μεγέθους των αντικειμένων video, η προσωρινή αποθήκευση ολόκληρου του αντικειμένου σε μια περιορισμένου μεγέθους κρυφή μνήμη (cache) θα ήταν αναποτελεσματική, ιδιαίτερα στην περίπτωση όπου ο χρήστης παρακολουθήσει μόνο ένα μικρό μέρος του video, προτού διακόψει τη σύνδεση του. Στρατηγικές μερικής προσωρινής αποθήκευσης αντικειμένων video, που έχουν προταθεί στην βιβλιογραφία, έχει αποδειχθεί ότι έχουν υψηλότερες αποδόσεις από τεχνικές που αποθηκεύουν ολόκληρα τα αντικείμενα, καθώς η ίδια έκταση της κρυφής μνήμης θα μπορούσε να ικανοποιήσει περισσότερα αιτήματα χρηστών. Σε ένα τυπικό σύστημα παροχής υπηρεσιών κατ' απαίτηση, η πιθανότητα ζήτησης ενός αντικειμένου ή ενός τμήματος του αυξάνει με τη δημοτικότητα του. Ορισμένες στρατηγικές μερικής προσωρινής

αποθήκευσης εκτός της δημοτικότητας, λαμβάνουν επίσης υπόψη την τελευταία φορά που το συγκεκριμένο αντικείμενο ζητήθηκε από κάποιον χρήστη, καθώς και τη συχνότητα των αιτήσεων των χρηστών για το αντικείμενο. Στις περισσότερες από αυτές τις στρατηγικές θεωρείται ότι δεν έχουμε μεταβολές στη δημοτικότητα των videos, αλλά ότι οι πιθανότητες αίτησης τους από τους χρήστες παραμένουν σταθερές. Όμως, σε ένα πραγματικό σύστημα παροχής υπηρεσιών κατ' απαίτηση, η δημοτικότητα των αντικειμένων μεταβάλλεται με την πάροδο του χρόνου. Μια ταινία μπορεί να θέλει ένα χρονικό διάστημα για να γίνει δημοφιλής και από την άλλη πλευρά, μια δημοφιλής ταινία μπορεί να χάσει τη δημοτικότητα της με το πέρασμα του χρόνου. Στην περίπτωση που το video χωρίζεται σε ένα σταθερό αριθμό τμημάτων, η παρακολούθηση των αλλαγών στη δημοτικότητα των τμημάτων του είναι σημαντική για τον αποτελεσματικό σχεδιασμό συστημάτων προσωρινής αποθήκευσης.

Στην εργασία αυτή παίρνουμε υπόψη τη συχνότητα με την οποία εμφανίζεται και το πόσο πρόσφατη είναι μια αίτηση για τη προσωρινή αποθήκευση ενός τμήματος κάποιου video και προτείνουμε μία μέθοδο που υπολογίζει δυναμικά τη χρησιμότητα του τμήματος, καθώς η δημοτικότητα των videos μεταβάλλεται. Για να αποφύγουμε να αποβληθούν από τη κρυφή μνήμη παλαιότερα δημοφιλή αντικείμενα και την θέση τους να πάρουν αντικείμενα με ξαφνική δημοτικότητα, προτείνουμε το διαμερισμό της κρυφής μνήμης σε δύο επιμέρους caches. Κάθε cache χρησιμοποιεί διαφορετική μέθοδο για τον υπολογισμό της χρησιμότητας ενός τμήματος που είναι αποθηκευμένα σε αυτήν. Τέλος, εξαιτίας της έλλειψης προσαρμοστικότητας στις δυναμικές αλλαγές της δημοτικότητας των αντικειμένων και των προτύπων πρόσβασης των χρηστών, τα υπάρχοντα στη βιβλιογραφία συστήματα μερικής αποθήκευσης δεν μπορούν να εγγυηθούν συνεχή παράδοση των αντικειμένων στους χρήστες, επειδή συχνά τα ζητούμενα τμήματα των videos δεν είναι αποθηκευμένα στις κρυφές μνήμες-caches, όταν ζητούνται. Το κλειδί για να μειώσουμε τις καθυστερήσεις των χρηστών είναι να φέρουμε έγκαιρα στη cache τα ζητούμενα τμήματα του video, ώστε οι χρήστες να παρακολουθούν το μέρος του video που επιθυμούν χωρίς καθυστερήσεις και διακοπές.

1.2 Σχετική έρευνα

Το σύστημα το οποίο παρουσιάζεται στην εργασία [1] αποτελεί τη βάση των ιδεών και της μελέτης που αναπτύσσεται σε αυτήν την εργασία. Βασικές ιδέες όπως: i) η τμηματοποίηση των videos σε τμήματα σταθερού και μέτριου μεγέθους, ii) η διαμέριση της μνήμης cache σε δύο επιμέρους caches, iii) η μέθοδος με την οποία υπολογίζεται η χρησιμότητα των τμημάτων που αποθηκεύονται στην $Cache_1$ και iv) η μελέτη μέσω προσομοίωσης της απόδοσης του συστήματος, τόσο στην περίπτωση στατικού, όσο και στην περίπτωση δυναμικού server, προέρχονται από την παραπάνω εργασία. Επιπλέον των παραπάνω, στην εργασία μας τροποποιήσαμε κατάλληλα τη μέθοδο με την οποία υπολογίζεται η χρησιμότητα των τμημάτων που αποθηκεύονται στην $Cache_2$ και χρησιμοποιήσαμε κάποιες επιπλέον λειτουργίες όσο αφορά στη διαχείριση των αιτήσεων από τις δύο caches όπως: i) κάθε τμήμα αντικειμένου εισέρχεται στην $Cache_1$ την χρονική στιγμή που χρειάζεται να το δει ο χρήστης που υπέβαλε την αίτηση, ii) όταν το σύστημα γνωρίζει ότι κάποιο τμήμα αντικειμένου θα παρακολουθηθεί από κάποιον χρήστη (το οποίο επιτυγχάνεται μέσω ελέγχου μιας λίστας αναμονής) τότε το τμήμα αυτό δεν απομακρύνεται από την cache. Τέλος, στην εργασία μας συγκρίναμε το προτεινόμενο σύστημα προσωρινής αποθήκευσης με τις τεχνικές μερικής προσωρινής αποθήκευσης LRLFU και LRU, εξετάζοντας τις μετρικές Byte Hit Ratio και Delayed Starts.

Στην εργασία [2] προτείνεται ένα σχήμα lazy segmentation, το οποίο αποθηκεύει προσωρινά ολόκληρο το αντικείμενο video και μετά υπολογίζει το μήκος των τμημάτων που θα κρατήσει στην cache, βασιζόμενο στο μέσο μήκος του video που παρακολουθείται και τον αριθμό των προσβάσεων. Τα αντικείμενα αποκτούν χρησιμότητα σύμφωνα με τους παράγοντες που αναφέραμε παραπάνω και τα τμήματα των αντικειμένων με τη μικρότερη χρησιμότητα, αν χρειασθεί, οδηγούνται εκτός cache. Η παραπάνω μέθοδος μπορεί να μετατραπεί σε προσωρινή αποθήκευση ολόκληρων αντικειμένων, αν ο μέσος χρόνος αναπαραγωγής του video είναι σχεδόν ίσος με το πλήρες μήκος του ή δεν ακολουθήσει επόμενη αίτηση χρήστη για το συγκεκριμένο video.

Η εργασία [3] προτείνει την ανάγκη μερικής προσωρινής αποθήκευσης αντικειμένων σε ένα κυψελωτό δίκτυο κινητής τηλεφωνίας. Υπολογίζει τη χρησιμότητα κάθε τμήματος ενός video με βάση το πόσο πρόσφατη είναι η αίτηση σε

σχέση με την αντίστοιχη αμέσως προηγούμενη και τον αριθμό των αιτήσεων που έχει δεχθεί το τμήμα και στη συνέχεια, υπολογίζει προσαρμοστικά τα μεγέθη των τμημάτων του video που πρόκειται να αποθηκευτούν προσωρινά στη κρυφή μνήμη. Ωστόσο, δε διαμερίζει τη κρυφή μνήμη σε δύο μέρη και, όπως αναφέραμε, τα μεγέθη των τμημάτων των αντικειμένων που αποθηκεύονται δεν είναι σταθερά.

Η εργασία [4] υπογραμμίζει την ανάγκη ένα σύστημα παροχής υπηρεσιών κατ' απαίτηση να εξετάζει την ευμετάβλητη φύση της δημοτικότητας των αντικειμένων video. Το προτεινόμενο σύστημα εμφανίζει καλύτερες επιδόσεις από τις τεχνικές LRU και LFU. Ωστόσο, το προτεινόμενο σύστημα δε λαμβάνει υπόψη του αλλαγές στη δημοτικότητα των τμημάτων των αντικειμένων (τμήματα του ίδιου αντικειμένου έχουν ίδια δημοτικότητα) και δεν υποστηρίζει αποτελεσματικά δυναμικές αλλαγές στα πρότυπα προβολής των χρηστών (play, stop, seek).

Η εργασία [7] προτείνει τον χωρισμό ενός αντικειμένου σε τμήματα διαφορετικού μήκους. Θεωρεί ότι η δημοτικότητα των αντικειμένων είναι αντιστρόφως ανάλογη με τα μήκη τους. Χρησιμοποιεί δύο μεθόδους: i) pyramid τμηματοποίηση, όπου τα μήκη των τμημάτων μεγαλώνουν εκθετικά και ii) skyscraper τμηματοποίηση, όπου τα μήκη των τμημάτων μεγαλώνουν σταδιακά. Η εργασία διαμερίζει την κρυφή μνήμη σε δύο επιμέρους caches, μία για την αποθήκευση ενός αριθμού προθεμάτων σταθερού μήκους με σκοπό την μείωση της καθυστέρησης εκκίνησης των χρηστών και την δεύτερη για την αποθήκευση τμημάτων μεταβλητού μήκους. Η πρώτη cache χρησιμοποιεί πολιτική αντικατάστασης LRU και η δεύτερη μία πολιτική αντικατάστασης με βάση την προτεραιότητα των τμημάτων. Η βασική ιδέα της δεύτερης πολιτικής αντικατάστασης είναι ότι τα αρχικά τμήματα των videos έχουν προτεραιότητα στην προσωρινή αποθήκευση τους στην μνήμη cache. Το προτεινόμενο σύστημα δεν λαμβάνει υπόψη του αλλαγές στα πρότυπα προβολής των χρηστών με το πέρασμα του χρόνου.

Η εργασία [8] προτείνει μία συνδυαστική στρατηγική προσωρινής αποθήκευσης και ενός σχήματος προφόρτωσης (prefetching) αντικειμένων στην μνήμη. Πιο συγκεκριμένα, διαμερίζει την κρυφή μνήμη σε δύο επιμέρους caches, η μία cache προφορτώνει ένα μέρος των πιο δημοφιλών αντικειμένων και η άλλη αποθηκεύει προσωρινά αντικείμενα χρησιμοποιώντας ως μέθοδο αντικατάστασης την LRU.

Ένα άλλο σχήμα προσωρινής μερικής αποθήκευσης ονομάζεται proportional segmentation και χρησιμοποιείται κυρίως σε συστήματα peer-to-peer εμφανίζεται στο [9]. Η εν λόγω μέθοδος τμηματοποίησης αποθηκεύει αρχικά ένα πρώτο τμήμα του video και στη συνέχεια, αυξάνει το μέγεθος του video που αποθηκεύεται σύμφωνα με τον αριθμό προσβάσεων, το μέσο μήκος τμήματος που παρακολουθήθηκε και το τρέχον μήκος τμήματος που είναι ήδη αποθηκευμένο στην κρυφή μνήμη. Ωστόσο, η μέθοδος αυτή δεν εξετάζει την μεταβλητή φύση της δημοτικότητας των αντικειμένων και αυξάνει αργά τα μήκη των τμημάτων που αποθηκεύονται με τη πάροδο του χρόνου.

Η εργασία [11] ήταν ίσως η πρώτη που πρότεινε μερική προσωρινή αποθήκευση αντικειμένων και μάλιστα με τη τεχνική προθέματος για την αποθήκευση των αρχικών τμημάτων δημοφιλών αντικειμένων video στη cache, ώστε να μειωθεί η καθυστέρηση εκκίνησης που βιώνει ο χρήστης και να μειωθεί το φορτίο του δικτύου. Ωστόσο, η εργασία αυτή δεν επικεντρώθηκε στη μείωση του φορτίου του δικτύου μέσω της βελτίωσης της μετρικής byte hit ratio.

Ένας τρόπος για τη βελτίωση της αποτελεσματικότητας των μεθόδων αντικατάστασης στην προσωρινή μνήμη είναι να ενσωματωθεί στις μεθόδους η δυνατότητα πρόβλεψης της δημοτικότητας κατά την απόφαση της αντικατάστασης. Η εργασία [13] προτείνει τη μέθοδο P-LFU, μια προσαρμογή της LFU, η οποία καθορίζει ποιό αντικείμενο θα απομακρυνθεί από τη μνήμη cache με βάση την προβλεπόμενη μελλοντική του ζήτηση. Τέσσερις γενικές μεθόδους (γραμμική, power-law, εκθετική και Gaussian) έχουν χρησιμοποιηθεί για την πρόβλεψη της μελλοντικής ζήτησης ενός αντικειμένου, με την εκθετική να προκύπτει ότι δίνει τα καλύτερα αποτελέσματα. Οι συγγραφείς της εργασίας δείχνουν ότι η συγκεκριμένη μέθοδος πρόβλεψης της δημοτικότητας μπορεί να αυξήσει το cache hit ratio σε σύγκριση με την μέθοδο LFU.

Κεφάλαιο 2

2.1 Τοπολογία δικτύου

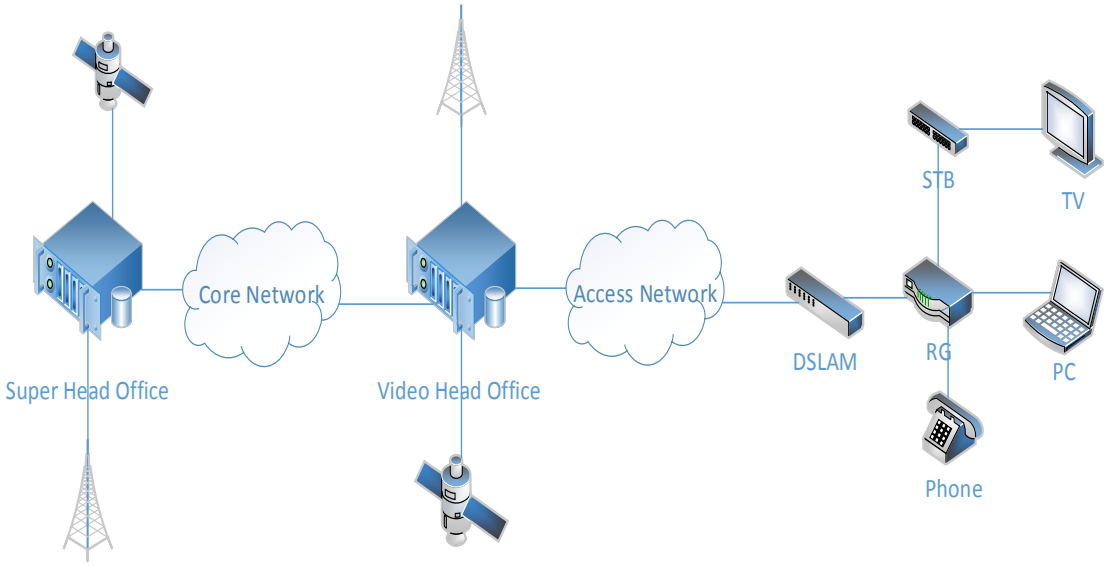
Το *[Σχήμα 1]* απεικονίζει την αρχιτεκτονική μιας IPTV υπηρεσίας που διανέμει αντικείμενα video στους οικιακούς χρήστες. Ένα τυπικό σύστημα IPTV χωρίζεται στα εξής επιμέρους τμήματα:

Ο SHO (Super Head Office) είναι η καρδιά του συστήματος IPTV. Εδώ συγκεντρώνεται το περιεχόμενο και διαμοιράζεται. Η συγκέντρωση του περιεχομένου γίνεται από διάφορες πηγές (studios, δορυφόρους, άλλες πηγές). Σε επόμενο στάδιο ο SHO κωδικοποιεί τα video συνεχούς ροής και τα μεταδίδει σε πολλαπλές υπηρεσίες VHO μέσω ενός IP δικτύου υψηλών ταχυτήτων.

Ο VHO (Video Head Office) είναι υπεύθυνος για μια ευρύτερη γεωγραφική περιοχή, συγκεντρώνει τα αντικείμενα πολυμέσων και τα μεταβιβάζει στους τελικούς χρήστες κατ' απαίτηση.

Ο CPE (Customer Premise Equipment) σε συνδυασμό με την καλωδίωση στο σπίτι αποτελεί το οικιακό δίκτυο. Τυπικά, η διάταξη περιλαμβάνει ένα ADSL Modem/Router (Residential Gateway - RG) κι ένα δέκτη IPTV (Set Top Box). Ο IPTV δέκτης (STB) αναλαμβάνει την αποκωδικοποίηση των IPTV ροών αντικειμένων, αλλά και την παροχή εξελιγμένων υπηρεσιών προστιθέμενης αξίας στον συνδρομητή. Το RG διαχειρίζεται επίσης, τα αιτήματα ελέγχου από κάθε Set Top Box πίσω στο δίκτυο διανομής.

Τα πολυμέσα, ανάλογα με τον πάροχο υπηρεσίας, περνούν από μια σειρά δρομολογητών (routers) ή διακοπτών (switches), όπως Digital Subscriber Line Access Multiplexers (DSLAM), ψηφιακών συνδρομητικών γραμμών DSL, Intermediate Offices (IOS) και Central Offices (COs), προτού φθάσουν στους τελικούς χρήστες. Στο σύστημα μας, οι χρήστες θα μπορούσαν να είναι συνδεδεμένοι μεταξύ τους με πολυπλέκτες *DSLAM* ψηφιακών συνδρομητικών γραμμών DSL. Σε ένα IPTV σύστημα τα πολυμέσα συνεχούς ροής κωδικοποιούνται από μια σειρά IP πακέτα και διακινούνται μέσω ευρυζωνικών δικτύων.



Σχήμα 1 Αρχιτεκτονική συστήματος IPTV

2.2 Μερική προσωρινή αποθήκευση αντικειμένων

Όπως αναφέραμε και στην εισαγωγή, η προσωρινή αποθήκευση ολόκληρου του αντικειμένου πολυμέσων εξαντλεί γρήγορα τη χωρητικότητα της κρυφής μνήμης, καθιστώντας την αναποτελεσματική, διότι το μέγεθος ενός αντικειμένου video είναι αρκετά μεγάλο. Οι καθυστερήσεις που συμβαίνουν κατά τη μεταφορά δεδομένων μέσω του Διαδικτύου είναι αποδεκτές σε περιηγήσεις αντικειμένων web. Ωστόσο, για τη ροή δεδομένων πολυμέσων ο χρήστης βιώνει αυτή την καθυστέρηση. Αυτό είναι ενοχλητικό και θα μπορούσε να οδηγήσει τους χρήστες μακριά από υπηρεσίες συνεχούς ροής πολυμέσων.

Υπάρχουν πολλές τεχνικές μερικής προσωρινής αποθήκευσης που διαιρούν τα αντικείμενα πολυμέσων σε μικρότερες μονάδες. Μερικές από τις πιο γνωστές στρατηγικές μερικής προσωρινής αποθήκευσης που προτείνονται για αποθήκευση πολυμέσων είναι η τεχνική αποθήκευσης προθέματος (prefix), η οποία αποθηκεύει τα αρχικά τμήματα του αντικειμένου, η τεχνική που αποθηκεύει διαφορετικά μεγέθη τμημάτων των αντικειμένων με βάση το χρονικό διαχωρισμό των αιτήσεων (sliding interval) και η τεχνική που αποθηκεύει τμήματα ίσου μεγέθους των αντικειμένων (segments) με βάση τη συχνότητα των αιτήσεων [5]. Η μέθοδος προθεματικής τμηματοποίησης μειώνει την καθυστέρηση εκκίνησης που βιώνει ο χρήστης, όμως δε

μειώνει το απαιτούμενο bandwidth του δικτύου αν ο χρήστης βλέπει διαφορετικά μέρη του video. Η μέθοδος sliding interval ομαδοποιεί τα αιτήματα των χρηστών που φθάνουν σε σύντομο χρονικό διάστημα για ένα αντικείμενο video και το τμήμα που είναι προσωρινά αποθηκευμένο απομακρύνεται από την cache, όταν ικανοποιηθεί η τελευταία αίτηση για αυτό. Η παραπάνω μέθοδος μπορεί να μετατραπεί σε προσωρινή αποθήκευση ολόκληρων αντικειμένων, αν το διάστημα της επόμενης αίτησης για το ίδιο video είναι μεγαλύτερο από τη διάρκεια αναπαραγωγής του. Η μέθοδος με προκαθορισμένα τμήματα ίσου μεγέθους, ενώ είναι μια γενικευμένη εκδοχή της μερικής προσωρινής αποθήκευσης, μπορεί να οδηγήσει σε βελτίωση των επιδόσεων της μνήμης cache, αν ληφθούν υπόψη τα πρότυπα προβολής των χρηστών.

Στην εργασία αυτή θεωρούμε ότι τα αντικείμενα είναι χωρισμένα σε ίσα τμήματα σταθερού μήκους. Σύμφωνα με συζητήσεις που έγιναν με μηχανικούς της Netflix, η σταθερή τμηματοποίηση με τμήματα μέτριου μεγέθους προτιμάται της προσαρμοστικής τμηματοποίησης, λόγω των ζητημάτων που προκύπτουν με την τελευταία στην πολυπλοκότητα της διαχείριση της μνήμης [1]. Ένα πρόβλημα με τη σταθερή τμηματοποίηση είναι ότι ο χρήστης αποκτά πρόσβαση σε αντικείμενα πολυμέσων με ασύμμετρο τρόπο (skewed pattern). Οι περισσότερες προσβάσεις είναι για λίγα δημοφιλή αντικείμενα και είναι πιθανόν οι χρήστες να τα παρακολουθήσουν στο σύνολο τους ή κοντά στο σύνολο τους. Αυτό είναι αρκετά συνηθισμένο για περιεχόμενο video σε υπηρεσίες video κατ' απαίτηση. Μία μέθοδος με προκαθορισμένο ομοιόμορφο ή εκθετικό μέγεθος τμηματοποίησης των αντικειμένων μπορεί να οδηγήσει σε ευνοϊκή προσωρινή αποθήκευση τμημάτων, χωρίς να λαμβάνει υπόψη ότι οι περισσότερες προσβάσεις στοχεύουν σε μερικά δημοφιλή αντικείμενα. Στην περίπτωση του δυναμικού server, οι δημοτικότητες αντικειμένων ή τμημάτων τους αλλάζουν συνεχώς με τον χρόνο. Για παράδειγμα, ένα αντικείμενο μπορεί να είναι δημοφιλές για σύντομο χρονικό διάστημα και οι χρήστες να το ζητήσουν ολόκληρο. Σε αυτό το σενάριο η χρήση μιας σταθερής στρατηγικής τμηματοποίησης μπορεί να επιβαρύνει το δίκτυο, καθώς για κάθε αντικείμενο που παροδικά είναι δημοφιλές, το σύστημα πρέπει να ανακτήσει τα περισσότερα τμήματα του αντικειμένου σε κάθε πρόσβαση του χρήστη.

Για την αποφυγή των παραπάνω προβλημάτων, λαμβάνουμε υπόψη στον υπολογισμό της χρησιμότητας των τμημάτων τον αριθμό των bytes που παρακολουθούνται από τους χρήστες από κάθε τμήμα. Με αυτό τον τρόπο,

διαφορετικά τμήματα του ίδιου αντικειμένου μπορεί να έχουν διαφορετική δημοτικότητα. Στην εργασία αυτή χρησιμοποιούμε το τμήμα (segment) ως βασική μονάδα των αντικειμένων (videos) για προσωρινή αποθήκευση και αντικατάσταση. Επίσης υποθέτουμε ότι το κάθε τμήμα αποτελείται από μικρότερες δομικές μονάδες (chunks), οι οποίες είναι χρήσιμες για τη μέτρηση των bytes που παρακολουθήθηκαν και τον υπολογισμό της χρησιμότητας του κάθε τμήματος μετά από κάθε αίτηση χρήστη.

2.3 Μέθοδοι αντικατάστασης LRU, LFU και LRLFU

Γνωστές στρατηγικές για την επιλογή του αντικειμένου ή τμήματος του αντικειμένου που αποθηκεύεται και αντικαθίσταται σε μία cache είναι οι εξής:

Λιγότερο προσφάτως χρησιμοποιημένο (least recently used-LRU). Ως κριτήριο για την επιλογή του τμήματος αντικατάστασης χρησιμοποιείται μόνο η χρονική στιγμή κατά την οποία κάθε τμήμα αντικειμένου που βρίσκεται στη μνήμη είχε την πιο πρόσφατη αίτηση από κάποιο χρήστη. Το τμήμα που ζητήθηκε πιο πρόσφατα έχει την υψηλότερη χρησιμότητα και τοποθετείται στην κορυφή της στοίβας, ενώ τα παλαιότερα τμήματα τοποθετούνται στο κάτω μέρος της στοίβας. Θύμα επιλέγεται εκείνο το αντικείμενο που δεν έχει ζητηθεί για μεγαλύτερο χρονικό διάστημα. Η χρησιμότητα ενός τμήματος δίνεται από την έκφραση $\text{Utility}_{\text{LRU}} = \frac{1}{\text{Tcurrent} - \text{Tlast accessed}}$, όπου Tcurrent είναι η τρέχουσα χρονική στιγμή και Tlast accesed είναι η χρονική στιγμή της τελευταίας αίτησης για το τμήμα. Η μέθοδος LRU δεν δίνει καλά αποτελέσματα όταν οι αιτήσεις των video από τους χρήστες δεν φθάνουν με χρονική ομοιομορφία. Πιο συγκεκριμένα, ένα αντικείμενο μπορεί να αποκτήσει μεγάλη χρησιμότητα μόνο μετά από δύο διαδοχικές αιτήσης για αυτό.

Λιγότερο συχνά χρησιμοποιημένο (least frequency used-LFU). Για την επιλογή του τμήματος αντικατάστασης, χρησιμοποιείται μόνο ο αριθμός των αιτήσεων για τα διάφορα τμήματα αντικειμένων που βρίσκονται στην cache. Θύμα επιλέγεται εκείνο το τμήμα που έχει τον μικρότερο αριθμό αιτήσεων σε κάποιο χρονικό διάστημα. Η μέθοδος LFU δεν δίνει καλά αποτελέσματα, όταν ο εξυπηρετητής είναι δυναμικός και η δημοτικότητα των αντικειμένων μειώνεται με το χρόνο. Ένας εξυπηρετητής είναι

δυναμικός, όταν το μέγεθος της βάσης δεδομένων του δεν είναι σταθερό, δηλαδή έχουμε εισαγωγή νέων αντικειμένων video και απομάκρυνση παλαιότερων αντικειμένων video.

Λιγότερο πιο πρόσφατα και συχνά χρησιμοποιημένο (least recently least frequency used-LRLFU) [16]. Η πολυπλοκότητα του αλγορίθμου είναι αυξημένη σε σχέση με αυτήν στους LRU και LFU, δεδομένου ότι χρησιμοποιεί ένα σύνθετο κριτήριο απόφασης για την επιλογή του τμήματος που θα βγει από την cache. Ο αλγόριθμος λαμβάνει υπόψη τη συχνότητα με την οποία ζητούνται τα διάφορα τμήματα και τις χρονικές στιγμές κατά τις οποίες κάθε τμήμα αντικειμένου που βρίσκεται στη μνήμη cache είχε την πιο πρόσφατη αναζήτηση από κάποιον χρήστη. Η χρησιμότητα ενός τμήματος δίνεται από την έκφραση $\text{Utility}_{\text{LRFLU}} = \frac{\text{RF}}{\text{Tcurrent} - \text{Tlast accessed}}$, όπου RF είναι ο μετρητής που μετρά τον αριθμό των φορών που το τμήμα έχει ζητηθεί από τη χρονική στιγμή που εισήχθη στη μνήμη cache μέχρι τον τρέχοντα χρόνο και Tcurrent, Tlast accessed είναι όπως ορίσθηκαν στην συζήτηση της μεθόδου LRU παραπάνω. Αν ένα τμήμα απομακρυνθεί ολοκληρωτικά από την cache, τότε ο αντίστοιχος μετρητής του αριθμού αναζητήσεων μηδενίζεται και όταν το τμήμα ξαναμπεί αργότερα στη μνήμη cache η καταγραφή του αριθμού των αναζητήσεων του θα ξεκινήσει από το ένα. Η μέθοδος αντικατάστασης που προτείνεται στην διπλωματική είναι παρόμοια με τον αλγόριθμο LRLFU.

2.4 Σχήμα διαμέρισης της κρυφής μνήμης

Η δημοτικότητα ενός αντικειμένου video όπως εκτιμάται στη μνήμη cache αυξάνεται με το πέρασμα του χρόνου, καθώς αυξάνεται ο αριθμός των φορών που αυτό ζητείται από τους χρήστες. Όμως, σε ένα σύστημα κατ' απαίτηση, η δημοτικότητα ενός αντικειμένου video μπορεί να μειωθεί με το χρόνο, αν οι χρήστες δεν το παρακολουθούν συχνά. Έτσι είναι απαραίτητο να εξεταστεί η τελευταία φορά που το αντικείμενο ζητήθηκε από κάποιον χρήστη, προτού ανανεωθεί η χρησιμότητα του αντικειμένου στην cache.

Συνήθως απλά συστήματα caching υποθέτουν σταθερές τις πιθανότητες ζήτησης των αντικειμένων και δε μεταβάλλουν τη δημοτικότητα των videos με το

πέρασμα του χρόνου (στατικός server). Συστήματα caching που προσπαθούν να μοντελοποιήσουν την γήρανση (μείωση) της δημοτικότητας των videos πρέπει να λαμβάνουν υπόψη τους τη συχνότητα με την οποία ζητούνται τα αντικείμενα video και τη χρονική στιγμή κατά την οποία ένα αντικείμενο στη μνήμη cache είχε την τελευταία αναζήτηση. Δεδομένων των παραπάνω, ένα αντικείμενο δεν θα πρέπει να εκδιωχθεί σε σύντομο χρόνο από την εισαγωγή του στην μνήμη cache, αλλά πρέπει να παραμείνει προσωρινά αποθηκευμένο σε αυτήν, ώστε ο υπολογισμός της χρησιμότητας να είναι αποτελεσματικός και να αποτυπώνει τις μεταβολές της δημοτικότητας των αντικειμένων. Νέα τμήματα αντικειμένων που μπαίνουν στη μνήμη cache με παροδικά υψηλή δημοτικότητα δεν θα πρέπει να απομακρύνουν από αυτήν τμήματα αντικειμένων που έχουν μακροπρόθεσμα μεγάλη δημοτικότητα.

Λαμβάνοντας υπόψη τα παραπάνω, προτείνουμε την διαίρεση της μνήμης cache σε δύο μέρη, $Cache_1$ για την πρώτη και $Cache_2$ για τη δεύτερη. Η $Cache_1$ χρησιμοποιείται για αποθήκευση τμημάτων που ζητούνται για πρώτη φορά (ή δε βρίσκονται στις κρυφές μνήμες) και η $Cache_2$ για αποθήκευση τμημάτων που βρίσκονται ήδη στην cache και η χρησιμότητα τους παραμένει υψηλή με το πέρασμα του χρόνου. Οι δύο μνήμες χρησιμοποιούν διαφορετικές μεθόδους υπολογισμού της χρησιμότητας των τμημάτων των αντικειμένων, διότι η έξωση τμημάτων από αυτές θα πρέπει να γίνεται με διαφορετικό τρόπο.

Ο υπολογισμός της χρησιμότητας ενός τμήματος στην $Cache_1$ βασίζεται στη συχνότητα ζήτησης και στο πόσο πρόσφατες είναι οι τελευταίες δύο αιτήσεις των χρηστών για το συγκεκριμένο τμήμα. Ο υπολογισμός αυτός είναι παρόμοιος με αυτόν στο κριτήριο χρησιμότητας της τεχνικής LRLFU, εμείς έχουμε όμως προσθέσει και μια επιπλέον παράμετρο που μετρά πόσο μέρος του τμήματος παρακολουθούν οι χρήστες. Όταν η τιμή χρησιμότητας κάποιου τμήματος ενός αντικειμένου στην $Cache_1$ περάσει ένα προκαθορισμένο κατώφλι (threshold), το τμήμα φεύγει από την $Cache_1$ και εισέρχεται στην $Cache_2$. Όπως θα δούμε και παρακάτω, ο υπολογισμός της χρησιμότητας ενός τμήματος στην $Cache_2$ εμπεριέχει τη παράμετρο πρόβλεψης της ζήτησης του τμήματος στο μέλλον. Επισημαίνουμε ότι τα τμήματα που είναι αποθηκευμένα στην $Cache_2$ είναι τμήματα αντικειμένων που έχουν μακροπρόθεσμα εμφανίσει μεγάλη δημοτικότητα, σε σχέση με τα τμήματα που είναι αποθηκευμένα στην $Cache_1$ τα οποία είναι τμήματα αντικειμένων που έχουν βραχυπρόθεσμα εμφανίσει μεγάλη δημοτικότητα.

Κεφάλαιο 3

3.1 Δυναμικό σχήμα προσωρινής αποθήκευσης

Το *[Σχήμα 2]* απεικονίζει την αρχιτεκτονική του συστήματος caching. Ο χρήστης έρχεται αρχικά σε επαφή με την $Cache_1$, ζητώντας ένα μέρος ή ολόκληρο το αντικείμενο video όπως θα το ζητούσε από τον απομακρυσμένο εξυπηρετητή στον οποίο βρίσκεται αποθηκευμένο ολόκληρο το ζητούμενο αντικείμενο. Τα πιθανά σενάρια που μπορούν να συμβούν με τη λήψη ενός τέτοιου αιτήματος από την $Cache_1$ είναι τα παρακάτω :

- *Το ζητούμενο μέρος του αντικειμένου δεν βρίσκεται στην $Cache_1$ ή στην $Cache_2$.*

Η $Cache_1$ πρέπει να προσκομίσει το ζητούμενο τμήμα από τον απομακρυσμένο εξυπηρετητή για λογαριασμό του χρήστη. Προς τούτο, η $Cache_1$ εγκαθιδρύει μια σύνδεση με τον απομακρυσμένο εξυπηρετητή και ο εξυπηρετητής μεταχειρίζεται την $Cache_1$ σαν ένα χρήστη.. Η $Cache_1$ διαβιβάζει τα λαμβανόμενα τμήματα του αντικειμένου video στον χρήστη, ενώ τα αποθηκεύει προσωρινά στην $Cache_1$ για πιθανή μελλοντική χρήση από άλλη αίτηση. Η παραπάνω προσέγγισή είναι σαφώς επιθετική για τα τμήματα των οποίων η χρησιμότητα δεν έχει ακόμα καθοριστεί και άρα, δεν είναι γνωστή στην $Cache_1$.

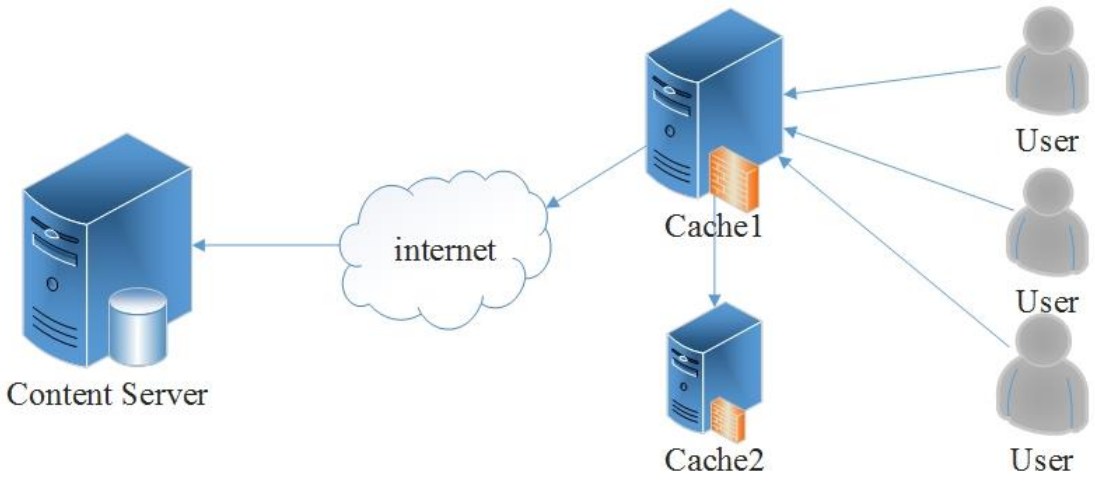
- *Το ζητούμενο μέρος του αντικειμένου βρίσκεται στην $Cache_1$.*

Η $Cache_1$ ξεκινά την παράδοση του τμήματος στον χρήστη που το ζήτησε. Το τμήμα αποκτά χρησιμότητα, η οποία καθορίζεται από το μέσο μήκος του τμήματος που έχει ιδωθεί από τους χρήστες, το πόσο πρόσφατη είναι η αίτηση χρήστη σε σχέση με την αντίστοιχη προηγούμενη για το συγκεκριμένο τμήμα και τον αριθμό των αιτήσεων που έχει δεχθεί το τμήμα κατά την παρουσία του στην $Cache_1$. Στη συνέχεια, ελέγχεται αν η χρησιμότητα του τμήματος έχει περάσει το προκαθορισμένο όριο, σε μια τέτοια περίπτωση το τμήμα απομακρύνεται από την $Cache_1$ και εισάγεται στην $Cache_2$. Τέλος, εφόσον το τμήμα παραμείνει στην $Cache_1$ ανανεώνεται ο χρόνος παραμονής του σε αυτήν.

- *Το ζητούμενο μέρος του αντικειμένου βρίσκεται στην $Cache_2$.*

Η $Cache_2$ ξεκινά την παράδοση του τμήματος στον χρήστη που το ζήτησε. Ανανεώνεται η χρησιμότητα του τμήματος η οποία εδώ εξαρτάται από το μέσο μήκος

του τμήματος που έχει ιδωθεί από τους χρήστες, την πιθανότητα να ζητηθεί το τμήμα σε επόμενο αίτημα χρήστη και τον αριθμό των αιτήσεων που έχει δεχθεί το τμήμα κατά την παρουσία του στις $Cache_1$ και $Cache_2$.



Σχήμα 2 Αρχιτεκτονική συστήματος Cache

3.1.1 Μέθοδος αντικατάστασης στην Cache1

Μετά από κάθε αίτηση που φθάνει για κάποιο τμήμα ενός αντικειμένου που βρίσκεται στην $Cache_1$, η τιμή της χρησιμότητας του ανανεώνεται. Δεδομένου ότι η δημοτικότητα του τμήματος δεν μπορεί να είναι γνωστή στο σύστημα, χρησιμοποιούμε μια συνάρτηση για τον υπολογισμό της χρησιμότητας, η οποία είναι ανάλογη με τον μέσο αριθμό των μικρών κομματιών (chunks) που έχουν ζητηθεί να παιχτούν από το τμήμα και το πόσο πρόσφατη είναι η αίτηση του τμήματος σε σχέση με την τελευταία αίτηση για το ίδιο τμήμα και αντιστρόφως ανάλογη με τον συνολικό αριθμό των μικρών κομματιών του τμήματος και τον αριθμό των αιτήσεων που έχει δεχθεί το τμήμα κατά την παρουσία του στην $Cache_1$. Εφόσον χρειαστεί το τμήμα με τη μικρότερη χρησιμότητα, οδηγείται σε έξωση. Συνεπώς, υπολογίζουμε τη χρησιμότητα των τμημάτων που είναι αποθηκευμένα στην $Cache_1$ ως:

$$\text{Utility}_1 = \frac{\text{Nchunks played} * \text{Frecency}}{\text{Nchunks in segment} * \text{Nrequest}} \quad (1)$$

όπου $\text{Frecency} = \frac{1}{1+(\frac{\text{Tcurrent} - \text{Tlast accessed}}{\text{b}})}$

Ο πίνακας Ι παρουσιάζει τις παραμέτρους της παραπάνω έκφρασης.

Συμβολισμός	**Επεξήγηση**
Nchunks played	συνολικός αριθμός μικρών κομματιών (chunks) που παίζονται από το τμήμα κατά την παραμονή του στην cache
Frecency	παράμετρος που εξαρτάται από το πόσο πρόσφατη είναι η τελευταία αίτηση για το τμήμα σε σχέση με την αντίστοιχη αμέσως προηγούμενη αίτηση
Nrequest	συνολικός αριθμός των αιτήσεων του τμήματος κατά την παραμονή του στην cache
Nchunks in segment	συνολικός αριθμός των μικρών κομματιών που αποτελούν ένα τμήμα
Tcurrent	χρονική στιγμή που φθάνει η τρέχουσα αίτηση
Tlast accessed	χρονική στιγμή της τελευταίας αίτησης για το συγκεκριμένο τμήμα πριν το Tcurrent κατά την παραμονή του στην Cache_1
b	παράμετρος εξομάλυνσης της παραμέτρου Frecency

Πίνακας Ι: Παράμετροι της συνάρτησης Utility_1

Παρατηρείται ότι η χρησιμότητα που βασίζεται στην Εξ.(1) μειώνεται καθώς αυξάνεται ο χρόνος κατά τον οποίο το τμήμα ενός αντικειμένου παραμένει στην $Cache_1$ χωρίς να ζητηθεί. Αυτό εξασφαλίζει ότι τέτοια τμήματα οδηγούνται με προτεραιότητα σε έξωση από την $Cache_1$.

Μετά από μια αίτηση για ένα τμήμα γίνεται έλεγχος αν το τμήμα βρίσκεται σε κάποια από τις δύο μνήμες και στην περίπτωση που δε βρίσκεται σε καμία από αυτές το τμήμα εισέρχεται στη $Cache_1$, χωρίς αρχική τιμή χρησιμότητας. Αν το τμήμα αυτό ζητηθεί ξανά κατά τη διάρκεια παρουσίας του στη $Cache_1$, το τμήμα αποκτά χρησιμότητα αφού έχουν ενημερωθεί οι παράμετροι της εξίσωσης (1). Η παράμετρος b χρησιμοποιείται για την εξομάλυνση του παράγοντα που καθορίζει το πόσο πρόσφατη είναι η τρέχουσα αίτηση για το τμήμα σε σχέση με την αντίστοιχη αμέσως προηγούμενη (Frecency). Αυτό εξασφαλίζει ότι η χρησιμότητα ενός τμήματος δεν μεταβάλλεται δραστικά σε σύντομο χρονικό διάστημα, πράγμα το οποίο αν συνέβαινε θα οδηγούσε με μεγάλη πιθανότητα σε λανθασμένη πρόβλεψη της δημοτικότητας του.

Πρέπει εδώ να σημειωθεί, ότι ένα τμήμα δεν επιτρέπεται να απομακρυνθεί από την $Cache_1$ όταν θεωρείται ενεργό (κατά την διάρκεια δηλαδή που παρακολουθείται από κάποιον χρήστη), με αυτόν τον τρόπο αποφεύγεται μια αστοχία (cache miss). Όταν όλα τα τμήματα στην $Cache_1$ είναι ενεργά και ένα νέο αίτημα για ένα τμήμα αντικειμένου που δεν είναι αποθηκευμένο στις δύο μνήμες φθάσει, τότε το αίτημα απορρίπτεται και δεν ικανοποιείται. Τέλος, η $Cache_1$ δεν αφαιρεί τμήματα που γνωρίζει ότι πρόκειται να παρακολουθηθούν στο σύντομο μέλλον από κάποιον χρήστη, αλλά επιλέγει να αφαιρέσει το τμήμα με την αμέσως μικρότερη χρησιμότητα υπό την προϋπόθεση ότι αυτό δεν είναι ενεργό εκείνη τη στιγμή.

3.1.2 Μέθοδος αντικατάστασης στην $Cache_2$

Όπως αναλύθηκε στην παράγραφο 3.1.1, μετά από κάθε αίτηση που φθάνει για κάποιο τμήμα ενός αντικειμένου που βρίσκεται στην $Cache_1$ η τιμή της χρησιμότητας του ανανεώνεται. Όταν αυτή η τιμή περάσει ένα προκαθορισμένο κατώφλι (threshold), τότε το τμήμα μεταφέρεται από την $Cache_1$ στην $Cache_2$. Δεδομένου ότι το σύστημα έχει αποκτήσει μια καλή εκτίμηση για την δημοτικότητα των τμημάτων αυτών,

χρησιμοποιούμε μια διαφορετική συνάρτηση για τον υπολογισμό της χρησιμότητας τους στην $Cache_2$, η οποία είναι ανάλογη με τον μέσο αριθμό των μικρών κομματιών που παίζονται από το τμήμα κατά την παραμονή του στις δύο μνήμες και με την πιθανότητα να ζητηθεί το συγκεκριμένο τμήμα σε επόμενο αίτημα που θα φθάσει στο σύστημα για το δεδομένο αντικείμενο video και αντιστρόφως ανάλογη με τον συνολικό αριθμό των μικρών κομματιών του τμήματος και τον συνολικό αριθμό των αιτήσεων που έχει δεχθεί το τμήμα κατά την παραμονή του στις δύο μνήμες ($Cache_1$ και $Cache_2$). Αν χρειασθεί, το τμήμα με τη μικρότερη χρησιμότητα οδηγείται σε έξωση. Ο υπολογισμός τη χρησιμότητας του τμήματος i από το video j το οποίο είναι αποθηκευμένο στην $Cache_2$ γίνεται ως εξής:

$$\text{Utility}_2(i,j) = \frac{\text{Nchunks played} * \text{Pnext request}(i,j)}{\text{Nchunks in segment} * \text{Nrequest}} \quad (2)$$

όπου $\text{Pnext request}(i,j) = \frac{\frac{1}{\lambda} * \text{Pvideo}(j) * \text{Psegment}(i)}{\max\{\frac{1}{\lambda} * \text{Pvideo}(j) * \text{Psegment}(i), \text{Tsince last request}\}}$

Ο πίνακας ΙΙ παρουσιάζει τις παραμέτρους της παραπάνω έκφρασης.

Συμβολισμός	**Επεξήγηση**
Pnext request	πρόβλεψη της πιθανότητας να ζητηθεί το τμήμα i του video j σε επόμενο αίτημα χρήστη
Tsince last request	χρονικό διάστημα που έχει περάσει από την τελευταία φορά που ζητήθηκε το τμήμα κατά την παρουσία του στην $Cache_2$
Pvideo	πιθανότητα αίτησης του video j από έναν χρήστη
Psegment	πιθανότητα του τμήματος i να ζητηθεί σε επόμενο αίτημα χρήστη

Πίνακας ΙΙ: Παράμετροι της συνάρτησης $Utility_2$

Η χρησιμότητα που βασίζεται στην Εξ.(2), αντί τις παραμέτρου που εκτιμά το πόσο πρόσφατη είναι μία αίτηση ενός χρήστη για κάποιο τμήμα (Frecency) χρησιμοποιεί την παράμετρο πρόβλεψης (Pnext request). Η πρόβλεψη καθορίζεται με βάση το μέσο διάστημα μεταξύ διαδοχικών αιτήσεων για το κάθε τμήμα ενός αντικειμένου. Όσο η δημοτικότητα ενός τμήματος μειώνεται, ο χρόνος μεταξύ αφίξεων δύο διαδοχικών αιτήσεων για το συγκεκριμένο τμήμα αυξάνεται.

Η χρησιμότητα ενός τμήματος κατά την είσοδο του στη $Cache_2$ διατηρείται και η χρονική στιγμή της τελευταίας αίτησης για το τμήμα συμπίπτει με τη χρονική στιγμή της εισόδου του στην $Cache_2$. Αν κατά τη παρουσία του τμήματος στην $Cache_2$ αυτό ζητηθεί ξανά, τότε η χρησιμότητα του ανανεώνεται με βάση την εξίσωση (2). Σε αυτήν την περίπτωση οι τιμές των παραμέτρων Nrequest, Nchunks played και Nchunks in segment της εξίσωσης (1) μεταφέρονται για τον υπολογισμό της εξίσωσης (2).

Κεφάλαιο 4

4.1 Μετρικές απόδοσης του συστήματος

Η κύρια αποστολή της μεθόδου διαχείρισης της κρυφής μνήμης είναι να εκμεταλλευτεί αποτελεσματικά τους πόρους αποθήκευσης, ώστε να μειωθεί ο όγκος των τμημάτων των videos που προσκομίζονται αναγκαστικά από τον κεντρικό εξυπηρετητή. Το πόσο αποδοτική είναι η χρήση της κρυφής μνήμης σε αυτή την κατεύθυνση φαίνεται από την τιμή της μετρικής Byte hit ratio (BHR). Η BHR αντιστοιχεί στο μέσο κλάσμα των δεδομένων που εξυπηρετούνται άμεσα από τις τοπικές μνήμες-cache. Πιο συγκριμένα, η μετρική BHR ορίζεται ως ο συνολικός αριθμός των bytes που βρίσκονται στις cache, μετά μια αίτηση για το τμήμα ενός video, δια του συνολικού αριθμού των bytes για όλα τα τμήματα των video που έχουν ζητηθεί κατά τη διάρκεια της προσομοίωσης. Πρόκειται για το hit ratio εκφρασμένο σε bytes, δηλαδή δεν υπολογίζουμε τον αριθμό των αιτήσεων που ικανοποιούνται από τις τοπικές μνήμες, αλλά τον αριθμό των bytes που αντιστοιχούν σε αυτές τις αιτήσεις. Με αυτόν τον τρόπο υψηλές τιμές BHR χαρακτηρίζουν την σωστή διαχείριση μνήμης και κατά συνέπεια επιφέρουν χαμηλή κίνηση στο δίκτυο. Το BHR παίρνει τιμές μεταξύ του 0 και 1.

Η μετρική Delayed Starts αντιπροσωπεύει το κλάσμα των αιτήσεων αντικειμένων video από τους χρήστες για τις οποίες οποίες υπάρχει cache miss. Πιο συγκεκριμένα, η μετρική αντιστοιχεί στον αριθμό των αιτήσεων (για τα πρώτα τμήματα των αντικειμένων) που οδήγησαν σε cache miss σε όλη τη διάρκεια της προσομοίωσης προς το συνολικό αριθμό των αιτήσεων στη διάρκεια της προσομοίωσης. Η παραπάνω μετρική αντιστοιχεί στο ποσοστό των αιτήσεων των χρηστών που βιώνουν αρχική καθυστέρηση στην αναπαραγωγή του video, δεδομένου ότι η cache πρέπει να εγκαθιδρύσει σύνδεση με τον απομακρυσμένο εξυπηρετητή προτού αρχίσει να παραδίδει το video στον χρήστη.

4.2 Η κατανομή MZipf

Στην παρούσα εργασία, οι πιθανότητες ζήτησης των αντικειμένων από τους χρήστες υπολογίζονται πριν αρχίσει η προσομοίωση σύμφωνα με την κατανομή Zipf Mandelbrot (MZipf). Όπως προκύπτει από την βιβλιογραφία [6], η δημοτικότητα σε πολυμέσα συνεχούς ροής αποκλίνει από Zipf και Zipf-like κατανομές. Οι εργασίες [1] και [6] επιβεβαιώνουν ότι η μοντελοποίηση της δημοτικότητας ταινιών ή αντικειμένων video με Zipf κατανομή μπορεί να οδηγήσει σε σημαντικό σφάλμα. Σύμφωνα με τις παραπάνω εργασίες, μια Zipf κατανομή θα υπερεκτιμήσει τα αντικείμενα video με χαμηλή δημοτικότητα. Η MZipf κατανομή ορίζεται ως εκείνη για την οποία η πιθανότητα ζήτησης του video i από ένα σύνολο N διαθέσιμων videos, δίνεται από τον τύπο $P_{i=} \frac{K}{(i+p)^s}$, όπου $K= \frac{1}{\sum_{i=1}^{N} \frac{1}{(i+p)^s}}$, s παράμετρος της κατανομής που καθορίζει την πολικότητα (skew) και p παράμετρος που καθορίζει το πλάτωμα (plateau) της κατανομής. Αξίζει να σημειωθεί ότι όσο μεγαλύτερη είναι η παράμετρος p τόσο πιο πεπλατυσμένο γίνεται το αριστερό μέρος (κεφαλή) της κατανομής. Αν η παράμετρος p γίνει μηδέν τότε η Mzipf κατανομή εκφυλίζεται σε Zipf κατανομή. Για τις τιμές των παραμέτρων της MZipf που χρησιμοποιούνται γίνεται εκτενέστερη αναφορά στη παράγραφο 4.3.1.3.

4.3 Προσομοίωση του προτεινόμενου σχήματος προσωρινής αποθήκευσης

Στην ενότητα αυτή, παρουσιάζονται οι λεπτομέρειες και τα αποτελέσματα της προσομοίωσης του συστήματος. Ο χρόνος είναι διαιρεμένος σε ίσα χρονικά παράθυρα, όπου κάθε παράθυρο έχει διάρκεια 600sec. Οι αφίξεις των αιτήσεων των χρηστών στο σύστημα ακολουθούν κατανομή Poisson με ρυθμό λ ίσο με 1 αίτηση ανά 100 δευτερόλεπτα. Ο χρόνος λειτουργίας του συστήματος τίθεται 24 ημέρες και αντιστοιχεί περίπου σε 3500 παράθυρα. Στη διάρκεια αυτή έχουμε κατά μέσο όρο 3500*6= 21.000 αιτήσεις για αντικείμενα video, που αντιστοιχούν κατά μέσο όρο σε 120.000 αιτήματα για τμήματα αντικειμένων video. Ο συνολικός αριθμός των videos είναι 2.000 και υποθέτουμε ότι όλα έχουν το ίδιο μέγεθος. Τα videos έχουν διάρκεια 100 λεπτά, το

οποίο προσεγγιστικά αντιστοιχεί σε μέγεθος ταινίας ίσο με 1.5GB. Όπως αναφέραμε, κάθε ταινία διαμερίζεται ομοιόμορφα, και υποθέτουμε ότι αποτελείται από 10 ίσου μεγέθους τμήματα (segments), καθένα των οποίων έχει διάρκεια 10 λεπτών και κάθε τμήμα διαιρείται σε 10 ίσου μεγέθους μικρά κομμάτια (chunks), καθένα των οποίων έχει διάρκεια 1 λεπτού.

Υποθέτουμε ότι οι αιτήσεις των χρηστών ξεκινούν πάντα από την αρχή της ζητούμενης ταινίας, μπορούν όμως να τελειώνουν όμως σε διαφορετικά σημεία της. Η χρονική στιγμή κατά την οποία ένας χρήστης σταματά να παρακολουθεί την ταινία που ζήτησε υποθέτουμε ότι ακολουθεί μία trimodal κατανομή σύμφωνα με την οποία ο χρήστης σταματά να παρακολουθεί την ταινία ομοιόμορφα στο πρώτο εικοσάλεπτο της διάρκειας της με πιθανότητα 0.4, ομοιόμορφα στο διάστημα από το $21^{ο}$ έως $80^{ο}$ λεπτό της με πιθανότητα 0.2 και ομοιόμορφα στο διάστημα από το $81^{ο}$ έως $100^{ο}$ λεπτό της με πιθανότητα 0.4. Στην πρώτη περίπτωση η αίτηση αφορά το πολύ τα πρώτα δύο τμήματα του video, οπότε ο αριθμός μικρών κομματιών (chunks) που θα παρακολουθήσει ο χρήστης προκύπτει από την ομοιόμορφη κατανομή στο διάστημα 1 έως 20. Έτσι, ανάλογα με το αποτέλεσμα της παραπάνω ομοιόμορφης κατανομής στην $Cache_1$ θα εισέλθει μόνο το πρώτο ή και τα δύο πρώτα τμήματα του ζητούμενου video. Στην δεύτερη περίπτωση, η αίτηση αφορά το πολύ τα τμήματα 1 έως 8 του video, ο χρήστης θα παρακολουθήσει σίγουρα τα πρώτα δύο τμήματα (20 chunks) και ο υπολειπόμενος αριθμός chunks που θα παρακολουθήσει προκύπτει από μία ομοιόμορφη κατανομή στο διάστημα 21 έως 80. Έτσι, στην δεύτερη περίπτωση, στην $Cache_1$ θα εισέλθουν τα πρώτα δύο τμήματα του video και ανάλογα με το αποτέλεσμα της παραπάνω ομοιόμορφης κατανομής θα εισέλθουν και τα τμήματα 3 έως 8. Στην τρίτη περίπτωση, η αίτηση αφορά τα τμήματα 1 έως 10 του video, ο χρήστης θα παρακολουθήσει σίγουρα τα πρώτα οκτώ τμήματα (80 chunks) και ο υπολειπόμενος αριθμός chunks που θα παρακολουθήσει προκύπτει από μία ομοιόμορφη κατανομή στο διάστημα 81 έως 100. Έτσι, στην περίπτωση αυτή, στην $Cache_1$ θα εισέλθουν τα πρώτα οχτώ τμήματα του video και ανάλογα με το αποτέλεσμα της παραπάνω ομοιόμορφης κατανομής θα εισέλθουν και τα τμήματα 9 ή 9 και 10. Από τις τρεις περιπτώσεις που αναφέραμε προκύπτει η παράμετρος Psegment της εξίσωσης (2), η οποία είναι η πιθανότητα του κάθε τμήματος να είναι αυτό που θα ζητηθεί σε επόμενο αίτημα του χρήστη. Ο πίνακας IV δείχνει τις τιμές που παίρνει η παράμετρος αυτή για το κάθε τμήμα ενός αντικειμένου.

Μετά από μια αίτηση που αφορά ένα αριθμό τμημάτων ενός αντικειμένου, κάθε ένα από αυτά εισέρχεται στην $Cache_1$ την χρονική στιγμή που θα το δει ο χρήστης που υποβάλλει την αίτηση. Δηλαδή, το πρώτο τμήμα του αντικειμένου εισέρχεται απευθείας στην $Cache_1$ και τα εναπομείναντα τμήματα που περιλαμβάνονται στο αίτημα εισέρχονται μετά από 10λεπτά* (αριθμό_τμήματος-1) για καθένα από αυτά. Επομένως, τα τμήματα αναμένουν για την προκαθορισμένη είσοδο τους στην $Cache_1$ και κάθε φορά κατά την είσοδο ενός τμήματος στην $Cache_1$ γίνεται έλεγχος της λίστας τμημάτων που αναμένουν για την είσοδο τους στην μνήμη, ώστε να μην εκδιωχθεί τμήμα που το σύστημα γνωρίζει ότι πρόκειται να παρακολουθηθεί από κάποιον χρήστη. Κάθε τμήμα αντικειμένου έχει ελάχιστο χρόνο παραμονής στις μνήμες-cache, ο οποίος είναι ανάλογος με το μήκος του (αριθμό chunks που έχουν ζητηθεί να παιχτούν από το τμήμα), καθώς εισέρχεται στο σύστημα. Πιο συγκεκριμένα, ο ελάχιστος χρόνος παραμονής καθορίζεται από το χρόνο αναπαραγωγής του, δηλαδή είναι 1λεπτό (διάρκεια 1 chunk) * επί τον αριθμό των chunks που έχουν ζητηθεί να παιχτούν από το τμήμα. Ένα τμήμα που δεν έχει ολοκληρώσει τον ελάχιστο χρόνο παραμονής του στην $Cache_1$ δεν μπορεί να εκδιωχθεί από αυτήν. Όπως αναφέραμε και ανωτέρω, αν όλα τα τμήματα στην $Cache_1$ είναι ενεργά, τότε τυχόν αίτημα για ένα νέο τμήμα δεν ικανοποιείται. Στις προσομοιώσεις μας δε συναντήσαμε τέτοια περίπτωση εκτός των περιπτώσεων που μειώσουμε τη χωρητικότητα της cache σε πολύ μικρά μεγέθη, για παράδειγμα σε μία cache χωρητικότητας μόνο 100 τμημάτων video παρατηρούνται αιτήματα χρηστών που δεν ικανοποιούνται.

Στη μελέτη μας αρχικά υποθέτουμε ότι η $Cache_1$ έχει χωρητικότητα 180 videos ή αλλιώς 1800 τμήματα video και η $Cache_2$ έχει χωρητικότητα 20 videos ή αλλιώς 200 τμήματα video. Η αναλογία επομένως προς της συνολική χωρητικότητα της μνήμης είναι $Cache_1$=90% και $Cache_2$=10%. Για τον υπολογισμό της μετρικής BHR αρχίζουμε να συλλέγουμε στοιχεία, αφού το σύστημα έχει δεχτεί αιτήματα χρηστών για 2 ημέρες (warm up period). Ο πίνακας ΙΙΙ παρουσιάζει το σύνολο των προεπιλεγμένων παραμέτρων που χρησιμοποιούνται στην προσομοίωση.

Συμβολισμός	**Επεξήγηση(Τιμή)**
w	διάρκεια παραθύρου (600sec)
n	αριθμός των video που είναι αποθηκευμένα στον server (2000)
m	αριθμός τμημάτων για κάθε video (10)
chunks	αριθμός chunks για κάθε τμήμα (10)
c_1	συνολική χωρητικότητα τμημάτων video στην $Cache_1$ (1800)
c_2	συνολική χωρητικότητα τμημάτων video στην $Cache_2$ (200)
1/λ	ρυθμός άφιξης αιτήσεων χρηστών (100sec)
s	παράμετρος πολικότητας της MZipf (0.8)
p	παράμετρος πλατώματος της MZipf (20)
b	παράμετρος εξομάλυνσης της παραμέτρου Frecency (1000)
threshold	κατώφλι (0.4)
Runs	αριθμός παραθύρων στη διάρκεια της προσομοίωσης (3500), αντιστοιχεί σε διάρκεια λειτουργίας του συστήματος ίση με 24 ημέρες

Πίνακας ΙΙΙ: Προεπιλεγμένες παραμέτροι της προσομοίωσης

Trimodal	Psegment για το τμήμα i
ο χρήστης σταματά να παρακολουθεί το video στο διάστημα από το 1ο έως 20ο λεπτό του (συμβαίνει με πιθανότητα P_1=0.4)	P=1, για το τμήμα1
	P=1 - 1/2* P_1 = 0.8, για το τμήμα 2
	P=1 - P_1= 0.6, για το τμήμα 3
ο χρήστης σταματά να παρακολουθεί το video στο διάστημα από το 21ο έως 80ο λεπτό του (συμβαίνει με πιθανότητα P_2=0.2)	P=1 - P_1 - 1/6* P_2 = 0.56, για το τμήμα 4
	P=1 - P_1 - 2/6* P_2 = 0.53, για το τμήμα 5
	P=1 - P_1 - 3/6* P_2 = 0.5, για το τμήμα 6
	P=1 - P_1 - 4/6* P_2 = 0.46, για το τμήμα 7
ο χρήστης σταματά να παρακολουθεί το video στο διάστημα από το 81ο έως 100ο λεπτό του (συμβαίνει με πιθανότητα P_3=0.4	P=1 - P_1 - 5/6* P_2 = 0.43, για το τμήμα 8
	P=1 - P_1 - P_2 = 0.4, για το τμήμα 9
	P=1 - P_1 - P_2-1/2* P_3 =0.2, για το τμήμα 10

Πίνακας IV: Παράμετρος Psegment της συνάρτησης $Utility_2$

4.3.1 Προσομοίωση σε στατικό server

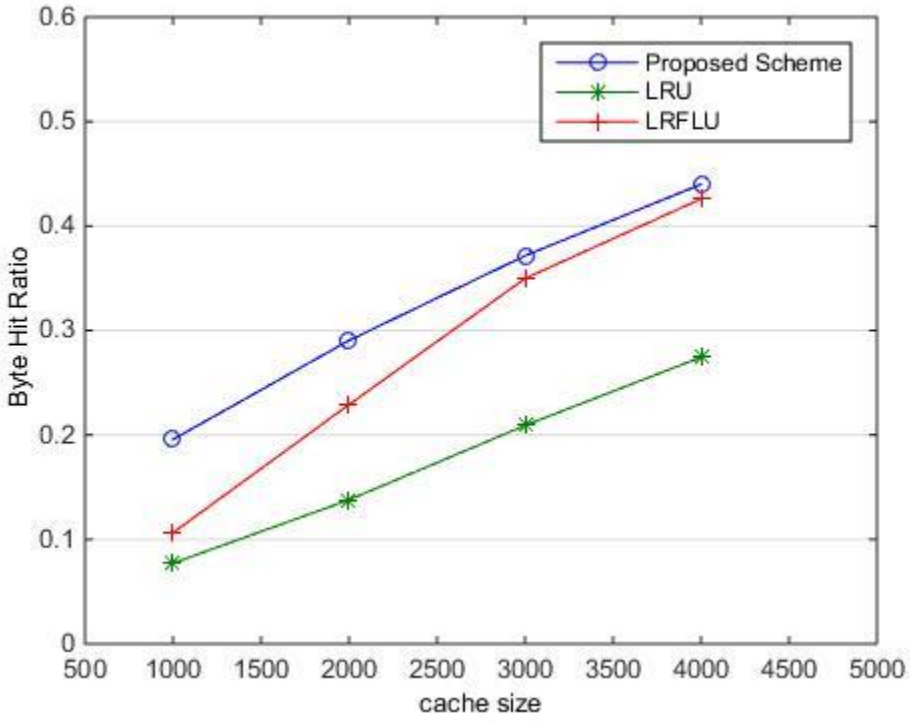
Στην ενότητα αυτή προσομοιώνουμε την περίπτωση στατικού εξυπηρετητή (server), πράγμα το οποίο σημαίνει ότι στον εξυπηρετητή είναι αποθηκευμένος ένας αριθμός αντικειμένων video με σταθερές δημοτικότητες που δεν μεταβάλλονται με το χρόνο. Οι πιθανότητες εμφάνισης των videos υπολογίζονται πριν αρχίσει η προσομοίωση σύμφωνα με κατανομή Mzipf και παραμένουν αμετάβλητες μέχρι το τέλος της προσομοίωσης.

4.3.1.1 Η επίδραση διαφορετικών μεγεθών μνήμης στην cache στις μετρικές απόδοσης

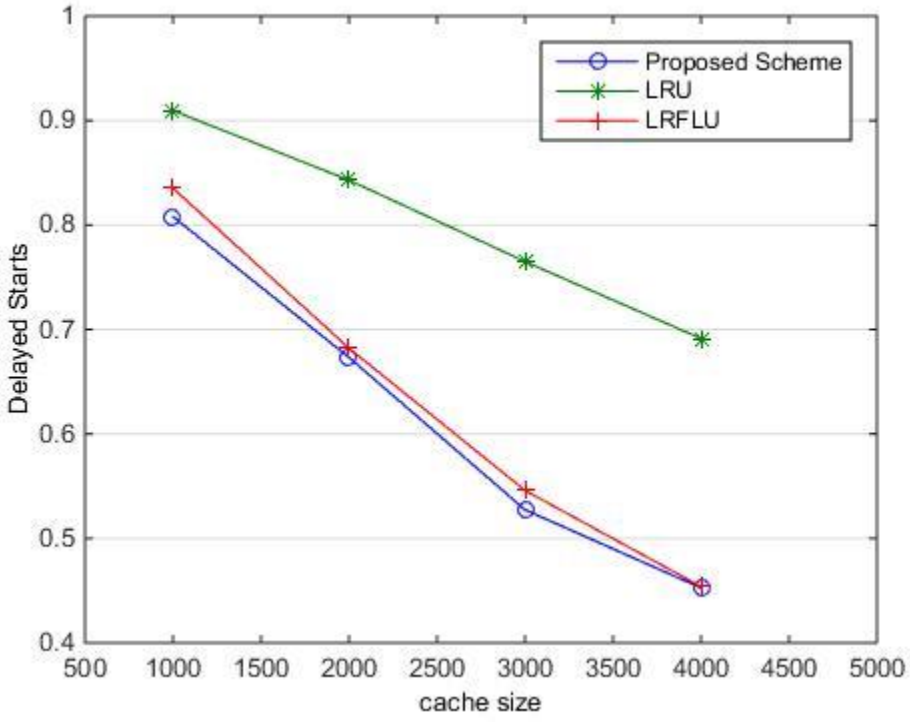
Στην ενότητα αυτή προσομοιώνουμε την περίπτωση στατικού server, για διαφορετικά μεγέθη της συνολικής μνήμης cache. Διατηρούμε σταθερό το μέγεθος της βάσης δεδομένων του εξυπηρετητή, καθώς μεταβάλλουμε το μέγεθος της μνήμης cache από 1000 έως 4000 τμήματα videos, πράγμα το οποίο σημαίνει ότι το ποσοστό των τμημάτων που μπορούν προσωρινά να αποθηκευτούν στη μνήμη cache κυμαίνεται από 5% έως 20% του συνολικού αριθμού των αντικειμένων video του εξυπηρετητή. Η αναλογία ως προς της συνολική χωρητικότητα της μνήμης είναι για την $Cache_1$=90% και για την $Cache_2$=10%. Στην προσομοίωση θέτουμε s=0.8 και p=20 ως τιμές των παραμέτρων της Mzipf κατανομής.

Στο *[Σχήμα 3]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης BHR σαν συνάρτηση του μεγέθους της μνήμης για τις μεθόδους αντικατάστασης LRU, LRFLU και το προτεινόμενο σχήμα σε αυτήν την διπλωματική εργασία. Στο *[Σχήμα 4]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος Delayed Starts σαν συνάρτηση του μεγέθους της μνήμης για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα σε αυτήν την εργασία.

Παρατηρούμε ότι το προτεινόμενο σχήμα εμφανίζει υψηλότερο BHR και χαμηλότερο Delayed Starts συγκριτικά με τις μεθόδους LRFLU και LRU. Όσο αυξάνεται το μέγεθος της μνήμης cache μειώνεται η διαφορά BHR του προτεινόμενου σχήματος σε σχέση με αυτό της μεθόδου LRFLU. Στην περίπτωση που η μνήμη cache έχει χωρητικότητα 4000 τμήματα video (20% του μεγέθους της βάσης δεδομένων του εξυπηρετητή) το BHR των δύο μεθόδων έχει μικρή διαφορά. Τα Delayed Starts του προτεινόμενου σχήματος με αυτά της μέθοδο LRFLU έχουν μικρή διαφορά για όλες τις εξεταζόμενες τιμές του μεγέθους της μνήμης cache. Η μέθοδος LRU εμφανίζει χειρότερα αποτελέσματα όσο αφορά τις μετρικές BHR και Delayed Starts συγκριτικά με τις άλλες εξεταζόμενες μεθόδους.



Σχήμα 3 BHR versus cache size (static server)

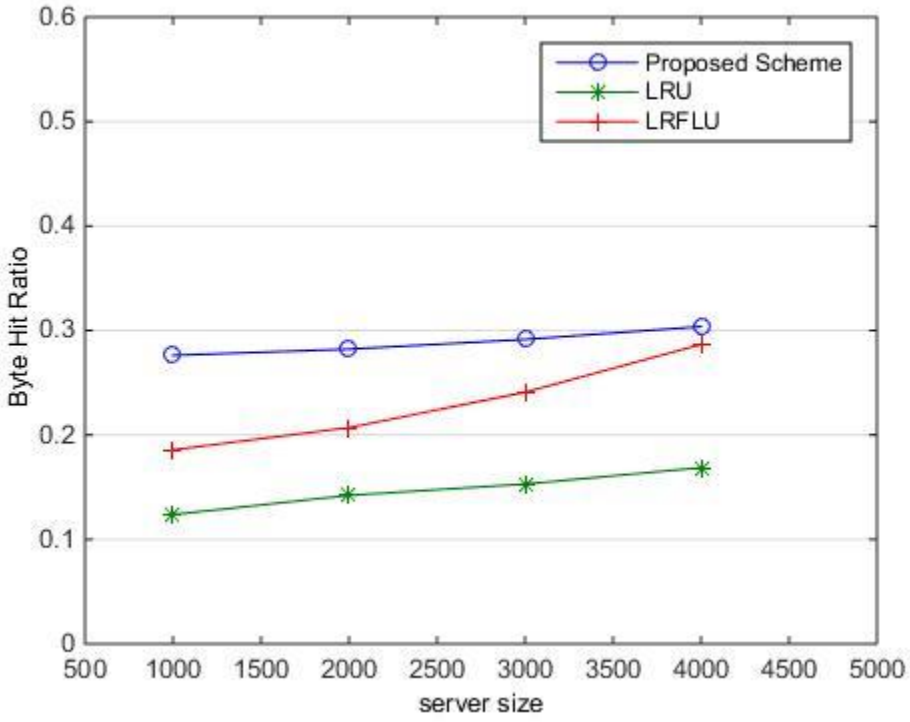


Σχήμα 4 Delayed Starts versus cache size (static server)

4.3.1.2 Η επίδραση διαφορετικού αριθμού αντικειμένων στον server στις μετρικές απόδοσης

Στην ενότητα αυτή προσομοιώνουμε την περίπτωση στατικού server, για διαφορετικά μεγέθη της βάσης δεδομένων του εξυπηρετητή. Διατηρούμε σταθερή τη χωρητικότητα της μνήμης cache στο 10% του εκάστοτε μεγέθους της βάσης δεδομένων του εξυπηρετητή και μεταβάλλουμε τον αριθμό των αντικειμένων στη βάση δεδομένων του εξυπηρετητή από 1000 έως 4000 αντικείμενα video. Από τη μία μεριά αυξάνοντας το μέγεθος του server, οι αιτήσεις των χρηστών αφορούν περισσότερα videos, πράγμα το οποίο σημαίνει ότι το σύστημα της cache πρέπει να μπορεί να κρατήσει προσωρινά αποθηκευμένα τα πιο δημοφιλή τμήματα video ανάμεσα από περισσότερα videos. Η αναλογία ως προς της συνολική χωρητικότητα της μνήμης είναι για την $Cache_1$=90% και για την $Cache_2$=10%. Στην προσομοίωση θέτουμε s=0.8 και p=20 ως τιμές των παραμέτρων της Mzipf κατανομής. Από την άλλη με την αύξηση του μεγέθους της βάσης του server, αυξάνεται ανάλογα το μέγεθος της συνολικής μνήμης cache, που σημαίνει ότι περισσότερα τμήματα video μπορούν να αποθηκεύονται προσωρινά στην cache. Στο *[Σχήμα 5]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση του μεγέθους του server για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα σε αυτήν την εργασία.

Το προτεινόμενο σχήμα εμφανίζει υψηλότερο BHR σε σχέση με τις μεθόδους LRFLU και LRU. Όσο αυξάνουμε το μέγεθος της βάσης δεδομένων του εξυπηρετητή, μειώνεται η διαφορά BHR του προτεινόμενου σχήματος σε σχέση με αυτό της μεθόδου LRFLU. Στην περίπτωση που ο εξυπηρετητής έχει χωρητικότητα 4000 videos το BHR των δύο αυτών μεθόδων έχει μικρή διαφορά. Η μέθοδος LRU εμφανίζει την χειρότερη επίδοση σε BHR σε σχέση με την μέθοδο LRFLU και αυτήν του προτεινόμενου σχήματος σε αυτήν την εργασία.



Σχήμα 5 BHR versus server size (static server)

4.3.1.3 Η επίδραση διαφορετικών τιμών των παραμέτρων της Mzipf στις μετρικές απόδοσης

Στην ενότητα αυτή προσομοιώνουμε την περίπτωση στατικού server, για διαφορετικές τιμές της παραμέτρου s της κατανομής Mzipf. Διατηρούμε σταθερή τη παράμετρο p που καθορίζει το πλάτωμα της κατανομής δημοτικότητας των αντικειμένων video, καθώς μεταβάλλουμε τη παράμετρο s (η οποία καθορίζει την πολικότητα) από 0.6 έως 0.9. Μικρές τιμές της παραμέτρου s οδηγούν σε ένα ευρύ φάσμα δημοφιλών αντικειμένων, ενώ οι υψηλότερες τιμές οδηγούν σε ένα μικρότερο σύνολο δημοφιλών αντικειμένων. Έτσι, όταν η παράμετρος s αυξάνεται είναι ευκολότερο να γίνει διάκριση μεταξύ των πιο δημοφιλών αντικειμένων επειδή οι πιθανότητες εμφάνισης τους είναι υψηλότερες από τις πιθανότητες των υπόλοιπων αντικειμένων, ενώ όταν η παράμετρος s μειώνεται οι πιθανότητες εμφάνισης των διάφορων αντικειμένων είναι κοντά η μία με την άλλη. Αυτό το σενάριο είναι

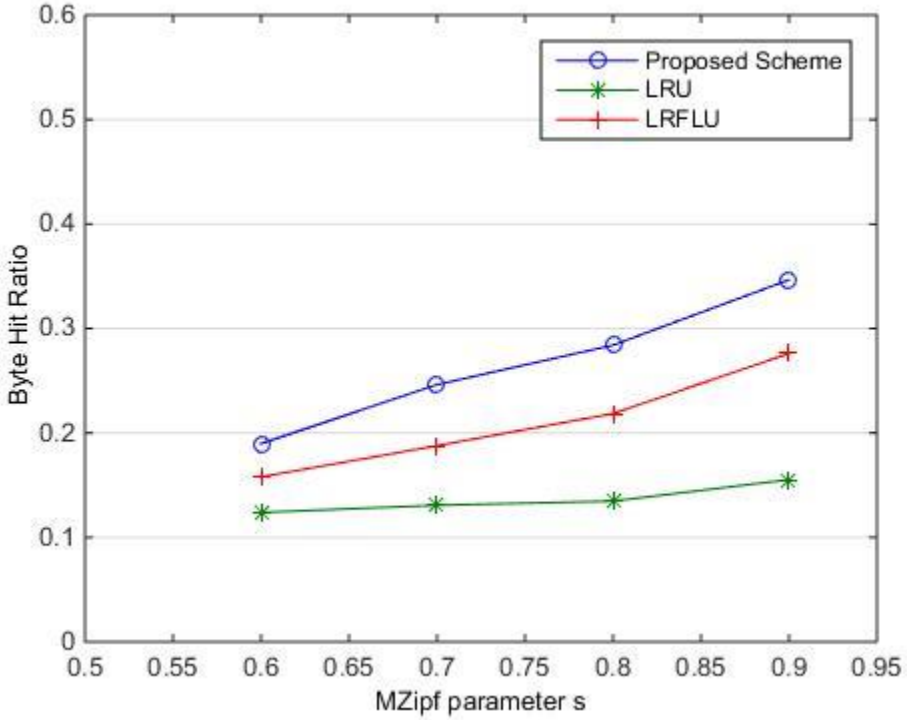
σημαντικό, διότι μπορεί να μας δώσει μια ένδειξη για τη συμπεριφορά των μεθόδων διαχείριση της μνήμης, σε ένα σύστημα στο οποίο οι χρήστες έχουν σχεδόν τις ίδιες προτιμήσεις για τα διάφορα αντικείμενα και σε ένα διαφορετικό σύστημα στο οποίο οι χρήστες έχουν αρκετά διαφορετικές προτιμήσεις για τα διάφορα αντικείμενα.

Στη συνέχεια, προσομοιώνουμε την περίπτωση στατικού server, για διαφορετικές τιμές της παραμέτρου p της κατανομής Mzipf. Διατηρούμε σταθερή τη παράμετρο s που καθορίζει την πολικότητα της κατανομής δημοτικότητας των videos και μεταβάλλουμε τη παράμετρο p (η οποία καθορίζει το πλάτωμα της κατανομής) από 10 έως 40. Η σημασία της παραμέτρου p είναι ότι ελέγχει το αριστερό μέρος της κατανομής, όσο μεγαλύτερη είναι η τιμή του p τόσο πιο πεπλατυσμένη είναι η καμπύλη στα αντικείμενα video μεγάλης δημοτικότητας. Αυτό σημαίνει ότι σε μια τέτοια περίπτωση οι αιτήσεις των χρηστών για video χαμηλής δημοτικότητας είναι περισσότερες συγκριτικά με τις αιτήσεις των χρηστών για video μεγάλης δημοτικότητας, με αποτέλεσμα να έχουμε μικρότερο όφελος από την προσωρινή αποθήκευση.

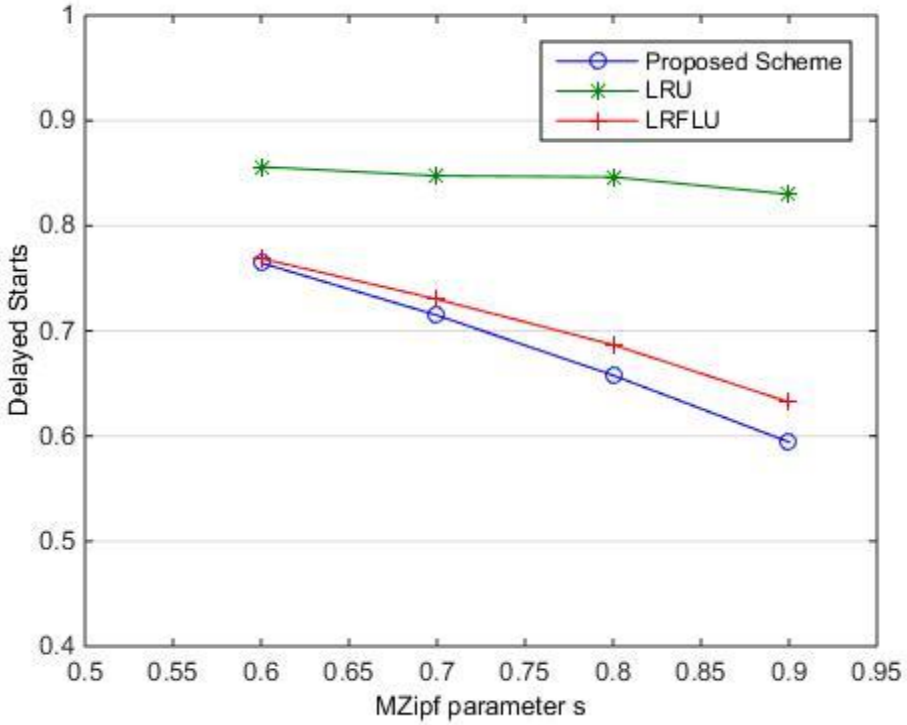
Στο *[Σχήμα 6]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους s για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα σε αυτήν την εργασία. Στο *[Σχήμα 7]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος Delayed Start σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους s για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα. Στο *[Σχήμα 8]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους p για τις μεθόδους αντικατάστασης LRU, LRFLU και το προτεινόμενο σχήμα. Στο *[Σχήμα 9]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος Delayed Starts σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους p για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα.

Παρατηρούμε ότι το προτεινόμενο σχήμα εμφανίζει υψηλότερο BHR και χαμηλότερο Delayed Starts σε σχέση με τις μεθόδους LRFLU και LRU. Το προτεινόμενο σχήμα φαίνεται να λειτουργεί πιο αποδοτικά, όταν οι αιτήσεις αφορούν ένα μικρό υποσύνολο δημοφιλών αντικειμένων video. Πιο συγκεκριμένα, παρατηρούμε ότι η διαφορά του BHR και των Delayed Starts του προτεινόμενου σχήματος μεγαλώνει σε σχέση με τις τιμές που επιτυγχάνουν οι μέθοδοι LRFLU και

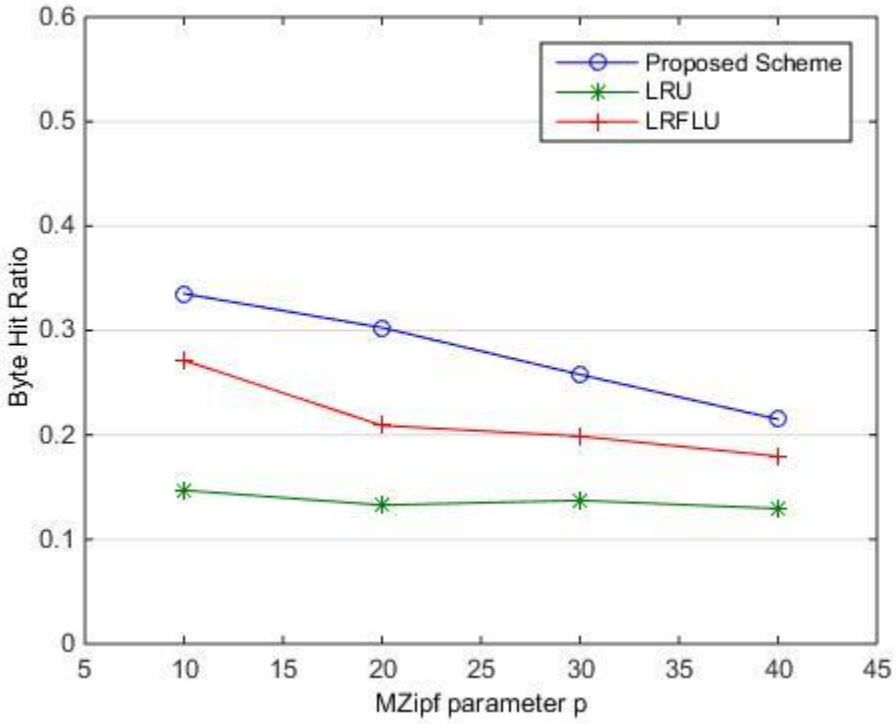
LRU, τόσο στην περίπτωση που αυξάνεται η παράμετρος s, όσο και στην περίπτωση που μειώνεται η παράμετρος p της κατανομής MZipf.



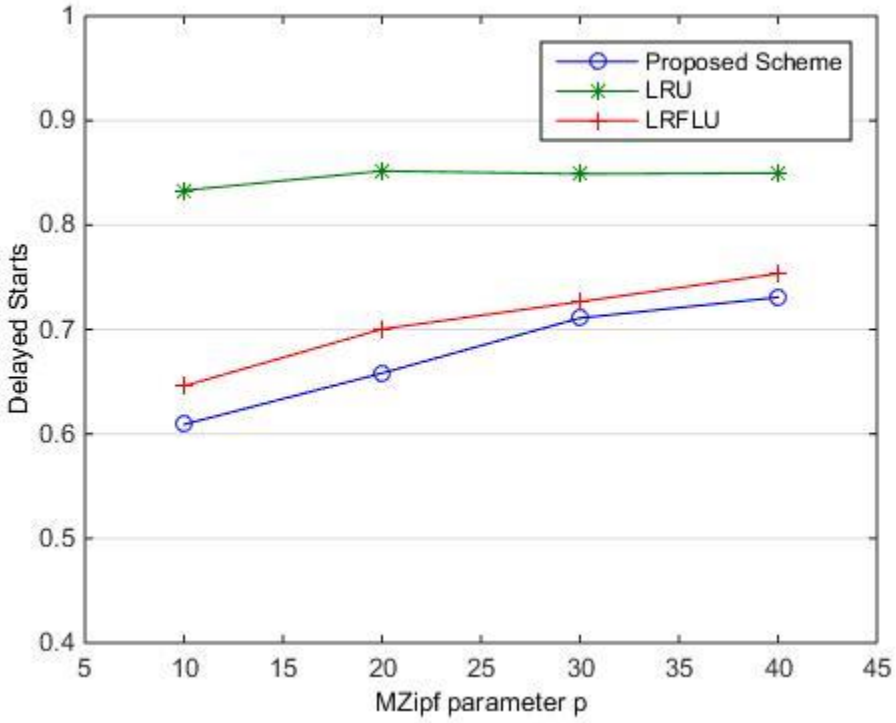
Σχήμα 6 BHR versus MZipf parameter s (static server)



Σχήμα 7 Delayed Starts versus MZipf parameter s (static server)



Σχήμα 8 BHR versus MZipf parameter p (static server)

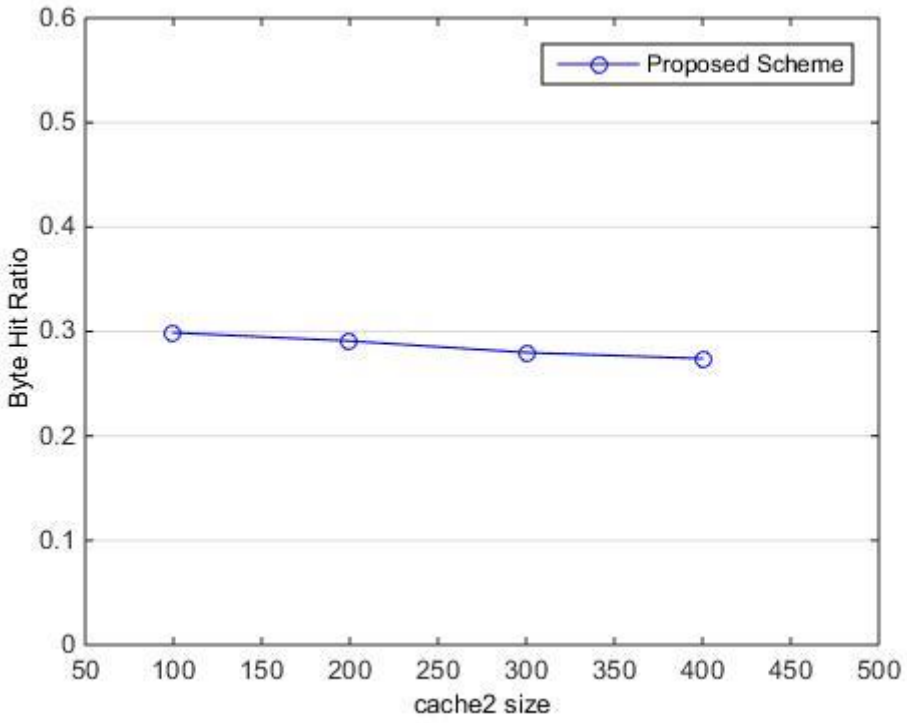


Σχήμα 9 Delayed Starts versus MZipf parameter p (static server)

4.3.1.4 Η επίδραση διαφορετικής αναλογίας μεγεθών της $Cache_1$ και $Cache_2$ στις μετρικές απόδοσης

Στην ενότητα αυτή προσομοιώνουμε την περίπτωση στατικού server, για διαφορετική αναλογία του μεγέθους της $Cache_1$ σε σχέση με το μέγεθος τη $Cache_2$. Διατηρούμε σταθερό το συνολικό μέγεθος της μνήμης cache (10% του μεγέθους της βάσης δεδομένων του εξυπηρετητή), καθώς μεταβάλλουμε το μέγεθος της $Cache_2$ από 100 έως 400 τμήματα videos, πράγμα το οποίο σημαίνει ότι μεταβάλλαμε την αναλογία της $Cache_2$ από 5% έως 20% του συνολικού μεγέθους της μνήμης cache. Παράλληλα, μεταβάλλουμε το κατώφλι (threshold) από 0.4 έως 0.25 (μειώνουμε το κατώφλι κατά 0.5 καθώς αυξάνουμε τα τμήματα της $Cache_2$ κατά 100), έτσι ώστε τα τμήματα των videos να μεταφέρονται πιο εύκολα από την $Cache_1$ στην $Cache_2$, καθώς αυξάνεται το μέγεθος της $Cache_2$, ώστε να γίνεται καλύτερη χρησιμοποίηση της.

Στο *[Σχήμα 10]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση του μεγέθους της $Cache_2$ προς τη συνολική μνήμη cache για την περίπτωση s=0.8 και p=20. Παρατηρούμε ότι η διαφορετικής αναλογία του μεγέθους της $Cache_1$ σε σχέση με το μέγεθος τη $Cache_2$ για σταθερό συνολικό μέγεθος cache, δεν επηρεάζει αισθητά το BHR.



Σχήμα 10 BHR versus cache2 size (static server)

4.3.2 Προσομοίωση σε δυναμικό server

Στην ενότητα αυτή προσομοιώνουμε την περίπτωση δυναμικού εξυπηρετητή (server), πράγμα το οποίο σημαίνει ότι στην διάρκεια προσομοίωσης έχουμε εισαγωγή νέων αντικειμένων video στον server με παράλληλη απομάκρυνση παλαιότερων αντικειμένων video και επιπλέον η δημοτικότητα των αντικειμένων video αλλάζει με την πάροδο του χρόνου.

Σε μια πρώτη προσέγγιση, εισάγουμε X νέα αντικείμενα στον server. Προκειμένου να διατηρήσουμε την ίδια κατανομή δημοτικότητας των videos και το ίδιο μέγεθος της βάσης δεδομένων του server, αφαιρούνται τα X αντικείμενα που βρίσκονται στο κάτω μέρος της λίστας δημοτικότητας και τη θέση τους παίρνουν X νέα αντικείμενα στην κορυφή της λίστας. Η εισαγωγή των νέων αντικειμένων γίνεται στο τέλος κάθε ημέρας προσομοίωσης. Οι δύο caches στο τέλος κάθε ημέρας προσομοίωσης ενημερώνονται και απομακρύνουν τα αποθηκευμένα τμήματα των videos που έχουν αφαιρεθεί από το σύνολο των videos του server. Επίσης, ενημερώνεται η λίστα αναμονής των τμημάτων videos (τμήματα που περιμένουν για την είσοδο τους στην $Cache_1$), γιατί αυτή ενδέχεται να περιέχει τμήματα των videos που έχουν αφαιρεθεί από τον server. Ο χρόνος παραμονής κάθε video στον server είναι N/X ημέρες, όπου N είναι η χωρητικότητα του server. Στην ενότητα αυτή ο ρυθμός εισαγωγής νέων videos είναι X=10 videos ανά ημέρα, το οποίο αντιστοιχεί στο 0.5% του συνολικού αριθμού αντικειμένων video του server.

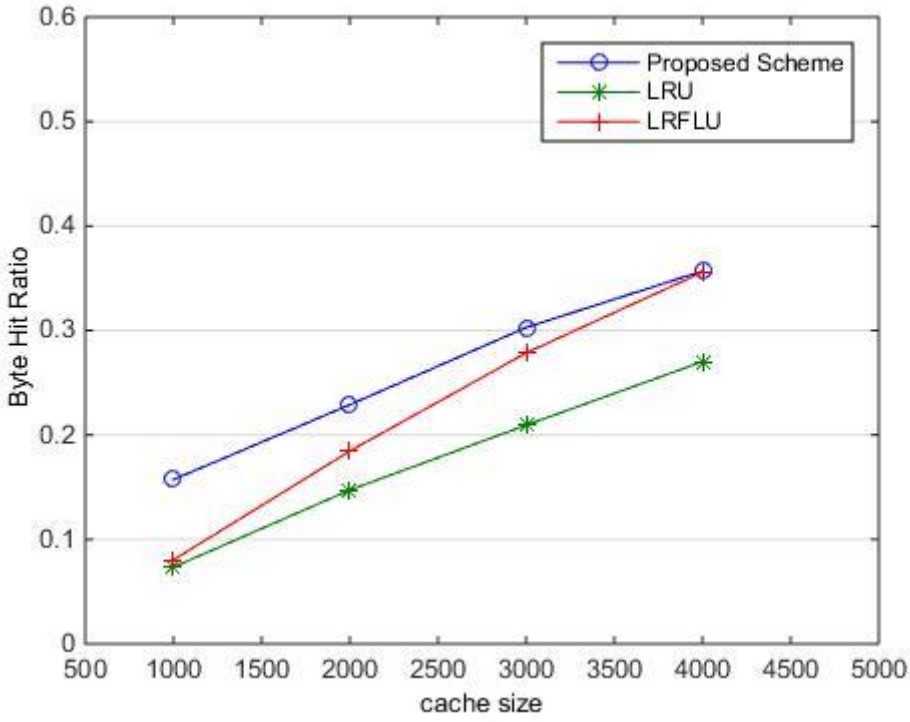
4.3.2.1 Η επίδραση διαφορετικών μεγεθών μνήμης στην cache στις μετρικές απόδοσης

Στην ενότητα αυτή προσομοιώνουμε την περίπτωση δυναμικού server, για διαφορετικά μεγέθη της συνολικής μνήμης cache. Διατηρούμε σταθερό το μέγεθος της βάσης δεδομένων του εξυπηρετητή, καθώς μεταβάλλουμε το μέγεθος της μνήμης cache από 1000 έως 4000 τμήματα videos, πράγμα το οποίο σημαίνει ότι το ποσοστό των τμημάτων που μπορούν προσωρινά να αποθηκευτούν στη μνήμη cache κυμαίνεται από 5% έως 20% του συνολικού αριθμού των αντικειμένων video του εξυπηρετητή. Η αναλογία ως προς της συνολική χωρητικότητα της μνήμης είναι για την $Cache_1$=90%

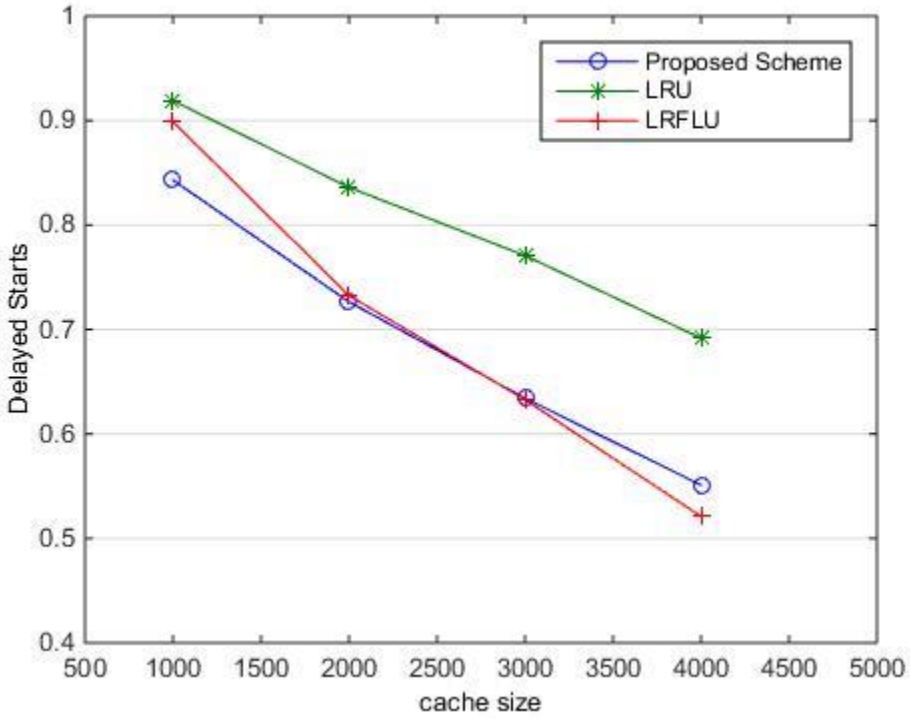
και για την $Cache_2$=10%. Στην προσομοίωση θέτουμε s=0.8 και p=20 ως τιμές των παραμέτρων της Mzipf κατανομής.

Στο *[Σχήμα 11]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση του μεγέθους της μνήμης cache για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σε αυτήν την εργασία σχήμα. Στο *[Σχήμα 12]* παρουσιάζονται τα αντίστοιχα αποτελέσματα της μετρικής απόδοσης του συστήματος Delayed Starts.

Όπως είναι αναμενόμενο, όταν η μνήμη cache μπορεί να αποθηκεύσει περισσότερα τμήματα των videos έχουμε υψηλότερο BHR και χαμηλότερο Delayed Starts. Παρατηρούμε ότι προτεινόμενο σχήμα εμφανίζει υψηλότερο BHR και χαμηλότερο Delayed Starts συγκριτικά με τις μεθόδους LRFLU και LRU. Όσο αυξάνεται το μέγεθος της μνήμης cache μειώνεται η διαφορά BHR του προτεινόμενου σχήματος σε σχέση με αυτό της μεθόδου LRFLU. Στην περίπτωση που η μνήμη cache έχει χωρητικότητα 4000 τμήματα video (20% του μεγέθους της βάσης δεδομένων του εξυπηρετητή), το επιτυγχανόμενο BHR των δύο μεθόδων είναι σχεδόν το ίδιο. Το Delayed Starts του προτεινόμενου σχήματος με αυτό της μεθόδου LRFLU έχει μικρή διαφορά για όλες τις τιμές του μεγέθους της μνήμης cache. Τέλος, η μέθοδος LRU εμφανίζει τα χειρότερα αποτελέσματα, συγκριτικά με τις δύο άλλες μεθόδους.



Σχήμα 11 BHR versus cache size (dynamic server)



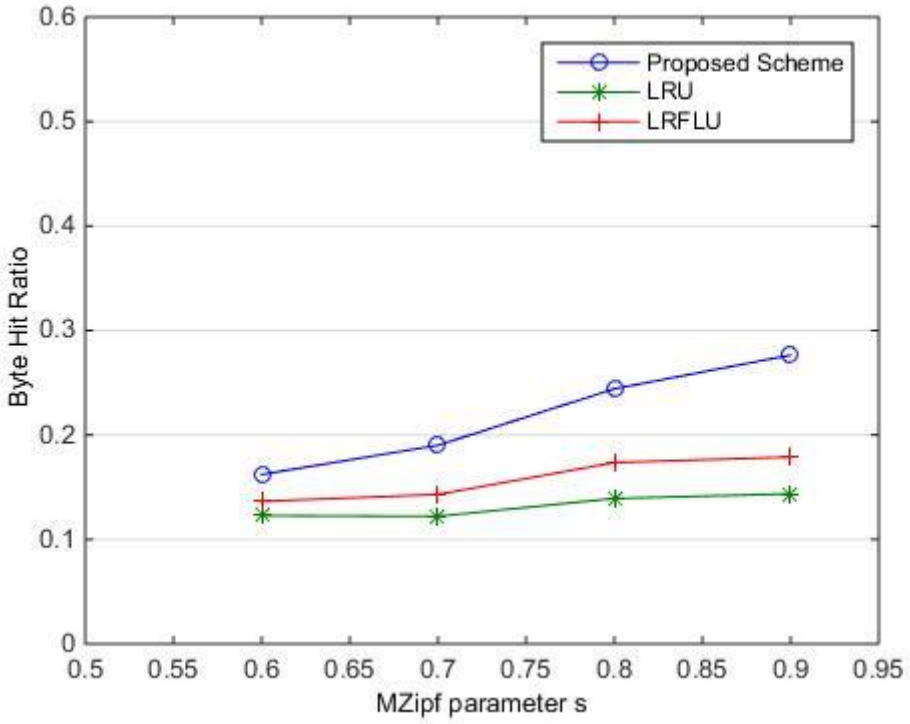
Σχήμα 12 Delayed Starts versus cache size (dynamic server)

4.3.2.2 Η επίδραση διαφορετικών τιμών των παραμέτρων της Mzipf στις μετρικές απόδοσης

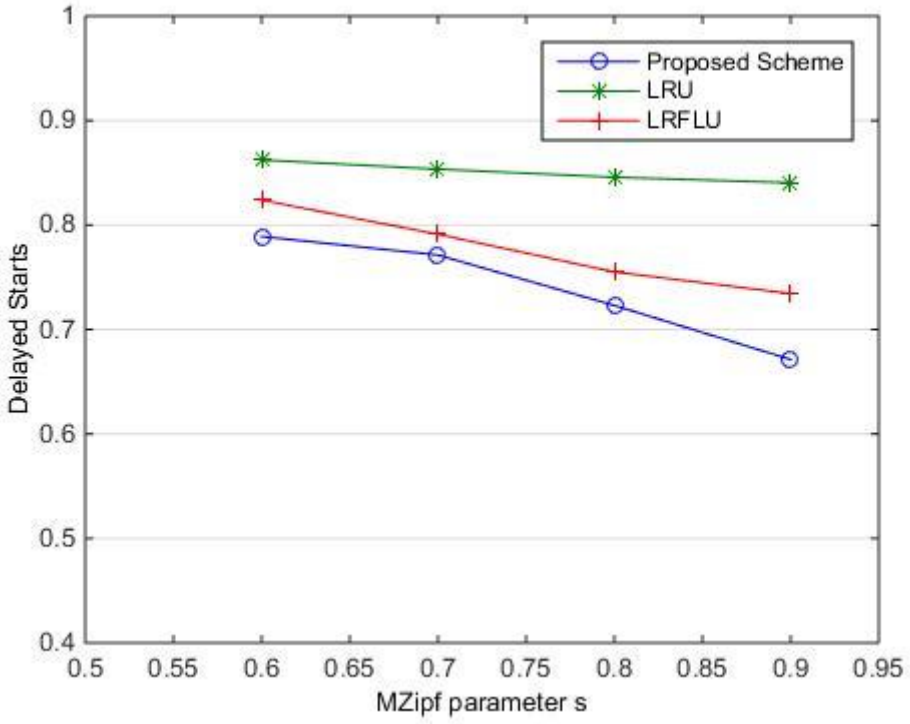
Στην ενότητα αυτή προσομοιώνουμε την περίπτωση δυναμικού server, για διαφορετικές τιμές της παραμέτρου s της κατανομής Mzipf. Διατηρούμε σταθερή την παράμετρο p που καθορίζει το πλάτωμα της κατανομής δημοτικότητας των videos, καθώς μεταβάλλουμε την παράμετρο s (η οποία καθορίζει την πολικότητα) από 0.6 έως 0.9. Στη συνέχεια, προσομοιώνουμε την περίπτωση δυναμικού server, για διαφορετικές τιμές της παραμέτρου p της κατανομής Mzipf. Διατηρούμε σταθερή την παράμετρο s που καθορίζει την πολικότητα της κατανομής δημοτικότητας των videos και μεταβάλλουμε την παράμετρο p (η οποία καθορίζει το πλάτωμα της κατανομής) από 10 έως 40.

Στο *[Σχήμα 13]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους s για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα αυτής της εργασίας. Στο *[Σχήμα 14]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος Delayed Starts σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους s για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα. Στο *[Σχήμα 15]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους p για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα. Στο *[Σχήμα 16]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος Delayed Starts σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους p για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα.

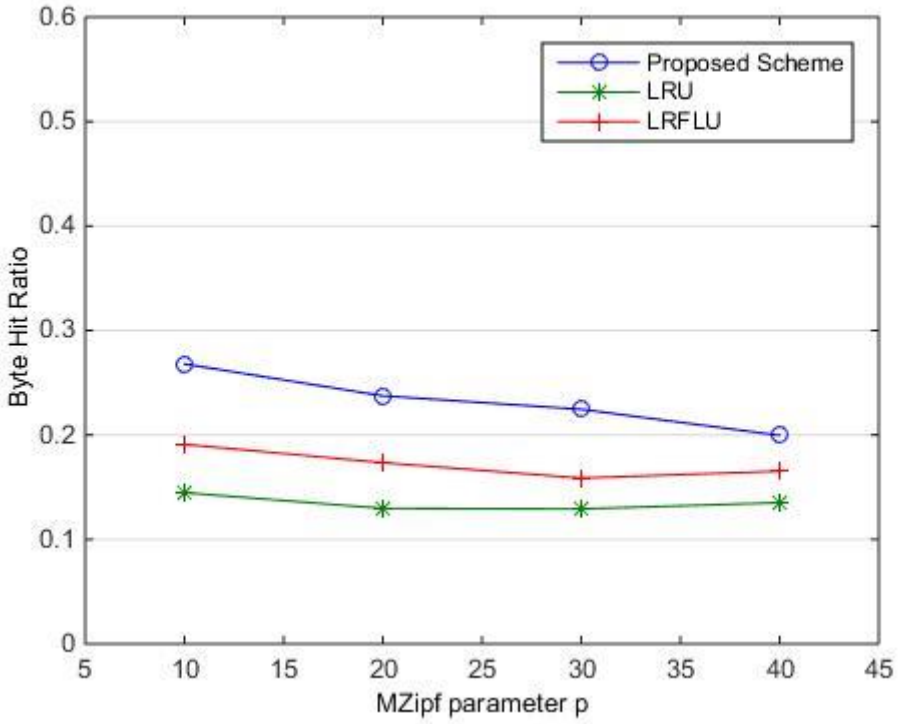
Το προτεινόμενο σχήμα εμφανίζει υψηλότερο BHR και χαμηλότερο Delayed Starts σε σχέση με τις μεθόδους LRFLU και LRU. Το προτεινόμενο σχήμα φαίνεται να λειτουργεί πιο αποδοτικά (όπως και στην περίπτωση της προσομοίωσης με στατικό server), όταν οι αιτήσεις αφορούν ένα μικρό υποσύνολο δημοφιλών αντικειμένων video. Πιο συγκεκριμένα, παρατηρούμε ότι η διαφορά του BHR και Delayed Starts του προτεινόμενου σχήματος μεγαλώνει σε σχέση με αυτά των μεθόδων LRFLU και LRU, τόσο στην περίπτωση που αυξάνεται η παράμετρος s, όσο και στην περίπτωση που μειώνεται η παράμετρος p της κατανομής MZipf.



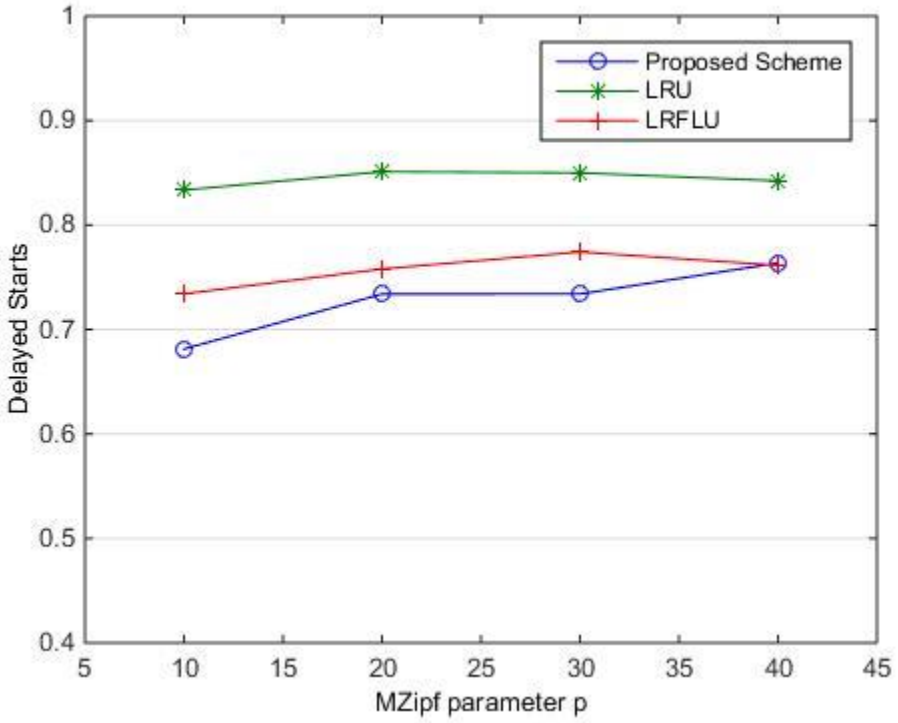
Σχήμα 13 BHR versus MZipf parameter s (dynamic server)



Σχήμα 14 Delayed Starts versus MZipf parameter s (dynamic server)



Σχήμα 15 BHR versus MZipf parameter p (dynamic server)

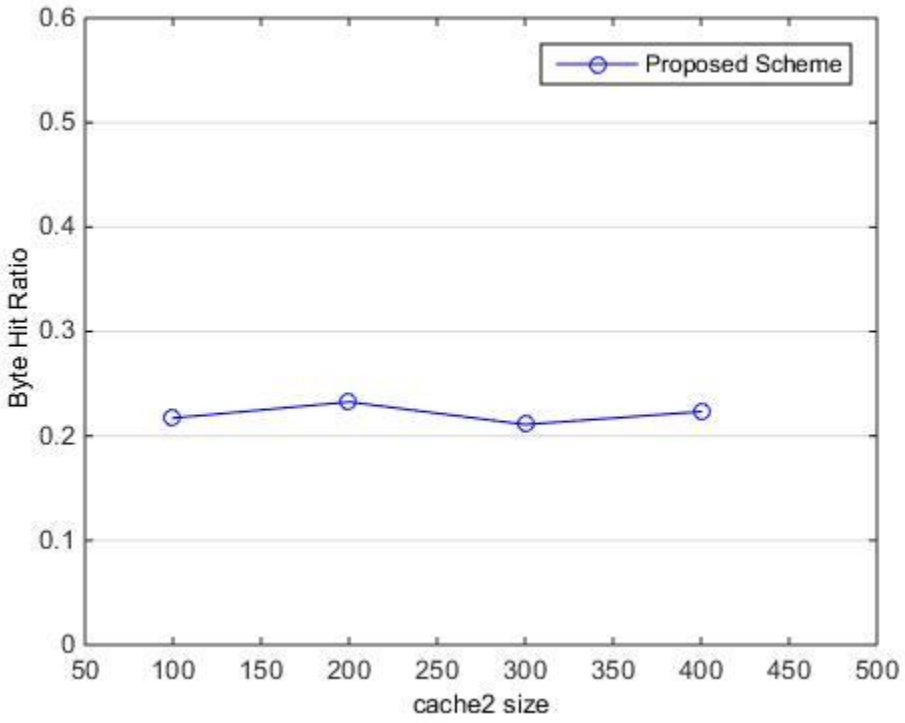


Σχήμα 16 Delayed Starts versus MZipf parameter p (dynamic server)

4.3.2.3 Η επίδραση διαφορετικής αναλογίας μεγεθών της $Cache_1$ και $Cache_2$ στις μετρικές απόδοσης

Στην ενότητα αυτή προσομοιώνουμε την περίπτωση δυναμικού server, για διαφορετική αναλογία του μεγέθους της $Cache_1$ σε σχέση με το μέγεθος τη $Cache_2$. Διατηρούμε σταθερό το συνολικό μέγεθος της μνήμης cache (10% του μεγέθους της βάσης δεδομένων του εξυπηρετητή), καθώς μεταβάλλουμε το μέγεθος της $Cache_2$ από 100 έως 400 τμήματα videos, πράγμα το οποίο σημαίνει ότι μεταβάλλαμε την αναλογία της $Cache_2$ από 5% έως 20% του συνολικού μεγέθους της μνήμης cache. Παράλληλα, μεταβάλλουμε το κατώφλι (threshold) από 0.4 έως 0.25 (μειώνουμε το κατώφλι κατά 0.5 καθώς αυξάνουμε τα τμήματα της $Cache_2$ κατά 100), έτσι ώστε τα τμήματα των videos να μεταφέρονται πιο εύκολα από την $Cache_1$ στην $Cache_2$, καθώς αυξάνεται το μέγεθος της $Cache_2$, ώστε να γίνεται καλύτερη χρησιμοποίηση της.

Στο *[Σχήμα 17]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση του μεγέθους της $Cache_2$ προς τη συνολική μνήμη cache για την περίπτωση s=0.8 και p=20. Παρατηρούμε ότι η διαφορετικής αναλογία του μεγέθους της $Cache_1$ σε σχέση με το μέγεθος τη $Cache_2$ για σταθερό συνολικό μέγεθος cache, δεν επηρεάζει αισθητά το BHR.



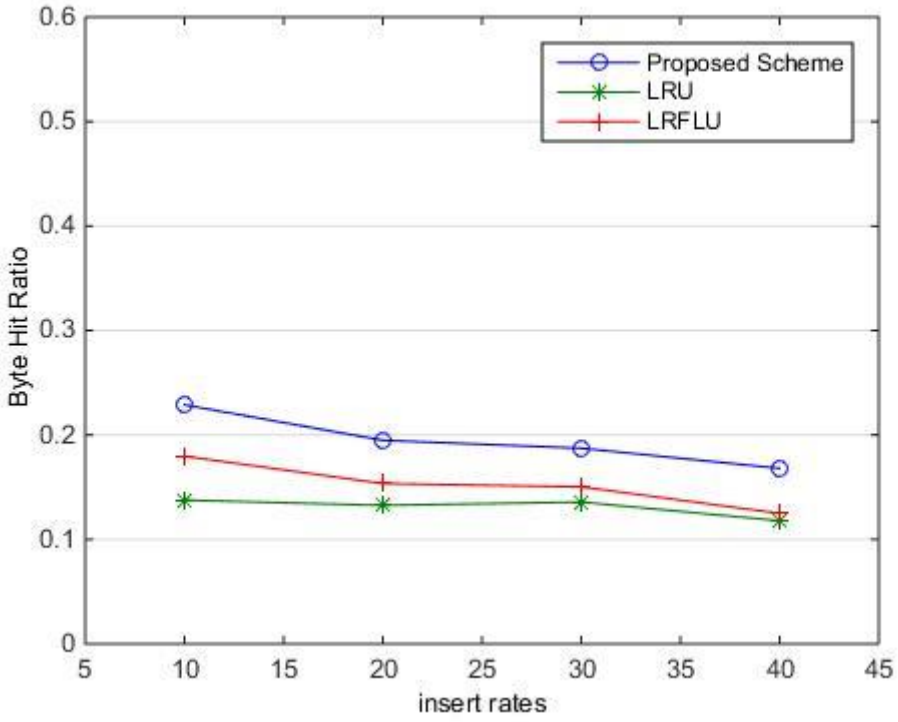
Σχήμα 17 BHR versus $cache_2$ (dynamic server)

4.3.2.4 Η επίδραση διαφορετικού ρυθμού εισαγωγής αντικειμένων video στον server στις μετρικές απόδοσης

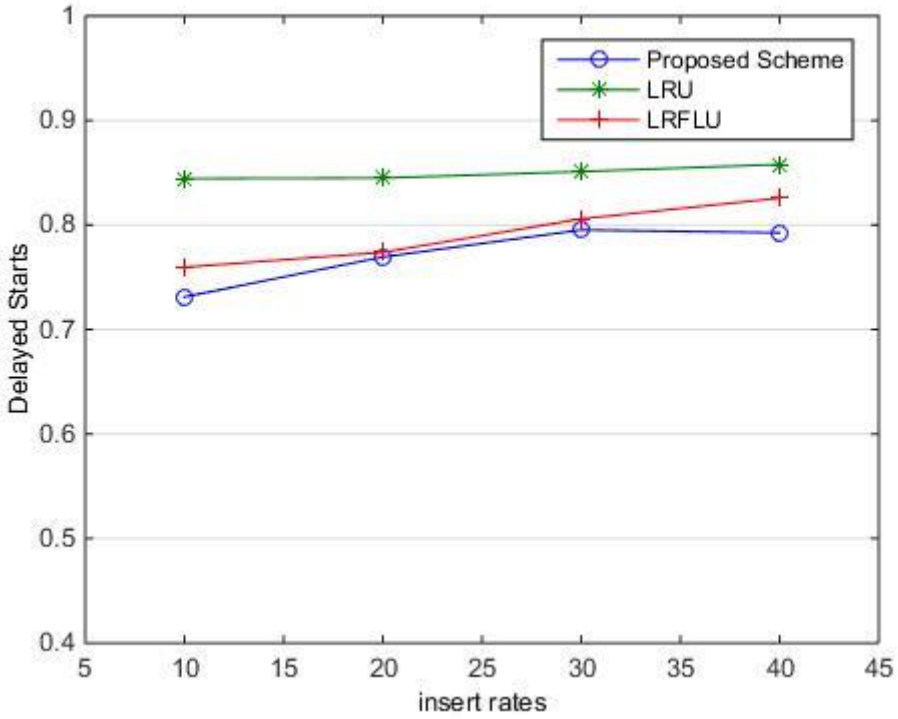
Στην ενότητα αυτή μεταβάλλουμε τον αριθμό εισαγωγής νέων αντικειμένων από 10 έως 40 videos, το οποίο αντιστοιχεί από 0.5% έως 2% του μεγέθους της βάσης δεδομένων του εξυπηρετητή. Η αναλογία ως προς της συνολική χωρητικότητα της μνήμης cache είναι για την $Cache_1$=90% και για την $Cache_2$=10%. Στην προσομοίωση θέτουμε s=0.8 και p=20 για τις τιμές των παραμέτρων της Mzipf κατανομής.

. Στο *[Σχήμα 18]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση του αριθμό εισαγωγής αντικειμένων video για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σε αυτήν την εργασία σχήμα. Στο *[Σχήμα 19]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος Delayed Starts σαν συνάρτηση του αριθμό εισαγωγής αντικειμένων video για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα.

Παρατηρούμε ότι το προτεινόμενο σχήμα εμφανίζει υψηλότερο BHR και χαμηλότερο Delayed Starts συγκριτικά με τις μεθόδους LRFLU και LRU. Όσο ο ρυθμός εισαγωγής των videos αυξάνεται, τόσο το BHR μειώνεται και το Delayed Starts αυξάνεται. Ο αλγόριθμος μας περιορίζεται από την εισαγωγή νέων αντικειμένων video στον εξυπηρετητή και τα οφέλη του περιορίζονται, καθώς αυξάνουμε τον αριθμό. Τα αποτελέσματα της προσομοίωσης δείχνουν ότι η επίδοση της μεθόδου LRFLU επηρεάζεται περισσότερο όσο αυξάνεται ο ρυθμός εισαγωγής των νέων videos.



Σχήμα 18 BHR versus insert rates (dynamic server)



Σχήμα 19 Delayed Starts versus insert rates (dynamic server)

4.3.3 Η επίδραση του ρυθμού γήρανσης της δημοτικότητας των αντικειμένων video στις μετρικές απόδοσης

Στην ενότητα αυτή μελετούμε την επίδραση του ρυθμού γήρανσης της δημοτικότητας των αντικειμένων video, πράγμα το οποίο σημαίνει ότι η δημοτικότητα των αποθηκευμένων αντικειμένων video αλλάζει με τη πάροδο του χρόνου. Μελετούμε πως η μεταβολή της δημοτικότητας των videos επηρεάζει το σύστημα διαχείρισης της μνήμης προσωρινής αποθήκευσης. Ο συνολικός αριθμός των videos στον εξυπηρετητή παραμένει σταθερός και δεν έχουμε εισαγωγή/εξαγωγή αντικειμένων video (στατικός server).

Υποθέτουμε ότι στο τέλος κάθε ημέρας ένας περιοριορισμένος αριθμός X αντικειμένων από τα λιγότερο δημοφιλή αντικείμενα στον εξυπηρετητή αυξάνουν τη δημοτικότητα τους και μπαίνουν στην κορυφή της λίστας δημοτικότητας, ενώ τα υπόλοιπα αντικείμενα χάνουν X θέσεις κατάταξης στη λίστα. Αυτό γίνεται για να μιμηθούμε το παρακάτω σενάριο: i) οι ταινίες που είναι δημοφιλείς κάποια χρονική στιγμή παύουν να είναι δημοφιλείς με το πέρασμα του χρόνου και ii) ένας αριθμός αντικειμένων video έχει ραγδαία αύξηση της δημοτικότητας τους (viral). Στην ενότητα αυτή ο ρυθμός γήρανσης της δημοτικότητας είναι ίσο με X=10 videos ανά ημέρα.

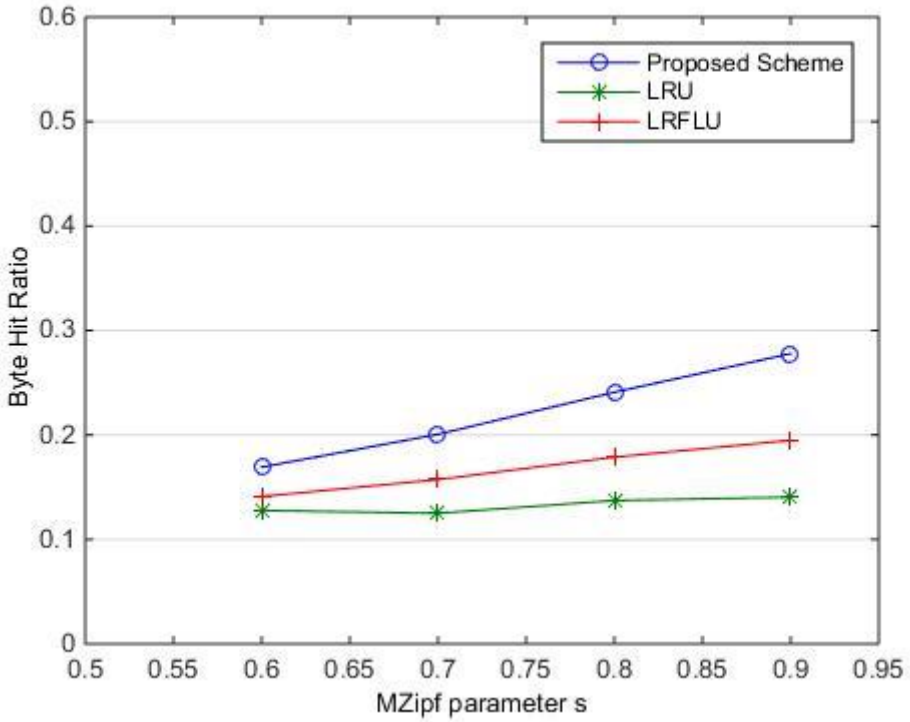
4.3.3.1 Η επίδραση διαφορετικών τιμών των παραμέτρων της Mzipf στις μετρικές απόδοσης

Στην ενότητα αυτή προσομοιώνουμε την περίπτωση γήρανσης της δημοτικότητας των videos, για διαφορετικές τιμές της παραμέτρου s της κατανομής Mzipf. Διατηρούμε σταθερή την παράμετρο p που καθορίζει το πλάτωμα της κατανομής δημοτικότητας των videos, καθώς μεταβάλλουμε την παράμετρο s (η οποία καθορίζει την πολικότητα) από 0.6 έως 0.9. Στη συνέχεια προσομοιώνουμε την περίπτωση γήρανσης της δημοτικότητας των videos, για διαφορετικές τιμές της παραμέτρου p της κατανομής Mzipf. Διατηρούμε σταθερή την παράμετρο s που καθορίζει την πολικότητα της κατανομής δημοτικότητας των videos και μεταβάλλουμε την παράμετρο p (η οποία καθορίζει το πλάτωμα της κατανομής) από 10 έως 40.

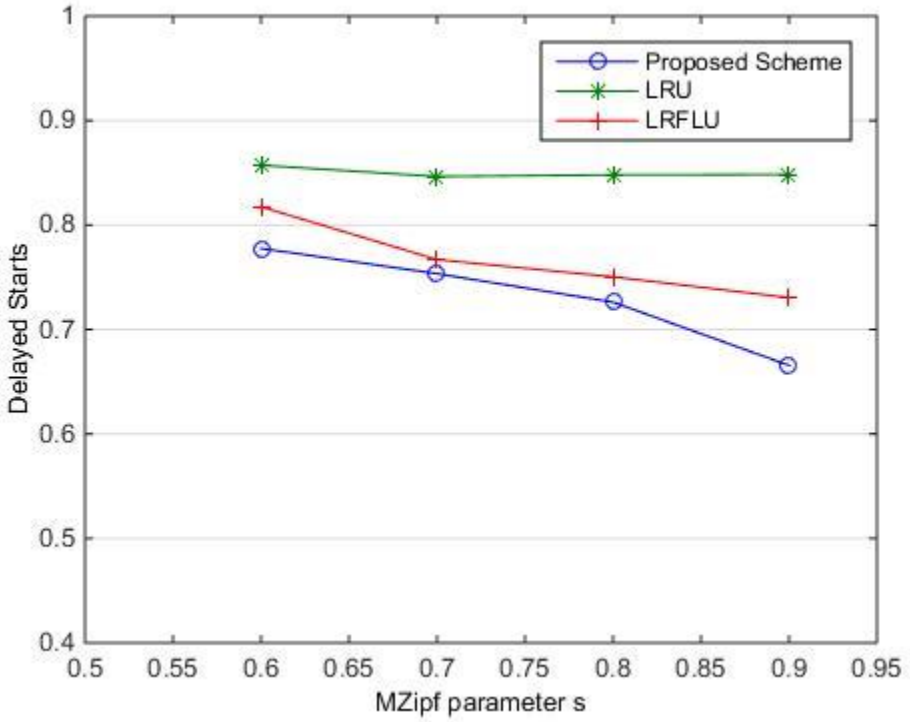
Στο *[Σχήμα 20]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους s για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα. Στο *[Σχήμα 21]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος Delayed Starts σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους s για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα. Στο *[Σχήμα 22]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους p για τις μεθόδους αντικατάστασης LRU, LRFLU και το προτεινόμενο σχήμα. Στο *[Σχήμα 23]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος Delayed Starts σαν συνάρτηση των διαφορετικών τιμών της παραμέτρους p για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σχήμα.

Παρατηρούμε ότι το προτεινόμενο σχήμα εμφανίζει υψηλότερο BHR και χαμηλότερο Delayed Starts σε σχέση με αυτά των μεθόδων LRFLU και LRU. Το προτεινόμενο σχήμα φαίνεται να λειτουργεί πιο αποδοτικά (όπως και στις περιπτώσεις με στατικό και δυναμικό server), όταν οι αιτήσεις αφορούν ένα μικρό υποσύνολο δημοφιλών αντικειμένων video. Πιο συγκεκριμένα, παρατηρούμε ότι η διαφορά του BHR και των Delayed Starts του προτεινόμενου σχήματος μεγαλώνει σε σχέση με τις αντίστοιχες τιμές των μεθόδων LRFLU και LRU, τόσο στην περίπτωση που αυξάνεται η παράμετρος s, όσο και στην περίπτωση που μειώνεται η παράμετρος p της κατανομής MZipf.

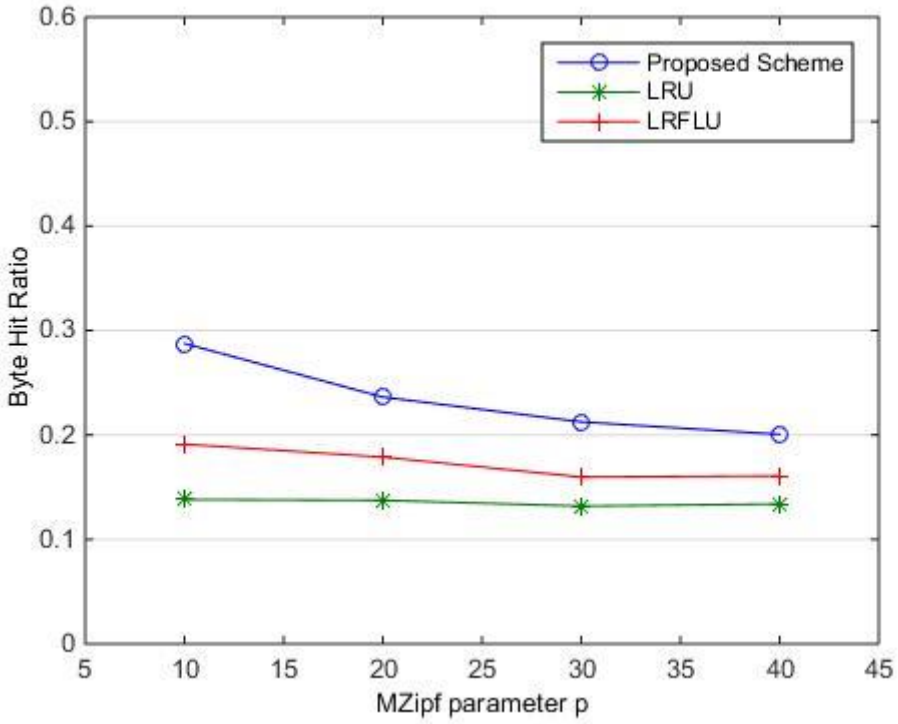
.



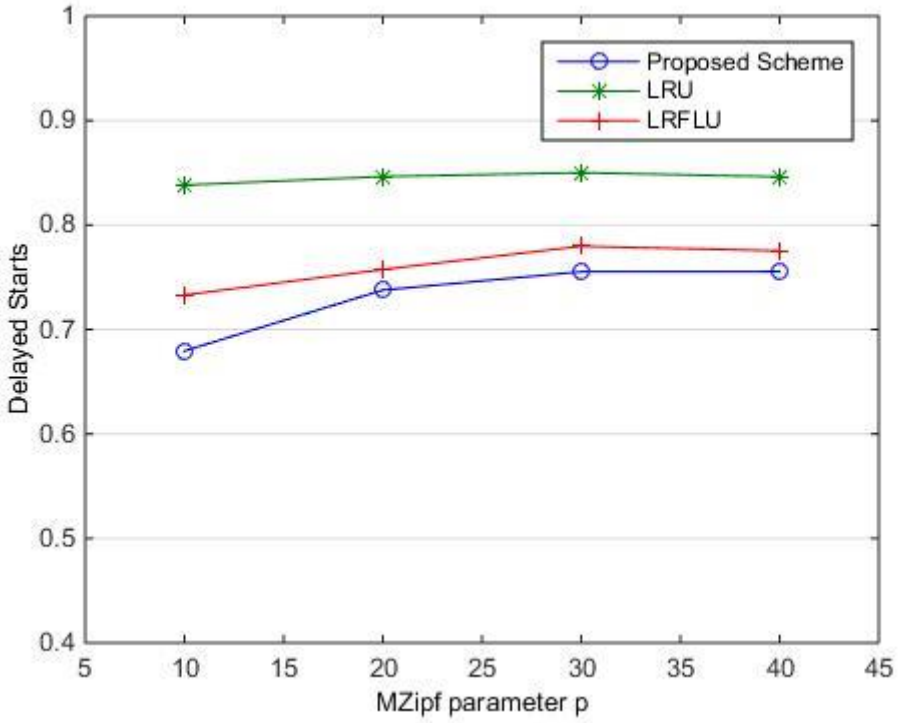
Σχήμα 20 BHR versus MZipf parameter s (ageing)



Σχήμα 21 Delayed Starts versus MZipf parameter s (ageing)



Σχήμα 22 BHR versus MZipf parameter p (ageing)



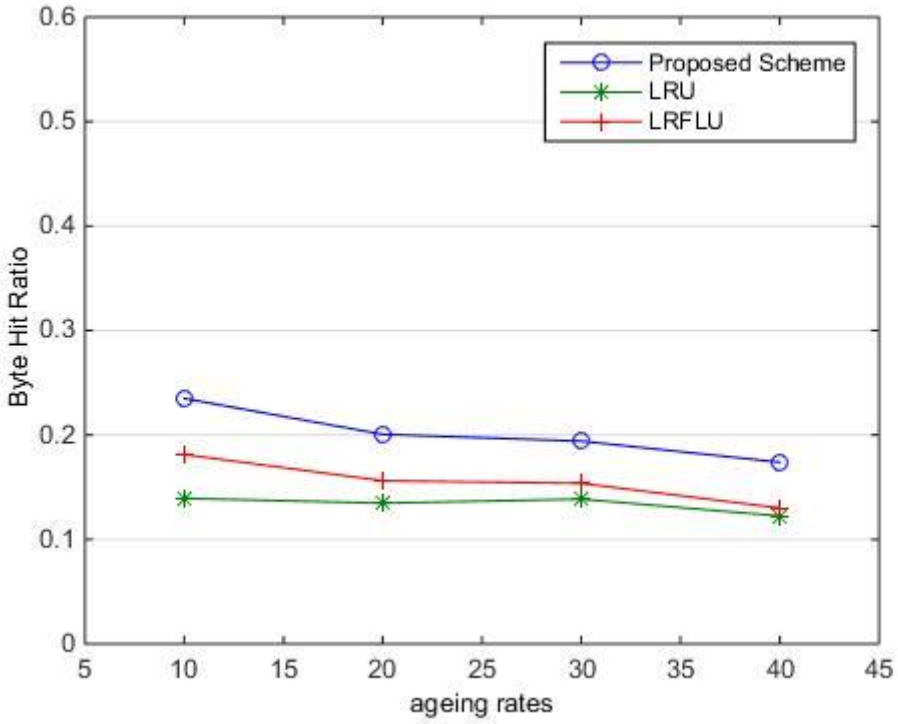
Σχήμα 23 Delayed Starts versus MZipf parameter p (ageing)

4.3.3.2 Η επίδραση διαφορετικού ρυθμού γήρανσης στις μετρικές απόδοσης

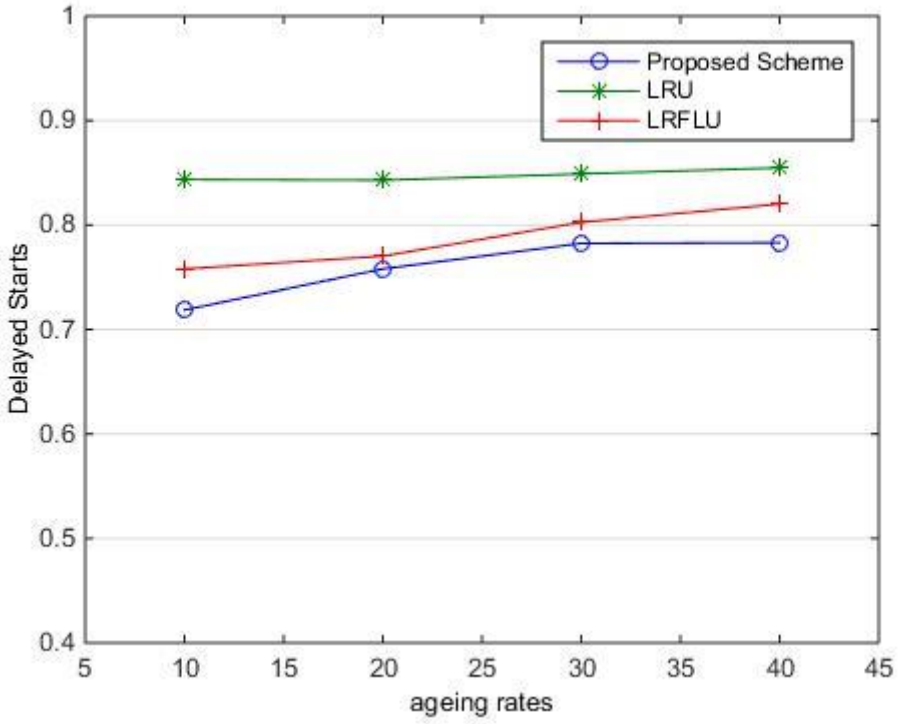
Στην ενότητα αυτή μεταβάλλουμε τον ρυθμό γήρανσης της δημοτικότητας των αντικειμένων από 10 έως 40 videos, το οποίο αντιστοιχεί από 0.5% έως 2% του μεγέθους της βάσης δεδομένων του εξυπηρετητή. Σε αυτή την περίπτωση, ένας περιοριορισμένος αριθμός από X=10 έως 40 αντικείμενα με τα λιγότερο δημοφιλή αντικείμενα μπαίνουν στην κορυφή της λίστας δημοτικότητας, ενώ τα υπόλοιπα αντικείμενα χάνουν X θέσεις κατάταξης στη λίστα στο τέλος κάθε ημέρας προσομοίωσης. Η αναλογία προς της συνολική χωρητικότητα της μνήμης cache είναι για την $Cache_1$=90% και για την $Cache_2$=10%. Στην προσομοίωση θέτουμε s=0.8 και p=20 για τις τιμές των παραμέτρων της Mzipf κατανομής.

. Στο *[Σχήμα 24]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος BHR σαν συνάρτηση του ρυθμού γήρανσης της δημοτικότητας των αντικειμένων video για τις μεθόδους αντικατάστασης LRU, LRFLU και για το προτεινόμενο σε αυτήν την εργασία σχήμα. Στο *[Σχήμα 25]* παρουσιάζονται τα αποτελέσματα της μετρικής απόδοσης του συστήματος Delayed Starts σαν συνάρτηση του ρυθμού γήρανσης της δημοτικότητας των αντικειμένων video για τις μεθόδους αντικατάστασης LRU, LRFLU και το προτεινόμενο σχήμα.

Παρατηρούμε ότι το προτεινόμενο σχήμα εμφανίζει υψηλότερο BHR και χαμηλότερο Delayed Starts συγκριτικά με τις αντίστοιχες τιμές των μεθόδων LRFLU και LRU. Όσο ο ρυθμός γήρανσης της δημοτικότητας των videos αυξάνεται, τόσο το BHR μειώνεται και αυξάνεται το Delayed Starts. Τα οφέλη του αλγορίθμου μας περιορίζονται, καθώς αυξάνεται ο ρυθμός γήρανσης της δημοτικότητας των videos. Τα αποτελέσματα της προσομοίωσης δείχνουν ότι η επίδοση της μεθόδου LRFLU επηρεάζεται περισσότερο όσο αυξάνεται ο ρυθμός γήρανσης της δημοτικότητας των videos.



Σχήμα 24 BHR versus ageing rates (ageing)



Σχήμα 25 Delayed Starts versus ageing rates (ageing)

Κεφάλαιο 5

5.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία προτείνεται ένα δυναμικό σχήμα προσωρινής αποθήκευσης τμημάτων αντικειμένων video σε ένα σύστημα παροχής υπηρεσιών κατ' απαίτηση. Το προτεινόμενο σχήμα διαμερίζει την κρυφή μνήμη σε δύο επιμέρους caches για να αποφευχθεί η πρώιμη έξοδος δημοφιλών αντικειμένων που παροδικά δε ζητούνται συχνά, από μη δημοφιλή αντικείμενα που παροδικά ζητούνται με μεγαλύτερη από την συχνότητα που τους αντιστοιχεί. Τα αντικείμενα video είναι χωρισμένα σε ίσα τμήματα σταθερού μήκους ώστε να μειώνεται η πολυπλοκότητα της διαδικασίας διαχείρισης της κρυφής μνήμης. Τα αποτελέσματα της μελέτης μας μέσω προσομοίωσης έδειξαν ότι το προτεινόμενο σύστημα διαχείρισης της κρυφής μνήμης λειτουργεί πιο αποτελεσματικά, όσον αφορά τις μετρικές απόδοσης Byte Hit Ratio (BHR) και Fraction of Delayed Starts, σε σύγκριση με τις τεχνικές προσωρινής αποθήκευσης LRU και LRLFU στις παρακάτω περιπτώσεις που εξετάστηκαν αναλυτικά: i) με στατικό εξυπηρετητή παροχής αντικειμένων video, ii) με δυναμικό εξυπηρετητή παροχής αντικειμένων video και iii) με ύπαρξη ρυθμού γήρανσης στις δημοτικότητες των αντικειμένων video.

5.2 Ιδέες για μελλοντική εργασία

Ένα ενδιαφέρον χαρακτηριστικό που θα μπορούσε να προστεθεί στο σύστημα και να εξετασθεί είναι η υποστήριξη περισσότερων της μίας εκδόσεων για κάθε αντικείμενο video, όπου η κάθε έκδοση θα αντιστοιχεί σε διαφορετική ποιότητα ανάλυσης και θα έχει φυσικά διαφορετικό μέγεθος video. Ένα άλλο χαρακτηριστικό που θα μπορούσε να προστεθεί στο σύστημα και να εξετασθεί είναι η υποστήριξη δυναμικών αλλαγών στα πρότυπα προβολής των χρηστών (skip, pause, rewind, fast forward). Τέλος, η διαμέριση της cache θα μπορούσε να λειτουργήσει με διαφορετικό τρόπο από τον τρόπο που χρησιμοποιήθηκε στην εργασία, να χρησιμοποιηθεί το ένα μέρος της cache για να προφορτώνει (prefetch) ένα μέρος των πιο δημοφιλών αντικειμένων video.

Βιβλιογραφία

[1] Krishna Mohan Agrawal, T.Venkatesh , Deep Medhi, "A Dynamic Popularity-based Partial Caching Scheme for Video on Demand Service in IPTV Networks", in Proceedings of the Sixth International Conference on Communication Systems and Networks, Bangalore, India, January 2014.

[2] S. Chen, H. Wang, X. Zhang, B. Shen, S. Wee, "Segment-based proxy caching for Internet streaming media delivery", IEEE Multimedia, vol. 12, no. 3, pp. 59–67, July-September 2005.

[3] UmaMaheswari Devi, Ramana Polavarapu, Malolan Chetlur, Shivkumar Kalyanaraman, "On the partial caching of streaming video", in Proceedings of the 2012 IEEE 20th International Workshop on Quality of Service.

[4] D. De Vleeschauwer, K. Laevens, "Performance of caching algorithms for IPTV on-demand services", IEEE Transactions on Broadcasting, vol. 55, no. 2, pp. 491–501, 2009.

[5] J. Liu and J. Xu, "Proxy caching for media streaming over the Internet ", IEEE Communications Magazine, vol. 42, no. 8, pp. 88–94, 2004.

[6] Osama Saleh, «Modeling and Caching of peer-to-peer traffic", in Proceedings of the 2006 IEEE International Conference on Network Protocols.

[7] K.-L. Wu, P. S. Yu, J. L. Wolf, "Segment-based proxy caching of multimedia streams", in Proceedings of the 10th International Conference on World Wide Web, 2001, pp. 36–44.

[8] D. K. Krishnappa et al., "On the feasibility of prefetching and caching for online tv services: a measurement study on hulu", in Proceedings of the 12th international conference on Passive and active measurement, Berlin, Heidelberg: Springer-Verlag, 2011, pp. 72–80.

[9] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," IEEE/ACM Transactions on Networking, vol. 16, no. 6, pp. 1447–1460, 2008.

[10] B. Krogfoss, L. Sofman, A. Agrawal, “Caching architectures and optimization strategies for IPTV networks”, Bell Labs Technical Journal, vol. 13, no. 3, pp. 13–28, 2008.

[11] S. Sen, J. Rexford, D. Towsley, “Proxy prefix caching for multimedia streams”, in Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communication Societies, 1999, pp. 1310–1319.

[12] Henrik Abrahamsson, Mats Björkman, “Simulation of IPTV caching strategies”, in Proceedings of the 2010 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS).

[13] Jeroen Famaey, , Frédéric Iterbeke, Tim Wauters , Filip De Turck, “Towards a predictive cache replacement strategy for multimedia content”, Journal of Network and Computer Applications, Volume 36, Issue 1, Pages 219–227, 2013.

[14] S. Zeadally, H. Moustafa, and F. Siddiqui, “Internet protocol television (IPTV): Architecture, trends, and challenges”, IEEE Systems Journal, vol. 5, no. 4, pp. 518–527, 2011.

[15] N. Degrande, K. Laevens, D. De Vleeschauwer, R. Sharpe, “Increasing the user perceived quality for IPTV services”, IEEE Communications Magazine, vol. 46, no. 2, pp. 94–100, Feb. 2008.

[16] Anna Satsiou, Michael Paterakis, “Impact of Frequency-Based Cache Management Policies on the Performance of Segment Based Video Caching Proxies”, in Proc. of the 3rd IFIP Networking Conference, Athens, Greece, May, 2004.