



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχήμα Προφόρτωσης Περιεχομένου WWW

Άγγελος Γ. Κατσής

Επιβλέπων: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΜΑΙΟΣ 2006

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχήμα Προφόρτωσης Περιεχομένου WWW

Άγγελος Γ. Κατσης

A.M.: M595

ΕΠΙΒΛΕΠΩΝ:

Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:

Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

Λάζαρος Μεράκος, Καθηγητής ΕΚΠΑ

Ημερομηνία Εξέτασης 20/05/2006

ΠΕΡΙΛΗΨΗ

Η ραγδαία ανάπτυξη του Internet έχει καταστήσει απαραίτητη την χρήση ενδιάμεσων μνημών (caches) έτσι ώστε να επιτευχθεί τόσο η μείωση του καταναλισκόμενου εύρους ζώνης στο δίκτυο κορμού, ο φόρτος στους εξυπηρετές πηγής (origin servers), αλλά και η εκλαμβανόμενη από τους χρήστες καθυστέρηση. Στόχος είναι οι πόροι να βρίσκονται όσο πιο κοντά γίνεται στους χρήστες. Μια τεχνική που χρησιμοποιείται επιπλέον της απλής αποθήκευσης αντικειμένων που έχουν ήδη ζητηθεί σε ενδιάμεσες μνήμες (caching) είναι η προανάκτηση αντικειμένων. Στόχος αυτής της τεχνικής είναι η περαιτέρω μείωση της εκλαμβανόμενης από τους χρήστες καθυστέρησης στην λήψη ενός πόρου. Στην παρούσα εργασία εξετάστηκε ένας μηχανισμός διαιτησίας για τον κατάλληλο περιορισμό αιτήσεων προανάκτησης πόρων που σχηματίζονται σύμφωνα με τον αλγόριθμο προανάκτησης πόρων Top-10 ώστε να πληρείται ένας περιορισμός χρόνου όσον αφορά το χρονικό περιθώριο ικανοποίησης της αίτησης προανάκτησης αλλά και την όσο πιο δίκαιη εξυπηρέτηση των χρηστών. Μελετήθηκαν οι πλέον δημοφιλείς αρχιτεκτονικές προανάκτησης αντικειμένων (WWW prefetching) (όπως η TOP-10) καθώς και οι εσωτερικοί αλγόριθμοι αυτών. Για το σχεδιασμό του σχήματος διαιτησίας επηρεαστήκαμε από τη Θεωρία Παιγνίων και συγκεκριμένα από τη λύση διαπραγμάτευσης του Nash (Nash Bargaining Solution) Για τον τελικό καθορισμό της μορφής των αιτήσεων προανάκτησης χρησιμοποιήθηκε μια μορφή του προβλήματος του σακιδίου. Ο αλγόριθμος διαιτησίας προσομοιώθηκε για την αρχιτεκτονική τεσσάρων τερματικών κόμβων και δυο επιπέδων ιεραρχίας από caching proxies. Τα αποτελέσματα δείχνουν ότι το σχήμα διαιτησίας οδηγεί στη μείωση των απαιτούμενων πόρων του δικτύου για την πραγματοποίηση της προανάκτησης ενώ ναι μεν το λόγος επιτυχιών μειώνεται, ωστόσο δεν μειώνεται το ίδιο και η ποιότητα υπηρεσίας που απολαμβάνουν οι χρήστες όσον αφορά την καθυστέρηση. Συγκεκριμένα η τελευταία είναι παραπλήσια αυτής του αλγορίθμου Top 10 χωρίς σχήμα διαιτησίας.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Προανάκτηση Πόρων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: προανάκτηση, σχήμα διαιτησίας, θεωρία παιγνίων, αλγόριθμος Top 10, πρόβλημα σακιδίου

ABSTRACT

The rapid development of Internet has rendered essential the use of caches so as to achieve the reduction of consumed bandwidth in the core network, the load in the origin servers, as well as the delay considered from users. The objective is that the resources are as near as possible to the users. A technique that is used on top of caching is the prefetching of resources. The objective of this technique is a further reduction of the delay considered from users during the reception of a resource. In the present work a mechanism of arbitration was examined for the suitable restriction of prefetch requests which are constructed according to the Top-10 prefetching algorithm so that a certain time restriction concerning the time limit for the accomplishment of the prefetch request is not violated and the fairer service of users is achieved. The most popular architectures of WWW pre-fetching (as the TOP-10) were studied as well as the internal algorithms of these architectures. As far as the design of the arbitration mechanism is concerned we have been influenced by the Theory of Games and precisely by the Nash Bargaining Solution. For the final formation of the prefetch requests, a solution to a variation of the knapsack problem was used. The algorithm of arbitration was simulated in an architecture of four terminal nodes and a two level hierarchy of caching proxies. The results show that the proposed scheme of arbitration leads to the reduction of the required network resources that are needed for the prefetching, while although the Hit Rate is decreased, the quality of service that the users enjoy with regard to the delay does not have a similar decrease. In fact the last one is similar to that of the Top 10 algorithm without the application of arbitration.

SUBJECT AREA: Web Prefetching

KEYWORDS: prefetching, arbitration scheme, game theory, Top 10 algorithm, knapsack problem

ΠΡΟΛΟΓΟΣ	7
ΚΕΦΑΛΑΙΟ: 1 CACHING ΣΤΟ ΔΙΑΔΙΚΤΥΟ	8
1.1 Γενική περιγραφή και στόχοι.....	8
1.2 Οφέλη από την ανάπτυξη της τεχνολογίας ενδιάμεσης αποθήκευσης	9
1.3 Πόροι που μπορούν να αποθηκευθούν σε ενδιάμεση μνήμη.....	9
1.4 Είδη ενδιάμεσων μνημών.....	12
1.5 Πώς υλοποιείται η ενδιάμεση αποθήκευση.....	14
1.6 Διαγραφή και Αποθήκευση Πόρων στην Ενδιάμεση μνήμη	14
1.7 Επιστροφή αποθηκευμένου πόρου ή προώθηση αίτησης.....	15
1.8 Συντήρηση ενδιάμεσης μνήμης	16
1.9 Στρατηγικές αντικατάστασης αντικειμένων ενδιάμεση μνήμη.....	16
1.10 Συνάφεια της ενδιάμεσης μνήμης.....	18
1.11 Επικοινωνία μεταξύ ενδιάμεσων μνημών	20
1.11.1 Internet Cache Protocol (ICP).....	21
1.11.2 Cache Array Resolution Protocol (CARP).....	22
1.11.3 Cache Digest Protocol (CDP).....	22
1.11.4 Web Cache Coordination Protocol (WCCP)	23
1.12 Διανομή Περιεχομένου (Content Distribution).....	23
1.13 Αρχιτεκτονικές ενδιάμεσων μνημών	25
1.13.1 Μεσάζων εξυπηρέτης με ενδιάμεση μνήμη	26
1.13.2 Ανάστροφος μεσάζων εξυπηρέτης με ενδιάμεση μνήμη.....	26
1.13.3 Διαφανής ενδιάμεση αποθήκευση.....	27
ΚΕΦΑΛΑΙΟ: 2 ΠΡΟΑΝΑΚΤΗΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ	28
2.1 Γενική περιγραφή.....	28
2.2 Αλγόριθμοι προανάκτησης αντικειμένων.....	30
2.2.1 Top-10.....	30
2.2.2 Προανάκτηση βασισμένη στην πρόβλεψη (Predictive Prefetching)	34
2.3 Επιπτώσεις της προανάκτησης στο δίκτυο	37
2.4 Αλγόριθμοι προανάκτησης σε δίκτυο διανομής περιεχομένου.....	38
2.4.1 Γενικά	38
2.4.2 Popularity.....	39
2.4.3 Prefetching by Lifetime	39
2.4.4 Prefetching by Good-Fetch.....	39
2.4.5 Prefetching by APL	40
2.4.6 Χαρακτηριστικά σταθερής κατάστασης.....	40
2.5 Αξιολόγηση των αλγορίθμων	42
2.5.1 Προβλήματα των προαναφερθέντων αλγορίθμων.....	43
2.6 H/B Greedy προανάκτηση	43
2.6.1 Hit Rate Greedy προανάκτηση	45
2.6.2 Bandwidth Greedy προανάκτηση	46
ΚΕΦΑΛΑΙΟ: 3 ΘΕΩΡΙΑ ΠΑΙΓΝΙΩΝ	47

Σχήμα Προφόρτωσης Περιεχομένου WWW

3.1	Γενικά	47
3.2	Συνεργασία	49
3.3	Λύση διαπραγμάτευσης του Nash	52
3.4	Απόδειξη	55
ΚΕΦΑΛΑΙΟ: 4 ΠΑΙΓΝΙΟΘΕΩΡΗΤΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΣΕ ΔΙΚΤΥΑΚΑ ΠΡΟΒΛΗΜΑΤΑ.57		
4.1	Εκχώρηση εύρους ζώνης σε εικονικά μονοπάτια για δίκτυα πολλαπλών χρηστών	57
4.2	Παιγιοθεωρητική ανάλυση των αρχών ελέγχου συμφόρησης	60
4.3	Αρχιτεκτονική μη-συνεργατικών δικτύων	63
4.4	Παιγιοθεωρητική κατανομή εύρους ζώνης σε ευρυζωνικά δίκτυα	63
4.4.1	<i>Βέλτιστες για το δίκτυο εκχωρήσεις ρυθμού</i>	<i>65</i>
4.4.2	<i>Το πρόβλημα από τη μεριά το χρήστη</i>	<i>67</i>
4.4.3	<i>Τιμολόγηση</i>	<i>68</i>
4.5	Περίπου βέλτιστος σχεδιασμός δικτύου με πράκτορες που συμπεριφέρονται εγωιστικά	69
4.6	Εγωιστική ενδιάμεση αποθήκευση σε καταναμημένα συστήματα: μια παιγιοθεωρητική ανάλυση	72
4.6.1	<i>Βασικό παίγνιο</i>	<i>73</i>
4.6.2	<i>Παίγνιο πληρωμών</i>	<i>75</i>
ΚΕΦΑΛΑΙΟ: 5 ΠΡΟΤΕΙΝΟΜΕΝΟ ΣΧΗΜΑ ΠΡΟΑΝΑΚΤΗΣΗΣ WEB ΠΕΡΙΕΧΟΜΕΝΟΥ76		
5.1	Σχήμα Διαχείρισης προανάκτησης WWW Πόρων	78
5.2	Λειτουργία του διαιτητή με τη μέθοδο του σακιδίου	84
5.3	Αποτελέσματα Προσομοίωσης	88
5.4	Παρατηρήσεις	89
ΚΕΦΑΛΑΙΟ: 6 ΣΥΜΠΕΡΑΣΜΑΤΑ90		
ΑΚΡΩΝΥΜΙΑ.....92		
ΟΡΟΛΟΓΙΑ.....93		
ΑΝΑΦΟΡΕΣ94		

Εικόνες

Εικόνα 1: a) αυτόνομο, b) διάφανο στον δρομολογητή, και c) διάφανο στον μεταγωγέα proxy caching... 26	
Εικόνα 2: Top-10 προανάκτηση	32
Εικόνα 3: Συνεργασία πράκτορα προανάκτησης στην μεριά του πελάτη και http εξυπηρέτη	33
Εικόνα 4: Αρχιτεκτονική συστήματος προανάκτησης	35
Εικόνα 5: Τμήμα του γράφου εξαρτήσεων	36
Εικόνα 6: Αλγόριθμος H/B-Greedy	45
Εικόνα 7: Δικτυακή Αρχιτεκτονική	76

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του Μεταπτυχιακού Προγράμματος Σπουδών του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Το αντικείμενο μελέτης της είναι ο σχεδιασμός και η ανάπτυξη ενός συστήματος διαιτησίας για την προανάκτηση αντικειμένων στον παγκόσμιο ιστό. Το σχήμα διαιτησίας χρησιμοποιεί δεδομένα όπως η μέση καθυστέρηση για την λήψη ενός αντικειμένου από κάθε εξυπηρέτη, τη δημοφιλία των αντικειμένων, αλλά και το από πόσους χρήστες ταυτόχρονα ζητείται η προανάκτηση αντικειμένων από ένα συγκεκριμένο εξυπηρέτη. Στόχος είναι μείωση τη εκλαμβανόμενης από τους χρήστες καθυστέρησης με ένα δίκαιο τρόπο καθώς και η μείωση καταναλισκόμενων πόρων του δικτύου

Θα ήθελα να εκφράσω τις θερμές ευχαριστίες μου στον επίκουρο καθηγητή κ. Ευστάθιο Χατζηευθυμιάδη για την πολύ σημαντική καθοδήγησή του και τις εύστοχες επισημάνσεις του καθ' όλη την διάρκεια εκπόνησης της εργασίας. Θα ήθελα επίσης να ευχαριστήσω την οικογένεια μου και τα κοντινά μου πρόσωπα για την συμπαράσταση και στήριξη που μου προσέφεραν.

Άγγελος Γ. Κασής

ΚΕΦΑΛΑΙΟ: 1

CACHING ΣΤΟ ΔΙΑΔΙΚΤΥΟ

1.1 Γενική περιγραφή και στόχοι

Με την ραγδαία ανάπτυξη της κίνησης στο Διαδίκτυο, το caching ήταν η πρώτη βασική τεχνική που στόχευε στην ελαχιστοποίηση της υστέρησης που αντιλαμβάνονται οι χρήστες του Διαδικτύου καθώς και στον περιορισμό της μετάδοσης πλεονάζουσας πληροφορίας στο δίκτυο. Επίσης διαπιστώθηκε ότι πολλαπλοί χρήστες σε ένα οργανισμό επισκέπτονται το ίδιο σύνολο λίγων σχετικά δικτυακών τόπων. Το RFC 2616 καθορίζει ότι μία ενδιάμεση μνήμη (cache) είναι ένας τοπικός χώρος αποθήκευσης μηνυμάτων απάντησης. Ένας λιγότερο αυστηρός ορισμός του caching είναι η μετακίνηση web περιεχομένου πλησιέστερα στους τελικούς χρήστες. Το Web caching είναι η πλέον μελετημένη εφαρμογή του Παγκόσμιου Ιστού. Σήμερα υπάρχουν πολλαπλά προϊόντα για caching είτε στην μορφή λογισμικού είτε στην μορφή υλικού.

Ο πρώτος εξυπηρέτης Διαδικτύου που κατασκευάστηκε στο CERN είχε ένα συσχετιζόμενο proxy server που συμπεριλάμβανε μία ενδιάμεση μνήμη. Ένα από τα πρώτα έργα πάνω στην τεχνολογία του caching ήταν το harvest το οποίο δεικτοδοτούσε πληροφορία στο πλαίσιο του Διαδικτύου. Το harvest επεδίωκε την ενδιάμεση αποθήκευση (caching) και την διατήρηση αντιγράφων (replicate) πληροφορίας η οποία συγκεντρώνονταν με διάφορα εργαλεία. Μία ιεραρχική ενδιάμεση μνήμη αντικειμένων ήταν το βασικό τμήμα της αρχιτεκτονικής του harvest. Ένας μηχανισμός εγγραφής των εξυπηρετών (registry) επέτρεπε την συλλογή πληροφοριών για άλλες ενδιάμεσες μνήμες (caches). Το μητρώο (registry) ανταποκρινόταν σε αιτήματα για την τοποθεσία των ενδιάμεσων μνημών. Οι ενδιάμεσες μνήμες που βρίσκονταν στην ίδια τοποθεσία (collocated) μπορούσαν να οργανωθούν σε δεξαμενές (pools) πόρων.

Η ενδιάμεση αποθήκευση αποτέλεσε σημαντικότερο συστατικό του WWW από την περίοδο που δημοσιεύτηκε η προδιαγραφή HTTP/1.0. Τα πρώτα πειράματα στην τεχνολογία ενδιάμεσης αποθήκευσης επέδειξαν σημαντικές μειώσεις στον όγκο της ανταλλασσόμενης πληροφορίας. Οι ενδιάμεσες μνήμες (caches) άρχισαν να διατηρούν μία πληθώρα πληροφοριών σχετικά τα έγγραφα που είχαν αποθηκευμένα όπως π.χ., τύπος περιεχομένου (content type), μέγεθος και μέσος χρόνος μεταξύ αλλαγών-μεταβολών. Θέματα συνέπειας καθώς και η εξουσιοδοτημένη πρόσβαση στους πόρους

της ενδιάμεσης μνήμης άρχισαν να απασχολούν την κοινότητα του WWW. Ομάδες ενδιάμεσων μνημών (caches) οργανώθηκαν σε ιεραρχίες οι οποίες κάλυπταν ολόκληρες περιοχές ή ακόμη και κράτη.

Αναλυτικά οι στόχοι της ενδιάμεσης αποθήκευσης είναι οι εξής:

- Να μειώσει τον χρόνο που αντιλαμβάνεται ο τελικός χρήστης από την υποβολή μίας αίτησης μέχρι την παρουσίαση του ζητούμενου πόρου στο τερματικό του,
- Να μειώσει τον φόρτο στο δίκτυο αποφεύγοντας την επανάληψη της μετάδοσης του ιδίου πόρου,
- Να μειώσει τον φόρτο στον εξυπηρέτη πηγής με την παρεμβολή ενός μεσάζοντα μεταξύ του πελάτη και του εξυπηρέτη πηγής που θα χειρίζεται αιτήσεις.

1.2 Οφέλη από την ανάπτυξη της τεχνολογίας ενδιάμεσης αποθήκευσης

Οι εταιρίες φιλοξενίας δικτυακών τόπων (WWW hosting) πρέπει να πληρώσουν για το εύρος ζώνης το οποίο χρησιμοποιούν και θέλουν να αυξήσουν την δυνατότητα ενδιάμεσης αποθήκευσης για να μειώσουν τα έξοδά τους. Όλοι οι εμπλεκόμενοι στην αλυσίδα του WWW ωφελούνται από την εισαγωγή της τεχνολογίας ενδιάμεσης αποθήκευσης. Οι τελικοί χρήστες ωφελούνται γιατί μειώνεται ο χρόνος απόκρισης του συστήματος. Ένα σημαντικό ποσοστό των εγκαταλελειμμένων προσπαθειών (aborts) κατά την διάρκεια της συνόδου ενός χρηστή οφείλεται στην δυσαρέσκεια του χρήστη για την βραδύτητα άντλησης δεδομένων. Το εύρος ζώνης το οποίο δαπανάται σαν αποτέλεσμα της επαναλαμβανόμενης μετάδοσης πλεονάζουσας πληροφορίας είναι ιδιαίτερα μεγάλο. Δεδομένου του προβλήματος συμφόρησης (congestion) στο Διαδίκτυο, η ελάττωση της κίνησης ή η μετακίνηση του φόρτου προς τα άκρα του δικτύου είναι ιδιαίτερα ωφέλιμη. Τα πλεονεκτήματα για το δίκτυο είναι δύο: μόνο τα αναγκαία δεδομένα διασχίζουν το δίκτυο και υπάρχει διαθέσιμο εύρος ζώνης και για άλλα δεδομένα.

1.3 Πόροι που μπορούν να αποθηκευθούν σε ενδιάμεση μνήμη

Μία ενδιάμεση μνήμη μπορεί να αποφασίσει το αν κάποια απάντηση μπορεί να υποστεί αποθήκευση σε ενδιάμεση μνήμη ή όχι βάσει δύο παραγόντων: απαιτήσεις που σχετίζονται με το πρωτόκολλο HTTP καθώς και το υπό εξέταση περιεχόμενο. Σχετικά με το HTTP πρωτόκολλο απαιτείται οι ενδιάμεσες μνήμες να λαμβάνουν υπόψη κάποιες ντιρεκτίβες σχετικά με την δυνατότητα αποθήκευσης κάποιου συγκεκριμένου

μηνύματος. Οι απαιτήσεις οι οποίες εξαρτώνται από το περιεχόμενο επηρεάζονται από τις επιχειρηματικές απαιτήσεις μίας ενδιάμεσης μνήμης καθώς και τις πολιτικές που επηρεάζουν την συχνότητα ελέγχου της ορθότητας των περιεχομένων της ενδιάμεσης μνήμης (revalidation). Οι πολιτικές αυτές, με την σειρά τους, επηρεάζονται από τα χαρακτηριστικά όπως π.χ., το μέγεθος και τον τύπο.

Το HTTP/1.1 καθορίζει απλούς κανόνες για το ποιοι πόροι επιδέχονται αποθήκευση σε ενδιάμεση μνήμη. Η μέθοδος αίτησης, τα πεδία επικεφαλίδας της αίτησης, ο κωδικός κατάστασης της απάντησης (response status) καθώς και τα πεδία επικεφαλίδας της απάντησης πρέπει να υποδεικνύουν εάν ο πόρος επιδέχεται αποθήκευση σε ενδιάμεση μνήμη. Οι απαντήσεις στα αιτήματα OPTIONS, PUT και DELETE δεν επιδέχονται αποθήκευση σε ενδιάμεση μνήμη. Οι απαντήσεις στα αιτήματα POST δεν επιδέχονται αποθήκευση σε ενδιάμεση μνήμη εκτός εάν η απάντηση φέρει τις κατάλληλες Cache-Control και Expires επικεφαλίδες. Εάν η ενδιάμεση μνήμη δεν υποστηρίζει το πεδίο επικεφαλίδα Range, όλες οι απαντήσεις που φέρουν κωδικό κατάστασης 206 (Partial Content) δεν μπορούν να αποθηκευτούν σε ενδιάμεση μνήμη.

Ορισμένες απαντήσεις περιέχουν πληροφορία από τον εξυπηρέτη πηγής που αποκλείει την αποθήκευση του μηνύματος σε ενδιάμεσες μνήμες. Υπάρχουν 2 είδη τέτοιας πληροφορίας: πληροφορίες σχετικά με την δυνατότητα αποθήκευσης και ντιρεκτίβες προς τις ενδιάμεσες μνήμες. Εάν η απάντηση περιέχει το πρώτο είδος πληροφορίας, η απόφαση για αποθήκευση σε ενδιάμεση μνήμη θα πρέπει να βασίζεται σε αυτήν. Για παράδειγμα, ο εξυπηρέτης πηγής μπορεί να καθορίσει ακριβώς το διάστημα για το οποίο ο πόρος θα πρέπει να θεωρείται έγκυρος μέσω του πεδίου Expires. Η ντιρεκτίβα cache-control μπορεί να αποκλείσει την αποθήκευση σε ενδιάμεση μνήμη ορισμένων πόρων. Για παράδειγμα η cache-control: private καθορίζει ότι μία διαμοιραζόμενη ενδιάμεση μνήμη δεν μπορεί να αποθηκεύσει τον πόρο. Ένα μήνυμα απάντησης που περιέχει την ντιρεκτίβα cache-control: no-store δεν θα πρέπει να αποθηκευτεί καθόλου. Η ντιρεκτίβα cache-control: no-cache περιορίζει την πιθανότητα αποθήκευσης του αντικειμένου στη ενδιάμεση μνήμη γιατί θα πρέπει να ελέγχεται η ορθότητα του αντικειμένου πριν από κάθε φορά που επιστρέφεται ως επιτυχία ενδιάμεσης μνήμης (cache hit). Οι ντιρεκτίβες δεν περιορίζονται μόνο στις απαντήσεις από τον εξυπηρέτη πηγής. Μπορεί να ενσωματώνονται και στις ερωτήσεις από τον αντιπρόσωπο του χρήστη (user agent). Για παράδειγμα η cache-control: no-store μπορεί να εμφανιστεί τόσο σε απάντηση όσο και σε ερώτηση. Το πρωτόκολλο διαθέτει αυτές τις ντιρεκτίβες για να προστατεύει τον προσωπικό χαρακτήρα μίας απάντησης (privacy) και για

υποδείξει την πτητικότητα του πόρου (volatile resource) δηλαδή ότι μπορεί να αλλάξει αμέσως μετά την αποστολή του.

Η παρουσία πεδίων επικεφαλίδας όπως τα Authorization και Vary ελαχιστοποιούν τις πιθανότητες αποθήκευσης του πόρου σε ενδιάμεση μνήμη. Η επικεφαλίδα αίτησης Authorization υποδεικνύει ότι ο ζητούμενος πόρος δεν είναι διαθέσιμος για όλους. Επίσης, η παρουσία του Vary δεικνύει ότι μία αποδεκτή απάντηση για ενδιάμεση μνήμη θα πρέπει να περιορίζεται από τις τιμές που ορίζονται στο Vary πεδίο.

Μία ενδιάμεση μνήμη μπορεί να έχει και τους δικούς της κανόνες για τον έλεγχο της δυνατότητας αποθήκευσης σε ενδιάμεση μνήμη κάποιου συγκεκριμένου πόρου, άσχετα από τους περιορισμούς που επιβάλλονται μέσω του πρωτοκόλλου. Δηλαδή, αν και ο πόρος επιδέχεται αποθήκευση σε ενδιάμεση μνήμη δεν σημαίνει, κατ' ανάγκη ότι θα αποθηκευτεί και στην ενδιάμεση μνήμη. Τα μηνύματα μπορεί να είναι μεγάλα σε όγκο, να παράγονται δυναμικά, να περιέχουν cookies, παράμετροι που επηρεάζουν σημαντικά την δυνατότητα αποθήκευσης του πόρου σε ενδιάμεσες μνήμες. Οι πολιτικές αποθήκευσης σε ενδιάμεση μνήμη επηρεάζονται από χαρακτηριστικές του μηνύματος και όχι από τους περιορισμούς του πρωτοκόλλου.

Οι επιχειρήσεις μπορεί να θέλουν να περιορίσουν το κόστος μεταφοράς αγνοώντας ορισμένους περιορισμούς που σχετίζονται με την ενδιάμεση μνήμη. Για παράδειγμα μπορεί να αποθηκεύουν πόρους που δεν θα πρέπει να αποθηκεύουν (π.χ., cache-control: private). Επίσης, οι ενδιάμεσες μνήμες μπορεί να λαμβάνουν υπόψη την επιβάρυνση αποθηκευτικού χώρου (storage overhead) και να μην αποθηκεύουν ορισμένα μεγάλα αρχεία παρά το γεγονός ότι επιδέχονται αποθήκευση σε ενδιάμεση μνήμη. Εάν το αντικείμενο είναι μεγάλο, πολλοί πόροι θα πρέπει να εκτοπιστούν από την ενδιάμεση μνήμη. Αναφορικά με την χρονική υστέρηση, το κόστος ανάκτησης, από τους εξυπηρετές πηγής, μικρών σε όγκο πόρων είναι μεγαλύτερο από το κόστος ανάκτησης τους από ενδιάμεσες μνήμες. Έτσι, μία ενδιάμεση μνήμη μπορεί να αποφύγει την αποθήκευση μεγάλων πόρων. Από την άλλη πλευρά, μία μεγάλη απάντηση αποθηκευμένη σε ενδιάμεση μνήμη, εάν ζητηθεί ορισμένες φορές από τους πελάτες (clients) επιφέρει σημαντικά οφέλη εύρους ζώνης.

Πολλές ενδιάμεσες μνήμες δεν αποθηκεύουν τις απαντήσεις από scripts με το σκεπτικό ότι οι παράμετροι των σχετικών ερωτήσεων δεν πρόκειται να ξαναχρησιμοποιηθούν. Η παρουσία πρόσθετης πληροφορίας σχετικά με τη δυνατότητα αποθήκευσης σε ενδιάμεση μνήμη στις επικεφαλίδες μίας δυναμικά διαμορφούμενης απάντησης (π.χ.,

Expires ή ETag) μπορεί να σημαίνει ο πόρος μπορεί να αποθηκευτεί σε ενδιάμεση μνήμη (π.χ., ένα CGI script το οποίο επιστρέφει το n-οστό ψηφίο του αριθμού π). Πολλά WWW ερωτήματα συχνά καταλήγουν στις ίδιες απαντήσεις και υπάρχουν σήμερα ενδιάμεσες μνήμες που το λαμβάνουν αυτό υπόψη. Μία άλλη κατηγορία απαντήσεων που θεωρούνται ως μη επιδεχόμενες αποθήκευσης είναι οι απαντήσεις που εξατομικεύονται (tailored). Για παράδειγμα, οι απαντήσεις που συνοδεύονται από cookies θεωρούνται μη επιδεχόμενα ενδιάμεσης αποθήκευσης (uncacheable), γιατί εκτιμάται ότι διαφέρουν από χρήστη σε χρήστη.

Η απόφαση για την αποθήκευση κάποιου πόρου σε ενδιάμεση μνήμη εξαρτάται από τον ρυθμό αλλαγών στο συγκεκριμένο πόρο. Ορισμένοι πόροι μεταβάλλονται σπάνια, ίσως και καθόλου (π.χ., ηλεκτρονικές εκδόσεις βιβλίων). Μία παλιά ευρεστική τεχνική για τον προσδιορισμό της δυνατότητας αποθήκευσης σε ενδιάμεση μνήμη ενός πόρου ήταν η ημερομηνία τελευταίας αλλαγής. Υιοθετούνταν η υπόθεση ότι εάν ο πόρος δεν έχει μεταβληθεί για μεγάλο χρονικό διάστημα είναι χαμηλή η πιθανότητα αλλαγής του στο άμεσο μέλλον. Αυτός ο πόρος θεωρούνταν υποψήφιος για αποθήκευση στην ενδιάμεση μνήμη. Σε περίπτωση που ο πόρος αποθηκεύονταν στην ενδιάμεση μνήμη, ο χρόνος τελευταίας αλλαγής υποδεικνύει το διάστημα στο οποίο θα έπρεπε να επανελεγχθεί η ορθότητα (revalidate) του πόρου. Αντίστροφα, η λογική αυτή υποθέτει ότι αν ο πόρος έχει αλλαχθεί πρόσφατα είναι υψηλή η πιθανότητα αλλαγής του σε σύντομο χρονικό διάστημα. Ο πόρος αυτός δεν θα μείνει επίκαιρος (fresh) στην ενδιάμεση μνήμη για μεγάλο διάστημα. Ένας άλλος προβληματισμός που λαμβάνεται επίσης υπόψη για τη δυνατότητα αποθήκευσης σε ενδιάμεση μνήμη ενός πόρου είναι ότι οι πόροι που αλλάζουν συχνά είναι δημοφιλείς και, κατ' επέκταση, χαρακτηρίζονται από υψηλό ρυθμό προσπέλασης (rate of access). Αυτοί οι πόροι θα πρέπει να υποστούν αποθήκευσης.

1.4 Είδη ενδιάμεσων μνημών

Οι ενδιάμεσες μνήμες εντοπίζονται στους φυλλομετρητές και σε όλους τους ενδιάμεσους (intermediaries) μεταξύ του user agent και του εξυπηρετή πηγής. Συνήθως, η ενδιάμεση μνήμη εντοπίζεται σε ένα μεσάζοντα (proxy) ή στους φυλλομετρητές (browsers). Συχνά προτιμάται η αποθήκευση πόρων σε περισσότερες από μία τοποθεσίες (locations) γιατί:

- Η ενδιάμεση μνήμη του φυλλομετρητή μπορεί να αποφύγει να επαναμεταφέρει πόρους που ζήτησε ο χρήστης κατά την διάρκεια της ίδιας συνόδου. Ο

φυλλομετρητής όμως δεν εκμεταλλεύεται τους πόρους που ζητούνται ιδιαίτερα συχνά από άλλους χρήστες στο ίδιο περιβάλλον.

- Ένας μεσάζοντας με ενδιάμεση μνήμη (caching proxy) μπορεί να υποστηρίξει δεκάδες ίσως και εκατοντάδες χρήστες. Μία ενδιάμεση μνήμη φυλλομετρητή μπορεί να αποθηκεύσει ένα λογικό σύνολο από απαντήσεις που μεταδόθηκαν πρόσφατα για μεγαλύτερο διάστημα απ' ό,τι ένας μεσάζοντας με ενδιάμεση μνήμη. Ένας μεσάζοντας με ενδιάμεση μνήμη, όντας πόρος διαμοιραζόμενος από πολλούς, θα πρέπει να εκτοπίσει ορισμένες απαντήσεις από την ενδιάμεση μνήμη ταχύτερα από τον φυλλομετρητή.
- Μία περιφερειακή ενδιάμεση μνήμη μπορεί να υποστηρίξει πολλαπλές ενδιάμεσες μνήμες που χαρακτηρίζονται από μία γεωγραφική εγγύτητα και ανήκουν σε μία ή περισσότερες διαχειριστικές οντότητες. Μία εθνική ενδιάμεση μνήμη μπορεί να ομαδοποιήσει ένα σύνολο από περιφερειακές ενδιάμεσες μνήμες και να ελαττώσει το κόστος σε χώρες όπου η διεθνής κίνηση χαρακτηρίζεται από υψηλό κόστος.

Άλλες μορφές ενδιάμεσων μνημών είναι οι ανάστροφες (reverse) και οι ενδιάμεσες μνήμες αναχαίτισης (interception caches). Ένα ανάστροφο μεσάζων (reverse proxy) συμπεριφέρεται ως ένα front-end σε ένα ή περισσότερους εξυπηρετές πηγής και μπορεί να περιέχει και ενδιάμεση μνήμη. Σε ένα ανάστροφο μεσάζοντα, η ενδιάμεση αποθήκευση υλοποιείται για λογαριασμό των εξυπηρετών πηγής και όχι των αντιπροσώπων χρήστη. Στόχος του ανάστροφου μεσάζοντα είναι η μείωση του φόρτου στους εξυπηρετές πηγής. Οι πιο δημοφιλείς πόροι που ζητούνται από ένα εξυπηρετή πηγής είναι πολύ πιθανόν να εντοπίζονται στον ανάστροφο μεσάζων. Ένας ανάστροφος μεσάζων προωθεί την αίτηση στον εξυπηρετή πηγής εάν η απάντηση δεν εντοπιστεί στην ενδιάμεση μνήμη. Η προώθηση πραγματοποιείται μέσω ενός tunnel, δηλαδή ο μεσάζων εξυπηρετήτης συμπεριφέρεται ως ένας απλός μεταγωγέας (blind relay).

Οι μεσάζοντες εξυπηρετές αναχαίτισης (interception proxies) μπορεί να βρίσκονται οπουδήποτε σε ένα δίκτυο και να ελέγχουν τα επίπεδα δικτύου και μεταφοράς. Οι μεσάζοντες εξυπηρετές αναχαίτισης αναχαιτίζουν την HTTP αίτηση και λαμβάνουν ρητά την απάντηση. Συνήθως τοποθετούνται κοντά στους clients. Οι μεσάζοντες εξυπηρετές αναχαίτισης δεν χρειάζεται να βρίσκονται πάνω στην διαδρομή των πακέτων. Μία συσκευή που θα μπορούσε να ελέγξει τα πακέτα σε επίπεδο μεταφοράς θα μπορούσε να ανακατευθύνει την κίνηση σε ένα μεσάζοντα εξυπηρετή αναχαίτισης (interception

roxy). Μετά την αποθυλάκωση των πακέτων, η αίτηση θα μπορούσε προαιρετικά να ανακατευθυνθεί σε κοντινές ενδιάμεσες μνήμες κάτω από τον ίδιο χειριστικό έλεγχο όπως στον μεσάζοντα εξυπηρέτη αναχαίτισης. Αυτή η αναχαίτιση είναι γενικά διαφανής στον χρήστη, ο οποίος αντιλαμβάνεται μειωμένη καθυστέρηση, αν ο πόρος εντοπιστεί σε στην ενδιάμεση μνήμη.

1.5 Πώς υλοποιείται η ενδιάμεση αποθήκευση

Η υλοποίηση του μηχανισμού της ενδιάμεσης αποθήκευσης βασίζεται σε συγκεκριμένα βήματα: η ενδιάμεση μνήμη πρέπει, αρχικά, να αποφασίσει αν το μήνυμα επιδέχεται αποθήκευση σε ενδιάμεση μνήμη, μετά να διαγνώσει αν υπάρχει διαθέσιμος χώρος για την αποθήκευση του, και, στην περίπτωση που δεν υπάρχει χώρος, να αποφασίσει ποια από τα ήδη αποθηκευμένα στην ενδιάμεση μνήμη αντικείμενα θα αντικατασταθούν. Η ενδιάμεση μνήμη, με την λήψη κάποιας αίτησης θα πρέπει να αποφασίσει αν είναι σε θέση να την εξυπηρετήσει. Εάν είναι όντως σε θέση να διεκπεραιώσει την αίτηση, θα πρέπει να επιστρέψει τον πόρο στον αιτούμενο client και να ενημερώσει κάποια τοπική πληροφορία. Η ενδιάμεση μνήμη θα πρέπει να έχει μία πολιτική συνέπειας (συνάφειας) για να διατηρεί την κατάσταση επικαιρότητας (freshness information) του πόρου.

Διαφορετικές ενδιάμεσες μνήμες υλοποιούν διαφορετικές προσεγγίσεις για την απόφαση ενδιάμεσης αποθήκευσης του πόρου. Όπως έχει ήδη επισημανθεί, τα διαφορετικά κριτήρια για την απόφαση αυτή είναι:

- Υπάρχουν απαιτήσεις του πρωτοκόλλου που καθορίζουν ότι ο συγκεκριμένος πόρος δεν μπορεί να αποθηκευτεί στην ενδιάμεση μνήμη.
- Επιδέχεται συνήθως το περιεχόμενο αποθήκευση σε ενδιάμεση μνήμη;
- Είναι πιθανή η επαναχρησιμοποίηση του πόρου;
- Μπορεί η απόφαση για αποθήκευση της συγκεκριμένης απάντησης να οδηγήσει σε αντικατάσταση ενός ή περισσότερων πόρων;

Μία ενδιάμεση μνήμη χρησιμοποιεί κάποια ή όλα τα παραπάνω κριτήρια για να αποφασίσει αν πρέπει να εφαρμόσει caching σε κάποιο πόρο.

1.6 Διαγραφή και Αποθήκευση Πόρων στην Ενδιάμεση μνήμη

Μετά την απόφαση για αποθήκευση ενός πόρου, η ενδιάμεση μνήμη ελέγχει αν μπορεί να προχωρήσει στην αποθήκευση χωρίς να διαγράψει κάποια από τα αντικείμενα που

ήδη βρίσκονται στην ενδιάμεση μνήμη. Εάν όχι, ενεργοποιείται ο μηχανισμός αντικατάστασης (cache replacement). Ο μηχανισμός αυτός εισάγει κάποια επιβάρυνση, ειδικά αν μικρότερα αντικείμενα που είναι ήδη αποθηκευμένα στην ενδιάμεση μνήμη πρέπει να διαγραφούν. Πρόσθετη επιβάρυνση προκαλείται όταν λαμβάνονται, μελλοντικά, αιτήσεις για αντικείμενα τα οποία έχουν διαγραφεί. Σε αυτήν την περίπτωση πρέπει να εγκατασταθούν νέες συνδέσεις για την άντληση τους από τους εξυπηρέτες πηγής. Συχνά, πόροι οι οποίοι θεωρούνται ότι δεν είναι επίκαιροι (stale) διαγράφονται από την ενδιάμεση μνήμη ακόμη και στην περίπτωση που αυτή δεν είναι πλήρης. Έτσι περιορίζεται η ανάγκη για ενεργοποίηση του μηχανισμού αντικατάστασης (cache replacement) την στιγμή διεκπεραίωσης μίας αίτησης (μείωση της υστέρησης που εκλαμβάνεται ο χρήστης).

Μόλις προκύπτει ελεύθερος χώρος, η ενδιάμεση μνήμη εξάγει πληροφορία από το μήνυμα όπως το χρόνο τελευταίας μεταβολής καθώς και πληροφορία για την λήξη του πόρου. Επικεφαλίδες όπως οι `Expire` και `Cache-control: max-stale` μεταφέρουν πληροφορία σχετική με την λήξη (expiration) του πόρου. Αυτά τα πεδία συντελούν στο να είναι η ενδιάμεση μνήμη συμβατή με τους περιορισμούς του πρωτοκόλλου HTTP για το βάθος χρόνου στο οποίο η απάντηση μπορεί να επιστραφεί ως σημασιολογικά έγκυρη. Μία ενδιάμεση μνήμη η οποία είναι συμβατή με το πρωτόκολλο είναι υποχρεωμένη να εξασφαλίζει ότι οι απαντήσεις που επιστρέφει θεωρούνται από τον εξυπηρέτη πηγής ως επίκαιρες (fresh). Αν απουσιάζουν πληροφορίες λήξης του πόρου, η ενδιάμεση μνήμη προσδιορίζει ευρεστικά ένα χρόνο λήξεως για να αποφασίσει πότε καθίσταται μη-έγκυρος (stale). Ο αλγόριθμος μπορεί να βασίζεται στη τιμή του πεδίου `Last-Modified` που είναι συνυφασμένη με τον πόρο. Για παράδειγμα, μία ενδιάμεση μνήμη μπορεί να προσθέσει ένα καθορισμένο χρονικό διάστημα π.χ. 10' στην τιμή `Last-modified` και να χρησιμοποιήσει την νέα τιμή σαν διάστημα ελέγχου επικαιρότητας (freshness interval). Επίσης, παράγεται ένα κλειδί για χρήση σε μελλοντικές αναφορές (lookups). Το κλειδί αυτό είναι μία τιμή κατακερματισμού (hash value) η οποία βασίζεται στο URL του πόρου.

1.7 Επιστροφή αποθηκευμένου πόρου ή προώθηση αίτησης

Εάν ένας πόρος που αντιπροσωπεύει κάποιο κλειδί που αναζητείται στην ενδιάμεση μνήμη εντοπιστεί, θεωρείται ότι συνέβη μια επιτυχία ενδιάμεσης μνήμης (cache hit). Τότε, ανάλογα με την πολιτική της ενδιάμεσης μνήμης και ενδεχόμενους περιορισμούς που επιβάλλονται από πεδία επικεφαλίδας, ένας επανέλεγχος ορθότητας μπορεί να

εκτελεστεί για να διαπιστωθεί εάν ο πόρος είναι επίκαιρος. Εάν ο έλεγχος αυτός αποβεί θετικός, η αίτηση ικανοποιείται από την ενδιάμεση μνήμη. Διαφορετικά, η ενδιάμεση μνήμη ανακτά ένα νέο αντίγραφο του πόρου και εφαρμόζει την πολιτική της για να αποφανθεί αν ο πόρος θα πρέπει να αποθηκευτεί παράλληλα με την προώθηση του στον αιτούμενο πελάτη (client). Εάν ο πόρος δεν εντοπιστεί στην ενδιάμεση μνήμη, (περίπτωση αποτυχίας ενδιάμεσης μνήμης (cache miss)) η αίτηση προωθείται.

1.8 Συντήρηση ενδιάμεσης μνήμης

Περιοδικά, μία ενδιάμεση μνήμη μπορεί να ελέγξει εάν τα αντικείμενα που είναι αποθηκευμένα σε αυτή είναι επίκαιρα και να πυροδοτήσει την διαγραφή των «παλιών» αντικειμένων. Μία ενδιάμεση μνήμη μπορεί επίσης να ελέγξει τον ρυθμό αιτήσεων για αντικείμενα που είναι αποθηκευμένα σε αυτή για να αποφασίσει ποιοι πόροι είναι δημοφιλείς και να προβεί σε ειδικές ενέργειες για λογαριασμό τους. Για παράδειγμα, μία ενδιάμεση μνήμη μπορεί να προελέγχει την ορθότητα – εγκυρότητα (pre-validation) για να διαπιστώσει αν τα αντικείμενα που ζητούνται περισσότερο είναι επίκαιρα. Αυτός ο προέλεγχος μπορεί να υλοποιηθεί με την HTTP Head μέθοδο που αντλεί μόνο τα μεταδεδομένα για τον υπό συζήτηση πόρο. Μία ενδιάμεση μνήμη μπορεί και προδραστικά, να επικοινωνήσει με τον εξυπηρετή πηγής και να ελέγξει εάν ο πόρος έχει μεταβληθεί. Εάν έχει όντως μεταβληθεί μπορεί να εκκινήσει την διαδικασία της προανάκτησης αντικειμένων για να ενημερωθεί η ενδιάμεση μνήμη.

1.9 Στρατηγικές αντικατάστασης αντικειμένων ενδιάμεση μνήμη

Μόλις η ενδιάμεση μνήμη είναι πλήρης, αντικείμενα πρέπει να διαγραφούν για να δημιουργήσουν χώρο για την αποθήκευση νέων απαντήσεων. Πολλές στρατηγικές για την αντικατάσταση αντικειμένων έχουν προταθεί. Ορισμένες προέρχονται από το παραδοσιακό χώρο του διαχείρισης ενδιάμεσων μνημών (cache management) σε συστήματα αρχείων, ενώ άλλες είναι εξειδικευμένες στο Web περιβάλλον. Μία ιδιαίτερα γνωστή προσέγγιση είναι το Least Recently Used (LRU) – αντικατάσταση του αντικειμένου που χρησιμοποιήθηκε λιγότερο. Οι στόχοι της ενδιάμεσης αποθήκευσης, δηλαδή η μείωση του όγκου της πληροφορίας που ανταλλάσσεται στο δίκτυο καθώς και της υστέρησης που αντιλαμβάνεται ο χρήστης, οδηγούν σε σύνθετες αποφάσεις για την αντικατάσταση περιεχομένου ενδιάμεσης μνήμης. Οι σύνθετες αποφάσεις αποτελούν ένα συνδυασμό μετρικών που περιλαμβάνουν το μέγεθος των απαντήσεων που αποθηκεύονται, τον τύπο αντικειμένου ακόμη και την έννοια της απόστασης προς τον

εξυπηρέτη πηγής.

Η χρησιμότητα διατήρησης ενός πόρου στην ενδιάμεση μνήμη μπορεί να υπολογιστεί από πολλούς παράγοντες όπως:

- Το κόστος ανάκτησης του πόρου: το κόστος ανάκτησης ενός πόρου από ένα εξυπηρέτη πηγής προσδιορίζεται από την διασυνδεσιμότητα της ενδιάμεσης μνήμης και την απόσταση που πρέπει να διανύσει ο πόρος μέχρι να καταχωρηθεί στην ενδιάμεση μνήμη. Αντικαθιστώντας ένα πόρο του οποίου η ανάκτηση ήταν “ακριβή”, το ίδιο κόστος θα πρέπει να αντιμετωπιστεί στην περίπτωση που ο πόρος ζητηθεί πάλι στο μέλλον.
- Το κόστος αποθήκευσης του πόρου: Μία ενδιάμεση μνήμη έχει σταθερό μέγεθος και η αποθήκευση ενός αντικειμένου σημαίνει λιγότερο χώρο για άλλα αντικείμενα. Ένας μεγάλος σε όγκο πόρος καταλαμβάνει σημαντικό χώρο αλλά ενδεχόμενη αντικατάσταση του σημαίνει ότι η ανάκτηση του πάλι θα κοστίζει σημαντικά.
- Ο αριθμός των προσβάσεων στο πόρο κατά το παρελθόν: ένα αντικείμενο που έχει προσπελαστεί πολλές φορές στο παρελθόν είναι πολύ πιθανόν να προσπελαστεί και στο μέλλον και, κατά συνέπεια, είναι επωφελές να παραμείνει στην ενδιάμεση μνήμη για μεγαλύτερο διάστημα.
- Η πιθανότητα προσπέλασης του πόρου στο άμεσο μέλλον: Εάν ο πόρος είναι πιθανόν να ανακτηθεί στο άμεσο μέλλον δεν ενδείκνυται η απόρριψη του από την ενδιάμεση μνήμη. Η πιθανότητα πρόσβασης σε ένα πόρο θα μπορεί να είναι γνωστή *a priori* ή να προσδιορίζεται βάσει της ιστορικότητας προσπέλασης (access patterns).
- Ο χρόνος από την τελευταία μεταβολή του πόρου: Ένας πόρος που δεν έχει μεταβληθεί για μεγάλο διάστημα είναι λιγότερο πιθανό να αλλάξει στο κοντινό μέλλον. Ένας πόρος που παρήχθη πρόσφατα μπορεί να είναι δυναμικός ή να υπάρχει μεγάλη πιθανότητα να αλλάξει πάλι στο μέλλον. Οι πόροι που υπάρχει μεγάλη πιθανότητα να αλλάξουν είναι συνήθως δημοφιλείς. Αυτοί οι πόροι μπορούν να μεταβληθούν σαν αποτέλεσμα του δημοφιλοφούς τους χαρακτήρα και είναι έτσι καλοί υποψήφιοι για αποθήκευση σε ενδιάμεση μνήμη. Η αποθηκευμένη απάντηση μπορεί όμως να πρέπει να αντικατασταθεί συχνά με τον μεταβαλλόμενο πόρο. Ο χρόνος τελευταίας μεταβολής ενός πόρου μπορεί έτσι να χρησιμοποιηθεί για να προσδιοριστούν οι πιθανοί υποψήφιοι για αντικατάσταση.

- Ο χρόνος λήξης που προσδιορίζεται ευρεστικά: Εάν δεν υπάρχει χρόνος λήξης προσδιορισμένος από τον εξυπηρέτη, η ενδιάμεση μνήμη προσδιορίζει ευρεστικά ένα χρόνο λήξης. Εάν δεν υπάρχουν πόροι για τους οποίους έχει παρέλθει ο χρόνος λήξης, τότε αυτοί που βρίσκονται κοντινότερα στην λήξη τους αποτελούν υποψήφιους για αντικατάσταση.

Οι αλγόριθμοι που χρησιμοποιούνται ως επί το πλείστον για αντικατάσταση αντικειμένων σε μια ενδιάμεση μνήμη (cache replacement) είναι οι ακόλουθοι:

- Least Recently Used (LRU)
- Least Frequently Used (LFU)
- Size of object (SIZE)
- Hyper-G (LRU/LFU/SIZE): Το σύστημα Hyper-G συνδυάζει τις πολιτικές LRU/LFU και Size. Η πρώτη απόφαση για αντικατάσταση (replacement) βασίζεται στο LFU. Εάν υπάρχουν περισσότεροι από ένας πόροι που πληρούν το παραπάνω κριτήριο, εφαρμόζεται η πολιτική LRU. Εάν πάλι, δεν προσδιορίζεται ένας πόρος προς αντικατάσταση, επιλέγεται ο μεγαλύτερος σε όγκο πόρος.
- GreedyDual-Size: Ο αλγόριθμος αυτός είχε προταθεί για αντικατάσταση σελίδων στην μνήμη υπολογιστικών συστημάτων. Ο αρχικός προσανατολισμός του δηλαδή αφορούσε ένα σύστημα με σταθερό μέγεθος πόρων ενώ το κόστος ανάκτησης από την δευτερεύουσα αποθήκευση (συστήματα δίσκων) ήταν βασικότατος παράγοντας. Ο αλγόριθμος επεκτάθηκε για να καλύψει την περίπτωση την ποικιλότητα των μεγεθών των www πόρων. Ο μετασχηματισμένος αλγόριθμος συσχετίζει μία τιμή χρησιμότητας (utility value) και αντικαθιστά τον πόρο με την χαμηλότερη τιμή χρησιμότητας. Εκτός από το κόστος μεταφοράς του πόρου στην ενδιάμεση μνήμη και το μέγεθος του, η τιμή χρησιμότητας επηρεάζεται από τον παράγοντα παλαιώσης (age factor) που ενημερώνονται καθώς πόροι απομακρύνονται από την ενδιάμεση μνήμη.

1.10 Συνάφεια της ενδιάμεσης μνήμης

Ο εξυπηρέτης πηγής αποφασίζει την διάρκεια για την οποία ο πόρος θα πρέπει να θεωρείται έγκυρος (freshness duration). Μία ενδιάμεση μνήμη θα πρέπει να εξασφαλίσει ότι μία αποθηκευμένη απάντηση είναι ακόμη έγκυρη πριν να απαντήσει σε κάποιον πελάτη που αιτείται τον πόρο. Η συνάφεια της ενδιάμεσης μνήμης αποτελεί ένα

σημαντικό πρόβλημα. Πολλοί σχετικοί αλγόριθμοι έχουν προταθεί κατά τα τελευταία έτη για το πρόβλημα της συνάφειας των ενδιάμεσων μνημών. Η ανάγκη για συνάφεια ενδιάμεσης μνήμης εξαρτάται από τους πόρους και τις πολιτικές οι οποίες έχουν επιβληθεί στην ενδιάμεση μνήμη. Οι ενδιάμεσες μνήμες μπορεί απλά να επιστρέφουν ένα παλιό αποθηκευμένο πόρο μαζί με μία αιτία για την απαξίωση του πόρου. Μεταξύ των αιτιών είναι η αδυναμία εγκατάστασης σύνδεσης προς τον εξυπηρέτη πηγής ή ο υψηλός φόρτος της ενδιάμεσης μνήμης. Η επικεφαλίδα Warning του HTTP/1.1 μπορεί να χρησιμοποιηθεί για να υποδηλώσει ότι επιστρέφεται μία μη-επίκαιρη απάντηση.

Το πρωτόκολλο HTTP/1.1 παρέχει πολλούς τρόπους για την διατήρηση της συνάφειας των ενδιάμεσων μνημών. Εάν ο εξυπηρέτης πηγής θέσει ένα συγκεκριμένο χρόνο λήξης για ένα πόρο, ο μεσάζων εξυπηρέτης που παρέχει ενδιάμεση αποθήκευση ανεξάρτητα από την σημασιολογία του πόρου, είναι υποχρεωμένος να υιοθετήσει τον ίδιο χρόνο λήξης. Η μόνη διαφοροποίηση σε αυτό είναι ο περιορισμός που μπορεί να τεθεί στο αίτημα του πελάτη μέσω επικεφαλίδας Cache-control: only-if-cached που αναγκάζει τον μεσάζοντα εξυπηρέτη να επιστρέψει μία ήδη αποθηκευμένη απάντηση χωρίς να ελέγξει την ορθότητα της στον εξυπηρέτη πηγής. Εάν ο εξυπηρέτης πηγής δεν θέσει ένα χρόνο λήξης, ο μεσάζων εξυπηρέτης μπορεί να προσδιορίσει ευρεστικά ένα χρόνο λήξης. Ο πλέον συνήθης τρόπος ελέγχου της συνάφειας στο www είναι η αποστολή ενός GET ή HEAD αιτήματος με μία επικεφαλίδα if-modified-since. Η επικεφαλίδα μεταφέρει μία χρονοσφραγίδα που δεικνύει το χρόνο τελευταίας μεταβολής του πόρου όπως αυτός υποδεικνύεται από τον εξυπηρέτη πηγής. Σε ορισμένες περιπτώσεις, ο χρόνος παραγωγής της απάντησης μπορεί να είναι ο χρόνος τελευταίας μεταβολής του πόρου. Οι ετικέτες οντότητας (entity tags) του HTTP/1.1 σε συνδυασμό με την επικεφαλίδα if-modified-since μπορούν να χρησιμοποιηθούν για την πραγματοποίηση ελέγχων συνάφειας. Ο εξυπηρέτης πηγής μπορεί να απαντήσει με ένα πλήρες αντίγραφο του πόρου ή με την απάντηση 304 Not Modified (και χωρίς σώμα στην απάντηση). Παρόλα αυτά ένας έλεγχος συνάφειας προϋποθέτει πλήρη διαδοχή μηνυμάτων HTTP αίτησης/ απάντησης.

Εάν ο μεσάζοντας με ενδιάμεση μνήμη στέλνει ένα αίτημα ελέγχου εγκυρότητας κάθε φορά που συμβαίνει μια επιτυχία ενδιάμεσης μνήμης, η πολιτική καλείται ισχυρή συνάφεια (strong consistency). Εάν ο μεσάζων εξυπηρέτης χρησιμοποιεί ένα ευρεστικό αλγόριθμο για να αποφανθεί εάν ο πόρος είναι επίκαιρος, χωρίς να συμβουλευτεί τον εξυπηρέτη πηγής, η πολιτική καλείται ασθενής συνάφεια (weak consistency). Οι δύο

ευρεστικοί αλγόριθμοι για έλεγχο συνάφειας είναι οι: leased-based και time-to-live.

- Lease-based προσέγγιση: Η ενδιάμεση μνήμη συμφωνεί να αποθηκεύσει ένα πόρο για συγκεκριμένο χρονικό διάστημα (περίοδος χρονομίσθωσης – lease) χωρίς να ελέγχει την ορθότητα του (revalidation). Ο εξυπηρέτης «υπόσχεται» να ειδοποιήσει την ενδιάμεση μνήμη για ενδεχόμενες αλλαγές στον αποθηκευμένο πόρο κατά την διάρκεια της περιόδου χρονομίσθωσης. Εάν η περίοδος παρέλθει η ενδιάμεση μνήμη μπορεί να ελέγξει την ορθότητα του πόρου ή να ανανεώσει την χρονομίσθωση. Αυτή η προσέγγιση μεταφέρει το κόστος του revalidation στον εξυπηρέτη πηγής ο οποίος θα πρέπει να γνωρίζει και παρακολουθεί όλους τους proxies στους οποίους έχει υποσχεθεί ενημερώσεις. Η προσέγγιση δεν μπορεί να κλιμακωθεί αν ο εξυπηρέτης πηγής είναι υποχρεωμένος να ειδοποιήσει εκατοντάδες – χιλιάδες proxies.
- Time-to-live προσέγγιση: Οι πόροι έχουν συσχετιστεί με ένα χρόνο λήξης αποθήκευσης. Όταν παρέλθει αυτό το χρονικό διάστημα, οι πόροι παύουν να θεωρούνται επίκαιροι. Κατά την διάρκεια της περιόδου time-to-live (TTL), η ενδιάμεση μνήμη δεν επικυρώνει τις απαντήσεις διασώζοντας εύρος ζώνης. Η απόδοση τιμής στον χρόνο TTL μπορεί να επηρεαστεί ένα πλήθος παραμέτρων όπως ο χρόνος λήξης που αναφέρεται στην επικεφαλίδα της απάντησης, η συχνότητα αναφοράς στην απάντηση, ο χρόνος τελευταίας αλλαγής του πόρου.

1.11 Επικοινωνία μεταξύ ενδιάμεσων μνημών

Ανάλογα με το πως οργανώνονται οι ενδιάμεσες μνήμες, μπορούν να δέχονται και να λαμβάνουν πληροφορίες για τους πόρους για τους οποίους ενδιαφέρονται. Αυτή η επικοινωνία είναι εξωτερική, ανεξάρτητη από τα μηνύματα αίτησης/ απάντησης που ρέουν μεταξύ πελατών και εξυπηρετών πηγής. Η επικοινωνία μεταξύ ενδιάμεσων μνημών μπορεί να βασίζεται στο HTTP αλλά συνήθως χρησιμοποιεί εξειδικευμένα, light-weight πρωτόκολλα. Εάν ένα σύνολο από ενδιάμεσες μνήμες οργανώνεται σε ιεραρχία, μία ενδιάμεση μνήμη μπορεί να επικοινωνήσει με τις υπόλοιπες ενδιάμεσες μνήμες στο ίδιο επίπεδο να διαπιστώσει εάν ο ζητούμενος πόρος είναι διαθέσιμος σε αυτές. Ένα ερώτημα για κάποιο πόρο μπορεί να απαντηθεί από μία ή περισσότερες ενδιάμεσες μνήμες που τυχαίνει να έχουν τον πόρο. Συχνά, η ανάκτηση ενός πόρου από μία τοπική ενδιάμεση μνήμη είναι προτιμότερη από την ανάκτηση από τον εξυπηρέτη πηγής. Η αναμονή για την απάντηση από όλες τις ενδιάμεσες μνήμες στην ιεραρχία μπορεί να αυξήσει σημαντικά την υστέρηση στον χρήστη. Για την υποβοήθηση

της επικοινωνίας μεταξύ των ενδιάμεσων μνημών εξετάζονται τέσσερα πρωτόκολλα: τα Internet Cache Protocol (ICP), Cache Array Resolution Protocol (CARP), Cache Digest Protocol (CDP) και Web Cache Coordination Protocol (WCCP).

1.11.1 Internet Cache Protocol (ICP)

Μία ενδιάμεση μνήμη που δεν διαθέτει τον ζητούμενο πόρο μπορεί να θέλει να ελέγξει την διαθεσιμότητα του πόρου σε μία άλλη γειτονική ενδιάμεση μνήμη. Αυτή η επικοινωνία είναι διαφορετική από την παραδοσιακή αίτηση για ένα πόρο από τον εξυπηρέτη πηγής. Σε αυτή την περίπτωση οι ενδιάμεσες μνήμες αποτελούν την πηγή καθώς και τον προορισμό των ανταλλασσόμενων μηνυμάτων. Ένα διαφορετικό πρωτόκολλο απαιτείται για την επικοινωνία μεταξύ των ενδιάμεσων μνημών. Ένα από τα πρώτα πρωτόκολλα που καθιερώθηκαν σε αυτήν την επικοινωνία είναι το Internet Cache Protocol (ICP). Το ICP είναι ένα πρωτόκολλο ερωταποκρίσεων. Το μήνυμα που στέλνεται από μία ενδιάμεση μνήμη πελάτη είναι μία ερώτηση για το αν ο ομότιμος κόμβος έχει ένα αντίγραφο από τον πόρο που χρειάζεται η συγκεκριμένη ενδιάμεση μνήμη. Ο δημοφιλής χαρακτήρας του ICP οφείλεται στο γεγονός ότι χρησιμοποιείται από το Squid.

Το ICP χρησιμοποιείται σε ιεραρχίες από ενδιάμεσες μνήμες, σύνολα ενδιάμεσων μνημών που συνδέονται μεταξύ τους κάτω από ένα κοινό γονέα. Η διαδικασία αυτή επαναλαμβάνεται, και η μετακίνηση προς τα ανώτερα επίπεδα της ιεραρχίας σημαίνει μετακίνηση προς μία περισσότερο κεντρική ενδιάμεση μνήμη. Οι κεντρικές ενδιάμεσες μνήμες μπορεί να έχουν μια περιφερειακή (regional) ενδιάμεση μνήμη ως άμεσο πρόγονο ενώ οι περιφερειακές ενδιάμεσες μνήμες μπορεί να έχουν μία εθνική ενδιάμεση μνήμη ως πρόγονο. Αν υποθεθεί ότι η ενδιάμεση μνήμη Original Cache δεν έχει κάποιο ζητούμενο πόρο, θα σταλούν ICP αιτήσεις (πάνω από UDP) σε όλους τους ομότιμους κόμβους ταυτόχρονα. Εάν κάποιος από τους ομότιμους κόμβους διαθέτει τον αιτούμενο πόρο, η Original Cache θα ενημερωθεί σχετικά και θα ζητήσει την ανάκτηση του χρησιμοποιώντας HTTP. Εάν κανείς από τους ομότιμους κόμβους δεν διαθέτει τον πόρο, η Original Cache θα προωθήσει την αίτηση στον γονέα της. Ο γονέας της Original Cache επαναλαμβάνει την διαδικασία. Εάν καμία από τις ενδιάμεσες μνήμες δεν διαθέτει τον πόρο, η Original Cache θα πρέπει να προωθήσει την αίτηση στον εξυπηρέτη πηγής. Η φιλοσοφία βάσει της οποίας λειτουργεί το ICP είναι ότι η αποστολή των ICP queries, ακόμη και αν αυτή επαναληφθεί πολλές φορές σε διάφορα επίπεδα της ιεραρχίας, είναι σημαντικά γρηγορότερη την επικοινωνία με τον εξυπηρέτη πηγής.

1.11.2 Cache Array Resolution Protocol (CARP)

Το CARP καθορίζει ένα μηχανισμό μέσω του οποίου ένα σύνολο από μεσάζοντες με ενδιάμεσες μνήμες (caching proxies) μπορούν να λειτουργήσουν ως μία λογικά ενιαία ενδιάμεση μνήμη. Ο μηχανισμός χειρίζεται το σύνολο των απαντήσεων που αποθηκεύονται σε ενδιάμεση μνήμη συλλογικά μεταξύ της ομάδα (array) των μεσαζόντων εξυπηρετών ως μία μεγάλη ενδιάμεση μνήμη. Μία συνάρτηση κατακερματισμού του κλειδιού (hash function) χρησιμοποιείται για να διαιρεθεί το σύνολο των URL μεταξύ των ενδιάμεσων μνημών. Ένας πελάτης που προσπαθεί να εντοπίσει ένα αποθηκευμένο (cached) πόρο μπορεί να κατευθύνει την αίτηση στην κατάλληλη ενδιάμεση μνήμη εφαρμόζοντας την συνάρτηση κατακερματισμού. Η συνάρτηση κατακερματισμού χρησιμοποιεί το αιτούμενο URL καθώς και την ταυτότητα των μεσαζόντων εξυπηρετών για να διαμορφώσει ένα μονοπάτι επίλυσης (resolution path). Εάν συγκριθεί με το ICP, το CARP έχει ντετερμινιστικό μονοπάτι επίλυσης της αίτησης, εξαλείφοντας έτσι την ανάγκη για μηνύματα ερωταποκρίσεων (queries). Επίσης, υπάρχουν λιγότερα διπλότυπα αποθηκευμένων πόρων στο CARP από το ICP. Το CARP χρησιμοποιεί το HTTP καθώς και απομακρυσμένες κλήσεις διεργασιών (Remote Procedure Calls) για την επικοινωνία μεταξύ των μεσαζόντων εξυπηρετών. Ο κάθε μεσάζων εξυπηρετής συσχετίζεται με ένα παράγοντα φορτίου (load factor) που λαμβάνεται υπόψη πριν μία αίτηση οδηγηθεί σε ένα συγκεκριμένο μεσάζοντα εξυπηρετή. Το CARP διατίθεται ως προϊόν από την Microsoft.

1.11.3 Cache Digest Protocol (CDP)

Το Cache Digest Protocol αποτελεί μία επέκταση του ICP. Η βασική ιδέα στο Cache Digest είναι η δυνατότητα ανταλλαγής μίας περίληψης (digest) των περιεχομένων της ενδιάμεσης μνήμης. Η περίληψη αποτελεί μία ένδειξη της συλλογής των αντικειμένων σε μία ενδιάμεση μνήμη. Όταν μία ενδιάμεση μνήμη έχει στην διάθεση της μία περίληψη από όλους τους ομότιμους κόμβους μπορεί, πολύ εύκολα να ανατρέξει στην περίληψη και να εξετάσει εάν το αιτούμενο αντικείμενο είναι διαθέσιμο σε μία από τις ενδιάμεσες μνήμες. Εάν η διερεύνηση αυτή επιτύχει, ο συγκεκριμένος ομότιμος κόμβος είναι υποψήφιος να δεχθεί μία αίτηση ανάκτησης του πόρου. Εάν ο έλεγχος στις περιλήψεις αποτύχει, οι αντίστοιχες ενδιάμεσες μνήμες δεν ερωτώνται με προφανές αποτέλεσμα την δραστική μείωση των ερωτημάτων που απευθύνονται στο σύνολο των ομότιμων κόμβων.

Ένα πρόβλημα του μηχανισμού Cache Digest είναι η εγκυρότητα των περιλήψεων και

τα λανθασμένα ερωτήματα τα οποία οφείλονται σε αυτήν. Ένα αντικείμενο μπορεί να αφαιρεθεί από μία ενδιάμεση μνήμη μετά την διαμόρφωση της σχετικής περίληψης. Ένα ακόμη πρόβλημα είναι το μέγεθος των περιλήψεων και η ανταλλαγή τους μεταξύ των ομότιμων κόμβων. Οι περιλήψεις ανταλλάσσονται μέσα σε HTTP μηνύματα, πάνω από το TCP, για λόγους αξιοπιστίας. Μία περίληψη μπορεί να θεωρηθεί σαν ένας κανονικός πόρος και οι τεχνικές ελέγχου της ορθότητας του HTTP (resource revalidation) μπορούν να χρησιμοποιηθούν για την διερεύνηση του επίκαιρου της περίληψης.

1.11.4 Web Cache Coordination Protocol (WCCP)

Το WCCP είναι ένας μηχανισμός συντονισμού, στενά δεμένος με το επίπεδο δικτύου. Ο σκοπός του WCCP είναι η αναχαίτιση (intercept) της HTTP αίτησης και η ανακατεύθυνση της στην ενδιάμεση μνήμη. Επειδή η αίτηση θα αποτύχει εάν η ενδιάμεση μνήμη δεν είναι, για κάποιο λόγο, διαθέσιμη, ένας μηχανισμός συντονισμού απαιτείται. Το αντικείμενο του μηχανισμού συντονισμού είναι να εξισορροπεί τον φόρτο μεταξύ διαφορετικών ενδιάμεσων μνημών, έχοντας πλήρη γνώση της διαθεσιμότητας τους. Ελέγχοντας περιοδικά την διαθεσιμότητα μίας ενδιάμεσης μνήμης, ο μηχανισμός εξασφαλίζει ότι δεν πρόκειται να προωθηθούν πακέτα σε μία ενδιάμεση μνήμη που δεν ανταποκρίνεται σε έλεγχο διαθεσιμότητας (heartbeat check). Ένας τέτοιος μηχανισμός αποτελεί την βάση του πρωτοκόλλου WCCP που υλοποιείται σαν τμήμα της Cisco Cache Engine. Η ενδιάμεση μνήμη ρυθμίζεται ώστε να δέχεται WWW αιτήσεις που ανακατευθύνονται σε αυτήν από ένα δρομολογητή. Ο δρομολογητής, που έχει ενεργοποιημένο το WCCP, μπορεί να επεξεργάζεται όλες τις IP επικεφαλίδες. Ένα TCP πακέτο που στοχεύει στην θύρα 80 ανακατευθύνεται στην ενδιάμεση μνήμη. Επιπλέον, οι δρομολογητές που διαθέτουν WCCP επικοινωνούν περιοδικά με τις μηχανές ενδιάμεσης αποθήκευσης για να εξασφαλίσουν την διαθεσιμότητα τους.

1.12 Διανομή Περιεχομένου (Content Distribution)

Η τεχνική του content distribution αναφέρεται στην εφαρμογή επιλεκτικού κατοπτρισμού (selective mirroring). Η βασική ιδέα στην διανομή περιεχομένου είναι να μειωθεί ο φόρτος στον εξυπηρετή πηγής. Αυτό μπορεί να επιτευχθεί παρέχοντας τμήμα ή όλο το περιεχόμενο από ένα σύνολο αντιγράφων (replicas). Διάφορες τεχνικές χρησιμοποιούνται για την ανακατεύθυνση των αιτήσεων στα αντίγραφα (π.χ., τεχνικές που βασίζονται στο DNS). Ένας τρόπος για την διαίρεση (partitioning) ενός πόρου είναι στα συστατικά βάσης (base) και εμφωλευμένων στοιχείων (embedded). Ένα έγγραφο

βάσης (base document) αποτελεί το container έγγραφο και τα εμφωλευμένα συστατικά είναι οι εικόνες ή τα scripts τα οποία αποτελούν τμήμα της WWW σελίδας. Οι εξυπηρέτες που χρησιμοποιούνται για να εξυπηρετήσουν το non-container τμήμα του πόρου καλούνται εξυπηρέτες διανομής περιεχομένου (content distribution servers). Μπορεί να εντοπίζονται κοντά στον εξυπηρέτη πηγής ή οπουδήποτε στο διαδίκτυο. Τα εμφωλευμένα στοιχεία υπάρχουν ως αντίγραφα (replicated) σε αυτούς τους εξυπηρέτες. Κατά την αίτηση, η υπηρεσία διανομής περιεχομένου (content distribution service) προσπαθεί να εντοπίσει τον εξυπηρέτη διανομής περιεχομένου που βρίσκεται «πλησιέστερα» στον χρήστη για να του επιστρέψει τα εμφωλευμένα στοιχεία. Η εγγύτητα ενός εξυπηρέτη διανομής περιεχομένου μπορεί να αναφέρεται σε γεωγραφική απόσταση, σε δικτυακή απόσταση και μετρικές υστέρησης (latency metrics). Αυτή η προσέγγιση ελαττώνει τον φόρτο στον εξυπηρέτη πηγής και βελτιώνει τον χρόνο απόκρισης για τον τελικό χρήστη.

Ο στόχος του διανομής περιεχομένου δεν είναι διαφορετικός από αυτόν της ενδιάμεσης αποθήκευσης. Και οι δύο προσεγγίσεις μετακινούν το περιεχόμενο κοντά στον τελικό χρήστη φιλοδοξώντας να μειώσουν την υστέρηση που αντιλαμβάνεται ο χρήστης καθώς και τον φόρτο στον εξυπηρέτη πηγής. Με το ενδιάμεση αποθήκευση, οι μεσάζοντες εξυπηρέτες πρέπει να διατηρούν την συνέπεια (consistency) και να επαληθεύουν την ορθότητα των αποθηκευμένων πόρων. Με το μηχανισμό διανομής περιεχομένου, οι εξυπηρέτες περιεχομένου έχουν πλήρη έλεγχο επί του περιεχομένου και μπορούν να προβούν σε ρυθμίσεις με τους εξυπηρέτες που διαθέτουν περιεχόμενο για λογαριασμό τους.

Στο μηχανισμό κατοπτρισμού ενδιάμεσης μνήμης (cache mirroring), μεγάλα τμήματα ενός δικτυακού τόπου (site) κατοπτρίζονται (mirrored) σε διάφορους κόμβους του διαδικτύου. Στην προσέγγιση διανομής περιεχομένου οι εξυπηρέτες πηγής αποφασίζουν ποιοι πόροι μπορούν να αντιγραφούν, και μεταφέρουν το έργο του κατοπτρισμού σε άλλο οργανισμό. Οι εξυπηρέτες πηγής είναι υποχρεωμένοι να ειδοποιούν την εταιρία που αναλαμβάνει το διανομή περιεχομένου όταν οι πόροι τους υφίστανται μεταβολές.

Ένα παράδειγμα εταιρίας που αναλαμβάνει διανομή περιεχομένου είναι η Akamai. Ένας δικτυακός τόπος που θέλει να διανεμηθούν τα περιεχόμενα του μέσω του Akamai, μετονομάζει αυτά τα URLs με συγκεκριμένο πρόθεμα. Το πρόθεμα περιέχει το όνομα ενός κόμβου (hostname string). Η DNS επίλυση του ονόματος κόμβου επιστρέφει την IP

διεύθυνση ενός Akamai εξυπηρέτη αντικατοπτρισμού (mirror server) που είναι πολύ πιθανό να περιέχει αντίγραφο του πόρου. Η απόφαση για την επιστροφή μίας IP διεύθυνσεως λαμβάνεται από τον DNS εξυπηρέτη του Akamai δικτύου. Σχεδιαστικά, ο προσδιοριζόμενος Akamai εξυπηρέτης είναι πλησιέστερα στον τοπικό DNS εξυπηρέτη του πελάτη που αιτήθηκε τον πόρο. Η προσδοκία είναι ότι ο πελάτης είναι αρκετά κοντά στον DNS εξυπηρέτη (δικτυακή απόφαση) και ο πόρος θα πρέπει να μεταφερθεί για μία μικρή σχετικά απόσταση. Επειδή πρέπει να απεικονιστεί το όνομα κόμβου που περιέχεται στο URL, είναι δυνατό για το Akamai να χρησιμοποιήσει τον DNS εξυπηρέτη για να προσδιορίσει τον κατάλληλο Akamai εξυπηρέτη που διαθέτει τον ζητούμενο πόρο.

Η τεχνική του Akamai πρέπει να εξασφαλίσει ότι το DNS lookup επιστρέφει το πλησιέστερο mirror site. Οι DNS τιμές TTL πρέπει να ρυθμιστούν κατάλληλα ώστε να αποφευχθεί να παραμένουν αποθηκευμένες σε ενδιάμεση μνήμη οι DNS απαντήσεις για μεγάλο διάστημα. Διαφορετικά, ένας πελάτης που αναζητάει κάποιον κόμβο μπορεί να χρησιμοποιήσει μία παλιά IP διεύθυνση ενός Akamai εξυπηρέτη που δεν αποτελεί, πλέον, την καταλληλότερη επιλογή για τον ζητούμενο πόρο. Υπάρχει ένας συμβιβασμός (trade-off) μεταξύ του εντοπισμού της καλύτερης επιλογής και του κόστους των DNS ερωτημάτων.

Υπάρχουν όμως ορισμένα προβλήματα με την προσέγγιση διανομής περιεχομένου. Ο εξυπηρέτης πηγής ωφελείται από τον περιορισμένο φόρτο ενώ οι τελικοί χρήστες ωφελούνται από την άντληση πόρων από «κοντινούς» χρήστες. Η θέση των εξυπηρετών διανομής περιεχομένου μπορεί να είναι ένα πρόβλημα για τους πελάτες. Είναι δυνατόν ορισμένοι πελάτες να έχουν καλύτερη δικτυακή συνδεσιμότητα (χαμηλότερα RTT) προς τον εξυπηρέτη πηγής από τους εξυπηρέτες διανομής περιεχομένου. Τεχνικά, οι εξυπηρέτες διανομής περιεχομένου εργάζονται για λογαριασμό των εξυπηρετών πηγής. Οι πελάτες εγκαθιστούν απευθείας συνδέσεις με τους εξυπηρέτες διανομής περιεχομένου και αναμένουν να είναι αυτοί συμβατοί με το HTTP πρωτόκολλο. Οι δικτυακοί τόποι διανομής περιεχομένου χρησιμοποιούν διαφορετικά πρωτόκολλα στην επικοινωνία τους με τους εξυπηρέτες πηγής και μπορούν να χρησιμοποιούν και άλλους μηχανισμούς για να διασφαλίσουν ότι διαθέτουν επικαιροποιημένο περιεχόμενο.

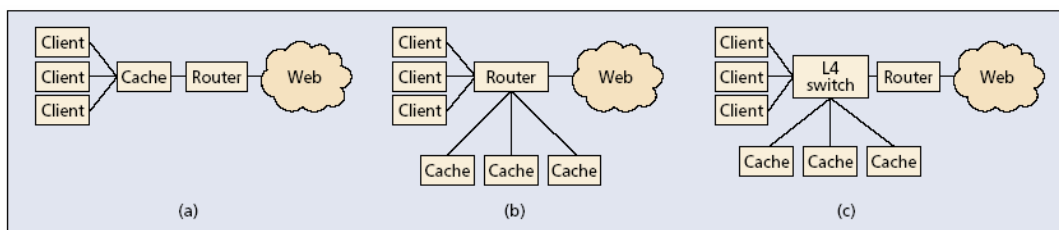
1.13 Αρχιτεκτονικές ενδιάμεσων μνημών

1.13.1 Μεσάζων εξυπηρέτης με ενδιάμεση μνήμη

Ο μεσάζων εξυπηρέτης με ενδιάμεση μνήμη (proxy cache server) παρεμβάλλεται ανάμεσα στις HTTP αιτήσεις των πελατών (clients), και αν βρει το αντικείμενο που ζητείται στην ενδιάμεση μνήμη του τότε το επιστρέφει στον χρήστη. Αν το αντικείμενο δεν βρεθεί, η ενδιάμεση μνήμη απευθύνεται στον πηγαίο εξυπηρέτη του αντικειμένου και για λογαριασμό του χρήστη, παίρνει το αντικείμενο, πιθανόν το αποθηκεύει, και τελικώς επιστρέφει το αντικείμενο στον χρήστη.

Οι ενδιάμεσες μνήμες μεσαζόντων (proxy caches) συνήθως αναπτύσσονται στις άκρες του δικτύου (π.χ. στις εταιρικές πύλες, ή στα firewall των hosts) έτσι ώστε να μπορούν να εξυπηρετούν ένα μεγάλο αριθμό από εσωτερικούς χρήστες. Η χρήση ενδιάμεσων μνημών μεσαζόντων τυπικά επιδρά θετικά στην μείωση του απαιτούμενου εύρους ζώνης, στην βελτίωση του χρόνου απόκρισης και στην αύξηση της διαθεσιμότητας στατικών πόρων του παγκόσμιου ιστού.

Μια αυτόνομη (standalone) διαμόρφωση ενός μεσάζοντα εξυπηρέτη απεικονίζεται στην Εικόνα 1. Σημειώστε ότι ένα μειονέκτημα αυτής της σχεδίασης είναι ότι η ενδιάμεση μνήμη αποτελεί ένα μοναδικό σημείο αστοχίας στο δίκτυο. Συνεπώς, όταν η ενδιάμεση μνήμη δεν είναι διαθέσιμη, το δίκτυο επίσης εμφανίζεται μη διαθέσιμο στους χρήστες. Επίσης, αυτή η προσέγγιση απαιτεί ότι όλοι οι φυλλομετρητές των χρηστών να διαμορφωθούν κατάλληλα για την χρήση της κατάλληλης ενδιάμεσης μνήμης μεσάζοντα. Σε συνέχεια, αν ο μεσάζων εξυπηρέτης δεν είναι διαθέσιμος, όλοι οι χρήστες πρέπει να αναμορφώσουν τους φυλλομετρητές με στόχο να χρησιμοποιήσουν ένα διαφορετική ενδιάμεση μνήμη.



Εικόνα 1: a) αυτόνομο, b) διάφανο στον δρομολογητή, και c) διάφανο στον μεταγωγέα proxy caching.

1.13.2 Ανάστροφος μεσάζων εξυπηρέτης με ενδιάμεση μνήμη

Μία ενδιαφέρουσα προσέγγιση είναι η έννοια της αποθήκευσης αντικειμένων σε ενδιάμεση μνήμη ανάστροφου μεσάζοντα (reverse proxy caching), στην οποία οι

ενδιάμεσες μνήμες τοποθετούνται κοντά στην πηγή του περιεχομένου και όχι κοντά στον πελάτη. Αυτή είναι μια ελκυστική λύση για εξυπηρέτες που περιμένουν αρκετές αιτήσεις και θέλουν να σιγουρέψουν μια υψηλού επιπέδου ποιότητα υπηρεσίας. Το ανάστροφο (Reverse) Proxy caching είναι επίσης ένας χρήσιμος μηχανισμός όταν υποστηρίζονται Web hosting farms (εικονικά domains που αντιστοιχούν σε ένα και μόνο φυσικό site), που αποτελεί μια αυξανόμενη κοινή υπηρεσία σε πολλούς ISPs.

Σημειώστε ότι αυτό το ανάστροφο Proxy caching είναι ολοκληρωτικά ανεξάρτητο από το Proxy caching στην περιοχή του πελάτη. Ουσιαστικά, αυτά μπορούν να συνυπάρχουν και επιλεκτικά να βελτιώνουν ολόκληρη την απόδοση.

1.13.3 Διαφανής ενδιάμεση αποθήκευση

Η διαφανής ενδιάμεση αποθήκευση (transparent caching) εξαφανίζει ένα από τα μεγαλύτερα μειονεκτήματα της αρχιτεκτονικής μεσαζόντων εξυπηρετών που είναι η απαίτηση για κατάλληλη διαμόρφωση των φυλλομετρητών. Οι διάφανες ενδιάμεσες μνήμες λειτουργούν ανακόπτοντας HTTP αιτήσεις και τις ανακατευθύνουν σε εξυπηρέτες με ενδιάμεση μνήμη (Web cache servers) ή συστοιχίες ενδιάμεσων μνημών (clusters). Αυτή η μέθοδος ενδιάμεσης αποθήκευσης στοιχειοθετεί ένα σημείο στο οποίο διαφορετικά σημεία διαχειριστικού ελέγχου είναι πιθανά, για παράδειγμα, στην απόφαση για το πως θα εξισορροπηθεί το φορτίο των αιτήσεων ανάμεσα σε πολλαπλές ενδιάμεσες μνήμες.

Το δυνατό αυτό σημείο του διαφανούς ενδιάμεσης αποθήκευσης είναι επίσης η κύρια του αδυναμία: η τεχνική αυτή αθετεί την από άκρη σε άκρη συμφωνία με την μη υποστήριξη σταθερών τερματικών σημείων της σύνδεσης. Αυτό είναι το πρόβλημα όταν μια εφαρμογή απαιτεί η διατήρηση πληροφοριών κατάστασης κατά την διάρκεια συνεχόμενων αιτήσεων ή κατά την διάρκεια μιας λογικής αίτησης που περιέχει πολλαπλά αντικείμενα.

Το φιλτράρισμα των HTTP αιτήσεων από όλη την εξερχόμενη Internet κίνηση προσθέτει επιπλέον καθυστέρηση. Για παράδειγμα, ενδιάμεσες μνήμες οι οποίες είναι τοποθετημένες σε συνδυασμό με τους μεταγωγείς επιπέδου 4 (L4) βασίζονται στο γεγονός ότι αυτοί οι μεταγωγείς ανακόπτουν την TCP κίνηση στη θύρα 80 και στέλνουν όλη την άλλη κίνηση κατευθείαν στον WAN δρομολογητή.

Υπάρχουν δύο τρόποι για την ανάπτυξη διαφανούς ενδιάμεσης αποθήκευσης: στο επίπεδο μεταγωγής και στο επίπεδο δρομολόγησης. Διαφανής ενδιάμεση αποθήκευση

βασισμένο στη δρομολόγηση χρησιμοποιεί δρομολόγηση βασισμένη στην πολιτική (policy-based) ώστε να κατευθύνει τις αιτήσεις στις κατάλληλες ενδιάμεσες μνήμες. Για παράδειγμα, αιτήσεις από συγκεκριμένους πελάτες μπορούν να συσχετίζονται με ένα συγκεκριμένο αποθεματικό.

Στη διαφανή ενδιάμεση αποθήκευση βασισμένη στην μεταγωγή, ο μεταγωγέας ενεργεί ως αποκλειστικός ισορροπιστής φορτίου. Αυτή η προσέγγιση είναι εναλλακτική επειδή μειώνει την απότομη αύξηση του φορτίου κίνησης που μπορεί να υπάρξει στην περίπτωση της βασισμένης σε πολιτική (policy-based) δρομολόγησης. Αν και αυτό προσθέτει επιπλέον κόστος στην φάση της ανάπτυξης, οι μεταγωγείς είναι γενικότερα πιο φθινοί από τους δρομολογητές.

Χρησιμοποιώντας μεταγωγείς επιπέδου 4 για διαφανή ενδιάμεση αποθήκευση είναι ένα παράδειγμα που δείχνει πως άλλα δικτυακά συστατικά στοιχίζουν ένα ρόλο στην αποτελεσματικότητα της ενδιάμεσης αποθήκευσης. Για τη διαφανή ενδιάμεση αποθήκευση, αυτοί οι μεταγωγείς εξασφαλίζουν μια μορφή τοπικής ισορρόπησης του φορτίου. Υπάρχουν άλλοι μεταγωγείς και λύσεις για τοπική ισορρόπηση του φορτίου καθώς και λύσεις για συνολική ισορρόπηση του φορτίου.

ΚΕΦΑΛΑΙΟ: 2

ΠΡΟΑΝΑΚΤΗΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

2.1 Γενική περιγραφή

Μετά την εγκατάσταση μίας TCP σύνδεσης, ένας πελάτης εκπέμπει μία HTTP αίτηση προς τον WWW εξυπηρέτη. Ακόμη και στην περίπτωση που ο πελάτης έχει ένα αποθηκευμένο (cached) αντίγραφο του ζητούμενου πόρου, η σύνδεση με τον εξυπηρέτη πρέπει να εγκατασταθεί για να πραγματοποιηθεί ο έλεγχος της ορθότητας (revalidation) του πόρου. Η καθυστέρηση στην λήψη της HTTP απάντησης εξαρτάται από πολλούς παράγοντες όπως π.χ., ο χρόνος παραγωγής της απάντησης στον εξυπηρέτη, το μέγεθος της απάντησης καθώς και το εύρος ζώνης από τον εξυπηρέτη στον πελάτη. Ο πελάτης μπορεί να αποκρύψει την καθυστέρηση από τον χρήστη παράγοντας εκ των προτέρων μία HTTP αίτηση και αποθηκεύοντας την απάντηση στην ενδιάμεση μνήμη. Εάν ο χρήστης αιτηθεί τον πόρο, ο πελάτης παρέχει την απάντηση άμεσα από την ενδιάμεση μνήμη. Με την υπόθεση ότι το περιεχόμενο του πόρου είναι επίκαιρο, ο πελάτης μπορεί να εξυπηρετήσει την αίτηση χωρίς να επιλύσει την IP

Άγγελος Γ. Κατσής

διεύθυνση του εξυπηρέτη, να εγκαταστήσει την σύνδεση και να αποστείλει την HTTP αίτηση. Η απάντηση θα είναι άμεση. Επίσης, η προανάκτηση αντικειμένων μπορεί να εκμεταλλευτεί αποδοτικότερα τις TCP συνδέσεις απ' ό,τι οι συνήθεις μεταφορές WWW πόρων που ενεργοποιούνται ως αποτέλεσμα των ενεργειών του χρήστη. Ο πελάτης μπορεί να διασωληνώσει (pipeline) ένα σύνολο αιτήσεων σε μία, μόνιμη (persistent) σύνδεση. Οι συνεχείς (back-to-back) μεταφορές αυτές των απαντήσεων μπορούν να συντελέσουν στην αποφυγή της φάσης αργής έναρξης (slow-start) του ελέγχου συμφόρησης.

Η προανάκτηση αντικειμένων των HTTP απαντήσεων έχει αποτελέσει αντικείμενο πολλών ερευνητικών προσπαθειών από το 1995 έως σήμερα. Παρά την δυνητική μείωση της υστέρησης που εκλαμβάνεται ο χρήστης, η προ-ανάκτηση WWW πόρων μπορεί να επιφέρει ένα σημαντικότατο φόρτο στον WWW εξυπηρέτη και το δίκτυο. Η προανάκτηση αντικειμένων δεν είναι χρήσιμο αν οι πόροι δεν ζητούνται από τον χρήστη και το περιεχόμενό τους είναι επίκαιρο. Επίσης, η προ-ανάκτηση απαντήσεων έρχεται σε ανταγωνισμό με τις εξελισσόμενες μεταφορές πόρων για το εύρος ζώνης του δικτύου.

Αντί της προανάκτησης του ιδίου του πόρου, ο πελάτης μπορεί να προανακτήσει την μεταπληροφορία για τον πόρο. Για παράδειγμα, ο πελάτης μπορεί να στείλει ένα αίτημα HEAD προς τον εξυπηρέτη. Η HTTP απάντηση θα πρέπει να έχει τις ίδιες επικεφαλίδες όπως η απάντηση σε ένα GET αίτημα. Αυτή η πληροφορία είναι ιδιαίτερα χρήσιμη εάν ο πελάτης έχει ήδη ένα αποθηκευμένο αντίγραφο του πόρου. Ανάλογα με το αν ο πόρος έχει μεταβληθεί ή όχι στον εξυπηρέτη πηγής, ο πελάτης θα μπορούσε να ακυρώσει τον αποθηκευμένο πόρο ή να ανανεώσει τον χρόνο εγκυρότητας του (freshness lifetime). Οι επικεφαλίδες του μηνύματος απάντησης μπορούν να φανούν χρήσιμες ακόμη και όταν ο εξυπηρέτης δεν έχει αποθηκευμένο αντίγραφο του πόρου. Για παράδειγμα, ο πελάτης μπορεί να ελέγξει τις επικεφαλίδες που σχετίζονται με την ενδιάμεση αποθήκευση όπως οι Last-Modified, Cache-Control ή Content-Type και να αποφασίσει εάν αξίζει να προχωρήσει με την προανάκτηση ολοκλήρου του πόρου. Ο πελάτης μπορεί να αποφασίσει να μην προανακτήσει ένα πόρο που δεν επιδέχεται αποθήκευση σε ενδιάμεση μνήμη ή αλλάζει πολύ τακτικά. Επίσης, ο πελάτης θα μπορούσε να εξετάσει το πεδίο Content-Length για να προσδιορίσει το ακριβές μέγεθος του πόρου. Ο πελάτης μπορεί να αποφασίσει να προ-ανακτήσει κάποιο πόρο αν η τιμή του πεδίου Content-Length βρίσκεται κάτω από συγκεκριμένο κατώφλι. Έτσι αποφεύγονται πόροι που

εισάγουν σημαντική δικτυακή επιβάρυνση.

Αντί να μεταδώσει ένα αίτημα HEAD προς τον εξυπηρέτη για να προσδιορίσει το μέγεθος του πόρου, ο πελάτης μπορεί να ελέγξει άμεσα τον όγκο της προανακτώμενης πληροφορίας παράγοντας μία GET ερώτηση με το πεδίο Range. Για παράδειγμα, ο πελάτης θα μπορούσε να αιτηθεί τα πρώτα 1000 bytes από τον πόρο. Αυτή η τακτική επιτρέπει στον πελάτη να αντλήσει πλήρως ένα μικρό πόρο ή το αρχικό τμήμα ενός μεγάλου πόρου. Σε ορισμένες περιπτώσεις η προανάκτηση του αρχικού τμήματος (prefix) ενός πόρου μπορεί να αποβεί ιδιαίτερα χρήσιμη. Για παράδειγμα, τα πρώτα bytes ενός GIF αρχείου περιέχουν το μέγεθος του αρχείου και μπορούν να περιέχουν ένα αντίγραφο της εικόνας χαμηλής ευκρίνειας. Η εκ των προτέρων διαθεσιμότητα του prefix της εικόνας επιτρέπει στον πελάτη να ξεκινήσει την γραφική απόδοσης της (graphic rendering). Το πρόθεμα (prefix) ενός αρχείου video ή audio περιέχει επαρκή αριθμό πλαισίων (frames) ώστε να ξεκινήσει το playback. Όταν ο χρήστης αιτηθεί τον πόρο, ένα δεύτερο αίτημα με το πεδίο range προκαλεί την ανάκτηση του υπολοίπου του πόρου.

2.2 Αλγόριθμοι προανάκτησης αντικειμένων

2.2.1 Top-10

Ο αλγόριθμος Top-10 προτάθηκε στις εργασίες [8], [9] και βασίζεται στην γνώση από την πλευρά του εξυπηρέτη για τον εντοπισμό των πλέον δημοφιλών εγγράφων, την χρήση μεσαζόντων εξυπηρετών για την συνάθροιση αιτήσεων προς τους εξυπηρέτες καθώς και τον επιμερισμό του κόστους προανάκτησης σε ένα μεγάλο αριθμό από πελάτες καθώς και στην προσαρμογή (adaptation) στα εξελισσόμενα πρότυπα πρόσβασης (access patterns) του χρήστη. Ειδικότερα, τα παραπάνω συστατικά του αλγορίθμου έχουν ως εξής:

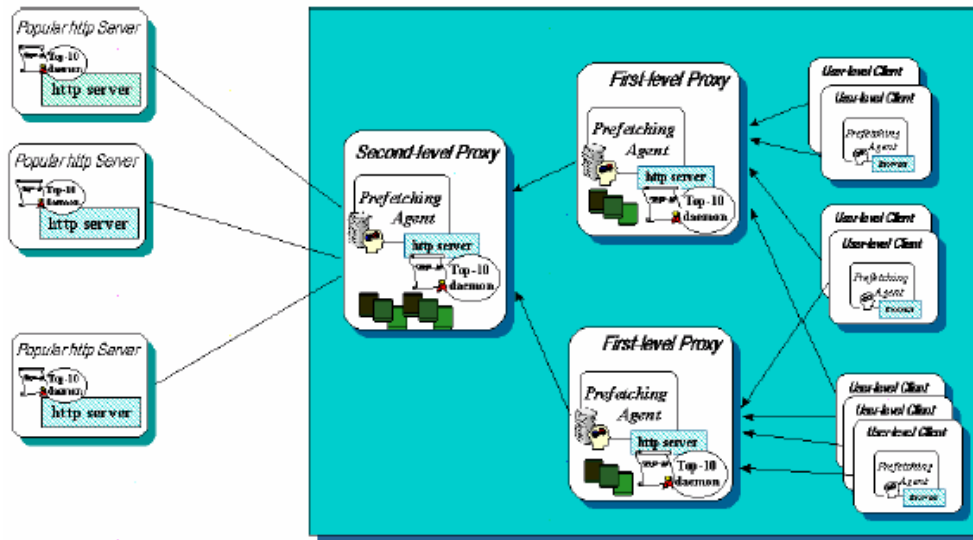
- Γνώση του εξυπηρέτη: Στο WWW τα δημοφιλή έγγραφα είναι πολύ δημοφιλή. Σε κάθε εξυπηρέτη ένα μικρό ποσοστό των αρχείων του αντιστοιχεί στο μεγαλύτερο ποσοστό των αιτήσεων σε αυτόν τον εξυπηρέτη. Αυτό το σύνολο αρχείων καλείται Top-10. Μόνο τα έγγραφα που αντιστοιχούν στο Top-10 λαμβάνονται υπόψη για την προανάκτηση. Ο πραγματικός αριθμός των αρχείων που περιέχεται στο Top-10 ρυθμίζεται βάσει των profiles των χρηστών λαμβάνοντας υπόψη τον όγκο των αιτήσεων που πραγματοποιήθηκαν στο πρόσφατο παρελθόν από τον πελάτη και το χώρο δευτερεύουσας αποθήκευσης που είναι διαθέσιμος για προανακτηθέντες

πόρους.

- Proxying: Ο τυπικός WWW χρήστης παράγει ένα μικρό αριθμό αιτήσεων προς ένα εξυπηρέτη. Το 30% των πελατών ενός εξυπηρέτη πραγματοποιούν μόνο μία αίτηση και δεν ζητούν κάτι από τον ίδιο εξυπηρέτη στην συνέχεια. Εάν κάποιος πελάτης πρόκειται να ζητήσει μόνο ένα έγγραφο από τον εξυπηρέτη, ο μηχανισμός της προανάκτησης δεν έχει ιδιαίτερο νόημα. Εάν, όμως οι πελάτες επικοινωνούν μέσω κάποιου μεσάζοντα εξυπηρέτη, η προανάκτηση των εγγράφων στον πελάτη μπορεί να βελτιώσει σημαντικά τις επιδόσεις του συστήματος εφόσον τα έγγραφα που προανακτώνται για λογαριασμό κάποιου πελάτη μπορούν να χρησιμοποιηθούν και από άλλους.
- Προσαρμοστικότητα (adaptation): Ο όγκος των αιτήσεων που παράγονται από τους διάφορους www πελάτες ποικίλει. Ορισμένοι παράγουν ιδιαίτερα υψηλούς όγκους ενώ άλλοι αιτούνται μόνο λίγα αντικείμενα. Ο αλγόριθμος Top-10 έχει σχεδιαστεί ώστε να προσαρμόζεται σε αυτές τις περιπτώσεις. Επιτρέπει έτσι σε χρήστες που ζητούν συχνά πόρους (frequent users) να προανακτήσουν πολλά έγγραφα ενώ οι περιστασιακοί χρήστες επιτρέπεται να προανακτήσουν σημαντικά λιγότερους πόρους ενδεχομένως και καθόλου. Η προανάκτηση ελέγχεται από το access profile του χρήστη. Το profile αυτό περιέχει τον αριθμό των εγγράφων που έχουν ζητηθεί από τον κάθε πελάτη στο πρόσφατο παρελθόν. Βάσει αυτού προσδιορίζεται το πλήθος των πόρων που θα πρέπει να προανακτηθούν από τον εξυπηρέτη στο μέλλον. Εάν το πρότυπο προσπέλασης (access pattern) αλλάξει, το Top-10 θα προσαρμοστεί και θα ξεκινήσει προανάκτηση από τους πλέον δημοφιλείς κόμβους.

Όπως αναφέρθηκε παραπάνω, ο αλγόριθμος βασίζεται στην συνεργασία πελάτη–εξυπηρέτη για την επιτυχή υλοποίηση ενεργειών προανάκτησης. Η πλευρά του εξυπηρέτη είναι υπεύθυνη για τον περιοδικό υπολογισμό της λίστας με τους πλέον δημοφιλείς πόρους (Top-10) και την προώθηση της στους πελάτες. Για να είναι βέβαιο ότι οι πόροι προανακτώνται μόνο στους πελάτες που μπορούν να τα χρησιμοποιήσουν, το Top-10 δεν αντιμετωπίζει όλους τους πελάτες ενιαία. Ο χρόνος θεωρείται διαιρεμένος σε διαστήματα (intervals) και η προανάκτηση από κάθε εξυπηρέτη ενεργοποιείται μόνο αφού ο συγκεκριμένος πελάτης έχει πραγματοποιήσει επαρκή αριθμό αιτήσεων στον συγκεκριμένο εξυπηρέτη (Access Threshold). Έτσι, σε περιστασιακούς πελάτες δεν προανακτώνται έγγραφα ενώ οι «συχνοί» πελάτες εντοπίζονται και λαμβάνονται σημαντικά υπόψη κατά την προανάκτηση.

Ορισμένοι πελάτες π.χ., μεσάζοντες εξυπηρέτες, παράγουν σημαντικά περισσότερες αιτήσεις από άλλους και ένας σωστά σχεδιασμένο αλγόριθμος θα πρέπει να προανακτήσει διαφορετικούς αριθμούς αντικειμένων στους διαφορετικούς πελάτες. Για παράδειγμα, δεν έχει νόημα να προανακτηθούν τα 500 πιο δημοφιλή έγγραφα σε ένα πελάτη που έκανε μόνο 10 αιτήσεις κατά το προηγούμενο χρονικό διάστημα. Λαμβάνοντας το πρόσφατο παρελθόν ως ένδειξη για το άμεσο μέλλον, ο πελάτης θα πραγματοποιήσει περίπου 10 αιτήσεις στο επόμενο χρονικό διάστημα και κατά το μέγιστο $10/500 = 2\%$ των προανακτηθέντων εγγράφων θα χρησιμοποιηθούν. Ένας άλλος πελάτης ο οποίος πραγματοποίησε 20000 αιτήσεις στο προηγούμενο χρονικό διάστημα μπορεί να ωφεληθεί σημαντικά από τα 500 προανακτηθέντα έγγραφα. Ο μηχανισμός Top-10 προσαρμόζει τον όγκο της προανάκτησης στους διάφορους πελάτες με βάση τον αριθμό των αιτήσεων που έχουν παραχθεί στο παρελθόν. Ένας πελάτης δεν μπορεί να προχωρήσει σε προανάκτηση περισσότερων εγγράφων από αυτά που ζήτησε στο πρόσφατο παρελθόν. Τέλος, για να διασφαλιστεί ότι η πολιτική Top-10 θα είναι λιγότερο ή περισσότερο επιθετική όπως απαιτείται, η παράμετρος Top-10 καθορίζει τον μέγιστο αριθμό πόρων που μπορούν να ανακτηθούν σε κάθε χρονικό διάστημα από οποιοδήποτε κόμβο. Ένας πελάτης δεν μπορεί να ζητήσει προανάκτηση περισσότερων από Top-10 πόρων ακόμη και αν έχει προσπελάσει σημαντικότερο αριθμό πόρων στο προηγούμενο διάστημα.

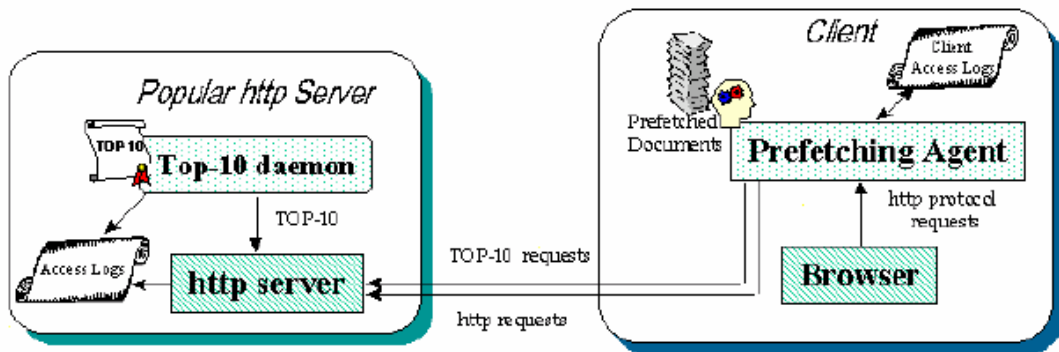


Εικόνα 2: Top-10 προανάκτηση

Η προανάκτηση μπορεί να υλοποιηθεί τόσο στο επίπεδο του πελάτη όσο και σε

επίπεδο μεσάζοντα εξυπηρέτη (Εικόνα 2). Οι πελάτες επιπέδου χρήστη ενεργοποιούν η προανάκτηση από τους μεσάζοντες εξυπηρέτες πρώτου επιπέδου λαμβάνοντας πρόνοια για τους χρήστες που εξυπηρετούν. Οι μεσάζοντες εξυπηρέτες πρώτου και δευτέρου επιπέδου παίζουν επίσης τους ρόλους του πελάτη και του εξυπηρέτη. Οι μεσάζοντες εξυπηρέτες 1^{ου} επιπέδου αποτελούν πελάτες για τα μεσάζοντες εξυπηρέτες 2^{ου} επιπέδου και προανακτούν και αποθηκεύουν έγγραφα για λογαριασμό των πελατών επιπέδου χρήστη. Οι μεσάζοντες εξυπηρέτες 2^{ου} επιπέδου αποτελούν πελάτες σε δημοφιλείς εξυπηρέτες από τους οποίους προανακτούν διάφορα έγγραφα τα οποία προωθούν στους πελάτες τους.

Η υλοποίηση του μηχανισμού Top-10 βασίζεται στην συνεργασία των οντοτήτων του πελάτη και του εξυπηρέτη (Εικόνα 3). Στη πλευρά του εξυπηρέτη, ο Top-10 daemon επεξεργάζεται όλα τα logs και διαμορφώνει την Top-10 λίστα, δηλαδή την λίστα των πλέον δημοφιλών πόρων στον συγκεκριμένο εξυπηρέτη. Ενημερώνει μία συγκεκριμένη web σελίδα ώστε η πληροφορία Top-10 να εμφανίζεται ως ένας άλλος πόρος στον εξυπηρέτη. Η συχνότητα εκτίμησης του Top-10 εξαρτάται από το πόσο συχνά μεταβάλλεται το περιεχόμενο του εξυπηρέτη.



Εικόνα 3: Συνεργασία πράκτορα προανάκτησης στην μεριά του πελάτη και http εξυπηρέτη

Στην πλευρά του πελάτη, ο πράκτορας προανάκτησης καταγράφει όλες τις http αιτήσεις του πελάτη και προσαρμόζει την δραστηριότητα προανάκτησης αυτού σε αυτές. Ο agent προανάκτησης συνεργάζεται με ένα μεσάζοντα εξυπηρέτη που φιλτράρει τις http αιτήσεις που υποβάλλονται από τον πελάτη. Εάν μία http αίτηση μπορεί να εξυπηρετηθεί από την τοπική ενδιάμεση μνήμη με τα έγγραφα προανακτηθέντα, ο μεσάζων εξυπηρέτης διεκπεραιώνει την αίτηση από την ενδιάμεση μνήμη. Σε διαφορετική περίπτωση, η αίτηση προωθείται στο μεσάζοντα εξυπηρέτη του επόμενου επιπέδου. Σε καθημερινή ή εβδομαδιαία βάση, ανάλογα με την ρύθμιση, ο πράκτορας

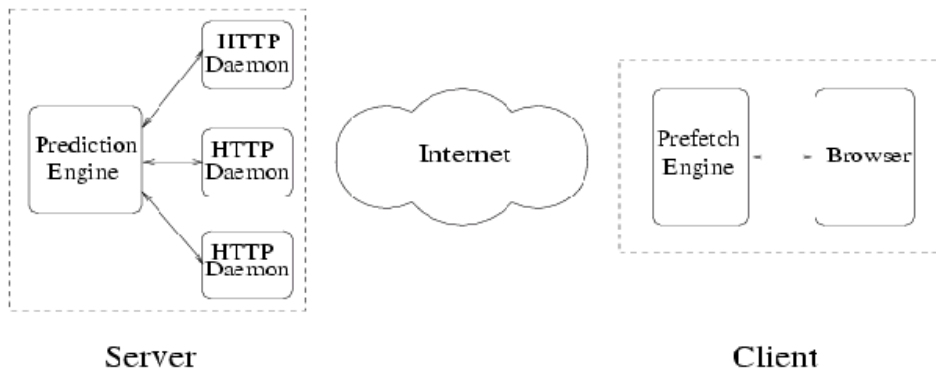
προανάκτησης διατρέχει τα client access logs τα οποία περιέχουν όλες τις http αιτήσεις που έχει υποβάλλει ο πελάτης και διαμορφώνει το προφίλ προανάκτησης αυτού του χρήστη (δηλαδή, την λίστα των εξυπηρετών από τους οποίους πρέπει να προανακτηθούν πόροι). Ο αριθμός των πόρων που έχουν ζητηθεί από τους συγκεκριμένους εξυπηρετές κατά την προηγούμενο χρονικό διάστημα υπερβαίνει το όριο Access_Threshold. Τελικά, βασιζόμενος στο προφίλ προανάκτησης αυτού του χρήστη, ο πράκτορας προανάκτησης ζητάει τα πιο δημοφιλή έγγραφα από τους εξυπηρετές που έχουν ενεργοποιηθεί για την διαδικασία της προανάκτησης. Ο αριθμός των εγγράφων που ζητούνται από κάθε εξυπηρετή ισούται με το ελάχιστο του αριθμού των αιτήσεων σε αυτόν τον εξυπηρετή ή το πλήθος της λίστας Top-10.

2.2.2 Προανάκτηση βασισμένη στην πρόβλεψη (Predictive Prefetching)

Στην εργασία [14] περιγράφεται ένας αλγόριθμος για την πρόβλεψη και την προανάκτηση πόρων που είναι πολύ πιθανόν να ζητηθούν σύντομα. Συγκεκριμένα, ο εξυπηρετής υπολογίζει την πιθανότητα να ζητηθεί ένας συγκεκριμένος πόρος στο άμεσο μέλλον και προωθεί αυτή την πληροφορία στον πελάτη. Ο πελάτης αποφασίζει αν θα προχωρήσει στην προ-ανάκτηση του πόρου. Ο εξυπηρετής έχει την δυνατότητα παρατήρησης των προτύπων πρόσβασης από διάφορους πελάτες και χρησιμοποιεί αυτή την πληροφορία για να προβεί σε ευφυείς προβλέψεις. Ο πελάτης είναι σε καλύτερη θέση να αποφασίσει αν θα εκμεταλλευτεί την παρεχόμενη πληροφορία για την προ-ανάκτηση των πόρων ανάλογα με την κατάσταση της τοπικής ενδιάμεσης μνήμης και το εκτιμώμενο κόστος ανάκτησης.

Αρχιτεκτονική του συστήματος προανάκτησης

Στην πλευρά του εξυπηρετή υπάρχουν δύο τύποι user-level διεργασιών. Το ένα αφορά ένα σύνολο HTTP daemon διεργασιών με υποστήριξη persistent συνδέσεων. Μία http διεργασία γεννάται για να διεκπεραιώσει το αίτημα ενός πελάτη. Εφόσον υποστηρίζονται persistent συνδέσεις, εκτελείται μία διεργασία ανά πελάτη και όχι ανά αίτημα. Η άλλη διεργασία είναι ο prediction daemon (predictd) ο οποίος παράγει τις προβλέψεις. Υπάρχει μόνο μία διεργασία prediction daemon ανά εξυπηρετή. Ο predictd επικοινωνεί μόνο με τον εξυπηρετή και όχι απευθείας με τους πελάτες.



Εικόνα 4: Αρχιτεκτονική συστήματος προανάκτησης

Όταν λάβει μία αίτηση από ένα πελάτη, το http περνάει την ταυτότητα του πελάτη και τα URL των αιτούμενων πόρων στον predictd. Ο predictd ενδιαφέρεται μόνο για τα αιτήματα που υποβάλλονται με την μέθοδο GET (ή παραλλαγές της). Ο predictd εφαρμόζει τον αλγόριθμο πρόβλεψης που αναλύεται στην επόμενη παράγραφο για να προσδιορίσει ποια αρχεία είναι υποψήφια για προανάκτηση βάσει της πιθανότητας ανάκτησης τους στο άμεσο μέλλον. Η πληροφορία αυτή μεταδίδεται στον πελάτη, riggy-backed στις απαντήσεις που στέλνονται από τον εξυπηρέτη, σε ειδικό πεδίο.

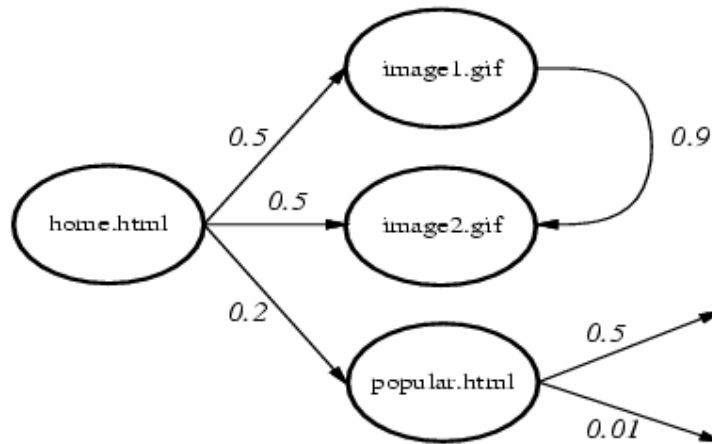
Στον πελάτη εκτελούνται ένας συμβατικός φυλλομετρητής και ένα μηχανή προανάκτησης. Η μηχανή προανάκτησης χρησιμοποιεί την πληροφορία πρόβλεψης που έχει μεταδώσει ο εξυπηρέτης για να αποφασίσει για την προανάκτηση των αρχείων. Η απόφαση επηρεάζεται από πολλές παραμέτρους όπως τα περιεχόμενα της τοπικής ενδιάμεσης μνήμης, τον φόρτο του συστήματος, την τρέχουσα κατάσταση λειτουργίας του φυλλομετρητή, κλπ.

Μόλις αποφασιστεί η προανάκτηση κάποιων αρχείων από το μηχανή προανάκτησης, κατάλληλες αιτήσεις μεταδίδονται προς τον εξυπηρέτη. Στις αιτήσεις αυτές προσδιορίζεται ότι δεν πρόκειται για την συνήθη ανάκτηση πόρων αλλά για προανάκτηση. Η πληροφορία αυτή μπορεί να φανεί χρήσιμη στον εξυπηρέτη. Ο predictd αποφασίζει να μην συμπεριλάβει τις αιτήσεις αυτές στον υπολογισμό των προβλέψεων του αφού πρόκειται για αιτήσεις προανάκτησης. Επίσης, σε περίπτωση που το σύστημα διεκπεραιώνει ήδη κάποιες αιτήσεις, οι αιτήσεις προανάκτησης λαμβάνουν χαμηλή προτεραιότητα.

Αλγόριθμος πρόβλεψης

Ο αλγόριθμος πρόβλεψης έχει ως εξής: κατασκευάζεται ένας γράφος εξαρτήσεων (dependency graph) που αποτυπώνει το πρότυπο πρόσβασης (access pattern) στα

διαφορετικά αρχεία του εξυπηρέτη. Ο γράφος έχει ένα κόμβο για κάθε αρχείο για το οποίο το σύστημα έχει δεχθεί αιτήσεις ανάκτησης. Υπάρχει μία ακμή από το κόμβο A στον B μόνο και μόνο αν κάποια χρονική στιγμή υπήρξε προσπέλαση στο B μετά από w προσπελάσεις μετά την επίσκεψη του A. Το w καλείται lookahead μέγεθος παραθύρου. Το βάρος της ακμής είναι το πηλίκο του αριθμού αναφορών το B μετά από ένα παράθυρο από την επίσκεψη στο A προς τον αριθμό αιτήσεων για το A. Αυτό το βάρος δεν είναι στην πραγματικότητα η πιθανότητα επίσκεψης του B αμέσως μετά το A. Έτσι, τα βάρη των ακμών που ξεκινούν από ένα συγκεκριμένο κόμβο δεν απαιτείται να έχουν άθροισμα το 1. Στο Εικόνα 5 παρουσιάζεται ένα τμήμα ενός υποθετικού γράφου εξαρτήσεων.



Εικόνα 5: Τμήμα του γράφου εξαρτήσεων

Ο γράφος εξαρτήσεων ενημερώνεται δυναμικά καθώς ο εξυπηρέτης δέχεται νέες αιτήσεις. Υπεύθυνη για την ενημέρωση είναι η διεργασία predictd, η οποία δέχεται πληροφορία από τις httpd διεργασίες που διεκπεραιώνουν αιτήσεις πελάτη. Διατηρεί ένα κυκλικό (ring) buffer, μεγέθους ίσου με το μέγεθος του παραθύρου, w , για κάθε πελάτη που είναι συνδεδεμένος με το σύστημα (υπόθεση: persistent συνδέσεις). Όταν λάβει μία νέα αίτηση από μία από τις httpd διεργασίες εισάγει τον προσδιοριστή (ID) του αρχείου που ζητήθηκε στον αντίστοιχο κυκλικό (ring) buffer. Μόνο οι καταχωρήσεις μέσα στο ίδιο ring buffer θεωρούνται συσχετιζόμενες, έτσι μόνο οι αντίστοιχες ακμές του γράφου εξαρτήσεων ενημερώνονται. Έτσι διαχωρίζονται οι αιτήσεις από διαφορετικούς πελάτες και αποφεύγονται προβλήματα λανθασμένων συσχετίσεων.

Το predictd βασίζει τις προβλέψεις του στον γράφο εξαρτήσεων. Όταν το A

προσπελαστεί, θα έχει νόημα να γίνει προ-ανάκτηση του B εφόσον το βάρος της ακμής από το A στο B είναι υψηλό (δηλαδή υπάρχει μεγάλη πιθανότητα προσπέλασης του B στο άμεσο μέλλον). Γενικά, το `predictd` ανακηρύσσει το B υποψήφιο για προανάκτηση εάν η ακμή από το A στο B έχει βάρος υψηλότερο από ένα όριο προανάκτησης (`prefetch threshold`), p . Είναι δυνατόν η μεταβλητή p να λαμβάνει διαφορετικές τιμές για κάθε πελάτη και να μεταβάλλεται δυναμικά.

2.3 Επιπτώσεις της προανάκτησης στο δίκτυο

Η προανάκτηση ως γνωστό έχει σαν στόχο την μείωση της εκλαμβανόμενης από το χρήστη καθυστέρησης. Αν γίνονται λανθασμένες προβλέψεις για τις αιτήσεις που πρόκειται να κάνει ο χρήστης στο μέλλον, τότε προστίθεται νέα επιπλέον κίνηση στο δίκτυο. Πολλές ερευνητικές εργασίες είχαν ως στόχο τη μείωση αυτού του επιπλέον εύρους ζώνης που απαιτεί ένας αλγόριθμος προανάκτησης, πράγμα το οποίο αντιμετωπιζόταν σαν το κυριότερο κόστος που εισήγαγε η προανάκτηση.

Στην εργασία [19] η προσοχή στρέφεται στη μείωση των καθυστερήσεων του δικτύου. Δείχνεται ότι ακόμα και στην περίπτωση που η προανάκτηση δεν εισάγει επιπλέον κίνηση, φέρνοντας δηλαδή πόρους που τελικά δεν ζητούνται, μπορεί να έχει σημαντικές αρνητικές επιπτώσεις στην επίδοση του δικτύου. Αυτό συμβαίνει διότι η προανάκτηση αλλάζει το μοτίβο σύμφωνα με το οποίο παράγονται αιτήσεις από τον κάθε πελάτη και άρα το μοτίβο κίνησης στο δίκτυο.

Είναι γνωστό ότι η παραγόμενη κίνηση έχει *ON/OFF* χαρακτηριστικά [24]. Το αποτέλεσμα μιας χωρίς την κατάλληλη προσοχή εφαρμογής της προανάκτησης οδηγεί στην αύξηση της διάρκειας τόσο των *ON* όσο και *OFF* περιόδων. Αυτό γίνεται διότι από τη μια όταν μετά από μια αίτηση του χρήστη ακολουθεί η προανάκτηση ενός ή και παραπάνω πόρων που ο αλγόριθμος υπολογίζει ότι θα ζητηθούν στο άμεσο μέλλον, και αυτή η μεταφορά των πόρων γίνεται στον μέγιστο ρυθμό που εκείνη τη στιγμή μπορεί να δώσει το δίκτυο, τότε είναι προφανές ότι έχουμε αύξηση της διάρκειας της *ON* περιόδου για κάθε πελάτη. Από την άλλη έπεται ότι λόγω ακριβώς της εφαρμογής της προανάκτησης θα υπάρχουν περιπτώσεις όπου ο χρηστής θα κάνει μια αίτηση η οποία θα ικανοποιείται πχ από την ενδιάμεση μνήμη του φυλλομετρητή, άρα δεν θα παραχθεί στην συγκεκριμένη περίπτωση κίνηση στο δίκτυο. Ως εκ τούτου έχουμε αύξηση και των *OFF* περιόδων.

Λόγω των παραπάνω αυξάνεται η διακύμανση της παραγόμενης κίνησης πράγμα που

όπως δείχνεται στο [19] έχει ως αποτέλεσμα αύξηση του μέσου μεγέθους των ουρών στους *routers* του δικτύου και συνεπώς την αύξηση της μέσης καθυστέρησης παράδοσης των πακέτων στον πελάτη.

Αυτό που προτείνεται για την εξάλειψη αυτού του φαινομένου, ώστε να γίνονται αισθητά από τους χρήστες τα οφέλη που μπορούν να προσφέρουν οι διάφοροι αλγόριθμοι που έχουν προταθεί στη βιβλιογραφία είναι το εξής: Μετά από την άφιξη της απάντησης στην αίτηση του χρήστη, ακολουθεί όπως είπαμε ένα *OFF* διάστημα. Προτείνεται ένα σχήμα πρόβλεψης της διάρκειας αυτού του διαστήματος. Μέσα στο προβλεφθέντα αυτό χρόνο, πριν δηλαδή την αμέσως επόμενη αίτηση του χρήστη θα πρέπει ο πόρος που σύμφωνα με τις προηγούμενες αιτήσεις του πελάτη και έναν αλγόριθμο πρόβλεψης, θα ζητηθεί στην επόμενη αίτηση, να φτάσει στον πελάτη. Με τον κατάλληλο περιορισμό του παραθύρου του *TCP* γίνεται εφικτό να έρθει αυτός ο πόρος πριν γίνει η επόμενη αίτηση, αλλά όχι στο μέγιστο ρυθμό που εκείνη τη στιγμή μπορεί να δώσει το δίκτυο, αλλά σε τέτοιο ώστε ο πόρος να φτάσει στον πελάτη πριν την επόμενη αίτηση.

Με τη παραπάνω διαδικασία δείχνεται [19] ότι μειώνονται η δυσάρεστες συνέπειες που περιγράφηκαν παραπάνω.

2.4 Αλγόριθμοι προανάκτησης σε δίκτυο διανομής περιεχομένου

2.4.1 Γενικά

Η ενδιάμεση αποθήκευση χρησιμοποιείται ευρέως για τη μείωση της καθυστέρησης που προσλαμβάνει ο χρήστης του Διαδικτύου. Όπως έχει αναφερθεί υπάρχουν διάφορα είδη ενδιάμεσης αποθήκευσης. Αυτό μπορεί να γίνει για παράδειγμα στον φυλλομετρητή του χρηστή αλλά και σε ενδιάμεσους μεσάζοντες εξυπηρέτες ανάμεσα σε αυτόν και τον εξυπηρέτη πηγής. Ωστόσο αυτές οι τεχνικές χρησιμοποιούν ενδιάμεση αποθήκευση κατ' απαίτηση (*on demand caching*) ή βραχυπρόθεσμη ενδιάμεση αποθήκευση (*short term caching*) και τα αντικείμενα που αποθηκεύονται σε ενδιάμεση μνήμη καθορίζονται από τις πρόσφατες προσπελάσεις του χρηστή. Η μακροπρόθεσμη ενδιάμεση αποθήκευση (*long term caching*) από την άλλη είναι ένας μηχανισμός που επιτρέπει στους πελάτες (πχ μεσάζοντες εξυπηρέτες) να εγγράφονται για κάθε αντικείμενο που θέλουν να έχουν πάντα 'φρέσκο' και έτσι αυτό να τους προωθείται από τον εξυπηρέτη πηγής οποτεδήποτε αυτό αλλάζει. Αυτό κάνει τον λόγο επιτυχίας ενδιάμεσης μνήμης (*cache hit rate*) για αυτά τα αντικείμενα πάντα ίσο με 1. Κριτήρια για την επιλογή των αντικειμένων στα οποία θα εγγραφεί ένας πελάτης-μεσάζων είναι μακροπρόθεσμα (*long-term*

(statistical)) χαρακτηριστικά, όπως μέση συχνότητα πρόσβασης, διαστήματα ενημέρωσης και μέγεθος των αντικειμένων και όχι βραχυπρόθεσμα μοτίβα πρόσβασης (short-term (recent) access patterns) σε μεμονωμένες ενδιάμεσες μνήμες. Οι τιμές αυτών των μακροπρόθεσμων χαρακτηριστικών για τα όλα τα αντικείμενα συγκεντρώνονται και διαμοιράζονται μεταξύ εξυπηρετών διανομής περιεχομένου. Έχουν προταθεί διάφοροι αλγόριθμοι που έχουν σαν στόχο την βελτιστοποίηση κάποιων μετρικών.

2.4.2 Popularity

Το 1998 ο Μαρκάτος [9] πρότεινε ένα top10 αλγόριθμο για προανάκτηση. Σύμφωνα με αυτόν κάθε εξυπηρετής κρατάει στατιστικά στοιχεία για το ποια είναι τα 10 πιο δημοφιλή του αντικείμενα. Έτσι μπορούν οι πελάτες αυτού του εξυπηρετή να προανακτήσουν αυτά τα αντικείμενα και επειδή αυτά είναι δημοφιλή να αυξάνουν το λόγο επιτυχίας φτάνοντας το στο 60% με μόνο 10% αύξηση στο απαιτούμενο εύρος ζώνης. Στο prefetch by popularity χρησιμοποιείται μια παραλλαγή αυτού του αλγορίθμου σύμφωνα με την οποία τα n πιο δημοφιλή αντικείμενα διατηρούνται στην ενδιάμεση μνήμη και ενημερώνονται οποτεδήποτε αυτά αλλάζουν στον εξυπηρετή πηγής που είναι υπεύθυνος για αυτά. Δεδομένου ότι κρατάμε στην ενδιάμεση μνήμη τα πιο δημοφιλή αντικείμενα περιμένουμε να έχουμε τη μεγαλύτερη δυνατή αύξηση στο λόγο επιτυχίας. Επειδή ακριβώς όμως δεν λαμβάνουμε υπόψη το μέγεθος αυτών των πόρων, έπεται ότι η αύξηση στο απαιτούμενο εύρος ζώνης θα είναι εξίσου μεγάλη

2.4.3 Prefetching by Lifetime

Η διάρκεια ζωής (lifetime) ενός αντικειμένου είναι το χρονικό διάστημα που μεσολαβεί μεταξύ δύο διαδοχικών τροποποιήσεων του αντικειμένου. Είναι φανερό λοιπόν πως αν φέρνουμε αντικείμενα των οποίων η διάρκεια ζωής είναι μεγάλο περιμένουμε η αύξηση στο απαιτούμενο εύρος ζώνης να είναι η ελάχιστη δυνατή.

2.4.4 Prefetching by Good-Fetch

Αυτός ο αλγόριθμός προτάθηκε από τον Venkataramani [16]. Αυτός ο αλγόριθμος δεν επιλέγει αποκλειστικά τα πιο δημοφιλή αντικείμενα ή αυτά που έχουν μεγαλύτερο lifetime αλλά κάτι ενδιάμεσο. Είναι γνωστό ότι ένα από τα προβλήματα της προανάκτησης είναι ότι ένα αντικείμενο το οποίο προανακτήθηκε μπορεί να μην χρησιμοποιηθεί πριν γίνει stale. Αν ένα αντικείμενο ζητηθεί πριν γίνει stale τότε θεωρείται good fetch. Αυτός ο αλγόριθμος υπολογίζει την πιθανότητα ένα αντικείμενο

να είναι good fetch και προανακτά τα αντικείμενα των οποίων η πιθανότητα αυτή ξεπερνά ένα συγκεκριμένο κατώφλι. Για ένα αντικείμενο i έστω ότι το lifetime είναι l_i , η πιθανότητα να ζητηθεί είναι p_i και ο ρυθμός άφιξης αιτήσεων από χρηστές για ολόκληρη την ενδιάμεση μνήμη είναι α . Τότε η πιθανότητα αυτό το αντικείμενο να ζητηθεί πριν να γίνει stale είναι:

$$P_{goodfetch} = 1 - (1 - p_i)^{\alpha l_i} \quad (\text{Εξ.2.1})$$

Οπού αl_i είναι ο αριθμός των αιτήσεων που καταφθάνουν όσο το αντικείμενο είναι ακόμα φρέσκο, $(1 - p_i)^{\alpha l_i}$ είναι η πιθανότητα ότι καμιά από αυτές τις αιτήσεις δεν αφορά το συγκεκριμένο αντικείμενο i , και έτσι $1 - (1 - p_i)^{\alpha l_i}$ είναι η πιθανότητα το αντικείμενο αυτό να ζητηθεί πριν γίνει stale.

2.4.5 Prefetching by APL

Ισχύει ο ίδιος συμβολισμός με παραπάνω. Έτσι για ένα αντικείμενο i το $\alpha p_i x l_i$ είναι ο αριθμός των φορών που αυτό ζητείται πριν γίνει stale. Σύμφωνα λοιπόν με αυτόν τον αλγόριθμο επιλέγονται για προανάκτηση τα αντικείμενα τα οποία αναμένεται να ζητηθούν τον μεγαλύτερο αριθμό φορών. Ο αλγόριθμος μπορεί να τροποποιηθεί ώστε να δίνει μεγαλύτερη έμφαση στην δημοφιλία των αντικείμενων. Τότε το κριτήριο με βάση το οποίο γίνεται η επιλογή είναι $\alpha x (p_i)^n x l_i$. Με αυτόν τον τρόπο όταν $n > 1$ αυτός ο αλγόριθμος δουλεύει περισσότερο σαν τον popularity και έτσι χρησιμοποιείται περισσότερο εύρος ζώνης για την βελτίωση του χρόνου απόκρισης. Όταν αντιθέτως το $n < 1$ τότε ο αλγόριθμος δουλεύει περισσότερο σαν τον Lifetime

2.4.6 Χαρακτηριστικά σταθερής κατάστασης

Υποθέτουμε ότι ο ρυθμός των αιτήσεων ακολουθεί την κατανομή Poisson και ότι ο μέσος ρυθμός αφίξεων είναι α . Επιπλέον σύμφωνα με το [27] η διάρκεια ζωής ενός αντικειμένου ακολουθεί την εκθετική κατανομή. Επιπλέον είναι γνωστό ότι η πιθανότητα να ζητηθεί ένα αντικείμενο που έχει popularity= i ακολουθεί την κατανομή μια κατανομή τύπου Zipf.

Παρουσιάζουμε σε αυτή την ενότητα ένα αναλυτικό μοντέλο για την σύγκριση μέσω προσομοίωσης αυτών των αλγορίθμων όπως αυτό παρουσιάζεται στο [25].

Έστω $P_{Ai}(t)$ η πιθανότητα να ζητηθεί το αντικείμενο i t χρονικές μονάδες πριν από την

Σχήμα Προφόρτωσης Περιεχομένου WWW

τρέχουσα τιμή του χρόνου και $P_{Bi}(t)$ η πιθανότητα ότι δεν θα αλλάξει στον εξυπηρέτη πηγής αυτό το αντικείμενο για t χρονικές μονάδες. Τότε η πιθανότητα να έχουμε επιτυχία σε μια ενδιάμεση μνήμη οπου δεν χρησιμοποιείται προανάκτηση είναι

$$P_{hit_d}(i) = \sum_i p_i \int_0^{\infty} P_{A_i}(t) P_B(t) dt \quad (\text{Εξ.2.2})$$

Έστω ότι $P_{(a,t)}(k)$ είναι η πιθανότητα να ζητηθεί ένα αντικείμενο k φορές μέσα σε t χρονικές μονάδες δεδομένου ενός ρυθμού άφιξης αιτήσεων ίσου με a . Δεδομένου ότι ο ρυθμός αφίξεων ακολουθεί την κατανομή Poisson η πιθανότητα αυτή είναι $P_{(a,t)}(k) = e^{-at} \frac{(at)^k}{k!}$. Επιπλέον η πιθανότητα το αντικείμενο i να ζητηθεί σε μια αίτηση είναι p_i . Έτσι η πιθανότητα να μην ζητηθεί το αντικείμενο i σε αυτό το χρονικό διάστημα τα είναι

$$P(0 \text{ reqs for } i \text{ in } t \text{ time}) = \sum_{k=0}^{\infty} P(k \text{ reqs in } t) P(\text{none of those for } i) = \sum_{k=0}^{\infty} \left(e^{-at} \frac{(at)^k}{k!} \right) (1 - p_i)^k = e^{-(ap_i)t} \quad (\text{Εξ.2.3})$$

Η παραπάνω εξίσωση υπονοεί ότι η κατανομή του χρόνου μεταξύ διαδοχικών προσπελάσεων στο αντικείμενο i ακολουθεί την εκθετική κατανομή με μέσο $(1/ap_i)$

Έτσι η πιθανότητα να ζητηθεί το αντικείμενο i t χρονικές μονάδες μετά την προηγούμενη ακριβώς φορά που αυτό ζητήθηκε είναι $(ap_i)e^{-(ap_i)t}$ το οποίο αποτελεί επίσης και την κατανομή του $P_{A_i}(t)$.

Επιπλέον δεδομένου ότι ο χρόνος ζώνης ενός αντικειμένου i ακολουθεί την εκθετική κατανομή με μέσο l_i έπεται ότι η πιθανότητα να μην γίνει καμία ενημέρωση στο αντικείμενο i σε t χρονικές μονάδες είναι $P_{B_i}(t) = e^{-t/l_i}$

Από τα παραπάνω έπεται ότι η πιθανότητα επιτυχίας σε μία αίτηση είναι

$$P_{hit_d}(i) = \sum_i p_i \int_0^{\infty} ((ap_i)e^{-(ap_i)t})(e^{-t/l_i}) dt = \sum_i p_i \left(\frac{ap_i l_i}{ap_i l_i + 1} \right) \quad (\text{Εξ.2.4})$$

Το κλάσμα $\frac{ap_i l_i}{ap_i l_i + 1}$ είναι ο λόγος επιτυχιών για τις requests που ζητούν το αντικείμενο i και από δω και πέρα θα το λέμε freshness factor(i) και θα το συμβολίζουμε με $ff(i)$.

Δεδομένου ότι σε αυτή τη σειρά αλγορίθμων τα αντικείμενα που έχουν επιλεγεί για προανάκτηση ενημερώνονται στους μεσάζοντες εξυπηρέτες οποτεδήποτε αυτά ενημερωθούν στον αντίστοιχο εξυπηρέτη πηγής έπεται ότι $ff(i)=1$ για αυτά τα αντικείμενα. Επομένως ο λόγος επιτυχίας σταθερής κατάστασης για μια ενδιάμεση μνήμη προανάκτησης με κατώφλι για την επιλογή ενός αντικειμένου το T είναι

$$P_{hit_d}(i, T) = \sum_i p_i h_i \text{ όπου } h_i = 1 \text{ αν } P_{goodFetch}(i) > T \text{ διαφορετικά } \frac{ap_i l_i}{ap_i l_i + 1} \quad (\text{Εξ.2.5})$$

Για τους ίδιους λόγους που εξηγήσαμε και παραπάνω έπεται ότι το απαιτούμενο εύρος ζώνης στην περίπτωση μιας ενδιάμεσης μνήμης προανάκτησης είναι

$$BW_{pref} = \sum_i b_i \text{ όπου } b_i = ap_i(1 - f(i))s_i \text{ αν το } i \text{ δεν προανακτάται, και } \frac{s_i}{l_i} \text{ αν } \quad (\text{Εξ.2.6})$$

προανακτάται

2.5 Αξιολόγηση των αλγορίθμων

Στο [25] η αξιολόγηση του Good Fetch αλγόριθμου έγινε με βάση μια μετρική που δείχνει πόσο βελτιώνεται ο λόγος επιτυχιών της ενδιάμεσης μνήμης σε σχέση με την αύξηση του εύρους ζώνης που προκύπτει από την προανάκτηση. Με την υιοθέτηση μικρότερου κατωφλιού έχουμε μεγαλύτερο λόγο επιτυχιών αλλά και κατανάλωση περισσότερου εύρους ζώνης λόγω των περισσότερων αντικειμένων που προανακτώνται. Ανάλογα με το πόσο σημαντική είναι η από τον χρήστη εκλαμβανόμενη καθυστέρηση σε σχέση με την κατανάλωση παραπάνω εύρος ζώνης, μπορούν να επιλεγούν διαφορετικά κατώφλια για αυτόν τον αλγόριθμο. Η μετρική λοιπόν που αναφέραμε παραπάνω για την αξιολόγηση του αλγόριθμου είναι

$$H / B = \frac{Hit_{pref} / Hit_{demand}}{BW_{pref} / BW_{demand}} \quad (\text{Εξ.2.7})$$

Μια παραλλαγή αυτής της μετρικής μπορεί να χρησιμοποιηθεί για να δοθεί σχετικά μεγαλύτερη έμφαση στην αύξηση του λόγου επιτυχιών ή του εύρους ζώνης

$$H^k / B = \frac{(Hit_{pref} / Hit_{demand})^k}{BW_{pref} / BW_{demand}} \quad (\text{Εξ.2.8})$$

Αν $k > 1$ τονίζει την σημαντικότητα της βελτίωσης του λόγου επιτυχιών (δηλαδή μπορεί για παράδειγμα να έχουμε απεριόριστο εύρος ζώνης οπότε να μας ενδιαφέρει η αύξηση του λόγου επιτυχιών με οποιοδήποτε κόστος στην αύξηση του απαιτούμενου εύρους ζώνης). Αν από την άλλη το διαθέσιμο εύρος ζώνης είναι περιορισμένο τότε θέτουμε $k < 1$. Στο [20] μολονότι χρησιμοποιήθηκαν οι παραπάνω μετρικές για την αξιολόγηση των Good Fetch, Popularity, αλλά και των Lifetime και APL δεν προτάθηκε ωστόσο κάποιος αλγόριθμος που θα βελτιστοποιεί αυτές τις μετρικές.

2.5.1 Προβλήματα των προαναφερθέντων αλγορίθμων

Πρώτα θα δούμε ότι οι Good Fetch και APL αντιμετωπίζουν διάφορα προβλήματα

Για παράδειγμα στην περίπτωση του Good Fetch προανακτάται το αντικείμενο που έχει τη μεγαλύτερη πιθανότητα να ζητηθεί όσο αυτό είναι ακόμα φρέσκο. Αν λοιπόν ένα αντικείμενο δεν προανακτάται και κάποια στιγμή αλλάξει η συχνότητα σύμφωνα με την οποία αυτό ζητείται και ενημερώνεται και έτσι αυτό ζητείται ακριβώς μία φορά ανάμεσα σε δύο ενημερώσεις, τότε ο λόγος επιτυχιών κατ' απαίτηση (on demand hit rate) είναι 0 και η κατανάλωση εύρους ζώνης είναι $\frac{S_i}{L_i}$. Αν τώρα αποφασίσουμε να προανακτήσουμε

αυτό το αντικείμενο, τότε ο λόγος επιτυχιών αυτού του αντικειμένου βελτιώνεται σε 1 ενώ το απαιτούμενο εύρος ζώνης παραμένει το ίδιο! Η παραπάνω παρατήρηση υπονοεί ότι το αντικείμενο i είναι ιδανικό για προανάκτηση. Ωστόσο αυτός ο «τέλειος υποψήφιος για prefetching» δεν ευνοείται πάντα από τον Good Fetch γιατί το $P_{goodfetch}$ δεν είναι πάντα ψηλό σε σχέση με αυτό των υπόλοιπων αντικειμένων. Ένα ανάλογης φύσεως πρόβλημα έχει και ο APL.

2.6 H/B Greedy προανάκτηση

Ως γνωστό ένας αλγόριθμος προανάκτησης επιλέγει ένα υποσύνολο των αντικειμένων της ενδιάμεσης μνήμης για να κάνει προανάκτηση. Όταν ένα αντικείμενο i προανακτάται

Σχήμα Προφόρτωσης Περιεχομένου WWW

τότε ο λόγος επιτυχιών του αυξάνεται από $\frac{ap_i l_i}{1 + ap_i l_i}$ σε 1 το οποίο είναι $\frac{1}{f(i)}$ φορές αυτό της κατ' απαίτηση ενδιάμεσης αποθήκευσης, ενώ το απαιτούμενο εύρος ζώνης αυξάνει από $\frac{s_i}{l_i + \frac{1}{ap_i}}$ σε $\frac{s_i}{l_i}$ το οποίο είναι επίσης $\frac{1}{f(i)}$ φορές αυτού που απαιτείται στην

περίπτωση της κατ' απαίτηση ενδιάμεσης αποθήκευσης.

Χρησιμοποιώντας την μετρική H/B έχουμε στην περίπτωση της προανάκτησης

$$(H/B)_{pref} = \frac{Hit_{pref}}{BW_{pref}} = \frac{\sum_i p_i h_i}{\sum_i b_i} \quad (\text{Εξ.2.9})$$

ενώ στην περίπτωση της κατ' απαίτηση ενδιάμεσης αποθήκευσης

$$(H/B)_{demand} = \frac{Hit_{demand}}{BW_{demand}} = \frac{\sum_i p_i f(i)}{\sum_i \frac{s_i}{l_i} f(i)} \quad (\text{Εξ.2.10})$$

Για τα αντικείμενα που επιλέγονται από τον αλγόριθμο αντικαθιστούμε το $f(i)$ με 1 για να πάρουμε το $(H/B)_{pref}$

Στο[25] παρουσιάζεται λοιπόν ένας greedy αλγόριθμος για την μεγιστοποίηση αυτής της μετρικής.

Είναι

$$S' = \arg \max_{S' \subset S, |S'|=m} \left[\left(\frac{H}{B} \right)_{pref} \right] \quad (\text{Εξ.2.11})$$

Σε κάθε βήμα επιλέγεται το αντικείμενο αυτό που θα αυξήσει περισσότερο την μετρική.

Η αύξηση αυτή είναι

$$\left(\frac{H}{B} \right)_{demand} \times incr(j) \quad (\text{Εξ.2.12})$$

Εδώ το $incr(j)$ είναι ο παράγοντας που δείχνει το κατά πόσο θα αυξηθεί το H/B αν το

αντικείμενο j επιλεγεί για προανάκτηση. Ο *greedy* αλγόριθμος επιλέγει λοιπόν σε κάθε βήμα το αντικείμενο που έχει τον μεγαλύτερο παράγοντα αύξησης (increase factor)

Algorithm 1 H/B-Greedy prefetching

```

1: procedure H/B-Greedy( $S, m, a$ )
2: Inputs:
3:   a set of objects of type  $\langle p_i, l_i, s_i \rangle$ :  $S$ 
4:   number of objects to be prefetched:  $n$ 
5:   the total access rate:  $a$ 
6:
7: for each object  $i \in S$  do
8:   compute the freshness factor:  $f(i) = \frac{ap_i l_i}{ap_i l_i + 1}$ 
9: end for
10: compute the on-demand overall hit rate  $Hit_{demand} = \sum_{i \in S} p_i f(i)$ 
11: compute the on-demand overall bandwidth  $BW_{demand} = \sum_{i \in S} \frac{s_i}{l_i} f(i)$ 
12: for each object  $i \in S$  do
13:   compute the increase factor  $incr(i)$  as defined in Equation (15)
14: end for
15: select  $m$  objects with the largest increase factors using randomized selection
16: mark the selected objects "prefetched"
17:
18:  $Hit_{pref} = Hit_{demand}$ 
19:  $BW_{pref} = BW_{demand}$ 
20: for each object  $j$  that is prefetched do
21:    $Hit_{pref} = Hit_{pref} + p_j(1 - f(j))$ 
22:    $BW_{pref} = BW_{pref} + \frac{s_j}{l_j}(1 - f(j))$ 
23: end for
24: return  $\frac{Hit_{pref}}{BW_{pref}}$ 

```

Εικόνα 6: Αλγόριθμος H/B-Greedy

Αλλάζοντας τα κριτήρια σύμφωνα με τα οποία γίνεται η επιλογή των αντικείμενων σε κάθε βήμα μπορούν να προκύψουν δύο διαφορετικοί αλγόριθμοι ο Hit Rate Greedy και ο Bandwidth Greedy.

2.6.1 Hit Rate Greedy προανάκτηση

Προανακτώντας το αντικείμενο i παρατηρούμε ότι η συνεισφορά του στην αύξηση του λόγου επιτυχιών είναι

$$H_contr(i) = \frac{P_i}{ap_i l_i + 1} \quad (\text{Εξ.2.13})$$

Αν λοιπόν γίνει αυτό το κριτήριο μας για την επιλογή αντικειμένου σε κάθε βήμα τότε μεγιστοποιείται ο τελικός λόγος επιτυχιών. Ο Hit Rate Greedy είναι μια ακραία περίπτωση του H^k / B για k να τείνει στο άπειρο

$$\lim_{k \rightarrow +\infty} \left[\frac{(Hit_{pref} / Hit_{demand})^k}{BW_{pref} / BW_{demand}} \right]^{1/k} = Hit_{pref} / Hit_{demand} \quad (\text{Εξ.2.14})$$

Το αξιοσημείωτο είναι ότι αυτός ο αλγόριθμος πετυχαίνει μεγαλύτερο λόγο επιτυχιών από τον Popularity! Αυτό συμβαίνει γιατί όπως φαίνεται από το κριτήριο επιλογής σε κάθε βήμα λαμβάνεται υπόψη και το διάρκεια ζωής (lifetime) του αντικείμενου πράγμα που στο Popularity δεν γίνεται πράγμα που έχει ως συνέπεια πως τα δημοφιλή αντικείμενα που προανακτώνται μπορεί να έχουν ελάχιστη διάρκεια ζωής και έτσι να μην προλάβουν να ζητηθούν πριν γίνουν stale!

2.6.2 BandwidthGreedy προανάκτηση

Αντίστοιχα έχουμε τον Bandwidth Greedy αλγόριθμο όπου το κριτήριο για την επιλογή ενός αντικείμενου σε κάθε βήμα είναι

$$B_contr(i) = \frac{s_i}{ap_i l_i^2 + l_i} \quad (\text{Εξ.2.15})$$

Αυτός ο αλγόριθμος είναι μια ακραία περίπτωση του H^k / B όταν το k τείνει στο 0.

Το αξιοσημείωτο είναι ότι αυτός ο αλγόριθμος απαιτεί μικρότερο εύρος ζώνης από τον Prefetch by Lifetime! Αυτό συμβαίνει γιατί όπως φαίνεται από το κριτήριο επιλογής σε κάθε βήμα λαμβάνεται υπόψη και η δημοφιλία του αντικείμενου πράγμα που στο Prefetch by Lifetime δεν γίνεται πράγμα που έχει ως συνέπεια πως τα αντικείμενα που έχουν μεγάλη διάρκεια ζωής να υπάρχει περίπτωση να είναι ελάχιστα δημοφιλή οπότε να μην ζητούνται καθόλου και έτσι να καταναλώνεται περισσότερο εύρος ζώνης για την εξυπηρέτηση των αιτήσεων λόγω του μικρότερου λόγου επιτυχιών!

ΚΕΦΑΛΑΙΟ: 3 ΘΕΩΡΙΑ ΠΑΙΓΝΙΩΝ

3.1 Γενικά

Η Θεωρία Παιγνίων μπορεί να οριστεί ως η μελέτη των μαθηματικών μοντέλων της αντιπαράθεσης (conflict) και συνεργασίας μεταξύ ευφυών και λογικών ληπτών αποφάσεων (decision makers). Η Θεωρία Παιγνίων παρέχει γενικές μαθηματικές τεχνικές για την ανάλυση καταστάσεων όπου δύο ή περισσότερα άτομα λαμβάνουν αποφάσεις που θα επηρεάσουν την ευημερία (welfare) των υπολοίπων. Οι καταστάσεις που εξετάζονται από τους παιγνιοθεωρητικούς δεν είναι απλά δραστηριότητες ψυχαγωγίας (όπως ο όρος games αφήνει να εννοηθεί). Οι όροι «ανάλυση αντιπαράθεσης» (conflict analysis) ή διαδραστική θεωρία αποφάσεων (interactive decision theory) είναι πιο κατάλληλοι όροι για το αντικείμενο μελέτης.

Οι παιγνιοθεωρητικοί προσπαθούν να κατανοήσουν την αντιπαράθεση και την συνεργασία μέσω της μελέτης ποσοτικών μοντέλων και υποθετικών παραδειγμάτων. Στη Θεωρία Παιγνίων η έννοια του «παιγνίου» υποδηλώνει μία κοινωνική κατάσταση όπου εμπλέκονται δύο ή περισσότερα άτομα. Τα άτομα αυτά καλούνται «παίκτες» (players). Οι παιγνιοθεωρητικοί κάνουν δύο βασικές υποθέσεις για τους παίκτες, ότι δηλαδή είναι λογικοί και έξυπνοι.

Ένας λήπτης απόφασης χαρακτηρίζεται λογικός (rational) εάν λαμβάνει αποφάσεις που βρίσκονται σε συνέπεια προς τις επιδιώξεις του. Στην θεωρία παιγνίων, θεωρείται ότι ο κάθε παίκτης έχει ως στόχο του την μεγιστοποίηση της μέσης τιμής της ωφέλειας του (payoff), η οποία μετράται σε μία συγκεκριμένη κλίμακα ωφέλειας. Η ιδέα ότι ένας λογικός λήπτης απόφασης θα πρέπει να λαμβάνει αποφάσεις που μεγιστοποιούν την αναμενόμενη ωφέλεια ανήκει αρχικά στον Bernoulli (1738) ενώ η σύγχρονη θεώρηση του προβλήματος ανήκει στους John von Neumann και Oscar Morgenstern (1947). Έδειξαν, με ιδιαίτερα ασθενείς υποθέσεις, ότι για κάθε λογικό λήπτη αποφάσεων πρέπει να υπάρχει ένας τρόπος ανάθεσης αριθμών ωφέλειας (utility numbers) στα διάφορα πιθανά ενδεχόμενα για τα οποία ενδιαφέρεται έτσι ώστε να διαλέγει πάντοτε την εναλλακτική εκείνη λύση η οποία μεγιστοποιεί την αναμενόμενη του ωφέλεια του. Αυτό το αποτέλεσμα καλείται θεώρημα μεγιστοποίησης της αναμενόμενης ωφέλειας (expected utility maximization theorem).

Τα λογικά αξιώματα τα οποία υποστηρίζουν το θεώρημα μεγιστοποίησης της

αναμενόμενης ωφέλειας είναι υποθέσεις ασθενούς συνέπειας (*weak consistency assumptions*). Η βασική υπόθεση είναι το αξίωμα αντικατάστασης (*substitution axiom*) το οποίο μπορεί να αποδοθεί ως εξής: εάν ένας λήπτης αποφάσεων προτιμάει την εναλλακτική λύση 1 από την 2, όταν συμβεί το συμβάν A και προτιμάει την λύση 1 ακόμη και αν δεν συμβεί η A, τότε θα προτιμήσει το ενδεχόμενο 1 ακόμη και αν δεν γνωρίσει εάν στο συμβάν A συμβεί ή όχι. Αυτή η υπόθεση είναι σε θέση να εξασφαλίσει ότι υπάρχει κάποια κλίμακα ωφέλειας στην οποία ο λήπτης απόφασης πάντα προτιμά τις εναλλακτικές επιλογές που μεγιστοποιούν την αναμενόμενη συνάρτηση ωφέλειας.

Συνεπής συμπεριφορά μεγιστοποίησης ωφέλειας μπορεί να προκύψει από μοντέλα εξελικτικής επιλογής. Σε ένα κόσμο (*universe*) αυξανόμενης αταξίας αποτελεί φυσικό νόμο ότι σύνθετοι οργανισμοί μπορούν να συνεχίσουν να υφίστανται μόνο εάν συμπεριφέρονται κατά τρόπο που αυξάνει την πιθανότητα επιβίωσης και αναπαραγωγής τους. Έτσι, ένα επιχείρημα υπέρ της εξελικτικής επιλογής καθορίζει ότι τα άτομα προσπαθούν να μεγιστοποιήσουν την αναμενόμενη τιμή ενός μέτρου γενικής επιβίωσης και ικανότητας αναπαραγωγής.

Γενικά, η μεγιστοποίηση της αναμενόμενης ωφέλειας δεν συμβαδίζει με την μεγιστοποίηση της χρηματικής απολαβής (*monetary payoff*). Ένα άτομο το οποίο αποφεύγει το ρίσκο (*risk averse*) μπορεί να ωφελείται (προσθετικά) περισσότερο από μία *extra* νομισματική μονάδα όταν είναι φτωχός παρά όταν είναι πλούσιος. Αυτή η παρατήρηση καθορίζει ότι, για ορισμένους λήπτες απόφασης, η ωφέλεια μπορεί να είναι μη γραμμική συνάρτηση της χρηματικής επιστροφής. Γενικότερα, η ωφέλεια (*utility payoff*) ενός ατόμου μπορεί να εξαρτάται από πολλαπλές μεταβλητές και όχι μόνο την χρηματική αξία.

Όταν υπάρχει αβεβαιότητα, οι αναμενόμενες ωφέλειες μπορεί να καθοριστούν και να υπολογιστούν μόνο αν έχουν ανατεθεί πιθανότητες στα αβέβαια ενδεχόμενα συμβάντα. Οι πιθανότητες αυτές προσδιορίζουν ποσοτικά την πιθανότητα (*likelihood*) ενός συμβάντος.

Η πρώτη εργασία στην περιοχή της Θεωρίας Παιγνίων αποδίδεται στον Γάλλο μαθηματικό Emil Borel ο οποίος πραγματεύτηκε την Θεωρία των Παιγνίων Στρατηγικής (*Theory of Games of Strategy*). Η εργασία των von Neumann και Morgenstern βασίστηκε στα αποτελέσματα του Borel.

Οι λογικές ρίζες της Θεωρίας Παιγνίων εντοπίζονται στην Bayesian Θεωρία Αποφάσεων (*Decision Theory*). Η Θεωρία Παιγνίων μπορεί να θεωρηθεί ως επέκταση της Θεωρίας Αποφάσεων (για δύο ή περισσότερους λήπτες αποφάσεων), ή ως η

αναγκαία λογική εκπλήρωση της. Τα βασικά αποτελέσματα της Θεωρίας Αποφάσεων αναφέρονται στο ερώτημα αν ένα απλό ποσοτικό μοντέλο μπορεί να δώσει μία λογική περιγραφή της ανθρώπινης συμπεριφοράς. Δείχνουν ότι ο οποιοσδήποτε λήπτης αποφάσεων, που ικανοποιεί ορισμένα αξιώματα, θα συμπεριφέρεται έτσι ώστε να μεγιστοποιεί την μαθηματική αναμενόμενη τιμή μίας συνάρτησης ωφέλειας (utility function), αναφορικά προς μία υποκειμενική πιθανοτική κατανομή. Δηλαδή, η συμπεριφορά ενός λογικού λήπτη απόφασης θα πρέπει να μπορεί να περιγραφεί από μία συνάρτηση ωφέλειας που δίνει ένα ποσοτικό χαρακτηρισμό των προτιμήσεων του αναφορικά με αποτελέσματα-επιβραβεύσεις, και μία υποκειμενική πιθανοτική κατανομή που υποδηλώνει τις πεποιθήσεις του για όλους τους σχετικούς αστάθμητους παράγοντες. Όταν νέες πληροφορίες καθίστανται διαθέσιμες στον λήπτη της απόφασης, οι υποκειμενικές πιθανότητες αναθεωρούνται βάσει του νόμου Bayes.

Οι αποφάσεις κάτω από αβεβαιότητα γενικά περιγράφονται από δύο μοντέλα: ένα πιθανοτικό μοντέλο (probability model) και ένα μοντέλο καταστάσεων – μεταβλητών (state variable model). Σε κάθε περίπτωση θεωρείται ότι ο λήπτης απόφασης επιλέγει μεταξύ διαφόρων λοταριών, αλλά τα δύο μοντέλα διαφέρουν στον ορισμό των λοταριών. Στο πιθανοκρατικό μοντέλο οι λοταρίες είναι κατανομές πιθανότητας πάνω σε ένα σύνολο τιμών. Στο μοντέλο καταστάσεων μεταβλητών οι λοταρίες είναι συναρτήσεις από ένα σύνολο από δυνατές καταστάσεις σε ένα σύνολο τιμών. Το κάθε ένα από τα μοντέλα αυτά είναι κατάλληλο για μία συγκεκριμένη οικογένεια εφαρμογών.

3.2 Συνεργασία

Η έννοια της συνεργασίας (cooperation) είναι βασική στην θεωρία παιγνίων αλλά λίγο «λεπτή». Ο όρος συνεργάζομαι σημαίνει «να ενεργώ μαζί, με ένα κοινό στόχο». Μπορούμε να υποθέσουμε ότι για μία συμμαχία δύο ή περισσότερων ατόμων η κοινή δράση με κοινούς στόχους σημαίνει ότι τα άτομα αυτά παραμερίζουν τις προσωπικές, διακριτές συναρτήσεις χρησιμότητας/ωφέλειας και διαμορφώνουν κάτι τελείως νέο, μία συλλογική συνάρτηση χρησιμότητας μέσω της οποίας καθορίζεται η συλλογική συμπεριφορά. Αυτή η εννοιολογική προσέγγιση δεν είναι απόλυτα συμβατή με την θεωρία παιγνίων όπου υπάρχει η υπόθεση για ευφυείς και λογικούς λήπτες αποφάσεων των οποίων η συμπεριφορά καθορίζεται αποκλειστικά και μόνο από την προσπάθεια μεγιστοποίησης της συνάρτησης χρησιμότητας τους. Απαιτείται έτσι ένα μοντέλο συνεργατικής συμπεριφοράς το οποίο όμως δεν θα πρέπει να εγκαταλείπει και τις επιμέρους αρχές της θεωρίας παιγνίων περί λήψεως αποφάσεων.

Ο Nash το 1951 πρότεινε η συνεργασία μεταξύ παικτών να μελετάται με την βασική έννοια της ισορροπίας Nash η οποία διέπει όλες τις περιοχές της μη-συνεργατική (non-cooperative) θεωρίας παιγνίων. Σύμφωνα με τον Nash, οι συνεργατικές ενέργειες είναι το αποτέλεσμα κάποια μορφής διαδικασίας διαπραγμάτευσης (*bargaining*) μεταξύ των συνεργαζόμενων παικτών. Στο πλαίσιο αυτής της διαδικασίας διαπραγμάτευσης ο κάθε παίκτης αναμένεται ότι θα συμπεριφέρεται σύμφωνα με κάποια στρατηγική η οποία μεγιστοποιεί το ίδιο κριτήριο της μεγιστοποίησης της προσωπικής ωφέλειας όπως σε κάθε άλλη μορφή παιχνιδιού. Δηλαδή, σε κάθε πραγματική κατάσταση, είναι δυνατή η μοντελοποίηση των ενεργειών των παικτών για την επίτευξη συμφωνίας σε μία κοινή, συνεργατική στρατηγική σε ένα παιχνίδι σε εκτεταμένη μορφή και εν συνεχεία, η πρόβλεψη του αποτελέσματος με την ανάλυση των καταστάσεων ισορροπίας.

Είναι δυνατός ο ορισμός ενός συνεργατικού μετασχηματισμού (cooperative transformation), της απεικόνισης Ψ ενός παιχνιδιού Γ σε μία άλλη μορφή $\Psi(\Gamma)$ η οποία αναπαριστά την κατάσταση όπου ο κάθε παίκτης, συμπληρωματικά προς τις στρατηγικές επιλογές τις οποίες είχε στο Γ , διαθέτει μία ευρεία γκάμα επιλογών για διαπραγμάτευση με τους άλλους παίκτες και τον κοινό προγραμματισμό συνεργατικών στρατηγικών. Σε καταστάσεις όπου η λογική δομή των ανεξάρτητων εναλλακτικών επιλογών επιτρέπει μόνο ισορροπίες που είναι δυσμενείς για όλους του παίκτες, οι παίκτες μπορεί να έχουν ένα ισχυρό κίνητρο ώστε να μετασχηματίσουν την δομή του παιχνιδιού προσθέτοντας την δυνατότητα επικοινωνίας (communication) ή σύναψης συμβολαίου (contract signing). Οι δυνατότητες αυτές δίνουν στον κάθε παίκτη, σε κάποιο βαθμό, την δυνατότητα ελέγχου του χώρου αποφάσεων των υπολοίπων. Τέτοιοι συνεργατικοί μετασχηματισμοί μπορούν να μετασχηματίσουν ένα παιχνίδι με πενιχρές ισορροπίες, σε ένα παιχνίδι ισορροπιών που είναι καλύτερες για όλους του παίκτες. Το Nash πρόγραμμα για την συνεργατική θεωρία παιγνίων καθορίζει έννοιες συνεργατικών λύσεων τέτοιες ώστε μία συνεργατική λύση για κάθε δοσμένο παιχνίδι είναι ισορροπία Nash για κάποιο συνεργατικό μετασχηματισμό του παιχνιδιού.

Είναι γενικά αποδεκτό ότι εάν περιγραφούν πλήρως όλες οι δυνατές ενέργειες στις οποίες μπορούν να προβούν οι εμπλεκόμενοι σε μία διαπραγμάτευση ή υπογραφή συμβολαίου, το παιχνίδι στο οποίο θα οδηγηθούμε θα έχει ένα πολύ μεγάλο αριθμό καταστάσεων ισορροπίας. Το πρόγραμμα Nash θα αποτύχει να προσδιορίσει μία μοναδική συνεργατική λύση. Για να αναδειχθούν πιο αυστηρά και περιορισμένα πλαίσια συνεργατικών λύσεων, απαιτείται μία θεωρία για την επιλογή των καταστάσεων

συνεργατικής ισορροπίας (cooperative equilibrium selection theory).

Η γενικότερη αντιμετώπιση ενός προβλήματος πολλαπλών καταστάσεων ισορροπίας είναι το φαινόμενο του εστιακού σημείου (focal point effect). Σύμφωνα με την θεωρία αυτή, σε ένα πρόβλημα με πολλαπλές καταστάσεις ισορροπίας οτιδήποτε εστιάζει την προσοχή των παιχτών σε μία συγκεκριμένη κατάσταση ισορροπίας, με ένα κοινά αποδεκτό τρόπο, τείνει να αναδείξει αυτή την κατάσταση στην κατάσταση ισορροπίας που αναμένουν και επιδιώκουν οι παίχτες. Η εστιακή ισορροπία (focal equilibrium) καθορίζεται από ένα μεγάλο αριθμό παραγόντων, ειδικές μαθηματικές ιδιότητες των καταστάσεων ισορροπίας καθώς και εκ των προτέρων δηλώσεις (pre-play statements) ή έναν εξωτερικό διαιτητή. Μία δυνατή ερμηνεία της υπόθεσης συνεργασίας μεταξύ των παικτών ενός παιχνιδιού είναι η χρήση εκ των προτέρων συνεννοήσεων για τον συντονισμό των αναμενόμενων αποτελεσμάτων προς την εστιακή ισορροπία η οποία έχει θετικά χαρακτηριστικά για μερικούς από αυτούς ή όλους. Έτσι, οι βάσεις της συνεργατικής θεωρίας παιγνίων εντοπίζονται στον ρόλο της διαιτησίας, της διαπραγμάτευσης καθώς και στις θετικές επιστροφές μίας εστιακής ισορροπίας σε παιχνίδια πολλαπλών ισορροπιών.

Ένας εστιακός διαιτητής (focal arbitrator) μπορεί να καθορίσει την εστιακή ισορροπία προτείνοντας στους παίκτες, δημόσια, την ισορροπία την οποία θα πρέπει να επιδιώξουν. Η πρόταση αυτή δεν είναι δεσμευτικής φύσεως αλλά εάν ο κάθε παίκτης πιστεύει ότι οι υπόλοιποι θα ακολουθήσουν την πρόταση του διαιτητή, τότε ο κάθε παίκτης θα προτιμήσει να πράξει σύμφωνα με τις κατευθύνσεις του διαιτητή οδηγούμενος στην κατάσταση ισορροπίας. Επίσης, θα πρέπει να αποτελεί κοινή γνώση μεταξύ των παικτών ότι λόγω του κύρους και της εξουσίας του διαιτητή, όλοι οι παίκτες θα κινηθούν προς την ισορροπία την οποία προτείνει. Ένας αμερόληπτος διαιτητής θα πρέπει να βασίσει τις επιλογές του σε κάποιες μορφής αντικειμενικά κριτήρια. Το ερώτημα δηλαδή είναι, ποια ισορροπία θα πρέπει να προτείνει ένας ιδανικός, αμερόληπτος διαιτητής σε κάθε δοθέν παιχνίδι, εάν η επιλογή του βασίζεται σε κριτήρια που αντιμετωπίζουν τους παίκτες συμμετρικά και εξαρτώνται μόνο από την δομή του παιχνιδιού σύμφωνα με την θεωρία λήψης αποφάσεων. Γενικά, όταν οι παίκτες μπορούν να προσδιορίσουν μία μοναδική ισορροπία η οποία θα αποτελούσε και επιλογή ενός αμερόληπτου διαιτητή, η ισορροπία αυτή μπορεί να γίνει εστιακή ακριβώς λόγω αυτής της ιδιότητας της. Δηλαδή τα χαρακτηριστικά της αποδοτικότητας (efficiency) και δικαιοσύνης (equity) μπορούν να καθορίσουν μία εστιακή ισορροπία

ανεξάρτητα με το αν υπάρχει διαιτητής ή όχι.

Εάν ο εστιακός διαιτητής είναι παίκτης στο ίδιο παιχνίδι, καλείται κύριος ή αρχηγός (principal) του παιχνιδιού. Εάν κάποιος παίκτης είναι αρχηγός με πλήρη πληροφορία, η εστιακή ισορροπία θα είναι μία ισορροπία που επιφέρει τα υψηλότερα ωφέλη στον συγκεκριμένο αρχηγό. Μία άλλη δυνατότητα είναι ο καθορισμός της εστιακής ισορροπίας μέσω μιας διαδικασίας εκ των προτέρων συνεννοήσεων (pre-play communication). Μέσω της διαδικασίας αυτής δύο ή περισσότεροι παίκτες έχουν την δυνατότητα να εκφράσουν την υποστήριξη τους για κάποια ισορροπία. Η διαδικασία εκ των προτέρων συνεννοήσεων που επηρεάζει την επιλογή κάποιας ισορροπίας καλείται εστιακή διαπραγμάτευση (focal negotiation).

3.3 Λύση διαπραγμάτευσης του Nash

Η λύση διαπραγμάτευσης του Nash (Nash bargaining solution) για παιχνίδια δύο παικτών βασίζεται στην παρακάτω υπόθεση. Εάν δύο παίκτες διαπραγματεύονται ή ένας αμερόληπτος διαιτητής διαιτητεύει, τα οφέλη (payoff allocations) τα οποία θα προκύψουν για τους δύο παίκτες τελικά εξαρτώνται από:

- τα οφέλη τα οποία ανέμεναν αν η διαπραγμάτευση ή η διαιτησία αποτύγχαναν να προσδιορίσουν μία συμβιβαστική λύση, και,
- το σύνολο των δυνατών τιμών ωφέλειας (payoff allocations) που είναι από κοινού δυνατές κατά την εξέλιξη της διαπραγμάτευσης ή της διαιτησίας.

Το πρόβλημα διαπραγμάτευσης δύο ατόμων (2-person bargaining problem) θεωρείται το ζεύγος (F, u) όπου F είναι ένα κυρτό (convex) υποσύνολο του \mathbf{R}^2 , $u=(u_1, u_2)$ ένα διάνυσμα στον \mathbf{R}^2 και το σύνολο

$$F \cap \{(x_1, x_2) \mid x_1 \geq u_1 \text{ and } x_2 \geq u_2\} \quad (\text{Εξ.3.1})$$

είναι μη-κενό και φραγμένο. Το F καθορίζει το σύνολο των δυνατών εκχωρήσεων ωφέλειας (payoff allocations) ή το εφικτό σύνολο (feasible set), ενώ το u καθορίζει το σημείο διαφωνίας (disagreement point). Η υπόθεση ότι το σύνολο F είναι κυρτό μπορεί να δικαιολογηθεί με την υπόθεση ότι οι παίκτες μπορεί να συμφωνήσουν σε κοινές, τυχαίες στρατηγικές (jointly randomized strategies), έτσι ώστε αν οι εκχωρήσεις - αναθέσεις ωφέλειας (utility allocations) $x=(x_1, x_2)$ και $y=(y_1, y_2)$ είναι εφικτές και $0 \leq \theta \leq 1$, τότε η αναμενόμενη ωφέλεια $\theta x+(1-\theta)y$ μπορεί να επιτευχθεί υλοποιώντας το x με πιθανότητα θ και το y με πιθανότητα $(1-\theta)$.

Ένα παιχνίδι διαπραγμάτευσης 2 ατόμων (F, u) καλείται ουσιώδες (essential) αν και μόνο αν υπάρχει τουλάχιστον ένα δυνατό διάνυσμα ανάθεσης (allocation) y στο F που είναι αυστηρά καλύτερο και για τους δύο παίκτες απ' ό,τι το σημείο διαφωνίας (disagreement allocation) u , ισχύει δηλαδή $y_1 > u_1$ και $y_2 > u_2$.

Η θεωρία διαπραγματεύσεων ή διαιτησίας μας επιτρέπει να προσδιορίσουμε για κάθε πρόβλημα διαιτησίας μεταξύ δύο ατόμων, (F, u) ένα διάνυσμα επιστροφών στον \mathbf{R}^2 , το οποίο συμβολίζεται ως $\varphi(F, u)$ και θα επιλεγεί ως αποτέλεσμα της διαπραγμάτευσης ή διαιτησίας στην περίπτωση που το F είναι το σύνολο των δυνατών επιστροφών ωφέλειας και u το σημείο διαφωνίας. Δηλαδή, το πρόβλημα εντοπίζεται στον σχεδιασμό μίας συνάρτησης επίλυσης φ από το σύνολο όλων των προβλημάτων διαπραγμάτευσης δύο ατόμων στον χώρο \mathbf{R}^2 δηλαδή το σύνολο των δυνατών επιστροφών ωφέλειας.

Ο Nash προσέγγισε αυτό το πρόβλημα αξιωματικά. Δηλαδή διαμόρφωσε ένα σύνολο από ιδιότητες τις οποίες θα πρέπει να πληρούν οι λύσεις των προβλημάτων διαπραγματεύσεων. Ο συμβολισμός $\varphi_i(F, u)$ αναπαριστά το i -οστό στοιχείο της $\varphi(F, u)$ δηλαδή:

$$\varphi(F, u) = (\varphi_1(F, u), \varphi_2(F, u)) \quad (\text{Εξ.3.2})$$

Επίσης, για οποιαδήποτε δύο διανύσματα x, y στον χώρο \mathbf{R}^2 μπορούμε να γράψουμε:

$x \geq y$ iff $x_1 \geq y_1$ and $x_2 \geq y_2$ και

$x > y$ iff $x_1 > y_1$ and $x_2 > y_2$

Τα αξιώματα για την λύση του Nash στο πρόβλημα της διαπραγμάτευσης είναι τα ακόλουθα:

Αξίωμα 1: Strong Efficiency

Η $\varphi(F, u)$ είναι μία δυνατή επιστροφή ωφέλειας στο F , και για κάθε x στο F , εάν $x \geq \varphi(F, u)$, τότε $x = \varphi(F, u)$.

Αξίωμα 2: Individual Rationality

$$\varphi(F, u) \geq u$$

Αξίωμα 3: Scale Convergence

Για οποιουδήποτε $\lambda_1, \lambda_2, \gamma_1$ και γ_2 τέτοιους ώστε $\lambda_1 > 0$ και $\lambda_2 > 0$, εάν

Σχήμα Προφόρτωσης Περιεχομένου WWW

$$\Gamma = \{(\lambda_1 x_1 + \gamma_1, \lambda_2 x_2 + \gamma_2) \mid (x_1, x_2) \in F\}$$

Και $w = (\lambda_1 u_1 + \gamma_1, \lambda_2 u_2 + \gamma_2)$, τότε

$$\varphi(G, w) = (\lambda_1 \varphi_1(F, u) + \gamma_1, \lambda_2 \varphi_2(F, u) + \gamma_2)$$

Αξίωμα 4: Independence of Irrelevant Alternatives

Για οποιοδήποτε κλειστό, κυρτό σύνολο G , εάν το $G \subseteq F$ και $\varphi(F, u) \in G$, τότε $\varphi(G, u) = \varphi(F, u)$

Αξίωμα 5: Symmetry

Εάν $u_1 = u_2$ και $\{(x_2, x_1) \mid (x_1, x_2) \in F\} = F$ τότε $\varphi_1(F, u) = \varphi_2(F, u)$

Το αξίωμα 1 εξασφαλίζει ότι η λύση σε κάθε πρόβλημα διαπραγμάτευσης 2 ατόμων θα πρέπει να είναι εφικτή (feasible) και Pareto αποδοτική (pareto efficient). Δηλαδή δεν θα πρέπει να υπάρχει άλλος δυνατός συνδυασμός εκχωρήσεων-αναθέσεων (allocation) που να είναι καλύτερος από την λύση για τον ένα παίκτη και όχι χειρότερος από την λύση για τον άλλον παίκτη. Γενικά, δεδομένου ενός εφικτού συνόλου F , ένα σημείο x θεωρείται strongly Pareto efficient τότε και μόνο τότε αν δεν υπάρχει άλλο σημείο y στο F τέτοιο ώστε $y \geq x$ και $y_i > x_i$ για τουλάχιστον ένα παίκτη i . Σε αντιδιαστολή, ένα σημείο x του F καλείται weakly Pareto efficient τότε και μόνο τότε αν δεν υπάρχει σημείο y στο F τέτοιο ώστε $y > x$.

Το αξίωμα 2 εξασφαλίζει ότι κανένας από τους δύο παίκτες δεν πρόκειται να λάβει λιγότερα απ' ό,τι στο σημείο διαφωνίας. Ένας συνδυασμός αναθέσεων (allocation) x του F καλείται ανεξάρτητα λογικός (individually rational) τότε και μόνο τότε αν $x \geq u$.

Το αξίωμα 3 καθορίζει ότι εάν ένα πρόβλημα διαπραγμάτευσης 2 ατόμων (G, u) μπορεί να παραχθεί από το πρόβλημα διαπραγμάτευσης (F, u) με συσχετιζόμενο (affine) μετασχηματισμό (αύξηση) των συναρτήσεων ωφέλειας (ο μετασχηματισμός αυτός δεν θα επηρεάσει τα παιγνιοθεωρητικά χαρακτηριστικά των συναρτήσεων), η λύση του (G, u) μπορεί να παραχθεί από την λύση (F, u) με τον ίδιο μετασχηματισμό. Δηλαδή, εάν μεταβληθεί ο τρόπος μέτρησης της ωφέλειας κατά την διαμόρφωση ενός 2-person παιχνιδιού διαπραγμάτευσης αλλά οι νέες κλίμακες ωφέλειας είναι παιγνιοθεωρητικά ισοδύναμες με τις παλιές, η διαπραγματευτική λύση στον χώρο των αναθέσεων ωφέλειας θα πρέπει να μεταβληθεί τελείως ανάλογα.

Το αξίωμα 4 καθορίζει ότι η απαλοιφή των εφικτών εναλλακτικών λύσεων (εκτός του σημείου διαφωνίας) τα οποία δεν πρόκειται να επιλεγούν, δεν πρόκειται να επηρεάσει

την λύση. Εάν κάποιος διαιτητής επιλέξει μία λύση για την μεγιστοποίηση ενός συνολικού (aggregate) μεγέθους κοινωνικής ωφέλειας (social gain) (δηλαδή $\{\phi(F,u)\} = \operatorname{argmax}_{x \in F} M(x,u)$, όπου $M(x,u)$ είναι το μέτρο κοινωνικής ωφέλειας από την επιλογή της ανάθεσης x αντί της u) τότε το αξίωμα 4 θα ικανοποιείται πάντα.

Το αξίωμα 5 καθορίζει ότι εάν οι θέσεις των παικτών 1 και 2 είναι τελείως συμμετρικές στο πρόβλημα διαπραγμάτευσης, τότε η λύση θα πρέπει επίσης να τους χειρίζεται συμμετρικά.

Το σημαντικό αποτέλεσμα της εργασίας του Nash είναι ότι υπάρχει μία ακριβώς διαπραγματευτική λύση, η οποία ονομάζεται λύση διαπραγμάτευσης του Nash (Nash bargaining solution) η οποία ικανοποιεί τα παρακάτω αξιώματα.

Υπάρχει μία μοναδική συνάρτηση λύσης $\phi(\cdot, \cdot)$ η οποία ικανοποιεί τα αξιώματα 1 – 5. Αυτή η συνάρτηση επίσης ικανοποιεί την σχέση

$$\phi(F, u) \in \operatorname{argmax}_{x \in F, x \geq u} (x_1 - u_1)(x_2 - u_2) \quad (\text{Εξ.3.3})$$

για κάθε παιχνίδι διαπραγμάτευσης 2 ατόμων (F, u) .

3.4 Απόδειξη

Εάν (F, u) είναι ένα ουσιώδες (essential) παιχνίδι διαπραγμάτευσης 2 ατόμων και υπάρχει κάποιο y στο σύνολο F τέτοιο ώστε $y_1 > x_1$ και $y_2 > x_2$. Έστω x το μοναδικό σημείο στο F το οποίο επιτυγχάνει το μέγιστο της συνάρτησης $(x_1 - u_1)(x_2 - u_2)$, η οποία επίσης καλείται και γινόμενο Nash (Nash product), για όλα τα x στο F για τα οποία ισχύει $x \geq u$. Αυτό το σημείο θα πρέπει να ικανοποιεί την $x > u$ για να επιτύχει τις αυστηρά θετικές τιμές του γινομένου Nash οι οποίες είναι δυνατές στο σύνολο F . Δεδομένων των F και u , αυτό το σημείο μεγιστοποίησης είναι μοναδικό αφού η μεγιστοποίηση του γινομένου Nash είναι ισοδύναμη με την μεγιστοποίηση του λογαρίθμου του, ο οποίος είναι αυστηρά κυρτή (concave) συνάρτηση του x .

Έστω $\lambda_i = 1/(x_i - u_i)$ και $\gamma_i = -u_i/(x_i - u_i)$ για κάθε i . Ορίζουμε την συνάρτηση $L: \mathbf{R}^2 \rightarrow \mathbf{R}^2$ τέτοια ώστε: $L(y) = (\lambda_1 y_1 + \gamma_1, \lambda_2 y_2 + \gamma_2)$

Και $G = \{L(y) \mid y \in F\}$. Για κάθε y στο \mathbf{R}^2 , εάν $z = L(y)$, τότε $z_1 z_2 = \lambda_1 \lambda_2 (y_1 - u_1)(y_2 - u_2)$ και $\lambda_1 \lambda_2$ είναι μία θετική σταθερά. Επειδή το x μεγιστοποιεί το γινόμενο Nash στο σύνολο F , η συνάρτηση $L(x)$ θα πρέπει να μεγιστοποιεί το γινόμενο $z_1 z_2$ για όλα τα z στο σύνολο G . Αλλά $L(x) = (1, 1)$ και η υπερβολή $\{z \in \mathbf{R}^2 \mid z_1 z_2 = 2\}$ έχει κλήση -1 στο σημείο $(1, 1)$. Η

Σχήμα Προφόρτωσης Περιεχομένου WWW

ευθεία $\{z \mid z_1 + z_2 = 2\}$ που έχει κλήση -1 και διέρχεται από το σημείο $(1,1)$ πρέπει να είναι από πάνω και εφαπτόμενη στο κυρτό σύνολο G στο σημείο $(1,1)$. Έστω $E = \{z \in \mathbf{R}^2 \mid z_1 + z_2 \leq 2\}$. Τότε το σύνολο G είναι περιέχεται στο ημιεπίπεδο E ($G \subseteq E$).

Για να πληρούνται τα Αξιώματα 1 και 5 θα πρέπει να ισχύει: $\varphi(E, (0,0)) = (1,1)$

Για να πληρείται το Αξίωμα 4 πρέπει να ισχύει: $\varphi(G, (0,0)) = (1,1)$

Το αξίωμα 3 απαιτεί $L(\varphi(F,u)) = \varphi(G,(0,0))$, οπότε: $\varphi(F,u) = x$

Για να ικανοποιήσει τα αξιώματα η φ θα πρέπει να επιλέξει εκείνη την ανάθεση-εκχώρηση η οποία μεγιστοποιεί το γινόμενο Nash μεταξύ όλων των μεμονωμένα λογικών αναθέσεων στο σύνολο F .

Αν υποθεθεί ότι το (F, u) είναι μη ουσιώδες (inessential) παιχνίδι, με το σκεπτικό ότι δεν υπάρχει κάποιο σημείο y στο F τέτοιο ώστε $y > u$. Από την κυρτότητα του F , θα πρέπει να υπάρχει τουλάχιστον ένας παίκτης i τέτοιος ώστε, για κάθε y στο F , εάν $y \geq u$, τότε $y_i = u_i$. Έστω x η ανάθεση στο F η οποία είναι καλύτερη για το παίκτη διαφορετικό του i , υποκείμενη στον περιορισμό $x_i = u_i$. Τότε το διάνυσμα x είναι το μοναδικό σημείο το οποίο είναι ισχυρά Pareto efficient στο F καθώς και ανεξάρτητα –μεμονωμένα λογικό σε σχέση με το u . Για την ικανοποίηση των Αξιωμάτων 1 και 2 θα πρέπει να ισχύει $\varphi(F, u) = x$. Προφανώς, το x μεγιστοποιεί την τιμή του Nash γινομένου η οποία είναι 0 για όλες τις μεμονωμένα λογικές αναθέσεις στο μη ουσιώδες (inessential) πρόβλημα διαπραγμάτευσης.

Με βάση τα παραπάνω, δείχθηκε ότι τα πέντε αξιώματα ικανοποιούνται από μία και μοναδική συνάρτηση φ , η οποία πάντα επιλέγει την μοναδική, ισχυρά αποδοτική ανάθεση που μεγιστοποιεί το γινόμενο Nash.

ΚΕΦΑΛΑΙΟ: 4

ΠΑΙΓΝΙΟΘΕΩΡΗΤΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΣΕ ΔΙΚΤΥΑΚΑ ΠΡΟΒΛΗΜΑΤΑ

4.1 Εκχώρηση εύρους ζώνης σε εικονικά μονοπάτια για δίκτυα πολλαπλών χρηστών

Μία άλλη σημαντική εργασία στον χώρο της μελέτης δικτυακών προβλημάτων με εφαρμογή παιγνιοθεωρητικών αρχών είναι η [6]. Η εργασία αυτή ασχολείται με το πρόβλημα του επιμερισμού του διαθέσιμου εύρους ζώνης ενός εικονικού μονοπατιού ATM (ATM virtual path) μεταξύ διαφόρων χρηστών. Με την παραδοσιακή τακτική, η λύση του παραπάνω προβλήματος αποτελεί αντικείμενο μίας και μοναδικής διαχειριστικής οντότητας. Οι χρήστες δηλαδή είναι παθητικοί και συχνά υποβιβάζεται η δική τους επίδοση προς όφελος του δικτύου. Η εφαρμογή ενιαίας διαχείρισης έχει κριθεί αναποτελεσματική για μεγάλα ευρυζωνικά δίκτυα. Το δίκτυο μπορεί να θεωρηθεί ως ένας διαμοιραζόμενος πόρος για τον οποίο ανταγωνίζονται εγωιστικά σκεπτόμενοι χρήστες (selfish). Αυτό αποτελεί και το βασικό σενάριο για την θεώρηση ενός παιχνιδιού.

Μία προφανής επίπτωση του σχήματος virtual path είναι ότι μέρος του διαθέσιμου εύρους ζώνης μπορεί να μείνει αδιάθετο. Μία άλλη επίπτωση είναι ότι πολλές φορές ένας χρήστης μπορεί να διαπιστώσει αδυναμία του συστήματος να δεχθεί νέες κλήσεις (call blocking) λόγω κορεσμού του διαθέσιμου εύρους ζώνης. Σε ένα σενάριο μη συνεργατικού παιχνιδιού, ο κάθε χρήστης μπορεί να προσπαθήσει να δεσμεύσει το μέγιστο δυνατό εύρος ζώνης. Αυτή η «άπληστη» συμπεριφορά σε συνδυασμό με το πεπερασμένο χαρακτήρα του εύρους ζώνης μπορεί να οδηγήσει σε ανεπιθύμητα αποτελέσματα (π.χ., αστάθεια και αδικία).

Όπως συνηθίζεται στην Θεωρία Παιγνίων θα πρέπει να επιβληθεί μία συγκεκριμένη αλληλεπίδραση μεταξύ των παικτών στην βάση των επιμέρους στρατηγικών τους. Αυτό μπορεί να επιτευχθεί με την υιοθέτηση μίας συγκεκριμένης συνάρτησης κόστους (cost function) η οποία εξαρτάται από το διαθέσιμο εύρος ζώνης. Για την μοντελοποίηση του συστήματος εξετάζεται ένα σύνολο N χρηστών που μοιράζονται τον δικτυακό πόρο συνολικού εύρους ζώνης B . Ο κάθε χρήστης δεσμεύει τμήμα του πόρου για να εγκαταστήσει ένα εικονικό μονοπάτι για τις εισερχόμενες κλήσεις. Οι χρήστες δεν συνεργάζονται μεταξύ τους (non-cooperative) και ο καθένας προσπαθεί να

βελτιστοποιήσει την επίδοση του. Στο ίδιο πλαίσιο προσπαθεί να ελαχιστοποιήσει την συνάρτηση κόστους. Αυτή η συνάρτηση θα πρέπει να ακολουθεί την λογική που περιγράφεται παρακάτω. Ο κάθε χρήστης θα πρέπει να ελαχιστοποιεί τη blocking probability των εισερχομένων κλήσεων στο επίπεδο του εικονικού μονοπατιού. Αυτή η πιθανότητα είναι φθίνουσα συνάρτηση της χωρητικότητας που έχει δεσμευτεί στα virtual paths του χρήστη. Η δέσμευση χωρητικότητας όμως γίνεται σταδιακά πιο δύσκολη (και ακριβότερη) όσο περιορίζεται η διαθεσιμότητα των πόρων του συστήματος. Οι παραπάνω, αντικρουόμενες λογικές μπορούν να ποσοτικοποιηθούν όπως περιγράφεται παρακάτω. Έστω C_i η στρατηγική του παίκτη i δηλαδή το ποσό των πόρων (εύρους ζώνης) που δεσμεύει ο χρήστης i . Μπορεί να θεωρηθεί, χωρίς απώλεια της γενικότητας, ότι ένας χρήστης λαμβάνει πάντα την ποσότητα των πόρων την οποία αιτείται. Ο χώρος στρατηγικών (strategy space) του χρήστη i είναι το διάστημα $[0, B]$. Έστω $C = (C_1, C_2, \dots, C_N)$ το διάνυσμα στρατηγικών (game strategy vector). Επίσης, το C υποδηλώνει το συνολικό ποσό δεσμευμένης χωρητικότητας, δηλαδή $C = \sum_i C_i$. Η συνάρτηση κόστους για τον χρήστη i είναι η J_i η μορφή της οποίας είναι η ακόλουθη:

$$J_i(C) = J_i(C_i, C) = F_i(C_i, C) + G_i(C_i) \quad (\text{Εξ.4.1})$$

Η συνάρτηση F_i υποδηλώνει την διαθεσιμότητα των πόρων όπως την αντιλαμβάνεται ο χρήστης i . Η συνάρτηση G_i υποδηλώνει την επίδραση που έχει στις επιδόσεις του συγκεκριμένου χρήστη η δεσμευμένη χωρητικότητα. Οι παραπάνω συναρτήσεις θεωρείται, επίσης, ότι έχουν τα εξής χαρακτηριστικά:

- Η $F_i(\cdot, \cdot)$ αυξάνεται μονότονα με κάθε ένα από τα δύο ορίσματα της (ιδιότητα F1).
- Η $F_i(C_i, C)$ είναι διαρκώς παραγωγίσιμη αναφορικά με το C_i (ιδιότητα F2).
- Η $F_i(C_i, C)$ είναι αυστηρά κυρτή (strictly convex) στο C_i (ιδιότητα F3).
- Η μερική παράγωγος $\partial F_i(C_i, C) / \partial C_i$ είναι μη-φθίνουσα αναφορικά με το C και επίσης αυστηρά αύξουσα αναφορικά με το C_i (ιδιότητα F4).
- Το όριο $\lim_{C_i \rightarrow B} F_i(C_i, C) = \infty$ (ιδιότητα F5).
- Η $G_i(C_i)$ είναι διαρκώς παραγωγίσιμη (ιδιότητα G1).
- Η $G_i(C_i)$ είναι αυστηρά φθίνουσα (ιδιότητα G2).
- Η $G_i(C_i)$ είναι κυρτή (ιδιότητα G3).

- Το όριο $\lim_{C_i \rightarrow 0} G_i(C_i) = \infty$ (ιδιότητα G4).

Η συνάρτηση F_i αυξάνεται με τις αιτήσεις για δέσμευση εύρους ζώνης από τον χρήστη i καθώς και με το συνολικό όγκο δεσμεύσεων. Η ιδιότητα F5 επιβάλλει ότι ο κάθε χρήστης δεν μπορεί να αναλώσει μόνος του το συνολικό εύρος ζώνης. Η συνάρτηση G_i μειώνεται με την C_i δεικνύοντας έτσι την βελτίωση στην επίδοση του χρήστη i . Η G4 δηλώνει ότι ο κάθε χρήστης έχει ανάγκη από κάποιο όγκο δεσμεύσεων.

Εφόσον η συνάρτηση κόστους του κάθε χρήστη εξαρτάται από την στρατηγική όλων των υπολοίπων, το όλο σύστημα μπορεί να θεωρηθεί ως μη-συνεργατικό παιχνίδι. Η λύση ισορροπίας Nash (NE, Nash Equilibrium) είναι ένα διάνυσμα στρατηγικών του παιχνιδιού από το οποίο κανείς παίκτης δεν ενδιαφέρεται να αποκλίνει. Με δεδομένες τις ιδιότητες (υποθέσεις) οι οποίες αναλύθηκαν παραπάνω μπορεί να εξασφαλιστεί η ύπαρξη NE για το συγκεκριμένο πρόβλημα. Επίσης αποδεικνύεται η ύπαρξη μοναδικού NE. Επίσης, αποδεικνύεται ότι η κατάσταση ισορροπίας είναι «δίκαια» (*fairness property*) για τους εμπλεκόμενους παίκτες/χρήστες. Το χαρακτηριστικό αυτό εγγυάται ότι όποιος χρήστης έχει περισσότερη ανάγκη από τον διαμοιραζόμενο πόρο, θα λάβει περισσότερο εύρος ζώνης στην κατάσταση ισορροπίας.

Στην ίδια εργασία μελετάται η δυναμική συμπεριφορά του παραπάνω μηχανισμού. Αποδεικνύεται η επίτευξη της σύγκλισης στην κατάσταση ισορροπίας κάτω από τα επαναληπτικά σχήματα Gauss-Seidel και Jacobi. Μέσω των σχημάτων αυτών, οι χρήστες επανυπολογίζουν την χωρητικότητα την οποία έχουν δεσμεύσει και βελτιστοποιούν τις συναρτήσεις κόστους τους σε κάθε κατάσταση του συστήματος.

Μέσω των πειραμάτων τα οποία παρουσιάζονται στην εργασία [6] μελετάται η σχέση (εξάρτηση) της σύγκλισης του προτεινόμενου σχήματος από την συνάρτηση κόστους. Η μελετούμενη μορφή συνάρτηση κόστους είναι η ακόλουθη:

$$F_i(C_i, C) = C_i \left(a_1 + \frac{a_2}{\left(1 - \frac{C}{B}\right)^n} \right) \quad (\text{Εξ.4.2})$$

Οι παράγοντες a_1 , a_2 και n είναι θετικοί πραγματικοί αριθμοί. Ο παράγοντας a_1 υποδηλώνει ένα σταθερό κόστος ανά μονάδα χωρητικότητας. Ο παράγοντας a_2 υποδηλώνει ένα μοναδιαίο κόστος οφειλόμενο σε παράγοντες όπως η δικτυακή συμφόρηση (η συνολική δεσμευμένη χωρητικότητα C τείνει προς την χωρητικότητα του

καναλιού B). Η παράμετρος n δεικνύει πόσο νωρίς ανιχνεύεται η κατάσταση συμφόρησης.

Για την συνάρτηση G_i χρησιμοποιήθηκε η συνάρτηση απώλειας (loss function) Erlang-B. Η διαδικασία άφιξης/γέννησης κλήσεων του χρήστη i θεωρείται Poisson με ρυθμό α_i . Η Erlang-B formula που αντιστοιχεί στον χρήστη i είναι συνάρτηση του α_i και C_i και υποδηλώνεται από τον όρο $E(\alpha_i, C_i)$. Εάν το κ_i είναι το πάνω όριο στο call blocking probability του χρήστη i , η G_i ορίζεται ως εξής:

$$G_i(C_i) = \begin{cases} \frac{1}{\kappa_i - E(\alpha_i, C_i)}, & \text{if } \kappa_i > E(\alpha_i, C_i) \\ \infty, & \text{otherwise} \end{cases} \quad (\text{Εξ.4.3})$$

4.2 Παιγιοθεωρητική ανάλυση των αρχών ελέγχου συμφόρησης

Μία ιδιαίτερα σημαντική εργασία στο χώρο της παιγιοθεωρητικής μελέτης δικτυακών προβλημάτων είναι η [15]. Η εργασία αυτή βασίζεται στην παραδοχή ότι είναι αδύνατο (και δεν ενδείκνυται) να επιβληθεί ένας ενιαίος τρόπος συμπεριφοράς στους τελικούς χρήστες ενός δικτύου. Θεωρείται ότι οι χρήστες συμπεριφέρονται «εγωιστικά» (selfishly) προωθώντας τα προσωπικά τους ενδιαφέροντα. Ο ρόλος του σχεδιαστή του δικτύου περιορίζεται στον καθορισμό της συμπεριφοράς των στοιχείων μεταγωγής του δικτύου (network switches). Ο στόχος της συγκεκριμένης διερεύνησης είναι ο σωστός σχεδιασμός της λογικής λειτουργίας η οποία θα πρέπει να επιβληθεί στα στοιχεία του δικτύου ώστε να επιτευχθούν καλές επιδόσεις παρά την εγωιστική συμπεριφορά των τελικών χρηστών.

Τίθεται ο προβληματισμός ότι η προσέγγιση που δίνει έμφαση στα κίνητρα των χρηστών δεν είναι συμβατή με την συμβατική μελέτη των καταναμημένων συστημάτων. Πώς μπορεί κάποιος να μοντελοποιήσει την συμπεριφορά χρηστών αν αυτοί δεν ακολουθούν ένα αλγόριθμο που τους έχει επιβληθεί συλλογικά; Σε ανάλογα μη-συνεργατικά περιβάλλοντα πως μπορεί να περιγραφεί το τελικό σημείο λειτουργίας του δικτύου; Ο σκοπός της εργασίας [15] είναι να δείξει πως μπορούν να εφαρμοστούν οι αρχές της θεωρίας παιγνίων για να απαντηθούν τα παραπάνω ερωτήματα.

Η παιγιοθεωρητική προσέγγιση εφαρμόζεται στο απλό σύστημα ενός μεταγωγέα (network switch) η δυνατότητα του οποίου μοιράζεται σε N χρήστες. Ο κάθε χρήστης στέλνει κίνηση Poisson προς τον μεταγωγέα με ρυθμό r_i . Ο αλγόριθμος μέσω του

οποίου ρυθμίζεται η παράμετρος r_i καλείται έλεγχος ροής (flow control). Ο μεταγωγέας συμπεριφέρεται ως εκθετικός εξυπηρέτης με pre-emption. Ένα μέτρο της συμφόρησης την οποία υφίσταται κάποιος χρήστης είναι ο μέσος αριθμός των πακέτων του συγκεκριμένου χρήστη τα οποία υπάρχουν στην ουρά του εξυπηρέτη. Αυτό το μέτρο συμφόρησης, ή μέσο μέγεθος ουράς, συμβολίζεται με c_i και εξαρτάται από τους διάφορους ρυθμούς εισόδου στον μεταγωγέα (r_j) καθώς και την πολιτική που αυτός εφαρμόζει. Η ανάλυση βασίζεται στις εξής αρχές:

- 1) η ευχαρίστηση του χρήστη είναι συνάρτηση του ποσού και της ποιότητας υπηρεσίας την οποία παρέχει ο μεταγωγέας. Η πρώτη αρχή αντιμετωπίζεται φορμαλιστικά μέσω συναρτήσεων ωφέλειας (*utility functions*) $U_i(r_i, c_i)$ οι οποίες εκφράζουν την προτίμηση του χρήστη για συγκεκριμένα επίπεδα παροχής της υπηρεσίας.
- 2) Ο κάθε χρήστης συμπεριφέρεται εγωιστικά. Ο κάθε χρήστης εφαρμόζει ένα αλγόριθμο έλεγχου ροής για να μεγιστοποιήσει την ωφέλεια του μεταβάλλοντας τον ρυθμό r_i . Όταν ο κάθε χρήστης συμπεριφέρεται με αυτόν τον τρόπο, το λειτουργικό σημείο του συστήματος αποτελεί σημείο ισορροπίας Nash.
- 3) Η επίδοση του συστήματος εκτιμάται μόνο βάσει του επιπέδου ευχαρίστησης που παρέχει στον χρήστη.
- 4) Ο αλγόριθμος βάσει του οποίου λειτουργεί ο μεταγωγέας τελεί υπό κεντρικό, διαχειριστικό έλεγχο.

Η σχεδιαστική προσπάθεια εστιάζεται στον καθορισμό πολιτικών λειτουργίας του μεταγωγέα ώστε το σύστημα να εμφανίζει καλές επιδόσεις παρά την εγωιστική συμπεριφορά των χρηστών. Η «καλή συμπεριφορά» συνίσταται στο να είναι το σημείο ισορροπίας Nash δίκαιο (fair) και αποδοτικό (efficient). Επίσης, το σημείο ισορροπίας Nash θα πρέπει να είναι γρήγορα και εύκολα προσβάσιμο μέσω απλών τεχνικών βελτιστοποίησης. Τέλος, το σύστημα θα πρέπει να προσφέρει ορισμένες εγγυήσεις επιδόσεων ακόμη και αν λειτουργεί εκτός ισορροπίας.

Η υπόθεση της «εγωιστικής» συμπεριφορά, η οποία αποτελεί βάση για την μη-συνεργατική θεώρηση του συστήματος, αντιμετωπίζεται, γενικά, με σκεπτικισμό. Ο αντίλογος επικεντρώνεται στην εφαρμογή συνεργατικών αλγορίθμων για το πρόβλημα του ελέγχου ροής. Συνεργατικοί αλγόριθμοι όπως απαιτούν a-priori αφαιρετική γνώση των συναρτήσεων ωφέλειας και δυνατότητα σύγκρισης, από την πλευρά των χρηστών,

των εκχωρήσεων (allocations) χωρίς να τις λάβουν στην πραγματικότητα. Αυτή η λογική είναι σημαντικά διαφορετική συμπεριφορά από την ανάβαση λόφου (hill-climbing) όπου οι χρήστες πραγματοποιούν συγκρίσεις τοπικά (local hill climbing) για τις εκχωρήσεις τις οποίες λαμβάνουν στην πραγματικότητα. Το δεύτερο πρόβλημα με την συνεργατική νοοτροπία είναι ότι ένας παγκόσμια αποδεκτός αλγόριθμος ελέγχου ροής παγιδεύει το δίκτυο σε τεχνολογία η οποία πολύ σύντομα θα απαξιωθεί. Τέλος, δεν θα πρέπει να παραβλεφθεί ο κίνδυνος των «εγωιστικά» συμπεριφερόμενων χρηστών και η δυνατότητα προσβολής ενός συνεργατικού συστήματος από αυτούς.

Πέρα από την αναφορά στις βασικές αρχές που διέπουν τα σημεία ισορροπίας Nash, στην [15] μελετάται ο μηχανισμός λειτουργίας των μεταγωγέων που είναι γνωστός ως αναλογική εκχώρηση (proportional allocation) καθώς και ο fair share (δίκαιου μεριδίου). Στον πρώτο μηχανισμό, το μέσο μήκος της ουράς ενός χρήστη είναι ανάλογο του ρυθμού αυτού του χρήστη. Η λογική αυτή απεικονίζεται στην πολιτική FIFO των μεταγωγέων. Στον fair share μηχανισμό η λογική είναι αυτή ενός συστήματος χρονοδρομολόγησης με προτεραιότητες (preemptive priority scheduling). Εάν οι ρυθμοί r_i ταξινομηθούν κατά φθίνουσα σειρά, ο χρήστης με ρυθμό r_1 έχει την μεγαλύτερη προτεραιότητα, ενώ ο ρυθμός του χρήστη με την αμέσως επόμενη προτεραιότητα είναι r_2-r_1 κλπ.

Ο μηχανισμός δίκαιου μοιράσματος (fair share) έχει το χαρακτηριστικό ότι διαθέτει πάντα ένα σημείο ισορροπίας Nash το οποίο επίσης είναι Pareto efficient. Επίσης, η ίδια ισορροπία είναι δίκαιη (fair). Ορίζεται η έννοια της envy-free ισορροπίας δηλαδή της κατάστασης όπου ο κάθε παίκτης δεν «ζηλεύει» την ανάθεση ενός άλλου. Η ισορροπία Nash η οποία επιτυγχάνεται με την fair share λογική είναι δίκαια επειδή διατηρεί το χαρακτηριστικό envy-free. Η λογική αναλογικής εκχώρησης οδηγεί σε ισορροπίες Nash που δεν είναι Pareto efficient και μπορεί να μην είναι δίκαιες. Για την ίδια πολιτική, η σύγκλιση στην κατάσταση ισορροπίας δεν είναι εγγυημένη.

Όλοι οι γενικευμένοι αλγόριθμοι ανάβασης λόφου αν εφαρμοστούν στην περίπτωση fair share οδηγούν σε ισορροπία Nash. Επίσης μελετάται μία παραλλαγή της κατάστασης ισορροπίας η οποία είναι γνωστή ως ισορροπία Stackelberg. Στο παιχνίδι Stackelberg κάποιος από τους χρήστες συμπεριφέρεται κατά τρόπο βελτιωμένο, αφήνοντας τους υπολοίπους χρήστες να φτάσουν σε ισορροπία και επιλέγοντας, εκ των υστέρων, την ενδεδειγμένη για αυτόν στρατηγική. Ο χρήστης αυτός χαρακτηρίζεται αρχηγός (leader). Ο χρήστης, πρακτικά, διαθέτει κάποια πληροφορία για τις συναρτήσεις ωφέλειας των

υπολοίπων. Στην κατάσταση ισορροπίας Stackelberg η ωφέλεια του αρχηγού δεν είναι χαμηλότερη από αυτή στην απλή ισορροπία Nash. Η ισορροπία Nash η οποία επιτυγχάνεται από τον fair share αλγόριθμο είναι επίσης ισορροπία Stackelberg.

4.3 Αρχιτεκτονική μη-συνεργατικών δικτύων

Μία άλλη σημαντική εργασία στον χώρο των παιγνιοθεωρητικών προσεγγίσεων δικτυακών προβλημάτων είναι η [26]. Η εργασία εστιάζει στο πρόβλημα της δρομολόγησης και εισάγει δύο μεθοδολογίες για την συγκρότηση μη συνεργατικών δικτύων όπου οι χρήστες λαμβάνουν αποφάσεις ελέγχου (control decisions) προσπαθώντας να μεγιστοποιήσουν την ωφέλεια τους. Οι μεθοδολογίες αναφέρονται στις φάσεις provisioning του δικτύου καθώς και στην φάση της λειτουργίας (run-time). Κατά την φάση provisioning, διαστασιοποιούνται οι βασικές λειτουργικές παράμετροι του δικτύου.

Σκοπός της μεθοδολογίας στην φάση provisioning είναι η διάθεση της χωρητικότητας των συνδέσεων ώστε το σύστημα να οδηγηθεί σε κατάσταση ισορροπίας Nash (η οποία θα έχει επίσης το χαρακτηριστικό της αποδοτικότητας «efficiency»). Τα κριτήρια για την αποδοτικότητα του συστήματος είναι η τιμή (οριακό κόστος) την οποία αντιλαμβάνεται ο χρήστης, το συνολικό κόστος για τον κάθε χρήστη ή συνδυασμός των ανωτέρω. Η λύση του προβλήματος εκχώρησης χωρητικότητας δεν συμβαδίζει με τα αναμενόμενα αφού η επαύξηση της χωρητικότητας συνδέσεων μπορεί να οδηγήσει σε υποβάθμιση των επιδόσεων των χρηστών. Ο σχεδιαστής δηλαδή δεν θα πρέπει να προσπαθήσει να εκμεταλλευτεί όλους τους διαθέσιμους πόρους. Η επίδειξη αυτής της μη-αναμενόμενης συμπεριφοράς βασίζεται στο παράδοξο του Braess.

Κατά την διάρκεια της φάσης λειτουργίας (run-time) ένας συντονιστής (manager) ελέγχει την δρομολόγηση τμήματος της δικτυακής ροής. Ο συντονιστής γνωρίζει την μη-συνεργατική συμπεριφορά των χρηστών του δικτύου και λαμβάνει αποφάσεις δρομολόγησης βάσει αυτής της συμπεριφοράς, προσπαθώντας πάντα να βελτιώσει την επίδοση του συστήματος. Το σενάριο αυτό είναι ακριβώς το σενάριο ενός παιχνιδιού Stackelberg όπως αυτό συζητήθηκε στην προηγούμενη παράγραφο. Ο συντονιστής-αρχηγός μπορεί να επιβάλλει κατάσταση ισορροπίας η οποία ταυτίζεται με την βέλτιστη κατάσταση του δικτύου (κεντρικός έλεγχος όλων των δικτυακών ροών).

4.4 Παιγνιοθεωρητική κατανομή εύρους ζώνης σε ευρυζωνικά δίκτυα

Τα σημερινά δίκτυα υψηλών ταχυτήτων πρέπει να μπορούν να υποστηρίξουν

εφαρμογές που δεν έχουν τρόπο να προβλέψουν τις μελλοντικές τους ανάγκες σε εύρος ζώνης, αλλά μπορούν να ανεχτούν μεταβολές σε καθυστερήσεις μεταφοράς, έχοντας συγκεκριμένες απαιτήσεις όσον αφορά τις απώλειες. Αυτά τα χαρακτηριστικά σημαίνουν ότι οι πηγές αυτού του είδους μπορούν να αλλάζουν τους ρυθμούς μεταφοράς τους ανάλογα με την κατάσταση του δικτύου. Αυτές οι υπηρεσίες αναφέρονται ως ελαστικές (elastic services). Τυπικές υπηρεσίες αυτού του είδους είναι υπηρεσίες που βασίζονται στο TCP/IP ή στο ABR του ATM. Αυτές οι υπηρεσίες τρέχουν πάνω από άλλες που απαιτούν εγγύηση συγκεκριμένου εύρους ζώνης και έτσι κάθε φορά μοιράζονται το υπολειπόμενο εύρος ζώνης. Από την στιγμή που το ελεύθερο αυτό εύρος ζώνης αλλάζει ανάλογα με τις υπηρεσίες που τρέχουν στο παρασκήνιο, έπεται ότι οι ελαστικές υπηρεσίες (elastic services) θα πρέπει να μεταβάλουν αναλόγως το ρυθμό με τον οποίο στέλνουν πακέτα στο δίκτυο βασιζόμενες σε κάποια ειδοποίηση από το δίκτυο για το ελεύθερο εύρος ζώνης. Από τη στιγμή που υπάρχουν αρκετές τέτοιες πηγές που ανταγωνίζονται για αυτό το εύρος ζώνης, προκύπτουν τα παρακάτω θέματα με τα οποία ασχολείται η εργασία [5] .

1. αποδοτική κατανομή του ελεύθερου εύρους ζώνης στις διαφορετικές πηγές με βάση τις διαφορετικές ανάγκες τους.
2. δικαιοσύνη
3. η δυνατότητα να υλοποιηθεί αυτό το σχήμα κατανεμημένα με την ελάχιστη δυνατή επιπλέον επικοινωνία μεταξύ των κόμβων.
4. το ζήτημα κοστολόγησης του εύρους ζώνης με τέτοιο τρόπο ώστε το κέρδος του δικτύου να μεγιστοποιείται όταν το εύρος ζώνης μοιράζεται στις πηγές σύμφωνα με τα 1,2.

Σε αυτή την εργασία προτείνεται μια λύση που βασίζεται στη θεωρία παιγνίων και συγκεκριμένα στο Nash bargaining framework από την συνεργατική θεωρία παιγνίων (cooperative game theory)

Το πρόβλημα λοιπόν μελοποιείται ως εξής:

Υπάρχουν N χρήστες που ανταγωνίζονται για το διαμοιρασμό του εύρους ζώνης. Κάθε χρήστης i έχει μια συνάρτηση επίδοσης f_i και μια επιθυμητή αρχική επίδοση u_i^0 η οποία είναι η ελάχιστη επίδοση που απαιτείται από τον χρήστη i χωρίς καμία συνεργασία προκειμένου να μπει στο παιχνίδι. Κάθε συνάρτηση επίδοσης ορίζεται σε ένα υποσύνολο του R^N , το X που είναι το σύνολο των στρατηγικών των N χρηστών. Στη

συγκεκριμένη περίπτωση το X θα μπορούσε να είναι ο χώρος των διαφορετικών διανυσμάτων που προκύπτουν από την ανάθεση διαφορετικής ποσότητας εύρους ζώνης σε κάθε χρήστη. Η αρχική επίδοση που ζητά ο κάθε χρήστης αναπαριστά ουσιαστικά την ελάχιστη επίδοση που θα πρέπει να μπορεί να εγγυηθεί το δίκτυο στον συγκεκριμένο χρήστη. Άρα κάνοντας την υπόθεση ότι κάθε χρήστης που εισέρχεται στο παιχνίδι λαμβάνει τουλάχιστον την αρχική του επίδοση ή και παραπάνω, υποθέτουμε ουσιαστικά ότι υπάρχει τουλάχιστον ένα διάνυσμα στο X για το οποίο το διάνυσμα επίδοσης $f = (f_1, \dots, f_N)$ είναι τουλάχιστον ίσο με το αρχικό διάνυσμα επίδοσης u^0 .

Η λύση λοιπόν στο πρόβλημα θα πρέπει να ικανοποιεί τα 5 αξιώματα που έχουν αναφερθεί και παραπάνω και θα πρέπει να μέγιστοι το παρακάτω γινόμενο

$$\text{Max} \prod_{j \in J} (f_j(x) - u_j^0) \quad x \in X_0 \quad (\text{Εξ.4.4})$$

Οπού J είναι το σύνολο των χρηστών που παίρνουν περισσότερο εύρος ζώνης από την αρχική ελάχιστη απαίτησή τους.

Το παραπάνω πρόβλημα μεγιστοποίησης έχει ως ισοδύναμο το

$$\text{Max} \sum_{j \in J} \ln(f_j(x) - u_j^0) \quad x \in X_0 \quad (\text{Εξ.4.5})$$

Αρχικά το πρόβλημα αντιμετωπίζεται σφαιρικά. Σύμφωνα με αυτό το μοντέλο ο κάθε σύνδεσμος έχει ως στόχο την μεγιστοποίηση του εύρους ζώνης που δεσμεύεται για αυτό πέρα από την ελάχιστη απαίτηση που είχε καθοριστεί στην αρχή. Η λύση αυτή θα πρέπει να κατανέμει δίκαια το εύρος ζώνης στους διαφορετικούς χρήστες αλλά να είναι και η πιο επικερδής για το ίδιο το δίκτυο. Αυτή η λύση αντιστοιχεί στην εύρεση του *NBS* για το πρόβλημα κατανομής. Έπειτα παρουσιάζεται μια λύση κατά την οποία το πρόβλημα δεν λύνεται κεντρικά όπως στην πρώτη περίπτωση αλλά κατανεμημένα από κάθε χρήστη ξεχωριστά. Με κατάλληλες αλλαγές στο στόχο που τίθεται στον κάθε χρήστη δηλαδή με την εισαγωγή των κατάλληλων κοστών, επιτυγχάνεται η ίδια λύση όπως και στην πρώτη περίπτωση αφήνοντας πάντα τους χρήστες να δράσουν άπληστα.

4.4.1 Βέλτιστες για το δίκτυο εκχωρήσεις ρυθμού

Αρχικά λοιπόν αντιμετωπίζεται το πρόβλημα κεντρικά ως εξής

Σχήμα Προφόρτωσης Περιεχομένου WWW

Έχουμε N συνδέσεις κάθε μια από τις οποίες αναγνωρίζεται από το μονοπάτι που ακολουθεί στο δίκτυο. Επιπλέον έχουμε L συνδέσμους μέσα στο δίκτυο. Για κάθε σύνδεση γίνεται η υπόθεση ότι είναι ελαστική με μέγιστο ρυθμό PR_i και ελάχιστο ρυθμό MR_i , ο οποίος ελάχιστος αυτός ρυθμός εγγυάται από το δίκτυο. Αυτές N συνδέσεις ανταγωνίζονται για τους διαθέσιμους πόρους του δικτύου. Ο επιτρεπτός χώρος διανυσμάτων σε αυτή την περίπτωση είναι

$$X_0 = \{x \in R^N / x \geq MR, x \leq PR, Ax \leq C\} \quad (\text{Εξ.4.6})$$

όπου C είναι το διάνυσμα των χωρητικοτήτων των links και $A = (a_{lp})_{l,p}$ είναι μια μήτρα $L \times N$ όπου a_{lp} είναι 1 αν ο σύνδεσμος l ανήκει στη σύνδεση p και 0 διαφορετικά.

Στο συγκεκριμένο πρόβλημα είναι φυσικό να γίνει η υπόθεση ότι κάθε σύνδεση έχει ως στόχο να δεσμεύσει περισσότερο εύρος ζώνης το οποίο θα είναι περισσότερο από την ελάχιστη απαίτηση της MR_i και όσο πιο κοντά γίνεται στην μέγιστη απαίτηση της PR_i . Άρα η συνάρτηση επίδοσης f_i ορίζεται ως το δεσμευμένο από αυτή τη σύνδεση εύρος ζώνης x_i .

Η λύση στο παραπάνω πρόβλημα δίνεται από την επίλυση του παρακάτω προβλήματος μεγιστοποίησης

$$\begin{cases} \max_x \prod (x_i - MR_i) \\ x_i \geq MR_i, i \in \{1 \dots N\} \\ x_i \leq PR_i, i \in \{1 \dots N\} \\ (Ax)_l \leq (C)_l, l \in \{1 \dots L\} \end{cases} \quad (\text{Εξ.4.7})$$

Η λύση είναι η

$$x_i = MR_i + \min \left\{ (PR_i - MR_i); \frac{1}{\sum_{l=1}^L \mu_l a_{li}} \right\} \quad (\text{Εξ.4.8})$$

Ο πολλαπλασιαστής Lagrange μ_l ερμηνεύεται ως το κόστος χρήσης μιας μονάδας εύρους ζώνης από τον σύνδεσμο l .

4.4.2 Το πρόβλημα από τη μεριά το χρήστη

Παραπάνω λοιπόν λύθηκε το πρόβλημα κεντρικά. Ωστόσο μια τέτοια επίλυση επιφέρει μεγάλη επιβάρυνση επικοινωνίας μεταξύ των διαφόρων κόμβων του δικτύου. Παρουσιάζεται η λύση που δίνεται στην εργασία[5] και η οποία είναι κατανεμημένη. Όταν όμως ο κάθε χρήστης δρώντας με αποκλειστικό στόχο τη μεγιστοποίηση του προσωπικού του συμφέροντος είναι άπληστος τότε προκύπτει μια ισορροπία Nash (Nash equilibrium) ή οποία όμως όντας Pareto inefficient δεν είναι NBS. Έπεται λοιπόν ότι οι χρήστες θα πρέπει να δρουν με διαφορετικά κριτήρια αν θέλουμε να δουλεύει ο καθένας από αυτούς άπληστα και παράλληλα να καταλήξουμε σε NBS για το δίκτυο. Η ιδέα είναι η εισαγωγή κοστών έτσι ώστε ο χρήστης να μεγιστοποιεί το άθροισμα που αντιστοιχεί στην επιθυμητή επίδοση μείον το κόστος που επιφέρει αυτή η επίδοση στον χρήστη. Με αυτό τον τρόπο και αφήνοντας τους χρήστες να δρουν άπληστα για τη μεγιστοποίηση αυτής της συνάρτησης καταλήγουμε σε Pareto optimal point.

Εισάγονται λοιπόν στο μοντέλο N θετικές παράμετροι που ουσιαστικά αντιπροσωπεύουν το κόστος ανά μονάδα δεσμευμένου εύρος ζώνης από καθένα από τους N χρήστες.

Στόχος λοιπόν του χρήστη είναι η επίλυση του παρακάτω προβλήματος βελτιστοποίησης

$$\begin{cases} \max_x \ln(x_i - MR_i) - a_i x_i \\ x_i \geq MR_i, i \in \{1 \dots N\} \\ x_i \leq PR_i, i \in \{1 \dots N\} \end{cases} \quad (\text{Εξ.4.9})$$

Ενώ από την άλλη το δίκτυο έχει το παρακάτω πρόβλημα

$$\begin{cases} \max_x \prod_{i=1}^N (x_i - MR_i) \\ x_i \geq MR_i, i \in \{1 \dots N\} \\ x_i \leq PR_i, i \in \{1 \dots N\} \\ (Ax)_l \leq (C)_l, l \in \{1 \dots L\} \end{cases} \quad (\text{Εξ.4.10})$$

Με κατάλληλη επιλογή των $a_i (= \sum_{l=1}^L a_{i,l} \mu_l)$ δείχνεται ότι η λύση του *centralized* προβλήματος όπως αυτό παρουσιάστηκε στην προηγούμενη ενότητα μεγιστοποιεί την

ωφέλεια (utility) του κάθε χρήστη αλλά και το συνολικό κέρδος του δικτύου.

Παρατηρούμε ότι για να προσδιορίσουμε τις σωστές τιμές του α χρειάζεται η επίλυση του κεντρικοποιημένου προβλήματος οπότε δεν επωφελούμαστε από την τοπική (localized) επίλυση που ακολουθεί. Στο[5] παρουσιάζεται ένας κατανεμημένος αλγόριθμος που βασίζεται στην gradient projection method. Αυτός ο αλγόριθμος εφαρμόζεται στο δυϊκό του αρχικού προβλήματος πράγμα που έχει ως πλεονέκτημα το ότι επειδή ο αριθμός των συνδέσμων είναι συνήθως μικρότερος από αυτόν των χρηστών, η επίλυσή του είναι απλούστερη. Το σημαντικότερο χαρακτηριστικό αυτού του αλγορίθμου είναι ότι για την εκτέλεση του σε κάθε σύνδεσμο χρειάζονται μόνο πληροφορίες για τους χρήστες που χρησιμοποιούν αυτό το σύνδεσμο και συνεπώς δεν χρειάζονται καθολικές πληροφορίες.

4.4.3 Τιμολόγηση

Επιπλέον και με βάση τα παραπάνω παρουσιάζεται μια λύση που εκτός από την κατανομή των πόρων του δικτύου ανάμεσα στους χρήστες δίνει και μία λύση για την κοστολόγηση των πόρων που κατανέμονται στον κάθε χρήστη κατά τέτοιο τρόπο ώστε να μεγιστοποιείται το κέρδος του δικτύου. Αυτό γίνεται λαμβάνοντας υπόψη τόσο τις απαιτήσεις όσο και την προθυμία του κάθε χρήστη να πληρώσει για εύρος ζώνης πέρα από το ελάχιστο εγγυημένο MR_i , δηλαδή από το προϋπολογισμό (budget) του. Όπως έχει δείχθει το κέρδος του δικτύου μεγιστοποιείται όταν το εύρος ζώνης κατανέμεται στους χρήστες σύμφωνα με το NBS. Αυτή είναι η βασική ιδιότητα που επιτρέπει τη μοντελοποίηση αυτού του σχήματος τιμολόγησης των χρηστών.

Στην αρχή λοιπόν κατά τη εγκατάσταση μιας σύνδεσης ο χρήστης κοινοποιεί στο δίκτυο την προθυμία του να πληρώσει για επιπλέον εύρος ζώνης πέρα του MR_i , δηλαδή του budget του. Αυτό το budget μπορεί να επιλεγεί από τον χρήστη ή να επιλεγεί από ένα αριθμό προκαθορισμένων τιμών που έχουν καθοριστεί από τον λειτουργό του δικτύου.. Αυτό που αναλαμβάνει τώρα ο λειτουργός του δικτύου είναι να κατανείμει τους πόρους του δικτύου δίκαια σύμφωνα με τις απαιτήσεις των χρηστών αλλά και το budget του καθενός από αυτούς.

Από τη σκοπιά του δικτύου πρέπει να στηθεί ένα σχήμα κοστολόγησης έτσι ώστε να μην υπερβαίνεται ποτέ το budget του χρήστη αλλά και το δίκτυο να αυξάνει το κέρδος του κατανέμοντας το εύρος ζώνης σύμφωνα με τις παραπάνω απαιτήσεις.

Το πρόβλημα εδώ λοιπόν αντιμετωπίζεται όπως και παραπάνω με τη διαφορά ότι εδώ

Σχήμα Προφόρτωσης Περιεχομένου WWW

έχουμε επιπλέον την παράμετρο B_i που είναι το budget του χρήστη i . Η λύση λοιπόν έρχεται πάλι από το NBS. Επειδή όμως αυτή τη φορά θέλουμε κατανομή των πόρων ανάλογα το budget του κάθε χρήστη θέλουμε την ασύμμετρη NBS. Αυτή έχει όλες τις ιδιότητες της NBS εκτός από αυτή που αναφέρεται στη συμμετρία των λύσεων.

Αναζητείται λοιπόν η λύση στο

$$\max_{x \in X_0} \prod_{i=1}^N (x_i - MR_i)^{B_i} \quad (\text{Εξ.4.11})$$

Και είναι η

- If $B_i = 0$ then $x_i = MR_i$.
- If $B_i > 0$ and $\sum_{l=1}^L \mu_l a_{li} > 0$ then

$$x_i = MR_i + \min \left(PR_i - MR_i, \frac{B_i}{\sum_{l=1}^L \mu_l a_{li}} \right) \quad (\text{Εξ.4.12})$$

where for each $l \in \{1 \dots L\}$, $(Ax - C)_l \mu_l = 0$.

- If $\sum_{l=1}^L \mu_l a_{li} = 0$, then $x_i = PR_i$.

Προτείνεται λοιπόν το ακόλουθο σχήμα κοστολόγησης

$$p_i(x_i) = T(B_i) + (x_i - MR_i) \sum_{l=1}^L a_{li} \mu_l \quad (\text{Εξ.4.13})$$

Το $T(\cdot)$ μπορεί να θεωρηθεί ως μια συγκεκριμένη τιμή που πρέπει να πληρώσει κάποιος χρήστης προκειμένου να έχει εγγύηση για ελάχιστο εύρος ζώνης MR_i . Η κύρια ιδιότητα ενός τέτοιου σχήματός κοστολόγησης είναι ότι περιέχει δύο όρους, ο ένας εκ των οποίων είναι σταθερός και σχετίζεται με το MR_i αλλά και το *budget* του χρήστη ενώ ο άλλος μεταβάλλεται ανάλογα με τις συνθήκες συμφόρησης του δικτύου. Αν ο χρήστης κοστολογείται σύμφωνα με την $p_i(x_i)$ τότε δείχνεται ότι αυξάνεται το κέρδος του δικτύου.

4.5 Περίπου βέλτιστος σχεδιασμός δικτύου με πράκτορες που συμπεριφέρονται εγωιστικά

Σε πολλά δίκτυα μεταξύ των οποίων και στο Internet δρουν agents οι οποίοι δρουν εγωιστικά και έχουν ένα περιορισμένο αριθμό στόχων. Αυτό συνιστά συνήθως μια

παιγνιοθεωρητική προσέγγιση στην συμπεριφορά αυτών των πρακτόρων και στη δομή των δικτύων που δημιουργούν. Οι ισορροπίες που προκύπτουν από αλληλεπιδράσεις τέτοιου είδους πρακτόρων συνιστούν ισορροπίες Nash. Όπως όμως έχει μελετηθεί στη βιβλιογραφία οι ισορροπίες Nash μπορεί να είναι πολύ ακριβότερες από το καθολικό βέλτιστο. Σε μερικές περιπτώσεις η ισορροπία Nash είναι μοναδική ενώ σε άλλες η καλύτερη από αυτές αντιστοιχεί στη βέλτιστη λύση του προβλήματος. Σε αυτή την περίπτωση εξετάζεται πόσο άσχημη είναι η χειρότερη ισορροπία Nash σε σχέση με τη βέλτιστη λύση. Σε μερικές μάλιστα περιπτώσεις η ποιότητα ακόμα και της καλύτερης ισορροπίας Nash είναι πολύ χειρότερη από τη βέλτιστη λύση του προβλήματος (πχ prisoners dilemma). Στην εργασία [18] εξετάζεται, σε αντίθεση με άλλες εργασίες που εξετάζει ποσό άσχημη μπορεί να είναι μια ισορροπία Nash, το πόσο καλή σε σχέση πάντα με τη βέλτιστη λύση μπορεί να είναι μια ισορροπία Nash. Ο λόγος της καλύτερης Nash ισορροπίας προς την βέλτιστη λύση του προβλήματος λέγεται optimistic price of anarchy.

Στην εργασία [18] εξετάζεται η περίπτωση κατασκευής δικτύου από πράκτορες που δρουν εγωιστικά και έχουν ένα συγκεκριμένο στόχο. Καθένας από αυτούς έχει ένα σύνολο από τερματικά τα οποία θέλει να συνδέσει μεταξύ τους. Κάθε σύνδεσμος έχει ένα κόστος αγοράς-χρήσης. Σκοπός κάθε πράκτορα είναι να συνδέσει τα τερματικά του μεταξύ τους με το ελάχιστο κόστος. Ένα ενδιαφέρον χαρακτηριστικό αυτού του παιχνιδιού είναι ότι οι πράκτορες έχουν κίνητρο να συνεργαστούν μεταξύ τους μοιραζόμενοι κόστη για στην αγορά συγκεκριμένων συνδέσμων προκειμένου να συνδέσουν τα τερματικά τους ο καθένας. Συγκεκριμένα στρατηγική του κάθε παίχτη είναι μια συνάρτηση πληρωμής p_i , όπου $p_i(e)$ είναι το ποσό που προσφέρει ο παίκτης i για την αγορά της ακμής e . Αν για κάποιο σύνδεσμο το συνολικό ποσό που έχει μαζευτεί είναι μεγαλύτερο από το κόστος του $\sum_i p_i(e) \geq c(e)$ τότε αυτό θεωρείται αγορασμένο και μπορεί πλέον χρησιμοποιηθεί από οποιοδήποτε άλλο παίχτη ακόμα και αν αυτός δεν έχει πληρώσει για το συγκεκριμένο σύνδεσμο. Προκύπτει έτσι ο γράφος G_p που είναι τους αγορασμένους συνδέσμους ενώ οι παίχτες έχουν προσφέρει για την αγορά των ακμών αυτού του γράφου $p = (p_1, \dots, p_N)$. Στόχος του κάθε παίχτη είναι η ελαχιστοποίηση των εξόδων του $\sum_{e \in E} p_i(e)$

Μια ισορροπία Nash για αυτό το παιχνίδι είναι μια συνάρτηση πληρωμής τέτοια ώστε κανένας παίχτης δεν θα έχει κίνητρο να μετακινηθεί από τις επιλογές πληρωμής που

έχει κάνει. Ένα $(1+\epsilon)$ -approximate Nash equilibrium είναι από την άλλη μια συνάρτηση p τέτοια ώστε κανένας παίχτης δεν μπορεί να μειώσει τα έξοδά του περισσότερο από ένα παράγοντα $1+\epsilon$ αλλάζοντας τις επιλογές πληρωμής του.

Η εύρεση της κεντρικοποιημένης βέλτιστης λύσης αυτού του προβλήματος, δηλαδή η εύρεση των συνδέσμων που πρέπει να αγοραστούν προκειμένου η συνολική συνεισφορά των παιχτών να είναι η ελάχιστη συνιστά το πρόβλημα σχεδιασμού του γενικευμένου Steiner tree.

Η εργασία ασχολείται κυρίως με ντετερμινιστικές ισορροπίες Nash και όχι με mixed strategies. Επιπλέον εξετάζεται η έννοια που approximate Nash equilibrium και συγκεκριμένα εξετάζεται πόσο μακριά από μια πραγματική ισορροπία θα πρέπει να μετακινηθούμε προκειμένου να καταλήξουμε στη βέλτιστη λύση, δηλαδή πόσο δυσαρεστημένοι θα είναι οι πράκτορες αν τους επιβληθεί να πληρώσουν για την κοινωνικά βέλτιστη λύση (socially optimum solution).

Έστω ότι έχουμε ένα Nash equilibrium p και έστω T_i το μικρότερο δέντρο στο G_p που συνδέει όλα τα τερματικά του i . Ισχύει ότι

1. το G_p είναι δάσος
2. κάθε παίχτης i συνεισφέρει μονό για την αγορά των συνδέσμων που ανήκουν στο T_i
3. και κάθε σύνδεσμος είτε πληρώνεται εξ ολοκλήρου η καθόλου.

Υπάρχουν στιγμιότυπα του παραπάνω προβλήματος για τα οποία δεν υπάρχει pure Nash equilibrium. Ακόμα όμως και αν υπάρχουν αυτά ποσό άσχημα μπορεί να είναι σε σχέση με τη βέλτιστη λύση; Δείχνεται στη εργασία ότι το τιμή της αναρχίας είναι το πολύ N όπου N αριθμός των παιχτών.

Έπειτα αναλύεται η περίπτωση όπου έχουμε μια μόνο πηγή την οποία κάθε παίχτης θέλει να συνδέσει με όλα τα τερματικά του. Αποδεικνύεται ότι σε αυτή την περίπτωση υπάρχει πάντα ισορροπίας Nash και η αισιόδοξη τιμή αναρχίας (optimistic price of anarchy) είναι 1. Ο αλγόριθμος που παρουσιάζεται μας δίνει, δεδομένου του βέλτιστου Steiner tree, η βέλτιστη ισορροπία Nash το οποίο μάλιστα συμπίπτει και με τη βέλτιστη λύση του προβλήματος. Δεδομένης όμως της δυσκολίας εύρεσης του βέλτιστου Steiner tree στη γενική περίπτωση, αποδεικνύεται ότι δεδομένου ενός α -approximate minimum cost Steiner tree T μπορούμε σε πολυωνυμικό χρόνο να καταπλήξουμε σε ένα $(1+\epsilon)$ -

approximate Nash equilibrium πάνω σε ένα *Steiner tree* T' με $c(T') \leq c(T)$

Για την γενική περίπτωση όπου έχουμε πολλές πηγές και ο κάθε πράκτορας θέλει να συνδέσει κάποιες από αυτές με τα τερματικά του αποδεικνύεται ότι για κάθε βέλτιστη κεντρικοποιημένη λύση T^* υπάρχει ένα 3-approximate Nash equilibrium στο οποίο οι πράκτορες αγοράζουν links ακριβώς πάνω στο T^*

4.6 Εγωιστική ενδιάμεση αποθήκευση σε κατανεμημένα συστήματα: μια παιγνιοθεωρητική ανάλυση

Στην εργασία[10] γίνεται η υπόθεση ότι οι εξυπηρέτες ενδιάμεσης αποθήκευσης δεν υπακούουν σε κάποιο πρωτόκολλο σχεδιασμένο για τη βελτιστοποίηση της απόδοσης ολόκληρου του συστήματος ενδιάμεσης αποθήκευσης, αλλά δρουν εγωιστικά. Έτσι για παράδειγμα λαμβάνουν την απόφαση για το αν θα αποθηκεύσουν η όχι ένα αντικείμενο μονό με άξονα την καλύτερη εξυπηρέτηση του δικού τους domain και έτσι το σύστημα ξεφεύγει από την καθολικά βέλτιστη συμπεριφορά. Σε αυτή την εργασία λοιπόν μοντελοποιείται η εγωιστική ενδιάμεση αποθήκευση ως ένα μη συνεργατικό παίγνιο.

Στο βασικό μοντέλο που εξετάζεται κάθε εξυπηρέτης έχει δύο επιλογές για κάθε αντικείμενο. Αν ένα αντίγραφο αυτού του αντικειμένου βρίσκεται αρκετά σε ένα άλλο εξυπηρέτη κοντά στον πρώτο τότε αυτός ο εξυπηρέτης μπορεί να επιλέξει να μην το αποθηκεύσει στην ενδιάμεση μνήμη. Αν από την άλλη τα αντίγραφα είναι αποθηκευμένα πολύ μακριά από αυτόν τον εξυπηρέτη τότε επιλέγει να αποθηκεύσει ο ίδιος το αντικείμενο. Αποφάσεις για την μία ή την άλλη περίπτωση παίρνονται τοπικά παίρνοντας υπόψη μόνο τοπικά κόστη. Επιπλέον εισάγεται και ένα μοντέλο πληρωμών σύμφωνα με το οποίο κάθε εξυπηρέτης κάνει μια προσφορά για να αποθηκευθεί ένα αντικείμενο σε ένα άλλο εξυπηρέτη. Ο εξυπηρέτης που αποθηκεύει ένα αντικείμενο μαζεύει όλες τις σχετικές προσφορές. Αφού κάθε παίχτης έχει επιλέξει την στρατηγική του το παιχνίδι ορίζει μια διαμόρφωση (configuration) δηλαδή το σύνολο των εξυπηρετών που τελικά θα αποθηκεύσουν το κάθε αντικείμενο άλλα και τα έξοδα που σχετίζονται με καθένα από αυτούς.

Στη συγκεκριμένη εργασία εξετάζονται μόνο αμιγείς ισορροπίες Nash. Είναι όμως δυνατό, όπως έχει αναφερθεί και παραπάνω, να καταλήξουμε σε μια ισορροπία Nash της οποίας το κόστος θα είναι πολύ υψηλότερο από το αντίστοιχο της βέλτιστης λύσης του προβλήματος. Αυτή η μη-αποδοτικότητα καταστάσεων ισορροπίας στις οποίες περιέρχεται ένα σύστημα στο οποίο ο κάθε παίχτης δρα με μόνο άξονα το προσωπικό

του συμφέρον ποσοτικοποιείται με την τιμή αναρχίας που είναι ο λόγος της χειρότερης, από πλευράς κόστους, Nash ισορροπίας προς το κόστος της βέλτιστης λύσης. Ωστόσο σχεδόν πάντα υπάρχουν τρόποι σύμφωνα με τους οποίους μπορούμε να οδηγήσουμε το σύστημα σε μια προεπιλεγμένη ισορροπία Nash. Για αυτόν τον λόγο στη συγκεκριμένη εργασία λαμβάνεται υπόψη και η αισιόδοξη τιμή αναρχίας.

Σε αυτή την εργασία λοιπόν εξετάζεται

1. Αν υπάρχουν ισορροπίες Nash σε ένα παίγνιο ενδιάμεσης αποθήκευσης
2. Πόσο αποδοτικές είναι αυτές
3. Τι επίπτωση έχει η εισαγωγή πληρωμών όσον αφορά την Nash ισορροπία στην οποία καταλήγουμε.

4.6.1 Βασικό παίγνιο

Έχουμε ένα σύνολο N από n εξυπηρέτες και ένα σύνολο M από m αντικείμενα. Η απόσταση μεταξύ διαφορετικών εξυπηρετών μπορεί να αναπαρασταθεί με τον πίνακα D . Κάθε εξυπηρέτης έχει μια ζήτηση για καθένα από τα αντικείμενα, η οποία καθορίζεται από τις απαιτήσεις των πελατών που αυτός εξυπηρετεί. Η ζήτηση λοιπόν του κάθε εξυπηρέτη i για το αντικείμενο j αναπαρίσταται από τον πίνακα W που w_{ij} είναι η ζήτηση του αντικειμένου j από τον εξυπηρέτη i . Όταν ένας εξυπηρέτης i αποφασίσει να αποθηκεύσει ένα αντικείμενο j τότε επιβαρύνεται με ένα κόστος a_{ij} . Γίνεται η υπόθεση ότι δεν υπάρχει περιορισμός χώρου. Μια διαμόρφωση (configuration) X για ένα αντικείμενο είναι το σύνολο των εξυπηρετών που τελικά θα αποθηκεύσουν αυτό το αντικείμενο.

Έστω λοιπόν A_i το σύνολο των στρατηγικών που μπορεί να επιλέξει ο παίχτης i , και S_i η στρατηγική που επιλέγεται από τον παίχτη αυτόν (το ποια αντικείμενα δηλαδή θα αποθηκεύσει). Δεδομένων των στρατηγικών όλων των παιχτών $S = (S_1, S_2, \dots, S_n)$ το κόστος με το οποίο επιβαρύνεται ένας παίχτης i είναι

$$C_i(S) = \sum_{j \in \mathcal{S}_i} a_{ij} + \sum_{j \notin \mathcal{S}_i} w_{ij} d_{il(ij)} \quad (\text{Εξ.4.14})$$

όπου $l(i, j)$ είναι ο κοντινότερος εξυπηρέτης στον i που έχει αποθηκεύσει το αντικείμενο j .

Δεδομένου ότι δεν υπάρχουν περιορισμοί χώρου μπορούμε να αντιμετωπίσουμε την

Σχήμα Προφόρτωσης Περιεχομένου WWW

αποθήκευση κάθε αντικειμένου σαν ένα ξεχωριστό παιχνίδι. Για ένα και μόνο αντικείμενο λοιπόν ο κάθε εξυπηρέτης επιλέγει αν θα το αποθηκεύσει ή όχι. Το αντικείμενο που εξετάζεται είναι το j . Είναι $S_i = 1$ αν ο εξυπηρέτης i αποθηκεύσει το j και $S_i = 0$ διαφορετικά. Το κόστος με το οποίο επιβαρύνεται ο εξυπηρέτης i είναι

$$C_i(S) = \sum_{j \in S_i} a_{ij} + \sum_{j \notin S_i} w_{ij} d_{il(ij)} \quad (\text{Εξ.4.15})$$

Ξεκινάμε λοιπόν με μια τυχαία διαμόρφωση και το αφήνουμε να εξελιχθεί καθώς οι παίχτες αλλάζουν τις στρατηγικές τους προκειμένου να μειώσουν το κόστος τους ο καθένας. Μια ισορροπία Nash υφίσταται όταν κανένας από τους παίχτες δεν έχει κίνητρο να αλλάξει τη στρατηγική του δηλαδή δεν επωφελείται από μια τέτοια αλλαγή. Μια ισορροπία Nash για το βασικό παίγνιο (S_i^*, S_{-i}^*) ορίζει μια configuration X τέτοια ώστε $\forall i \in N, i \in X \Leftrightarrow S_i^* = 1$

Αν \mathcal{E} το σύνολο όλων των αμιγών ισορροπιών Nash τότε

$$X \in \mathcal{E} \Leftrightarrow \forall i \in N, \forall S_i \in A_i, C_i(S_i^*, S_{-i}^*) \leq C_i(S_i, S_{-i}^*) \quad (\text{Εξ.4.16})$$

Και για το βασικό παίγνιο μπορεί εύκολα να διαπιστωθεί ότι ισχύει

$$X \in \mathcal{E} \Leftrightarrow \forall i \in N, \exists j \in X \text{ s.t. } d_{ji} \leq \alpha \text{ and } \forall j \in X, \neg \exists k \in X \text{ s.t. } d_{kj} < \alpha \quad (\text{Εξ.4.17})$$

Αυτές οι ιδιότητες είναι απαραίτητες ώστε να μην έχει κανένας παίχτης όφελος αλλάζοντας τη στρατηγική του.

Το κοινωνικό κόστος (social cost) είναι

$$C(S) = \sum_{i=0}^{n-1} C_i(S) \quad (\text{Εξ.4.18})$$

ενώ το κοινωνικά βέλτιστο κόστος είναι

$$C(S_0) = \min_S C(S) \quad (\text{Εξ.4.19})$$

Αποδεικνύεται ότι υπάρχει αμιγής ισορροπία Nash για αυτό το βασικό παίγνιο. Έπειτα εξετάζονται τα χαρακτηριστικά ποιότητας μιας ισορροπίας Nash δηλαδή το πόσο άσχημη μπορεί να είναι (price of anarchy) αλλά και ποσό καλή μπορεί να είναι

(optimistic price of anarchy). Το χειρότερο Nash equilibrium είναι

$$C(S_w) = \max_{X \in E} (a | X | + \sum_{j \in X} \min d_{ij}) \quad (\text{Εξ.4.20})$$

Εξετάζονται φράγματα για την τιμή της αναρχίας καθώς μεταβάλλεται το κόστος αποθήκευσης a ενός αντικειμένου. αν $a \leq d_{\min}$ τότε $PoA=1$ αφού τόσο στην ισορροπία Nash όσο και στο κοινωνικό βέλτιστο κάθε εξυπηρέτης αποθηκεύει το αντικείμενο. Όταν $a > d_{\max}$ τότε επειδή το κόστος αποθήκευσης είναι μεγαλύτερο από τη μεγαλύτερη απόσταση στο δίκτυο, μονό ένας εξυπηρέτης αποθηκεύει το αντικείμενο ενώ όλοι οι άλλοι έχουν πρόσβαση σε αυτό μόνο μέσω αυτού του εξυπηρέτη. Ωστόσο στο κοινωνικό βέλτιστο (social optimum) ενδέχεται να τοποθετηθούν περισσότερα από ένα αντίγραφα. Επειδή $a \leq C(S_0) \leq a + \min_{j \in N} \sum_i d_{ij}$ όταν $a > d_{\max}$ είναι

$$\frac{a + \max_{j \in N} \sum_i d_{ij}}{a + \min_{j \in N} \sum_i d_{ij}} \leq PoA \leq \frac{a + \min_{j \in N} \sum_i d_{ij}}{a} \quad (\text{Εξ.4.21})$$

Ανάλογα με την τοπολογία του δικτύου ακόμα και το κάτω φράγμα μπορεί να είναι $O(n)$.

Τέλος παρατηρείται μια αλλαγή όταν $a \geq \min_{j \in N} \sum_i d_{ij}$. Τότε $PoA = \frac{a + \max_{j \in N} \sum_i d_{ij}}{a + \min_{j \in N} \sum_i d_{ij}}$ και

έχει άνω φράγμα το 2.

4.6.2 Παίγνιο πληρωμών

Σε αυτή την επέκταση του αρχικού παιχνίσιου ο κάθε παίχτης κάνει μια προσφορά σε κάποιον άλλο προκειμένου ο τελευταίος να αποθηκεύσει το αντικείμενο. Το κόστος της αποθήκευσης μοιράζεται έτσι μεταξύ των κόμβων που έκαναν προσφορές στον κόμβο που τελικά αποθηκεύει το αντικείμενο. Μια στρατηγική για ένα παίχτη i είναι μια τριπλέτα $(u_i, b_i, t_i) \in \{N, \mathbb{R}_+, \mathbb{R}_+\}$ όπου u_i ο παίκτης στον οποίο ο i έκανε προσφορά, b_i είναι η προσφορά και t_i είναι το κατώφλι προσφορών που πρέπει να πάρει ο i προκειμένου να αποθηκεύσει το αντικείμενο. Επιπλέον ως R_i ορίζεται το ποσό των συνολικών προσφορών που λαμβάνονται από ένα παίχτη i . Ένας παίχτης αποθηκεύει το αντικείμενο αν $R_i \geq t_i$. Το I_i είναι 1 αν ο i αποθηκεύει το αντικείμενο και 0 διαφορετικά..

Σε αυτή στην περίπτωση λοιπόν το κόστος ενός παίκτη i είναι

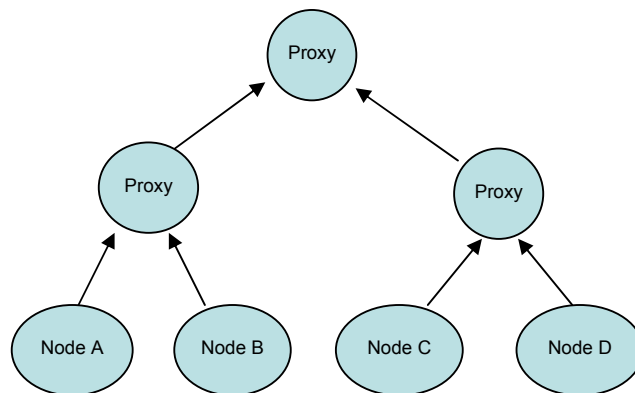
$$C_i(S) = a_{ij}I_i + w_{ij}d_{il(i,j)}(1 - I_i) + b_iI_{u_i} - R_iI_i \quad (\text{Εξ.4.22})$$

παρατηρείται ότι το κοινωνικά βέλτιστο για το payment game είναι το ίδιο με αυτό του βασικού παίγνιου. Δείχνεται επίσης ότι μια ισορροπία Nash στο αρχικό παίγνιο είναι επίσης ισορροπία Nash και για το payment game, άρα το PoA αυτού του παίγνιου είναι τουλάχιστο αυτό του βασικού παίγνιου. Το ενδιαφέρον όμως είναι ότι αποδεικνύεται ότι σε αυτό το παίγνιο υπάρχει πάντα ένα προφίλ στρατηγικών τέτοιο ώστε η ισορροπία Nash που προκύπτει είναι το κοινωνικό βέλτιστο. άρα η αισιόδοξη τιμή αναρχίας (optimistic price of anarchy) σε αυτό το παίγνιο είναι πάντα 1.

ΚΕΦΑΛΑΙΟ: 5

ΠΡΟΤΕΙΝΟΜΕΝΟ ΣΧΗΜΑ ΠΡΟΑΝΑΚΤΗΣΗΣ WEB ΠΕΡΙΕΧΟΜΕΝΟΥ

Η παρούσα εργασία στοχεύει στην εφαρμογή ενός σχήματος διαιτησίας για την προανάκτηση web περιεχομένου σε μια ιεραρχία ενδιάμεσων μνημών. Συγκεκριμένα, μελετάται η λειτουργία ενός συστήματος μεσαζόντων με ενδιάμεση μνήμη (WWW proxy caches). Η δικτυακή υποδομή η οποία εξετάζεται είναι αυτή που παρουσιάζεται στην Εικόνα 7.



Εικόνα 7: Δικτυακή Αρχιτεκτονική

Όπως φαίνεται στην Εικόνα 7, την εξεταζόμενη αρχιτεκτονική συγκροτούν ένα σύνολο από ενδιάμεσες μνήμες οργανωμένων σε μια ιεραρχία. Η κάθε μια από αυτές έχει δύο

παιδιά και ένα γονέα. Τα μεσάζοντες με ενδιαμέση μνήμη εξυπηρετούν τις απαιτήσεις σε Web περιεχόμενο των τελικών χρηστών. Από τη χρήση των μεσαζόντων ενδιαμέσης μνήμης επιδιώκεται η μείωση της εκλαμβανόμενης από τους χρήστες μέσης καθυστέρησης κατά την αίτηση ενός αντικειμένου από ένα εξυπηρέτη.

Για την βελτίωση της υπηρεσίας που θα παρέχει αυτή η ιεραρχία από ενδιαμέσες μνήμες στους χρήστες της, προτείνεται ένα σχήμα προανάκτησης πόρων. Ο στόχος που έχει τεθεί είναι η όσο το δυνατό καλύτερη εξυπηρέτηση ενός όσο το δυνατό μεγαλύτερου αριθμού από τους χρήστες της ενδιαμέσης μνήμης. Με την προανάκτηση ενός αριθμού από αντικείμενα αναμένουμε την μείωση στην μέση καθυστέρηση που θα εκλαμβάνεται καθένας από τους χρηστές του σχήματος ενδιαμέσης αποθήκευσης. Παράλληλα έχει τεθεί ως στόχος η προανάκτηση των πόρων να μην γίνεται παράλληλα με την εκτέλεση κανονικών http αιτήσεων αφού είναι πιθανό κάτι τέτοιο να επιφέρει επιπλέον καθυστερήσεις στην εξυπηρέτηση των τελευταίων. Έτσι γίνεται η υπόθεση ότι η προανάκτηση των πόρων από τους εξυπηρέτες γίνεται κατά τις βραδινές ώρες όταν ο φόρτος που αντιμετωπίζει το δίκτυο από κανονικές http αιτήσεις είναι μειωμένος. Ακριβώς για αυτό το λόγο θα θέλαμε η προανάκτηση των αντικειμένων να λάβει τέλος πριν αρχίσει πάλι η κανονική ροή των http αιτήσεων. Έτσι δεν θα καθυστερείται η εξυπηρέτηση των κανονικών http αιτήσεων λόγω επιβάρυνσης του δικτύου από ταυτόχρονη προανάκτηση αντικειμένων αλλά άπιστης θα έχουν έρθει και τα προανακτηθέντα αντικείμενα ώστε αν κάποια από τις κανονικές http αιτήσεις αιτηθεί κάποιο από αυτά να μπορεί αυτή να εξυπηρετηθεί τοπικά.

Οι λίστες επιθυμητών πόρων προσδιορίζονται από τους μεσάζοντες εξυπηρέτες του κατώτερου επιπέδου της ιεραρχίας βάσει του αλγορίθμου Top-10. Ο αλγόριθμος Top10 προκειμένου να κατασκευάσει αυτές τις αιτήσεις προανάκτησης αντικειμένων, λαμβάνει υπόψη του τις αιτήσεις που έχουν γίνει μέσω του συγκεκριμένου κάθε φορά μεσάζοντα εξυπηρέτη προς τους εξυπηρέτες. Αυτές οι αιτήσεις προανάκτησης προωθούνται στην συνέχεια στο ανώτερο επίπεδο της ιεραρχίας.

Η προανάκτηση των πόρων όπως προαναφέρθηκε πραγματοποιείται off-line σε ώρες που ο φόρτος του δικτύου είναι περιορισμένος. Όπως επίσης προαναφέρθηκε το σύστημα δρα κάτω από χρονικούς περιορισμούς όσον αφορά την προανάκτηση των αντικειμένων. Οι σχηματιζόμενες από τους μεσάζοντες εξυπηρέτες του πρώτου επιπέδου της ιεραρχίας αιτήσεις προανάκτησης βάσει του αλγορίθμου Top10 ενδέχεται να αιτούνται ένα σύνολο αντικειμένων το οποίο αν προανακτηθεί ως έχει, να

παραβιάζεται ο χρονικός περιορισμός που έχει τεθεί. Για αυτόν ακριβώς το λόγο προτείνεται ένας αλγόριθμος διαιτησίας που θα είχε ως στόχο τον περιορισμό αυτών των αιτήσεων από τον γονέα των μεσαζόντων εξυπηρετών πρώτου επιπέδου κατά τέτοιο τρόπο ώστε ενώ θα πληρείται ο περιορισμός που έχει τεθεί, παράλληλα όμως να εξυπηρετείται από τους πόρους που θα προανακτηθούν ένας όσο το δυνατό μεγαλύτερος αριθμός από μεσάζοντες εξυπηρετές πρώτου επιπέδου και μέσω αυτών οι αντίστοιχοι πελάτες τους.

Προτείνεται η εφαρμογή ενός μηχανισμού διαιτησίας (arbitration scheme). Για να υλοποιηθεί το προτεινόμενο σχήμα θα πρέπει να σχεδιαστεί και εφαρμοστεί μία συνάρτηση ωφέλειας (utility function). Η συνάρτηση ωφέλειας αφορά τα περιεχόμενα των λιστών πόρων που υποβάλλονται στο μηχανισμό pre-fetching.

5.1 Σχήμα Διαχείρισης προανάκτησης WWW Πόρων

Το προτεινόμενο σχήμα διαχείρισης προανάκτησης WWW πόρων βασίζεται σε ένα μηχανισμό διαιτησίας (arbitration scheme). Η αρχιτεκτονική υποθέτει ότι τα φύλλα μίας ιεραρχίας δικτύου διανομής WWW περιεχομένου (εταιρικοί μεσάζοντες εξυπηρετές, αυτόνομοι WWW πελάτες) προωθούν προς τα ανώτερα επίπεδα αιτήσεις προανάκτησης. Οι αιτήσεις αυτές, όπως προτείνεται στην βιβλιογραφία, υποδεικνύουν κάποιο συγκεκριμένο κόμβο (WWW εξυπηρετή) καθώς και το μέγεθος της προανάκτησης (πόσοι από τους πιο δημοφιλείς πόρους στον συγκεκριμένο κόμβο θα πρέπει να προανακτηθούν). Η επαυξημένη (augmented) BNF μορφή των αιτήσεων προανάκτησης είναι η ακόλουθη:

Αίτηση προανάκτησης = * (Στοιχείο Αίτησης)

Στοιχείο Αίτησης = Διεύθυνση Κόμβου + Πλήθος Στοιχείων για προανάκτηση +
Χρονικός περιορισμός

Ο χρονικός περιορισμός αναφέρεται στον αριθμό των χρονικών μονάδων από τη στιγμή της υποβολής της αίτησης προανάκτησης αντικειμένων από τον μεσάζοντα εξυπηρετή πρώτου επιπέδου, μέχρι τη στιγμή όπου θα πρέπει να έχουν φτάσει οι πόροι που θα προανακτηθούν από τον συγκεκριμένο εξυπηρετή για λογαριασμό αυτού του μεσάζοντα εξυπηρετή.

Παρακάτω αναλύεται ο μηχανισμός διαιτησίας που προτείνεται για τον περιορισμό στον γονέα μεσάζοντα εξυπηρετή, των αιτήσεων προανάκτησης που έχει δεχθεί από τα παιδιά μεσάζοντες εξυπηρετές. Παρακάτω αυτά τα παιδιά θα αναφέρονται ως πελάτες.

Οι πελάτες συγχρονισμένα μεταξύ τους και ανά τακτά χρονικά διαστήματα προωθούν αιτήσεις προανάκτησης που κατασκευάστηκαν με βάση τον αλγόριθμο Top10, στο γονέα μεσάζοντα εξυπηρέτη

Ο μηχανισμός διαιτησίας στο γονέα μεσάζοντα εξυπηρέτη, αφού δεχθεί τις αιτήσεις των κατωτέρων επιπέδων πρέπει να περιορίσει αυτές τις αιτήσεις κατά τέτοιο τρόπο ώστε ενώ θα τηρείται ο περιορισμός που έχει τεθεί, παράλληλα να προανακτηθούν τόσοι πόροι και από τέτοιους εξυπηρέτες πηγής έτσι ώστε να μειώνεται η εκλαμβανόμενη καθυστέρηση από ένα όσο το δυνατό μεγαλύτερο αριθμό πελατών. (από τη στιγμή που εδώ έχουμε μόνο δύο πελάτες ο παραπάνω στόχος μεταφράζεται σε προτεραιότητα στην προανάκτηση αντικειμένων από τέτοιους εξυπηρέτες έτσι ώστε να υπάρχει μείωση στην αναμενόμενη καθυστέρηση κατά την εξυπηρέτηση μια http αίτησης και από τους δυο πελάτες σε σχέση με την μείωση στην καθυστέρηση που θα αντιλαμβανόταν μονό ένας από αυτούς).

Το πρώτο βήμα που γίνεται στον γονέα μεσάζοντα εξυπηρέτη μετά τη λήψη και των δυο αιτήσεων προανάκτησης από τους δύο πελάτες του, είναι να παραγάγει μια νέα αίτηση προανάκτησης που θα προκύψει ως μίξη αυτών των δυο. Αυτή η μίξη των δύο αιτήσεων σε μία γίνεται ως εξής. Κάθε γραμμή μιας αίτησης που αιτείται αντικείμενα από εξυπηρέτη από τον οποίο η άλλη αίτηση προανάκτησης δεν αιτείται κανένα αντικείμενο, αντιγράφεται ως έχει στην νέα αίτηση. Σε αντίθετη περίπτωση, αν δηλαδή ένας εξυπηρέτης αναφέρεται και στις δύο αιτήσεις τότε δημιουργείται μια νέα γραμμή στην νέα αίτηση που κατασκευάζεται. Αυτή η γραμμή θα αιτείται πόρους από το συγκεκριμένο εξυπηρέτη. Ο αριθμός των αντικειμένων που θα αιτείται θα είναι ο μέγιστος των αντίστοιχων στις αιτήσεις προανάκτησης των πελατών. Ο χρονικός περιορισμός που θα τίθεται για την προανάκτηση αντικειμένων από το συγκεκριμένο εξυπηρέτη θα είναι αυτός ο οποίος είναι πιο περιοριστικός από πλευράς ρυθμού λήψης αντικειμένων από τον συγκεκριμένο εξυπηρέτη μεταξύ των περιορισμών που έχουν τεθεί από τους δυο πελάτες. Για παράδειγμα αν ο πελάτης1 ζητά 20 αντικείμενα από τον εξυπηρέτη B σε 1000sec και ο πελάτης2 ζητά 200 αντικείμενα σε 1500sec από τον εξυπηρέτη B μέσω της δικιάς του αίτησης προανάκτησης, τότε σαν περιορισμός στην mixed αίτηση προανάκτησης για προανάκτηση αντικειμένων από το συγκεκριμένο εξυπηρέτη θα τεθεί το 1500 επειδή $\max(20/1000, 200/1500) = 200/1500$. Αυτό έχει τη λογική ότι αν έχει ο γονέας τα 200 αντικείμενα που ζήτησε ο πελάτης2 σε 1500sec τότε προφανώς θα έχει τα 20 αντικείμενα που ζήτησε ο πελάτης1 σε 1000sec άρα θα

μπορέσει να του τα προωθήσει εγκαίρως.

Η πληροφορία που χρησιμοποιεί ο arbitrator προκειμένου να περιορίσει κατάλληλα την mixed αίτηση προανάκτησης που μόλις κατασκευάστηκε είναι:

- Από ποιους origin και πόσους πόρους από τον καθένα τους έχει ζητήσει ο κάθε πελάτης (γνώση της αίτησης προανάκτησης για κάθε πελάτη). Ο διαιτητής δεν ξχνά μετά την δημιουργία της mixed αίτηση προανάκτησης, πόσα αντικείμενα και από ποιους εξυπηρέτες ζήτησε ο κάθε πελάτης. Η διατήρηση αυτής της πληροφορίας είναι χρήσιμη ώστε να μπορεί να δοθεί προτεραιότητα στην λήψη αντικειμένων από εξυπηρέτες που έχουν ζήτηση μέσω των αιτήσεων προανάκτησης και των δύο πελατών. Με αυτό τον τρόπο θα καταστεί δυνατή η μείωση της αναμενόμενης καθυστέρησης για ένα όσο το δυνατό μεγαλύτερο αριθμό πελατών (εδώ δύο) με κατανάλωση του ελάχιστου δυνατού επιπλέον εύρους ζώνης (η χρόνου) οπότε θα μένει και επιπλέον εύρος ζώνης για προανάκτηση και άλλων αντικειμένων.
- Τη μέση καθυστέρηση για την λήψη ενός αντικειμένου από τον κάθε εξυπηρέτη πηγής (κανονικοποιημένη ως προς το μέγεθος της απάντησης). Αυτή η πληροφορία είναι απαραίτητη γιατί ουσιαστικά αντικατοπτρίζει τη μέση καθυστέρηση που υφίσταται η εξυπηρέτηση της αίτησης ενός πελάτη όταν αυτή αναφέρεται σε αντικείμενο ενός εξυπηρέτη X το οποίο εκείνη τη στιγμή δεν βρίσκεται αποθηκευμένο στη ενδιάμεση μνήμη του μεσάζοντα εξυπηρέτη. Ο διαιτητής έχοντας ως στόχο την μείωση της εκλαμβανόμενης από κάθε πελάτη καθυστέρησης χρησιμοποιεί αυτήν την πληροφορία ώστε να προτιμήσει την προανάκτηση αντικειμένων από εξυπηρέτες από τους οποίους η μέση (κανονικοποιημένη) καθυστέρηση κατά την λήψη ενός αντικειμένου είναι μεγαλύτερη. Προανακτώντας πόρους από «αργούς» εξυπηρέτες έχουμε την προσδοκία της μείωσης της μέσης καθυστέρησης κατά την εξυπηρέτηση μιας http αίτησης.
- Το μέσο μέγεθος των αντικειμένων για κάθε εξυπηρέτη πηγής. Ο περιορισμός χρόνου που υφίσταται μπορεί ,δεδομένου του περιορισμένου εύρους ζώνης που έχει ο γονέας μεσάζων εξυπηρέτης για την εξυπηρέτηση των αιτήσεων προανάκτησης, να μετατραπεί σε περιορισμό εύρους ζώνης. Άρα ο διαιτητής θα πρέπει να γνωρίζει το μέσο μέγεθος των αντικειμένων κάθε εξυπηρέτη έτσι ώστε να μπορεί να εκτιμήσει πόσα αντικείμενα μπορεί να προανακτήσει από τον κάθε

εξυπηρέτη μέσα στο ζητούμενο χρονικό διάστημα

Η λύση μπορεί να προσδιοριστεί μέσω συναρτήσεων ωφέλειας (utility functions). Οι συναρτήσεις ωφέλειας λαμβάνουν υπόψη το όφελος που προκύπτει για τον κάθε πελάτη. Έτσι λοιπόν κάθε πελάτης έχει ένα utility function που αναπαριστά το πόσο ευχαριστημένος είναι από την επιλογή των αντικειμένων προς προανάκτηση που κάνει ο arbitrator. Αυτό το utility function το ορίσαμε ως το άθροισμα των επιμέρους utility functions καθεμιά από τις οποίες είναι ένα μέτρο της ευχαρίστησης του πελάτη i από την προανάκτηση αντικειμένων από ένα συγκεκριμένο εξυπηρέτη πηγής j – ουσιαστικά είναι ένα μέτρο της αναμενόμενης μείωσης της εκλαμβανόμενης από αυτόν καθυστέρησης κατά την αίτηση κάποιου αντικειμένου από το συγκεκριμένο εξυπηρέτη j .

$$U_i = \sum u_{i,j} \tag{Εξ.5.1}$$

$$u_{i,j} = (w_{i,j} \cdot f_{i,j}) \cdot a_j \cdot d_j \cdot \sum_{l=1}^{w_{i,j} \cdot f_j} \frac{k}{l^z}$$

Επεξήγηση των όρων της (εξίσωσης 5.1):

$w_{i,j}$: αριθμός αντικειμένων που αιτείται ο πελάτης i από τον origin j στην αίτηση προανάκτησης του και προέκυψε από τον αριθμό των διαφορετικών αντικειμένων που ζήτησε ο πελάτης i από τον εξυπηρέτη j κατά το προηγούμενο time interval, $f_{i,j}$: ποσοστό των αντικειμένων που θα προανακτηθούν επί του αριθμού αυτών που ζητήθηκαν, a_j καλείται παράγοντας συνάφειας (coherency factor) και υποδηλώνει την ενδεχόμενη επανάληψη αιτήσεων για τον ίδιο πόρο από τον ίδιο ή διαφορετικούς πελάτες. d_j : μέση καθυστέρηση ανά λήψη αντικειμένου από τον origin j , a_j overlap factor, $\sum_{l=1}^{w_{i,j} \cdot f_j} \frac{k}{l^z}$ η πιθανότητα να ζητηθεί από μια συγκεκριμένη http αίτηση στον πελάτη i κάποιο από τα αντικείμενα που θα προανακτηθούν για αυτόν από τον origin j , δοθέντος ότι αυτή η αίτηση αιτείται κάποιο πόρο από τον origin j .

Ο παράγοντας $\sum_{l=1}^{w_{i,j} \cdot f_j} \frac{k}{l^z}$ προκύπτει από τον νόμο Zipf που καθορίζει την πιθανότητα $P(i)$ πρόσβασης στο i -οστό στοιχείο μίας ταξινομημένης, ως προς την συχνότητα προσπέλασης, ενδιάμεσης μνήμης ή ενός WWW site. Η συνάρτηση πυκνότητας πιθανότητας $P(i)$ προσδιορίζει το ποσοστό των αιτήσεων που αφορούν το στοιχείο i

Σχήμα Προφόρτωσης Περιεχομένου WWW

μετά την ταξινόμηση των στοιχείων ως προς την δημοτικότητα τους (popularity ranking). Ισχύει:

$$P(i) = ki^{-1} \quad (\text{Εξ.5.2})$$

Η σταθερά k εξασφαλίζει ότι το άθροισμα των πιθανοτήτων, για όλα τα θεωρούμενα στοιχεία-πόρους, είναι 1. Ο νόμος Zipf ισχύει σε διάφορες καταστάσεις όπως π.χ. η δημοτικότητα των λέξεων σε ένα κείμενο. Η γενική μορφή του νόμου Zipf είναι η ακόλουθη:

$$P(i) = ki^{-Z} \quad (\text{Εξ.5.3})$$

Ο όρος $\sum_{l=1}^{w_i \cdot f_j} \frac{k}{l^Z}$ προκύπτει από το άθροισμα των πιθανοτήτων για το πιο δημοφιλές τμήμα/ ποσοστό f_j του συνόλου w_j των πόρων που αιτείται ο πελάτης i . Η σταθερά Z είναι η σταθερά Zipf η οποία για τις περιπτώσεις των WWW ενδιάμεσων μηνυμάτων έχει προσδιοριστεί στην περιοχή 0.8 – 1.

Το $u_{i,j}$ είναι η ένα μέτρο του κατά πόσο ευχαριστημένος είναι ο πελάτης i αν επιλεχθούν από το διαιτητή $f_{i,j}$ ποσοστό επί του αριθμού των αντικειμένων που αυτός ζήτησε από τον origin j . Με βάση την παραπάνω επεξήγηση των όρων γίνεται σαφές ότι αυτή η ευχαρίστηση του πελάτη i είναι ουσιαστικά η αναμενόμενη μείωση της καθυστέρησης κατά την εξυπηρέτηση μιας αίτησης του προς τον origin j . Αυτό ισχύει διότι το $(w_{i,j} \cdot f_{i,j}) \cdot d_j$ είναι ουσιαστικά το μέγιστο ποσό καθυστέρησης που αναμένεται να γλιτώσει ο πελάτης i , αυτή δηλαδή που προκύπτει αν ζητήσει καθένα από τα αντικείμενα που θα προανακτηθούν από τον origin j για αυτόν. Η πιθανότητα αναφοράς όμως που συγκεντρώνουν τα αντικείμενα που θα προανακτηθούν από τον origin j είναι

$\sum_{l=1}^{w_{i,j} \cdot f_j} \frac{k}{l^Z}$ άρα η αναμενόμενη μείωση στην καθυστέρηση εξυπηρέτησης του πελάτη i για αιτήσεις του που έχουν σαν στόχο την αίτηση κάποιου αντικειμένου τον origin j είναι ανάλογη του γινομένου $(w_{i,j} \cdot f_{i,j}) \cdot d_j \cdot \sum_{l=1}^{w_{i,j} \cdot f_j} \frac{k}{l^Z}$ που αποτελεί ουσιαστικά το $u_{i,j}$.

Ανάλογη μείωση προκύπτει για τον πελάτη i από την προανάκτηση αντικειμένων από καθένα από τους εξυπηρέτες πηγής από τους οποίους αυτός αιτήθηκε προανάκτηση αντικειμένων στην αντίστοιχη αίτηση που έκανε προς τον γονέα. Το άθροισμα αυτών

Σχήμα Προφόρτωσης Περιεχομένου WWW

των αναμενόμενων μειώσεων στην καθυστέρηση εξυπηρέτησης για τον πελάτη i αποτελεί ουσιαστικά και την συναρτήσεων ωφέλειάς του (utility function). Έτσι προέκυψε και το $U_i = \sum u_{i,j}$.

Επιπλέον με κάθε εξυπηρέτη πηγής σχετίζεται ένα βάρος που έχει να κάνει με το πόσο εύρος ζώνης χρειάζονται τα αντικείμενα που θα προανακτήσουμε από αυτόν, ώστε να έρθουν μέσα στο χρονικά όρια που έχουν τεθεί ως περιορισμός. Συγκεκριμένα:

$$weight_j = (w_j \cdot F_j) \cdot s_j / t_j \quad (\text{Εξ.5.4})$$

Όπου w_j : Ο αριθμός των αντικειμένων που ζητούνται μέσω της mixed αίτησης προανάκτησης από τον origin j (έχει προέλθει από το $\max_i(w_{i,j})$) όπως είδαμε παραπάνω για το τρόπο συγχώνευσης δύο αιτήσεων προανάκτησης σε μία), F_j : το ποσοστό των αντικειμένων που θα προανακτηθούν από τον origin j , από αυτά που ζητούνται από τον origin j μέσω της μικτής (mixed) αίτησης προανάκτησης (w_j), s_j : το μέσο μέγεθος των αντικειμένων του origin j όπως αυτό είναι γνωστό στον μεσάζοντα εξυπηρέτη μέσω λήψεων αντικειμένων από αυτόν στο παρελθόν κατά την εξυπηρέτηση κανονικών http αιτήσεων, t_j : ο χρόνος μέσα στον οποίο θα πρέπει να έρθουν τα αντικείμενα που θα προανακτηθούν από τον origin j , έτσι όπως αυτός καθορίστηκε κατά την συγχώνευση των δύο αιτήσεων προανάκτησης.

Από τα παραπάνω προκύπτει ότι $(w_j \cdot F_j)$ είναι ο αριθμός των αντικειμένων που θα προανακτηθούν από τον origin j , $(w_j \cdot F_j) \cdot s_j$ είναι το συνολικό μέγεθος των αντικειμένων που θα προανακτηθούν από τον origin j , ενώ $(w_j \cdot F_j) \cdot s_j / t_j$ είναι το απαιτούμενο εύρος ζώνης για την προανάκτηση αυτόν των αντικειμένων μέσα στο χρονικό διάστημα που έχει τεθεί ως περιορισμός.

Ο διαιτητής θα επιλέξει να προανακτήσει ένα αριθμό αντικειμένων από κάθε origin που υπάρχει στην mixed αίτησης προανάκτησης. Παράμετροι με τις οποίες «παίζει» ο arbitrator είναι οι F_j που αποτελούν ποσοστά επί των αντίστοιχων w_j , δηλαδή του αριθμού των αντικειμένων που ζητούνται από τον κάθε origin μέσω της mixed αίτησης προανάκτησης. Θα πρέπει τα ποσοστά αυτά να επιλεγούν έτσι ώστε να μην παραβιάζονται οι χρονικοί περιορισμοί που έχουν τεθεί άρα θα πρέπει:

$$\sum_j weight_j \leq bw \quad (\text{Εξ.5.5})$$

όπου bw είναι το διαθέσιμο στον γονέα μεσάζοντα εξυπηρέτη εύρος ζώνης για την προανάκτηση αντικειμένων.

Τα ποσοστά F_j μεταφράζονται σε ποσοστά f_{ij} ώστε να μπορεί να υπολογιστεί η συνάρτηση ωφέλειας (utility function) του κάθε πελάτη ως εξής:

$$f_{i,j} = \min(1, F_j \cdot w_j / w_{i,j}) \quad (\text{Εξ.5.6})$$

Με το παραπάνω θέλουμε να μοντελοποιήσουμε το γεγονός ότι ένας πελάτης δεν επωφελείται από την προανάκτηση περισσότερων αντικειμένων από κάποιον συγκεκριμένο εξυπηρέτη από αυτά που ζήτησε μέσω της Top10 αίτησης προανάκτησης που έκανε. Αυτό συμβαίνει διότι κατά την κατασκευή της αίτησης προανάκτησης αυτού του πελάτη είχε ληφθεί υπόψη (από τον αλγόριθμο Top10) ο αριθμός των αντικειμένων που αυτός ζήτησε κατά το προηγούμενο χρονικό διάστημα από τον κάθε εξυπηρέτη. Από ένα εξυπηρέτη από τον οποίο ζήτησε λίγα αντικείμενα, αναμένεται να ζητήσει λίγα αντικείμενα και στον μέλλον, άρα η προανάκτηση παραπάνω αντικειμένων δεν αναμένεται να μειώσει στην εκλαμβανόμενη από αυτόν καθυστέρηση. Γι' αυτό το λόγο παραπάνω αντικείμενα που προανακτά ο γονέας μεσάζων εξυπηρέτης από κάποιον εξυπηρέτη για λογαριασμό κάποιου άλλου πελάτη που έχει ζητήσει περισσότερα αντικείμενα προς προανάκτηση από τον συγκεκριμένο εξυπηρέτη δεν προωθούνται στον πρώτο πελάτη που ζήτησε λιγότερα ,και δεν επηρεάζουν την τιμή της συνάρτησης ωφέλειας του.

5.2 Λειτουργία του διαιτητή με τη μέθοδο του σακιδίου

Προκειμένου ο arbitrator να προσδιορίσει το βέλτιστο ποσοστό που θα πρέπει να προανακτηθεί από κάθε εξυπηρέτη που αναφέρεται σε μια mixed αίτηση προανάκτησης προσπαθεί να επιλύσει το πρόβλημα του σακιδίου με τη μέθοδο του δυναμικού προγραμματισμού.

Συγκεκριμένα ο διαιτητής (arbitrator) δουλεύει ως εξής:

- Για κάθε εξυπηρέτη πηγής j χωρίζει το F_j σε percentiles
- Το συνολικό βάρος του σακιδίου χωρίζεται επίσης σε τόσα τμήματα ώστε το βάρος που αντιστοιχεί σε ένα percentile του sever j , για τον οποίο το $weight_j$ είναι

το ελάχιστο, να μην είναι μηδενικό. Όπως είδαμε το βάρος που συσχετίζεται με κάθε εξυπηρέτη είναι ουσιαστικά το αναμενόμενο εύρος ζώνης που απαιτείται για να προανακτηθεί ο ζητούμενος από την mixed αίτηση προανάκτησης αριθμός αντικειμένων από τον συγκεκριμένο εξυπηρέτη. Άρα το βάρος του σακιδίου που είναι και ο περιορισμός μας είναι ουσιαστικά το διαθέσιμο εύρος ζώνης για την προανάκτηση αντικειμένων από το συγκεκριμένο μεσάζοντα εξυπηρέτη.

- Οι εξυπηρέτες εξετάζονται σειριακά ένας-ένας. Πρώτα εξετάζονται με τη σειρά από το πρώτο έως το εκατοστό τα percentiles του πρώτου εξυπηρέτη μετά του δεύτερου κλπ.
- Για κάθε ένα από τα percentiles τώρα υπολογίζεται το κέρδος $g_j(F_j+0.01)$ που προκύπτει από την πρόσθεση του στην λύση (Το g_j ορίζεται παρακάτω και είναι ουσιαστικά η συνάρτηση υπολογισμού του επιπλέον κέρδους που προκύπτει για τους πελάτες αν προανακτηθεί ακόμα ένα percentile από τον εξυπηρέτη i). Έπειτα αυξάνεται το επιτρεπόμενο βάρος του σακιδίου από 0 μέχρι και τον περιορισμό που έχει τεθεί προσθέτοντας κάθε φορά και ένα επιπλέον «κομμάτι» βάρους το μέγεθος το οποίου υπολογίστηκε με τον τρόπο που αναφέρθηκε παραπάνω. Αυξάνοντας το βάρος αποφασίζεται την κάθε φορά αν το συγκεκριμένο percentile «χωράει» στην λύση για το συγκεκριμένο κάθε φορά βάρος. Αν όντως χωράει αποφασίζεται αν προσθέτοντας το συγκεκριμένο percentile το κέρδος που προκύπτει (το όφελος της βέλτιστης λύσης, για βάρος ίσο με το τρέχον μείον αυτό του συγκεκριμένου percentile, συν το $g_j(F_j+0.01)$) είναι μεγαλύτερο από το όφελος που προκύπτει από τα percentiles που έχουν επιλεγεί μέχρι τώρα για το συγκεκριμένο βάρος. Αν κάτι τέτοιο ο ισχύει τότε επιλέγεται το συγκεκριμένο percentile. Στις συντεταγμένες του διδιάστατου πίνακα που δηλώνουν ποιο percentile ποιανού εξυπηρέτη εξετάζεται και το τρέχον βάρος, γράφεται το συνολικό όφελος που προκύπτει από την μέχρι τώρα επιλογή percentiles για το συγκεκριμένο βάρος.
- Τέλος και αφού συμπληρωθεί ο πίνακας με τον τρόπο που προαναφέρθηκε διατρέχεται κατάλληλα ώστε να διαπιστωθεί τελικά ποια percentiles συνιστούν τη βέλτιστη λύση.

Η αύξηση του utility function του πελάτη i που προκύπτει αν προανακτηθεί ποσοστό F_j από τον αριθμό αντικειμένων που ζητούνται στην μικτή (mixed) αίτηση προανάκτησης από τον *origin* j , σε σχέση με την προανάκτηση του ποσοστού $F_j-0.01$ δίνεται από τον

παρακάτω τύπο.

$$\Delta U_1(F_j) = U_1(F_j) - U_1(F_j - 0.01) \quad (\text{Εξ.5.7})$$

Το επιπλέον κέρδος που προκύπτει για τους πελάτες αν προανακτηθεί ποσοστό F_j από τον αριθμό αντικειμένων που ζητούνται στην mixed αίτηση προανάκτησης από τον origin j , σε σχέση με την προανάκτηση του ποσοστού $F_j - 0.01$ δίνεται από τον παρακάτω τύπο και ουσιαστικά πρόκειται για την συνεισφορά του συγκεκριμένου percentile από τον εξυπηρέτη j στην αύξηση της ικανοποίησης των πελατών.

$$g_j(F_j) = \Delta U_1(F_j) + \Delta U_2(F_j) + \frac{\Delta U_1(F_j) \cdot \Delta U_2(F_j)}{\Delta U_1(F_j) + \Delta U_2(F_j)} \quad (\text{Εξ.5.8})$$

Με τον όρο $\frac{\Delta U_1(F_j) \cdot \Delta U_2(F_j)}{\Delta U_1(F_j) + \Delta U_2(F_j)}$ «πριμοδοτείται» η προανάκτηση πόρων που οδηγεί σε

παράλληλη αύξηση των utility functions και των δύο πελατών. (δηλαδή «πριμοδοτείται» η προανάκτηση πόρων από εξυπηρέτες πηγής που υπήρχαν στην αίτηση προανάκτησης και των δυο πελατών) Η προανάκτηση αντικειμένων από τέτοιους εξυπηρέτες πηγής έχει ως αποτέλεσμα την μείωση της αναμενόμενης καθυστέρησης παράλληλα για περισσότερους από ένα πελάτες με χωρίς την κατανάλωση επιπλέον εύρους ζώνης αφήνοντας έτσι το περιθώριο για προανάκτηση επιπλέον πόρων .

Με τον παραπάνω τρόπο επιλογής percentiles από κάθε origin τελικά θα προκύψει για κάθε εξυπηρέτη ένας αριθμός αντικειμένων που θα προανακτηθούν, ενώ παράλληλα θα πληρείται και ο περιορισμός χρόνου που έχει τεθεί. Αξίζει να παρατηρηθεί ότι το ποσοστό F_j που θα επιλεγεί τελικά από τον κάθε εξυπηρέτη δεν είναι απαραίτητα τέτοιο ώστε να μειώνεται η μέση εκλαμβανόμενη καθυστέρηση μεταξύ των πελατών. Κάτι τέτοιο θα συνέβαινε αν σε κάθε percentile ορίζαμε ως κέρδος από την προανάκτησή του το $g_j(F_j) = \Delta U_1(F_j) + \Delta U_2(F_j)$. Σε αυτή όμως την περίπτωση μεταξύ percentiles που έχουν το ίδιο κέρδος δεν δίνεται προτεραιότητα σε αυτό που δίνει όφελος ταυτόχρονα και στους δύο πελάτες από αυτό που δίνει μόνο στον ένα. Για παράδειγμα αν υποθέσουμε ότι έχουμε δύο πελάτες και δύο εξυπηρέτες και έστω ότι μπορούμε αν πάρουμε μόνο ένα percentile από ένα μόνο από τους δύο εξυπηρέτες λόγω των περιορισμών που έχουν τεθεί (χωρίς βλάβη της γενικότητας υποθέτουμε ότι η μόνη παράμετρος που διαφοροποιεί τα U_j σε αυτό το παράδειγμα είναι τα $w_{i,j}$ και d_j ενώ

όλες οι άλλες έχουν την ίδια τιμή). Αν $\Delta U_1(F_1) = 500$ και $\Delta U_2(F_1) = 0$ για $F_1 = 0.01$, και $\Delta U_1(F_2) = 210$, $\Delta U_2(F_2) = 210$ για $F_2 = 0.01$ τότε αν το κέρδος καθενός percentile υπολογιζόταν μόνο με βάση το άθροισμα των διαφορών στα utility functions όπως φαίνεται παραπάνω τότε θα γινόταν η επιλογή του πρώτου percentile του εξυπηρέτη 1. Αυτό μπορεί να δίνει την μέγιστη δυνατή μείωση της μέσης μεταξύ των πελατών καθυστέρησης, ωστόσο δεν μειώνει την καθυστέρηση παράλληλα για όσο το δυνατό μεγαλύτερο αριθμό πελατών. Η προτεινόμενη συνάρτηση κέρδους θα έχει ως αποτέλεσμα την επιλογή του πρώτου percentile από τον δεύτερο εξυπηρέτη. Σε αυτή την περίπτωση η μείωση στην μέση καθυστέρηση μεταξύ των πελατών δεν είναι τόσο μεγάλη (δεν είναι κοινωνικά βέλτιστη λύση) ωστόσο μειώνεται η καθυστέρηση για όλους τους πελάτες και όχι μόνο για ένα. Η λύση του αθροίσματος των διαφορών λοιπόν ενέχει τον κίνδυνο ένας μονό πελάτης να συγκεντρώσει όλο το όφελος. Μολονότι έτσι όντως μειώνεται στο μέγιστο δυνατό η μέση μεταξύ των πελατών καθυστέρησης, ωστόσο η συγκέντρωση όλου του οφέλους σε ένα πελάτη δεν είναι επιθυμητή.

Μια άλλη λύση θα ήταν η μεγιστοποίηση του γινομένου των utility functions των πελατών. Αυτό θα είχε σαν συνέπεια την πιο δίκαιη κατανομή του οφέλους από την προανάκτηση αντικειμένων μεταξύ των πελατών, ωστόσο δεν εξασφαλίζεται η μέγιστη μείωση της μέσης καθυστέρησης μεταξύ των πελατών. Για παράδειγμα αν υποθέσουμε ότι έχουμε τρεις πελάτες και δύο εξυπηρέτες και έστω ότι μπορούμε να πάρουμε συνολικά δύο percentiles από τους δύο εξυπηρέτες λόγω των περιορισμών που έχουν τεθεί. Έστω ότι οι πελάτες 2,3 ζητούν αντικείμενα από τον εξυπηρέτη 2 ενώ ο πελάτης 1 από τον εξυπηρέτη 1. Αν $\Delta U_1(F_1) = 100$ και $\Delta U_2(F_1) = 0$, $\Delta U_3(F_1) = 0$ για $F_1 = 0.01$, και $\Delta U_1(F_2) = 0$, $\Delta U_2(F_2) = 100$, $\Delta U_3(F_2) = 100$ για $F_2 = 0.01$ και $\Delta U_1(F_2) = 0$, $\Delta U_2(F_2) = 80$, $\Delta U_3(F_2) = 80$ για $F_2 = 0.02$. Τότε στην περίπτωση μεγιστοποίησης του γινομένου θα είχαμε την επιλογή του πρώτου percentile από καθένα από τους δύο εξυπηρέτες πράγμα που όντως είναι το δίκαιο. Ωστόσο μπορεί να επιτευχθεί μεγαλύτερη μείωση στην μέση μεταξύ των πελατών αντιλαμβανόμενης καθυστέρησης αν επιλεγούν τα δυο πρώτα percentiles από τον δεύτερο εξυπηρέτη και κανένα από τον πρώτο. Αυτό τον τελευταίο συνδυασμό επιλέγει το προτεινόμενο σχήμα διαιτησίας.

Παρατηρούμε λοιπόν ότι με την λύση που προτείνεται παίρνουμε κάτι ενδιάμεσο μεταξύ του κοινωνικά βέλτιστου -που όμως μπορεί όπως είδαμε να είναι πολύ άδικο, και του δίκαιου maxΠU το οποίο όπως είδαμε μπορεί να μειώνει λιγότερο από το προτεινόμενο

σχήμα την μέση μεταξύ των πελατών καθυστέρηση . Αυτό που παίρνουμε ως λύση είναι ποσοστά από κάθε εξυπηρέτη τέτοια ώστε να επιτευχθεί όσο το δυνατό μεγαλύτερη μείωση της μέση μεταξύ των πελατών καθυστέρησης για ένα όμως όσο το δυνατό μεγαλύτερο αριθμό πελατών. Με αυτή την λύση όπως είπαμε δεν εξασφαλίζεται η ελαχιστοποίηση της μέσης καθυστέρησης για τους πελάτες ως σύνολο (δεν είναι δηλαδή κοινωνικά βέλτιστη λύση) ούτε είναι απαραίτητα δίκαιη.

Μετά τον προσδιορισμό των ποσοστών F_j ζητούνται από τον γονέα μεσάζοντα εξυπηρέτη της ιεραρχίας τόσοι πόροι για προανάκτηση από κάθε εξυπηρέτη όσοι περιέχονται στη νέα arbitrated αίτηση.

5.3 Αποτελέσματα Προσομοίωσης

Για τους σκοπούς της προσομοίωσης του προτεινόμενου σχήματος διαιτησίας κατασκευάσαμε έναν event-driven προσομοιωτή χρησιμοποιώντας το εργαλείο J-sim (εργαλείο γραμμένο σε Java, βασίζεται στην *Autonomous Component Architecture*). Ο προσομοιωτής διαβάζει web traces της DEC [22] βάσει των οποίων δημιουργεί τα κατάλληλα events. Το πλήθος των αιτήσεων που χρησιμοποιήσαμε στις προσομοιώσεις ανέρχεται στις 550000.

Όταν καλείται ο αλγόριθμος αντικατάστασης LRU έχουμε ορίσει να απελευθερώνει το 20% του συνολικού χώρου της ενδιάμεσης μνήμης.

Συγκρίνουμε τρεις μηχανισμούς,

- έναν κατά τον οποίο έχουμε ενεργοποιημένο την ενδιάμεση αποθήκευση αλλά δεν γίνεται καθόλου προανάκτηση πόρων ,
- έναν κατά τον οποίο γίνεται προανάκτηση πόρων σύμφωνα με τον αλγόριθμο Top-10 χωρίς κανένα περιορισμό και τέλος,
- έναν κατά τον οποίο γίνεται προανάκτηση πόρων αλλά κάτω από περιορισμό χρόνου, ενώ ο περιορισμός της αίτησης προανάκτησης γίνεται με τον μηχανισμό διαιτησίας που περιγράψαμε
- Καταγράφουμε τα εξής στοιχεία:
 - Hit rate
 - Mean BW used (το μέσο εύρος ζώνης που ήταν απαραίτητο σε κάθε περίπτωση)

- Mean latency (μέση καθυστέρηση ανά πελάτη)

Arbitration/no prefetch			
	Hit rate	latency	Bandwidth
	1.025	0.959	1.017

Top10/no prefetch			
	Hit rate	latency	Bandwidth
	1.044	0.953	1.058

5.4 Παρατηρήσεις

Οι διαφορές είναι πολύ μικρές και αυτό οφείλεται σε μεγάλο μέρος στο ότι το Top-10 για κάθε εξυπηρετή πηγής καθορίστηκε από το συγκεκριμένο trace με το οποίο έγινε η προσομοίωση. Έτσι η προανάκτηση φέρνει αντικείμενα που ήδη έχουν ζητηθεί κατά το προηγούμενο time interval άρα με μεγάλη πιθανότητα υπάρχουν ήδη στην ενδιάμεση μνήμη. Παρόλα αυτά τα παραπάνω αποτελέσματα μας οδηγούν στα ακόλουθα συμπεράσματα

Μολονότι το Top10 επιφέρει μεγαλύτερη αύξηση στο λόγο επιτυχιών απ' ότι το σχήμα με διαιτησία, η επιπλέον μείωση που επέρχεται στην μέση καθυστέρηση των πελατών είναι ελάχιστη. Αυτό είναι μια ένδειξη ότι μολονότι στην περίπτωση του σχήματος με τη διαιτησία προανακτούνται λιγότεροι πόροι, επιλέγεται η προανάκτηση πόρων από εξυπηρετές πηγής από τους οποίους η λήψη ενός αντικειμένου είναι για τον οποιοδήποτε λόγο πιο αργή κατά μέσο όρο. Ο διαιτητής με άλλα λόγια

1. «κόβει» πόρους από εξυπηρετές πηγής από τους οποίους η καθυστέρηση λήψης μιας απάντησης είναι ήδη μικρή, άρα δεν επηρεάζεται σε μεγάλο βαθμό αρνητικά η μέση καθυστέρηση ανά πελάτη σε σχέση με την περίπτωση του Top10, μολονότι βέβαια όπως ήταν αναμενόμενο ο λόγος επιτυχιών μειώνεται
2. Φέρνει πόρους από εξυπηρετές πηγής από τους οποίους περισσότεροι πελάτες ζητούν προανάκτηση αντικειμένων. Έτσι μολονότι προανακτούνται λιγότεροι

πόροι, δεν επηρεάζεται τόσο αρνητικά η καθυστέρηση που αντιλαμβάνεται κάθε πελάτης

Επιπλέον παρατηρούμε ότι το απαιτούμενο εύρος ζώνης αυξάνεται παραπάνω στην περίπτωση του Top10 χωρίς να παρατηρείται αντίστοιχη μείωση στη μέση καθυστέρηση για κάθε πελάτη. Αυτό οφείλεται στο γεγονός ότι τα επιπλέον αντικείμενα που φέρνει ο μηχανισμός χωρίς διαιτησία είναι αντικείμενα για τα οποία ούτως ή άλλως η καθυστέρηση που αντιλαμβάνονταν ο χρήστης ήταν μικρή.

ΚΕΦΑΛΑΙΟ: 6 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα εργασία εξετάστηκε ένας μηχανισμός διαιτησίας για τον κατάλληλο περιορισμό αιτήσεων προανάκτησης πόρων που σχηματίζονται σύμφωνα με τον αλγόριθμο Top-10 ώστε να πληρείται ένας περιορισμός χρόνου όσον αφορά την ικανοποίηση της αίτησης προανάκτησης. Μελετήθηκαν οι πλέον δημοφιλείς προτάσεις - αρχιτεκτονικές WWW pre-fetching (όπως η TOP-10) καθώς και οι εσωτερικοί αλγόριθμοι αυτών. Ο προτεινόμενος μηχανισμός εφαρμόζεται σε συγκεκριμένη αρχιτεκτονική κόμβων στο πλαίσιο ενός ιεραρχίας ενδιάμεσων μνημών. Ο μηχανισμός προσπαθεί να μειώσει την εκλαμβανόμενη από τους χρήστες καθυστέρηση πριμοδοτώντας την προανάκτηση αντικειμένων από εξυπηρέτες που έχουν ζήτηση ταυτόχρονα από περισσότερους πελάτες ενώ παράλληλα χαρακτηρίζονται ως πιο «αργοί» κατά την εξυπηρέτηση http αιτήσεων. Βασικό στοιχείο για την υλοποίηση του μηχανισμού – αλγόριθμου αποτελεί η προτεινόμενη συνάρτηση ωφέλειας (utility function). Η συνάρτηση αυτή λαμβάνει υπόψη των αριθμό των διαφορετικών αντικειμένων που ζητά ο κάθε πελάτης από τον κάθε εξυπηρέτη στο προηγούμενο χρονικό διάστημα, την μέση καθυστέρηση για την ικανοποίηση μιας αίτησης από τον κάθε εξυπηρέτη καθώς και την μη γραμμική μορφή της κατανομής πυκνότητας πιθανότητας για την πρόσβαση στα στοιχεία μίας WWW ενδιάμεσης μνήμης (κανόνας Zipf).

Ο αλγόριθμος διαιτησίας προσομοιώθηκε για την αρχιτεκτονική τεσσάρων τερματικών κόμβων και δυο επιπέδων ιεραρχίας από μεσάζοντες ενδιάμεσης μνήμης. Τα αποτελέσματα της προσομοίωσης δείχνουν ότι ο προτεινόμενος μηχανισμός διαιτησίας κάνει επιλογές που είναι προς την σωστή κατεύθυνση όσον αφορά το στόχο μείωσης της εκλαμβανόμενης από τον χρήστη καθυστέρησης με δεδομένους τους περιορισμούς.

Σχήμα Προφόρτωσης Περιεχομένου WWW

Έτσι όπως βλέπουμε έχουμε για τις συγκεκριμένες παραμέτρους προσομοίωσης παρόμοια μείωση στην καθυστέρηση καταναλώνοντας παράλληλα λιγότερο εύρος ζώνης

ΑΚΡΩΝΥΜΙΑ

WWW	World Wide Web
HTTP	HyperText Transfer Protocol
URL	Uniform Resource Locator
LRU	Least Recently Used
LFU	Least Frequently Used
TTL	Time To Live
ICP	Internet Cache Protocol
CARP	Cache Array Resolution Protocol
CDP	Cache Digest Protocol
WCCP	Web Cache Coordination Protocol
UDP	User Datagram Protocol
DNS	Domain Name System
RTT	Round Trip Time
ISP	Internet Service Provider
NE	Nash Equilibrium
FIFO	First In First Out
ABR	Available Bit Rate
ATM	Asynchronous Transfer Mode
NBS	Nash Bargaining Solution
PoA	Price of Anarchy

ΟΡΟΛΟΓΙΑ

cache	ενδιάμεση μνήμη
origin server	εξυπηρετής πηγής
prefetching	προανάκτηση αντικειμένων
caching	αποθήκευση σε ενδιάμεση μνήμη
client	πελάτης
server	εξυπηρετής
bandwidth	εύρος ζώνης
browser	φυλλομετρητής
proxy server	μεσάζων εξυπηρετής
interception proxy	μεσάζων εξυπηρετής αναχαίτισης
cache hit	επιτυχία ενδιάμεσης μνήμης
content distribution	διανομή περιεχομένου
cache mirroring	κατοπτρισμός ενδιάμεσης μνήμης
transparent caching	διαφανής ενδιάμεση αποθήκευση
cache hit rate	λόγος επιτυχίας ενδιάμεσης μνήμης
lifetime	διάρκεια ζωής
price of anarchy	τιμή της αναρχίας
optimistic Price of anarchy	αισιόδοξη τιμή αναρχίας
pure Nash equilibrium	αμιγής ισορροπία Nash
mixed Nash equilibrium	μικτή ισορροπία Nash

ΑΝΑΦΟΡΕΣ

1. S. Hadjiefthymiades, S. Papayiannis, D. Martakos, "A Game Theoretic Approach to WWW Resource Prefetching"
2. Stathes Hadjiefthymiades, Yiannis Georgiadis, Lazaros F. Merakos: A Game Theoretic Approach to Web Caching. NETWORKING 2004: 1332-1337
3. F. Douglis, A. Feldman, B. Krishnamurthy, and J. Mogul. Rate of Change and Other Metrics: a Live Study of the World Wide Web. In Proc. of USENIX Symposium on Internet Technologies and Systems, Monterey, CA, Dec. 1997.
4. P. Barford and M. Crovella. Generating Representative Web Workloads for Network and Server Performance Evaluation. In ACM SIGMETRICS, pages 151–160, July 1998.
5. H.Yaïche, R.R. Mazumdar, C. Rosenberg, "A Game Theoretic Framework for Bandwidth Allocation and Pricing in Broadband Networks" IEEE/ACM TON, Vol.8, Issue 5, pp. 667 - 678 October 2000
6. Lazar, A. Orda and D. Pendarakis, "Virtual Path Bandwidth Allocation in Multiuser Networks", IEEE/ACM Transactions on Networking, Vol. 5, No. 6, pp.861-871, December 1997.
7. B. Duska, D. Marwood, M. Feeley, The measured access characteristics of world wide web client proxy caches, in: Proceedings of the USENIX Symposium on Internet Technologies and Systems, Monterey, CA, 1997, pp. 23–35
8. Markatos, E., Chronaki, C., "A Top-10 Approach to Prefetching on the Web," Tech. Rpt. No. 173, Aug. 1996, ICS-FORTH, Heraklion Crete, Greece.
9. E. Markatos, C. Chronaki, "A Top-10 Approach to Prefetching the web", Proceedings of INET '98 Geneva, Switzerland, July 1999.
10. Byung-Gon Chun, Kamalika Chaudhuri, Hoeteck Wee, Marco Barreno, Christos H. Papadimitriou, and John Kubiawicz, "Selfish caching in distributed systems: A gametheoretic analysis," in Proc. ACM Symposium on Principles of Distributed Computing (ACM PODC), Newfoundland, Canada, July 2004.
11. A.M. Glicksman, «An Introduction to Linear Programming and the Theory of Games», Dover Publications, 2001.
12. B. Krishnamurthy, and J. Rexford, «Web Protocols and Practice», Addison Wesley, 2001.
13. R.D. Luce, and H. Raiffa, «Games and Decisions: Introduction and Critical Survey», John Wiley & Sons, NY, 1957.

14. V. Padmanabhan, and J.C.Mogul, "Using Predictive Prefetching to Improve WWW Latency", ACM SIGCOMM Computer Communications Review, Vol. 26, No.3, July 1996.
15. S.J.Shenker, «Making Greed Work in Networks: A Game Theoretic Analysis of Switch Service Disciplines», IEEE/ACM Transactions on Networking, Vol.3, No.6, December 1995.
16. Venkataramani, P. Yalagandula, R. Kokku, S. Sharif, and M. Dahlin. Potential costs and benefits of long-term prefetching for content-distribution. Elsevier Computer Communications, Vol. 25, No. 4, pp.367–375, Mar. 2002.
17. G. Barish and K. Obraczka. World Wide Web Caching: Trends and Techniques. IEEE Communications Magazine, Vol. 38, No. 5, 2000, pp. 178-185
18. E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler, "Nearoptimal network design with selfish agents", in Proceedings of the 35th Annual ACM Symposium on the Theory of Computing, 2003, pp. 511-520.
19. M. Crovella and P. Barford. The Network Effects of Prefetching. In *Proc. Infocom 98*. IEEE, April 1998
20. Y. Jiang, M. Wu, W. Shu, "Web Prefetching: Costs, Benefits and Performance" Dep. of Electrical and Computer Engineering, University of New Mexico
21. Michael Rabinovich, Oliver Spatscheck, Web Caching and Replication, Addison Wesley, 2003.
22. Digital's Web Proxy Traces (V1.2 format) from <http://gatekeeper.dec.com/pub/DEC/traces/proxy/webtraces.v1.2.html>,
23. D. Foygel, D. Strelow, "Reducing Web Latency with Hierarchical Cache-based Prefetching", "ICPP Workshop, 2000, pp. 103-
24. M. F. Arlitt, C. L. Williamson, "Web Server Workload Characterization: The Search for Invariants", March 1996, pp.126-137
25. B. Wu, A. D. Kshemkalyani, "Objective-greedy Algorithms for Long-Term Web Prefetching," in *Proceedings of IEEE Network Computing and Applications Conference (NCA 04)*, pp. 61-68
26. Y.A. Korilis, A.A. Lazar, and A. Orda, «Architecting Noncooperative Networks», IEEE JSAC, Vol. 13, No. 8, 1995, pp.1241-1251
27. Brian E. Brewington, George Cybenko, How dynamic is the Web?, Proc. 9th Int'l WWW Conf Computer networks : the international journal of computer and telecommunications networking, p.257-276, June 2000, Amsterdam, The Netherlands.