ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ Επιστήμης Υπολογιστών

Ένα Σύστημα Αποδοτικής Μεταφοράς WWW Πληροφορίας βασισμένο σε Προτιμήσεις του Χρήστη και στην Εκτίμηση της Κατάστασης του Δικτύου

Μανόλης Πετσαγγουράκης

Μεταπτυχιακή Εργασία

Ηράκλειο, Οκτώβριος 1996

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ Επιστήμης Υπολογιστών

Ένα Σύστημα Αποδοτικής Μεταφοράς WWW Πληροφορίας βασισμένο σε Προτιμήσεις του Χρήστη και στην Εκτίμηση της Κατάστασης του Δικτύου

Εργασία που υποβλήθηκε από τον
Μανόλη Πετσαγγουράκη
ως μερική εκπλήρωση των απαιτήσεων
για την απόκτηση
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

Μανόλης Πετσαγγουράκης
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

Κώστας Κουρκουμπέτης
Καθηγητής, Επόπτης

Γιώργος Σταμούλης
Επίκουρος Καθηγητής, Μέλος

Πάνος Τραχανιάς
Επίκουρος Καθηγητής, Μέλος

Δεκτή:

Πάνος Κωνσταντόπουλος
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Οκτώβριος 1996

Στους γονείς μου,

Κωστή & Ειρήνη

Ευχαριστίες

θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή Κώστα Κουρκουμπέτη για την καθοδήγησή του στην εκπόνηση της εργασίας αυτής και γενικότερα, για την φιλική και παραγωγική συνεργασία μας. Ευχαριστώ επίσης τους καθηγητές μου Γιώργο Σταμούλη, για τις πολύτιμες συστάσεις του κατά τη διάρκεια της εργασίας και τις διορθώσεις του στην σύνταξη της παρούσας αναφοράς, και Πάνο Τραχανιά για την συμμετοχή του στην επιτροπή εξέτασης.

Ευχαριστίες θέλω να εκφράσω και στον φίλο και συνεργάτη Κώστα Παντερή για τις πολύωρες και γόνιμες συζητήσεις που είχαμε κατά τη διάρκεια της διαμόρφωσης της παρούσας εργασίας.

Ευχαριστώ ακόμα το Βασίλη Σύρη για την χρήσιμη ανταλλαγή απόψεων που είχαμε σε θέματα σχετικά με την εργασία καθώς και τη Μαρία Μαμαλάκη για την άμεση παροχή προνομίων υπερχρήστη στα συστήματα του Πανεπιστημίου Κρήτης που ήταν απαραίτητα για την πραγματοποίηση των πειραματικών μετρήσεων.

Ευχαριστώ τέλος το Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης για το υψηλό επίπεδο σπουδών που μου προσέφερε καθώς και το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για την υλικοτεχνική υποδομή που μου παρείχε.

Ένα Σύστημα Αποδοτικής Μεταφοράς WWW Πληροφορίας βασισμένο σε Προτιμήσεις του Χρήστη και στην Εκτίμηση της Κατάστασης του Δικτύου

Μανόλης Πετσαγγουράκης
Μεταπτυχιακή εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Ο Παγκόσμιος Δικτυακός Ιστός, γνωστότερος ως World Wide Web (WWW), είναι σήμερα η πιο διαδεδομένη αλλά και η πιο γρήγορα εξελισσόμενη εφαρμογή στο χώρο των υπολογιστών. Ένα μειονέκτημά του είναι η έλλειψη γνώσης των συνθηκών που επικρατούν στο δίκτυο. Έτσι δεν μπορεί να προσαρμόσει ανάλογα την συμπεριφορά του, με συνέπεια πολλές φορές να συμβάλει στην δημιουργία καταστάσεων συμφόρησης. Επίσης το HTTP, το πρωτόκολλο του WWW, έχει κάποιες εγγενείς αδυναμίες που προσθέτουν καθυστέρηση στην μεταφορά της πληροφορίας. Αποτέλεσμα των παραπάνω είναι η μείωση της απόδοσης που παρέχεται στο χρήστη.

Η παρούσα εργασία πραγματεύεται με τα δύο αυτά θέματα. Σχεδιάσαμε και υλοποιήσαμε ένα σύστημα που χρησιμοποιεί τροποποιημένους WWW client και server. Μια άλλη διεργασία που εκτιμά παραμέτρους του δικτύου (καθυστέρηση, απώλειες πακέτων, διέλευση), συνεργάζεται με τον WWW server. Η γνώση της κατάστασης του δικτύου που του παρέχει, χρησιμοποιείται για την επιλογή της πιο ‘‘κατάλληλης’’ μεταξύ των διαφορετικών μορφών της ίδιας πληροφορίας. Η επιλογή δεν εξαρτάται μόνο από την κατάσταση του δικτύου αλλά και από τις προτιμήσεις του χρήστη. Για το σκοπό αυτό ορίσαμε την έννοια του profile του χρήστη. Ο χρήστης μπορεί να δώσει διαφορετικές προτεραιότητες στις διαφορετικές μορφές της πληροφορίας που ζητά (π.χ video, audio, text), εισάγοντας και χρονικούς περιορισμούς. Επεκτείναμε το HTTP ώστε να είναι εφικτή η γνωστοποίησή του από τον WWW client στον server. Από τα πειράματα που κάναμε με ποικίλα σενάρια λειτουργίας του δικτύου και πληροφοριών σε πολλαπλούς τύπους, φάνηκε ότι το σύστημά μας ανταποκρίνεται πολύ καλά στις απαιτήσεις και τους περιορισμούς που θέτει ο χρήστης.

Επιπλέον στο HTTP αντικαταστήσαμε την τεχνική ‘‘σύνδεση ανά αίτηση’’ με ‘‘συνδέσεις διαρκείας’’. Με αυτόν τον τρόπο περισσότερα από ένα ζεύγη HTTP αιτήσεων/απαντήσεων εξυπηρετούνται από την ίδια σύνδεση, αποφεύγοντας το κόστος και την καθυστέρηση που προκαλείται με την εγκατάσταση/περάτωση πολλών συνδέσεων. Οι μετρήσεις που κάναμε με αντιπροσωπευτικές WWW σελίδες (ως προς το μέγεθος και τον αριθμό των ενσωματωμένων, σε αυτές, εικόνων), έδειξαν σημαντική βελτίωση της απόδοσης του HTTP.

Επόπτης : Κώστας Κουρκουμπέτης
Καθηγητής
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης.

A System for Efficient Transfer of WWW Information based on User Preferences and Network Status Estimation

Manolis Petsagourakis
Master of Science Thesis

Department of Computer Science
University of Crete

Abstract

World Wide Web (WWW) is the most common and faster evolved computer application nowadays. One of it's weaknesses is the lack of knowledge about the conditions that exists on the underlining network. So, it cannot adapt it's behavior appropriately, causing or contributing in congestion. Also HTTP, the protocol of WWW, has some inherent inefficiencies which add unnecessary delay on documents transfer. The above items results in a degradation of performance offered to the end users.

The present work deals with both of these issues. The system we designed and implemented uses a modified WWW client and server. In addition a process that estimates network parameters (delay, packet loss, throughput) cooperates with the WWW server. The knowledge of network status is used for the selection of the ''best'' among different types of the same information. The selection depends not only on network status but also on user preferences. For that reason we define the notion of user profile. This allows users to order their preference for the format of the document they wish to retrieve (e.g. video, audio, text), including time constrains. We extend the HTTP protocol enabling the WWW client to communicate it to server. A case study with various scenarios of network activity and information of multiple types, showed that our system responds properly on user demands and constrains.

Further more in the HTTP protocol we replaced the technique of one connection per request, with persistent connections. In this way, many HTTP requests/responses can be passed over the same connection, avoiding the overhead and latency caused by multiple connections setups. The measurements we have got with representative WWW pages (in size and number of inline images), showed a significant improvement of HTTP performance.

Advisor : Costas Courcoubetis
Professor
Department of Computer Science
University of Crete.

Περιεχόμενα

1 Εισαγωγή **1**

1.1 Στόχοι της Εργασίας 2

1.2 Το Περιεχόμενο της Εργασίας 3

2 WWW: Η Εξέλιξη, τα Χαρακτηριστικά και τα Δομικά Τμήματά του **5**

2.1 Τα Κύρια Χαρακτηριστικά του WWW 7

2.2 Η Γλώσσα HTML 10

2.3 Ομοιόμορφα Αναγνωριστικά Πόρων 11

2.4 Το Πρωτόκολλο HTTP 12

2.5 Προβλήματα στην Απόδοση του HTTP 16

2.5.1 Το Πρόβλημα Απόδοσης της Τεχνικής Σύνδεση ανά Αίτηση . . 16

2.5.2 Χρήση Υπολογιστικών Πόρων 17

2.5.3 Συνέπειες του Αλγορίθμου Αργής Εκκίνησης του TCP 18

2.6 Προσεγγίσεις στο Πρόβλημα Απόδοσης του HTTP 19

3 Εκτίμηση Κατάστασης Δικτύου **21**

3.1 Δυσκολίες στον Υπολογισμό Παραμέτρων του Δικτύου 21

3.2 Το Πρωτόκολλο ICMP 23

3.3 Σχετικές Εργασίες 24

4 Το Προτεινόμενο Σύστημα **26**

4.1 Η Αρχιτεκτονική του Συστήματος 27

4.2 Ο Αλγόριθμος Επιλογής 29

4.3 Καθορισμός Προτιμήσεων Χρήστη 31

4.3.1 Η Επικεφαλίδα UserProfile 31
4.4 Μέθοδος Εκτίμησης Throughput 33
4.5 HTTP Συνδέσεις Διαρκείας 37
4.5.1 Η Επικεφαλίδα Connection 37
4.5.2 Η Υλοποίηση των Συνδέσεων Διαρκείας 38
4.6 Θέματα Συμβατότητας 40

5 Αξιολόγηση - Μετρήσεις **41**
5.1 Μετρήσεις Παραμέτρων Δικτύου 42
5.2 Μελέτη Σεναρίων Χρήσης του Συστήματος 45
5.3 Η Απόδοση του WWW με την Χρήση Συνδέσεων Διαρκείας 49

6 Συμπεράσματα - Μελλοντική Εργασία **53**
6.1 Συμπεράσματα 53
6.2 Μελλοντική Εργασία 54

A Θέματα Σχεδίασης και Υλοποίησης **56**
A.1 Η Γεννήτρια Κυκλοφορίας 56
A.2 Η Δομή Καταχώρησης Throughput 58

B Σύγκριση Πραγματικού και Εκτιμούμενου Throughput Δικτύου **60**

Ευρετήριο Όρων **69**

Βιβλιογραφία **71**

Κατάλογος Σχημάτων

2.1 Αριθμός κόμβων στο Internet 6

2.2 Proxy servers 9

2.3 Το HTTP στη στοίβα των πρωτοκόλλων 13

2.4 Τα μηνύματα του HTTP 14

2.5 Καθυστέρηση των HTTP μηνυμάτων λόγω της τεχνικής σύνδεση ανά αίτηση 17

2.6 Στοιχεία σελίδων του WWW 19

3.1 Μη συμμετρική δρομολόγηση 22

3.2 Ενσωμάτωση των ICMP μηνυμάτων στα IP πακέτα 23

3.3 ICMP Echo Request/Reply μήνυμα 24

3.4 Υπολογισμός διαθέσιμου bandwidth συνδέσμου 24

3.5 Ροή πακέτων μέσα από περιοριστικό σύνδεσμο 25

4.1 Η αρχιτεκτονική του συστήματος 28

4.2 Ο αλγόριθμος επιλογής 30

4.3 Πιθανά στιγμιότυπα ακολουθιών ICMP Echo Request/Reply μηνυμάτων 34

4.4 Πίνακας αναχωρήσεων - αφίξεων 34

4.5 Διαγράμματα καταστάσεων 38

5.1 Τοπολογία Δικτύου 41

5.2 Μετρήσεις μεταξύ Ηρακλείου-Θεσσαλονίκης 44

A.1 Σύνδεση μεταφοράς πραγματικής κυκλοφορίας 57

A.2 Σημεία λήψεως χρόνου 57

A.3 Η δομή αποθήκευσης της πληροφορίας κατάστασης δικτύου 58

B.1 Μετρήσεις μεταξύ Ηράκλειου-Θεσσαλονίκης 63
B.2 Μετρήσεις μεταξύ Ηρακλείου-Γαλλίας 64
B.3 Μετρήσεις μεταξύ Ηράκλειου-Φινλανδίας 65
B.4 Μετρήσεις μεταξύ Ηρακλείου-Μ. Βρετανίας 66
B.5 Μετρήσεις μεταξύ Ηρακλείου-ΗΠΑ 67
B.6 Μετρήσεις μεταξύ Ηρακλείου-Αυστραλίας 68

Κατάλογος Πινάκων

2.1 Αριθμός WWW κόμβων στο Internet 6

2.2 Ποσοστό κυκλοφορίας ανά εφαρμογή στο NSFNET 7

5.1 Πληροφορίες διαφόρων τύπων 46

5.2 Χαρακτηριστικά WWW σελίδων που χρησιμοποιήθηκαν 50

5.3 Χρόνοι μεταφοράς πληροφορίας με συνδέσεις διαφορετικών τύπων (Ι.Π. - Π.Κ) 51

5.4 Χρόνοι μεταφοράς πληροφορίας με συνδέσεις διαφορετικών τύπων (FORTHnet - Π.Κ) 51

Β.1 Συγκεντρωτικά αποτελέσματα μετρήσεων 61

Κεφάλαιο 1

Εισαγωγή

Ο Παγκόσμιος Δικτυακός Ιστός, γνωστότερος ως World Wide Web (WWW), αποτελεί στις μέρες μας την πιο διαδεδομένη εφαρμογή των υπολογιστών. Η επιτυχία του οφείλεται κύρια στο μεγάλο εύρος της πληροφορίας που μπορεί να παράσχει και στην διαφάνεια (tranparency) για το χρήστη ως προς τη διαδικασία που χρησιμοποιεί για τη μεταφορά των δεδομένων, καθώς και το μέρος που αυτά βρίσκονται αποθηκευμένα. Η έρευνα στον τομέα αυτόν εστιάζεται σήμερα στην διεύρυνση των δυνατοτήτων του WWW, στην αναζήτηση μηχανισμών για την αναβάθμιση της ποιότητας των υπηρεσιών του και την βελτίωση της απόδοσής του. Στα πλαίσια της προσπάθειας αυτής εντάσσεται και η παρούσα εργασία.

Τα βασικότερα συμπεράσματα που αφορούν τον WWW και άπτονται της παρούσας εργασίας συνοψίζονται στα ακόλουθα:

- Η εκρηκτική αύξηση της κυκλοφορίας στο δίκτυο, ως συνέπεια της επιτυχίας του WWW, έχει σαν αποτέλεσμα την εμφάνιση του φαινομένου της συμφόρησης με ολοένα και μεγαλύτερη συχνότητα. Η αύξηση της χωρητικότητας των συνδέσμων επικοινωνίας δεν αποτελεί μακροπρόθεσμη λύση γιατί ούτε μπορεί να ακολουθήσει την αύξηση της ζήτησης αλλά μπορεί αυτό το ίδιο το γεγονός να επιτείνει το πρόβλημα σε μονοπάτια που υπάρχουν πιο αργοί σύνδεσμοι.

- Στον WWW δεν υπάρχει γνώση των συνθηκών που επικρατούν στο δίκτυο και συνεπώς δεν μπορεί να προσαρμόσει την συμπεριφορά του ανάλογα.

- Στο HTTP, το πρωτόκολλο του WWW, δεν υπάρχει δυνατότητα ο χρήστης να ορίσει προτιμήσεις-προτεραιότητες βάσει των οποίων θα γίνει η επιλογή της πληροφορίας που θα πάρει. Κάτι τέτοιο θα ήταν πολύ σημαντικό για λόγους μικρότερης χρέωσης ή καθυστέρησης.

- Το HTTP είναι το πρωτόκολλο με την μεγαλύτερη χρήση στο Internet. Η απόδοση του ωστόσο δεν είναι η βέλτιστη δυνατή, όσον αφορά το χρόνο μεταφοράς της πληροφορίας, με συνέπεια να μειώνεται η ποιότητα υπηρεσιών που λαμβάνει ο χρήστης.

Τα παραπάνω στοιχεία οριοθετούν τα πλαίσια της παρούσας εργασίας. Το ζητούμενο είναι *η υλοποίηση ενός συστήματος αποδοτικής μεταφοράς WWW πληροφορίας με δυναμική αναπροσαρμογή σύμφωνα με τις πληροφορίες που του παρέχει μια μέθοδος εκτίμησης παραμέτρων του δικτύου, βασισμένο παράλληλα στις προτιμήσεις που ορίζει ο χρήστης.*

1.1 Στόχοι της Εργασίας

Βασικοί στόχοι της εργασίας αυτής είναι:

- Η επέκταση του πρωτοκόλλου του WWW με την εισαγωγή προτιμήσεων από το χρήστη για την ποιότητα της πληροφορίας που είναι διατεθειμένος να δεχτεί ή το χρόνο που μπορεί να περιμένει για να γίνει αυτό.
- Η κατά προσέγγιση εκτίμηση της κατάστασης του δικτύου μεταξύ των σημείων που θα μεταφερθεί η πληροφορία, με ένα αποδοτικό τρόπο και σε πραγματικό χρόνο.
- Η χρήση της παραπάνω γνώσης για την επιλογή του είδους της πληροφορίας που τελικά θα αποσταλεί.
- Η τροποποίηση της διαδικασίας εγκατάστασης HTTP συνδέσεων για την ταχύτερη μεταφορά πληροφορίας από τον αποστολέα στον παραλήπτη.

Για την επίτευξη των στόχων αυτών υλοποιήσαμε το πρωτότυπο ενός τέτοιου συστήματος που αποτελείται: α) από τον διαχειριστή της πληροφορίας (server) στον οποίο έχουμε ενσωματώσει την διαδικασία επιλογής της πληροφορίας που θα σταλεί, β) την διεργασία υπολογισμού (estimator) κάποιων παραμέτρων του δικτύου και γ) τον αντιπρόσωπο χρήστη (client) ο οποίος συνδέεται στον κατάλληλο διαχειριστή και προωθεί την αίτηση και τις προτιμήσεις-περιορισμούς του χρήστη.

Τα κύρια θέματα που μας απασχόλησαν κατά το σχεδιασμό και την υλοποίηση του συστήματος είναι:

1. Ο τρόπος καθορισμού των παραμέτρων για την έκφραση των προτιμήσεων του χρήστη και ο μηχανισμός γνωστοποίησής τους στο διαχειριστή της πληροφορίας.
2. Ο έγκαιρος υπολογισμός του επιπέδου λειτουργίας (καθυστέρηση, ποσοστό απώλειας δεδομένων, διέλευση) του παρεμβαλλόμενου δικτύου, με την μικρότερη δυνατή επιβάρυνση.
3. Η αποτελεσματική ενημέρωση του διαχειριστή για την κατάσταση του δικτύου με την ελαχιστοποίηση του χρόνου αναμονής υπολογισμού και του κόστους επικοινωνίας.

4. Ο εφοδιασμός του διαχειριστή της πληροφορίας με έναν αλγόριθμο εύρεσης της κατάλληλης πληροφορίας με παραμέτρους τα στοιχεία των 1, 2 και των εναλλακτικών δυνατοτήτων που ο ίδιος μπορεί να παράσχει.

5. Η βελτίωση της απόδοσης του HTTP με τη δημιουργία TCP συνδέσεων που εξυπηρετούν περισσότερα από ένα, ζεύγη HTTP αιτήσεων/απαντήσεων (συνδέσεις διαρκείας).

Με την ολοκλήρωση του συστήματος θέλουμε ακόμα να πετύχουμε τα εξής:

- Η λειτουργία όλου του συστήματος να βρίσκεται στο επίπεδο των εφαρμογών όπου ανήκει και το HTTP.
- Να είναι συμβατό με τις υπάρχουσες υλοποιήσεις που δεν υποστηρίζουν τις προτεινόμενες αλλαγές και επεκτάσεις.
- Να κάνει καλύτερη χρησιμοποίηση του δικτύου με τη μεταφορά της πληροφορίας που ‘‘αντιστοιχεί’’ στην εκτιμούμενη κατάσταση αλλά και με την εισαγωγή λιγότερων μηνυμάτων ελέγχου εξαιτίας του πρωτοκόλλου. Με τον τρόπο αυτό έμμεσα μειώνεται η πιθανότητα συμφόρησης.
- Η διεργασία εκτίμησης της κατάστασης του δικτύου να πληροί τις ακόλουθες προϋποθέσεις:
  - -- Ανεξαρτησία από την συνεργασία του δικτύου ή του προς έλεγχο κόμβου
  - -- Ελάχιστη επιβάρυνση τόσο στο δίκτυο όσο και στον προς έλεγχο κόμβο.
  - -- Σταθερότητα και σωστή λειτουργία κάτω από διαφορετικές συνθήκες δικτύου .
  - -- Ακριβής και έγκαιρη παροχή της υπολογιζόμενης πληροφορίας.

1.2 Το Περιεχόμενο της Εργασίας

Η εργασία ξεκινάει με μια αναφορά στην εξέλιξη και τα χαρακτηριστικά του WWW και μια αναλυτική περιγραφή των δομικών τμημάτων από τα οποία συντίθεται (κεφάλαιο 2). Στις ενότητες 2.5, 2.6 παραθέτουμε τα προβλήματα που υπάρχουν στην απόδοση του WWW και κάνουμε μια ανασκόπηση των προτάσεων που έχουν γίνει για την επίλυση τους. Στο επόμενο κεφάλαιο αναφερόμαστε σε θέματα σχετικά με την εκτίμηση της κατάστασης του δικτύου, ποια εμπόδια συναντώνται, ποιο πρωτόκολλο έχει αναπτυχθεί για το σκοπό αυτό και ποιες προσεγγίσεις σχετικές με την παρούσα εργασία υπάρχουν.

Στο κεφάλαιο 4 παρουσιάζουμε το σύστημα το οποίο αναπτύξαμε για την επίτευξη των στόχων που έχουμε θέσει. Οι ενότητες του πραγματεύονται την αρχιτεκτονική και τις βασικές λειτουργίες του, περιγράφουν τη μέθοδο για τον υπολογισμό των παραμέτρων του δικτύου, τον αλγόριθμο επιλογής πληροφορίας καθώς και την

υλοποίηση των συνδέσεων διαρκείας. Εκτενέστερη αναφορά σε θέματα σχεδίασης και υλοποίησης γίνεται στο παράρτημα Α.

Για την αξιολόγηση του συστήματος έγιναν πειράματα με ποικίλα σενάρια περιπτώσεων, παρατηρήσεις για τις επιλογές του, μετρήσεις για την απόδοση του συστήματος και συγκρίσεις με το υπάρχον μοντέλο. Η παρουσίαση και ανάλυση των αποτελεσμάτων γίνεται στο κεφάλαιο 5. Η εργασία κλείνει με το κεφάλαιο 6, στο οποίο αναφέρονται τα συμπεράσματα που αποκομίσαμε και γίνονται κάποιες προτάσεις για μελλοντική εργασία.

Τέλος στο παράρτημα Β παρατίθενται τα αποτελέσματα μιας σειράς από πειράματα που έγιναν για να επαληθεύσουμε την ορθότητα της μεθόδου για την εκτίμηση της κατάστασης του δικτύου.

Κεφάλαιο 2

WWW: Η Εξέλιξη, τα Χαρακτηριστικά και τα Δομικά Τμήματά του

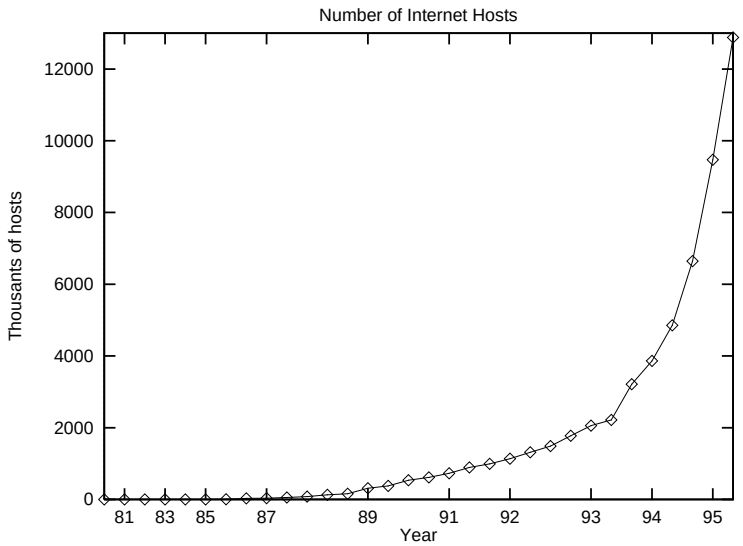
Το μεγαλύτερο δίκτυο δεδομένων στον κόσμο είναι το Internet (INTERnetworking NETworks). Βασίζεται στη στοίβα πρωτοκόλλων TCP/IP, είναι μικτής τοπολογίας και συντίθεται από τη διασύνδεση των πολύ γρήγορων τοπικών δικτύων με συνδέσμους μείζονος περιοχής. Οι ταχύτητες που χρησιμοποιούνται για τη διασύνδεση των πολύ γρήγορων τοπικών δικτύων (Ethernets, FDDIs) ξεκινούν από πολύ χαμηλές, της τάξεως των δεκάδων Kbits/sec και φτάνουν τα 155 Mbits/sec. Η εξέλιξη της χρησιμοποιούμενης τεχνολογίας (ATM) επιτρέπει την χρήση ακόμα μεγαλύτερων ταχυτήτων.

Τα δίκτυα υπολογιστών και η λειτουργία τους εξαρτώνται από ένα σύνολο πολλών και ποικίλων παραγόντων. Οι τοπολογίες, οι ταχύτητες, οι αλγόριθμοι που διέπουν την ροή της πληροφορίας (αλγόριθμοι δρομολόγησης, ελέγχου ροής κ.α), τα είδη των εφαρμογών και η κυκλοφορία που παράγουν στα δίκτυα, είναι μερικοί από τους παράγοντες αυτούς.

Τα τελευταία τέσσερα χρόνια το Internet, έχει γνωρίσει μια αλματώδη εξέλιξη. Με βάση μετρήσεις [Lot92] που έχουν γίνει ο αριθμός των κόμβων (hosts) που συνδέονται σε αυτό αυξήθηκε από 1.7 εκατομμύρια το 1993, σε 12.8 εκατομμύρια στα μέσα του 1996 (σχήμα 2.1).

Υπήρξε δηλαδή μια ποσοστιαία αύξηση της τάξης του 750%. Στηριζόμενοι στα παραπάνω και χρησιμοποιώντας την πιο *συντηρητική* μέθοδο (αντιστοιχώντας ένα χρήστη ανά κόμβο) προκύπτει ότι οι χρήστες του Internet είναι τουλάχιστον 13 εκατομμύρια. Στην πραγματικότητα φυσικά ο αριθμός τους είναι πολύ μεγαλύτερος.

Ένας από τους βασικούς λόγους που ώθησαν την εξάπλωση του Internet είναι η εφαρμογή World Wide Web (WWW). Αυτό αποδεικνύεται από τα στοιχεία του πίνακα 2.1, όπου φαίνεται ο αριθμός των WWW κόμβων στο Internet από τα μέσα του 1993 και μετά. Σε διάστημα τριών ετών ο αριθμός των WWW κόμβων αυξήθηκε από 130 σε 230000!!! Παρατηρούμε δηλαδή ένα εκθετικό ρυθμό αύξησης αυτών, με περίοδο διπλασιασμού κάτω των 6 μηνών ακόμα και σήμερα.



Σχήμα 2.1: Αριθμός κόμβων στο Internet

Ημερομηνία	Αριθμός WWW κόμβων
6/93	130
12/93	623
6/94	2738
12/94	10022
6/95	23500
1/96	100000
6/96	230000

Πίνακας 2.1: Αριθμός WWW κόμβων στο Internet

Παρόμοια εικόνα προκύπτει και από το ποσοστό της κυκλοφορίας (πίνακας 2.2) που αναλογεί σε διάφορες εφαρμογές, που βρίσκονται ‘‘πάνω’’ από το TCP πρωτόκολλο, στο δικτυακό κορμό NSFNET, των Ηνωμένων Πολιτειών (στοιχεία μέχρι 3/95 [Mer95]) [1]. Όπως φαίνεται στον πίνακα 2.2 πριν από τρία χρόνια η εφαρμογή μεταφοράς αρχείων (FTP), εισήγαγε το 42.9% της συνολικής κυκλοφορίας ενώ κατά σειρά ακολουθούσαν η εφαρμογές μεταφοράς news (9.3%), ηλεκτρονικού ταχυδρομείου (6.4 %), πρόσβασης σε απομακρυσμένους κόμβους (5.6%) και τελευταίος ο WWW με ποσοστό μόλις 0.5%. Στη συνέχεια όμως έχουμε μια σταδιακή μείωση της κυκλοφορίας όλων των εφαρμογών, με κύρια εκείνη της εφαρμογής FTP, εκτός από την κυκλοφορία του WWW που παρουσιάζει μια εντυπωσιακή αύξηση. Σήμερα, κρίνοντας και από την καθημερινή μας εμπειρία, η κυκλοφορία του WWW πρέπει να έχει υπερκαλύψει όλες τις υπόλοιπες.

Ο World Wide Web είναι ένα κατανεμημένο σύστημα πληροφοριών βασισμένο

[1]Το NSFNET έπαψε να είναι ο δικτυακός κορμός των ΗΠΑ τον Απρίλιο του 1995 και αντικαταστάθηκε από το vBNS (very high speed Backbone Network Service).

Ημερομηνία	FTP (%)	TELNET (%)	NNTP (%)	SMTP (%)	**WWW** (%)
6/93	42.9	5.6	9.3	6.4	**0.5**
12/93	40.9	5.3	9.7	6.0	**2.2**
6/94	35.2	4.8	10.9	6.4	**6.1**
12/94	31.7	3.9	10.9	5.6	**16.0**
3/95	24.2	2.9	8.3	4.9	**23.9**

Πίνακας 2.2: Ποσοστό κυκλοφορίας ανά εφαρμογή στο NSFNET

στην τεχνολογία των υπερκειμένων (hypertexts), το οποίο ξεκίνησε από ερευνητές στο CERN (Conseil Europeen pour la Recherche Nucleaire) της Ελβετίας το 1990. Το κύριο χαρακτηριστικό του είναι η διασύνδεση της διασκορπισμένης σε ολόκληρο το δίκτυο πληροφορίας, μέσω συνδέσμων υπερκειμένου. Αποτελεί έτσι ένα εξαιρετικά ισχυρό αλλά και εύχρηστο σύστημα περιπλάνησης και εύρεσης πληροφοριών ανά τον κόσμο.

Μέσα σε ελάχιστο χρονικό διάστημα έγινε αποδεκτός από την επιστημονική και όχι μόνο, κοινότητα [PR95] και σήμερα είναι η πιο διαδεδομένη εφαρμογή. Εξάλλου, ο WWW, συνιστά στις μέρες μας ένα από τα σημεία αιχμής της έρευνας όχι μόνο στον τομέα των υπολογιστών αλλά και στην οικονομία, τη διαφήμιση, το εμπόριο, την εκπαίδευση κτλ.

Τα βασικά δομικά του κομμάτια είναι τρία. Η **γλώσσα HTML**, τα **Ομοιόμορφα Αναγνωριστικά Πόρων** (URLs) και το **πρωτόκολλο HTTP**. Η γλώσσα HTML περιγράφει τις δυνατότητες σύνταξης των υπερκειμένων (hypertexts), το πρωτόκολλο HTTP καθορίζει την διαδικασία μεταφοράς που το δίκτυο ακολουθεί όταν ο χρήστης επιλέξει κάποιο αντικείμενο, ενώ το σύστημα των URLs προσδιορίζει κάθε αντικείμενο στο ‘‘χώρο’’ του WWW, δίνοντας του μια μοναδική ‘‘διεύθυνση’’. Για να μπορέσει λοιπόν ο WWW να λειτουργήσει και να αναπτυχθεί, πρέπει όλοι να ακολουθούν τις παραπάνω προδιαγραφές και να συμφωνούν στις αλλαγές που γίνονται σε αυτές. Γι’ αυτό ακριβώς το λόγο έχουν δημιουργηθεί ειδικές ομάδες εργασίας υπό την αιγίδα του International Standards Organisation, οι οποίες έχουν βασικό στόχο την επέκταση αλλά και προτυποποίηση των δομικών λίθων του WWW. Σε επόμενες ενότητες περιγράφουμε εκτενέστερα τον ρόλο και τη λειτουργία καθενός απ’ αυτούς.

2.1 Τα Κύρια Χαρακτηριστικά του WWW

Όπως ήδη αναφέραμε ο WWW επιτυγχάνει την διασύνδεση της κατανεμημένης στο δίκτυο πληροφορίας και παρέχει ένα εύκολο τρόπο αναζήτησής της. Τα άλλα χαρακτηριστικά του WWW που συνέβαλαν στην επιτυχία του είναι:

1. Η συνεργασία ή ενοποίηση με ένα πλήθος άλλων εφαρμογών.

Μέσω του WWW και με τη χρήση κατάλληλων URLs μπορούμε να μεταφέρουμε αρχεία από απομακρυσμένους ftp servers (συνεργασία με το πρωτόκολλο FTP [PR85]), να ανταλλάξουμε μηνύματα ηλεκτρονικού ταχυδρομείου (συνεργασία με το πρωτόκολλο SMTP [Pos82]), να διαβάσουμε news από τις λίστες του Usenet (συνεργασία με το πρωτόκολλο NNTP [CL86]), να κάνουμε απομακρυσμένη πρόσβαση (συνεργασία με το πρωτόκολλο TELNET [PR83]) και άλλα.

2. Η εκφραστικότητα και ο τρόπος παρουσίασης της πληροφορίας που παρέχει η γλώσσα HTML.

   Παρόλο που η γλώσσα HTML είναι αρκετά απλή και εύκολη έχει πολύ μεγάλες δυνατότητες όσον αφορά την παρουσίαση της πληροφορίας. Έτσι υποστηρίζει: 1. πολλαπλά μεγέθη και χρώματα γραμμάτων, 2. λίστες, 3. πολύπλοκους πίνακες, 4. φόρμες (forms) με ποικίλες επιλογές, 5. εικόνες ενσωματωμένες (inline) σε οποιοδήποτε σημείο του κειμένου, 6. ‘‘παράθυρα’’ (frames) με πληροφορίες, που διαχειρίζονται διαφορετικά το ένα από το άλλο, 7. μαθηματικούς τύπους, και φυσικά τους συνδέσμους υπερκειμένου οι οποίοι επιτρέπουν τη διασύνδεση της πληροφορίας οπουδήποτε στο δίκτυο και αν βρίσκεται αυτή. Όλα αυτά βέβαια δημιουργούν ένα πιο ελκυστικό περιβάλλον για τους χρήστες, σε σχέση με τα προηγούμενα, απλά και βασισμένα μόνο σε κείμενο, συστήματα πληροφοριών (πχ. WAIS [DKM+90]).

3. Η δυναμική παραγωγή πληροφορίας.

   Ο WWW μέσω του προτύπου Common Gateway Interface (CGI) μπορεί να συνεργάζεται με εξωτερικά προγράμματα τα οποία παράγουν δυναμικά την πληροφορία και την επιστρέφουν στο χρήστη. Δίνεται έτσι η δυνατότητα αποδέσμευσης από τα πλαίσια που θέτει το πρωτόκολλο HTTP (στο οποίο θα αναφερθούμε στη συνέχεια), οπότε μπορούν να παρουσιαστούν όχι μόνο ‘‘στατικά κείμενα’’ αλλά και πληροφορίες που πιθανόν να μεταβάλλονται από χρήστη σε χρήστη, ανάλογα με τις απαιτήσεις του. Με τον τρόπο αυτό για παράδειγμα, επεξεργάζονται τα στοιχεία που δίνονται κατά τη διάρκεια των ηλεκτρονικών εμπορικών συναλλαγών.

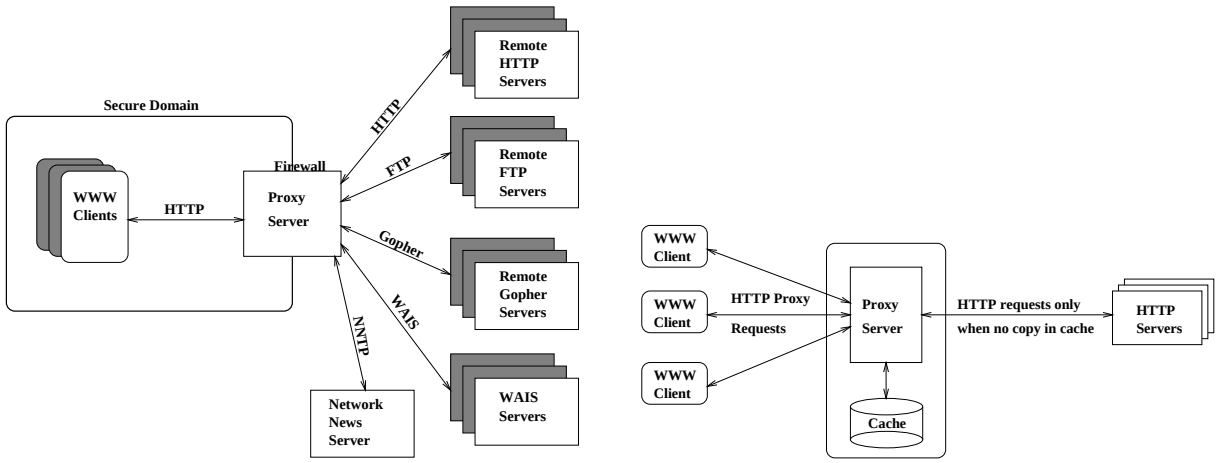
4. Η δυνατότητα για αναζήτηση στο χώρο του WWW.

   Με την βοήθεια ειδικών προγραμμάτων, των ‘‘μηχανών’’ αναζήτησης (search engines) ή των ψηφιακών βιβλιοθηκών (digital libraries), μπορούμε να ψάξουμε για οποιοδήποτε πληροφορία ορίζοντας περιορισμούς ή λογικές συνθήκες μεταξύ των ‘‘κλειδιών’’ (keywords), βάσει των οποίων θα γίνει η αναζήτηση. Αυτή μπορεί να γίνει είτε στατικά (δηλ. από κάποιο ήδη φτιαγμένο ευρετήριο), είτε δυναμικά (δηλ. με αφετηρία κάποιο σημείο ακολουθείται κάθε φορά ένας νέος σύνδεσμος υπερκειμένου, με βάση κάποια κριτήρια προτεραιότητας, για την εύρεση πληροφορίας που σχετίζεται με τα συγκεκριμένα ‘‘κλειδιά’’).

   Το χαρακτηριστικό αυτό, αποτελεί ένα βασικό πλεονέκτημα του WWW και ένα ισχυρό κίνητρο για τους χρήστες, αφού εύκολα και γρήγορα μπορούν, τις περισσότερες φορές, να βρουν την πληροφορία που ζητούν. Κάποιες από τις γνωστές μηχανές αναζήτησης του WWW είναι οι: Yahoo, Magellan, Lycos, Excite, WebCrawler, Infoseek, Alta Vista, WWW Worm κ.α.

5. Η χρήση Διαχειριστών-Αντιπροσώπων (Proxy-Servers).

   Η αρχιτεκτονική του WWW υποστηρίζει την έννοια της αντιπροσώπευσης (proxying). Σύμφωνα με αυτήν ο client αντί να συνδεθεί απευθείας με τον server που έχει τη ζητούμενη πληροφορία, στέλνει την αίτηση στον proxy server. Στη συνέχεια αυτός αναλαμβάνει να φέρει την πληροφορία (λειτουργώντας σαν client και ο ίδιος) και να την δώσει στη συνέχεια στον αρχικό client. Εκ πρώτης όψεως η διαδικασία αυτή, αν εξαιρέσουμε τις πιθανές μετατροπές από ένα πρωτόκολλο σε άλλο που μπορεί να κάνει ο proxy server, δεν φαίνεται να έχει άλλα πλεονεκτήματα. Στην πραγματικότητα όμως το μοντέλο αυτό δίνει μια λύση για την *ασφαλή παροχή δικτυακών υπηρεσιών* στους υπολογιστές του εσωτερικού δικτύου ενός οργανισμού, χωρίς αυτοί να έχουν άμεση επικοινωνία με τον ‘‘έξω κόσμο’’ (σχήμα 2.2(a)). Εκτός αυτού ο proxy server μπορεί να παράσχει υπηρεσίες αποθήκευσης πληροφορίας (caching facilities). Κρατάει δηλαδή τις πληροφορίες που έρχονται σαν απάντηση σε διάφορες αιτήσεις. Έτσι για κάθε νέα αίτηση που έρχεται, ελέγχει αν υπάρχει αποθηκευμένη η απάντησή της. Αν υπάρχει και είναι έγκυρη ακόμα τότε την δίνει αμέσως στον client, διαφορετικά συνδέεται στον κατάλληλο server, φέρνει την πληροφορία, την δίνει στον client και παράλληλα κρατάει ένα αντίγραφο στην δική του cache για πιθανές επόμενες αιτήσεις (σχήμα 2.2(b)).



(a) Proxy server σαν σύστημα ασφάλειας

(b) Proxy server με προσωρινή αποθήκευση

Σχήμα 2.2: Proxy servers

Με τον τρόπο αυτό βελτιώνεται η ποιότητα της υπηρεσίας που λαμβάνει ο χρήστης (η πληροφορία μεταφέρεται πιο γρήγορα) αλλά και εξοικονομείται εύρος ζώνης (bandwidth) του δικτύου αφού δεν χρειάζεται να μεταφέρουμε πολλές φορές τα ίδια δεδομένα.

6. Οι μηχανισμοί ασφάλειας που χρησιμοποιεί.

   Ο διαχειριστής της πληροφορίας μπορεί να περιορίσει ή να απαγορεύσει την πρόσβαση κόμβων του δικτύου ή χρηστών σε συγκεκριμένες περιοχές της πληροφορίας που διαθέτει ή ακόμα και σε μεμονωμένα αρχεία. Υπάρχει

καθορισμένη διαδικασία σύμφωνα με την οποία μπορεί να γίνει εξακρίβωση της ταυτότητας του αποστολέα της αίτησης (authentication), με τη χρήση γνωστών αλγορίθμων ασφάλειας (π.χ MD5 [Riv92]). Επίσης με τη χρήση μεθόδων κρυπτογραφίας (π.χ DES [Nat77]) υπάρχει δυνατότητα μεταφοράς κρίσιμης πληροφορίας, όπως οι αριθμοί των πιστωτικών καρτών, χωρίς τον κίνδυνο υποκλοπής ή παραποίησής τους από τρίτους. Τα χαρακτηριστικά αυτά επιτρέπουν την διεξαγωγή εμπορικών πράξεων πάνω από το δίκτυο όπως παραγγελίες προϊόντων ή κρατήσεις θέσεων.

Δημιουργείται έτσι ένας νέος χώρος με εκατομμύρια χρήστες και εν δυνάμει αγοραστές που αποτελεί πρόκληση για διαφήμιση, προβολή και εμπορικές συναλλαγές. Αυτό ωθεί πάρα πολλές επιχειρήσεις, οργανισμούς και μεμονωμένα άτομα στη χρήση του WWW. Μετατρέπεται έτσι το δίκτυο από ένας καθαρά ακαδημαϊκός-ερευνητικός χώρος που ήταν τα προηγούμενα χρόνια, σε χώρο όπου η εμπορική δραστηριότητα (με την ευρεία έννοια) λαμβάνει ολοένα και μεγαλύτερο μερίδιο.

2.2 Η Γλώσσα HTML

Η γλώσσα HTML (HyperText Markup Language) [BLC95] είναι μια απλή γλώσσα σύνταξης (markup language) που χρησιμοποιείται για την παραγωγή υπερκειμένων, συμβατών με οποιαδήποτε υπολογιστική πλατφόρμα. Δίνει την δυνατότητα οργάνωσης και δόμησης των κειμένων χρησιμοποιώντας ειδικές ετικέτες (tags). Βάσει εκείνων γίνεται η αναγνώριση, και ο προσδιορισμός των οντοτήτων που συναποτελούν το κείμενο και τελικά η παρουσίαση του στο χρήστη. Η HTML αποτελεί εφαρμογή του ISO προτύπου 8879:1986 (Standard Generalized Markup Language (SGML)) [Org86].

Είναι κατάλληλη για την αναπαράσταση πληροφορίας από ένα ευρύ πεδίο εφαρμογών. Μπορεί να αναπαραστήσει σε μορφή υπερκειμένου τεχνικές αναφορές, news, μηνύματα ηλεκτρονικού ταχυδρομείου, υπερμέσα (hypermedia), καταλόγους επιλογών, αποτελέσματα ερωτήσεων σε βάσεις δεδομένων, δομημένα κείμενα με ενσωματωμένες εικόνες, τμήματα του συστήματος αρχείων κτλ.

Η γλώσσα HTML χρησιμοποιείται από τον World Wide Web από το 1990. Ένα μικρό παράδειγμα μιας WWW σελίδας γραμμένης σε HTML δίνεται στη συνέχεια.

```
<HTML>
<HEAD>
<TITLE> Παράδειγμα WWW Σελίδας </TITLE>
</HEAD>
<BODY>
<CENTER>
<H1> Δυνατότητες Γλώσσας HTML </H1>
</CENTER>
Με τη σελίδα αυτή θέλουμε να δείξουμε την σύνταξη της γλώσσας
HTML για την υλοποίηση μερικών δυνατοτήτων της.
```

```
Οι λέξεις που είναι μεταξύ των συμβόλων ``<'' και ``>'' είναι
οι λεγόμενες ετικέτες της γλώσσας HTML, δεν παρουσιάζονται στον
χρήστη αλλά καθορίζουν τον δομή και τρόπο παρουσίασης της
υπόλοιπης πληροφορίας.
<BR>
Έχουμε λοιπόν:
<BR>
Ενσωμάτωση εικόνας:
<IMG SRC = "home. gif" ALIGN=middle> εικόνα σπιτιού
<BR>
Σύνδεσμος Υπερκειμένου:
<A HREF = "http: //www. csd. uch. gr/welcome. html">
Τμήμα Επιστήμης Υπολογιστών </A>
<BR>
Καθορισμός μεγέθους και τύπου γραμμάτων: <BR>
<STRONG> <FONT SIZE="+2">T</FONT>ο αρχικό "T" θα είναι πιο μεγάλο από
τα υπόλοιπα γράμματα</STRONG>
<BR>
Στο μικρό αυτό παράδειγμα δεν μπορούμε να συμπεριλάβουμε
τη σύνταξη και άλλων χαρακτηριστικών όπως πίνακες, φόρμες,
μαθηματικοί τύποι, λίστες κ. τ. λ που η HTML υποστηρίζει.
</BODY>
</HTML>
```

2.3 Ομοιόμορφα Αναγνωριστικά Πόρων

Τα Ομοιόμορφα Αναγνωριστικά Πόρων (Uniform Resource Locators ή URLs) [2] [MBM94] είναι τυποποιημένες ακολουθίες χαρακτήρων οι οποίες προσδιορίζουν μοναδικά ένα πόρο δικτύου. Κωδικοποιούν δηλαδή μέσα σε ένα μικρό αριθμό χαρακτήρων, όλη την απαιτούμενη πληροφορία για την εύρεση ενός ‘‘αντικειμένου’’ στο δίκτυο. Ουσιαστικά θα μπορούσαμε να πούμε ότι ορίζεται ένας νέος ‘‘χώρος ονοματολογίας και διευθυνσιοδότησης’’ όπου ανήκουν όλα τα αντικείμενα του δικτύου και το καθένα έχει την δική του *μοναδική* διεύθυνση.

Ένα URL χωρίζεται σε τρία τμήματα:

1. Το πρώτο τμήμα είναι το όνομα του σχήματος (πρωτόκολλο τις περισσότερες φορές) που θα χρησιμοποιηθεί για την ανταλλαγή της πληροφορίας (HTTP, FTP, Gopher, NNTP, Telnet, WAIS, Prospero, Mailto κ.α). Το χρησιμοποιούμενο σχήμα καθορίζει και την σύνταξη του υπόλοιπου URL. Για τα Internet πρωτόκολλα που κυρίως μας ενδιαφέρουν η σύνταξη των URLs φαίνεται στη συνέχεια:

   ```
   <scheme>: //<user>: <password>@<host>: <port>/<url-path>
   ```

[2] Στο υπόλοιπο της παρούσας εργασίας χρησιμοποιείται ο όρος URL

2. Το δεύτερο τμήμα του URL περικλείεται μεταξύ δύο διαδοχικών ‘‘/’’ και περιέχει τα ακόλουθα πεδία:
   • το όνομα και τον κωδικό πρόσβασης (password) του χρήστη (προαιρετικό πεδίο)
   • το DNS όνομα ή την IP διεύθυνση του κόμβου στον οποίο απευθυνόμαστε
   • το port μέσω του οποίου επικοινωνεί ο server (αν δεν υπάρχει χρησιμοποιείται το default)

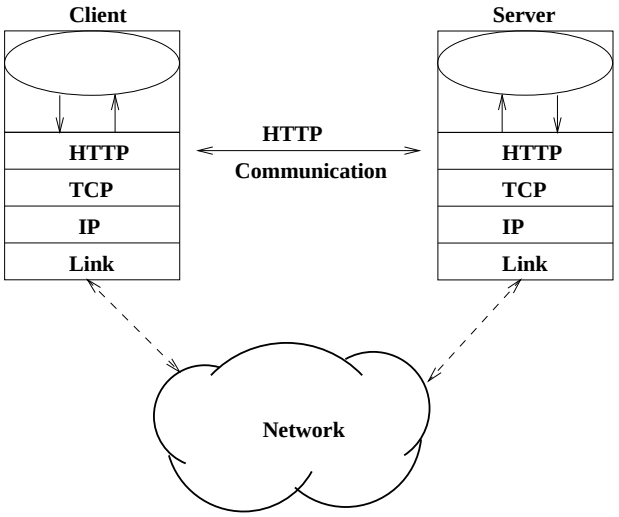
3. Το τελευταίο τμήμα του URL περιγράφει το αντικείμενο που ζητάμε από τον server (π.χ το απόλυτο όνομα ενός αρχείου στο τμήμα του συστήματος αρχείων που ο server επιτρέπει την πρόσβαση ή το όνομα ενός εκτελέσιμου προγράμματος με τις παραμέτρους εισόδου του).

Στη συνέχεια δίνουμε τρία χαρακτηριστικά παραδείγματα πραγματικών URLs:
• *http://www.csd.uch.gr/services.html*
(<scheme> = http || <host> = www.csd.uch.gr || <url-path> = services.html)
Αντιστοιχεί στο αρχείο *servises.html* του WWW server του Τμήματος Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης.
• *http://www.csd.uch.gr:80/cgi-bin/finger*
(<scheme> = http || <host> = www.csd.uch.gr, <port> = 80 ||
<url-path> = cgi-bin/finger)
Αντιστοιχεί στο εκτελέσιμο πρόγραμμα finger του ίδιου server.
• *ftp://ftp.csi.forth.gr/tech-reports/index.txt*
(<scheme> = ftp || <host> = www.csi.forth.gr || <url-path> = tech-reports/index.txt)
Αντιστοιχεί στο αρχείο /tech-report/index.txt του ftp server του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας.

Η γενικότητα της σύνταξης των URLs διευκολύνει την επεκτασιμότητά τους. Έτσι μπορούμε με ελάχιστες αλλαγές να εισάγουμε νέα πρωτόκολλα και να ελέγξουμε τις δυνατότητες τους, εκμεταλλευόμενοι τα πλεονεκτήματα που παρέχουν τα URLs.

2.4 Το Πρωτόκολλο HTTP

Το τρίτο δομικό κομμάτι του WWW θα το αναλύσουμε σε μεγαλύτερο βάθος αναφερόμενοι και σε θέματα απόδοσης και προτάσεις βελτίωσής του. Ο λόγος είναι ότι η παρούσα εργασία τροποποιεί και επεκτείνει αυτό το κομμάτι. Το πρωτόκολλο HTTP (HyperText Transfer Protocol) [FBF96] είναι ένα πρωτόκολλο του επιπέδου εφαρμογών και αποτελεί τη βάση του World Wide Web (WWW). Η πρώτη του έκδοση κυκλοφόρησε μόλις το 1990. Τα βασικά του πλεονεκτήματα είναι η απλότητα και η ευελιξία του. Η επικοινωνία βασίζεται στο μοντέλο πελάτη-εξυπηρετητή (client-server) και τα δεδομένα ανταλλάσσονται σύμφωνα με την τεχνική της αίτησης/απάντησης (request/response). Το HTTP απαιτεί αξιόπιστη μεταφορά δεδομένων γι’ αυτό και έχει ‘‘χτιστεί’’ πάνω από το πρωτόκολλο TCP [Pos81c] (default port 80) του επιπέδου μεταφοράς, το οποίο παρέχει αυτήν την υπηρεσία (σχήμα 2.3).



Σχήμα 2.3: Το HTTP στη στοίβα των πρωτοκόλλων

Η έναρξη κάθε δοσοληψίας ξεκινάει από τον client. Αφού εγκατασταθεί η σύνδεση με το server, ο client προετοιμάζει και στέλνει τη αίτηση στην οποία προσδιορίζει το ζητούμενο αντικείμενο [3], μαζί με ένα σύνολο από πληροφορίες που αφορούν το αντικείμενο, ή τον ίδιο. Με την λήψη της αίτησης, ο server την αναλύει, εκτελεί τις κατάλληλες ενέργειες και στέλνει την απάντηση. Μετά το πέρας της αποστολής η σύνδεση κλείνει και πάλι.

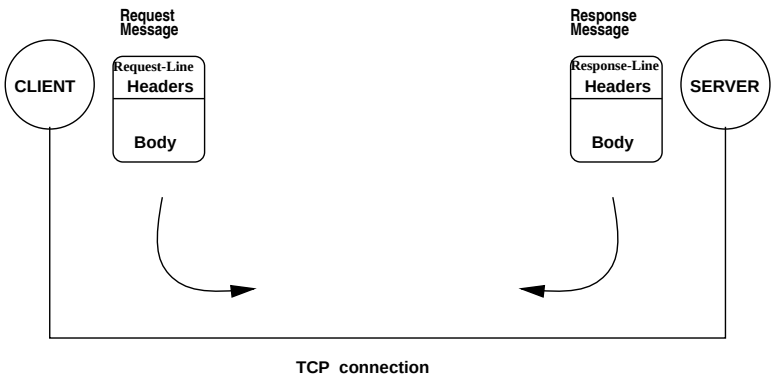
Η ευελιξία του πρωτοκόλλου έγκειται στο γεγονός ότι στην αίτηση του client ή στην απάντηση του server μπορεί να μεταφερθεί και ένα πλήθος από άλλες πληροφορίες οι οποίες και καθοδηγούν την συμπεριφορά των οντοτήτων (του client ή του server) που τις λαμβάνουν αντίστοιχα. Αυτό επιτυγχάνεται μέσω της δομής των HTTP μηνυμάτων. Τα μηνύματα αυτά αποτελούνται από δύο μέρη όπως φαίνεται στο σχήμα 2.4. Στο πρώτο μεταφέρεται η πληροφορία ελέγχου και στο άλλο η ζητούμενη πληροφορία.

Μια **HTTP αίτηση** αποτελείται από τα εξής στοιχεία:

- Την μέθοδο που πρέπει να χρησιμοποιήσει ο server για να επιστρέψει την απάντηση (π.χ GET, POST, HEAD, PUT, DELETE, TRACE κτλ.) καθώς και το URL που καθορίζει το ζητούμενο αντικείμενο. Όλα αυτά αποτελούν την *γραμμή αίτησης* (Request-Line)
- Μια σειρά από επικεφαλίδες (Headers)
- Ένα προαιρετικό τμήμα δεδομένων (Body), που χρησιμοποιείται μόνο από συγκεκριμένες μεθόδους (π.χ POST)

Μια **HTTP απάντηση** αποτελείται από τα εξής στοιχεία:

[3] Με τον όρο αντικείμενο στην παρούσα εργασία εννοούμε οποιαδήποτε μορφή αρχείου, από απλό text, μέχρι εικόνες, γραφικά ή αρχεία ήχου και video



Σχήμα 2.4: Τα μηνύματα του HTTP

- Ένα κωδικό κατάστασης (status code), για να δείξει την επιτυχία ή μη, της αίτησης και το λόγο της αποτυχίας στην δεύτερη περίπτωση. Όλα αυτά αποτελούν την *γραμμή απάντησης* (Response-Line).
- Ένα σύνολο επικεφαλίδων (Headers)
- Ένα τμήμα δεδομένων που περιέχει το ζητούμενο αντικείμενο (Body) σε περίπτωση επιτυχίας

Οι πιο συχνά χρησιμοποιούμενες μέθοδοι του πρωτοκόλλου είναι οι *GET, HEAD και POST*. Η μέθοδος GET ορίζει την μεταφορά ενός αντιγράφου του αντικειμένου που προσδιορίζεται από το URL ή την μεταφορά των αποτελεσμάτων της διεργασίας, σε περίπτωση που το URL αναφέρεται σε μια τέτοια. Η μέθοδος HEAD είναι παρόμοια με την GET μόνο που ο server δεν συμπεριλαμβάνει το κυρίως μέρος στην απάντηση του. Στέλνει όμως όλες τις βοηθητικές πληροφορίες που θα στέλνονταν σε μια αίτηση με μέθοδο GET. Η μέθοδος HEAD χρησιμοποιείται κυρίως για έλεγχο δυνατότητας πρόσβασης, εγκυρότητας και χρόνου τροποποίησης ενός συνδέσμου υπερκειμένου (hypertext link). Η μέθοδος POST χρησιμοποιείται για την μεταφορά δεδομένων από τον client στον server όπως μηνύματα για newsgroup και λίστες ηλεκτρονικού ταχυδρομείου, δεδομένα από HTML φόρμες, στοιχεία για επέκταση βάσεων δεδομένων κτλ.

Ο *κωδικός κατάστασης* είναι ένας τριψήφιος αριθμός που περιγράφει (κωδικοποιημένα) το αποτέλεσμα μιας αίτησης. Υπάρχουν πέντε κλάσεις τέτοιων κωδικών:

Κωδικός	**Κλάση**	**Αιτιολογία**
1xx	Πληροφοριακά	Η αίτηση ελήφθη, και η διαδικασία μπορεί να συνεχιστεί κανονικά.
2xx	Επιτυχία	Όλα τα βήματα έγιναν με επιτυχία.
3xx	Ανακατεύθυνση	Περαιτέρω ενέργειες απαιτούνται για την ολοκλήρωση της αίτησης.
4xx	Πρόβλημα στον client	Η αίτηση έχει λάθος σύνταξη ή δεν μπορεί να διεκπεραιωθεί.
5xx	Πρόβλημα στον server	Ο server απέτυχε να ικανοποιήσει μια έγκυρη αίτηση.

Οι επικεφαλίδες μεταφέρουν όλη την υπόλοιπη πληροφορία ελέγχου. Έχουν απλή δομή της μορφής:

```
HTTP-HEADER-SYNTAX = <Header-Name> ": " <Header-Value>
```

και χωρίζονται στις ακόλουθες κατηγορίες ανάλογα με το σκοπό που εξυπηρετούν:

- **Επικεφαλίδες Γενικού Περιεχομένου** (General Headers)

  Οι επικεφαλίδες αυτές μπορούν να χρησιμοποιηθούν τόσο στα μηνύματα αίτησης όσο και στα μηνύματα απάντησης και δεν έχουν καμία σχέση με το κυρίως μέρος του μηνύματος π.χ.
  Date :: Επικεφαλίδα που κρατάει το χρόνο αποστολής του μηνύματος
  Forwarded :: Επικεφαλίδα που συμπληρώνεται από τους proxy-servers και περιέχει τους ενδιάμεσους σταθμούς που διήλθε ένα HTTP μήνυμα.

- **Επικεφαλίδες Αίτησης** (Request Headers)

  Οι επικεφαλίδες αυτές επιτρέπουν στον client να παράσχει επιπρόσθετες πληροφορίες σχετικές με την αίτηση ή τον ίδιο π.χ.
  Accept :: Επικεφαλίδα που ενημερώνει το server για τα είδη των αντικειμένων που ο client μπορεί να χειριστεί.
  Authorization :: Επικεφαλίδα που μεταφέρει τα πιστοποιητικά εξακρίβωσης ταυτότητας του αποστολέα.

- **Επικεφαλίδες Απάντησης** (Response Headers)

  Οι επικεφαλίδες αυτές επιτρέπουν στον server να παράσχει επιπρόσθετες πληροφορίες σχετικές με την απάντηση. Οι πληροφορίες αυτές δεν αφορούν τόσο το προς μεταφορά αντικείμενο όσο τον τρόπο πρόσβασης σε αυτό ή στον ίδιο τον server π.χ.
  Server :: Επικεφαλίδα με στοιχεία για το λογισμικό του server που εξυπηρέτησε την αίτηση.
  WWW-authenticate :: Επικεφαλίδα που δεικνύει τον τρόπο εξακρίβωσης ταυτότητας που πρέπει να χρησιμοποιηθεί για την πρόσβαση στο ζητούμενο αντικείμενο.

- **Επικεφαλίδες Κύριου Μέρους** (Entity Headers)

  Οι επικεφαλίδες αυτές ορίζουν βοηθητικές πληροφορίες (metainformation) για για το κυρίως μέρος του μηνύματος π.χ.
  Content-Length :: Επικεφαλίδα με το μέγεθος, σε bytes, του κυρίου μέρους
  Content-Encoding :: Επικεφαλίδα που μεταφέρει το μηχανισμό σύμφωνα με τον οποίο έχει κωδικοποιηθεί το κυρίως μέρος του μηνύματος
  Expires :: Επικεφαλίδα που κρατάει το χρονικό διάστημα που θεωρείται έγκυρο το παρόν μήνυμα, σε περίπτωση caching.

Τέλος ένα άλλο πλεονέκτημα του HTTP είναι ότι θεωρεί όλες τις πληροφορίες σαν μια ακολουθία από bytes δηλαδή χαρακτήρες των οκτώ bits. Έτσι σε μια HTTP δοσοληψία μπορεί να μεταφερθεί οποιαδήποτε μορφή δεδομένων (characters, binary κ.τ.λ) χωρίς κανένα πρόβλημα.

2.5 Προβλήματα στην Απόδοση του HTTP

Η επιτυχία του WWW οφείλεται σε μεγάλο βαθμό στην απλότητα και επομένως στην ευκολία υλοποίησης του πρωτοκόλλου που στηρίζεται, δηλ. του HTTP. Το HTTP πλεονεκτεί συγκρινόμενο με το άλλο πρωτόκολλο μεταφοράς αρχείων, το FTP, όσον αφορά την απόδοση, ενώ παράλληλα δεν χρειάζεται κάποιο επιπλέον κανάλι ελέγχου στον client ή στο server. Παρόλα αυτά και το HTTP κάνει μη αποδοτική χρήση των δικτυακών και υπολογιστικών πόρων και προσθέτει καθυστέρηση με την δημιουργία μιας TCP σύνδεσης για κάθε αίτηση [PM94]. Στη συνέχεια παρουσιάζουμε και αναλύουμε τα προβλήματα που προκύπτουν εξαιτίας της τεχνικής αυτής.

2.5.1 Το Πρόβλημα Απόδοσης της Τεχνικής Σύνδεση ανά Αίτηση

Όπως προαναφέραμε τα βήματα για τη μεταφορά ενός αντικειμένου από το HTTP είναι τέσσερα: 1. εγκατάσταση σύνδεσης μεταξύ client-server, 2. αποστολή αίτησης από τον client, 3. επιστροφή της απάντησης από τον server, 4. κλείσιμο της σύνδεσης από τον server. Η δημιουργία όμως μιας TCP σύνδεσης για κάθε αίτηση έχει σαν αποτέλεσμα να απαιτείται χρόνος **τουλάχιστον δύο round-trips** για την επιστροφή της απάντησης στον client, και λέμε τουλάχιστον δύο, γιατί αντικείμενα μεγάλα, που δεν χωράνε σε ένα TCP segment, αντιμετωπίζουν επιπλέον καθυστέρηση.

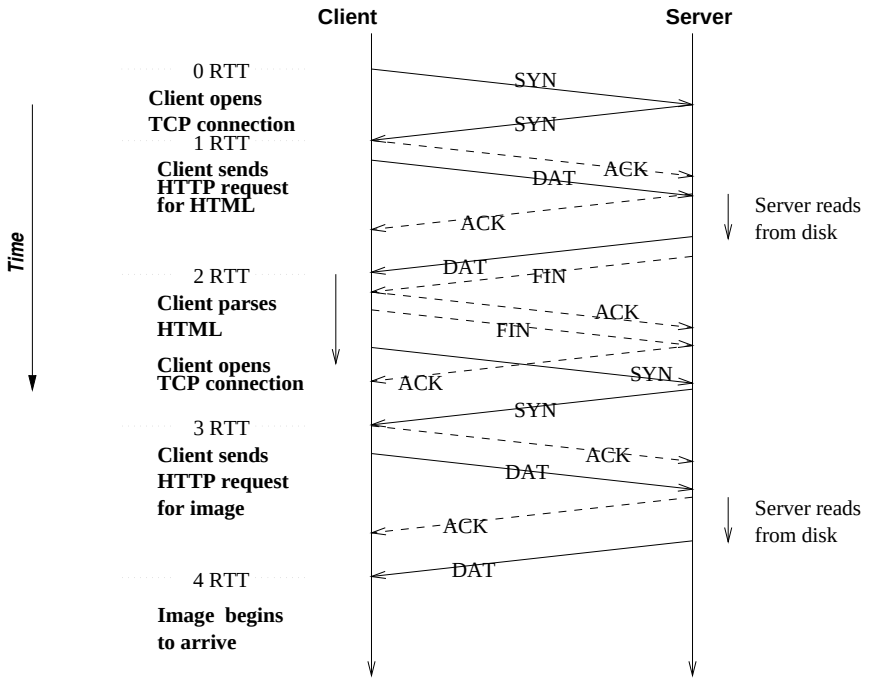
Στη συνέχεια δίνουμε ένα παράδειγμα και εξηγούμε αναλυτικά πόσος χρόνος χρειάζεται κάθε βήμα. Στο σχήμα 2.5 φαίνονται τα μηνύματα που ανταλλάσσονται από τη έναρξη μιας τυπικής δοσοληψίας, για την ανάκληση μιας HTML σελίδας με μία τουλάχιστον ενσωματωμένη εικόνα σε αυτήν.

Τα διαγώνια βέλη παριστάνουν τα TCP πακέτα ενώ τα μικρά κάθετα βέλη τοπικές καθυστερήσεις στον client ή τον server. Στο αριστερό μέρος του σχήματος φαίνονται οι *αναπόφευκτοι χρόνοι round trip* (RTTs) που επιβάλλονται από την συνεργασία των HTTP και TCP πρωτοκόλλων. Οι χρόνοι αυτοί είναι αποτέλεσμα των αλληλεξαρτήσεων μεταξύ των πακέτων που σημειώνουμε με τα βέλη με συνεχή γραμμή. Τα πακέτα που αντιστοιχούν στα βέλη με διακεκομμένη γραμμή, είναι αναγκαία μεν για το TCP αλλά δεν επηρεάζουν άμεσα την καθυστέρηση (latency) επειδή ο client δεν χρειάζεται να περιμένει γι' αυτά για να προχωρήσει στις επόμενες ενέργειές του. Τα αναπόφευκτα RTTs είναι:

RTT 1: Ο client ανοίγει μια TCP σύνδεση ενεργοποιώντας την διαδικασία διαπραγμάτευσης τριών φάσεων (three-way handshake) με αποτέλεσμα την ανταλλαγή πακέτων συγχρονισμού (SYN packets).

RTT 2: Ο client στέλνει την HTTP αίτηση στον server. Ο server εκτελεί τις κατάλληλες ενέργειες για την διεκπεραίωση της αίτησης στέλνει πίσω την απάντηση [4] και στη συνέχεια κλείνει την σύνδεση.

[4] Στο συγκεκριμένο παράδειγμα υποθέτουμε ότι η απάντηση είναι αρκετά μικρή ώστε να χωράει σε ένα πακέτο δεδομένων, αν και γενικότερα αυτό δεν ισχύει.



Σχήμα 2.5: Καθυστέρηση των HTTP μηνυμάτων λόγω της τεχνικής σύνδεση ανά αίτηση

RTT 3: Ο client ‘‘διαβάζει’’ (parses) την HTML σελίδα που έλαβε, εντοπίζει τα URLs των ενσωματωμένων εικόνων και ανοίγει μια νέα TCP σύνδεση, οπότε έχουμε νέα φάση συγχρονισμού.

RTT 4: Ο client στέλνει την HTTP αίτηση για την πρώτη εικόνα. Ο server την λαμβάνει, βρίσκει το ζητούμενο αντικείμενο και αρχίζει να το στέλνει.

Έτσι ο ελάχιστος χρόνος για την έναρξη παρουσίασης της πρώτης ενσωματωμένης εικόνας είναι τέσσερα RTTs μετά την αίτηση του χρήστη. Κάθε επιπλέον εικόνα χρειάζεται *τουλάχιστον* δύο ακόμα RTTs. Το πόσο μεγαλύτερος των δύο RTTs είναι ο απαιτούμενος χρόνος, εξαρτάται από το μέγεθος της κάθε εικόνας και την τιμή του παραθύρου της TCP σύνδεσης πάνω από την οποία μεταφέρονται τα δεδομένα.

2.5.2 Χρήση Υπολογιστικών Πόρων

Η δημιουργία μιας TCP σύνδεσης για κάθε αίτηση προκαλεί και άλλα επιπλέον κόστη που έχουν να κάνουν με τις απαιτήσεις σε υπολογιστική ισχύ και σε μνήμη του client και του server.

- Η εγκατάσταση της σύνδεσης απαιτεί κάποιο χρόνο επεξεργασίας και από τα δύο μέρη που συμμετέχουν, κυρίως για την εκχώρηση νέων ports, την δημιουργία των αναγκαίων δομών δεδομένων κτλ. Αντίστοιχη διαδικασία απαιτείται και κατά το κλείσιμο της σύνδεσης αλλά σε μικρότερο βαθμό.

- Οι TCP συνδέσεις μπορεί να είναι ενεργές για λίγα δευτερόλεπτα, αλλά όπως αναφέρεται στις προδιαγραφές (specifications) του TCP το μέρος που κλείνει την σύνδεση (στην συγκεκριμένη περίπτωση ο server) πρέπει να κρατάει κάποιες πληροφορίες για την σύνδεση για 4 min επιπλέον. Αυτό για έναν ‘‘δημοφιλή’’ server μπορεί να σημαίνει υπερβολικό κόστος για την διαχείριση των PCBs (Protocol Control Blocks) ή ακόμα και την πλήρωση όλων των θέσεων με συνδέσεις που έχουν κλείσει, μη αφήνοντας χώρο για νέες.

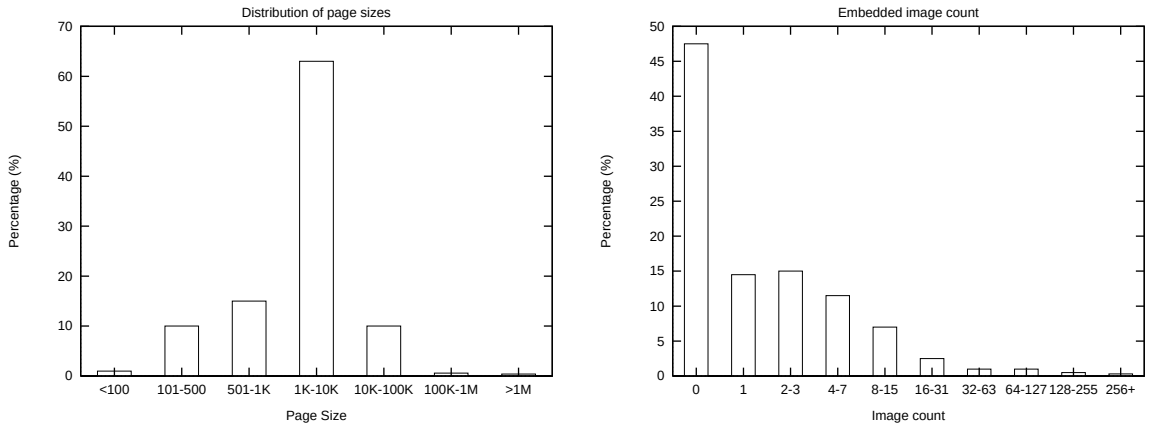
Ουσιαστικά δηλαδή δημιουργείται μια επιβάρυνση η οποία παίζει σημαντικό ρόλο, ιδίως όταν ο server δέχεται πολλές αιτήσεις ή λειτουργεί σε μηχάνημα μικρής υπολογιστικής ισχύος.

2.5.3 Συνέπειες του Αλγορίθμου Αργής Εκκίνησης του TCP

Από αναλύσεις που έχουν γίνει [Bra96], [Mar96], προκύπτει ότι το μεγαλύτερο ποσοστό (πάνω από το 80%) των WWW σελίδων που μεταφέρονται με το HTTP δεν ξεπερνά σε μέγεθος τα 10Kbytes (σχήμα 2.6(a)). Πρόκειται δηλαδή για σχετικά μικρά αρχεία τα οποία δεν συμβάλλουν στην καλή απόδοση του HTTP. Αυτό οφείλεται στο γεγονός ότι το TCP δεν εκμεταλλεύεται όλο το διαθέσιμο εύρος ζώνης (bandwidth) στο αρχικό στάδιο μιας σύνδεσης, λόγω του **αλγορίθμου αργής εκκίνησης** (slow-start algorithm) [Jac88] ο οποίος στοχεύει στην αποφυγή της συμφόρησης στο δίκτυο. Σύμφωνα με αυτήν την τεχνική ο αποστολέας διπλασιάζει τον αριθμό των πακέτων που στέλνει στο δίκτυο, κάθε φορά που λαμβάνει ένα μήνυμα επιβεβαίωσης (aknowledgement) για κάποιο προηγούμενο πακέτο δηλαδή αν δεν υπάρχουν απώλειες μετά από χρόνο round-trip.

Ο αλγόριθμος αργής εκκίνησης λοιπόν περιορίζει το διέλευση (throughput) που λαμβάνει μια TCP σύνδεση, μειώνοντας την πιθανότητα συμφόρησης στο δίκτυο αυξάνοντας όμως ταυτόχρονα την καθυστέρηση σε συνδέσεις μικρής διάρκειας, όπως αυτές του HTTP. Έτσι συνδέσεις με μεγάλα round-trip πρέπει να μεταφέρουν μεγάλη ποσότητα δεδομένων πριν λάβουν το μέγιστο διαθέσιμο εύρος ζώνης. Αν δεν έχουν πολλή πληροφορία προφανώς τελειώνουν χωρίς να πιάσουν το μέγιστο όριο.

Επιπρόσθετα η εκτεταμένη χρήση ενσωματωμένων εικόνων και άλλων σχετικών αντικειμένων μέσα στα html αρχεία (σχήμα 2.6(b)) επιτείνει το πρόβλημα. Ο client αναγκάζεται να κάνει πολλές συνδέσεις/αιτήσεις στον ίδιο server σε σύντομο χρονικό διάστημα. Κάθε μια από τις συνδέσεις αυτές υφίσταται εξ αρχής τον αλγόριθμο αργής εκκίνησης, μη μπορώντας να εκμεταλλευτεί τη γνώση που έχει αποκτηθεί από τις προηγούμενες, ως προς το μέγεθος του ωφέλιμου παραθύρου του TCP, δηλαδή τον μέγιστο αριθμό πακέτων που μπορεί να στείλει χωρίς να προκαλέσει συμφόρηση.



(a) Μεγέθη WWW σελίδων (b) Αριθμός εικόνων ανά WWW σελίδα

Σχήμα 2.6: Στοιχεία σελίδων του WWW

2.6 Προσεγγίσεις στο Πρόβλημα Απόδοσης του HTTP

Για τη λύση στο πρόβλημα καθυστέρησης (latency) του HTTP έχουν γίνει διάφορες προτάσεις. Μια προσέγγιση η οποία δεν απαιτεί τροποποίηση του πρωτοκόλλου είναι εκείνη που συχνά ξεκινάει *πολλαπλές παράλληλες* συνδέσεις [5]. Για παράδειγμα αν μια WWW σελίδα εμπεριέχει δέκα εικόνες, τότε η παραπάνω τεχνική ανοίγει μια TCP σύνδεση για την μεταφορά του HTML αρχείου και στη συνέχεια μπορεί να ανοίξει ακόμα και δέκα επιπλέον συνδέσεις, παράλληλα, για να φέρει τις εικόνες. Η παραλληλοποίηση των TCP συνδέσεων εξαλείφει μεγάλο μέρος της περιττής καθυστέρησης του HTTP.

Τα μειονεκτήματα της προσέγγισης αυτής είναι: *α)* Αυξάνει τις πιθανότητες για συμφόρηση στο δίκτυο, αφού πολλές παράλληλες TCP συνδέσεις είναι περισσότερο πιθανό να επηρεάσουν η μια την άλλη και να οδηγήσουν σε συμφόρηση απ' ότι μία μόνη της, *β)* δεν επιτρέπει στα δύο μέρη μιας σύνδεσης να μάθουν για την κατάσταση του δικτύου και έτσι δεν αντιμετωπίζει το κόστος που προκαλεί ο αλγόριθμος αργής εκκίνησης, με συνέπεια να υφίσταται το κόστος αυτό για κάθε αρχείο που μεταφέρει.

Μια άλλη πρόταση είναι η χρήση από το HTTP του Transaction TCP (T/TCP) [Bra94] αντί του TCP. Το T/TCP δεν χρησιμοποιεί την διαδικασία διαπραγμάτευσης τριών φάσεων για συγχρονισμό στην έναρξη και το πέρας μιας σύνδεσης. Επομένως είναι απαλλαγμένο από την καθυστέρηση που προκαλείται από αυτήν. Λύνονται έτσι κάποια από τα προβλήματα που αναφέραμε. Το βασικό μειονέκτημα στην περίπτωση αυτή είναι ότι το T/TCP είναι ακόμα ένα ''πειραματικό'' πρωτόκολλο. Έτσι είναι ελάχιστα διαδεδομένο σε αντίθεση με την τεράστια βάση του TCP στο δίκτυο σήμερα.

Η προσέγγιση της παρούσας εργασίας για την αντιμετώπιση του προβλήματος αυτού γίνεται με την δημιουργία συνδέσεων διαρκείας (persistent connections),

[5] Την προσέγγιση αυτή χρησιμοποιεί ο browser της εταιρίας NetScape.

συνδέσεων δηλαδή οι οποίες με τη σύμφωνη γνώμη των δύο μερών που συμμετέχουν, παραμένουν ανοιχτές και εξυπηρετούν περισσότερες από μία HTTP αιτήσεις. Η ανάλυση και ο τρόπος υλοποίησής τους περιγράφεται στην ενότητα 4.5.2.

Κεφάλαιο 3

Εκτίμηση Κατάστασης Δικτύου

Ένα δίκτυο όπως όλα τα σύνθετα συστήματα για να λειτουργεί καλά, πρέπει να λειτουργούν και να συνεργάζονται σωστά όλα τα τμήματα που το αποτελούν. Φυσικά αυτό δεν συμβαίνει πάντα, είτε γιατί κάποιο από τα μηχανήματα έχει πρόβλημα, είτε γιατί ένας σύνδεσμος επικοινωνίας είναι εκτός λειτουργίας, είτε γιατί κάποιο σημείο έχει μεγάλη συμφόρηση και προκαλεί απώλειες δεδομένων κτλ. Θα ήταν λοιπόν πολύ χρήσιμο για τους διαχειριστές των δικτύων, να γνωρίζουν τα σημεία στα οποία εμφανίζεται το πρόβλημα και τις αιτίες που το προκαλούν. Στην ενότητα 3.1 αναφέρουμε ενδεικτικά μερικά από τα θέματα που δυσχεραίνουν την παρακολούθηση και τον έλεγχο του δικτύου, τα οποία και μας απασχόλησαν στην προσπάθειά μας για την εκτίμηση της κατάστασης του δικτύου. Στην ενότητα 3.2 περιγράφουμε συνοπτικά το πρωτόκολλο ICMP, το οποίο αποτελεί ένα ισχυρό εργαλείο για τον εντοπισμό και απομόνωση των σημείων που προκαλούν προβλήματα. Μηνύματα του ICMP έχουμε χρησιμοποιήσει στην παρούσα εργασία για τον υπολογισμό των παραμέτρων του δικτύου που μας ενδιαφέρουν.

3.1 Δυσκολίες στον Υπολογισμό Παραμέτρων του Δικτύου

Ο υπολογισμός των παραμέτρων ενός δικτύου είναι μια αρκετά δύσκολη διαδικασία. Οταν πρόκειται για ένα τοπικό (LAN) ή έστω ένα μείζονος περιοχής (WAN) δίκτυο (που η σύνθεσή του δεν ξεπερνά τις μερικές δεκάδες συνδέσμων επικοινωνίας) μπορούμε με σχετικά αποτελεσματικό και γρήγορο τρόπο να το παρακολουθήσουμε, να εντοπίσουμε τυχόν προβλήματα και να μετρήσουμε κάποιες παραμέτρους. Αυτό μπορεί να επιτευχθεί είτε με τη βοήθεια του πρωτοκόλλου SNMP (Simple Network Management Protocol) [JCD90], είτε με κάποιες άλλες μεθόδους παρακολούθησης [Str94]. Άλλωστε σε τέτοιες περιπτώσεις είναι γνωστά και η συνολική τοπολογία του δικτύου και οι χωρητικότητες των συνδέσμων επικοινωνίας και η πολιτική δρομολόγησης που χρησιμοποιείται.

Αντίθετα σε ένα ‘‘ανοικτό’’ δίκτυο και ειδικά όταν αναφερόμαστε στο Internet

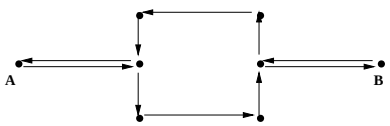
τίποτα από τα παραπάνω δεν ισχύει. Γι' αυτό το λόγο λίγες από τις παραμέτρους του μπορούν να υπολογιστούν και αυτές μόνο κατά προσέγγιση. Στη συνέχεια της ενότητας αυτής παραθέτουμε μερικά από τα στοιχεία που δεικνύουν το μέγεθος του προβλήματος.

- Το Internet συντίθεται από την διασύνδεση ετερογενών δικτύων με συνδέσμους μείζονος περιοχής. Αποτέλεσμα αυτού είναι η δημιουργία ενός δικτύου τυχαίας τοπολογίας ενώ ταυτόχρονα κατά μήκος ενός μονοπατιού μπορούμε να συναντήσουμε συνδέσμους επικοινωνίας με διαφορά στις ταχύτητες τους μία τάξη μεγέθους (από μερικά Kbps μέχρι δεκάδες Mbps). Η **ανομοιογένεια** αυτή στην ταχύτητα μεταφοράς δεδομένων, προκαλεί συμφόρηση στους πιο αργούς συνδέσμους επηρεάζοντας την απόδοση όλων των συνδέσεων που διέρχονται από αυτούς. Το ποσό του εύρους ζώνης λοιπόν που μπορεί να πάρει μια σύνδεση που διέρχεται από ένα μονοπάτι, εξαρτάται από το εύρος ζώνης που θα της δώσει ο πιο κορεσμένος σύνδεσμος κατά μήκος αυτού.

- Οι μέχρι τώρα υλοποιήσεις των πρωτοκόλλων της TCP/IP στοίβας **δεν μπορούν να δώσουν και να εγγυηθούν συγκεκριμένη και σταθερή ποσότητα εύρους ζώνης σε κάποια εφαρμογή**. Τα πρωτόκολλα αυτά δέχονται όλες τις νέες αιτήσεις για σύνδεση, ανεξάρτητα με το αν υπάρχει αρκετό ή όχι εύρος ζώνης για να ικανοποιήσουν τις απαιτήσεις τους που δεν δηλώνονται στο δίκτυο (best effort protocols). Όλες οι συνδέσεις ανταγωνίζονται μεταξύ για το ανεπαρκές συνήθως εύρος ζώνης. Αποτέλεσμα αυτού είναι να επηρεάζονται και οι ήδη υπάρχουσες συνδέσεις, να μειώνεται το λαμβανόμενο throughput και πολλές φορές να οδηγούμαστε σε καταστάσεις συμφόρησης.

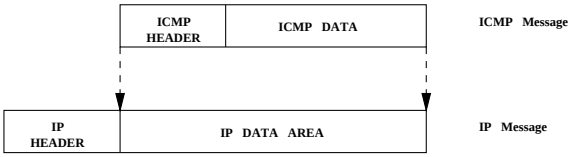
- Στη διαδρομή μεταξύ δύο σημείων είναι δυνατόν η **μία κατεύθυνση να έχει περισσότερη κυκλοφορία** και συνεπώς μεγαλύτερη συμφόρηση από την άλλη. Σαν χαρακτηριστικό παράδειγμα μπορούμε να αναφέρουμε την γνωστή εφαρμογή μεταφοράς αρχείων FTP, όπου η φορά από τον ftp server προς τον ftp client προφανώς έχει περισσότερη κυκλοφορία από την αντίστροφη.

- **Η δρομολόγηση των πακέτων μερικές φορές δεν είναι συμμετρική**. Δηλαδή η διαδρομή που τα πακέτα ακολουθούν από το σημείο Α στο Β μπορεί να είναι διαφορετική από εκείνη που ακολουθούν από το Β στο Α (σχήμα 3.1).



Σχήμα 3.1: Μη συμμετρική δρομολόγηση

Κάτι τέτοιο μπορεί να είναι απαραίτητο για την καλύτερη κατανομή και εξισορρόπηση της κυκλοφορίας (traffic balancing) στους συνδέσμους επικοινωνίας ή απλά να επιβάλλεται από την πολιτική δρομολόγησης που ακολουθούν οι οργανισμοί παροχής δικτυακών υπηρεσιών. Τέτοιες καταστάσεις ωστόσο δημιουργούν μεγάλα προβλήματα σε μεθόδους στις οποίες οι υπολογισμοί στηρίζονται στην μέτρηση του χρόνου round-trip, αφού δεν μπορούμε να



Σχήμα 3.2: Ενσωμάτωση των ICMP μηνυμάτων στα IP πακέτα

ξέρουμε πιο πραγματικά ήταν το μονοπάτι που εισήγαγε την καθυστέρηση [1].

3.2 Το Πρωτόκολλο ICMP

Στην προσπάθεια για την παρατήρηση της συμπεριφοράς των δικτύων και ανίχνευσης των ανωμαλιών που παρουσιάζονται, αναπτύχθηκε ένα νέο πρωτόκολλο, το ICMP (Internet Control Message Protocol) [Pos81a] το οποίο παρέχει ένα ενιαίο και απλό μηχανισμό με μηνύματα ελέγχου και πληροφορίες για μη αναμενόμενες καταστάσεις. Το ICMP πρωτόκολλο θεωρείται μέρος του IP (Internet Protocol) [Pos81b] πρωτοκόλλου.

Όπως όλα τα δεδομένα έτσι και τα ICMP μηνύματα μεταφέρονται πάνω από το δίκτυο μέσα στο πεδίο δεδομένων του IP μηνύματος (σχήμα 3.2). Τα IP πακέτα που περιέχουν ICMP μηνύματα, δρομολογούνται ακριβώς όπως και τα υπόλοιπα, χωρίς κανένα επιπλέον βαθμό αξιοπιστίας ή προτεραιότητας. Επομένως και αυτά μπορεί να χαθούν ή να απορριφθούν. Ο τελικός προορισμός ενός ICMP μηνύματος δεν είναι μια εφαρμογή ή ένας χρήστης αλλά το IP λογισμικό στο συγκεκριμένο μηχάνημα.

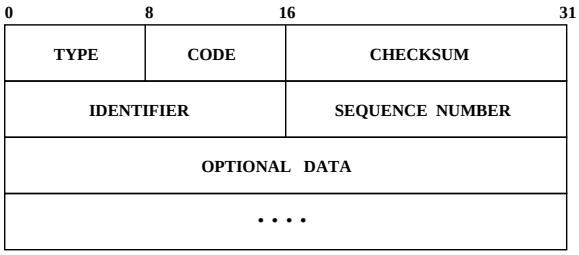
Οι κυριότερες κατηγορίες ICMP μηνυμάτων είναι:

Echo request/reply	Αίτηση/Απάντηση
Destination unreachable	Πρόβλημα στην παράδοση του μηνύματος
Source Quench	Επείγον μήνυμα ειδοποίησης
Redirect	Αλλαγή δρομολόγησης
Parameter problem	Πρόβλημα σε κάποιο πεδίο του IP πακέτου

Τα ICMP μηνύματα με τη συχνότερη χρήση είναι τα *echo request* και *echo reply*. Ένας κόμβος δικτύου που λαμβάνει ένα echo request μήνυμα, φτιάχνει ένα echo reply μήνυμα και το επιστρέφει στον αρχικό αποστολέα. Η δομή των μηνυμάτων αυτών φαίνεται στο σχήμα 3.3. Το πεδίο *TYPE* καθορίζει το είδος του μηνύματος (echo request ή echo reply). Το πεδίο *CHECKSUM* χρησιμοποιείται για εντοπισμό λαθών κατά τη μεταφορά. Τα πεδία *IDENTIFIER* και *SEQUENCE NUMBER* βοηθούν στην αντιστοίχιση των request/reply μηνυμάτων. Το πεδίο *OPTIONAL DATA* είναι προαιρετικό και μεταβλητού μεγέθους. Οτιδήποτε δεδομένα

[1] Μεταβολή στο μονοπάτι που ακολουθούν τα πακέτα δημιουργεί και η δυσλειτουργία (failure) ενός συνδέσμου επικοινωνίας, οπότε αυτόματα αλλάζουν οι πίνακες δρομολόγησης. Ωστόσο αυτό δεν είναι τόσο σημαντικό στην περίπτωσή μας, αν βέβαια οι αλλαγές δεν συμβαίνουν με συχνότητα της τάξης των δευτερολέπτων.

τοποθετήσει ο αποστολέας, τα ίδια ακριβώς περιέχονται και στο μήνυμα που του επιστρέφεται. Ανταλλάσσοντας τα παραπάνω μηνύματα μπορούμε να δούμε κατά πόσο είναι προσβάσιμος (reachable) ο προς έλεγχο προορισμός. Σωστή λήψη των απαντήσεων, επιβεβαιώνει ότι όλα τα τμήματα, του συστήματος μεταφοράς, που μεσολαβούν (gateways, σύνδεσμοι επικοινωνίας, πίνακες δρομολόγησης, τελικός προορισμός) λειτουργούν σωστά.

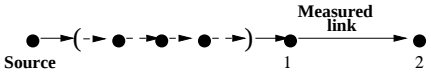


Σχήμα 3.3: ICMP Echo Request/Reply μήνυμα

3.3 Σχετικές Εργασίες

Όσον αφορά τον υπολογισμό παραμέτρων του δικτύου υπάρχουν δύο εργασίες οι οποίες χρησιμοποιούν μηνύματα του ICMP πρωτοκόλλου και παρουσιάζουν ενδιαφέρον για την περίπτωσή μας γιατί προσπαθούν να υπολογίσουν στοιχεία παρόμοια με αυτά της παρούσας εργασίας.

Η προσπάθεια της πρώτης [Bey95] εστιάζεται στην εκτίμηση του διαθέσιμου bandwidth πάνω σε *ένα* σύνδεσμο επικοινωνίας του δικτύου. Για να το επιτύχει αυτό στέλνει δύο πακέτα διαφορετικού μεγέθους (ένα μικρό και ένα μεγάλο) σε κάθε ένα από τα άκρα του προς έλεγχο συνδέσμου και καταγράφει τους χρόνους round-trip που απαιτούνται για το καθένα.



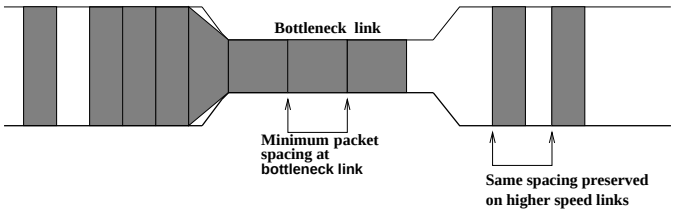
Σχήμα 3.4: Υπολογισμός διαθέσιμου bandwidth συνδέσμου

Αν $T_1^{(s)}, T_1^{(b)}, T_2^{(s)}, T_2^{(b)}$ είναι οι χρόνοι round-trip του μικρού και του μεγάλου πακέτου, μέχρι την πρώτη και την δεύτερη άκρη του συνδέσμου αντίστοιχα τότε για το round-trip πάνω στο συγκεκριμένο σύνδεσμο μόνο, ισχύουν οι σχέσεις:
$T_{1,2}^{(s)} = T_2^{(s)} - T_1^{(s)} = 2 * (sizesmall/R_{1,2} + WQ_{1,2} + PropagationTime)$ και
$T_{1,2}^{(b)} = T_2^{(b)} - T_1^{(b)} = 2 * (sizebig/R_{1,2} + WQ_{1,2} + PropagationTime)$ όπου $R_{1,2}$, $WQ_{1,2}$ και $PropagationTime$ ο ρυθμός αποστολής , ο χρόνος αναμονής μεταξύ των 1 και 2 και ο χρόνος μετάδοσης αντίστοιχα.

Έτσι με βάση την υπόθεση ότι *όλα τα πακέτα αντιμετωπίζουν τον ίδιο χρόνο καθυστέρησης στην ουρά του πρώτου άκρου*, το διαθέσιμο bandwidth δίνεται από τη σχέση $R_{1,2} = 2 * (sizebig - sizesmall)/(T_{1,2}^{(b)} - T_{1,2}^{(s)})$

Η μέθοδος αυτή δίνει ικανοποιητικά αποτελέσματα όταν μετράται το διαθέσιμο bandwidth μεταξύ σημείων που συνδέονται με ένα σύνδεσμο επικοινωνίας. Όταν όμως μεταξύ των σημείων αυτών παρεμβάλλονται πολλοί κόμβοι, τότε η υπόθεση ‘‘ίδια καθυστέρηση στις ουρές’’ *ισχύει με αρκετά μικρότερη ακρίβεια* και επομένως δύσκολα μπορεί να εφαρμοστεί. Ειδικά μάλιστα στην περίπτωση που ο πρώτος κόμβος είναι ο ίδιος ο οποίος στέλνει τα ICMP πακέτα, οπότε οι χρόνοι που παίρνουμε από τα πακέτα που κατευθύνονται σε αυτόν είναι αμελητέοι.

Στην [CC96] παρουσιάζεται μια μέθοδος για την εκτίμηση της ταχύτητας του περιοριστικού συνδέσμου σε ένα μονοπάτι του δικτύου. Η εκτίμηση αυτή γίνεται με την αποστολή μιας σειράς ICMP πακέτων και τον υπολογισμό του χρόνου μεταξύ των αφίξεων διαδοχικών πακέτων (inter-arrival time). Η ιδέα στην οποία στηρίζεται η μέθοδος αυτή είναι ότι ‘‘*αν δύο πακέτα που ταξιδεύουν μαζί, εισαχθούν διαδοχικά στην ουρά του περιοριστικού συνδέσμου, χωρίς να παρεμβάλλεται κανένα άλλο πακέτο μεταξύ τους, τότε η χρονική απόσταση (spacing) που θα δημιουργηθεί μεταξύ τους είναι ανάλογο του χρόνου που απαιτεί ο περιοριστικός δρομολογητής να επεξεργαστεί το δεύτερο πακέτο*’’ (σχήμα 3.5).



Σχήμα 3.5: Ροή πακέτων μέσα από περιοριστικό σύνδεσμο

Έτσι η ταχύτητα του περιοριστικού συνδέσμου λαμβάνεται από τη σχέση $BottleneckSpeed = PacketSize/InterArrivalGap$. Το πρόβλημα στην συγκεκριμένη εργασία είναι ότι για να δώσει ένα αξιόπιστο αποτέλεσμα πρέπει να εισάγει μεγάλη ποσότητα κυκλοφορίας στο δίκτυο και έτσι καθυστερεί αρκετά. Το γεγονός αυτό δεν επιτρέπει την χρησιμοποίηση της σε μια on-line διαδικασία. Εξάλλου εκείνο που ενδιαφέρει στην παρούσα εργασία δεν είναι τόσο το συνολικό bandwidth του περιοριστικού συνδέσμου, αλλά το bandwidth που μπορεί να διατεθεί σε μια σύνδεση την συγκεκριμένη χρονική στιγμή.

Η μέθοδός μας για τον υπολογισμό του προαναφερθέντος στοιχείου και η συμβολή της στο προτεινόμενο σύστημα παρουσιάζεται στο επόμενο κεφάλαιο, ενώ στο παράρτημα Β δίνουμε τις μετρήσεις που συλλέξαμε με αυτήν.

Κεφάλαιο 4

Το Προτεινόμενο Σύστημα

Στο κεφάλαιο 2 περιγράψαμε τα χαρακτηριστικά και τις δυνατότητες του WWW. Παρά το μεγάλο εύρος των δυνατοτήτων του μια βασική έλλειψη είναι η αδυναμία αυτόματης επιλογής μεταξύ διαφορετικών τύπων της ίδιας πληροφορίας. Τέτοιες περιπτώσεις πληροφορίας υπάρχουν αρκετές και αυξάνονται συνεχώς με την εξέλιξη της τεχνολογίας των δικτύων και των πολυμέσων. Ως παραδείγματα θα μπορούσαμε να αναφέρουμε τις τεχνικές αναφορές (technical reports) οι οποίες υπάρχουν σε μορφή html, postscript και doc ή την παρουσίαση ειδήσεων σε μορφή video ή ήχου. Η αυτόματη επιλογή φυσικά πρέπει να είναι συνάρτηση των προτιμήσεων του χρήστη και των εναλλακτικών τύπων πληροφορίας που υπάρχουν. Η έννοια των προτιμήσεων θα πρέπει να εμπεριέχει και την παράμετρο της χρονικής διάρκειας αναμονής για την λήψη της πληροφορίας.

Οι λόγοι για τους οποίους ένας χρήστης επιλέγει κάποιους μόνο από τους τύπους πληροφορίας ή τους δίνει διαφορετικές προτεραιότητες είναι:

- **μικρότερη χρέωση**: Σε ορισμένες περιπτώσεις ο χρήστης πληρώνει για την πρόσβασή του στο Internet ανάλογα με την ποσότητα των δεδομένων που μεταφέρει ή το με συνολικό χρόνο σύνδεσης. Τότε τον ενδιαφέρει ίσως περισσότερο εκείνος ο τύπος πληροφορίας που θα επιφέρει το χαμηλότερο κόστος, ακόμα και αν η ‘‘ποιότητά’’ του δεν είναι η καλύτερη δυνατή.

- **έλλειψη των κατάλληλων εργαλείων**: Ο χρήστης δεν διαθέτει τα απαραίτητα προγράμματα (π.χ video tools, image viewers) για να δει ή να επεξεργαστεί κάποιους τύπους πληροφορίας και επομένως δεν υπάρχει λόγος να φέρει πληροφορία που τους αντιστοιχεί.

- **ταχύτερη λήψη της πληροφορίας**: Ο χρήστης θέλει να πάρει τις πληροφορίες στο συντομότερο δυνατό χρονικό διάστημα. Έτσι δίνοντας προτεραιότητα στον τύπο πληροφορίας που έχει το μικρότερο μέγεθος επιτυγχάνει το ζητούμενο.

- **περιορισμένος χρόνος**: Ο χρήστης έχει στη διάθεσή του περιορισμένο χρόνο και επομένως θα ήθελε να ξέρει αν η πληροφορία που ζητάει μπορεί να μεταφερθεί στο διάστημα αυτό, διαφορετικά να μην περιμένει καθόλου.

Τα παραπάνω αποτέλεσαν τις κατευθυντήριες γραμμές για την υλοποίηση του συστήματος της παρούσας εργασίας. Το σύστημα αυτό, η αρχιτεκτονική του οποίου περιγράφεται αναλυτικά στην ενότητα 4.1, ακολουθεί και πάλι το μοντέλο client-server. Ο server τώρα χρειάζεται να γνωρίζει επιπλέον τις προτιμήσεις κάθε χρήστη που του ζητάει πληροφορία. Οι προτιμήσεις αυτές αποτελούν αυτό που ονομάζουμε profile του χρήστη.

Για λόγους που εξηγούμε σε παρακάτω ενότητες, επιλέξαμε οι προτιμήσεις αυτές να μεταφέρονται μέσα στις HTTP αιτήσεις και μόνο όταν υπάρχει δυνατότητα για επιλογή μεταξύ διαφορετικών τύπων απάντησης. Η επέκταση αυτή του πρωτοκόλλου HTTP και η δομή που ορίσαμε για τα profiles περιγράφονται στην ενότητα 4.3.1.

Η ύπαρξη χρονικών περιορισμών σε ένα profile και η λήψη μιας απόφασης βάσει αυτών, συνεπάγεται τον υπολογισμό της ικανότητας μεταφοράς δεδομένων μέσω του δικτύου και της αντίστοιχης καθυστέρησης τη συγκεκριμένη χρονική στιγμή. Απαιτείται λοιπόν μια μέθοδος η οποία θα εκτιμά σχετικά καλά την παράμετρο αυτή, αλλά σε πολύ μικρό χρόνο, της τάξης των δευτερολέπτων, και χωρίς να επιβαρύνει υπερβολικά την κυκλοφορία στο δίκτυο. Στην ενότητα 4.4 περιγράφουμε την μέθοδο που αναπτύξαμε για το σκοπό αυτό.

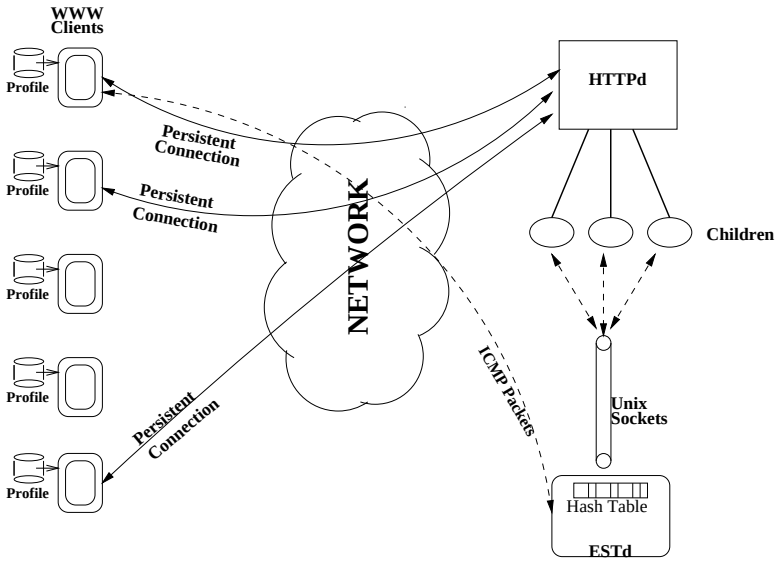
Παράλληλα με τα παραπάνω και ανεξάρτητα από την ύπαρξη πολλαπλών επιλογών για κάποιες από τις αιτήσεις, προσπαθήσαμε να μειώσουμε το χρόνο απόκρισης του συστήματος σε όλες τις HTTP αιτήσεις. Για να το πετύχουμε, έπρεπε να αποφύγουμε τα προβλήματα απόδοσης του πρωτοκόλλου HTTP. Η λύση που υλοποιήσαμε για το σκοπό αυτό είναι η δημιουργία συνδέσεων διαρκείας, συνδέσεων δηλαδή που δεν κλείνουν μετά την επιστροφή της απάντησης από τον server.

4.1 Η Αρχιτεκτονική του Συστήματος

Οι βασικές οντότητες οι οποίες συνεργάζονται κατά την λειτουργία του προτεινόμενου συστήματος είναι ο αντιπρόσωπος χρήστη (WWW client), ο διαχειριστής της πληροφορίας (HTTPd) και η διεργασία εκτίμησης των παραμέτρων του δικτύου (ESTd). Στη συνέχεια περιγράφουμε τις λειτουργίες καθεμιάς εστιάζοντας κυρίως στις νέες δυνατότητες που έχουμε προσθέσει.

Ο client προετοιμάζει και στέλνει τις HTTP αιτήσεις σύμφωνα με τις επιλογές του χρήστη. Αν πρόκειται να στείλει αίτηση που αντιστοιχεί σε πολλαπλά είδη πληροφορίας, τότε στο πρώτο μέρος του μηνύματος συμπεριλαμβάνει και μια νέα επικεφαλίδα με την τιμή του profile που έχει ορίσει ο χρήστης. Με τη λήψη του πρώτου μέρους της HTTP απάντησης, δηλαδή του τμήματος με τις επικεφαλίδες, εντοπίζει το είδος της πληροφορίας που επιστρέφεται από τον server και ενημερώνει σχετικά τον χρήστη, δεδομένου ότι δεν ξέρει εκ των προτέρων ποιο θα είναι αυτό.

Ο HTTPd είναι αυτός που διαχειρίζεται την πληροφορία. Για κάθε αίτηση που λαμβάνει δημιουργεί (forks) μια διεργασία-παιδί η οποία είναι υπεύθυνη στο εξής για την ολοκλήρωση της διαδικασίας απόκρισης στον αιτούντα. Με τον τρόπο



Σχήμα 4.1: Η αρχιτεκτονική του συστήματος

αυτό μπορούν παράλληλα να εξυπηρετούνται αιτήσεις από πολλούς clients, χωρίς να χρειάζεται η μία να αναμένει το πέρας των άλλων. Η διεργασία-παιδί ελέγχει και αναλύει τα στοιχεία του HTTP μηνύματος και εκτελεί τις κατάλληλες ενέργειες ή παίρνει τις αποφάσεις που τους αντιστοιχούν.

Μόλις εντοπίσει την ύπαρξη επικεφαλίδας με το profile του χρήστη τότε ξεκινάει την διαδικασία επιλογής της πληροφορίας που πρέπει να στείλει, ώστε να ανταποκρίνεται στις προτεραιότητες που έχει θέσει ο χρήστης με βάση φυσικά και τις δικές του εναλλακτικές λύσεις. Στην ενότητα 4.2 περιγράφουμε αναλυτικά τα βήματα της διαδικασίας αυτής την οποία ονομάζουμε αλγόριθμο επιλογής.

Για να ολοκληρωθεί η παραπάνω διαδικασία χρειάζεται να γνωρίζουμε το χρόνο που απαιτείται για τη μεταφορά καθενός από τα διαφορετικά είδη πληροφορίας που υπάρχουν. Ουσιαστικά δηλαδή χρειάζεται η γνώση του throughput που μπορεί να πάρει μια σύνδεση στο μονοπάτι μεταξύ client και server.

Αρχικά υπήρχε η σκέψη η μέθοδος υπολογισμού να εφαρμόζεται από τον client που κάνει την αίτηση. Αυτός ο τρόπος όμως δεν εκμεταλλεύεται υπολογισμούς που μπορεί να έχουν γίνει από άλλους clients που εξυπηρετήθηκαν ή εξυπηρετούνται από τον ίδιο server. Επέρχεται έτσι καθυστέρηση η οποία πιθανόν να μπορεί να αποφευχθεί. Επιλέξαμε λοιπόν οι υπολογισμοί να γίνονται στην πλευρά του HTTPd. Για τον ίδιο λόγο, οι υπολογισμοί δεν γίνονται από κάθε παιδί του HTTPd ξεχωριστά, αλλά από μία άλλη διεργασία, την διεργασία εκτίμησης κατάστασης δικτύου (ESTd). Οι παραπάνω επιλογές ενισχύονται από την αρχή της τοπικότητας (reference locality) η οποία φαίνεται να διέπει το σύνολο των αιτήσεων που γίνεται στους WWW servers [Bra96] [CAV96]. Αυτό σημαίνει ότι αν έλθει μια αίτηση από κάποιο client, τότε με μεγάλη πιθανότητα σε σύντομο χρονικό διάστημα, θα έλθουν και άλλες αιτήσεις από τον ίδιο client.

Η επικοινωνία των δύο οντοτήτων (HTTPd και ESTd) γίνεται με την ανταλλαγή

UDP μηνυμάτων μέσω δύο καναλιών, ένα για κάθε κατεύθυνση, τα οποία υλοποιούνται με τη χρήση unix sockets ώστε να ελαχιστοποιήσουμε το κόστος επικοινωνίας [1]. Κάθε φορά που ο HTTPd λαμβάνει μια αίτηση που αντιστοιχεί σε πληροφορία πολλαπλών τύπων, εντοπίζει το όνομα του κόμβου από τον οποίο προήλθε και το δίνει στον ESTd, ζητώντας του να υπολογίσει το throughput στο μεταξύ τους μονοπάτι.

Ο ESTd από τη μεριά του διατηρεί και ενημερώνει μια δομή δεδομένων, την *δομή καταχώρησης throughput*, στην οποία αποθηκεύει **α)** την τιμή του throughput που υπολογίζει κάθε φορά, **β)** τη χρονική στιγμή και **γ)** το όνομα του κόμβου που αντιστοιχεί. Τα στοιχεία αυτά θεωρούνται έγκυρα και μπορούν να ξαναχρησιμοποιηθούν μόνο αν ο χρόνος που έχει περάσει από τη στιγμή καταγραφής τους είναι μικρότερος από κάποιο διάστημα. Η διάρκεια του διαστήματος αυτού πρέπει να είναι μικρή (της τάξης των minutes) ώστε να μην δίνονται λανθασμένα αποτελέσματα στην περίπτωση που επέλθει μεταβολή στις συνθήκες που επικρατούν στο δίκτυο.

Όταν λοιπόν έλθει ένα νέο μήνυμα, ο ESTd ελέγχει πρώτα την παραπάνω δομή. Αν *δεν υπάρχουν* τα ζητούμενα στοιχεία ή *δεν είναι έγκυρα* πια, τότε ξεκινάει τη διαδικασία για τον υπολογισμό τους. Με την ολοκλήρωσή της στέλνει πίσω την απάντηση, αν χρειάζεται, και παράλληλα ενημερώνει την δομή καταχώρησης throughput. Με τον τρόπο αυτό βελτιώνουμε το χρόνο απόκρισης του ESTd και ελαττώνουμε το φόρτο του ιδίου και του δικτύου.

4.2 Ο Αλγόριθμος Επιλογής

Όταν ένας HTTP server πάρει μία αίτηση που περιέχει επικεφαλίδα με το profile του χρήστη, αρχικά ελέγχει αν υποστηρίζει αυτό το χαρακτηριστικό. Αν ναι, τότε προχωρεί στη λήψη των κατάλληλων ενεργειών, διαφορετικά το αγνοεί.

Το είδος της πληροφορίας που τελικά θα αποσταλεί εξαρτάται από το αποτέλεσμα του **αλγορίθμου επιλογής** (σχήμα 4.2), τα βήματα του οποίου περιγράφονται στη συνέχεια:

- **Βήμα 1ο**: Με είσοδο το ζητούμενο αντικείμενο, ο server αναζητεί στο χώρο δικαιοδοσίας του αντικείμενα που ταιριάζουν με αυτό. Όσα βρει τα τοποθετεί στη **λίστα ταύτισης** (*matched_list*), και προχωρεί στο επόμενο βήμα. Αν δεν βρει κανένα τότε σταματά και στέλνει σχετική απάντηση στον client (κωδικός κατάστασης 404-Not Found).

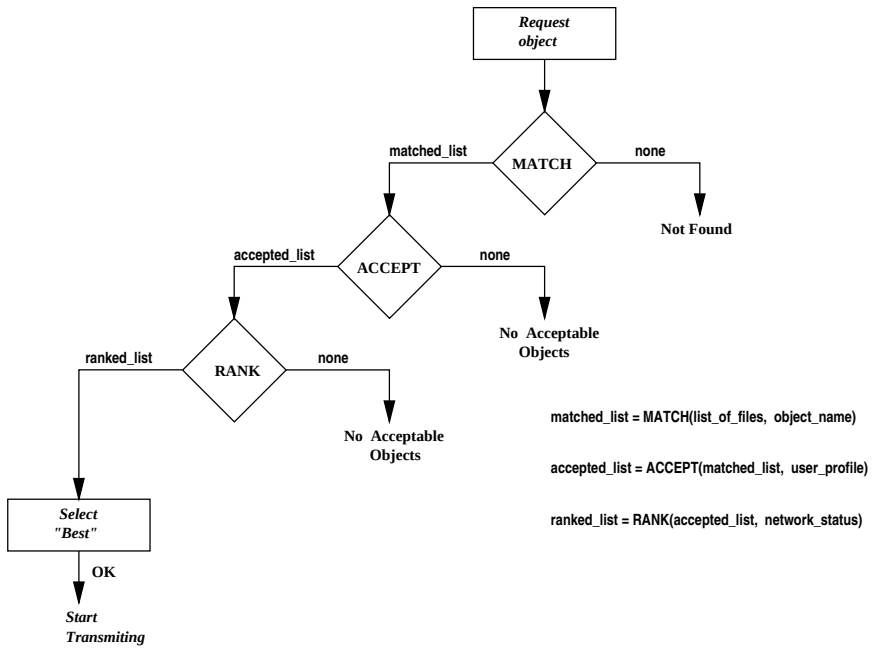
- **Βήμα 2ο**: Από την λίστα του προηγούμενου βήματος αφαιρούνται όσα αντικείμενα δεν είναι αποδεκτά σύμφωνα με το λαμβανόμενο profile του χρήστη. Για παράδειγμα αν ο χρήστης έχει ορίσει ότι δέχεται μόνο αντικείμενα τύπου video ή ήχου, τότε όποια δεν αντιστοιχούν σε κάτι τέτοιο (π.χ αρχεία κειμένου) διαγράφονται. Μετά την παραπάνω διαδικασία όσα αντικείμενα παραμείνουν αποτελούν την **λίστα αποδοχής** (*accepted_list*). Αν δεν βρεθεί

[1] Από τα τρία είδη sockets, TCP, UDP και UNIX, το τελευταίο είναι το ταχύτερο.

κανένα αντικείμενο που να ικανοποιεί τις απαιτήσεις του χρήστη, δηλ. η λίστα αποδοχής είναι κενή, τότε ο αλγόριθμος τελειώνει, και ο χρήστης ενημερώνεται ανάλογα. Για το σκοπό αυτό ορίσαμε ένα νέο κωδικό κατάστασης. Ο κωδικός αυτός ανήκει στην κατηγορία 4xx (η αίτηση δεν μπορεί να διεκπεραιωθεί), είναι ο 420 και σημαίνει ότι κανένα από τα αντικείμενα που υπάρχουν δεν ταιριάζει με το profile του χρήστη ή δεν πληροί τις προϋποθέσεις που τίθενται σε αυτό (No Acceptable Objects).

- **Βήμα 3ο**: Τα αντικείμενα της λίστας αποδοχής κατατάσσονται με βάση τις προτεραιότητες που έχει θέσει ο χρήστης (η δομή του profile τις ορίζει), και την εκτίμηση της παρούσας κατάστασης του δικτύου μεταξύ του client και του server. Έτσι δημιουργείται η **λίστα κατάταξης** (*ranked_list*). Για παράδειγμα έστω ότι ο χρήστης αποδέχεται ποιότητα υπηρεσίας video, ήχου και κειμένου με προτεραιότητες σύμφωνα με την αναγραφόμενη σειρά. Αν υπάρχουν αντικείμενα που αντιστοιχούν σε αυτού του είδους τις απαιτήσεις, αλλά η κατάσταση του δικτύου δεν επιτρέπει την έγκαιρη μεταφορά του video στα χρονικά πλαίσια που ο χρήστης έχει θέσει στο profile του, τότε το αντικείμενο που αντιστοιχεί στο video απορρίπτεται και κατατάσσονται μόνο τα υπόλοιπα. Αν όλα τα αντικείμενα απορριφθούν και η λίστα κατάταξης μείνει κενή, τότε επιστρέφεται και πάλι ο κωδικός κατάστασης 420 για να ενημερωθεί σχετικά ο χρήστης.

- **Βήμα 4ο**: Αμέσως μετά το βήμα 3, ο server αρχίζει την αποστολή του αντικειμένου που βρίσκεται πρώτο στη λίστα κατάταξης (κωδικός κατάστασης 200-OK).



Σχήμα 4.2: Ο αλγόριθμος επιλογής

4.3 Καθορισμός Προτιμήσεων Χρήστη

Υπάρχουν περιπτώσεις όπως είπαμε στην αρχή του κεφαλαίου, όπου μια πληροφορία μπορεί να βρίσκεται σε πολλές μορφές, και εξηγήσαμε τους λόγους για τους οποίους διαφορετικοί χρήστες δίνουν διαφορετικές προτεραιότητες σε καθένα από αυτούς. Αυτή η διαφοροποίηση είναι που ορίζει το profile του κάθε χρήστη. Όμως το πλήθος και η ταυτότητα των χρηστών ενός WWW server δεν είναι, ούτε και μπορεί να γίνει γνωστή. Έτσι δεν είναι εφικτή η δημιουργία μιας βάσης δεδομένων με αντιστοίχιση χρήστη-profile. Λύση δεν αποτελεί επίσης η γνωστοποίηση του profile μόνο την πρώτη φορά που ο χρήστης συνδέεται στον server (π.χ μέσω μιας html φόρμας), δεδομένου ότι το profile μπορεί να αλλάξει, αλλά και γιατί κάτι τέτοιο προϋποθέτει περισσότερη δουλειά από αυτόν που παρέχει την πληροφορία.

Επιλέξαμε έτσι η γνωστοποίηση του profile να γίνεται μέσω του πρωτοκόλλου HTTP, λύση που είναι απλή αλλά ταυτόχρονα απόλυτα δυναμική.

4.3.1 Η Επικεφαλίδα UserProfile

Τα στοιχεία λοιπόν που καθορίζουν τις προτεραιότητες και τις κατηγορίες αποδοχής του χρήστη μεταφέρονται στον server, στο τμήμα των επικεφαλίδων του HTTP μηνύματος. Γι' αυτό το σκοπό έχουμε ορίσει μια **νέα** επικεφαλίδα, την οποία ονομάσαμε **UserProfile**. Η επικεφαλίδα αυτή ανήκει στην κατηγορία των *επικεφαλίδων αίτησης* και πρέπει να μεταφέρεται στον τελικό server που έχει την πληροφορία άσχετα αν η αίτηση διέρχεται ή όχι μέσα από proxy servers. Η περιγραφή της σε BNF (Backus-Naur Form) μορφή [Cro82] είναι:

```
UserProfileHeader = "UserProfile" ":" preference*4(MIDpreference)
          preference = category["["timelimit"]"]
          category = "Video" |"Audio" |"Image"
               |"Application" |"Text"
          MID = "-->"
          timelimit = 1*DIGIT[("." | ",")1*DIGIT]
          DIGIT = "0"..."9"
```

Στόχοι μας ήταν ο ορισμός του profile να είναι απλός, σύντομος, εύκολα κατανοητός και αρκετά εκφραστικός. Ο χρήστης έχει να επιλέξει μία ή περισσότερες από τις πέντε κατηγορίες πληροφορίας (Video, Audio, Image, Application [2], Text) και να τις διατάξει σε σειρά σύμφωνα με τις προτιμήσεις του, διαχωρίζοντάς τες με την ακολουθία χαρακτήρων ‘‘-->’’. Η προτεραιότητα μιας επιλογής καθορίζεται από τη θέση της σε σχέση με τις άλλες και ελαττώνεται καθώς προχωράμε από αριστερά προς τα δεξιά. Παράλληλα ο χρήστης μπορεί για μερικές ή όλες τις επιλογές του να

[2]Στην κατηγορία Application ανήκουν τα αρχεία postscript, pdf, tiff, doc κ.α

ορίσει το χρονικό διάστημα [3] το οποίο είναι διατεθειμένος να περιμένει για να λάβει την αντίστοιχη πληροφορία.

Ο ορισμός των πέντε κατηγοριών αντιστοιχεί σε πέντε από τους βασικούς τύπους πληροφορίας όπως έχουν τυποποιηθεί για τα MIME (Multipurpose Internet Mail Extensions) μηνύματα [BF93].

Στη συνέχεια δίνουμε μερικά χαρακτηριστικά παραδείγματα profiles και εξηγούμε τους πιθανούς λόγους επιλογής τους από κάποιους χρήστες.

- *Text* :: Ο χρήστης χρησιμοποιεί έναν text-based browser, συνεπώς δεν τον ενδιαφέρει πληροφορία που δεν έχει την δυνατότητα να δει. Επίσης, μπορεί ο χρήστης να συνδέεται με ένα πολύ αργό σύνδεσμο επικοινωνίας (π.χ έχει συνδεθεί με modem) οπότε δεν θέλει πιο σύνθετη πληροφορία, που προκαλεί μεγάλη καθυστέρηση.

- *Application[3]−− >Text* :: Ο χρήστης θέλει να πάρει κείμενα καθένα από τα οποία βρίσκεται σε πολλούς τύπους (π.χ τα RFCs τα συναντάμε σε απλή και σε postsrcipt μορφή), αλλά δεν μπορεί να περιμένει παραπάνω από 3 min για κάποιο τύπο της κατηγορίας Application. Το κείμενο σε απλή μορφή, το οποίο είναι συνήθως μικρότερο σε μέγεθος και συνεπώς θα έλθει πιο γρήγορα, είναι αρκετό.

- *Video−− >Audio* :: Ο χρήστης έχει συγκεκριμένους λόγους (π.χ θέλει να δει ή να ακούσει τουλάχιστον ειδήσεις) ή συνδέεται μόνο με servers που βρίσκονται στο δικό του τοπικό δίκτυο, οπότε μπορεί να πάρει πληροφορία απαιτητική σε bandwidth (αρκεί αυτή να υπάρχει).

- *Video[5]−− >Audio[5]−− >Application* :: Ο χρήστης ενδιαφέρεται για κάποιες πληροφορίες (π.χ διαφημιστικές παρουσιάσεις), ωστόσο δεν επιθυμεί να περιμένει για διάστημα μεγαλύτερο των 5 min, προκειμένου να πάρει την πληροφορία που αντιστοιχεί σε video ή ήχο. Του αρκεί να τις πάρει και σε κείμενο (π.χ το διαφημιστικό φυλλάδιο).

- *Video[1]−− >Audio[1]−− >Application[1]−− >Image[1]−− >Text[1]* ::
  Ο χρήστης δεν αποκλείει κανένα τύπο πληροφορίας δίνοντας κατά φθίνουσα σειρά προτεραιότητα στο Video, Audio, Application, Image, Text. Εκείνο όμως που κυρίως τον ενδιαφέρει είναι ο χρόνος μεταφοράς οποιουδήποτε τύπου και αν σταλεί, να είναι της τάξης του 1 min.

Ανάλογα με την περίπτωση θα μπορούσαμε να σκεφτούμε άλλα παραδείγματα profiles χρηστών.

Ο τρόπος με τον οποίο ο client πληροφορείται το αν πρέπει ή όχι να στείλει την επικεφαλίδα *UserProfile*, είναι ίδιος με τη διαδικασία που οι HTTP servers ακολουθούν για να βρούν σε ποιά κατηγορία MIME μηνυμάτων ανήκει η πληροφορία που επιστρέφουν. Η εύρεση αυτή γίνεται με την εξέταση της κατάληξης του

[3] Το χρονικό διάστημα ορίζεται σε min.

ονόματος του αρχείου και την χρήση της αντιστοίχισης ‘‘κατάληξη--MIME τύπος’’ που υπάρχει στο αρχείο διαμόρφωσης (configuration file) του server. Έτσι για παράδειγμα ένα όνομα με κατάληξη *.gif* προφανώς ανήκει σε αρχείο εικόνας και κατατάσσεται στον MIME τύπο image. Επίσης αρχεία με κατάληξη *.html, .mpeg, .au* κατατάσσονται στους τύπους text, video και audio αντίστοιχα. Παρόμοια λοιπόν και εμείς ορίσαμε μια νέα κατάληξη, την *.multitype*, η οποία δεν αντιστοιχεί σε κανένα τύπο. Χρησιμοποιείται όμως για την γνωστοποίηση, στην πλευρά του client, του γεγονότος ότι η συγκεκριμένη επιλογή αντιστοιχεί σε πολλαπλά είδη πληροφορίας, και χρειάζεται να σταλεί το profile του χρήστη.

4.4 Μέθοδος Εκτίμησης Throughput

Μέχρι τώρα περιγράψαμε τη δομή και τον τρόπο γνωστοποίησης των profiles στους WWW servers. Τα profiles αυτά χρησιμοποιούνται στο 2ο βήμα του αλγορίθμου επιλογής για την δημιουργία της λίστας αποδοχής. Στο επόμενο βήμα του αλγορίθμου απαιτείται η γνώση της κατάστασης του δικτύου. Για να υπολογίσουμε λοιπόν το throughput σε κάποιο μονοπάτι του δικτύου, χρησιμοποιούμε μηνύματα του πρωτοκόλλου ICMP. Η επιλογή αυτή έγινε επειδή:

α) εισάγεται μικρό ποσό κυκλοφορίας στο δίκτυο

β) οι μετρήσεις λαμβάνονται αρκετά γρήγορα

γ) δεν χρειάζεται εγκατάσταση ειδικού λογισμικού στον απομακρυσμένο κόμβο

δ) δεν χρειάζεται να έχουμε πρόσβαση σε αυτόν

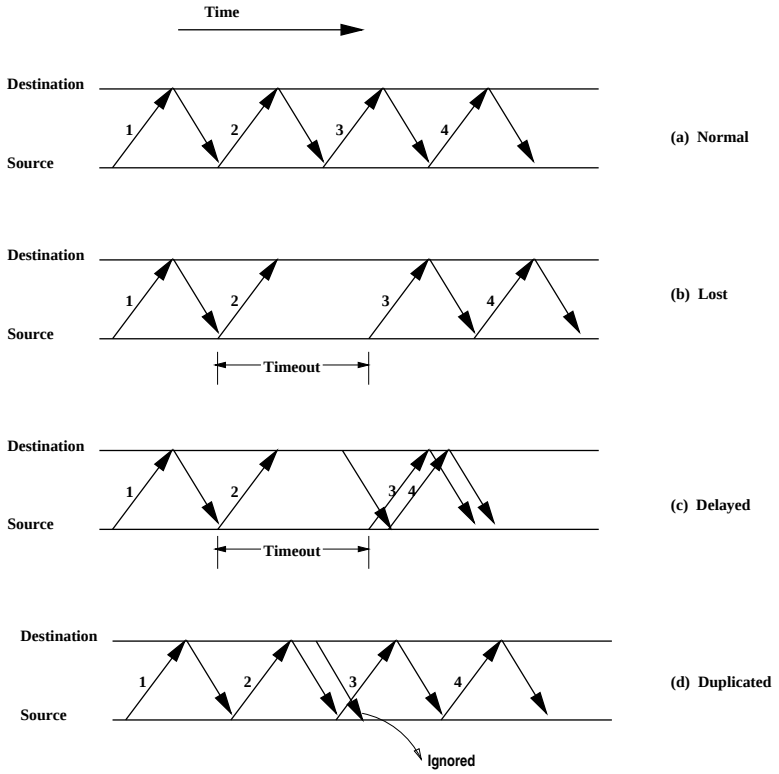
Όλα τα παραπάνω ανταποκρίνονται στους στόχους σχεδίασης που είχαμε θέσει στην ενότητα 1.1.

Ειδικότερα επιδιώκουμε να υπολογίσουμε την καθυστέρηση που ένα μήνυμα υφίσταται στην διαδρομή μεταξύ δύο σημείων, το ποσοστό απωλειών και το throughput που μπορεί να πάρει μια σύνδεση μεταξύ αυτών.

Για να το επιτύχουμε αυτό στέλνουμε ακολουθίες ICMP μηνυμάτων του είδους request/reply με προορισμό τον κόμβο που κάνει τις HTTP αιτήσεις. Στην αρχή του πεδίου *DATA* καταγράφουμε τη χρονική στιγμή της αποστολής τους, ενώ στο πεδίο *SEQUENCE* τον αύξοντα αριθμό που τους αντιστοιχεί. Αφού όπως αναφέραμε, η απάντηση που αντιστοιχεί στο συγκεκριμένο μήνυμα είναι το *ίδιο* μήνυμα, εκτός τα πεδία *TYPE* (έχει γίνει τώρα reply) και *CHECKSUM*, μπορούμε καταγράφοντας την χρονική στιγμή της άφιξης και αφαιρώντας τη χρονοσφραγίδα που είχαμε βάλει πιο πριν, να βρούμε το χρόνο που το συγκεκριμένο μήνυμα χρειάστηκε για να διανύσει τη διαδρομή *Source-Destination-Source.*

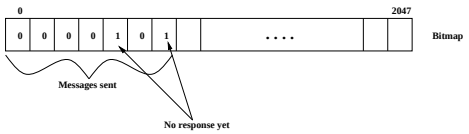
Όμως τα ICMP μηνύματα μεταφέρονται μέσα σε πακέτα, του αναξιόπιστου από άποψη εξασφάλισης μεταφοράς, IP πρωτοκόλλου και επομένως είναι δυνατόν κάποια

απ' αυτά να χαθούν, να καθυστερήσουν ή να δημιουργηθούν πολλαπλά αντίγραφά τους.



Σχήμα 4.3: Πιθανά στιγμιότυπα ακολουθιών ICMP Echo Request/Reply μηνυμάτων

Για να κάνουμε ευσταθείς (robust) αλλά και αποδοτικούς (effective) υπολογισμούς πρέπει να αντιμετωπίζουμε όλες τις προαναφερθείσες καταστάσεις. Ένα νέο πακέτο στέλνεται αμέσως μόλις ληφθεί η απάντηση για το προηγούμενο. Κάθε μήνυμα που στέλνεται σημειώνεται ως ''εν αποστολή'' (set) σε ένα πίνακα (σχήμα 4.4) στη θέση που αντιστοιχεί στον αύξοντα αριθμό του. Με την άφιξη της απάντησης σημειώνεται ως ''αφιχθέν'' (reset). Ο πίνακας αυτός, πίνακας αναχωρήσεων - αφίξεων, έχει μέγεθος 2048 και οι θέσεις του χρησιμοποιούνται κυκλικά όταν γεμίσει.



Σχήμα 4.4: Πίνακας αναχωρήσεων - αφίξεων

Επειδή κάποια από τα μηνύματα μπορεί να καθυστερήσουν ή να χαθούν (λόγω π.χ κάποιας στιγμιαίας δυσλειτουργίας ενός τμήματος του δικτύου ή λόγω συμφόρησης από την εισαγωγή νέας κυκλοφορίας) και επειδή δεν μπορούμε να περιμένουμε πολύ (δεν ξέρουμε άλλωστε πόσο) έχουμε προκαθορίσει ένα συγκεκριμένο όριο χρόνου αναμονής. Αν λοιπόν για τόσο χρόνο, από τη στιγμή αποστολής του πακέτου, δε ληφθεί καμία απάντηση, προχωράμε στην αποστολή του επόμενου (σχήμα 4.3(b)). Αν φτάσει καθυστερημένο, τα στοιχεία του και πάλι καταγράφονται και μαρκάρεται ως

‘‘αφιχθέν’’ στον πίνακα καταχώρησης, ενώ ταυτόχρονα ‘‘πυροδοτεί’’ την αποστολή νέου ICMP μηνύματος (σχήμα 4.3(c)). Αν μέχρι την λήξη του διαστήματος μέτρησης δεν έρθει απάντηση για το συγκεκριμένο μήνυμα τότε προφανώς θεωρείται χαμένο. Το διάστημα μέτρησης τελειώνει με την λήψη της απάντησης για το τελευταίο μήνυμα της ακολουθίας ή τη λήξη του χρόνου αναμονής γι’ αυτό.

Αν τώρα ένα μήνυμα επιστραφεί περισσότερες από μια φορές (duplication) τότε μόνο την πρώτη λαμβάνεται υπόψη, ενώ τις υπόλοιπες αγνοείται, εξασφαλίζοντας έτσι την ορθότητα των μετρήσεων (σχήμα 4.3(d)).

Χρησιμοποιώντας λοιπόν την παραπάνω τεχνική μπορούμε να εκτιμήσουμε προσεγγιστικά το throughput που μπορεί να πάρει μια σύνδεση μεταξύ δύο σημείων, ως εξής: στέλνουμε διαδοχικά προκαθορισμένο αριθμό ICMP echo request/replay πακέτων και καταγράφουμε το χρόνο round-trip εκείνων που επιστρέφουν. Όταν ολοκληρωθεί η λήψη ή ο χρόνος αναμονής για όλα τα πακέτα υπολογίζουμε το μέσο όρο (average) του χρόνου round-trip καθώς και το ποσοστό απωλειών:

$$avgrtt = (\sum_{i=1}^{N} rtt_i)/N \tag{4.1}$$

$$loss = (transmitted - received) * 100/transmitted \tag{4.2}$$

όπου *N = αριθμός πακέτων που ελήφθησαν*

Υπολογίζοντας λοιπόν αυτά και γνωρίζοντας το μέγεθος των ICMP πακέτων που στέλνουμε, βρίσκουμε το throughput σύμφωνα με τη σχέσεις:

$$thrpt = datalen/(avgrtt/2) \tag{4.3}$$

$$lthrpt = thrpt - (loss/100) * thprt \tag{4.4}$$

όπου *datalen = μέγεθος ICMP πακέτων*

Η σχέση (4.3) υπολογίζει το throughput όταν δεν υπάρχουν απώλειες, ενώ η σχέση (4.4) λαμβάνει υπόψη της και το ποσό του bandwidth που θα σπαταληθεί λόγω των επανεκπομπών που επιβάλλει το TCP ώστε να ληφθούν όλα τα πακέτα.

Προφανώς πρέπει να χρησιμοποιήσουμε το μισό του avgrtt (βλ. σχέση (4.3)) γιατί ο χρόνος που κρατάει αντιστοιχεί στην καθυστέρηση που το πακέτο υφίσταται για να φτάσει στον απέναντι κόμβο και να επιστρέψει.

Παρατηρήσεις:

Το throughput που μια σύνδεση λαμβάνει εξαρτάται από το χρόνο που τα δεδομένα της, χρειάζονται για να μπουν πάνω στον σύνδεσμο επικοινωνίας και είναι συνάρτηση της χωρητικότητας του συνδέσμου και των πηγών που ανταγωνίζονται γι’ αυτήν. Εξαρτάται δηλαδή από το χρόνο εισαγωγής (transmission time, t_{trans}) στο μέσο μετάδοσης. Ο χρόνος round trip που εμείς καταγράφουμε και χρησιμοποιούμε στους υπολογισμούς εμπεριέχει, εκτός από τον παραπάνω χρόνο, και το χρόνο διάδοσης (propagation time, t_{prop}) πάνω στο μέσο μετάδοσης, δηλαδή $RTT = t_{trans} + t_{prop}$. Έτσι οι εκτιμήσεις του throughput, που κάνουμε, δίνουν τιμές οι οποίες είναι μικρότερες

των πραγματικών. Η προσέγγιση θα είναι τόσο καλύτερη, όσο μικρότερο είναι το ποσοστό του χρόνου διάδοσης στον μετρούμενο χρόνο round trip.

Η σχέση 4.4 , με την οποία προσπαθούμε να εκτιμήσουμε το throughput, δίνει αξιόπιστα αποτελέσματα όταν υπάρχει *ένα* περιοριστικό σημείο στο μονοπάτι από το οποίο διέρχονται τα πακέτα. Αν τα σημεία στα οποία τα πακέτα αναγκάζονται να περιμένουν είναι περισσότερα, τότε τα αποτελέσματα που θα πάρουμε θα είναι αρκετά μικρότερα από ότι ισχύει στην πραγματικότητα. Σαν παράδειγμα θα μπορούσαμε να αναφέρουμε ένα μονοπάτι με δύο περιοριστικά σημεία τα οποία μπορούν να παράσχουν το ίδιο ακριβώς throughput. Αν τα ICMP πακέτα που στέλνουμε περιμένουν στον ενταμιευτή κάθε σημείου τότε το throughput που θα υπολογίσουμε θα είναι το μισό από το πραγματικό.

Τέτοια μοντέλα όμως είναι κυρίως θεωρητικά, αφού στην πράξη το περιοριστικό σημείο είναι πάντα ένα, εκείνο που είναι περισσότερο κορεσμένο και δίνει το *ελάχιστο* bandwidth στην σύνδεση. Εξάλλου σύμφωνα με μετρήσεις που έχουν γίνει [SB95] σε αρκετά μονοπάτια πάνω από το Internet, προκύπτει ότι ο σύνδεσμος συμφόρησης είναι πράγματι *ένας* και μάλιστα τις περισσότερες φορές *συγκεκριμένος* [4].

Εκείνο λοιπόν που έχει μεγαλύτερη σημασία στην περίπτωση μας είναι κατά πόσο ο φόρτος ή η θέση του περιοριστικού σημείου πάνω στο μονοπάτι αλλάζει και ειδικότερα πόσο συχνά γίνεται αυτό σε σχέση με την διάρκεια κατά την οποία θέλουμε να ισχύει η μέτρηση που πραγματοποιούμε. Η διάρκεια αυτή ουσιαστικά είναι το χρονικό διάστημα μεταφοράς των δεδομένων, το οποίο είναι ως επί το πλείστον, της τάξης των minutes. Πιστεύουμε ότι γενικά η συμπεριφορά του περιοριστικού συνδέσμου δεν μεταβάλλεται δραματικά σε τόσο μικρή χρονική κλίμακα. Κάτι τέτοιο εξάλλου φαίνεται στο [SB95] αλλά και στις μετρήσεις που κάναμε και παραθέτουμε στο παράρτημα Β.

Όσο μεγαλύτερο είναι το ICMP πακέτο τόσο περισσότερο επηρεάζεται από την υπόλοιπη κυκλοφορία ή τόσο μεγαλύτερη είναι η πιθανότητα να περιμένει στην ουρά κάποιου ενδιάμεσου συστήματος. Παραμένει δηλαδή στο δίκτυο περισσότερο χρόνο και επομένως μας μεταφέρει πιο αξιόπιστη πληροφορία. Το μέγεθός του όμως δεν πρέπει να ξεπερνάει τη *μέγιστη μονάδα μεταφοράς* (Maximum Transfer Unit, MTU) των συνδέσμων επικοινωνίας απ' όπου διέρχεται. Κάτι τέτοιο θα ξεκινούσε τη διαδικασία κερματισμού/επανασυγκόλλησης (fragmenation/reassemply) του πρωτοκόλλου IP στο πακέτο, με συνέπεια τα αποτελέσματα που θα πάρουμε να είναι ανακριβή. Γι' αυτούς τους λόγους επιλέξαμε τα πακέτα που στέλνουμε να είναι 1000 bytes (το MTU των συνδέσμων επικοινωνίας είναι συνήθως γύρω στα 1400 bytes).

4.5 HTTP Συνδέσεις Διαρκείας

Στην παρούσα εργασία για την βελτίωση της απόδοσης του WWW έχουμε κάνει επίσης τις απαραίτητες επεκτάσεις στο πρωτόκολλο HTTP και τις κατάλληλες

[4] Για παράδειγμα το περιοριστικό σημείο μεταξύ κόμβων της Ευρώπης και της Αμερικής είναι συνήθως οι υπερατλαντικοί σύνδεσμοι.

αλλαγές στις διεργασίες που το υλοποιούν έτσι ώστε να τους δώσουμε τη δυνατότητα να εγκαθιστούν συνδέσεις διαρκείας. HTTP σύνδεση διαρκείας είναι εκείνη η HTTP σύνδεση η οποία θα διαρκέσει και μετά από το παρόν ζεύγος αίτησης/απάντησης. Με τον τρόπο αυτό επιτρέπεται στον client να περάσει πάνω από την ίδια σύνδεση πολλές αιτήσεις, αποφεύγοντας ή διορθώνοντας αρκετά από τα προβλήματα απόδοσης που περιγράψαμε στην ενότητα 2.5.

Ειδικότερα τα πλεονεκτήματα των HTTP συνδέσεων διαρκείας είναι:

- Εξοικονόμηση κύκλων μηχανής και μνήμης με την εγκατάσταση λιγότερων TCP συνδέσεων και συνεπώς χρήση λιγότερων TCP PCBs (protocol control blocks),δομών δηλαδή που κρατούν στοιχεία σχετικά με τις συνδέσεις.
- Μείωση της συμφόρησης του δικτύου αφού λιγότερες συνδέσεις σημαίνουν μείωση των πακέτων συγχρονισμού που απαιτούνται κατά την έναρξη και το πέρας αυτών. Ταυτόχρονα δίδεται στον αλγόριθμο αργής εκκίνησης του TCP επαρκής χρόνος για την προσαρμογή στην κατάσταση συμφόρησης του δικτύου και την αποστολή δεδομένων από την πηγή με ικανοποιητικό ρυθμό.
- Μείωση της καθυστέρησης, με την δυνατότητα αποστολής των αιτήσεων διαδοχικά (pipelining), χωρίς να χρειάζεται αναμονή για την απάντηση της καθεμιάς.

4.5.1 Η Επικεφαλίδα Connection

Για την υλοποίηση των συνδέσεων διαρκείας ακολουθήσαμε τις προτάσεις της ομάδας εργασίας της IETF (Internet Engineering Task Force Working Group) για το HTTP [5]. Σύμφωνα με αυτήν για την ανταλλαγή πληροφορίας σχετικής με την σύνδεση μεταξύ του client και του server πρέπει να χρησιμοποιείται η επικεφαλίδα **Connection**, η οποία ανήκει στην κατηγορία των *επικεφαλίδων γενικού περιεχομένου.* Η περιγραφή της σε BNF μορφή είναι:

```
Connection Header = ``Connection'' ``:'' 1#field-name
```

Με βάση τον ορισμό αυτό στην επικεφαλίδα Connection μπορούμε να μεταφέρουμε μια λίστα από πεδία, καθένα από τα οποία έχει και κάποιο συγκεκριμένο σκοπό. Αν κάποιο από τα πεδία αυτά έχει την τιμή ‘‘Keep-Alive’’, τότε αυτό δεικνύει την δυνατότητα και ‘‘επιθυμία’’ του αποστολέα για σύνδεση διαρκείας.

Για να εγκατασταθεί λοιπόν μια σύνδεση διαρκείας πρέπει να εισαχθεί από τον client και στη συνέχεια από τον server, στο πρώτο τμήμα των HTTP μηνυμάτων που ανταλλάσσονται, η επικεφαλίδα: **Connection: Keep-Alive**. Η ύπαρξή της σημαίνει ότι η σύνδεση θα μείνει ανοιχτή και για την επόμενη αίτηση. Για την περαιτέρω διατήρηση της σύνδεσης, η επικεφαλίδα με την συγκεκριμένη τιμή πρέπει

[5]HTTP Internet Draft, έκδοση 1.1, Ιούλιος 1996 (draft-ietf-http-v11-spec-06)

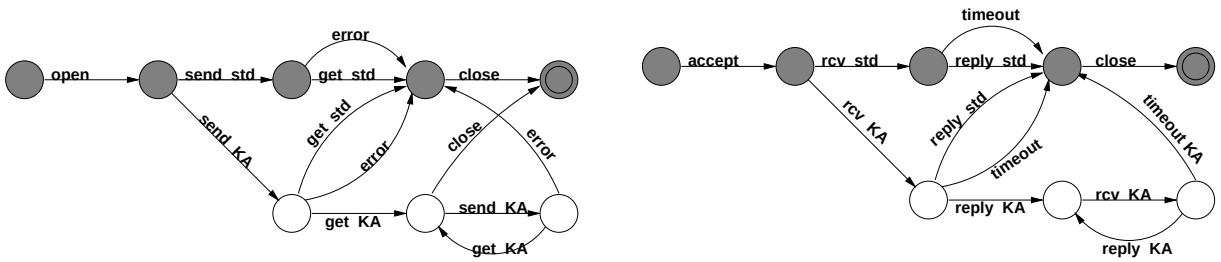
να στέλνεται σε κάθε αίτηση και απάντηση. Η σύνδεση κλείνει αν ένα από τα δύο μέρη την κλείσει.

Πρέπει να σημειώσουμε εδώ ότι η επικεφαλίδα *Connection: Keep-Alive* μιας αίτησης που εξυπηρετείται μέσω κάποιου proxy server, αφαιρείται από αυτόν πριν στείλει την στείλει στον επόμενο server, εκτός αν και ο proxy server υποστηρίζει το χαρακτηριστικό αυτό οπότε βάζει νέα επικεφαλίδα *Connection: Keep-Alive* κ.ο.κ. Ουσιαστικά δηλαδή η εγκατάσταση μιας σύνδεσης διαρκείας μπορεί να διαπραγματευτεί *μόνο* μεταξύ ''γειτονικών'' αποστολέων/παραληπτών.

4.5.2 Η Υλοποίηση των Συνδέσεων Διαρκείας

Με βάση τα προηγούμενα, η συμπεριφορά και οι ενέργειες των HTTP client και server, όπως τους επεκτείναμε, για την εγκατάσταση ή μη συνδέσεων διαρκείας, έχουν ως εξής:
Όταν ο server λάβει μια αίτηση χωρίς την επικεφαλίδα *Connection: Keep-Alive* την εξυπηρετεί κανονικά κλείνοντας την σύνδεση αμέσως μετά την ολοκλήρωση της αποστολής της απάντησης. Αν υπάρχει η συγκεκριμένη επικεφαλίδα, τότε στην απάντηση του περικλείει την ίδια επικεφαλίδα δηλώνοντας έτσι ότι θα διατηρήσει την σύνδεση ανοιχτή και για την επόμενη αίτηση. Ο client από την μεριά του με τη λήψη μιας απάντησης με την επικεφαλίδα *Connection: Keep-Alive* πληροφορείται ότι μπορεί να περάσει και την επόμενη αίτησή του πάνω από την ίδια σύνδεση, διαφορετικά είναι υποχρεωμένος να την κλείσει και αυτός από την μεριά του.

Στα σχήματα 4.5(a) και 4.5(b) φαίνονται τα διαγράμματα καταστάσεων του client και του server αντίστοιχα.



(a) HTTP client

(b) HTTP server

Σχήμα 4.5: Διαγράμματα καταστάσεων

Οι σκιασμένοι κύκλοι παριστάνουν τις καταστάσεις του πρότυπου HTTP ενώ οι υπόλοιποι παριστάνουν τις επιπλέον καταστάσεις που απαιτούνται για την υλοποίηση της νέας δυνατότητας. Οι μεταβάσεις κατάστασης που έχουμε σχεδιάσει σημαίνουν τα ακόλουθα:

- Μεταβάσεις Client

open	νέα σύνδεση με HTTP server
send std	αποστολή πρότυπης HTTP αίτησης (*χωρίς σύνδεση διαρκείας*)
get std	λήψη πρότυπης HTTP απάντησης
send KA	αποστολή αίτησης με την επικεφαλίδα *Connection: Keep-Alive*
get KA	λήψη απάντησης με την επικεφαλίδα *Connection: Keep-Alive*
error	ανίχνευση λάθους
close	περάτωση σύνδεσης

- Μεταβάσεις Server

accept	αποδοχή εγκατάστασης νέας σύνδεσης
rcv std	λήψη πρότυπης HTTP αίτησης
reply std	αποστολή πρότυπης HTTP απάντηση
rcv KA	λήψη HTTP αίτησης με επικεφαλίδα *Connection: Keep-Alive*
reply KA	HTTP απάντηση με την επικεφαλίδα *Connection: Keep-Alive*
timeout	λήξη της προθεσμίας χρόνου για την επιστροφή απάντησης
timeout KA	υπέρβαση προθεσμίας χρόνου αδράνειας της σύνδεσης διαρκείας
close	περάτωση σύνδεσης

Εάν πολλοί clients που υποστηρίζουν το παραπάνω χαρακτηριστικό συνδεθούν στον ίδιο server τότε μπορεί να προκληθεί υπερφόρτωση από τις διεργασίες-παιδιά που διατηρούν και εξυπηρετούν τις HTTP συνδέσεις. Το πρόβλημα θα εμφανιστεί πιο έντονο αν κάποιες από τις διεργασίες αυτές παραμένουν αδρανείς για μεγάλο χρονικό διάστημα επειδή ο client με τον οποίο διατηρούν τη σύνδεση δεν κάνει αιτήσεις. Για την αποφυγή αυτού του προβλήματος έχουμε προκαθορίσει τον αριθμό των διεργασιών-παιδιά που μπορούν να διατηρούν συνδέσεις διαρκείας ταυτόχρονα. Όταν συμπληρωθεί το όριο τότε όλες οι αιτήσεις από νέους clients εξυπηρετούνται με την μέθοδο της σύνδεσης ανά αίτηση ακόμα και αν υποστηρίζουν συνδέσεις διαρκείας. Παράλλληλα έχουμε καθορίσει και το μέγιστο χρόνο (timeout_Keep_Alive) που μια διεργασία-παιδί που διατηρεί σύνδεση διαρκείας επιτρέπεται να μείνει αδρανής. Όταν ο χρόνος αυτός εκπνεύσει τότε κλείνει η σύνδεση. Αν ο client με τον οποίο ήταν συνδεδεμένη κάνει νέα αίτηση κάποια στιγμή αργότερα τότε προφανώς εγκαθίσταται νέα σύνδεση. Με τον τρόπο αυτό ελαττώνεται ο φόρτος του server, βελτιώνεται η απόδοσή του και δίνουμε τη δυνατότητα σε νέους clients που κάνουν αιτήσεις να επωφεληθούν από τα πλεονεκτήματα των συνδέσεων διαρκείας.

Το πλήθος των διεργασιών-παιδιά που μπορούν να διατηρούν συνδέσεις διαρκείας ταυτόχρονα καθώς και ο χρόνος timeout_Keep_Alive είναι τιμές που επιτρέπουμε να τίθενται από τους διαχειριστές των HTTP servers μέσα στο αρχείο διαμόρφωσης. Οι τιμές αυτές εξαρτώνται από την υπολογιστική ισχύ των μηχανημάτων στα οποία ‘‘τρέχουν’’, το πλήθος των αιτήσεων που εξυπηρετούν και την τοπικότητα (locality) των αιτήσεων αυτών.

4.6 Θέματα Συμβατότητας

Ένας HTTP server μπορεί να υποστηρίζει ή να μην υποστηρίζει την εισαγωγή προτιμήσεων από τους χρήστες, ανάλογα με τους περιορισμούς ή την πολιτική των διαχειριστών του. Για να καταστεί αυτό δυνατό ορίσαμε ένα νέο κανόνα στο αρχείο διαμόρφωσης του server. Ο κανόνας αυτός έχει τη μορφή:

```
SupportUserProfile OPTION
```

όπου η παράμετρος OPTION μπορεί να πάρει τις τιμές YES ή NO, ανάλογα με το αν θέλουμε ή όχι να επιτρέψουμε αυτό το χαρακτηριστικό.

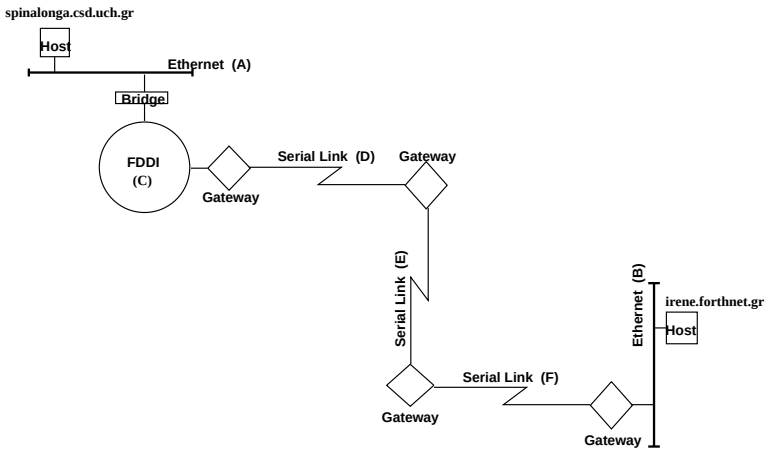
Για λόγους συμβατότητας, οι servers που δεν υποστηρίζουν το παραπάνω χαρακτηριστικό, απλά αγνοούν την επικεφαλίδα UserProfile και στέλνουν, την πρώτη επιλογή που βρίσκουν να αντιστοιχεί στην συγκεκριμένη αίτηση. Το ίδιο γίνεται και όταν ο server υποστηρίζει profiles χρηστών αλλά δεν λαμβάνει την αντίστοιχη επικεφαλίδα στην αίτηση του client.

Όσον αφορά τώρα την επέκταση του HTTP πρωτοκόλλου για την εγκατάσταση των συνδέσεων διαρκείας είναι και αυτή συμβατή με το standard HTTP αφού αιτήσεις που δεν περιλαμβάνουν την επικεφαλίδα *Connection: Keep-Alive* εξυπηρετούνται κανονικά όπως πριν, ενώ servers που δεν υποστηρίζουν το χαρακτηριστικό απλά δεν λαμβάνουν υπόψη τους την συγκεκριμένη επικεφαλίδα ούτε και την επιστρέφουν στον client, συνεπώς δεν εγκαθίσταται σύνδεση διαρκείας.

Κεφάλαιο 5

Αξιολόγηση - Μετρήσεις

Για την αξιολόγηση του συστήματος κάναμε πειράματα και μετρήσεις σε τυπικά σενάρια λειτουργίας του δικτύου, με ποικίλα profiles χρηστών και διάφορες περιπτώσεις πληροφορίας σε πολλαπλούς τύπους. Επειδή δεν είχαμε την δυνατότητα να εγκαταστήσουμε τον δικό μας client ή server σε πολλά μέρη έτσι ώστε να κάνουμε τα πειράματα σε διαφορετικές συνθήκες δικτύου, προσπαθήσαμε οι ίδιοι να προσομοιώσουμε τέτοιες συνθήκες στο μονοπάτι πάνω από το οποίο μεταφέρονται οι HTTP αιτήσεις και απαντήσεις. Για να πετύχουμε τη δημιουργία διαφορετικών συνθηκών, εγκαθιστούσαμε μία, δύο ή τρεις, ανάλογα με την περίπτωση, παράλληλες TCP συνδέσεις οι οποίες μετέφεραν σχετικά μεγάλη ποσότητα δεδομένων και επομένως χρησιμοποιούσαν ικανό ποσοστό από το bandwidth του περιοριστικού συνδέσμου του μονοπατιού. Η τοπολογία του δικτύου που παρεμβάλλεται μεταξύ των κόμβων που έγιναν τα πειράματα φαίνεται στο σχήμα 5.1.



Σχήμα 5.1: Τοπολογία Δικτύου

Ανάμεσα στα γρήγορα τοπικά δίκτυα A, B (Ethernets με χωρητικότητα 10Mbits) και C (FDDI με χωρητικότητα 100Mbits) παρεμβάλλονται τρεις WAN σύνδεσμοι, δύο σχετικά γρήγοροι (οι E, F) με χωρητικότητα 2Mbits, και ο τρίτος (D) πιο αργός (128Kbits). Ο σύνδεσμος D είναι βασικά το περιοριστικό σημείο του μονοπατιού. Το μέγεθος της ουράς εξυπηρέτησης του συνδέσμου αυτού είναι ίσο με 4 KBytes δηλαδή

8 TCP πακέτα μέγιστου μεγέθους (512 Bytes).

Αρχικά παρουσιάζονται οι τιμές των παραμέτρων του δικτύου στο μονοπάτι που έγιναν τα πειράματα, ώστε να δούμε πόσο αξιόπιστοι είναι οι υπολογισμοί μας. Στη συνέχεια δίνουμε μια σειρά από σενάρια λειτουργίας του συστήματός μας και την συμπεριφορά του σε κάθ' ένα από αυτά. Ακολουθούν τέλος, τα στοιχεία και η αξιολόγηση των συνδέσεων διαρκείας.

5.1 Μετρήσεις Παραμέτρων Δικτύου

Στην ενότητα αυτή παραθέτουμε τα αποτελέσματα των μετρήσεων που συλλέξαμε για τη διαδρομή μεταξύ Ηρακλείου-Θεσσαλονίκης και πιο συγκεκριμένα από τον κόμβο του Πανεπιστημίου Κρήτης, *spinalonga.csd.uch.gr*, στον κόμβο *irene.forthnet.gr*, του δικτύου FORTHnet. Τα χαρακτηριστικά του μονοπατιού που τους συνδέει είναι αυτά που περιγράψαμε πιο πριν. Στις γραφικές παραστάσεις που ακολουθούν απεικονίζονται οι μετρήσεις που έγιναν κατά το χρονικό διάστημα 5 συνεχόμενων ημερών, ανά περίοδο 10 min. Συνολικά έχουμε 724 δείγματα. Η γραφική παράσταση 5.2(a) δίνει το throughput που έλαβε μια TCP σύνδεση που μετέφερε ποσότητα δεδομένων ενός MByte κάθε φορά. Ο υπολογισμός του έγινε με την βοήθεια της γεννήτριας κυκλοφορίας που περιγράφουμε στο παράρτημα Α.1. Στην γραφική παράσταση 5.2(b) φαίνεται το εκτιμούμενο throughput που μας παρέχει η μέθοδος που αναλύσαμε στην ενότητα 4.4. Για τον υπολογισμό κάθε δείγματος χρησιμοποιούμε 5 ICMP πακέτα, μεγέθους 1000 bytes το καθένα. Τέλος στην γραφική παράσταση 5.2(c) φαίνεται ο μέσος χρόνος round-trip των πακέτων κάθε προσπάθειας.

Από τη μελέτη των στοιχείων των γραφικών παραστάσεων προκύπτει ότι ο μέσος όρος του throughput που πήρε η πραγματική σύνδεση είναι:

$$AvgThrpt = (\sum_{i=1}^{N} RealSample_i)/N = 80.20 Kbps \tag{5.1}$$

Συγκρίνοντας τα αντίστοιχα δείγματα προκύπτει ότι το *μέσο απόλυτο σφάλμα* είναι:

$$AvgErr = (\sum_{i=1}^{N} \mid RealSample_i - EstimatedSample_i \mid)/\mathrm{N} = 7.55 Kbps \tag{5.2}$$

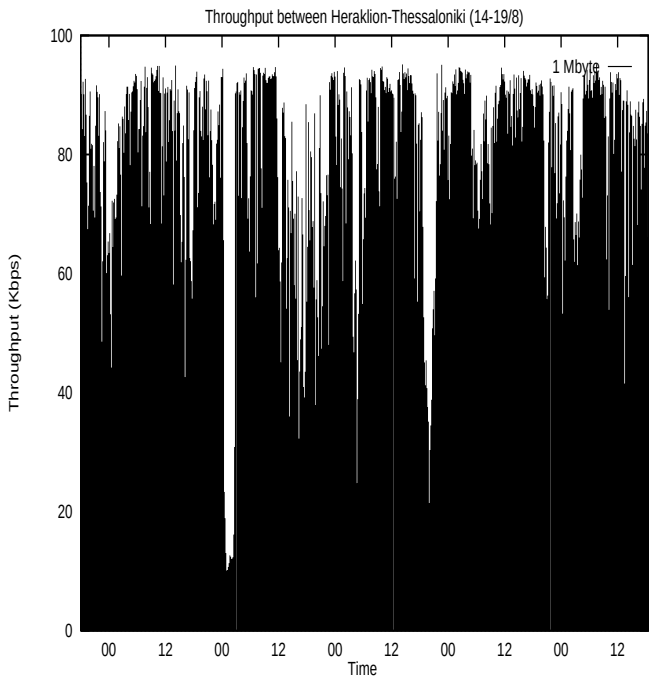
ή ποσοστό 9.41%, ενώ η *διασπορά του σφάλματος* είναι:

$$\sigma = \sqrt{\sum_{i=1}^{N}(\mid RealSample_i - EstimatedSample_i \mid^2 /N) - AvgErr^2} = 4.71 \tag{5.3}$$

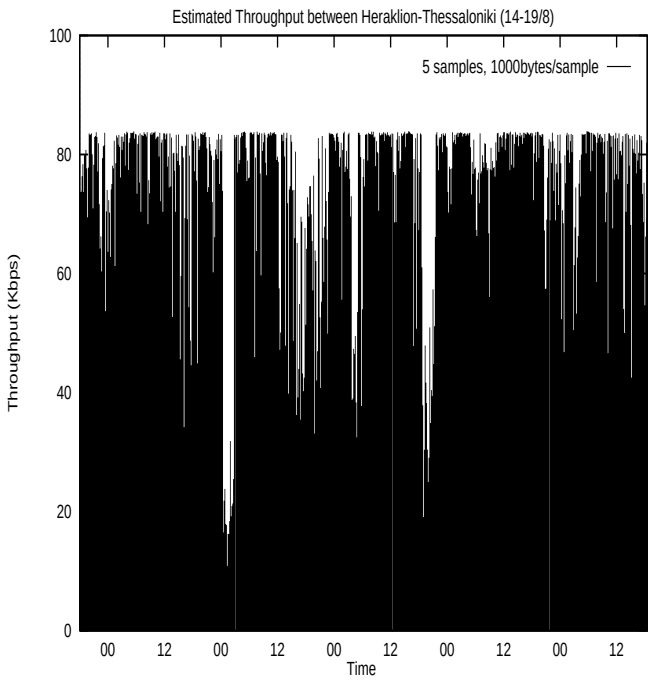
Παρατηρούμε δηλαδή ότι αν και εισάγουμε μικρό ποσό κυκλοφορίας (μόλις 5KBytes) η προσέγγιση στο πραγματικό throughput είναι πολύ ικανοποιητική και συνεπώς μπορεί να χρησιμοποιηθεί για την λήψη αποφάσεων όπως προτείνουμε στο σύστημα της παρούσας εργασίας. Στη γραφική παράσταση 5.2(c) φαίνεται

ότι ο χρόνος round-trip, όπως αναμενόταν, μεταβάλλεται αντιστρόφως ανάλογα με την τιμή του throughput. Οι μεταβολές του εκτιμούμενου throughput λοιπόν οφείλονται στις μεταβολές του χρόνου round trip που παρατηρούνται λόγω της διαφορετικής καθυστέρησης που τα πακέτα υφίστανται στους ενταμιευτές των ενδιάμεσων συστημάτων (gateways) από τα οποία διέρχονται. Για την συγκεκριμένη μέτρηση δεν έχουμε δώσει τη γραφική παράσταση με τα ποσοστά απώλειας πακέτων γιατί αυτά είναι σε όλα τα δείγματα από αμελητέα ως μηδενικά.

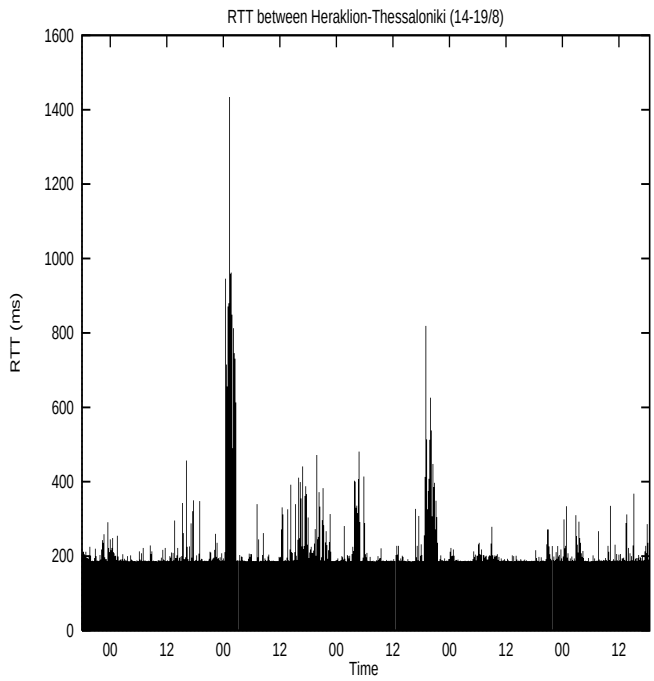
Στο παράρτημα Β παραθέτουμε αντίστοιχες μετρήσεις που έγιναν μεταξύ του κόμβου του Π.Κ και αρκετών άλλων εντός και εκτός Ελλάδας, κάνοντας επίσης τη σχετική ανάλυση και μερικές ενδιαφέρουσες παρατηρήσεις.



(a) Πραγματικό throughput



(b) Εκτιμούμενο throughput



(c) Χρόνος round trip

Σχήμα 5.2: Μετρήσεις μεταξύ Ηρακλείου-Θεσσαλονίκης

5.2 Μελέτη Σεναρίων Χρήσης του Συστήματος

Αναφέραμε στην ενότητα 2.5.3 ότι το 80% των WWW σελίδων δεν ξεπερνά τα 10 KBytes. Ταυτόχρονα όμως το υπόλοιπο 20%, είναι αυτό που εισάγει το μεγαλύτερο ποσοστό της WWW κυκλοφορίας στο δίκτυο. Πρόκειται για μεγάλες εικόνες, postscript αρχεία, αρχεία video και ήχου και άλλα σχετικά, με μέγεθος πολύ μεγαλύτερο από 10 KBytes. Το σύστημά μας εμφανίζεται ιδιαίτερα χρήσιμο, όπως θα φανεί και από τα στοιχεία που δίνουμε παρακάτω, σε εκείνες τις περιπτώσεις που έχει να επιλέξει μεταξύ τύπων μιας πληροφορίας που έχουν σημαντική διαφορά ώς προς το μέγεθος τους.

Το σύστημά μας πλεονεκτεί σε σχέση με το υπάρχον στατικό μοντέλο στις ακόλουθες περιπτώσεις:

- Όταν λόγω μεγάλης συμφόρησης αποφασίσει να στείλει πληροφορία που χρειάζεται λίγο bandwidth, σε αντίθεση με το στατικό που αγνοώντας το δίκτυο μπορεί να έστελνε μεγάλη ποσότητα πληροφορίας, η οποία θα χειροτέρευε την ήδη κακή κατάσταση του δικτύου.
- Όταν λόγω βελτίωσης των συνθηκών του δικτύου (π.χ τις νυχτερινές ώρες) αποφασίσει να στείλει ‘‘καλύτερης ποιότητας’’ πληροφορία, σε αντίθεση με το στατικό μοντέλο που στέλνει πάντα την ίδια

Στον πίνακα που ακολουθεί περιγράφουμε τις πληροφορίες, τους διαφορετικούς τύπους και τα μεγέθη τους, οι οποίες χρησιμοποιήθηκαν για τον έλεγχο του συστήματος μας. Οι πληροφορίες αυτές είναι:

- Η παρουσίαση μιας εργασίας σε html και postscript μορφή, με μικρή σχετικά διαφορά μεγεθών.
- Μια τεχνική αναφορά σε text και postscript μορφή. Το μέγεθος της δεύτερης οφείλεται κατά ένα μέρος σε κάποια σχήματα που εμπεριέχει.
- Ο χάρτης μιας περιοχής σε gif και postscript μορφή, με σημαντική διαφορά ως προς τα μεγέθη τους.
- Οι στίχοι (μορφή text) και το αρχείο ήχου (μορφή audio) ενός τραγουδιού.
- Το διαφημιστικό ταινίας κινουμένων σχεδίων σε μορφή video, audio και image.

Στη συνέχεια παρουσιάζουμε την απόφαση του συστήματός μας κάθε φορά που ζητάμε μια από τις παραπάνω πληροφορίες. Η απόφαση σε κάθε περίπτωση στηρίζεται στις προτιμήσεις του χρήστη και τις συνθήκες που επικρατούν στο δίκτυο. Φτιάχνοντας διάφορα ‘‘σενάρια’’ προτιμήσεων χρήστη και συνθηκών δικτύου, παρατηρούμε πώς αλλάζει η συμπεριφορά του συστήματος. Δίνουμε επίσης τις τιμές των παραμέτρων που το σύστημα υπολογίζει και τις τιμές που στην πραγματικότητα παίρνουμε τελικά. Πιο συγκεκριμένα, δίνουμε το εκτιμώμενο

	Video	Audio	Application	Image	Text
Παρουσίαση			36.8 KB		23 KB
Αναφορά			857 KB		42 KB
Χάρτης			1.63 MB	26.7 KB	
Τραγούδι		1.45 MB			0.4 KB
Διαφημιστικό Ταινίας	1.91 MB	130 KB		6.8 KB	

Πίνακας 5.1: Πληροφορίες διαφόρων τύπων

throughput και τους χρόνους που θα χρειαζόταν κάθε μορφή πληροφορίας για να μεταφερθεί. Δίνουμε ακόμα το πραγματικό throughput και χρόνο μεταφοράς της μορφής που τελικά επιλέχθηκε [1].

Τα πειράματά μας πραγματοποιήθηκαν σε διαφορετικές χρονικές στιγμές με διαφορετικές ‘‘αρχικές συνθήκες’’ κατάστασης δικτύου. Κατά τη διάρκεια των πειραμάτων, λόγω της αύξησης της ‘‘βοηθητικής’’ κυκλοφορίας που εισάγουμε στο δίκτυο, το σύστημά μας επιλέγει μορφές πληροφορίας που έχουν χαμηλότερη προτεραιότητα στις προτιμήσεις του χρήστη αλλά πληρούν τις προϋποθέσεις για να σταλούν (δηλ. ο χρόνος μεταφοράς τους είναι μέσα στα όρια που έχουν καθοριστεί στο profile) σε αντίθεση με άλλες μορφές, υψηλότερης προτεραιότητας, που δεν τις πληρούν. Στις περιπτώσεις που συνέβαινε αυτό, μεταφέραμε αμέσως μετά την μορφή που είχε απορριφθεί, χωρίς τη μεσολάβηση του συστήματος μας, για να δούμε αν πράγματι *δεν* πληροί τις προϋποθέσεις. Ο χρόνος που χρειάστηκε κάθε φορά και επιβεβαιώνει την ορθότητα των αποφάσεων του συστήματος, δίνεται στους παρακάτω πίνακες μέσα σε παρενθέσεις. Σε κάθε περίπτωση βλέπουμε ότι οι εκτιμήσεις που κάνει το σύστημά μας είναι πολύ κοντά στην πραγματική κατάσταση του δικτύου. Η σφάλμα στο throughput είναι πάντα μικρότερο από 1KByte/sec. Έτσι οι επιλογές του ανταποκρίνονται στους περιορισμούς που θέτει ο χρήστης.

Πρέπει να αναφέρουμε όμως ότι στις περιπτώσεις που μια απόφαση λαμβάνεται ‘‘οριακά’’ μπορεί τελικά να απαιτήσει χρόνο μεγαλύτερο απ’ όσο έχει καθοριστεί στο profile. Αυτό συμβαίνει στην 3η γραμμή του 3ου πίνακα, όπου ο εκτιμούμενος χρόνος (2.97 min) είναι μικρότερος από εκείνο που ορίζεται στο profile (3 min). Τελικά όμως χρειάστηκαν 3.13 min. Ωστόσο και σε αυτές τις περιπτώσεις ο επιπλέον χρόνος είναι πολύ μικρό ποσοστό σε σχέση με το όριο που έχει τεθεί. Στην προκειμένη περίπτωση μόλις +4.33%.

Στη συνέχεια δίνουμε το ‘‘σενάριο’’ κάθε προσπάθειας (πληροφορία, profile χρήστη) και τα αποτελέσματα των πειραμάτων όταν υπάρχουν καμία, μία, δύο ή τρεις παράλληλες συνδέσεις. Τα αποτελέσματα των πειραμάτων μας δίνονται στους παρακάτω πίνακες.

[1] Το throughput μετράται σε Kbps και οι χρόνοι σε min. Με έντονα γράμματα δείχνουμε τους χρόνους της πληροφορίας που τελικά απεστάλει.

Σενάριο 1ο
- **Πληροφορία**: Παρουσίαση
- **Profile**: Application$--$>Text

Ο χρήστης δηλώνει ότι προτιμά σε κάθε περίπτωση τα αντικείμενα που είναι σε application μορφή, ασχέτως πόσο θα καθυστερήσουν να μεταφερθούν. Αν όμως η πληροφορία δεν υπάρχει σε αυτή τη μορφή τότε δέχεται και το απλό text. Δεν υπάρχει λοιπόν περιορισμός σε χρόνο και επομένως το σύστημα αφού βρίσκει την πληροφορία στην μορφή που ο χρήστης έχει δώσει την μεγαλύτερη προτεραιότητα, στέλνει πάντα αυτή, ανεξάρτητα εάν η άλλη χρειάζεται μικρότερο χρόνο μεταφοράς. Επειδή δεν υπάρχει περιορισμός σε χρόνο, δεν χρησιμοποιούνται οι εκτιμήσεις του συστήματος γι' αυτό και δεν τις δίνουμε.

	πραγμ. χρ. μεταφοράς	
	ps	txt
αρχ. συνθ.	**0.06**	*(0.04)*
+1 σύνδεση	**0.11**	*(0.08)*
+2 συνδέσεις	**0.19**	*(0.12)*
+3 συνδέσεις	**0.23**	*(0.16)*

Σενάριο 2ο
- **Πληροφορία**: Αναφορά
- **Profile**: Application[3]$--$>Text

Στο profile του ο χρήστης ορίζει ότι προτιμά την πληροφορία σε μορφή text, από το να περιμένει περισσότερο από 3 min για μορφή application παρόμοιας πληροφορίας. Στην περίπτωση αυτή, επειδή η κατάσταση του δικτύου στο δεύτερο πείραμα ήταν καλύτερη από την πρώτη, χρειάστηκαν περισσότερες συνδέσεις για να δημιουργηθούν οι συνθήκες που προκαλούν την αλλαγή της επιλογής του συστήματος.

	εκτιμ. thrpt	εκτιμ. χρόνος		πραγμ. χρόνος		πραγμ. thrpt
		txt	ps	txt	ps	
αρχ. συνθ.	51.77	0.11	**2.21**	--	**2.1**	54.46
+1 σύνδεση	42.53	0.13	**2.69**	--	**2.85**	40.08
+2 συνδέσεις	30.60	**0.18**	3.73	**0.25**	*(3.75)*	22.40

	εκτιμ. thrpt	εκτιμ. χρόνος		πραγμ. χρόνος		πραγμ. thrpt
		txt	ps	txt	ps	
αρχ. συνθ.	80.79	0.7	**1.41**	--	**1.31**	87.23
+1 σύνδεση	52.40	0.11	**2.18**	--	**2.23**	51.16
+2 συνδέσεις	38.49	0.15	**2.97**	--	**3.13**	36.47
+3 συνδέσεις	26.47	**0.21**	4.32	**0.26**	*(4.28)*	21.54

Σενάριο 3ο
- **Πληροφορία**: Χάρτης
- **Profile**:Video[5]$--$>Application[3]$--$>Image[1]$--$>Text

Στο profile αυτό ορίζεται ότι είναι δεκτές οι μορφές video, application, image και text, με προτεραιότητα σύμφωνα την σειρά που δίνονται, αρκεί οι χρόνοι μεταφοράς τους να είναι της τάξης των 5, 3 και 1 min για τα τρεις πρώτες μορφές αντίστοιχα. Επειδή η μορφή με την μεγαλύτερη προτεραιότητα δεν υπάρχει, το σύστημα επιλέγει μεταξύ των δύο επόμενων. Αν το δίκτυο ήταν τόσο κορεσμένο που δεν μπορούσε να μεταφέρει ούτε την εικόνα, που είναι σχετικά μικρή, τότε ο χρήστης θα ειδοποιούνταν ότι κανένα από τα διαθέσιμα αντικείμενα δεν πληροί τις προϋποθέσεις του profile του, δεδομένου ότι δεν υπάρχει η πληροφορία σε text μορφή.

	εκτιμ. thrpt	εκτιμ. χρόνος		πραγμ. χρόνος		πραγμ. thrpt
		gif	ps	gif	ps	
αρχ. συνθ.	80.96	0.04	**2.69**	--	**2.45**	88.71
+1 σύνδεση	56.29	**0.06**	3.86	**0.07**	*(3.92)*	50.86
+2 συνδέσεις	37.00	**0.09**	5.80	**0.11**	*(6.07)*	32.36

Σενάριο 4ο
- **Πληροφορία**: Τραγούδι
- **Profile**: Video[5]$--$>Audio[5]$--$>Text

Ο χρήστης προτιμά το αρχείο video ή ήχου με τη σειρά που αναφέρονται αρκεί ο χρόνος μεταφοράς τους να είναι της τάξης των 5 min. Διαφορετικά του είναι αρκετή πληροφορία σε μορφή text (π.χ οι στίχοι του τραγουδιού ή άλλες σχετικές πληροφορίες).

	εκτιμ. thrpt	εκτιμ. χρόνος		πραγμ. χρόνος		πραγμ. thrpt
		txt	au	txt	au	
αρχ. συνθ.	80.05	< 0.01	**2.43**	--	**2.21**	87.88
+1 σύνδεση	51.00	< 0.01	**3.81**	--	**3.75**	51.77
+2 συνδέσεις	34.03	**0.01**	5.70	**0.01**	*(5.45)*	

Σενάριο 5ο
- **Πληροφορία**: Τραγούδι
- **Profile**: Video$--$>Audio

Ο χρήστης ενδιαφέρεται μόνο για video ή audio μορφή πληροφορίας χωρίς να θέτει χρονικά όρια. Η μορφή video έχει τη μέγιστη προτεραιότητα επειδή όμως δεν είναι διαθέσιμη στην προκειμένη περίπτωση, το σύστημα θα στέλνει πάντα την audio μορφή. Η μορφή αυτή μάλιστα, θα στέλνεται όσο κορεσμένο και αν είναι το δίκτυο, αφού δεν υπάρχει περιορισμός. Γι' αυτό το λόγο δεν δίνονται σχετικές μετρήσεις.

Σενάριο 6ο
- **Πληροφορία**: Διαφημιστικό Ταινίας
- **Profile**: Video[6]$--$>Audio[3]$--$>Image$--$>Application

Ο χρήστης ενδιαφέρεται κατά σειρά για πληροφορία video (με όριο 6 min), audio (με όριο 3 min), image και application (χωρίς χρονικό περιορισμό τα δύο τελευταία). Στα πειράματα που κάναμε για το ''σενάριο'' αυτό, το σύστημα επέλεγε στην αρχή την πληροφορία σε μορφή video και στη συνέχεια την μορφή audio όταν αυξήθηκε

η κυκλοφορία (με την εισαγωγή 2 ή 3 συνδέσεων παράλληλα). Η μορφή image θα δινόταν στο χρήστη μόνο αν το throughput του δικτύου έπεφτε κάτω από 6 Kbps, δηλαδή αν είχαμε πολύ υψηλά επίπεδα συμφόρησης.

	εκτιμ. thrpt	εκτιμ. χρόνος			πραγμ. χρόνος			πραγμ. thrpt
		txt	au	mpeg	txt	au	mpeg	
αρχ. συνθ.	77.69	0.01	0.22	**3.29**	--	--	**3.08**	82.85
+1 σύνδεση	54.56	0.02	0.32	**4.69**	--	--	**4.57**	55.93
+2 συνδέσεις	35.87	0.03	**0.48**	7.13	--	**0.53**	*(7.07)*	32.70
+3 συνδέσεις	29.51	0.03	**0.59**	8.66	--	**0.72**	--	24.07

Το χρονικό κόστος που το σύστημά μας εισήγαγε στα πειράματα που κάναμε ήταν το πολύ 1 sec και οφείλεται περισσότερο στο χρόνο για την εκτίμηση του throughput και λιγότερο στον αλγόριθμο επιλογής της μορφής της πληροφορίας που θα σταλεί. Η επιβάρυνση της κυκλοφορίας του δικτύου είναι μόλις 5 Kbytes. Το ποσό αυτό είναι αρκετά μικρό ειδικότερα μάλιστα αν αναλογιστούμε ότι μπορεί να οδηγήσει σε επιλογή μορφής πληροφορίας αρκετά μικρότερου μεγέθους, οπότε ουσιαστικά επιφέρει ελάττωση του φόρτου του δικτύου. Αν υπάρχει ήδη η τιμή του throughput, αν έχει δηλαδή υπολογιστεί σε πρόσφατη προηγούμενη αίτηση, τότε το χρονικό κόστος είναι αμελητέο ενώ η επιβάρυνση της κυκλοφορίας μηδενική.

5.3 Η Απόδοση του WWW με την Χρήση Συνδέσεων Διαρκείας

Στην ενότητα αυτή παρουσιάζουμε τα αποτελέσματα των μετρήσεων που κάναμε για να αξιολογήσουμε την επίπτωση των *συνδέσεων διαρκείας* στην απόδοση του WWW. Οι συγκρίσεις έγιναν μεταξύ clients που έκαναν τις ίδιες αιτήσεις, από τον ίδιο κόμβο στον ίδιο server και με τον ίδιο φόρτο δικτύου και server. Ο ένας client υποστήριζε τις συνδέσεις διαρκείας και ο άλλος όχι. Οι παρακάτω τιμές μετρήθηκαν στη μεριά του client και αντιπροσωπεύουν το χρόνο που απαιτήθηκε για την ικανοποίηση κάθε αίτησης/απάντησης, όχι όμως και την παρουσίαση της μεταφερθείσας πληροφορίας στο χρήστη. Δεν συμπεριλαμβάνεται δηλαδή το κόστος που εισάγει η επικοινωνία με τον X11 server, δεδομένου ότι αυτό δεν έχει σχέση με τις συνδέσεις διαρκείας. Για το σκοπό αυτό η λήψη των χρόνων γίνεται:

- *για τον client με μία σύνδεση ανά αίτηση*, λίγο πριν την έναρξη εγκατάστασης σύνδεσης και αμέσως μετά το κλείσιμό της.
- *για τον client με συνδέσεις διαρκείας*, λίγο πριν την εγκατάσταση σύνδεσης, αν πρόκειται για νέα, ή λίγο πριν την έναρξη αποστολής της αίτησης, αν η σύνδεση είναι ήδη ανοιχτή, και μετά από την λήψη ποσότητας δεδομένων, τόσης όσης λέει η επικεφαλίδα *Content-Length*.

Οι WWW σελίδες που χρησιμοποιήθηκαν για έλεγχο (πίνακας 5.2) προσπαθήσαμε να είναι κατά το δυνατό αντιπροσωπευτικές, σύμφωνα με τα στοιχεία που δώσαμε στην ενότητα 2.5.3, όσον αφορά τόσο το μέγεθός τους (κυμαίνεται από $< 1\text{K}Byte$ έως $> 700Kbytes$), όσο και τον αριθμό των ενσωματωμένων εικόνων σε αυτές (0 έως και 14).

Σελίδα (No)	Αριθ. Ενσωματ. Εικόνων	Συνολικό Μέγεθος (KBytes)
1	3	6.128
2	1	0.995
3	14	19.436
4	3	2.668
5	0	2.815
6	0	3.144
7	0	718.897
8	2	31.011
9	2	9.206
10	0	29.270
11	1	58.788

Πίνακας 5.2: Χαρακτηριστικά WWW σελίδων που χρησιμοποιήθηκαν

Οι μετρήσεις που ακολουθούν έγιναν μεταξύ clients και server που βρίσκονταν σε κόμβους του Ινστιτούτου Πληροφορικής και του Π.Κ αντίστοιχα με χρόνο round trip μεταξύ τους 4-5 ms (πίνακας 5.3). Οι ίδιες ακριβώς μετρήσεις έγιναν μεταξύ clients και server που βρίσκονται σε κόμβους του FORTHnet στη Θεσσαλονίκη και του Π.Κ αντίστοιχα, με χρόνο round trip μεταξύ τους 80-90 ms (πίνακας 5.4). Ο χρόνος round trip που αναφέρουμε αντιστοιχεί σε πακέτα μεγέθους 64 bytes, όσο είναι περίπου το μέγεθος των πακέτων συγχρονισμού που ανταλλάσσονται στη διαδικασία διαπραγμάτευσης τριών φάσεων.

Σε κάθε περίπτωση παρατηρούμε μια σημαντική βελτίωση στους χρόνους μεταφοράς της πληροφορίας σε ποσοστό από 15% μέχρι σχεδόν 60%. Εξαίρεση αποτελεί η ‘‘μεγάλη’’ σελίδα (No 7) στην οποία η βελτίωση είναι μικρότερη, γύρω στο 5%, και εξηγείται από το ότι ο χρόνος που μετράμε οφείλεται κατά το μεγαλύτερο μέρος στη μεταφορά των δεδομένων που είναι σχεδόν ίδιος και στους δύο τύπους συνδέσεων. Αντίθετα στις μικρές σελίδες ή τις σελίδες με αρκετές ενσωματωμένες εικόνες, μεγάλο ποσοστό του χρόνου οφείλεται στην εγκατάσταση και το κλείσιμο των συνδέσεων, οπότε η αποφυγή τους δίνει μεγαλύτερο κέρδος. Αυτό φαίνεται πιο καθαρά στον πίνακα 5.4.

Εξετάζοντας τους δύο τελευταίους πίνακες, παρατηρούμε ακόμα ότι όσο μεγαλύτερος είναι ο χρόνος round trip μεταξύ client και server τόσο μεγαλύτερη είναι και η βελτίωση που παρουσιάζεται.

Σελίδα (No)	Σύνδεση ανά αίτηση (ms)	Συνδέσεις Διαρκείας (ms)	Βελτίωση	
			ms	(%)
1	2584	1824	760	29.41
2	1117	775	342	30.62
3	8797	6295	2502	28.44
4	2398	1688	710	29.61
5	574	406	168	29.27
6	587	445	142	24.19
7	8050	7684	366	4.55
8	2013	1472	541	26.88
9	1907	1328	579	30.36
10	747	633	114	15.26
11	1465	1180	285	19.45

Πίνακας 5.3: Χρόνοι μεταφοράς πληροφορίας με συνδέσεις διαφορετικών τύπων (Ι.Π. - Π.Κ)

Σελίδα (No)	Σύνδεση ανά αίτηση (ms)	Συνδέσεις Διαρκείας (ms)	Βελτίωση	
			ms	(%)
1	6365	2605	3860	59.07
2	2095	1255	840	40.10
3	19433	9894	9539	49.09
4	5290	2412	2878	54.40
5	1480	794	686	46.35
6	1604	819	785	48.94
7	63714	60349	3365	5.28
8	7778	4314	3464	44.54
9	5555	2302	3253	58.56
10	4156	2796	1360	32.72
11	7212	5385	1827	25.33

Πίνακας 5.4: Χρόνοι μεταφοράς πληροφορίας με συνδέσεις διαφορετικών τύπων (FORTHnet - Π.Κ)

Τέλος πρέπει να σημειώσουμε ότι στην δική μας υλοποίηση με τη χρήση συνδέσεων διαρκείας αποφεύγουμε όχι μόνο το κόστος για την εγκατάσταση και κλείσιμο των συνδέσεων αλλά και το κόστος της δημιουργίας (forking) μιας διεργασίας-παιδί για κάθε αίτηση.

Κεφάλαιο 6

Συμπεράσματα - Μελλοντική Εργασία

Από την εργασία αυτή αποκομίσαμε πολλά χρήσιμα συμπεράσματα. Το σύστημα που υλοποιήσαμε αποτελεί το πρωτότυπο, το οποίο μπορεί να εξελιχθεί και να βελτιωθεί, ώστε να χρησιμοποιηθεί σε ευρεία κλίμακα. Τα συμπεράσματα αυτά αναλύονται παρακάτω και στη συνέχεια δίνουμε κάποιες ιδέες για μελλοντική δουλειά.

6.1 Συμπεράσματα

Ο WWW αναμφισβήτητα είναι σήμερα η πιο διαδεδομένη αλλά και πιο γρήγορα εξελισσόμενη εφαρμογή στο χώρο των υπολογιστών. Οι πληροφορίες που γίνονται διαθέσιμες μέσω αυτού αυξάνονται συνεχώς. Το ίδιο και η κυκλοφορία που ο WWW εισάγει στο δίκτυο.

Η ευρεία χρήση του WWW κάνει αναγκαία:

- τη βελτίωση της απόδοσής του με την απάλειψη των μειονεκτημάτων των τμημάτων που τον αποτελούν.
- την προσαρμογή της συμπεριφοράς του σύμφωνα με τις συνθήκες που επικρατούν στο δίκτυο, ώστε να προλαβαίνει ή τουλάχιστον να μην επιδεινώνει καταστάσεις συμφόρησης.

Η υλοποίηση των παραπάνω θα δώσει τη δυνατότητα καλυτέρευσης της ποιότητας των υπηρεσιών που παρέχονται στο χρήστη, με τη μείωση του χρόνου που απαιτείται για την λήψη της πληροφορίας.

Η παρούσα εργασία πραγματεύεται και τα δύο αυτά θέματα. Όσον αφορά το πρώτο, τροποποιήσαμε τη διαδικασία εγκατάστασης συνδέσεων του πρωτοκόλλου του WWW, δηλαδή του HTTP, πάνω από τις οποίες μεταφέρονται τα δεδομένα. Αντικαταστήσαμε την τεχνική ‘‘σύνδεση ανά αίτηση’’ με ’’συνδέσεις διαρκείας’’. Η ανάλυση και οι μετρήσεις που κάναμε με διάφορα μεγέθη σελίδων έδωσαν πολύ καλά αποτελέσματα. Πιστεύουμε λοιπόν ότι οι επόμενες εκδόσεις του HTTP προτύπου

πρέπει να υποστηρίζουν κάποιο χαρακτηριστικό που να εξασφαλίζει τις βελτιώσεις που επιτεύχθηκαν στην παρούσα εργασία.

Όσον αφορά την προσαρμογή στις συνθήκες που επικρατούν στο δίκτυο κάναμε μια πρώτη προσέγγιση με την συνεργασία του διαχειριστή της WWW πληροφορίας και μιας διεργασίας που εκτιμά παραμέτρους του δικτύου στα ζητούμενα μονοπάτια. Η γνώση της κατάστασης του δικτύου που παρέχεται, χρησιμοποιείται για την επιλογή της πιο ‘‘κατάλληλης’’ μεταξύ διαφορετικών μορφών της ίδιας πληροφορίας. Η επιλογή αυτή δεν μπορεί να είναι αυθαίρετη αλλά συνάρτηση των προτιμήσεων του χρήστη. Γι’ αυτό το σκοπό ορίσαμε την δομή των profiles και επεκτείναμε το πρωτόκολλο HTTP ώστε να είναι εφικτή η αυτόματη γνωστοποίησή του στον διαχειριστή της πληροφορίας. Από τα πειράματα που κάναμε με ποικίλα σενάρια λειτουργίας του συστήματος φάνηκε ότι ανταποκρίνεται σε αρκετά ικανοποιητικό βαθμό στις απαιτήσεις και τους περιορισμούς που θέτει ο χρήστης. Η δυναμική αναπροσαρμογή της συμπεριφοράς του μπορεί να συμβάλλει στην μείωση του φορτίου στο δίκτυο και σίγουρα πλεονεκτεί σε σχέση με το τωρινό στατικό μοντέλο. Πιστεύουμε πάντως ότι μπορεί να γίνει ακόμα πολλή εργασία στον τομέα της συνεργασίας του WWW με το δίκτυο. Μερικά σημεία αναφέρουμε στην επόμενη ενότητα.

6.2 Μελλοντική Εργασία

Το σύστημα που υλοποιήσαμε έδωσε ενθαρρυντικά αποτελέσματα από τη χρησιμοποίηση του στο τμήμα του δικτύου όπου είχαμε πρόσβαση και από τα πειράματα που διεξάγαμε σε αυτό. Έδειξε ότι μπορεί να βελτιώσει τις υπηρεσίες που παρέχονται στο χρήστη και έμμεσα να μειώσει την συμφόρηση του δικτύου. Ενδιαφέροντα θέματα τα οποία μπορούν να αποτελέσουν αντικείμενο έρευνας στο άμεσο μέλλον είναι:

- Εγκατάσταση και έλεγχος του προτεινόμενου συστήματος σε ευρύτερη κλίμακα. Συγκριτική μελέτη της απόδοσης του και αξιολόγηση των αποφάσεων που λαμβάνει ως προς το είδος της πληροφορίας που πρέπει να στείλει, σύμφωνα με τις συνθήκες που επικρατούν στο δίκτυο.

- Βελτίωση της μεθόδου εκτίμησης της κατάστασης του δικτύου με πιο ακριβείς υπολογισμούς και περισσότερες σχετικές παραμέτρους.

- Συνεργασία του συστήματός μας με συστήματα ελέγχου ροής των εφαρμογών, τα οποία έχουν τη δυνατότητα μορφοποίησης της κυκλοφορίας των συνδέσεων TCP, πετυχαίνοντας την επιθυμητή ποιότητα των παρεχόμενων υπηρεσιών.

- Μελέτη των δυνατοτήτων που μπορούν να δώσουν στο σύστημά μας νέα, υπο-εξέλιξη, πρωτόκολλα στο Internet, όπως η έκδοση 6 του IP [DH95], ή νέες τεχνολογίες όπως ο Ασύγχρονος Τρόπος Μετάδοσης (ATM). Και τα δύο παραπάνω, το καθένα με διαφορετικό τρόπο, μπορούν να εγγυηθούν μια συγκεκριμένη ποιότητα υπηρεσίας. Πρέπει λοιπόν να εξεταστεί πως μπορεί να γίνει η διαπραγμάτευση με το δίκτυο και η αντιστοίχιση των προτιμήσεων

του χρήστη στους διαθέσιμους πόρους του, με στόχο πάντα την μεγιστοποίηση της ικανοποίησης των χρηστών αλλά και της απόδοσης του δικτύου. Σε τέτοιες περιπτώσεις, τα profiles των χρηστών μπορούν να επεκταθούν και να εισαχθούν σε αυτά και στοιχεία χρέωσης.

- Χρήση της μεθόδου εκτίμησης της κατάστασης του δικτύου που αναπτύχθηκε για τον εντοπισμό των χρονικών διαστημάτων μικρότερη χρήσης του. Η γνώση αυτή είναι κρίσιμη σε μεθόδους μαζικής μεταφοράς δεδομένων, οι οποίες θέλουμε να μην έχουν σημαντική επίδραση στους διαλογικούς χρήστες του δικτύου. Τέτοιες μέθοδοι είναι το mirroring, η μεταφορά news, το prefetching κτλ.

Παράρτημα Α

Θέματα Σχεδίασης και Υλοποίησης

Στην ενότητα αυτή γίνεται αναφορά σε ορισμένα θέματα σχεδιασμού που μας απασχόλησαν κατά την υλοποίηση του συστήματος. Πιο συγκεκριμένα στην ενότητα Α.1 περιγράφουμε τη γεννήτρια κυκλοφορίας οι μετρήσεις της οποίας χρησιμοποιήθηκαν για την σύγκριση με την μέθοδο εκτίμησης που αναπτύξαμε. Στην ενότητα Α.2 αναλύεται ο τρόπος αποθήκευσης του υπολογιζόμενου throughput στην δομή καταχώρησης ώστε να πετύχουμε καλύτερη απόδοση.

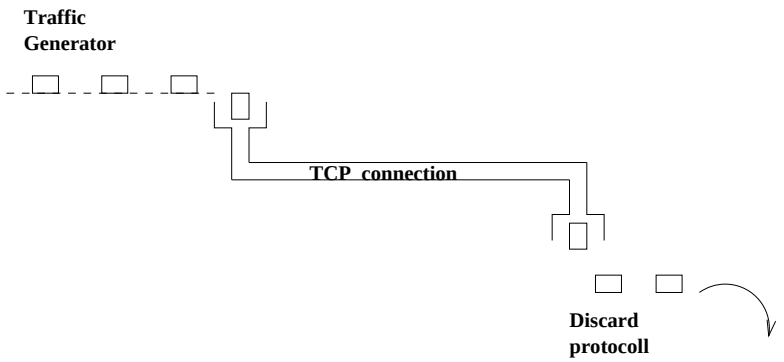
Α.1 Η Γεννήτρια Κυκλοφορίας

Για την επαλήθευση της ορθότητας της εκτίμησης του throughput που μας παρέχει η χρήση της προτεινόμενης μεθόδου, αναπτύξαμε μια γεννήτρια η οποία εισάγει πραγματική κυκλοφορία στο δίκτυο και μετρήσαμε τα αποτελέσματα που παίρνουμε από αυτήν.

Το πρόβλημα στην συγκεκριμένη περίπτωση είναι ότι θέλουμε να κάνουμε μετρήσεις με μακρινούς κόμβους (π.χ Ευρώπη, Η.Π.Α) στους οποίους προφανώς δεν έχουμε πρόσβαση για να εγκαταστήσουμε τα προγράμματα που κάνουν τους υπολογισμούς. Για το λόγο αυτό χρησιμοποιούμε το πρωτόκολλο DISCARD [Pos83], το οποίο βρίσκεται πάνω από το TCP (port 9) και η μόνη του λειτουργία (όπως λέει και το όνομα του) είναι να ‘‘πετάει’’ τα δεδομένα που κατευθύνονται σε αυτό.

Για να γίνουν οι μετρήσεις λοιπόν ανοίγουμε μια TCP σύνδεση μεταξύ της γεννήτριας κυκλοφορίας και του πρωτοκόλλου αυτού στον κόμβο προορισμού (σχήμα Α.1).

Το ποσό της κυκλοφορίας που εισάγουμε, επιλέγουμε να είναι αρκετά μεγάλο ώστε το throughput που παίρνει η συγκεκριμένη σύνδεση να είναι το μέγιστο δυνατό. Παράλληλα επιτυγχάνουμε η σύνδεση να είναι λιγότερο επιρρεπής σε διακυμάνσεις που οφείλονται στην κυκλοφορία άλλων συνδέσεων και στους χρόνους επεξεργασίας από τους τελικούς σταθμούς και τα ενδιάμεσα συστήματα. Ουσιαστικά δηλαδή επιδιώκουμε να δεσμεύσουμε όλο το διαθέσιμο bandwidth του περιοριστικού



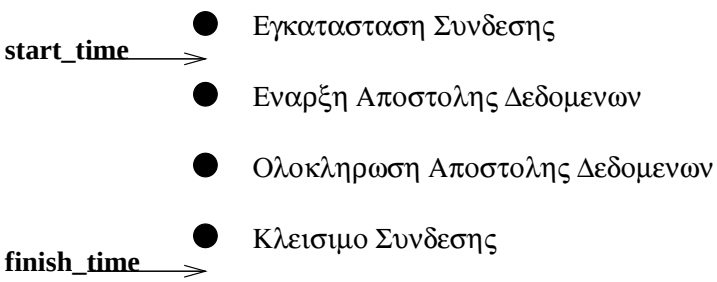
Σχήμα Α.1: Σύνδεση μεταφοράς πραγματικής κυκλοφορίας

συνδέσμου κατά μήκος της διαδρομής της σύνδεσης.

Η παρακάτω σχέση υπολογίζει το throughput που λαμβάνει μια τέτοια σύνδεση.

$$throughput = TrafficInjected/(FinishTime - StartTime) \qquad (A.1)$$

όπου *TrafficInjected* είναι το ποσό των δεδομένων που εισάγουμε και *StartTime*, *FinishTime* οι χρονικές στιγμές που αντιστοιχούν στα σημεία που φαίνονται στο σχήμα Α.2.



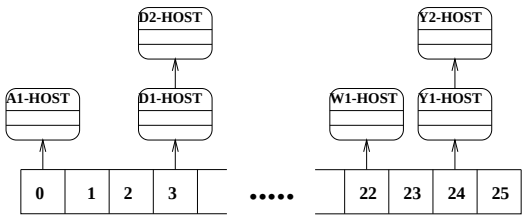
Σχήμα Α.2: Σημεία λήψεως χρόνου

Η πρώτη χρονική στιγμή λαμβάνεται λίγο πριν την έναρξη αποστολής δεδομένων, αποφεύγοντας έτσι το χρόνο για την εγκατάσταση σύνδεσης, ο οποίος μπορεί να επηρεάσει τα αποτελέσματα, ειδικά όταν χρησιμοποιούνται γρήγοροι σύνδεσμοι επικοινωνίας. Η δεύτερη χρονική στιγμή λαμβάνεται, αμέσως μετά το κλείσιμο της σύνδεσης. Ο λόγος που γίνεται αυτό είναι επειδή είμαστε αναγκασμένοι να κάνουμε τις μετρήσεις από την μεριά της πηγής. Περιμένοντας λοιπόν το κλείσιμο της σύνδεσης εξασφαλίζουμε (ο μηχανισμός παραθύρου και των μηνυμάτων επιβεβαίωσης του TCP το επιβάλλει) ότι τα δεδομένα που έχουμε στείλει έχουν πράγματι φτάσει στον παραλήπτη και δεν βρίσκονται ακόμα κάπου πάνω στο δίκτυο δηλαδή στους ενταμιευτές κάποιων ενδιάμεσων συστημάτων.

A.2 Η Δομή Καταχώρησης Throughput

Η αποθήκευση της εκτίμησης του throughput είναι ιδιαίτερα χρήσιμη γιατί μπορεί να αξιοποιηθεί: **α)** από επόμενες αιτήσεις που έρχονται από τον ίδιο client (δηλ. ίδιο κόμβο) σε σύντομο χρονικό διάστημα και **β)** από τις αιτήσεις *όλων* των clients (δηλ. διαφορετικοί κόμβοι) που βρίσκονται πίσω από ένα proxy server, δεδομένου ότι στην προκειμένη περίπτωση υπολογίζεται το throughput μεταξύ του δικού μας και του proxy server.

Η διεργασία εκτίμησης κατάστασης δικτύου διατηρεί και ενημερώνει την δομή δεδομένων που κρατούνται τα ονόματα των κόμβων και οι τιμές του υπολογιζόμενου throughput που τους αντιστοιχεί. Στόχος είναι να ελαχιστοποιήσουμε το χρόνο απόκρισης του ESTd προς τον HTTPd. Η παραπάνω πληροφορία λοιπόν οργανώνεται σε ένα hash-table, 26 θέσεων, και η αναφορά γίνεται με βάση τον πρώτο χαρακτήρα του ονόματος του κόμβου (σχήμα A.3).



Σχήμα A.3: Η δομή αποθήκευσης της πληροφορίας κατάστασης δικτύου

Μόνο αυτό όμως δεν αρκεί, γιατί οι περισσότερες αιτήσεις σε ένα server έρχονται από άλλους servers που λειτουργούν ως proxy και που τα ονόματα τους συνήθως αρχίζουν από ‘‘w’’. Δημιουργείται έτσι μια ανισοκατανομή όσον αφορά τις θέσεις του πίνακα και αδρανοποιούνται τα πλεονεκτήματα του hash-table, αφού τα περισσότερα στοιχεία μπαίνουν στην λίστα της ίδιας θέσης του πίνακα. Για το λόγο αυτό, με τη βοήθεια του DNS συστήματος [Moc87], βρίσκουμε και κρατάμε το επίσημο όνομα (official name) που αντιστοιχεί στον κόμβο που κάνει την αίτηση. Έτσι επιτυγχάνουμε πολύ καλύτερη κατανομή των πληροφοριών στις θέσεις του πίνακα και επομένως μειώνουμε το χρόνο αναφοράς σε αυτές. Παράλληλα με τη χρήση του επίσημου ονόματος αποφεύγουμε την κράτηση πολλαπλών θέσεων για τον ίδιο κόμβο και το επιπλέον κόστος υπολογισμού στοιχείων που μπορεί ήδη να υπάρχουν.

Ενδεικτικά παραθέτουμε παρακάτω το χρησιμοποιούμενο και το επίσημο όνομα τεσσάρων μηχανημάτων του πανεπιστημίου Κρήτης που ‘‘φιλοξενούν’’ ισάριθμους WWW servers.

Χρησιμοποιούμενο Όνομα	Επίσημο Όνομα
www.csd.uch.gr	crete.csd.uch.gr
www.cc.uch.gr	kerynia.cc.uch.gr
www.edu.uch.gr	artemis.edu.uch.gr
www.physics.uch.gr	selas.physics.uch.gr

Προκύπτει λοιπόν ότι αν έλθουν αιτήσεις από αυτούς, οι πληροφορίες που σχετίζονται με τους τέσσερις κόμβους, σύμφωνα με τον τρόπο που αναλύσαμε, θα μπούν στις λίστες τεσσάρων διαφορετικών θέσεων του πίνακα αντί της ίδιας που θα έμπαιναν αλλιώς.

Παράρτημα Β

Σύγκριση Πραγματικού και Εκτιμούμενου Throughput Δικτύου

Στην ενότητα αυτή παραθέτουμε τις μετρήσεις που κάναμε για να αξιολογήσουμε την ορθότητα της μεθόδου εκτίμησης που υλοποιήσαμε και επομένως δυνατότητα χρησιμοποίησης του συστήματος σε ευρύτερη κλίμακα. Η σύγκριση γίνεται με τα αποτελέσματα που μας παρέχει η γεννήτρια κυκλοφορίας του παραρτήματος Α. Οι κόμβοι με τους οποίους έγιναν οι μετρήσεις βρίσκονται στη Θεσσαλονίκη (Αριστοτέλειο Παν.), στη Γαλλία (Ebone), στη Φινλανδία (Funet), στη Μ. Βρετανία(Pipex), στις ΗΠΑ (Primenet) και στην Αυστραλία (Adelaide Univ.), ενώ πηγή των πακέτων ήταν πάντα ένας κόμβος του Π.Κ. (Τμήμα Επιστήμης Υπολογιστών).

Σε κάθε περίπτωση δίνουμε τέσσερις γραφικές παραστάσεις που αντιστοιχούν:

1. Στο throughput που πήρε μια TCP σύνδεση μεταξύ των δύο αναφερόμενων κόμβων, όπως αυτό μετρήθηκε από την γεννήτρια κυκλοφορίας. Το ποσό των δεδομένων που μετέφερε ήταν της τάξης του ενός MByte (στην Αυστραλία ήταν 512 KBytes), ώστε να δεσμεύει όλο το διαθέσιμο bandwidth του περιοριστικού συνδέσμου του μονοπατιού.

2. Στο throughput που υπολογίζει η δική μας μέθοδος, στέλνοντας 5 ICMP πακέτα μεγέθους 1000 bytes το καθένα.

3. Στο μέσο χρόνο round-trip των ICMP πακέτων κάθε προσπάθειας.

4. Στο ποσοστό απωλειών πακέτων που καταγράφηκε σε κάθε προσπάθεια.

Τα δείγματα παίρνονταν ανά τακτά χρονικά διαστήματα και η διάρκεια των μετρήσεων κυμαίνεται από 1.5 έως 7 μέρες.

Στον πίνακα Β.1 παρουσιάζουμε συγκεντρωτικά τα στοιχεία που συλλέξαμε (μέσο throughput πραγματικής σύνδεσης και μεθόδου εκτίμησης) και τα αποτελέσματα της σύγκρισής τους (μέσο απόλυτο σφάλμα, διασπορά).

Κόμβος	Μέσο throughput (Kbps)		Μέσο σφάλμα		σ
	TCP σύνδεση	Μέθ. Εκτίμησης	(Kbps)	(%)	
FORTHnet (Θεσσαλονίκη)	80.20	75.04	7.55	9.41	4.71
Αριστοτέλειο (Θεσσαλονίκη)	2.62	1.67			
Ebone (Γαλλία)	55.91	45.53	9.22	16.49	5.05
Pipex (Μ. Βρετανία)	19.53	14.81	5.86	30	5.13
Funet (Φινλανδία)	29.30	22.5	8.12	27.71	4.63
Primenet (ΗΠΑ)	20.85	16.38	5.72	27.40	3.92
Adelaide Univ. (Αυστραλία)	10.81	10.75	2.39	22.7	2.01

Πίνακας Β.1: Συγκεντρωτικά αποτελέσματα μετρήσεων

Τα ποσοστά του σφάλματος στον πίνακα ίσως φαίνονται μεγάλα, αλλά θα πρέπει να προσέξουμε τους απόλυτους αριθμούς βάσει των οποίων βγαίνουν. Οι αριθμοί αυτοί είναι μικροί και επομένως ακόμα και μικρές διαφορές μπορεί να μας δώσουν μεγάλα ποσοστά σφάλματος. Για παράδειγμα αν παίρναμε το σφάλμα των μετρήσεων με το Αριστοτέλειο πανεπιστήμιο, αυτό θα ήταν της τάξης του 60%. Στην πραγματικότητα όμως η εκτίμησή μας είναι πάρα πολύ καλή, λαμβάνοντας μάλιστα υπόψη ότι έγινε με τη αποστολή μόνο 5 ICMP πακέτων. Εξάλλου πρέπει να τονίσουμε ότι για τη λήψη της μιας ή της άλλης απόφασης από το σύστημα είναι αρκετή της περισσότερες φορές η εκτίμηση της ‘‘τάξης’’ του throughput του μονοπατιού και όχι η ακριβής τιμή του.

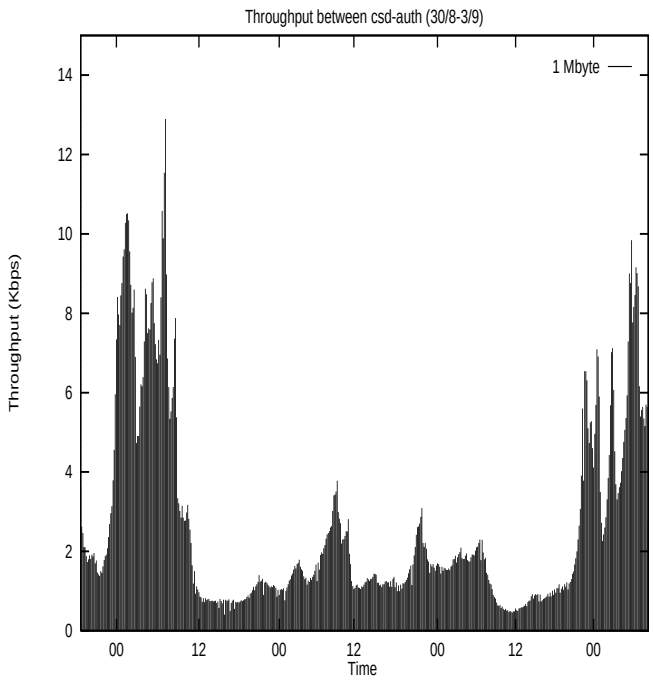
Στην περίπτωση των μετρήσεων με το Αριστοτέλειο πανεπιστήμιο, λόγω μεγάλης συμφόρησης του περιοριστικού συνδέσμου του μονοπατιού, το throughput που έπαιρνε κάθε σύνδεση ήταν πολύ μικρό ($<< 10Kbps$) με αποτέλεσμα να χρειάζεται πολλή ώρα για να ολοκληρωθεί. Για να κάνουμε λοιπόν καλύτερη σύγκριση με τη μέθοδό μας, παίρναμε περισσότερα δείγματα κατά την διάρκεια της σύνδεσης. Έτσι δεν μπορούμε να κάνουμε αντιστοίχιση των μετρήσεων των δύο μεθόδων για να βρούμε το σφάλμα, γι’ αυτό και δίνουμε μόνο το μέσο throughput κάθε μίας . Παρατηρούμε όμως από τις γραφικές παραστάσεις ότι η προσέγγιση είναι αρκετά καλή.

Αξίζει ακόμα να σημειώσουμε πόσο διαφορετική δικτυακή κατάσταση μπορεί να συναντήσει κανείς, ακόμα και μεταξύ σημείων που φυσικά βρίσκονται πολύ κοντά. Αυτό φαίνεται χαρακτηριστικά στον πίνακα, συγκρίνοντας τα στοιχεία που αντιστοιχούν στους κόμβους του FORTHnet και του Αριστοτελείου πανεπιστημίου στη Θεσσαλονίκη (μέσο πραγματικό throughput 80.20 και 2.62 αντίστοιχα).

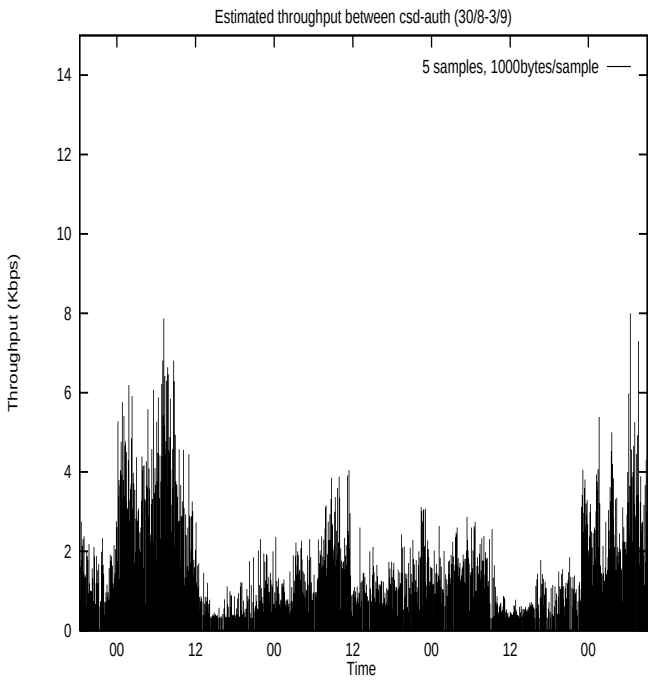
Όπως αναφέραμε στην ενότητα 3.1 ένα από τα προβλήματα που έχουμε να αντιμετωπίσουμε για την εκτίμηση της κατάστασης του δικτύου είναι το διαφορετικό ποσοστό χρησιμοποίησης του bandwidth των δύο κατευθύνσεων ενός συνδέσμου επικοινωνίας. Στις δικές μας μετρήσεις και ειδικά εκείνες για το Ebone και το Funet, το περιοριστικό σημείο είναι ο σύνδεσμος Ελλάδας-Γαλλίας, ο οποίος έχει αρκετά λιγότερη κυκλοφορία στην κατεύθυνση προς το εξωτερικό. Η ποσότητα

των δεδομένων που εισάγει η γεννήτρια κυκλοφορίας στο δίκτυο, στις περιπτώσεις αυτές, έχει φορά προς το εξωτερικό. Επομένως συναντάει λιγότερη ανταγωνιστική κυκλοφορία και υφίσταται μικρότερη καθυστέρηση στην ουρά εξυπηρέτησης του περιοριστικού συνδέσμου. Αντίθετα η καθυστέρηση που υφίστανται τα πακέτα της δικής μας μεθόδου, είναι το άθροισμα και το δύο κατευθύνσεων. Πιστεύουμε λοιπόν ότι αυτό επηρεάζει το αποτέλεσμα και προκαλεί μέρος του σφάλματος που δίνουμε στον πίνακα Β.1.

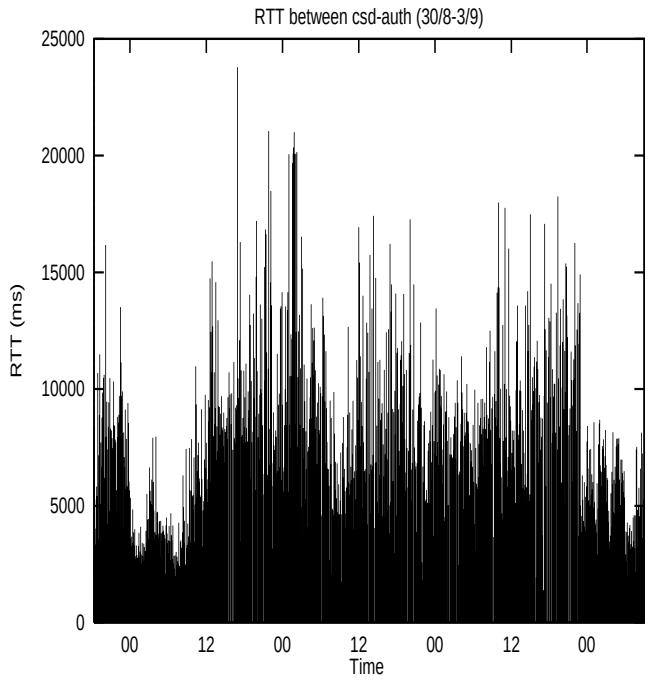
Όσον αφορά τις μετρήσεις με τους άλλους κόμβους του εξωτερικού δεν είμαστε σε θέση να γνωρίζουμε ποιο ή ποια είναι τα περιοριστικά σημεία και ποια κατεύθυνση έχει την μεγαλύτερη συμφόρηση. Μπορούμε μόνο να αναφέρουμε ότι η Pipex, το Primenet και Univ. Adelaide βρίσκονται αντίστοιχα 7, 9 και 17 βήματα δικτύου (hops) μακριά.



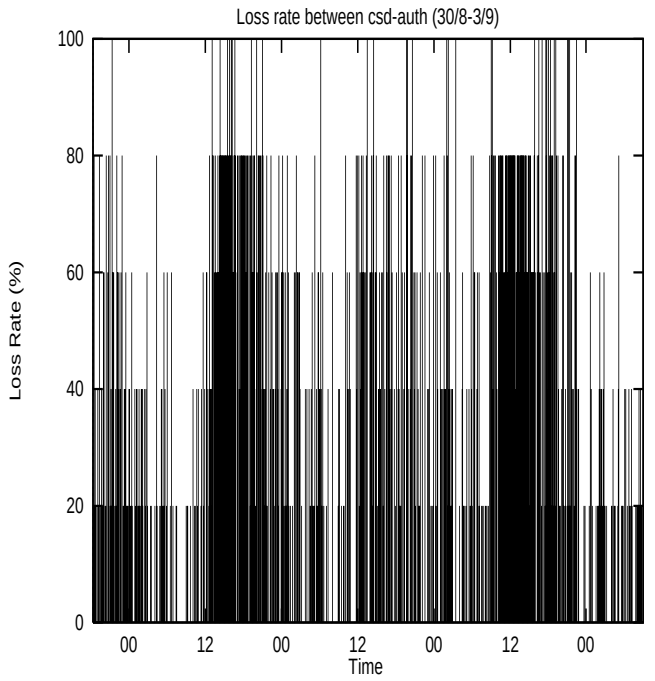
(a) Πραγματικό throughput



(b) Εκτιμούμενο throughput

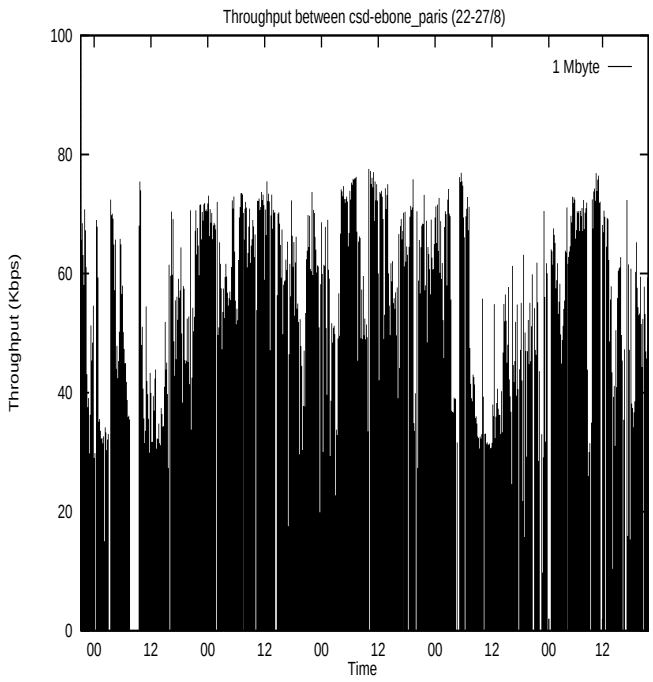


(c) Χρόνος round trip

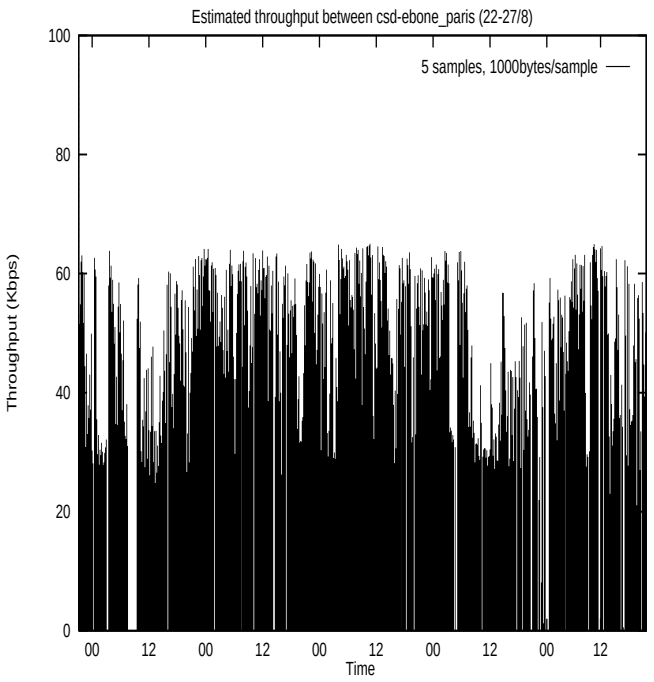


(d) Ποσοστό απωλειών πακέτων

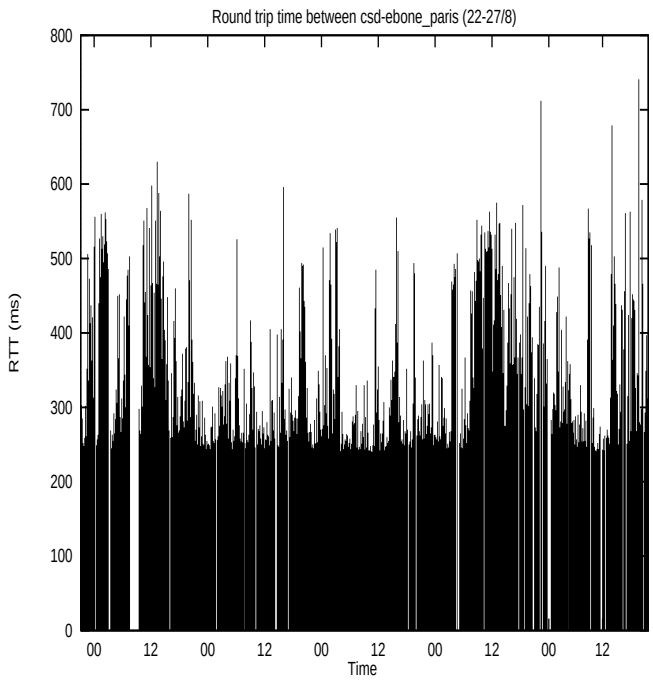
Σχήμα Β.1: Μετρήσεις μεταξύ Ηράκλειου-Θεσσαλονίκης



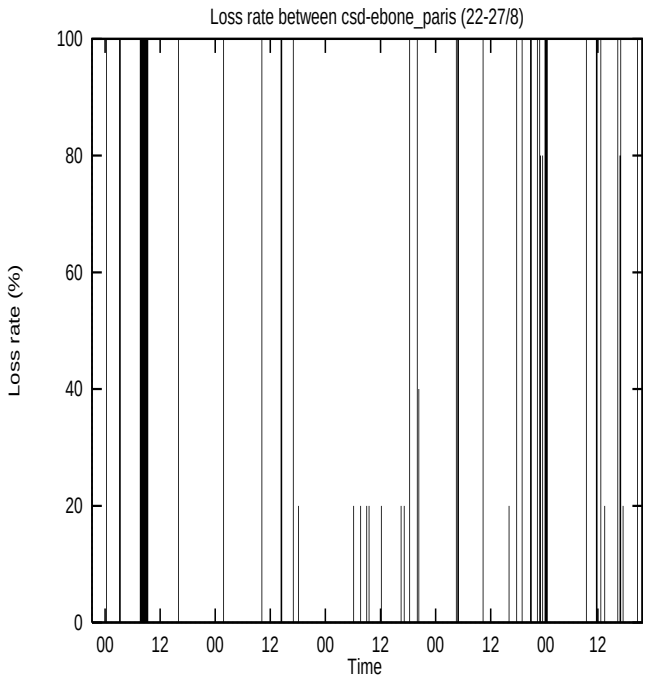
(a) Πραγματικό throughput



(b) Εκτιμούμενο throughput

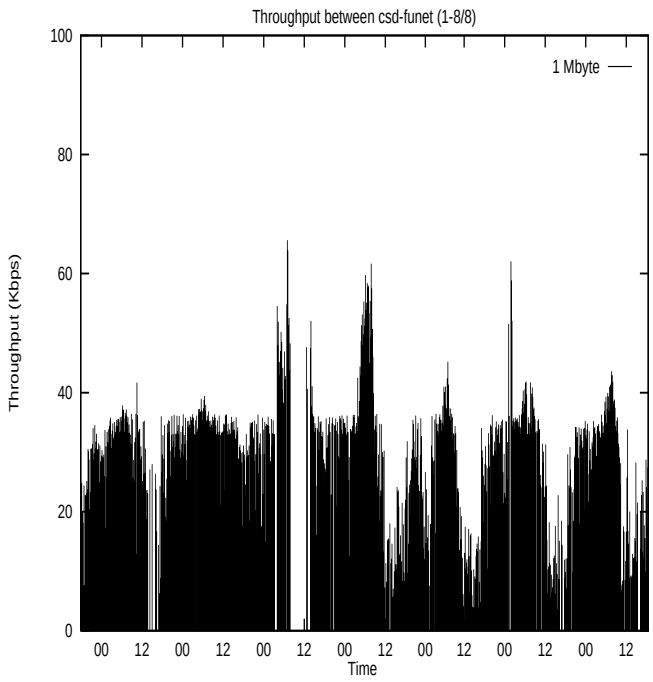


(c) Χρόνος round trip

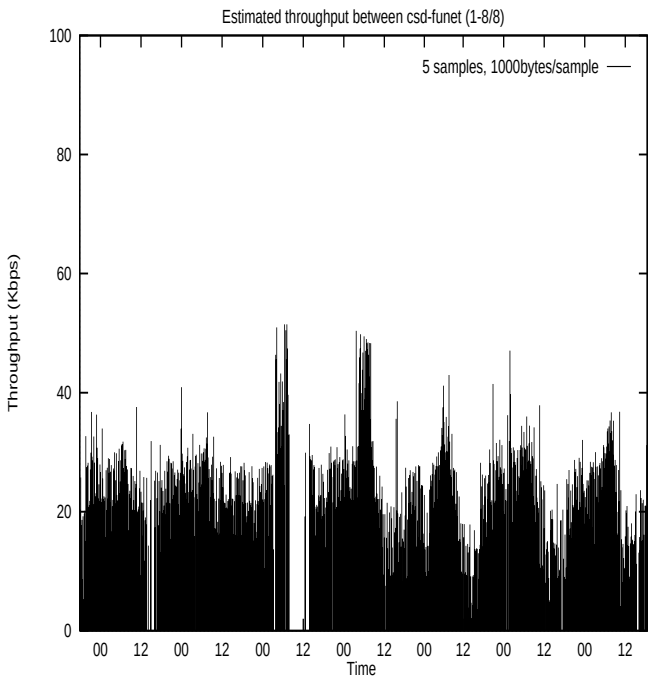


(d) Ποσοστό απωλειών πακέτων

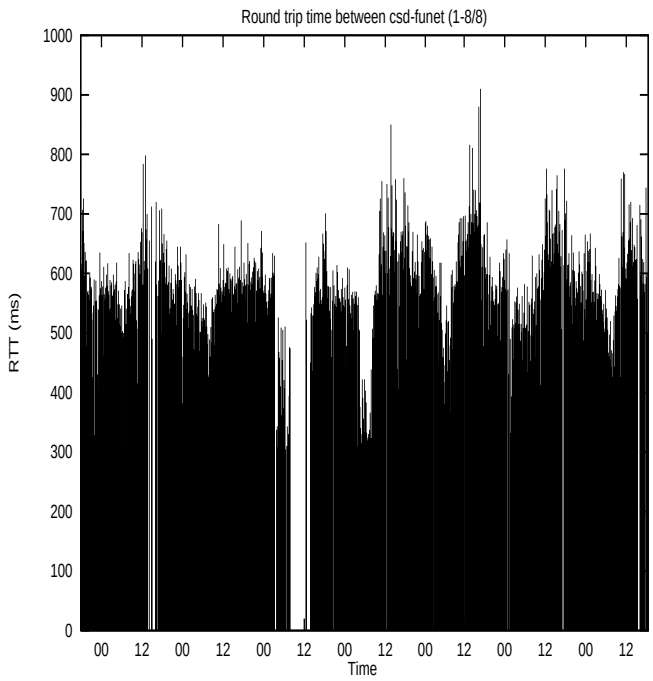
Σχήμα Β.2: Μετρήσεις μεταξύ Ηρακλείου-Γαλλίας



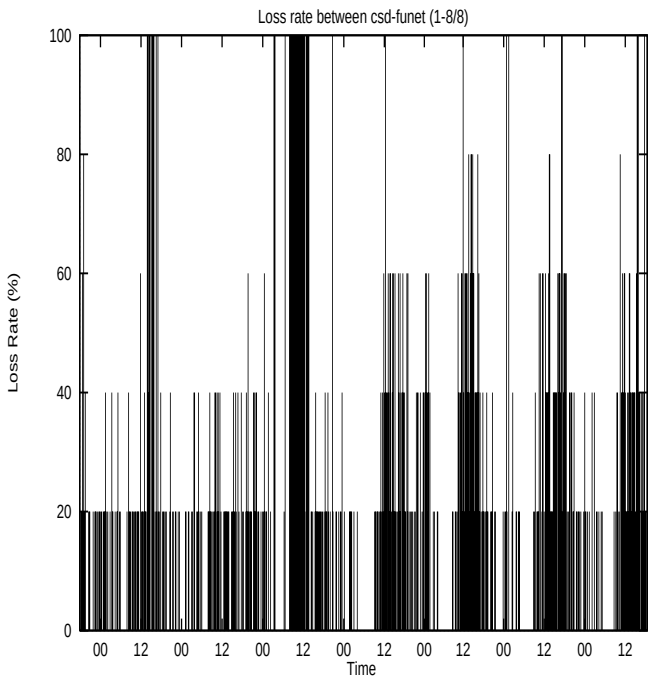
(a) Πραγματικό throughput



(b) Εκτιμούμενο throughput

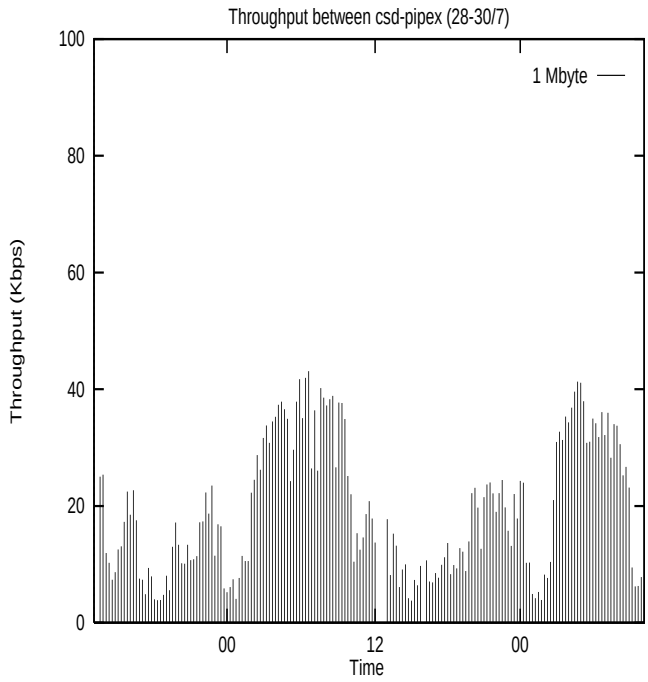


(c) Χρόνος round trip

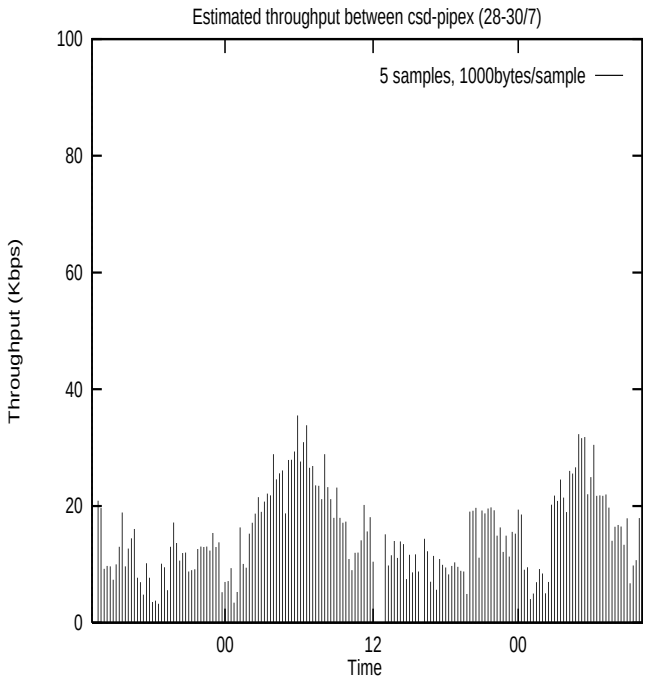


(d) Ποσοστό απωλειών πακέτων

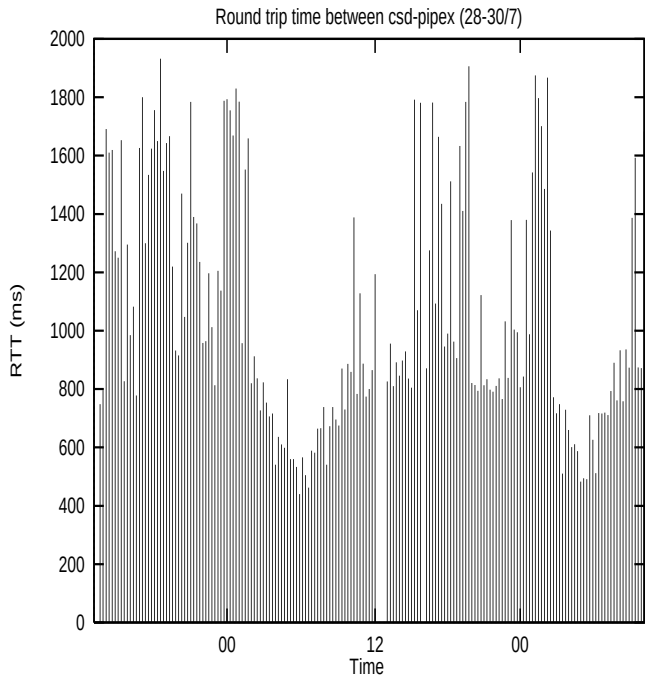
Σχήμα Β.3: Μετρήσεις μεταξύ Ηράκλειου-Φινλανδίας



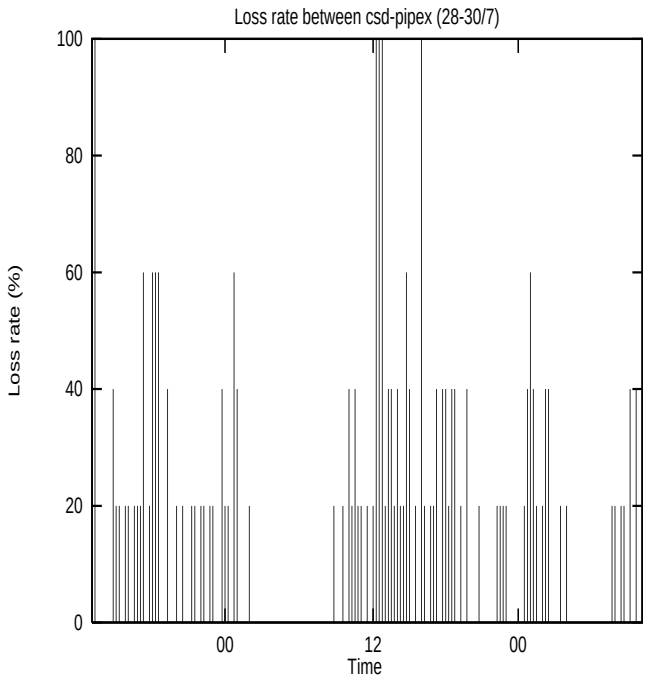
(a) Πραγματικό throughput



(b) Εκτιμούμενο throughput

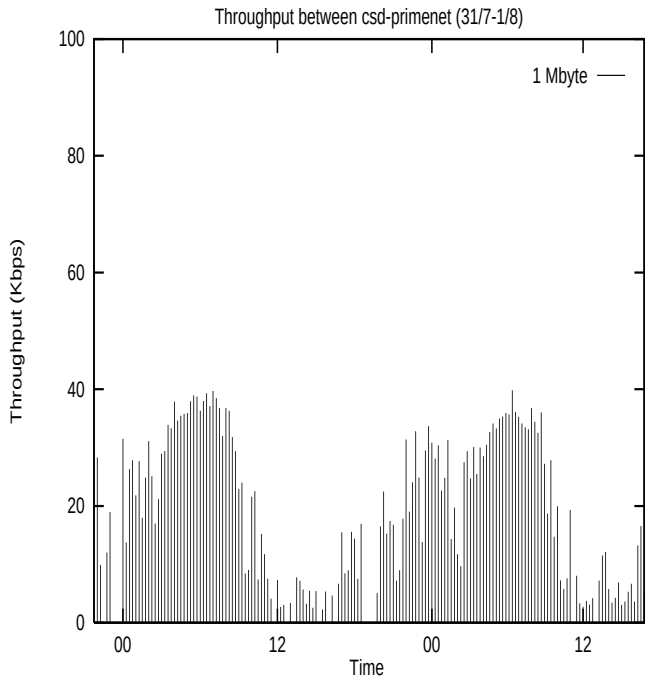


(c) Χρόνος round trip

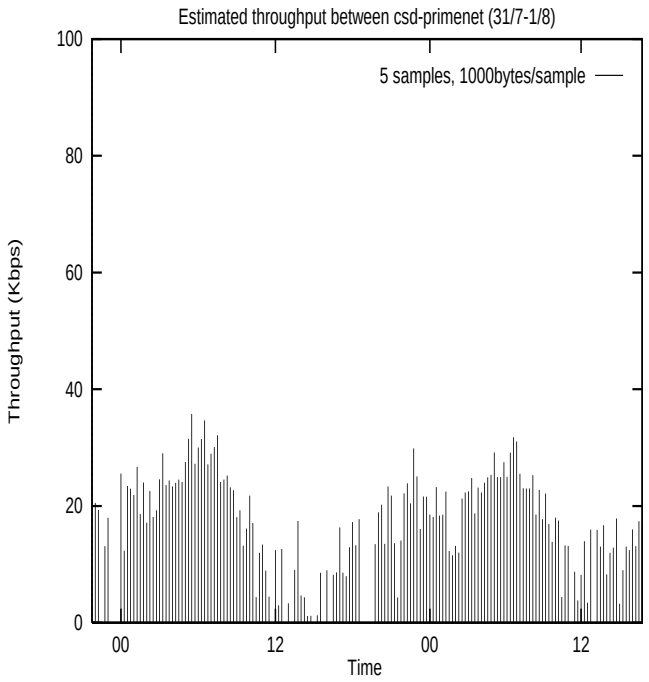


(d) Ποσοστό απωλειών πακέτων

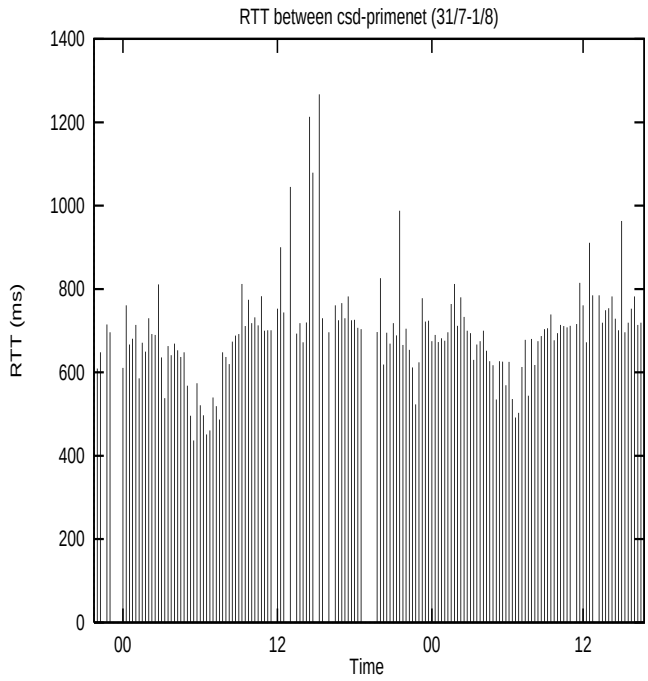
Σχήμα Β.4: Μετρήσεις μεταξύ Ηρακλείου-Μ. Βρετανίας



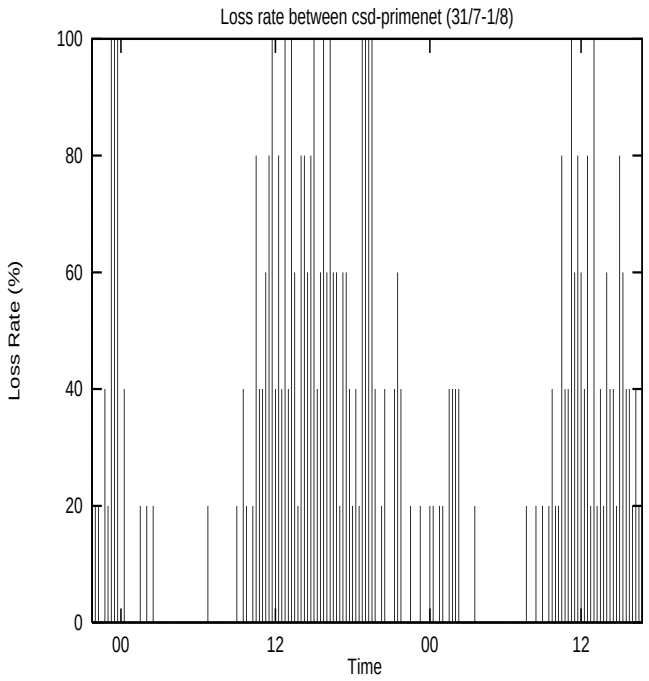
(a) Πραγματικό throughput



(b) Εκτιμούμενο throughput

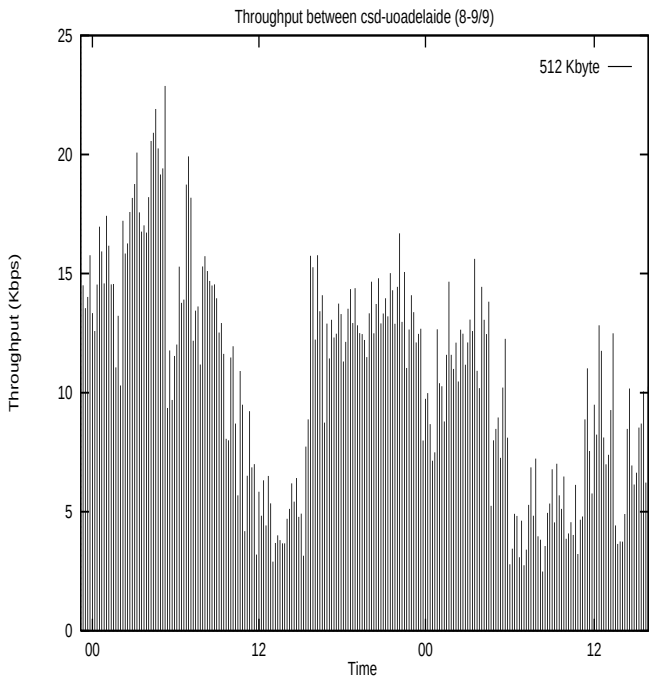


(c) Χρόνος round trip

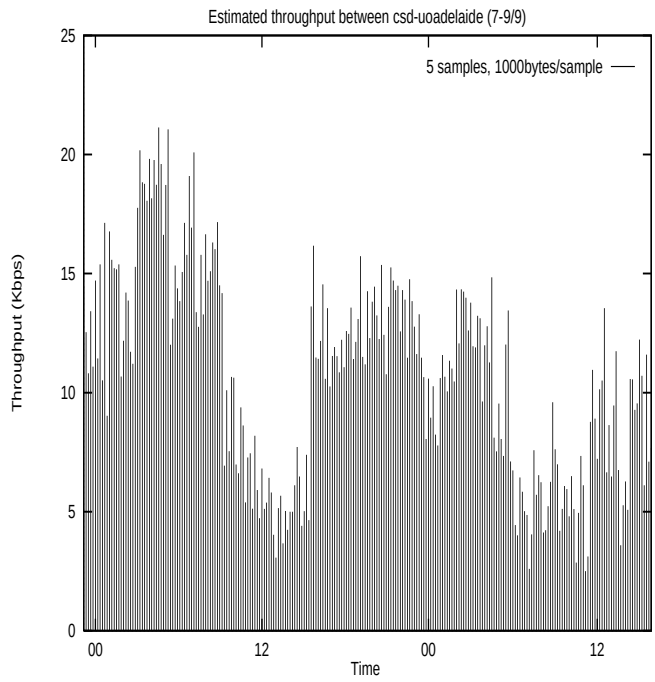


(d) Ποσοστό απωλειών πακέτων

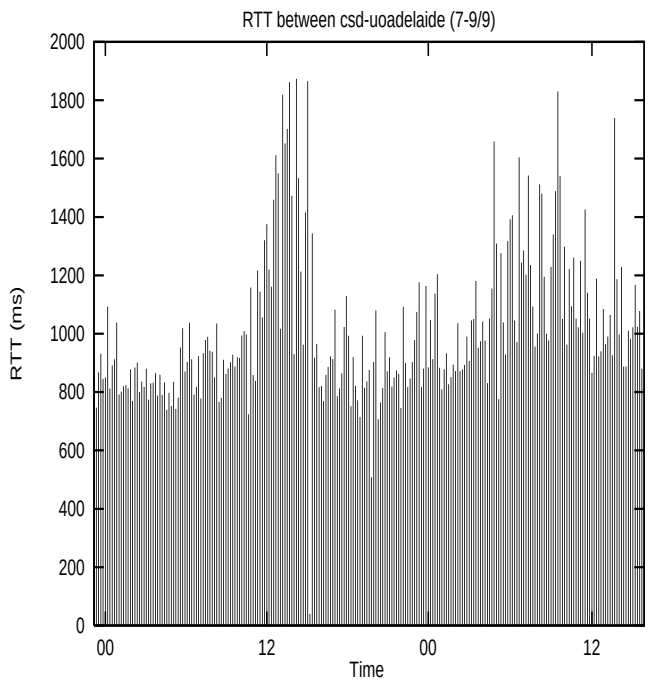
Σχήμα Β.5: Μετρήσεις μεταξύ Ηρακλείου-ΗΠΑ



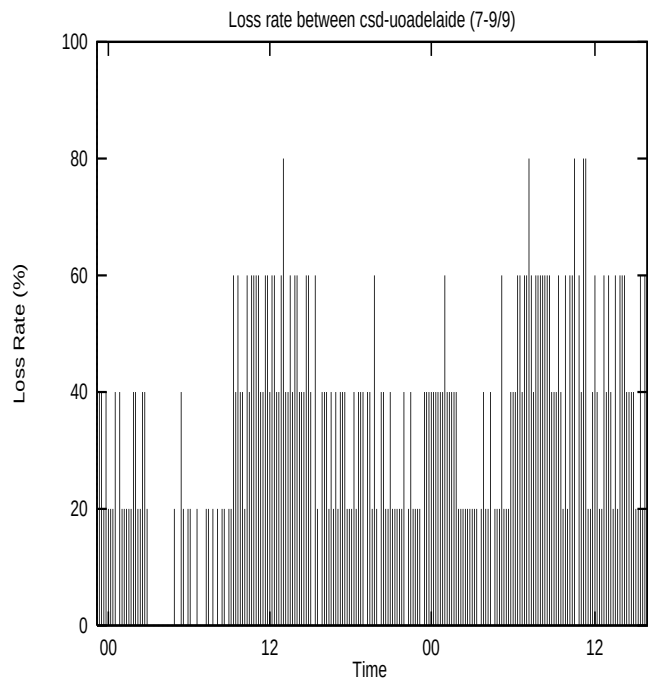
(a) Πραγματικό throughput



(b) Εκτιμούμενο throughput



(c) Χρόνος round trip



(d) Ποσοστό απωλειών πακέτων

Σχήμα Β.6: Μετρήσεις μεταξύ Ηρακλείου-Αυστραλίας

Ευρετήριο Όρων

Αίτηση	Request
Αλγόριθμος Αργής Εκκίνησης	Slow Start Algorithm
Αναζήτηση	Search
Ανάκληση	Retrieval
Αναμετάδοση	Retransmission
Ανενεργός-Αδρανής	Idle
Αντήχηση-Ηχώ	Echo
Αντικείμενο	Object
Αντιπρόσωπος Χρήστη	User Agent (UA)
Αξιόπιστος	Reliable
Αποδοτικότητα	Efficiency
Απόκριση - Απάντηση	Response
Αποστολέας	Sender
Απώλεια	Loss
Ασύγχρονος Τρόπος Μετάδοσης	Asynchronous Tranfer Mode (ATM)
Βήμα Δικτύου	Hop
Δέκτης - Παραλήπτης	Receiver
Δέσμευση Πόρων	Resource Reservation
Διαδρομή	Route
Διαμόρφωση	Configuration
Διαπραγμάτευση	Negotiation
Διαπραγμάτευση Τριών Φάσεων	Three Way Handshake
Διαφάνεια	Transparency
Διαχειριστής	Manager
Διαχειριστές - Αντιπρόσωποι	Proxy Servers
Διέλευση	Throughput
Δίκτυο Μείζονος (Ευρείας) Περιοχής	Wide Area Network (WAN)
Δοσοληψία	Transaction
Δρομολογητής	Router
Εγγυημένη Υπηρεσία	Guaranteed Service
Εκτιμητής	Estimator
Έλεγχος Κυκλοφορίας	Traffic Control
Έλεγχος Συμφόρησης	Congestion Control
Ενταμιευτής	Buffer
Εξακρίβωση Ταυτότητας	Authentication
Εξισορρόπηση Κυκλοφορίας	Traffic Balancing
Εξουσιοδότηση	Authorization
Επιβεβαίωση	Acknowledgement
Επικεφαλίδα	Header
Επίπεδο Εφαρμογών	Application Layer
Επιπλέον Φόρτος	Overhead
Ετικέτα	Tag
Ευέλικτος	Flexible

Εύρος Ζώνης	Bandwidth
Ευσταθής	Robust
Εφαρμογή	Application
Θύρα Επικοινωνίας	Port
Καθυστέρηση	Latency
Κερματισμός-Κατάτμηση	Fragmentation
Κλειδί	Keyword
Κοινή Χρήση Πόρων	Resource Sharing
Κόμβος	Host
Κυρίως Μέρος	Body
Λήξη Χρόνου-Χρονική Υπέρβαση	Timeout
Μέγιστη Μονάδα Μεταφοράς	Maximum Transfer Unit (MTU)
Μηχανισμός Κυλιόμενου Παραθύρου	Sliding Window Mechanism
Μοντέλο Πελάτη-Εξυπηρετητή	Client-Server Model
Μορφοποίηση	Shaping
Ομοιόμορφα Αναγνωριστικά Πόρων	Uniform Resource Locators (URLs)
Οντότητα	Entity
Παγκόσμιος Δικτυακός Ιστός	World Wide Web (WWW)
Περιοριστικός Σύνδεσμος	Bottleneck Link
Πλοήγηση-Περιπλάνηση	Navigation
Ποιότητα Υπηρεσίας	Quality of Service (QoS)
Πόρος	Resource
Προδιαγραφή	Specification
Προορισμός	Destination
Προσβάσιμος	Reachable
Προτίμηση	Preference
Πρότυπα-Προδιαγραφές	Standards
Πρωτόκολλο	Protocol
Συναρμολόγηση-Επανασυγκόλληση	Reassembly
Σύνδεση	Connection
Συνδέσεις Διαρκείας	Persistent Connections
Σύνδεσμος Επικοινωνίας	Link
Τηλεπικοινωνιακός Κορμός	Backbone Network
Τοπικό Δίκτυο	Local Area Network (LAN)
Τοπικότητα	Locality
Τυχαία Τοπολογία	Mesh Topology
Υπερκείμενο	Hypertext
Υπερμέσα	Hypermedia
Χρέωση	Charging
Χρονική απόσταση	Spacing
Χρόνος Αναμονής στην Ουρά	Queueing Delay
Χρόνος Δίαδοσης	Propagation Time
Χρόνος Μετάδοσης	Transmission Time
Χωρητικότητα	Capacity

Βιβλιογραφία

[WAB+96] A. Woodruff, P. Aoki, E. Brewer, P. Gauthier and L. Lawrence. An Investigation of Documents from the World Wide Web. Proceedings of Fifth WWW Conference, pages 963--980, Paris, France, May 1996.

[BCL+94] T. Berners-Lee, R. Calliau, A. Loutonen, H.F. Nielsen and A. Secret. The World Wide Web. Communications of the ACM, Volume 37, No 8, August 1994.

[BF93] N. Borenstein and N. Freed. MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies. Request for Comments 1521, Bellcore, Innosoft, September 1993.

[BLC95] T. Berners-Lee and D. Connolly. HyperText Markup Language Specification Version 2.0. Request for Comments 1866, MIT/LCS, November 1995.

[Bra96] T. Bray. Measuring the Web. Proceedings of Fifth WWW Conference, pages 993--1005, Paris, France, May 1996.

[Bra94] R. Braden. T/TCP -- TCP Extensions for Transactions, Functional Specification. Request for Comments 1644, ISI, July 1994.

[BC94] H. Braun and K. Claffy. Web Traffic Characterization: An Assesment of the Impact of caching documents frpm NCSA's Web. Proceedings of Second WWW Conference, pages 1007--1027, Chicago, Illinois, USA, October 1994.

[Jac88] V. Jacobson. Congestion Avoidance and Control. Proceedings of SIGCOMM '88 Symposium on Communications Architectures and Protocols, pages 314--329, Stanford, CA, USA, August 1988.

[CC96] R.L. Carter and M.E. Crovella. Measuring Bottleneck Link Speed in Packet-Switched Networks. Technical Report BU-CS-96-006, Computer Science Department, Boston University, March 1996.

[BCC95] A. Bestavros, C.R. Cunha and M.E. Crovella. Characteristics of WWW Client-Based Traces. Technical Report BU-CS-95-010, Boston University Computer Science Department, July 1995.

[CL86] B. Cantor and P. Lapsley. Network News Transfer Protocol. A Proposed Standard for the Stream-Based Transmission of News. Request for Comments 977, UC San Diego, UC Berkeley, February 1986.

[Cro82] D.H. Crocker. Standard for the Format of ARPA Internet Text Messages. Request for Comments 822, UDEL, August 1982.

[DKM+90] F. Davis, B. Kahle, H. Morris, J. Salem, T. Shen, R. Wang, J. Sui and M. Grinbaum. Wais interface protocol prototype functional specification (v1.5). Technical report, Thinking Machines Corporation, April 1990.

[DH95] S. Deering and R. Hinden Internet Protocol, Version 6 (IPv6) Specification. Request for Comments 1884, December 1995.

[HC95] M. Handley and J. Crowcroft. The World Wide Web: How Servers Work. *ConneXions*, 9:pages 12--24, February 1995.

[Mer95] Merit Network Inc. NSF network statistics. Available at ftp://nis.nsf.net/statistics/nsfnet, Merit Network, July 1995.

[Bey95] P. Beyssac. Bandwidth pING (BING). Available at http://web.cnam.fr/Network/bing.html, CNAM, July 1995.

[JCD90] M.L. Schoffstall, J.D Case, M. Fedor and C. Davin. Simple Network Management Protocol (SNMP). Request for Comments 1157, Internet Authority Board, May 1990.

[Lot92] M. Lottor. Internet growth. Request for Comments 1296, Network Information Systems Center, January 1992.

[Mar96] E.P. Markatos. Measuring the Web. Proceedings of Fifth WWW Conference, pages 993--1005, Paris, France, May 1996.

[MC95] M.Handley and J. Crowcroft. Problems with the World Wide Web. *ConneXions*, 11:pages 8--19, 1995.

[Nat77] National Bureau of Standards. Data Encryption Algorithm. Federal Information Processing Standards Publication 46, US Department of Commerce, January 1977.

[Org86] International Standards Organization. Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML). International Standard 8879, ISO, 1986.

[Moc87] P. Mockapetris. Domain Names - Concepts and Facilities. Request for Comments 1034, November 1987.

[PM94] V.N. Padmanabhan and J. Mogul. Improving HTTP Latency. Proceedings of Second WWW Conference, pages 995--1005, Chicago, Illinois, USA, October 1994.

[Pos81a] J.B. Postel. Internet Control Message Protocol. Request for Comments 792, ISI, September 1981.

[Pos81b] J.B. Postel. Internet Protocol. Request for Comments 791, Internet Authority Board, October 1981.

[Pos81c] J.B. Postel. Transmision Control Protocol. Request for Comments 793, Network Information Center, September 1981.

[Pos82] J.B. Postel. Simple Mail Transfer Protocol. Request for Comments 821, USC/ISI, August 1982.

[Pos83] J.B. Postel. Discard Protocol. Request for Comments 863, ISI, May 1983.

[PR95] J.E. Pitkow and M.M. Recker. Results from the Second WWW User Survey. Proceedings of Third WWW Conference, Darmstadt, Germany, April 1995.

[PR83] J.B. Postel and J. Reynolds. Telnet Protocol Specification. Request for Comments 854, ISI, May 1983.

[PR85] J.B. Postel and J.K Reynolds. File Transfer Protocol. Request For Comments 959, USC/ISI, October 1985.

[Riv92] R. Rivest. The MD5 Message Digest Algorithm. Request for Comments 1321, April 1992.

[SB95] J. Smith and R. Bohn. Throughput Research. Available at http://www.nlarn.net/Throughput, National Laboratory for Applied Network Research (NLANR), October 1995

[Sed94] J. Sedayao. Mosaic will kill my Network - Studying Network Traffic Patterns of Mosaic Use. Proceedings of Second WWW Conference, Chicago, Illinois, USA, October 1994.

[Str94] C. Strakadounas. An Application Monitoring and Link Performance Analysis System in TCP/IP Networks. Master's thesis, July 1994.

[FBF96] R. Fielding T. Berners-Lee and H. Frystyk. HyperText Transfer Protocol -- HTTP/1.0. Request for Comments 1945, MIT/LCS, UC Irvine, May 1996.

[MBM94] L. Masinter T. Berners-Lee and M. McCahill. Uniform Resource Locators (URLs). Request for Comments 1738, CERN, Xerox PARC, University of Minnesota, December 1994.

[CAV96] M.E. Crovella, V. Almeida, A. Bestavros and A. Oliveira. Characterizing Reference Locality in the WWW. Technical Report TR-96-11, Computer Science Department, Boston University, 1996.

[Wal91] J. Walrand. *Communication Networks: A First Course.* Aksen Associates, 1991.