

Πανεπιστήμιο Κρήτης
Σχολή Θετικών και Τεχνολογικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

Μέθοδοι πλοήγησης στο διαδίκτυο πάνω από συνδέσμους
περιορισμένης χωρητικότητας για Thin Clients.

Γνωριμία με το πρωτόκολλο CC/PP

Χρήστος Παπαχρήστος

Μεταπτυχιακή Εργασία

Ηράκλειο Κρήτης, Ιούνιος 2002

Πανεπιστήμιο Κρήτης
Σχολή Θετικών και Τεχνολογικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

Μέθοδοι πλοήγησης στο διαδίκτυο πάνω από συνδέσμους περιορισμένης χωρητικότητας για Thin Clients.

Γνωριμία με το πρωτόκολλο CC/PP

Εργασία που υποβλήθηκε από τον
Χρήστο Παπαχρήστο
ως μερική εκπλήρωση των απαιτήσεων για την απόκτηση
Μεταπτυχιακού Διπλώματος Ειδίκευσης στην Επιστήμη
Υπολογιστών

Συγγραφέας: _____

Χρήστος Παπαχρήστος
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή: _____

Ευάγγελος Μαρκάτος, Αναπληρωτής Καθηγητής, Επόπτης

Απόστολος Τραγανίτης, Αναπληρωτής Καθηγητής, Μέλος

Δημήτρης Πλεξουσάκης, Αναπληρωτής Καθηγητής, Μέλος

Δεκτή: _____

Πάνος Κωνσταντόπουλος, Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Ιούνιος 2002

Μέθοδοι πλοήγησης στο διαδίκτυο πάνω από συνδέσμους περιορισμένης χωρητικότητας για Thin Clients.

Γνωριμία με το πρωτόκολλο CC/PP

Χρήστος Παπαχρήστος

Μεταπτυχιακή Εργασία

Πανεπιστήμιο Κρήτης

Τμήμα Επιστήμης Υπολογιστών

Περίληψη

Τα τελευταία χρόνια ο ρυθμός εξάπλωσης του παγκόσμιου ιστού και η αύξηση των δεδομένων που διαθέτει είναι πραγματικά ένα εντυπωσιακό φαινόμενο. Μαζί όμως με τον παγκόσμιο ιστό έχουν γνωρίσει ραγδαία ανάπτυξη και οι συσκευές που προσπαθούν να τον προσπελάσουν. Πέρα από την τεχνολογική ανάπτυξη έχει αυξηθεί πάρα πολύ και το πλήθος τους. Οι συσκευές αυτές αν και προηγμένες έχουν περιορισμένες δυνατότητες σε σχέση με τους κλασσικούς υπολογιστές γραφείου. Κυριότερες αδυναμίες τους είναι το αργό (ασύρματο συνήθως) δίκτυο με το οποίο συνήθως συνδέονται και η έλλειψη ικανοποιητικού μεγέθους οθόνης για την απεικόνιση των δεδομένων. Για την καλύτερη εξυπηρέτηση όλων των συσκευών, τόσο από πλευράς ταχύτητας όσο και από πλευράς κατάλληλων δεδομένων, έχουν εφαρμοστεί κατά καιρούς διάφορες τεχνικές και μέθοδοι. Κάποιες από αυτές είναι η προσωρινή αποθήκευση των δεδομένων σε ενδιάμεσους κόμβους (proxies), που σε συνεργασία με την πρόβλεψη των δεδομένων που πρόκειται να ζητηθούν στο μέλλον μειώνουν αρκετά τον χρόνο στον οποίο εξυπηρετούνται οι χρήστες του Διαδικτύου. Μια σχετικά νεότερη τεχνική είναι η ελαχιστοποίηση των πληροφοριών που μεταφέρονται και πετυχαίνεται με την αφαίρεση των περιττών πληροφοριών και την συμπίεση των δεδομένων. Παρόμοια τεχνική είναι η μετατροπή των δεδομένων σε κατάλληλη μορφή σύμφωνα με τις δυνατότητες της κάθε συσκευής. Για να πραγματοποιηθούν αυτές οι μέθοδοι την κατάλληλη στιγμή, για την κατάλληλη συσκευή, απαιτείται επιπλέον πληροφορία που θα ενημερώνει τους εξυπηρετητές του παγκόσμιου ιστού για το είδος και τις δυνατότητες της συσκευής αλλά και του δικτύου που χρησιμοποιεί. Αυτή η πληροφορία μπορεί να μεταφερθεί με το νέο CC/PP πρωτόκολλο που προσπαθεί να γίνει πρότυπο σε αυτό τον τομέα.

Επόπτης: Μαρκάτος Ευάγγελος, Αναπληρωτής Καθηγητής

Navigation Technologies over low bandwidth links for Thin Clients

An introduction to the CC/PP protocol

Christos Papachristos
Master of Science Thesis

University of Crete
Department of Computer Science

Abstract

Over the last few years, the rapid growth of the world wide web itself and the increment of the amount of data it facilitates is amazing. Concurrently, along with the world wide web, rapid development is held in the area of the devices that are accessing the web. The technology used by such devices is improved and their variety is getting larger. Despite of their high technology they have limited capabilities when compared to the classic desktop personal computers. Their main weaknesses are the low bandwidth network that they usually use to connect to the Internet and the small screen size that is incapable of presenting many of the current available data. In order for such a device connected to the Web to be served in a reasonable amount of time, with content that can be appropriately handled by that device, many techniques and methods have been proposed. One of these techniques used is caching technology in proxies. Caching in conjunction with Prefetching reduces the latency of the data transfers. Another technique, that has been proposed lately, is the minimization of the information transferred through the Web, by removing all the unnecessary information. This method in collaboration with our approach of the URL substitution in an HTML file, can have satisfactory performance. There is also the method of data transcoding in a variety of formats depending on the capabilities of the device that requested the data. In order for the data transcoding and other methods to be applied efficiently for each device used, Web content providers need extra information. This information will supply to the servers a brief description of the characteristics of each device, and the kind of the network used. This kind of information can be transferred in the Web through the new CC/PP protocol that is proposed to become a standard in that area.

Supervisor: Markatos Evangelos, Assistant Professor

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επόπτη καθηγητή μου, κ. Ευάγγελο Μαρκάτο, ο οποίος μου έδωσε την δυνατότητα να ασχοληθώ με τον συγκεκριμένο τομέα, για τις εύστοχες πάντα παρατηρήσεις και υποδείξεις του και την δυνατότητα που μου έδωσε να γνωρίσω ερευνητές που ασχολούνται με παρόμοια θέματα στο εξωτερικό.

Επίσης, ευχαριστώ θερμά τους γονείς μου Βασίλειο και Σοφία που με στήριξαν σε όλη την διάρκεια των σπουδών μου και τον αδερφό μου Κοσμά για τις πολύτιμες συμβουλές του.

Θα ήθελα να ευχαριστήσω και τα μέλη του εργαστηρίου της Αρχιτεκτονικής Υπολογιστών για την άψογη συνεργασία που είχαμε όλο αυτό το χρονικό διάστημα.

Τέλος, θα ήθελα να ευχαριστήσω το Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης καθώς και το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για την υλικοτεχνική και οικονομική υποστήριξη που μου παρείχαν.

Περιεχόμενα

Κατάλογος Πινάκων	11
Κατάλογος Εικόνων	12
Κατάλογος Σχημάτων	13
1 Εισαγωγή	14
2 Σχετικές Εργασίες	16
2.1 Ενδιάμεσοι Εξυπηρετητές (Caching Proxies)	16
2.2 Κωδικοποίηση Delta	17
2.3 Συμπύεση Δεδομένων	18
2.4 Αποκωδικοποίηση και Κωδικοποίηση Δεδομένων (Transcoders)	19
2.5 Ελαχιστοποίηση μεγέθους ιστοσελίδων (HtmlMinimizers)	19
2.6 CC/PP (Composite Capability/Preference Profiles)	20
2.6.1 Η δομή του CC/PP	21
2.6.2 Τεχνική Περιγραφή του CC/PP και το HTTP-ext	24
2.6.2.1 Η Επικεφαλίδα Profile	25
2.6.2.2 Η επικεφαλίδα Profile-Diff	26
2.6.2.3 Η επικεφαλίδα Profile-warning	27
2.6.3 Πανεπιστήμιο Keio (CC/PP proxy)	28
2.6.4 Πανεπιστήμιο της Ουαλίας	29
2.6.5 DELI: Βιβλιοθήκη για το CC/PP	30
2.7 UAProf (User Agent Profile Specification)	30
3 Περιγραφή της παρούσης εργασίας	31
3.1 Μηχανή αναζήτησης βιβλίων (E-Book)	31
3.1.1 Περιγραφή του E-Book (http://ebook.ics.forth.gr)	31
3.1.2 Αρχιτεκτονική του συστήματος	37
3.1.3 Τεχνικές Caching και Prefetching	38
3.1.4 Τεχνολογίες που χρησιμοποιήθηκαν στο E-Book	38
3.2 Συμπιεστής Ιστοσελίδων (HtmlMinimizer)	41
3.2.1 Περιγραφή του συστήματος	42
3.2.2 Τεχνολογίες του Συστήματος	43
3.2.3 Τεχνικές του πρωτοτύπου	44
3.2.3.1 Βελτιστοποίηση της υπάρχουσας τεχνικής	46
3.2.4 Πειραματικές Μετρήσεις	49
3.3 Το πρωτόκολλο CC/PP (Composite Capability/Preference Profiles)	54
3.3.1 Επιλογή ή Μετατροπή των δεδομένων	55
3.3.2 Ο Apache web server και το CC/PP πρωτόκολλο	55
3.3.2.1 Ο Apache Web Server	56
3.3.2.2 CC/PP στον Server	56
3.3.2.3 CC/PP στον Proxy	60
3.3.3 Μετατροπή επιλεγμένων δεδομένων (Transcoding)	61
3.3.3.1 Transcoding στον Apache server	61
3.3.3.2 Transcoding στον Apache proxy	61
3.3.3.3 Οι βιβλιοθήκες λογισμικού ImageMagick και FFmpeg	62
3.3.3.4 Τα πρωτότυπα που υλοποιήθηκαν και ενδεικτικές μετρήσεις	62
4 Συμπεράσματα της εργασίας και μελλοντικές επεκτάσεις	63
5 Παράρτημα Α	66
6 Παράρτημα Β	72
7 Αναφορές και Βιβλιογραφία	75

Κατάλογος Πινάκων

Πίνακας 1 Η χρήση του xmlns:namespace μέσα στο CC/PP profile	22
Πίνακας 2 Παράδειγμα ενός profile για συσκευή της Nokia	23
Πίνακας 3 Η γραμματική σύμφωνα με την οποία θα προστεθεί η Profile επικεφαλίδα στο HTTP πρωτόκολλο	25
Πίνακας 4 Η γραμματική σύμφωνα με την οποία θα προστεθεί η Profile-Diff επικεφαλίδα στο HTTP πρωτόκολλο	26
Πίνακας 5 Παράδειγμα CC/PP με Profile και Profile-Diff επικεφαλίδες	27
Πίνακας 6 Η γραμματική και ένα παράδειγμα της επικεφαλίδας Profile-Warning	28
Πίνακας 7 Μετρήσεις για την μείωση του μεγέθους που επιτυγχάνει ο HtmlMinimizer σε 87 δημοφιλή ιστοσελίδες (Βάθος 0).....	50
Πίνακας 8 Η απόδοση του gzip πάνω στα ήδη επεξεργασμένα δεδομένα από τον HtmlMinimizer (Βάθος 0)	51
Πίνακας 9 Μετρήσεις για την μείωση του μεγέθους που επιτυγχάνει ο HtmlMinimizer σε 87 δημοφιλή ιστοσελίδες και στις θυγατρικές τους (Βάθος 1)	51
Πίνακας 10 Η απόδοση του gzip πάνω στα ήδη επεξεργασμένα δεδομένα από τον HtmlMinimizer (Βάθος 1)	52
Πίνακας 11 Μετρήσεις για την μείωση του μεγέθους που επιτυγχάνει ο HtmlMinimizer στις 87 δημοφιλέστερες ιστοσελίδες και στις θυγατρικές τους (Βάθος 2)	52
Πίνακας 12 Η απόδοση του gzip πάνω στα ήδη επεξεργασμένα δεδομένα από τον HtmlMinimizer (Βάθος 2)	53
Πίνακας 13 Ενδεικτικά αποτελέσματα transcoding κάποιων αρχείων με συγκεκριμένο Profile και παραμέτρους	63
Πίνακας 14 Το CC/PP profile που στέλνει ο Java proxy μαζί με τους HTTP headers στον Apache server	73
Πίνακας 15 Ολόκληρο το CC/PP profile που θα χρειαστεί να διαβάσει ο Apache μετά την υπόδειξη της συσκευής ότι περιγράφεται από αυτό	74

Κατάλογος Εικόνων

Εικόνα 1 Η αρχική σελίδα του E-Book όπου ο χρήστης θα εισάγει την λέξη κλειδί.....	33
Εικόνα 2 Με αυτό τον τρόπο παρουσιάζονται τα πρώτα αποτελέσματα της αναζήτησης για ένα βιβλίο σχετικό με XML	34
Εικόνα 3 Τα αποτελέσματα για την αναζήτηση τιμών για συγκεκριμένο βιβλίο. Φαίνονται μόνο όσα βιβλιοπωλεία το έχουν διαθέσιμο.....	35
Εικόνα 4 Τα ηλεκτρονικά βιβλιοπωλεία που χρησιμοποιεί το E-Book αυτή τη στιγμή για να ψάχνει τιμές βιβλίων.....	36
Εικόνα 5 Ο πηγαίος HTML κώδικας της αρχικής ιστοσελίδας του E-Book πριν την συμπίεση..	45
Εικόνα 6 Ο πηγαίος HTML κώδικας της αρχικής ιστοσελίδας του E-Book μετά την συμπίεση .	46
Εικόνα 7 Κατάσταση του browser μετά από επιλογή ενός επεξεργασμένου URL από τον HtmlMinimizer	48
Εικόνα 8 Η αρχιτεκτονική του Apache server	57
Εικόνα 9 Ο τρόπος με τον οποίο δημιουργούνται οι Apache διεργασίες από την αρχική	58
Εικόνα 10 Η μεταφορά της CC/PP πληροφορίας μέσω του Java Proxy.....	59
Εικόνα 11 Η μεταφορά της CC/PP πληροφορίας μέσω του Java Proxy και του Apache Proxy Server.....	60
Εικόνα 12 Αποτέλεσμα του transcoding σε μια GIF εικόνα. Η μετατροπή της σε μικρότερο μέγεθος	72
Εικόνα 13 Αποτέλεσμα του transcoding σε ένα MPG video. Το νέο μέγεθός του αν μικρύνουμε το παράθυρο.....	73

Κατάλογος Σχημάτων

Σχήμα 1 Η δενδρικά οργανωμένη πληροφορία στα profiles που μεταφέρονται με το CC/PP.....	22
Σχήμα 2 Η αρχιτεκτονική του E-Book.....	37

1 Εισαγωγή

Το Διαδίκτυο και τα δεδομένα που διακινούνται σε αυτό όπως τα γνωρίζουμε μέχρι σήμερα έχουν δημιουργηθεί με τέτοιο τρόπο ώστε να μπορούν να μεταφέρονται γρήγορα σε συσκευές όπως υπολογιστές γραφείου (Desktop PCs). Το τελευταίο διάστημα όμως εμφανίζονται όλο και περισσότερες συσκευές που συνδέονται στο Διαδίκτυο και προσπαθούν να προσπελάσουν αυτά τα δεδομένα. Τέτοιου είδους συσκευές όπως τα PDAs, Pocket PCs και κινητά τηλέφωνα δεν έχουν τις δυνατότητες που έχει ένας υπολογιστής γραφείου. Συνήθως διαθέτουν ασύρματες συνδέσεις οι οποίες έχουν περιορισμένη χωρητικότητα. Τέτοιες συνδέσεις χαμηλής χωρητικότητας όμως μπορούν να υπάρξουν και σε κανονικούς υπολογιστές αφού πολλοί χρήστες προσπελαίνουν το Διαδίκτυο μέσω modems. Επίσης, τέτοιες συσκευές εκ κατασκευής δεν μπορούν να διαθέτουν κάποια χαρακτηριστικά ενός παραδοσιακού υπολογιστή όπως είναι μια μεγάλη οθόνη.

Από την στιγμή που δίνεται πρόσβαση σε αυτές τις συσκευές περιορισμένων δυνατοτήτων στα δεδομένα του Διαδικτύου θα πρέπει να μπορούν να λαμβάνουν όσο το δυνατόν καταλληλότερα δεδομένα προσαρμοσμένα στις δυνατότητες αυτές. Ένα επιπρόσθετο σημαντικό βήμα, θα ήταν τα δεδομένα αυτά να μπορούν να προσαρμόζονται ανάλογα με κάποιες απαιτήσεις του χρήστη. Κάποιος χρήστης δεν επιθυμεί κάποια συγκεκριμένη μορφή δεδομένων αν η συσκευή του δεν μπορεί να τα επεξεργαστεί κατάλληλα. Αν δεν υπάρχει η δυνατότητα για ήχο τότε τα αρχεία που περιέχουν εικόνα και ήχο (video) θα μπορούσαν να μετατραπούν ώστε να μεταφερθεί μόνο η εικόνα στον τελικό προορισμό. Σε αυτό το σημείο ίσως γίνεται φανερή η αδυναμία να εξυπηρετηθεί το μεγάλο πλήθος των δυνατών συνδυασμών των δεδομένων που προκύπτει ανάλογα πάντα με την ευελιξία που θα έχει ο χρήστης ώστε να θέτει κάποιες συγκεκριμένες απαιτήσεις.

Δυο πρώτες, ακραίες ίσως, λύσεις που εμφανίζονται σε αυτό το σημείο είναι οι εξής: Στη μια κάθε εξυπηρετητής του Διαδικτύου (web server) θα πρέπει να διατηρεί διάφορες εκδόσεις από τα δεδομένα που έχει στην κατοχή του και κάθε φορά να επιλέγει πια είναι τα πιο κατάλληλα για αποστολή στον χρήστη που έκανε την αίτηση. Η δεύτερη λύση είναι ο web server να έχει την δυνατότητα να μπορεί να μετατρέπει τα δεδομένα στην κατάλληλη μορφή που ταιριάζουν στις απαιτήσεις κάθε αίτησης. Οι δύο αυτές λύσεις, στην ακραία μορφή τους, είναι τουλάχιστον ασύμφορες. Οι απαιτήσεις των χρηστών ποικίλουν και είναι δύσκολο να καλυφθούν απόλυτα με αποτέλεσμα να δημιουργούνται προβλήματα χώρου, διαχείρισης και προβλήματα πολυπλοκότητας του λογισμικού που θα κάνει τις μετατροπές για κάθε πιθανό συνδυασμό.

Για να υλοποιηθούν όμως οι προηγούμενες λύσεις θα πρέπει να καθοριστεί και ο τρόπος με τον οποίο ένας web server θα μαθαίνει από έναν πελάτη (client) τα διάφορα χαρακτηριστικά της συσκευής που διαθέτει και οποιουσδήποτε άλλους περιορισμούς-απαιτήσεις θέτει ο ίδιος ο χρήστης για τα δεδομένα που θέλει να λάβει. Μέχρι σήμερα ένας server είχε την δυνατότητα να μάθει από τον χρήστη για κάποια περιορισμένα χαρακτηριστικά της συσκευής που κάνει την αίτηση. Αυτό γίνεται με την βοήθεια του HTTP (HyperText Transfer Protocol) [3] πρωτοκόλλου το οποίο και χρησιμοποιείται ευρέως για μεταφορά πληροφοριών στο Διαδίκτυο. Η λύση με την βοήθεια του HTTP δεν παρέχει όμως κατάλληλα οργανωμένη πληροφορία στον server και δεν είναι αρκετά ευέλικτη σε μελλοντικές επεκτάσεις.

Για το σκοπό αυτό έχει εκφραστεί η επιθυμία από διάφορους οργανισμούς (W3C, World Wide Web Consortium) [4] και εταιρίες που θέλουν να οργανώσουν την πληροφορία και να την προτυποποιήσουν ώστε να μπορούν να την επεξεργάζονται καλύτερα οι web servers που διατηρούν τα δεδομένα. Οι εταιρίες (κυρίως εταιρίες κινητής τηλεφωνίας) με αυτό το τρόπο αποσκοπούν να προσφέρουν τις υπηρεσίες τους σε όλο και μεγαλύτερο κοινό και να αυξήσουν την αγοραστική πελατεία τους.

Το νέο πρωτόκολλο που έχει προταθεί για να μεταφέρει αυτή την πληροφορία είναι το CC/PP (Composite Capability/Preference Profiles) [4] και έχει προταθεί από τον οργανισμό W3C. Δεν έχει καθοριστεί πλήρως πως ακριβώς θα λειτουργεί μια και είναι σε εξέλιξη η ανάπτυξη του αλλά υπάρχουν κάποιες προτάσεις για το πως μπορεί να συνεισφέρει. Με βάση αυτή την πρόταση έχουν υλοποιηθεί κάποιες εφαρμογές για να μπορέσουν να ενισχύσουν και να προωθήσουν την γρήγορη διάδοση του CC/PP στο Διαδίκτυο.

Στο υπόλοιπο της αναφοράς θα περιγράψουμε κάποιες εφαρμογές που υλοποιήθηκαν στα πλαίσια αυτής της εργασίας. Κάποιες από αυτές εφαρμόζουν ήδη υπάρχουσες και δοκιμασμένες τεχνικές όπως caching και prefetching, (<http://ebook.ics.forth.gr>). Σε άλλη εφαρμογή θα παρουσιάσουμε πως γίνεται να μεταφερθούν όσο το δυνατόν λιγότερα δεδομένα από τον κώδικα μιας ιστοσελίδα στον χρήστη χωρίς αυτό να επηρεάζει την πλοήγησή του στο Διαδίκτυο. Τέλος θα παρουσιαστεί το πρωτόκολλο CC/PP όπως έχει οριστεί μέχρι αυτή τη στιγμή από το W3C, μια υλοποίηση που κάνει χρήση του νέου πρωτοκόλλου και πως μπορεί να χρησιμοποιηθεί σε 2 διαφορετικά βασικά τμήματα του Διαδικτύου (server-side & proxy-side). Σε αυτήν παρουσιάζεται όλη η αλληλεπίδραση μεταξύ client και server για να μπορέσει ο τελευταίος να στείλει τα κατάλληλα δεδομένα. Για την υλοποίηση αυτή χρησιμοποιήθηκε εκτενώς ο Apache web server [14].

2 Σχετικές Εργασίες

Σε όλη την πορεία εξέλιξης του Διαδικτύου βασικό και διαχρονικό μέλημα οργανισμών και ερευνητικών ιδρυμάτων που δημιουργούν τα πρότυπα, σύμφωνα με τα οποία λειτουργούν οι εφαρμογές του Διαδικτύου, είναι πως να εξυπηρετηθεί "καλύτερα" ο χρήστης που προσπαθεί να ανακτήσει πληροφορίες που τον ενδιαφέρουν. Το "καλύτερα" στην περίπτωση που θα μελετήσουμε σε αυτή την εργασία έχει να κάνει με το πόσο γρήγορα μπορεί να λάβει τα δεδομένα κάποιος χρήστης του Διαδικτύου και κατά πόσο αυτά τα δεδομένα μπορούν να του φανούν χρήσιμα και κατανοητά όταν υπάρχουν κάποιοι φυσικοί περιοριστικοί παράγοντες στην όλη διαδικασία. Προς αυτή την κατεύθυνση έχουν οδηγηθεί αρκετές μελέτες και έχουν δημιουργηθεί πλήθος εφαρμογών και μηχανισμών.

2.1 Ενδιάμεσοι Εξυπηρετητές (Caching Proxies)

Ευρέως γνωστοί μηχανισμοί που χρησιμοποιούνται εδώ και καιρό στο Διαδίκτυο είναι το Caching και το Prefetching που γίνεται από ενδιάμεσους κόμβους-servers. Οι κόμβοι αυτοί ονομάζονται Proxies [24][25]. Το παραδοσιακό μοντέλο πάνω στο οποίο βασίζεται το Διαδίκτυο αποτελείται από κάποιο πελάτη-client που κάνει μια αίτηση σε κάποιο εξυπηρετητή-server για ένα συγκεκριμένο αρχείο. Όλη η διαδικασία της αίτησης του συγκεκριμένου αρχείου μπορεί να γίνει πολλές φορές σε πολύ μικρό χρονικό διάστημα με αποτέλεσμα ο εξυπηρετητής να πρέπει να εκτελεί την ίδια συνεχώς διαδικασία για κάθε μια από τις αιτήσεις που δέχεται. Αυτό σημαίνει πως ο υπολογιστής στον οποίο θα λειτουργεί ο εξυπηρετητής θα είναι μονίμως φορτωμένος και πολλές φορές θα φτάνει σε σημείο να μην μπορεί να εξυπηρετήσει νέες αιτήσεις. Μια άλλη σημαντική διάσταση που έχει η επικοινωνία πελάτη-εξυπηρετητή είναι ο χρόνος. Ο χρόνος τον οποίο χρειάζεται ένα αρχείο να μεταφερθεί από τον server στον client εξαρτάται σε μεγάλο βαθμό από το μέγεθος του αρχείου και την χωρητικότητα των συνδέσμων από τους οποίους περνάει για να φτάσει στον προορισμό του. Όσο πιο λίγοι είναι οι ενδιάμεσοι σύνδεσμοι τόσο γρηγορότερα θα πάρει ο πελάτης το αρχείο.

Για να βελτιωθούν τα παραπάνω εγκαθιστούμε έναν ενδιάμεσο proxy server ο οποίος θα αποθηκεύει αυτά τα αρχεία που ζητάνε πιο συχνά οι χρήστες. Με αυτό τον τρόπο επόμενες αιτήσεις που θα ζητάνε αρχείο ήδη αποθηκευμένο στον proxy από προηγούμενη αίτηση, θα εξυπηρετούνται από τον proxy και όχι από τον server. Διαφορετικά η αίτηση προωθείται

κανονικά προς τον server. Συνήθως τέτοιου είδους proxies τοποθετούνται αρκετά κοντά στους πελάτες που θέλουν να εξυπηρετήσουν. Κάθε proxy έχει σαν σκοπό να εξυπηρετήσει κάποιους συγκεκριμένους πελάτες και δεν είναι συνηθισμένο να υπάρχουν proxies που να εξυπηρετούν όλους τους χρήστες του Διαδικτύου.

Στην ίδια κατηγορία servers ανήκουν και οι reverse proxies οι οποίοι όμως εγκαθίστανται σε σημείο του δικτύου κοντά στον server που περιέχει τα δεδομένα και όχι κοντά στα τερματικά των πιθανών χρηστών. Τέτοιου είδους proxies έχουν σαν σκοπό να ελαφρύνουν το φόρτο του πραγματικού server όταν δέχεται πολλές αιτήσεις. Μεγάλα sites διαθέτουν παραπάνω από έναν τέτοιους reverse proxies για καλύτερη εξισορρόπηση του φόρτου.

Επεκτείνοντας αυτό τον μηχανισμό, ένας proxy server θα μπορούσε με κάποιο τρόπο να "προβλέψει" τα δεδομένα που είναι πιθανό να του ζητήσουν οι χρήστες τους οποίους εξυπηρετεί και να τα μεταφέρει ο ίδιος για λογαριασμό τους (Prefetching) πριν ακόμα ζητηθούν. Αυτό μπορεί να επιτευχθεί ικανοποιητικά για στατικές ιστοσελίδες αλλά όχι και για δυναμικές ιστοσελίδες (όπως συμβαίνει και για το caching). Οι δυναμικές ιστοσελίδες δημιουργούνται βασισμένες πολλές φορές σε προσωπικά στοιχεία και παραμέτρους του χρήστη. Από την στιγμή που τα δεδομένα έχουν μεταφερθεί και έχουν αποθηκευτεί στον proxy οι χρήστες θα μπορούν να τα λαμβάνουν γρηγορότερα. Η δυσκολία σε αυτή τη μέθοδο όμως υπάρχει στο πως ένας caching proxy μπορεί να προβλέψει τι θα του ζητηθεί σε επόμενη χρονική στιγμή. Μια μέθοδος πρόβλεψης είναι να φέρει δεδομένα με παρόμοιο περιεχόμενο με αυτά που ήδη έχουν ζητήσει οι πελάτες [27]. Άλλη μέθοδος που προτείνεται είναι να επιλέγονται σαν πιθανά δεδομένα για μελλοντικές αιτήσεις οι διαδικτυακοί σύνδεσμοι της ιστοσελίδας που ο χρήστης βλέπει εκείνη τη στιγμή. Το πιθανότερο είναι να ακολουθήσει ένα από αυτά οπότε και η μέθοδος έχει επιτύχει το σκοπό της [43]. Επίσης μπορεί να βασιστεί στα πιο δημοφιλή αρχεία που έχουν ζητηθεί το τελευταίο χρονικό διάστημα [25]. Με αυτό το τρόπο μειώνεται δραστικά ο χρόνος που πέρασε από την στιγμή που κάποιος ζήτησε μια ιστοσελίδα μέχρι την στιγμή που θα εμφανιστεί το αποτέλεσμα (round trip time).

2.2 Κωδικοποίηση Delta

Ένας άλλος μηχανισμός που έχει προταθεί από ερευνητές παλαιότερα και μπορεί να συνεργαστεί και με μηχανισμούς caching, είναι η μεταφορά της πληροφορίας που δημιουργεί μια ιστοσελίδα σε σχέση με μια ιστοσελίδα που είχε μεταφερθεί σε προηγούμενη χρονική στιγμή (delta encoding) [16][17]. Αυτή η τεχνική προσπαθεί να κάνει μια μορφή κωδικοποίησης των ιστοσελίδων έτσι ώστε σε δυο διαδοχικές αιτήσεις από κάποιον χρήστη, την μεν πρώτη φορά να

μεταφερθεί ολόκληρη η ιστοσελίδα, ενώ για την δεύτερη και επόμενες ιστοσελίδες μπορούν να μεταφερθούν μόνο οι διαφορές αυτών από κάποια που ήδη έχει μεταφερθεί. Οι διαφορές αυτές σχετίζονται με το κείμενο που δημιουργεί την ιστοσελίδα, είτε αυτό ανήκει στο περιεχόμενο της ιστοσελίδας, είτε ανήκει στην δομή αυτής. Με την τεχνική αυτή επιτυγχάνεται να μεταφέρονται προς τον χρήστη πολύ λιγότερα δεδομένα. Επίσης γίνεται φανερό πως αυτή η μέθοδος αποδίδει για ιστοσελίδες που είναι αρκετά όμοιες μεταξύ τους, πράγμα που ισχύει όταν πρόκειται για ιστοσελίδες της ίδιας εταιρίας ή οργανισμού και διατηρούν κάποια συγκεκριμένη μορφή στην δομή τους.

Για να μπορέσει να δουλέψει αυτή η τεχνική χρειάζεται ένας proxy server που θα βρίσκει ποιες είναι οι διαφορές μεταξύ των ιστοσελίδων που έχουν ήδη ζητηθεί με αυτή που ζητείται εκείνη τη στιγμή. Μια μικρή αδυναμία αυτής της μεθόδου είναι πως και ο χρήστης, ανεξαρτήτως της συσκευής που θα διαθέτει θα πρέπει να διαθέτει κατάλληλο λογισμικό. Με αυτό η εφαρμογή που χρησιμοποιεί για να βλέπει τις ιστοσελίδες (browser) θα κάνει την αντίστροφη διαδικασία της επανασύνθεσης της ιστοσελίδας από τις διαφορές που έχουν μεταφερθεί εκείνη τη στιγμή και της ιστοσελίδας που είχε μεταφερθεί προηγουμένως. Επίσης αυτός ο μηχανισμός προϋποθέτει πως η βασική σελίδα που θα χρησιμοποιηθεί για την επανασύνθεση κάποιας επόμενης να βρίσκεται στην διάθεση της εφαρμογής. Σημερινές εφαρμογές για πλοήγηση στο Διαδίκτυο (π.χ Netscape, IExplorer) διαθέτουν τέτοιο μηχανισμό προσωρινής αποθήκευσης ιστοσελίδων (cache) ο οποίος μπορεί να χρησιμοποιηθεί για το σκοπό αυτό. Με την μέθοδο αυτή αναφέρουν πως περισσότερα από 35% του μεγέθους των συνολικών δεδομένων θα μπορούσε να μην έχει μεταφερθεί. Σε κάποιες ειδικές περιπτώσεις μάλιστα αναφέρουν ακόμα μεγαλύτερα ποσοστά.

2.3 Συμπύεση Δεδομένων

Η συμπύεση των δεδομένων είναι γνωστή για την εξοικονόμηση χώρου που επιτυγχάνεται σε δίσκους αφού τα αρχεία έχουν μικρότερο μέγεθος. Τέτοια μορφή συμπύεσης μπορούν να υποστούν και τα δεδομένα που μεταφέρονται με το HTTP πρωτόκολλο [28]. Ήδη οι πιο γνωστοί web servers (Apache, IIS, Netscape, Jigsaw) άλλα και οι διάφορες εφαρμογές πλοήγησης υποστηρίζουν την αποστολή δεδομένων σε συμπιεσμένη μορφή. Η επικεφαλίδα στο HTTP πρωτόκολλο που υποδεικνύει στην εφαρμογή πλοήγησης ότι τα δεδομένα που μεταφέρονται είναι συμπιεσμένα για να τα χειριστεί κατάλληλα είναι η «Content-Encoding». Στην εργασία για το delta encoding [17] αναφέρουν πως η συμπύεση των δεδομένων από HTTP κίνηση μπόρεσε να μειώσει το συνολικό μέγεθος που μεταφέρθηκε κατά 38.7%. Το λογισμικό που χρησιμοποιήθηκε

ήταν το gzip του unix λειτουργικού συστήματος. Όταν εφαρμοστεί αυτή η μέθοδος συμπίεσης σε απλό κείμενο μπορεί να επιτύχει συμπίεση περισσότερο από 50%.

2.4 Αποκωδικοποίηση και Κωδικοποίηση Δεδομένων (Transcoders)

Μια κατηγορία ενδιάμεσων servers οι οποίοι επεμβαίνουν στη μορφή και ποιότητα των δεδομένων που μεταφέρονται είναι οι transcoders. Αυτοί οι servers μπορούν και μετατρέπουν τα δεδομένα που περνάνε μέσα από αυτούς σε διαφορετικές μορφές. Όταν ένας χρήστης έχει ενημερώσει με κάποιο τρόπο έναν τέτοιο server πως μπορεί να λαμβάνει δεδομένα μόνο σε κάποια συγκεκριμένη μορφή π.χ μόνο JPEG εικόνες και όχι GIF τότε αυτός ο transcoder θα προσπαθήσει να μετατρέψει όλες τις εικόνες που προορίζονται για τον συγκεκριμένο χρήστη σε μορφή JPEG.

Μια υλοποίηση ενός τέτοιου transcoder είναι ο MOXY proxy [18] που προσπαθεί να εξυπηρετήσει αιτήσεις που προέρχονται από χρήστες με κινητές συσκευές. Προσπαθεί να μεταδώσει τα κατάλληλα δεδομένα ανάλογα πάντα με κάποια χαρακτηριστικά που του έχουν δώσει. Στην υλοποίηση τους οι απαιτήσεις του χρήστη για διάφορα επίπεδα ποιότητας που προσφέρει ο MOXY δίνονται σαν παράμετροι από τον ίδιο τον χρήστη σε ειδική ιστοσελίδα του που διατηρεί ο ίδιος ο proxy server. Άρα τα δεδομένα των προτιμήσεων του χρήστη δεν μεταφέρονται ενσωματωμένα μέσα στην επικεφαλίδα του HTTP ή με κάποιο διαφορετικό πρωτόκολλο.

Παρόμοια υλοποίηση έχει γίνει και από την IBM [28]. Το σύστημα που ανέπτυξαν είναι και αυτός ένας proxy που δέχεται τις προτιμήσεις του χρήστη μέσα από applets. Στο σύστημά τους ο χρήστης μπορεί να επιλέξει ανάμεσα στον χρόνο τον οποίο θα χρειαστεί μια εικόνα για να εμφανισθεί στον χρήστη και στην ποιότητα αυτής. Μπορεί παράλληλα να επέμβει και στο κείμενο της ιστοσελίδας και ανάλογα με το τι θα του υποδείξει ο χρήστης, το σύστημα μπορεί να αφαιρέσει αυτόνομα τμήματα κώδικα της ιστοσελίδας όπως είναι ο κώδικας JavaScript. Αυτό βέβαια έχει επιπτώσεις στη μορφή που θα παρουσιαστεί η ιστοσελίδα στον χρήστη αφού μπορεί να εξαφανιστούν από αυτήν διάφορα τμήματα. Ο χρήστης όμως υποτίθεται πως το γνωρίζει αφού ήταν δική του επιλογή.

2.5 Ελαχιστοποίηση μεγέθους ιστοσελίδων (HtmlMinimizers)

Υπάρχουν εφαρμογές που κάνουν αποκλειστικά επεξεργασία στο κείμενο το οποίο αποτελεί την ιστοσελίδα. Τέτοιου είδους εργαλεία προσπαθούν να κάνουν κάποιο είδος

μορφοποίησης στον HTML (HyperText Markup Language) [44] κώδικα της ιστοσελίδας ώστε να μεταφερθούν λιγότερα bytes. Ο κώδικας αυτό εκτελείται στο μηχάνημα του χρήστη. Η πληροφορία που παραλείπουν τέτοια εργαλεία είναι τα σχόλια της HTML, τα περιττά κενά, οι "λευκοί" χαρακτήρες και οι περιττές κενές γραμμές. Πολλές φορές επεξεργάζονται ακόμα και το μέγεθος των συνδέσμων (URL links) που οδηγούν σε άλλες ιστοσελίδες [30]. Όσον αφορά την περίπτωση των συνδέσμων αναφέρουν πως αυτό μπορεί να γίνει από μεριάς του server από τον διαχειριστή του server. Ο διαχειριστής είναι αυτός που θα υποδείξει κάποιους κανόνες σύμφωνα με τους οποίους ο server θα αντικαθιστά τα μεγάλα σε μέγεθος links με άλλα μικρότερα. Για παράδειγμα μια πιθανή αντιστοίχιση θα μπορούσε να είναι (`r/dc/48 <-----> /dhtml/column48/`) όπου ο αριστερός σχετικός σύνδεσμος (relative link) είναι μια σύντομη περιγραφή του περιγραφικού μεγάλου συνδέσμου χρησιμοποιώντας μόνο τα αρχικά και κάποια τμήματα κλειδιά όπως είναι το νούμερο 48 στην προκειμένη περίπτωση.

Συγκεκριμένα αναφέρουν τον Apache web server ο οποίος διαθέτει ειδικό κομμάτι λογισμικού (module rewrite) που μπορεί και κάνει μια μοναδική αντιστοίχιση των links που έχει γράψει ο προγραμματιστής της HTML με άλλα μικρότερου μεγέθους. Σχεδόν όλες οι ιστοσελίδες περιέχουν τα παραπάνω στοιχεία. Ο κώδικας HTML των ιστοσελίδων μπορεί να έχει δημιουργηθεί είτε με την βοήθεια κάποιου ειδικού εργαλείου (Html Editors) είτε απευθείας από τον προγραμματιστή. Σε κάθε περίπτωση ο κώδικας που δημιουργείται πρέπει να είναι ευανάγνωστος και με σχόλια για να μπορεί κάποιος προγραμματιστής να είναι σε θέση να κάνει αλλαγές σε περίπτωση που χρειαστεί. Για την εφαρμογή όμως (browser) που θα χρησιμοποιήσει αυτόν τον κώδικα για να δώσει το αποτέλεσμα στο χρήστη όλα τα παραπάνω δεν χρειάζονται και μπορούν να παραλειφθούν. Αυτή η διαδικασία μπορεί να γίνει είτε τη στιγμή που θα ζητηθεί κάποιο συγκεκριμένο αρχείο (real time, από κάποια εφαρμογή), είτε μπορεί ο διαχειριστής του διαδικτυακού τόπου (site), να τοποθετεί στον server τα φιλτραρισμένα αρχεία και όχι τα αρχικά. Αυτό βέβαια έχει κάποιο φόρτο εργασίας παραπάνω για τον διαχειριστή του συστήματος γιατί θα πρέπει να κρατάει πάντα ενήμερα τα φιλτραρισμένα αρχεία που είναι διαθέσιμα σε σχέση με τα αυθεντικά ακόμα και όταν υπάρχουν ελάχιστες αλλαγές.

2.6 CC/PP (Composite Capability/Preference Profiles)

Όπως αναφέρθηκε παραπάνω, ήδη κάποιες εφαρμογές έχουν αναπτυχθεί αρκετά ώστε να μπορούν να μεταφέρουν σε κάποιες συσκευές, γρήγορα, όσο το δυνατόν λιγότερα δεδομένα. Για να το κάνουν αυτό όμως θα πρέπει με κάποιο τρόπο αυτές οι εφαρμογές να έχουν μια ιδέα για το είδος της συσκευής που έχουν να εξυπηρετήσουν κάθε φορά. Αυτή τη στιγμή οι συσκευές που

μπορούν και προσπελαύνουν το Διαδίκτυο είναι πάρα πολλές οι δυνατότητες που έχει η κάθε μια ποικίλουν. Αν λάβουμε υπόψη μας και τις διαφορετικές απαιτήσεις των χρηστών τότε οι εφαρμογές αυτές θα πρέπει να έχουν μια λεπτομερή εικόνα του τι μπορεί να δεχτεί κάθε συσκευή και τι απαιτεί ο χρήστης της.

Οι web servers σήμερα έχουν μια περιορισμένη εικόνα για το είδος της συσκευής που τους ζητά ένα αρχείο. Αυτή η πληροφορία παρέχεται από ένα συγκεκριμένο πεδίο της επικεφαλίδας του HTTP πρωτοκόλλου. Αυτό είναι το User-Agent πεδίο που περιέχει κάποιες πληροφορίες όπως είναι το λειτουργικό που τρέχει η συσκευή, τον τύπο της εφαρμογής που χρησιμοποιεί τις μορφές δεδομένων που μπορεί να δεχτεί (π.χ κώδικα Html, απλό κείμενο κ.α) και τις γλώσσες (π.χ Αγγλικά ή Ελληνικά) που υποστηρίζει. Επειδή όμως δεν είναι ιδιαίτερα επεκτάσιμη αυτή η μέθοδος μια και οι συσκευές και οι δυνατότητές τους συνεχώς πληθαίνουν και υπάρχει ιδιαίτερη ανάγκη για καλύτερη οργάνωση αυτής της πληροφορίας ο οργανισμός W3C έχει προτείνει ένα νέο πρωτόκολλο.

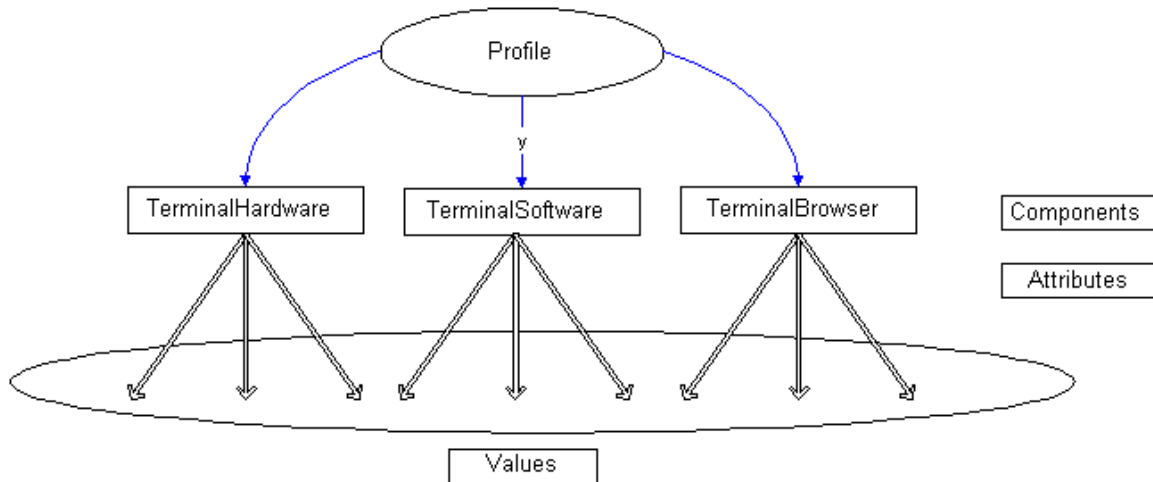
Τα αρχικά αυτού του πρωτοκόλλου είναι CC/PP [4] και έχει να κάνει με την συλλογή των ικανοτήτων της συσκευής και των προτιμήσεων του χρήστη. Σαν στόχο έχει να μεταφέρει στους servers που διατηρούν τα δεδομένα του Διαδικτύου μια οργανωμένη μορφή πληροφορίας που θα περιγράφει τις προδιαγραφές της συσκευής του χρήστη και κάποιες προτιμήσεις του ίδιου. Ο οργανισμός W3C θέλει να το προτείνει για πρότυπο άλλα ο σχεδιασμός του βρίσκεται ακόμη σε αρχικά στάδια.

2.6.1 Η δομή του CC/PP

Το CC/PP είναι βασισμένο στις περιγραφικές τεχνικές XML/RDF. Η RDF [6] (Resource Definition Framework) είναι ένα πλαίσιο με την βοήθεια του οποίου μπορούν να περιγραφούν τα μεταδεδομένα του Διαδικτύου. Η RDF περιγραφή χρησιμοποιήθηκε αφού οι πληροφορίες για τις συσκευές (user agent profiles) είναι μεταδεδομένα που χρησιμεύουν για την διαπραγμάτευση μεταξύ server και συσκευής. Για να μπορεί να είναι το CC/PP όσο το δυνατόν πιο επεκτάσιμο οι πληροφορίες που μεταφέρονται κωδικοποιούνται με την βοήθεια των τεχνολογιών XML/RDF. Το συντακτικό της XML χρησιμοποιείται για να περιγράψει και να αναπαραστήσει την πληροφορία που μεταφέρεται.

Το βασικό μοντέλο στο οποίο βασίζεται το πρωτόκολλο είναι ένα δένδρο. Οι αρχικοί κόμβοι του δένδρου θα είναι τα συστατικά του στοιχείου. Κάθε ένα από αυτά τα στοιχεία θα μπορεί να έχει συλλογές από πεδία. Τα συστατικά στοιχεία (components) ενός profile μπορεί να είναι το υλικό της συσκευής πάνω στο οποίο τρέχει το λογισμικό, το ίδιο το λογισμικό πάνω στο οποίο

τρέχουν οι εφαρμογές και οι εφαρμογές οι ίδιες. Γραφικά τα δομικά στοιχεία ενός profile παρουσιάζεται στο παρακάτω Σχήμα 1.



Σχήμα 1 Η δενδρικά οργανωμένη πληροφορία στα profiles που μεταφέρονται με το CC/PP

Το profile του CC/PP κάνει χρήση του χαρακτηριστικού namespace της XML με την χρήση του πεδίου xmlns ώστε να μπορεί να διαχωρίσει πολλαπλά πεδία και τιμές. Με αναφορά σε κάποιο XML namespace κάποια χαρακτηριστικά ανήκουν σε μια κατηγορία και κάποια άλλα σε άλλη. Παράδειγμα XML namespace μέσα σε RDF είναι το παρακάτω [5]:

```
<RDF xmlns=http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
```

Πίνακας 1 Η χρήση του xmlns μέσα στο CC/PP profile

Ένα παράδειγμα ενός profile όπως παρέχεται από την Nokia για μια συσκευή που τρέχει την εφαρμογή πλοήγησης Mozilla είναι στον επόμενο πίνακα (Πίνακας 2).

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

```

```
xmlns:prf="http://www.w3.org/TR/WD-profile-vocabulary#">
```

```
<rdf:Description about="HardwarePlatform">
```

```
<prf:Defaults
```

```
  Vendor="Nokia"
```

```
  Model="2160"
```

```
  Type="PDA"
```

```
  ScreenSize="800x600x24"
```

```
  CPU="PPC"
```

```
  Keyboard="Yes"
```

```
  Memory="16mB"
```

```
  Bluetooth="YES"
```

```
  Speaker="Yes" />
```

```
<prf:Modifications
```

```
  Memory="32mB" />
```

```
</rdf:Description>
```

```
<rdf:Description about="SoftwarePlatform">
```

```
<prf:Defaults
```

```
  OS="EPOC1.0"
```

```
  HTMLVersion="4.0"
```

```
  JavaScriptVersion="4.0"
```

```
  WAPVersion="1.0"
```

```
  WMLScript="1.0" />
```

```
<prf:Modifications
```

```
  Sound="Off"
```

```
  Images="Off" />
```

```
</rdf:Description>
```

```
<rdf:Description about="EpicEmail1.0">
```

```
<prf:Defaults
```

```
  HTMLVersion="4.0" />
```

```
</rdf:Description>
```

```
<rdf:Description about="EpicCalendar1.0">
```

```
<prf:Defaults
```

```
  HTMLVersion="4.0" />
```

```
</rdf:Description>
```

```
<rdf:Description about="UserPreferences">
```

```
<prf:Defaults
```

```
  Language="English"/>
```

```
</rdf:Description>
```

```
</rdf:RDF>
```

Πίνακας 2 Παράδειγμα ενός profile για συσκευή της Nokia

Με έντονο μαύρο χρώμα στον πίνακα φαίνονται τα components από τα οποία αποτελείται το profile της συσκευής. Θα εξηγήσουμε κάποιες λέξεις κλειδιά που εμφανίζονται σε αυτόν τον πίνακα. Η λέξη κλειδί **Modification** σημαίνει πως ο χρήστης έχει κάνει κάποια αλλαγή

στην συσκευή του και την αλλαγή αυτή την εισάγει με αυτό τον τρόπο για την λάβει υπόψη του ο server. Το κλειδί **Defaults** σημαίνει πως η συσκευή είναι ακριβώς όπως αγοράστηκε από τον κατασκευαστή, χωρίς να έχουν γίνει αλλαγές στα χαρακτηριστικά της.

Τα ενεργά στοιχεία του Διαδικτύου που θα πρέπει να γνωρίζουν το CC/PP θα πρέπει να είναι οι συσκευές χρήστη (clients), οι servers και οι ενδιάμεσοι κόμβοι. Οι ενδιάμεσοι κόμβοι θα πρέπει να μπορούν να διαβάζουν αυτή την πληροφορία και αν χρειάζεται να προσθέτουν για να μπορεί ο server να επιλέξει ή να μετατρέψει τα δεδομένα στην κατάλληλη μορφή. Πλεονέκτημα του να γνωρίζουν οι servers και οι ενδιάμεσοι κόμβοι την ύπαρξη του CC/PP είναι η τεχνική του caching που μπορεί να εφαρμοστεί ώστε να μην ζητείται κάθε φορά το profile της συσκευής. Αυτό ισχύει στην περίπτωση που έχουμε έμμεσες αναφορές για το profile κάποιας συσκευής.

2.6.2 Τεχνική Περιγραφή του CC/PP και το HTTP-ext

Δυο ιδέες που υπάρχουν για το CC/PP και πως αυτό θα μεταφέρει τις πληροφορίες από τον χρήστη στους servers, είναι είτε να αποτελεί ένα ξεχωριστό πρωτόκολλο, είτε να ενσωματωθεί σαν επέκταση του HTTP. Θα περιγραφεί το CC/PP πρωτόκολλο βασισμένο στο HTTP το οποίο μπορεί να δέχεται επεκτάσεις σύμφωνα πάντα με κάποιους κανόνες [1] [2].

Το κύριο σκεπτικό σε αυτή τη μέθοδο είναι να στέλνεται όσο το δυνατόν λιγότερη πληροφορία από τον χρήστη γιατί όπως φάνηκε και από τον πίνακα 2 το profile μπορεί να είναι αρκετά μεγάλο ώστε να ταξιδεύει πάνω από ένα αργό δίκτυο. Το πρόβλημα αυτό λύνεται με την τεχνική των URIs (Uniform Resource Identifiers) που μπορούν να χρησιμοποιηθούν μέσα στο CC/PP σαν δείκτες προς την πραγματική πληροφορία. Με τον τρόπο αυτό μια συσκευή δεν είναι ανάγκη να μεταδώσει όλα τα χαρακτηριστικά της η ίδια, άλλα μπορεί να ενημερώσει τον server από ποιο σημείο μπορεί αυτός να τα προμηθευτεί. Η συσκευή θα προσθέσει πληροφορία μόνο αν υπάρχουν διαφοροποιήσεις από τα αρχικά της χαρακτηριστικά (π.χ αναβάθμιση λογισμικού ή μνήμης). Με αυτή του την μορφή, μέσα στο HTTP, το CC/PP δεν διατηρεί πληροφορία για την κατάστασή του ανάμεσα στις διάφορες αιτήσεις (stateless).

Για να ενσωματωθεί η CC/PP πληροφορία με την επικεφαλίδα του HTTP πρωτοκόλλου θα πρέπει να προστεθούν κάποια νέα πεδία στο HTTP. Τα πεδία αυτά είναι τα:

Profile, Profile-Diff-X και Profile-Warning.

2.6.2.1 Η Επικεφαλίδα Profile

Η επικεφαλίδα αυτή θα προστίθεται στις αιτήσεις και θα μπορεί να περιέχει μια λίστα από αναφορές σε διευθύνσεις από όπου θα μπορεί να σχηματίζεται η CC/PP περιγραφή. Ο Πίνακας 3 παρουσιάζει την γραμματική την οποία θα ακολουθεί αυτό το πεδίο.

Profile	= profile-field-name ":" 1#reference
profile-field-name	= "Profile"
reference	= <"> (absoluteURI profile-diff-name) <">
profile-diff-name	= profile-diff-number "-" profile-diff-digest
profile-diff-number	= 1#DIGIT
profile-diff-digest	= sp; < MD5 message digest encoded by base64 >
DIGIT	= <any US-ASCII digit "0".."9">

Πίνακας 3 Η γραμματική σύμφωνα με την οποία θα προστεθεί η Profile επικεφαλίδα στο HTTP πρωτόκολλο

Η επικεφαλίδα Profile μπορεί να περιέχει είτε λίστα αναφορών σε άλλες διευθύνσεις, είτε έναν Profile-diff-number. Το Profile-diff-number είναι ένας αριθμός που χρησιμοποιείται για να αναγνωρίσει ο server ή ο proxy την επόμενη επικεφαλίδα (Profile-Diff- [0-9]). Παραδείγματα Profile επικεφαλίδων είναι τα:

Profile: "http://www.aaa.com/hw", "http://www.bbb.com/sw"

Profile: "http://www.aaa.com/hw", "1-SADfSDd354sdfsdfDSdfIo==", "http://www.bbb.com/"

Στην πρώτη περίπτωση περιέχονται 2 αναφορές σε διευθύνσεις από όπου μπορούν να βρεθούν πληροφορίες για την συσκευή. Η πρώτη έχει πληροφορίες για το υλικό της συσκευής και η δεύτερη για το λογισμικό αυτής.

Στην δεύτερη περίπτωση υπάρχουν πάλι οι ίδιες αναφορές με την προσθήκη του πεδίου "1-SADfSDd354sdfsdfDSdfIo==". Η πληροφορία αυτού του πεδίου δηλώνει στον παραλήπτη του πως το πεδίο profile-diff-number της επικεφαλίδας Profile-Diff που ακολουθεί είναι το 1 (δηλώνεται στην αρχή). Αυτό σημαίνει πως ο παραλήπτης (server) θα ψάξει για επικεφαλίδα με το όνομα Profile-Diff-1. Το υπόλοιπο τμήμα που φαίνεται υπογραμμισμένο είναι το αποτέλεσμα που παράγεται όταν δοθεί σαν είσοδος στον MD5 [40] αλγόριθμο η τιμή του Profile-Diff-1

πεδίου. Ο MD5 αλγόριθμος παίρνει σαν είσοδο ένα μήνυμα αυθαίρετου μεγέθους και παράγει σαν έξοδο ένα αποτύπωμα της εισόδου μεγέθους 128 bits. Το πεδίο αυτό που ονομάζεται profile-diff-digest χρησιμοποιείται για γρήγορη αναζήτηση του Profile-Diff στους proxies που έχουν αποθηκεύσει αυτή την πληροφορία.

2.6.2.2 Η επικεφαλίδα Profile-Diff

Η επικεφαλίδα Profile-Diff χρησιμοποιείται και αυτή όταν πρόκειται για αιτήσεις του χρήστη. Χρησιμοποιείται πάντα μαζί με την επικεφαλίδα Profile μέσα στην ίδια αίτηση. Ο

Πίνακας 4 περιγράφει την γραμματική που ακολουθεί η σύνταξή της.

```
Profile-Diff      = profile-diff-field-name ":" profile-desc
profile-diff-field-name = "Profile-Diff-" profile-diff-number
profile-desc      = < the CC/PP description based on XML/RDF text format
```

Πίνακας 4 Η γραμματική σύμφωνα με την οποία θα προστεθεί η Profile-Diff επικεφαλίδα στο HTTP πρωτόκολλο

Οι τιμές που μπορεί να πάρει το profile-diff-number είναι από 0-9. Αυτή η επικεφαλίδα περιέχει την CC/PP περιγραφή της συσκευής βασισμένη σε XML κείμενο και χρησιμοποιείται συνήθως για να δηλώσει κάποιες αλλαγές χαρακτηριστικών σε σχέση με αυτά που δηλώνουν οι αναφορές στην επικεφαλίδα Profile. Σε περίπτωση που δεν υπάρχει καμία αναφορά στο Profile τότε το Profile-Diff περιέχει όλη την απαραίτητη CC/PP πληροφορία. Ο Πίνακας 5 περιέχει ένα ολοκληρωμένο παράδειγμα που περιέχει και τις δυο αυτές επικεφαλίδες.

```
Profile: "1-P1GRkSjKK50aTWXXndFcSQ=="
Profile-Diff-1: <?xml version="1.0"?>
  <RDF xmlns="http://www.w3.org/TR/1999/PR-rdf-syntax-19990105#"
    xmlns:PRF="http://www.w3.org/TR/WD-profile-vocabulary#">
  <Bag>
  <Description about="HardwarePlatform">
  <Defaults>
  <Description PRF:Vendor="Nokia"
    PRF:Model="2160"
    PRF:Type="PDA"
```

```

PRF:ScreenSize="800x600x24"
PRF:CPU="PPC"
PRF:Keyboard="Yes"
PRF:Memory="16mB"
PRF:Bluetooth="YES"
PRF:Speaker="Yes" />
</Defaults>
<Modifications>
<Description PRF:Memory="32mB" />
</Modifications>
</Description>
<Description about="SoftwarePlatform">
.....
</RDF>

```

Πίνακας 5 Παράδειγμα CC/PP με Profile και Profile-Diff επικεφαλίδες

2.6.2.3 Η επικεφαλίδα Profile-warning

Ενώ οι προηγούμενες επικεφαλίδες χρησιμοποιούνται κατά την διαδικασία της αίτησης, η Profile-warning επικεφαλίδα χρησιμοποιείται κατά την διαδικασία της απάντησης. Σε περίπτωση που ο server δεν μπορεί να προμηθευτεί την CC/PP πληροφορία που του υπέδειξε η συσκευή τότε επιστρέφει σε αυτήν ανάλογο κωδικό. Ο κωδικός φανερώνει κατά πόσο τα δεδομένα που θα λάβει έχουν τροποποιηθεί ή επιλεγεί σύμφωνα με τις απαιτήσεις.

Η γραμματική του profile-warning και ένα παράδειγμα παρουσιάζεται στον πίνακα 6.

```

Profile-warning          = profile-warning-field-name ":" 1#warning-value
profile-warning-field-name = "Profile-Warning"
warning-value            = warn-code SP warn-target SP warn-text [SP warn-date]
warn-code                = 3DIGIT
warn-target              = (absoluteURI | host [ ":" port ])
warn-text                 = quoted-string
warn-date                 = "<" HTTP-date ">"

```

```

Profile-Warning: 102 http://www.aaa.com/hw "Not used profile",

```

```
202 www.w3.org "Content generation applied"  
Profile-Warning: 101 http://www.aaa.com/hw "Used stale profile",  
102 http://www.bbb.com/sw "Not used profile",  
200 18.23.0.23:80 "Not applied" "Wed, 31 Mar 1999 08:49:37 GMT"
```

Πίνακας 6 Η γραμματική και ένα παράδειγμα της επικεφαλίδας Profile-Warning

Ενδεικτικά μπορούν να αναφερθούν κάποιοι κωδικοί που μπορούν να επιστραφούν από τον server μέσα σε αυτή την επικεφαλίδα.

100 OK: Όλα λειτούργησαν κανονικά και τα κατάλληλα δεδομένα εστάλησαν από τον server

102 Not used Profile: Η CC/PP πληροφορία δεν μπορούσε να βρεθεί από τον server

201 Content Selection Applied: Εφαρμόστηκε η τεχνική της επιλογής των δεδομένων

203 Transformation Applied: Εφαρμόστηκε η τεχνική της μετατροπής των δεδομένων

2.6.3 Πανεπιστήμιο Keio (CC/PP proxy)

Από την στιγμή που εμφανίστηκε το CC/PP δημιουργήθηκαν και πρωτότυπες εφαρμογές που το υποστηρίζουν για να ελέγξουν την λειτουργικότητα του και να βοηθήσουν στην καλύτερη σχεδιάσή του.. Το πανεπιστήμιο του Keio στην Ιαπωνία είναι από τα πρώτα που υλοποίησαν εφαρμογή που να υποστηρίζει το CC/PP [9]. Η εφαρμογή τους στηρίζεται σε έναν proxy server (PandA). Ο server αυτός δέχεται αιτήσεις που περιέχουν και CC/PP πληροφορία από μια δική τους εφαρμογή πλοήγησης που την ονομάζουν SASA. Κατά ένα τρόπο αναγκάστηκαν να υλοποιήσουν και τον browser για να μπορέσουν να δημιουργήσουν αιτήσεις προς διάφορους servers και να μπορέσουν έπειτα να παρουσιάσουν τα αποτελέσματα ικανοποιητικά.

Οι διαθέσιμοι γνωστοί browsers δεν υποστηρίζουν το CC/PP αν και τελευταία έχει υλοποιηθεί μια τέτοια εφαρμογή που υποστηρίζει CC/PP και ονομάζεται X-Smiles αλλά δεν είναι ευρέως διαδεδομένη [19]. Ο X-Smiles browser μπορεί να υποστηρίξει αυτή τη στιγμή XML (Extensible Markup Language) [47], SMIL (Synchronized Multimedia Integration Language)[45] και SVG (Scalable Vector Graphics) [46] μορφές δεδομένων. Η XML χρησιμοποιείται για αναπαράσταση δεδομένων, η SMIL χρησιμοποιείται για εφαρμογές πολυμέσων και η SVG είναι μια καινούργια μορφή για εικόνες.

Σύμφωνα με την υλοποίηση τους η εφαρμογή στέλνει αιτήσεις που περιέχουν CC/PP πληροφορία προς τον proxy, αυτός αναλύει την πληροφορία, διαβάζει τα profiles που

αντιστοιχούν στον χρήστη και επιστρέφει τα ανάλογα δεδομένα. Η πληροφορία αυτή βρίσκεται σε συγκεκριμένο μέρος (CC/PP repository) που υποδεικνύεται από την αίτηση του χρήστη. Όταν τα δεδομένα έρθουν από τον web server και περάσουν από τον proxy αυτά θα μετατραπούν ανάλογα με τις υποδείξεις του profile. Οι πειραματικές μετρήσεις αφορούσαν τον χρόνο που χρειάζεται να μεταφερθεί μια αίτηση που περιέχει CC/PP πληροφορίες (profiles) από την συσκευή του χρήστη στον proxy που διάβαζε αυτή την πληροφορία, για διαφορετικά μεγέθη τέτοιων profiles. Παρατήρησαν πως όταν εφαρμόζεται η τεχνική της προσωρινής αποθήκευσης αυτής της πληροφορίας (caching) σε ενδιάμεσο κόμβο μειώνεται σημαντικά ο χρόνος μεταφοράς. Επίσης προτείνουν την χρήση μέσα στο πρωτόκολλο έμμεσων αναφορών (indirect references) και όχι άμεσων (direct references). Με τον τρόπο αυτό ο web server είναι υπεύθυνος να βρει το profile με βάση την αναφορά που του δίνει η συσκευή μέσα στην HTTP αίτηση. Αυτός σημαίνει πως έχουν ακολουθήσει το HTTP-ext σχήμα [1] αφού η πληροφορία του CC/PP εισάγεται σαν πεδίο μέσα στην επικεφαλίδα της HTTP αίτησης. Αποφεύγεται έτσι η μετάδοση ολόκληρου του profile μέσα από τον υποτιθέμενο μικρής χωρητικότητας σύνδεσμο με τον οποίο συνδέεται η συσκευή στο Διαδίκτυο.

2.6.4 Πανεπιστήμιο της Ουαλίας

Παρόμοια υλοποίηση με αυτή του πανεπιστημίου του Keio έχει γίνει και στην Ουαλία στα πλαίσια διπλωματικής εργασίας [10]. Η διαφορά με αυτή του Keio, όσον αφορά την υλοποίηση είναι πως σε αυτή δεν υπάρχει εφαρμογή πλοήγησης που να στέλνει CC/PP πληροφορία στον web server και υλοποίησαν δυο ενδιάμεσους κόμβους που υποστηρίζουν στην μεταξύ τους επικοινωνία το CC/PP. Ο ένας κόμβος ο οποίος και βρίσκεται κοντά στο μηχάνημα του χρήστη ονομάζεται browser proxy και ο άλλος βρίσκεται πιο κοντά στον server και ονομάζεται server proxy. Ο πρώτος προσθέτει στην αρχική αίτηση τους headers που χρειάζονται για το CC/PP και ο δεύτερος αφαιρεί αυτούς τους headers για να μπορέσει η αίτηση να γίνει κατανοητή από τους web servers. Οι web servers δεν γνωρίζουν την ύπαρξη του CC/PP. Όπως και στην προηγούμενη περίπτωση έτσι και σε αυτή χρησιμοποιήθηκε η HTTP επικεφαλίδα με ένα παραπάνω πεδίο.

Τα αποτελέσματα που παρουσιάζουν στο τέλος είναι μια ιστοσελίδα που μπορεί να παρουσιαστεί με διαφορετικό τρόπο σε έναν φορητό υπολογιστή, σε ένα κινητό τηλέφωνο και σε μια εφαρμογή πλοήγησης που δεν διαθέτει γραφικά (text browser).

2.6.5 DELI: Βιβλιοθήκη για το CC/PP

Μια από τις σημαντικότερες υλοποιήσεις που έχουν γίνει μέχρι στιγμής για το CC/PP είναι αυτή που έγινε από το ερευνητικό κέντρο της Hewlett Packard στο Bristol και ονομάζεται DELI (DELivery context LIbrary) [12]. Σε αυτή την εργασία υλοποιήθηκε μια βιβλιοθήκη λογισμικού σε γλώσσα προγραμματισμού Java που μπορεί να χρησιμοποιηθεί μαζί με τον Tomcat web server [48]. Αυτή η βιβλιοθήκη έχει σαν σκοπό να διαβάζει την πληροφορία που μεταφέρεται και αναφέρεται στο CC/PP. Έχει την δυνατότητα να εξάγει αυτή την πληροφορία σε κατάλληλη μορφή σε άλλες εφαρμογές οι οποίες θα κάνουν και τη τελική μορφοποίηση των δεδομένων. Πρόσφατα ξεκίνησαν οι διαδικασίες ώστε να ενσωματωθεί αυτή η βιβλιοθήκη με το εργαλείο Cocoon [15] που χρησιμοποιείται για κατάλληλη μορφοποίηση ιστοσελίδων. Μετά την συγχώνευση των δυο εργαλείων το Cocoon θα μπορεί να μορφοποιεί ιστοσελίδες με βάση την πληροφορία που θα εξάγεται από το DELI.

2.7 UAProf (User Agent Profile Specification)

Το πρωτόκολλο UAProf [20] έχει αναπτυχθεί από μια ομάδα που έχει σχέση με ασύρματες επικοινωνίες και ονομάζεται WAP forum. Σκοπός του είναι να αποκτήσουν οι κινητές συσκευές ικανοποιητική πρόσβαση σε όλα τα δεδομένα που υπάρχουν αυτή τη στιγμή στον παγκόσμιο ιστό. Οι τεχνικές που χρησιμοποιούνται σε αυτή την περίπτωση είναι το πρωτόκολλο UAProf που ενσωματώνεται μέσα στις επικεφαλίδες του πρωτοκόλλου WAP (Wireless Application Protocol) [20]. Το πρωτόκολλο αυτό μπορεί να μεταφέρει πληροφορίες για το λειτουργικό σύστημα της κινητής συσκευής, για το υλικό το οποίο διαθέτει, τα πρωτόκολλα τα οποία υποστηρίζει και διάφορες άλλες πληροφορίες. Όλες αυτές μεταφέρονται οργανωμένες σε RDF (Resource Description Framework) [6] μορφή. Στο profile αυτό, οι παραπάνω πληροφορίες είναι οργανωμένες σε τμήματα (components) τα οποία με την σειρά τους έχουν κάποια πεδία (attributes). Κάθε ένα από αυτά τα πεδία μπορεί να ανήκει σε ένα και μόνο component. Ακολουθείται δηλαδή μια μορφή ιεραρχικής δομής.

3 Περιγραφή της παρούσης εργασίας

Στα πλαίσια αυτής της εργασίας υλοποιήθηκαν διάφορα συστήματα που σκοπό έχουν να βελτιώσουν την ποιότητα πλοήγησης των χρηστών που χρησιμοποιούν συνδέσμους χαμηλής χωρητικότητας για τη διασύνδεσή τους με το Διαδίκτυο.

3.1 Μηχανή αναζήτησης βιβλίων (E-Book)

Η πρώτη εφαρμογή, που θα παρουσιαστεί στη συνέχεια είναι μια μηχανή αναζήτησης που κάνει χρήση τεχνικών όπως caching και prefetching για γρηγορότερη εξυπηρέτηση των χρηστών της. Η χρήση του caching επιτρέπει στον χρήστη να πλοηγείται στις δυναμικές σελίδες που δημιουργεί το E-Book με αρκετά γρήγορο τρόπο αφού τα αποτελέσματα αυτά μετά την πρώτη αίτηση είναι διαθέσιμα σε μικρό χρόνο σε επόμενες αιτήσεις. Επίσης, χρήστης που θα κάνει την ίδια αίτηση με κάποιον προηγούμενό του θα εξυπηρετηθεί πολύ γρηγορότερα. Σε αυτή την κατεύθυνση κινείται και η τεχνική του prefetching που χρησιμοποιείται στο E-Book. Η τεχνική βασίζεται στο ότι ακολουθούμε εσωτερικό URL της ιστοσελίδας που παρουσιάζεται στον χρήστη αφού είναι αρκετά πιθανό να το ακολουθήσει. Σε περίπτωση που το κάνει τότε η τεχνική απέδωσε, τα αποτελέσματα εμφανίζονται σε μικρό χρόνο και το prefetching συνεχίζεται για το επόμενο URL.

Η εφαρμογή E-Book δεν έχει αυτοσκοπό να εξυπηρετήσει χρήστες που συνδέονται μόνο με μικρούς συνδέσμους στο Διαδίκτυο αλλά όλους τους πιθανούς χρήστες της και θα χρησιμοποιηθεί σαν παράδειγμα για να παρουσιάσουμε χαρακτηριστικά και αποτελέσματα συστήματος που θα παρουσιαστεί στην συνέχεια (HtmlMinimizer).

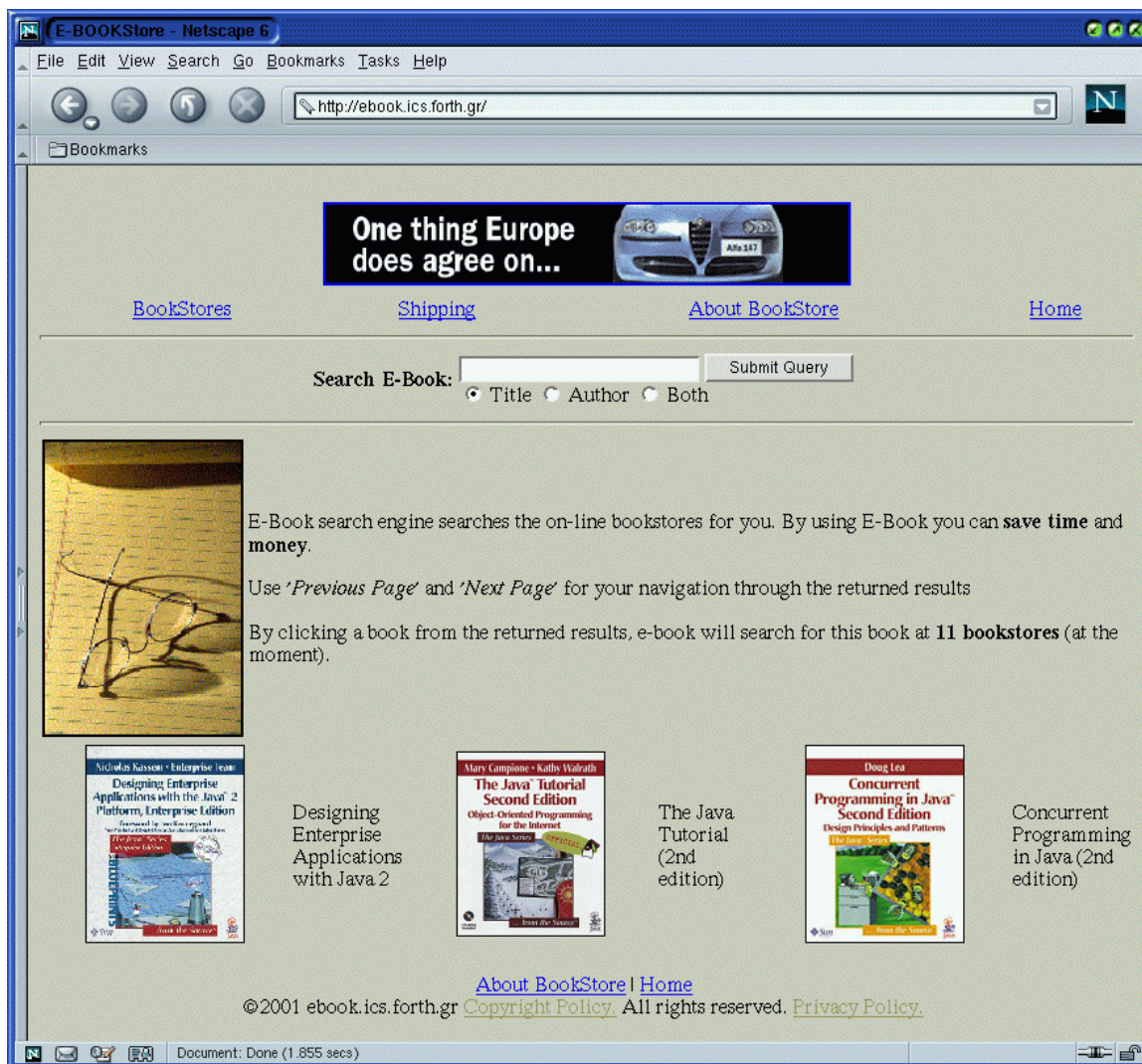
3.1.1 Περιγραφή του E-Book (<http://ebook.ics.forth.gr>)

Το E-Book είναι μια εφαρμογή που κάνει σε λιγότερο χρόνο αυτό που θα έκανε κάποιος χρήστης σε περίπτωση που αποφάσιζε να ψάξει στο Διαδίκτυο για την φθηνότερη τιμή ενός βιβλίου. Το πρόβλημα που είχαμε ως αρχικό κίνητρο για την υλοποίηση του E-Book ίσως να έχει παρουσιαστεί σε αρκετούς κατά καιρούς.

Έστω ότι κάποιος θέλει να ψάξει στο Διαδίκτυο για κάποιο βιβλίο και να αγοράσει φυσικά το φθηνότερο. Μια λύση που έχει είναι να επισκεφτεί όλα τα ηλεκτρονικά βιβλιοπωλεία που γνωρίζει και να ψάξει σε αυτά για το βιβλίο που επιθυμεί. Αφού έχει επισκεφτεί κάποια

βιβλιοπωλεία θα πρέπει μετά να συγκρίνει ποιο από όλα όσα επισκέφτηκε είχε το βιβλίο σε φθηνότερη τιμή. Άρα θα πρέπει να έχει σημειώσει κάπου την τιμή του βιβλίου από κάθε βιβλιοπωλείο που επισκέφτηκε, αλλά και τα έξοδα αποστολής που χρειάζονται για να παραλάβει το βιβλίο. Τα έξοδα αποστολής ποικίλλουν σε κάθε βιβλιοπωλείο και πολλές φορές παίζουν σημαντικό ρόλο στην τιμή. Για να γίνει όλη αυτή η διαδικασία απαιτείται αρκετός χρόνος από πλευράς του χρήστη. Στην περίπτωση όμως που γίνει χρήση του E-Book τότε όλα αυτά γίνονται με τρόπο γρήγορο και αυτοματοποιημένο. Το E-Book γνωρίζει εξ αρχής κάποια γνωστά ηλεκτρονικά βιβλιοπωλεία του Διαδικτύου. Όταν κάποιος το επισκεφτεί και ψάξει για ένα βιβλίο ο μηχανισμός του E-Book θα λειτουργήσει ως εξής:

Αρχικά ο χρήστης θα εισάγει την λέξη κλειδί με την οποία το εργαλείο θα προσπαθήσει να ψάξει το βιβλίο, όπως φαίνεται και στην Εικόνα 1.



Εικόνα 1 Η αρχική σελίδα του E-Book όπου ο χρήστης θα εισάγει την λέξη κλειδί

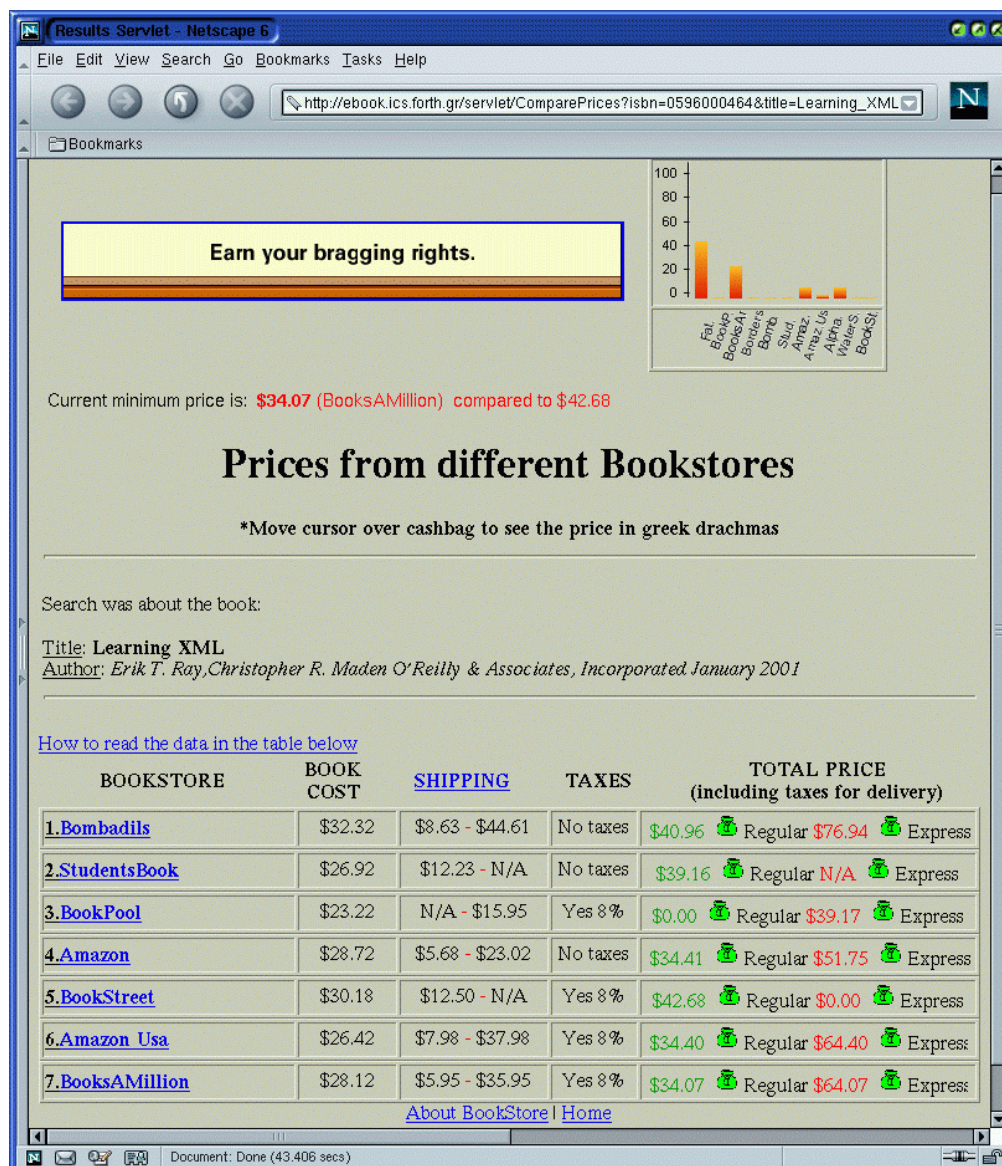
Σε αυτή την εικόνα φαίνονται και οι διαφημίσεις¹ που χρησιμοποιήθηκαν και οι οποίες δεν υπάρχουν στον web server το δικό μας αλλά σε απομακρυσμένους servers. Οι επιλογές που δίνονται στον χρήστη είναι να ψάξει για ένα βιβλίο με βάση τον τίτλο του, τον συγγραφέα του ή και τα δυο μαζί. Αρχικά η αναζήτηση θα γίνει σε ένα μεγάλο και κεντρικό βιβλιοπωλείο. Παλαιότερα χρησιμοποιούσαμε το βιβλιοπωλείο <http://www.fatbrain.com> για αυτή την πρώτη αναζήτηση αλλά τώρα χρησιμοποιούμε το <http://www.barnesandnoble.com> αφού μάλλον υπήρξε συγχώνευση μεταξύ τους. Όταν αυτή η πρώτη αναζήτηση τελειώσει το αποτέλεσμα που θα παρουσιαστεί στην οθόνη του χρήστη θα είναι παρόμοιο με αυτό στην Εικόνα 2.

¹ Για την διαφήμιση της Alfa Romeo δεν παίρνουμε προμήθεια. Ελπίζουμε να μας χαρίσουν ένα κάποτε!©



Εικόνα 2 Με αυτό τον τρόπο παρουσιάζονται τα πρώτα αποτελέσματα της αναζήτησης για ένα βιβλίο σχετικό με XML

Σε αυτό το στάδιο και αν ο χρήστης ενδιαφέρεται για την τιμή κάποιου βιβλίου που είναι στην οθόνη μπορεί να το επιλέξει και η εφαρμογή θα ψάξει σε 11 βιβλιοπωλεία για να παρουσιάσει στον χρήστη την τιμή του σε κάθε ένα από αυτά, αν βέβαια υπάρχει. Έστω ότι το βιβλίο που έψαχνε είναι το πρώτο που φαίνεται στα αποτελέσματα "Learning XML" και το επιλέγει. Μέσα σε μικρό χρονικό διάστημα ο χρήστης θα βλέπει μια προς μια την τιμή από κάθε βιβλιοπωλείο στην οθόνη του και το τελικό αποτέλεσμα θα είναι αυτό που φαίνεται στην Εικόνα 3.

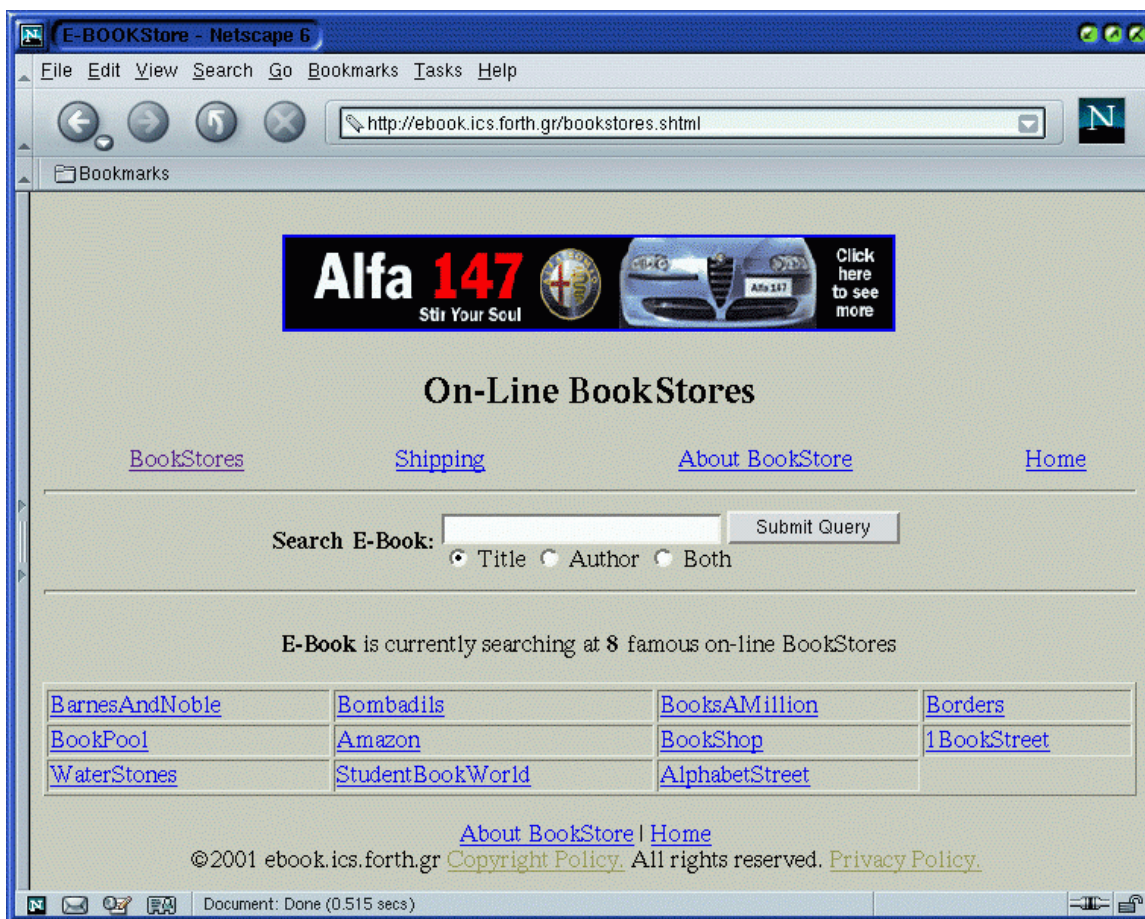


Εικόνα 3 Τα αποτελέσματα για την αναζήτηση τιμών για συγκεκριμένο βιβλίο. Φαίνονται μόνο όσα βιβλιοπωλεία το έχουν διαθέσιμο

Η πληροφορία που διαθέτει αυτή η εικόνα είναι η εξής: Δίνει στον χρήστη κάποια στατιστικά με μορφή ραβδογράμματος τα οποία τα διατηρεί το E-Book. Αυτά τα στατιστικά δείχνουν στον χρήστη ποιο ηλεκτρονικό βιβλιοπωλείο έχει δώσει τις καλύτερες τιμές σαν ποσοστό επί του συνόλου των αναζητήσεων που έχουν γίνει μέχρι εκείνη τη στιγμή στο E-Book. Τα ραβδογράμματα αυτά πρέπει να επισημάνουμε πως δεν δημιουργούνται δυναμικά αφού αυτό θα καθυστερούσε σημαντικά το σύστημα, αλλά υπάρχουν πάρα πολλές μικρές εικόνες στον server. Για κάθε ένα ποσοστό διαλέγει την κατάλληλη. Για την τρέχουσα ελάχιστη τιμή του

βιβλίου όσο το E-Book ψάχνει στα βιβλιοπωλεία ο χρήστης μπορεί να ενημερωθεί είτε από την κόκκινη επικεφαλίδα στην κορυφή της ιστοσελίδας είτε από τον συγκεντρωτικό πίνακα που ακολουθεί. Επίσης θα πρέπει να επισημανθεί πως το εργαλείο έχει σαν στόχο να εξυπηρετεί χρήστες στην Ελλάδα ή άλλα κράτη-μέλη της Ευρωπαϊκής Ένωσης αφού υπολογίζει τις τιμές με 8% φόρο συμπεριλαμβάνοντας και τα έξοδα αποστολής αν το βιβλιοπωλείο είναι εκτός της Ευρωπαϊκής Ένωσης. Αν το βιβλιοπωλείο είναι εντός της Ευρωπαϊκής Ένωσης τότε υπολογίζεται η τιμή με τα έξοδα αποστολής και μπορεί να εμφανιστεί και σε δραχμές².

Όλα τα βιβλιοπωλεία που χρησιμοποιεί το E-Book φαίνονται στον σύνδεσμο “Bookstores” και οι πληροφορίες που δίνουν για τα έξοδα αποστολής στο σύνδεσμο “Shipping”. Τα βιβλιοπωλεία και οι αντίστοιχοι σύνδεσμοι παρουσιάζονται στην παρακάτω Εικόνα 4.



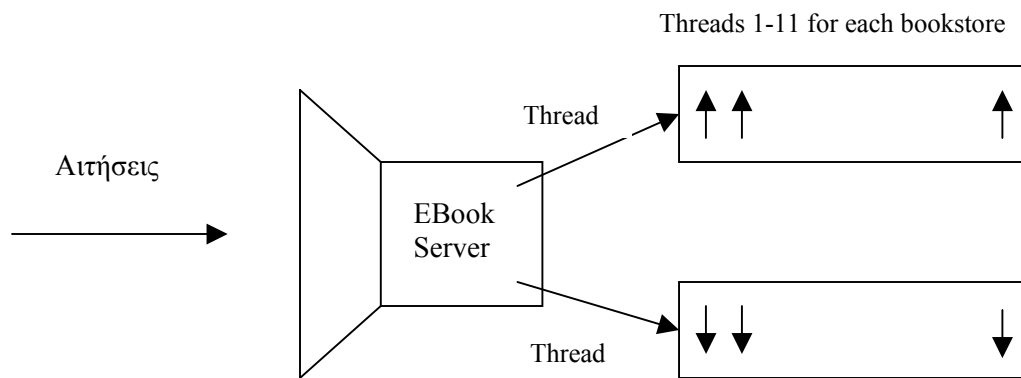
Εικόνα 4 Τα ηλεκτρονικά βιβλιοπωλεία που χρησιμοποιεί το E-Book αυτή τη στιγμή για να ψάχνει τιμές βιβλίων

² Το E-Book υλοποιήθηκε το 2000-2001. Μελλοντική έκδοση θα περιλαμβάνει και μετατροπή σε Euro

Τέλος θα πρέπει να επισημανθεί πως συστήματα όπως το E-Book υπήρχαν ήδη όπως για παράδειγμα το BestBookBuys [21], άλλα δεν αναφέρουν αν χρησιμοποιούν τις τεχνικές που θα περιγραφούν στη συνέχεια.

3.1.2 Αρχιτεκτονική του συστήματος

Το E-Book είναι ένας πολυνηματικός εξυπηρετητής (multithreaded server) όπως είναι οι περισσότεροι αυτή τη στιγμή στο Διαδίκτυο. Αυτή είναι και η λύση που έχει ακολουθηθεί στο παγκόσμιο ιστό για να αποφεύγεται να μονοπωλούνται οι servers από έναν και μόνο χρήστη [26].



Σχήμα 2 Η αρχιτεκτονική του E-Book

Για κάθε αίτηση χρήστη το E-Book δημιουργεί ένα “νήμα” (thread) για να το εξυπηρετήσει. Αυτό το νήμα με την σειρά του θα δημιουργήσει τόσα διαφορετικά νήματα όσα και τα βιβλιοπωλεία που γνωρίζει το E-Book (στην περίπτωση μας αυτά είναι έντεκα.) Άρα για κάθε εισερχόμενη αίτηση το σύστημα δημιουργεί συνολικά δώδεκα νήματα. Το κεντρικό νήμα που τα δημιούργησε περιμένει κάθε ένα από αυτά να επιστρέψουν κάποιο αποτέλεσμα. Σε περίπτωση που κάποιο από αυτά αργήσει αρκετά να επιστρέψει τα αποτελέσματα στο κεντρικό νήμα, είτε γιατί η σύνδεση δεν είναι εφικτή, είτε γιατί υπάρχει πρόβλημα με τον απομακρυσμένο server, τότε το νήμα τερματίζεται και ο χρήστης θα πάρει τα αποτελέσματα όσων επέστρεψαν τιμές μέσα σε χρονικό διάστημα το οποίο έχει καθοριστεί στα 15 δευτερόλεπτα (timeout).

Λόγω της αρχιτεκτονικής του συστήματος και τους τρόπους με τον οποίο υλοποιήθηκε, είναι πολύ εύκολο να προστεθούν και νέα βιβλιοπωλεία τα οποία το E-Book μπορεί να “ρωτά” για τις τιμές των βιβλίων. Αυτό μπορεί να γίνει χρησιμοποιώντας το API που απαιτείται ώστε ο parser που θα επιστρέφει τα αποτελέσματα και πρέπει να υλοποιηθεί για το νέο βιβλιοπωλείο να επικοινωνεί με το βασικό τμήμα του E-Book. Ένας νέος parser μπορεί να υλοποιηθεί αρκετά

γρήγορα ειδικά αν η μορφή των δεδομένων ταιριάζει με την μορφή κάποιου βιβλιοπωλείου που ήδη γνωρίζει το E-Book, οπότε και μπορεί να επαναχρησιμοποιηθεί ο ίδιος κώδικας.

3.1.3 Τεχνικές Caching και Prefetching

Για την καλύτερη εξυπηρέτηση των χρηστών του E-Book έχουν χρησιμοποιηθεί μέθοδοι που αποδίδουν ικανοποιητικά σε παρόμοια συστήματα. Οι τεχνικές αυτές που αναλύθηκαν στην ενότητα **Σχετικές Εργασίες** είναι το caching και το prefetching. Η περίπτωση κατά την οποία εφαρμόζονται αυτές οι τεχνικές στο σύστημα είναι όταν ο χρήστης προσπαθεί να βρει το βιβλίο που τον ενδιαφέρει. Όπως φαίνεται στην Εικόνα 2 όταν ο χρήστης ψάξει για ένα βιβλίο που να έχει σχέση με XML το E-Book θα φέρει τα πρώτα 25 αποτελέσματα (αν υπάρχουν τόσα βέβαια). Τα αποτελέσματα προκύπτουν από ανάλογη αναζήτηση στο ηλεκτρονικό βιβλιοπωλείο BarnesAndNoble. Αυτά τα 25 αποτελέσματα θα αποθηκευθούν προσωρινά (caching) σε δομές που διατηρεί εσωτερικά το E-Book. Μόλις εμφανιστούν όλα τα αποτελέσματα στον χρήστη τότε το E-Book ξεκινά ένα άλλο νήμα το οποίο θα αποθηκεύσει στις εσωτερικές δομές του τα επόμενα 25 αποτελέσματα (prefetching) για την ίδια ακριβώς αναζήτηση (XML). Με αυτό επιτυγχάνουμε να εκμεταλλευτούμε τον χρόνο που ο χρήστης θα καταναλώσει μέχρι να μελετήσει τα πρώτα 25 αποτελέσματα. Η πολιτική που ακολουθείται για την εισαγωγή νέων αποτελεσμάτων και την διαγραφή κάποιων άλλων είναι ένας απλός FIFO (First In First Out) αλγόριθμος.

Με την εφαρμογή των τεχνικών αυτών, ενώ την πρώτη φορά τα αποτελέσματα θα χρειαστούν τόσο χρόνο όσο χρειάζεται για να επικοινωνήσει και να μεταφέρει τα αποτελέσματα το E-Book από το BarnesAndNoble, τα επόμενα αποτελέσματα θα εμφανιστούν στον χρήστη σε πολύ λιγότερο χρόνο αφού θα έχουν ήδη μεταφερθεί. Αυτό βέβαια ισχύει καθ'όλη την διάρκεια της αναζήτησης με την ίδια λέξη κλειδί.

Επίσης, από την στιγμή που κάποια αποτελέσματα βρίσκονται αποθηκευμένα στις δομές του E-Book και κάποιος άλλος χρήστης ψάξει με την ίδια λέξη κλειδί τότε αυτά είναι άμεσα διαθέσιμα.

3.1.4 Τεχνολογίες που χρησιμοποιήθηκαν στο E-Book

Οι τεχνολογίες που χρησιμοποιήθηκαν κατά την υλοποίηση του συστήματος επιτρέπουν γρήγορη ανάπτυξη, επεκτασιμότητα (προσθήκη νέων βιβλιοπωλείων) και ευελιξία σε τυχόν αλλαγές του εργαλείου (πρόσφατα άλλαξε το κεντρικό βιβλιοπωλείο που χρησιμοποιούσαμε από

Fatbrain σε BarnesAndNoble όπως επίσης και η παρουσίαση των δεδομένων που προσφέρουν και χρειάστηκε να γίνουν κάποιες αλλαγές, χωρίς σημαντικό κόπο).

Όλη η εφαρμογή είναι υλοποιημένη στην γλώσσα προγραμματισμού Java [31]. Ένας από τους λόγους που προτιμήθηκε η Java είναι επειδή μας παρέχει αρκετά εργαλεία τα οποία σε μια άλλη γλώσσα προγραμματισμού όπως η C θα ήταν αρκετά χρονοβόρα να υλοποιηθούν. Οφείλουμε όμως να πούμε πως υστερεί σε απόδοση σε σύγκριση με αντίστοιχη έκδοση σε C. Χαρακτηριστικό παράδειγμα ένας αναλυτής HTML κώδικα (HtmlParser). Όλα τα δεδομένα που παίρνει το E-Book από τους servers των βιβλιοπωλείων είναι σε μορφή HTML. Ο κώδικας HTML είναι αυτός από τον οποίο και παράγονται οι ιστοσελίδες που υπάρχουν στο Διαδίκτυο. Με ανάλυση αυτών των δεδομένων το E-Book κρατά μόνο την πληροφορία που χρειάζεται (π.χ ονόματα συγγραφέων, τίτλοι βιβλίων και τιμές βιβλίων) και τα παρουσιάζει στους χρήστες με τον δικό του τρόπο.

Για την παρουσίαση όλων των αποτελεσμάτων έχει χρησιμοποιηθεί η τεχνολογία των servlets [32]. Τα servlets θεωρούνται ως απόγονοι των CGI scripts (Common Gateway Interface). Είναι τμήματα κώδικα γραμμένα σε Java και μπορούν να δημιουργηθούν με αυτά δυναμικές ιστοσελίδες. Είναι μία από τις πιο διαδεδομένες μεθόδους παραγωγής δυναμικών ιστοσελίδων σήμερα και έχουν καλύτερους χρόνους εκτέλεσης σε σχέση με αντίστοιχες εφαρμογές CGI. Για τα servlets χρειάζεται ειδικός μηχανισμός μέσα στον οποίο μπορούν να εκτελεστούν. Για το σκοπό αυτό χρησιμοποιήθηκε ο συνδυασμός του Apache web server με κατάλληλο πρόσθετο τμήμα λογισμικού που μπορεί να εκτελέσει τέτοιες εφαρμογές. Αυτό το πρόσθετο τμήμα λογισμικού είναι το module mod_jserv [33].

Το σενάριο που ακολουθεί μια τυπική αίτηση περιγράφεται στη συνέχεια. Αρχικά η αίτηση πηγαίνει στο web server (Apache). Αν η διεύθυνση αντιστοιχεί σε στατική ιστοσελίδα τύπου HTML, τότε η αίτηση θα εξυπηρετηθεί αποκλειστικά από τον Apache. Αν η αίτηση όμως αντιστοιχεί σε διεύθυνση κάποιου servlet τότε ο Apache θα προωθήσει την αίτηση στο ειδικό τμήμα του που γνωρίζει πως να χειρίζεται τέτοιες αιτήσεις (Apache+JServ) [33]. Τα servlets, στην περίπτωση του E-Book, χρησιμοποιήθηκαν για να παράγουν τον HTML κώδικα και να προωθούν τις παραμέτρους του χρήστη στην μηχανή αναζήτησης της εφαρμογής. Τα αποτελέσματα που θα λάβει ο χρήστης θα είναι μόνο HTML ή τμήματα κώδικα που μπορούν να εκτελεστούν στο μηχάνημα του χρήστη, όπως είναι η JavaScript.

Κατά την διαδικασία της ανάπτυξης του συστήματος χρησιμοποιήθηκε ο Tomcat application server. Αυτός είναι γραμμένος εξ' ολοκλήρου σε Java και ειδικότητά του είναι να εκτελεί τα java servlets. Έχει ικανοποιητική απόδοση στην εκτέλεση των servlets αλλά υστερεί στην απόδοση όταν πρόκειται για στατικές σελίδες σε σχέση με τον Apache. Το υβριδικό μοντέλο Apache για

τις στατικές ιστοσελίδες σε συνεργασία με τον Tomcat για τις δυναμικές σελίδες είναι το καλύτερο αλλά κάναμε την επιλογή Apache + JServ για λόγους καλύτερης διαχείρισης του συστήματος από μέρους μας.

Οι στατικές σελίδες που υπάρχουν στο E-Book όπως θα παρατηρήσει κάποιος που θα το επισκεφτεί έχουν κατάληξη .shtml (server parsed HTML). Αυτό σημαίνει πως έχει ενεργοποιηθεί το tag INCLUDE της HTML που παίρνει παράμετρο ένα άλλο αρχείο συνήθως HTML. Με αυτό το τρόπο κοινά τμήματα των ιστοσελίδων μπορούν να υπάρχουν σε ένα μόνο αρχείο και να μην επαναλαμβάνονται σε όλα.

3.2 Συμπιεστής Ιστοσελίδων (HtmlMinimizer)

Μια μέθοδος που υποστηρίζεται από το πρωτόκολλο HTTP 1.0 και 1.1 είναι η μεταφορά των δεδομένων σε συμπιεσμένη μορφή. Αυτό συνήθως επιτυγχάνεται με την βοήθεια της γνωστής εφαρμογής gzip [34]. Από την στιγμή όμως που θα εφαρμοστεί μια τέτοιου είδους συμπίεση από τον server, θα πρέπει και ο client να διαθέτει το ίδιο ή αντίστοιχο λογισμικό ώστε να επαναφέρει τα δεδομένα στην κανονική τους μορφή. Μια μορφή συμπίεσης που μπορεί να εφαρμοστεί χωρίς να είναι απαραίτητο πρόσθετο λογισμικό στην πλευρά του χρήστη είναι η ελαχιστοποίηση, στο μέτρο του δυνατού της πληροφορίας που υπάρχει στον κώδικα μιας ιστοσελίδας που μεταφέρεται.

Μια τέτοιου είδους ελαχιστοποίηση, βελτιστοποίηση όπως αναφέρεται σε σχετικά άρθρα στο Διαδίκτυο [30], μπορεί να επιτευχθεί για διάφορους λόγους. Θα αναφέρουμε απλώς τους κυριότερους οι οποίοι και μας ώθησαν να υλοποιήσουμε ένα τέτοιο σύστημα..

- Ο HTML κώδικας είναι γραμμένος από προγραμματιστές και προορίζεται για να κατανοηθεί όταν χρειαστεί από προγραμματιστές. Αυτό σημαίνει πως ο κώδικας μιας ιστοσελίδας είναι γραμμένος με αραιό και ευανάγνωστο τρόπο, με αρκετά σχόλια (επεξηγηματικά και μη), και με αρκετούς περιττούς χαρακτήρες
- Πολλές φορές οι ιστοσελίδες δημιουργούνται από εφαρμογές που χρησιμοποιούν σε μεγάλο βαθμό την παραπάνω τακτική.

Επειδή όμως ο κώδικας μιας ιστοσελίδας συνήθως δεν προορίζεται να διαβαστεί από τον μέσο χρήστη, παρά μόνο το αποτέλεσμα που παράγεται από αυτόν, όλα τα παραπάνω μπορούν να παραληφθούν χωρίς να υπάρχει διαφορά στο τελικό αποτέλεσμα της παρουσίασης. Η “συμπίεση” όμως που θα πετυχαίναμε αν αφαιρέσουμε σχόλια και περιττούς χαρακτήρες, όπως θα φανεί από κάποιες μετρήσεις, δεν θα ήταν σημαντική για αυτό προχωρήσαμε ένα βήμα παραπέρα και αποφασίσαμε να αυξήσουμε το ποσοστό της συμπίεσης με την τεχνική της ελαχιστοποίησης του μήκους των διαδικτυακών συνδέσμων (URLs) που υπάρχουν μέσα σε μια ιστοσελίδα.

Κάθε ένας από αυτούς τους συνδέσμους μπορεί να χρειάζεται μέχρι και αρκετές δεκάδες bytes για να αναπαρασταθεί και να υποδείξει στην εφαρμογή πλοήγησης του χρήστη σε ποιο δικτυακό τόπο (site) μπορεί να μετακινηθεί ο χρήστης. Αρκετά sites είναι αρκετά βελτιστοποιημένα αφού έχουν φροντίσει να διατηρούν αυτούς τους συνδέσμους σε μικρό μέγεθος. Έτσι πετυχαίνουν μικρό μέγεθος ιστοσελίδας και κατά συνέπεια μικρότερο χρόνο

μετάδοσης της ιστοσελίδας προς τους χρήστες. (π.χ google, yahoo). Στα περισσότερα sites όμως δεν έχει γίνει τέτοια πρόβλεψη και μεταφέρονται περιττές πληροφορίες.

3.2.1 Περιγραφή του συστήματος

Το λογισμικό που αναπτύξαμε υλοποιεί έναν τέτοιο HtmlMinimizer. Τα χαρακτηριστικά του κώδικα της HTML στα οποία βασιστήκαμε για να ελαχιστοποιήσουμε το μέγεθος μιας ιστοσελίδας είναι τα εξής:

- Όλα τα περιττά κενά “ ” (white spaces)
- Όλους τους tab χαρακτήρες “\t”
- Όλους τους περιττούς χαρακτήρες αλλαγής γραμμής (carriage return, newline) “\r\n”
- Όλα τα σχόλια του κώδικα της HTML <!-- -->
- Όλες τις περιττές αποστρώφους (quotes) “” και τέλος
- Αντικαταστήσαμε όλους τους συνδέσμους με νέους μικρότερου μεγέθους χρησιμοποιώντας χαρακτήρες δικής μας επιλογής και διατηρώντας αυξανόμενη αριθμηση

Σε πολλά από τα παραπάνω αναφέρουμε την λέξη “περιττό” γιατί σε πολλές περιπτώσεις η ύπαρξη ενός από αυτούς τους χαρακτήρες έχει μεγάλη σημασία στην παρουσίαση της ιστοσελίδας.

Χαρακτηριστικά μπορεί να αναφερθεί πως τα σχόλια της HTML δεν πρέπει να αφαιρεθούν όταν περιβάλλουν κώδικα JavaScript ή VBScript. Σε αυτή τη περίπτωση λειτουργούν σαν σχόλια μόνο για τους παλιούς browsers υποδεικνύοντας τους να μην λάβουν υπόψη τους τον κώδικα αυτό. Επίσης δεν πρέπει να αφαιρούνται οι χαρακτήρες αλλαγής γραμμής μέσα από τον κώδικα CSS (Cascade Style Sheets). Τα quotes που αφαιρέσαμε αναφέρεται [30] πως δεν δημιουργούν πρόβλημα τουλάχιστον στην HTML ενώ είναι υποχρεωτικά στην XHTML (Extensible HyperText Markup Language) εκτός και αν περιέχουν αλφαριθμητικό που περιέχει τον κενό χαρακτήρα.

Αν εφαρμοστούν όλα τα παραπάνω στον κώδικα μιας ιστοσελίδας το τελικό αποτέλεσμα που θα δει ο χρήστης θα είναι ακριβώς το ίδιο όπως και χωρίς την συμπίεση. Για να εφαρμοστεί όμως η τελευταία μέθοδος με την σμίκρυνση των συνδέσμων είναι απαραίτητη η παρουσία ενός ενδιάμεσου κόμβου (proxy) οποίος θα κάνει τελικά και την όλη συμπίεση. Ο ενδιάμεσος αυτός κόμβος χρειάζεται για να διατηρεί την αντιστοιχία των αυθεντικών διευθύνσεων που περιέχει η ιστοσελίδα και των μικρότερων σε μέγεθος διευθύνσεων που θα λάβουν οι τελικοί χρήστες. Εδώ

θα πρέπει να αναφερθεί πως δεν λάβαμε υπόψη μας την ύπαρξη κύκλων μεταξύ των διαδικτυακών συνδέσμων (σύνδεσμοι όπου βρίσκονται σε διαφορετικές ιστοσελίδες αλλά “ δείχνουν” στην ίδια ιστοσελίδα. Αυτό δεν δημιουργεί κανένα πρόβλημα αλλά σε μελλοντική έκδοση θα μπορούσε να προβλεφτεί για εξοικονόμηση μνήμης στον proxy που θα χρησιμοποιεί τον HtmlMinimizer. Για παράδειγμα έστω οι υποθετικές διευθύνσεις μέσα στον κώδικα της HTML της ιστοσελίδας του Google.

```
<A HREF="/advanced_search"> “Advanced Search” </A>
```

```
<A HREF="/preferences?hl=en"> “Preferences” </A>
```

```
<A HREF="/language_tools?hl=en"> “Language Tools” </A>
```

τότε οι διευθύνσεις που θα λάβει ο χρήστης θα είναι της μορφής

```
<A HREF=08>“Advanced Search”</A>
```

```
<A HREF=09>“Preferences”</A>
```

```
<A HREF=10>“Language Tools”</A>
```

Η Εικόνα 6 παρουσιάζει καλύτερα αυτούς τους μετασχηματισμούς.

Ο παραπάνω μετασχηματισμός πρέπει να αναφερθεί πως χρησιμοποιείται και στην περίπτωση που έχουμε και εικόνες.

Οι εικόνες για τις οποίες χρησιμοποιείται το tag

```
<img src=./images/logo.gif> </img>
```

αντικαθίστανται και αυτές με διευθύνσεις μικρότερου μεγέθους. Αυτές τις εικόνες όμως δεν τις ζητά ο ίδιος ο χρήστης αλλά ο browser του χρήστη δημιουργώντας αιτήσεις οι οποίες δεν φαίνονται στον ίδιο (sub-requests). Όπως και προηγουμένως οι αιτήσεις αυτές θα εξυπηρετηθούν από τον proxy server που θα κάνει την αντιστοιχία μικρών με μεγάλων διευθύνσεων.

Με τον τρόπο αυτό καταφέρνουμε να κατασκευάσουμε μια ιστοσελίδα από κώδικα μικρότερου μεγέθους ο οποίος όμως παρέχει την ίδια λειτουργικότητα με τον αρχικό.

3.2.2 Τεχνολογίες του Συστήματος

Ο HtmlMinimizer που υλοποιήθηκε βασίζεται σε έναν πολυνηματικό εξυπηρετητή (multithreaded server) όπως αυτός του E-Book που περιγράφηκε προηγουμένως, για να μπορεί να δέχεται πολλαπλές αιτήσεις χρηστών ταυτόχρονα. Επίσης διαθέτει και έναν αναλυτή (HtmlParser) της HTML γλώσσας ώστε να μπορούμε να γνωρίζουμε τι είδους τμήμα της ιστοσελίδας διαβάζουμε κάθε φορά και ανάλογα να επεμβαίνουμε όπου χρειάζεται. Ο κώδικας του HtmlParser έχει δημιουργηθεί από ένα εργαλείο το οποίο χρησιμοποιείται για την κατασκευή της γραμματικής μιας γλώσσας προγραμματισμού ή κάποιας άλλης γραμματικής. Είναι το

ανάλογο του YACC (Yet Another Compiler Compiler) και ονομάζεται JavaCC (Java Compiler Compiler) [35]. Ο κώδικας που δέχεται αυτό το εργαλείο είναι κανονικές εκφράσεις (regular expressions) και οι ενέργειες (actions) που θα εκτελεστούν αν υπάρχει αντιστοιχία της κανονικής έκφρασης με την είσοδο που δέχεται, είναι γραμμένες σε Java. Η γλώσσα προγραμματισμού στην οποία είναι υλοποιημένο το JavaCC είναι η Java και οι αναλυτές (Parsers) που δημιουργεί είναι επίσης σε κώδικα Java. Με τον τρόπο αυτό μπορέσαμε να αφαιρέσουμε όλη την περιττή πληροφορία από μια ιστοσελίδα ικανοποιητικά και γρήγορα αποφεύγοντας την απευθείας υλοποίηση του αναλυτή.

3.2.3 Τεχνικές του πρωτοτύπου

Οι τεχνικές που χρησιμοποιήθηκαν για το πρωτότυπο HtmlMinimizer αποσκοπούσαν στην μελέτη αν μπορεί να εφαρμοστεί κάτι τέτοιο σε ενδιάμεσο κόμβο του Διαδικτύου. Την αφαίρεση των περιττών πληροφοριών τις έκανε ο αναλυτής του HTML κώδικα. Για τις αντικατάσταση των διευθύνσεων από εικόνες και URLs χρησιμοποιήθηκε μια απλή τακτική.

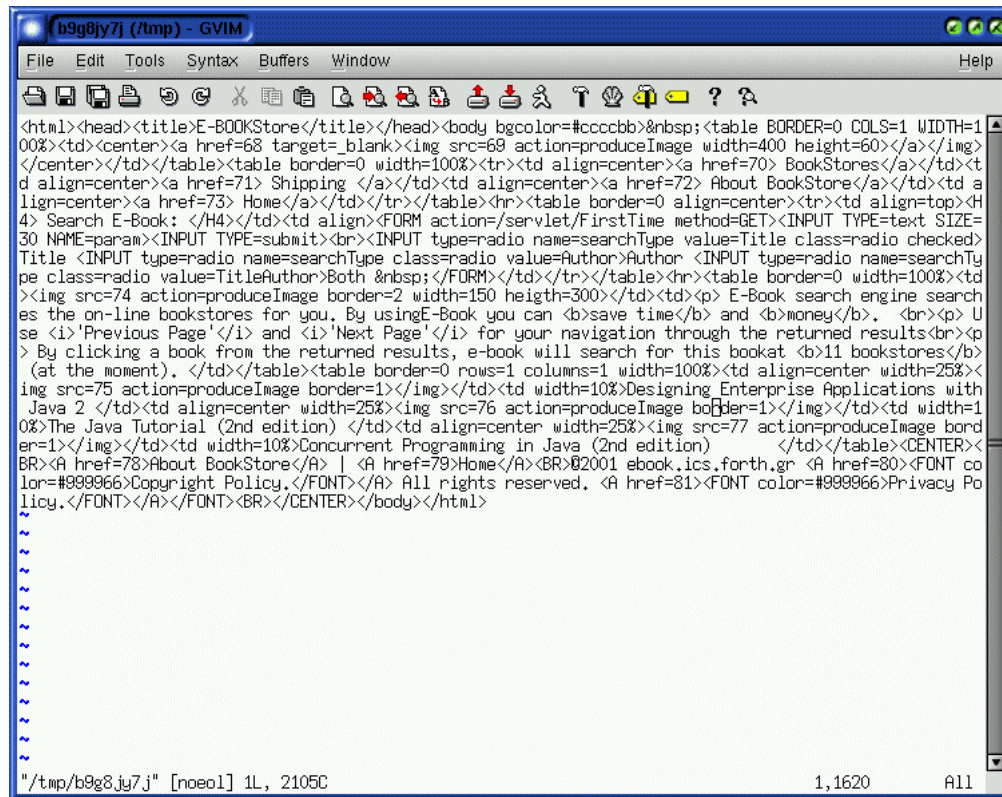
Καθώς αναλύεται ο HTML κώδικας της ιστοσελίδας, όταν εμφανιστεί κάποια διεύθυνση, είτε από URL (tag href) είτε από κάποια εικόνα (tag img), αυτή αντιστοιχίζεται με έναν αριθμό που αυξάνει πάντα κατά 1. Διαθέτουμε δηλαδή έναν απλό μετρητή. Αυτό έγινε για να επιβεβαιώσουμε πως μια κοινή ιστοσελίδα και η επικοινωνία χρήστης ενδιάμεσος κόμβος και server δεν επηρεάζονται στις μετατροπές που γίνονται στον ενδιάμεσο κόμβο. Επίσης η δομή στην οποία διατηρούμε την αντιστοιχία είναι μια απλά συνδεδεμένη λίστα. Οι κόμβοι από τους οποίους αποτελείται αυτή η λίστα είναι δομές που περιέχουν τις αντιστοιχίες για κάθε ιστοσελίδα. Στην Εικόνα 5 φαίνεται η μορφή του κώδικα της HTML πριν την εφαρμογή του HtmlMinimizer. Το τελικό αποτέλεσμα που βλέπει ο χρήστης και στις 2 περιπτώσεις είναι αυτό στην Εικόνα 1

```
<html>
<head>
  <title>
    E-BOOKStore
  </title>
</head>
<body bgcolor=#ccccbb>
  &nbsp;
  <table BORDER=0 COLS=1 WIDTH="100%" >
  <td>
    <center>
      <a href="http://www.alfa147.co.uk" target="_blank"></img>
    </center>
  </td>
</table>

  <table border=0 width=100%>
  <tr>
    <td align=center><a href="bookstores.shtml"> BookStores</a></td>
    <td align=center><a href="shipping.shtml"> Shipping </a></td>
    <td align=center><a href="about.shtml"> About BookStore</a></td>
    <td align=center><a href="index.shtml"> Home</a></td>
  </tr>
</table>
<hr>
<table border=0 align=center>
<tr>
  <td align=top> <H4> Search E-Book: </H4> </td>
  <td align=
    <FORM action=/servlet/FirstTime method=GET>
      <INPUT TYPE=text SIZE=30 NAME=param>
      <INPUT TYPE=submit>
    <br>
"ebook_normal" 94L, 2594C written                                21,58-65    Top
```

Εικόνα 5 Ο πηγαίος HTML κώδικας της αρχικής ιστοσελίδας του E-Book πριν την συμπίεση

Στην Εικόνα 6 παρουσιάζεται η μορφή του HTML κώδικα της σελίδας μετά την εφαρμογή της συμπίεσης.



Εικόνα 6 Ο πηγαίος HTML κώδικας της αρχικής ιστοσελίδας του E-Book μετά την συμπίεση

Όπως φαίνεται και από το κάτω μέρος των εικόνων η μείωση στο μέγεθος της αρχικής σελίδας δεν ήταν πολύ μεγάλη λόγω της λίγης πληροφορίας που διαθέτει και των περιορισμένων URLs. Το αρχικό μέγεθος ήταν 2594 bytes και μετά τη συμπίεση έγινε 2105 bytes. Μεγαλύτερη μείωση όμως πετύχαμε όταν εφαρμόστηκε η μέθοδος συμπίεσης στην ιστοσελίδα των αποτελεσμάτων που από 13503 bytes ελαττώθηκε σε 7805 bytes.

Το συνηθισμένο ποσοστό της συμπίεσης που πετυχαίνει η μέθοδός μας κυμαίνεται ανάμεσα στο 24-30% του μεγέθους της τελικής ιστοσελίδας.

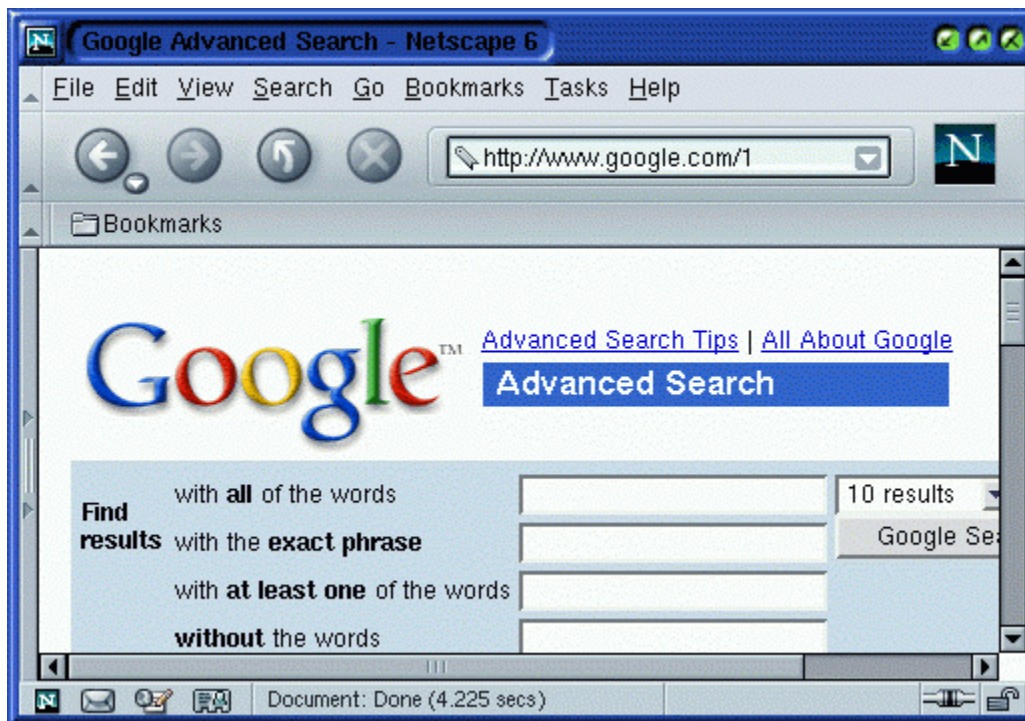
3.2.3.1 Βελτιστοποίηση της υπάρχουσας τεχνικής

Η τεχνική που περιγράψαμε στις προηγούμενες παραγράφους έχει κάποιες αδυναμίες όταν εφαρμοστεί σε έναν proxy server. Οι αδυναμίες αυτές υπάρχουν λόγω του ότι οι ιστοσελίδες μπορούν να αποθηκευθούν στην cache της εφαρμογής του χρήστη. Υπενθυμίζουμε σε αυτό το σημείο πως αρχικά θέλαμε να ελέγξουμε ότι μπορεί να γίνει η πλοήγηση κανονικά ακόμα και με τις νέες διευθύνσεις. Αφού είδαμε ότι κάτι τέτοιο μπορεί να επιτευχθεί προχωρήσαμε στο επόμενο στάδιο. Η συνήθης κατηγορία περιπτώσεων στην οποία ανήκουν οι αιτήσεις προς τον

proxy server (HtmlMinimizer), την οποία και καλύπτουμε πλήρως, είναι οι αιτήσεις που φτάνουν στον proxy server να αντιστοιχούν σε διευθύνσεις για τις οποίες υπάρχουν αντιστοιχίες στην μνήμη του. Δεδομένου όμως πως η μνήμη ενός proxy server θα είναι πάντα πεπερασμένου μεγέθους, θα πρέπει να εφοδιάσουμε το σύστημα με κατάλληλο μηχανισμό για την περίπτωση που λάβει αίτηση για διεύθυνση που δεν έχει αντιστοιχία εκείνη τη στιγμή ενώ παλιότερα είχε.

Μια τέτοια περίπτωση μπορεί να συμβεί όταν στην cache της εφαρμογής τους χρήστη υπάρχει μια παλιά ιστοσελίδα αποθηκευμένη που περιέχει διευθύνσεις όχι πραγματικές αλλά αυτές που έθεσε το σύστημα (HtmlMinimizer) τη στιγμή που ο χρήστης την είχε επισκεφθεί. Όταν ο χρήστης μετά από αρκετό χρονικό διάστημα προσπελάσει αυτή την ιστοσελίδα, η αίτηση που θα φτάσει στον proxy θα είναι ένα και μόνο αναγνωριστικό (ένα νούμερο). Κάποτε ο proxy για αυτό το νούμερο είχε την αντιστοιχία με την πραγματική διεύθυνση στο Διαδίκτυο. Εκείνη τη στιγμή όμως δεν είναι διαθέσιμη μια τέτοια αντιστοιχία ή σε διαφορετική περίπτωση η αντιστοιχία είναι λάθος και αναφέρεται σε μια πιο πρόσφατη σελίδα, αλλά όχι αυτή που θέλει ο χρήστης. Το αποτέλεσμα είναι ο χρήστης είτε να λάβει λάθος αποτελέσματα είτε να μην λάβει κανένα αποτέλεσμα. Το ότι στον proxy η πληροφορία που φτάνει είναι ένα απλό νούμερο έχει να κάνει με το εξής: Οι αιτήσεις που δημιουργούνται κάθε φορά από τον χρήστη είναι συνδυασμός της πληροφορίας που υπάρχει στο location bar της εφαρμογής εκείνη τη στιγμή και της διεύθυνσης που έχει επιλέξει να επισκεφτεί. Ένα σενάριο είναι το παρακάτω:

Έστω πως αρχικά ο χρήστης θα πληκτρολογήσει στο location bar το URL <http://www.google.com>. Η ιστοσελίδα του Google θα περάσει από τον HtmlMinimizer, θα αλλάξουν τα URLs αυτής και θα εμφανιστεί στον χρήστη. Κατόπιν, ο χρήστης επιλέγει το URL που οδηγεί στις οδηγίες για το advanced_search του Google. Η κατάσταση του browser μετά από αυτή την ενέργεια φαίνεται στην Εικόνα 7. Η πληροφορία στο location bar εκείνη τη στιγμή είναι ψευδής. Από την πληροφορία <http://www.google.com/1> το τμήμα που αναφέρει <http://www.google.com> τυχαίνει να συμπίπτει στην συγκεκριμένη περίπτωση με το Google, αφού αντί για το advanced_search θα μπορούσε να έχει επιλεγεί URL που αντιστοιχεί για παράδειγμα στο <http://www.cnn.com>. Άρα η μόνη χρήσιμη πληροφορία είναι το νούμερο '1'. Από αυτή την πληροφορία και μόνο δεν μπορούμε να γνωρίζουμε ανά πάσα στιγμή ποιο είναι το πραγματικό URL στο οποίο αναφέρεται ο χρήστης. Για να μπορέσουμε να το κάνουμε αυτό θα πρέπει ο,τιδήποτε υπάρχει στο location bar να ισχύει και να είναι πραγματικό.



Εικόνα 7 Κατάσταση του browser μετά από επιλογή ενός επεξεργασμένου URL από τον HtmlMinimizer

Το τμήμα αυτό του browser (location bar) μπορεί να το χειριστεί μόνο ο ίδιος ο browser. Κανένας εξωτερικός παράγοντας (π.χ πεδίο της HTTP απάντησης) δεν μπορεί να αλλάξει την τιμή του άμεσα. Υπάρχει η μέθοδος του redirection με την χρήση javascript αλλά δεν μας εξυπηρετεί λόγω της διπλής αίτησης που πρέπει να γίνει προς τον proxy. Μια αίτηση με το κωδικοποιημένο URL ώστε να λάβει από τον proxy το πραγματικό URL και μια αίτηση μετά με το πραγματικό URL.

Ένας απλός και αποτελεσματικός τρόπος για να δέχεται ο proxy αιτήσεις που μπορούν να διακριθούν μοναδικά στο σύνολο των αιτήσεων που μπορεί να δεχτεί είναι η χρήση του BASE πεδίου (tag) της HTML γλώσσας. Σύμφωνα με αυτό όλα τα URLs που δεν ξεκινάνε με το διακριτικό “**http://**” (relative URLs) έχουν σαν πρόθεμα την τιμή του BASE πεδίου. Έτσι σε κάθε ιστοσελίδα που περνά από τον proxy θα εισάγουμε στην σελίδα αυτό το πεδίο με τιμή το ίδιο URL της ιστοσελίδας. Έτσι σε κάθε αίτηση ο proxy θα έχει πληροφορία για μια την ιστοσελίδα πατέρα και το link παιδί που ζήτησε ο χρήστης μέσα σε αυτή. Σε αυτή την περίπτωση το location bar του browser έχει πληροφορία της μορφής

http://www.google.com/advanced_search/AF που σημαίνει πως ο χρήστης εκείνη τη στιγμή ο χρήστης βλέπει το 635ο (σύμφωνα με την δική μας αρίθμηση) URL της ιστοσελίδας advanced_search του Google. Η παραπάνω πληροφορία που προσθέτεται με αυτό τον τρόπο είναι 12 bytes λόγω συντακτικού για το BASE πεδίο και το μέγεθος του URL της ιστοσελίδας.

Η περίπτωση που δεν καλύπτεται από αυτή τη μέθοδο είναι να έχει αλλάξει η θέση του URL μέσα στην σελίδα που φιλοξενείται και να μεταφερθεί λάθος αποτέλεσμα στον χρήστη. Αυτό που μένει να κάνει μετά είναι να πληκτρολογήσει την διεύθυνση που θέλει ώστε να πάρει την σωστή ιστοσελίδα και να ενημερωθεί ταυτόχρονα με αυτό τον τρόπο και ο HtmlMinimizer για την καινούργια θέση του URL μέσα στην ιστοσελίδα..

Σε αυτή την νέα έκδοση του proxy server που θα κάνει την συμπίεση, για την εσωτερική αρίθμηση των URLs γίνεται χρήση της ακόλουθης τακτικής. Για την αρίθμηση των URLs μέσα στις ιστοσελίδες δεν θα διατηρούμε πλέον έναν αυξανόμενο μετρητή αλλά για κάθε μια ιστοσελίδα ο μετρητής θα αρχικοποιείται ξανά στην αρχική του τιμή. Η αρχική τιμή του μετρητή δεν θα είναι το μηδέν και η αρίθμηση δεν θα είναι στο γνωστό δεκαδικό σύστημα. Τα URLs μιας ιστοσελίδας θα κωδικοποιούνται με 2 χαρακτήρες το καθένα. Η τιμή κάθε χαρακτήρα θα μπορεί να κυμαίνεται από

First Char	0-9, A-Z, a-z
Second Char	0-9, A-Z, a-z

Συνολικά οι πιθανές τιμές κάθε χαρακτήρα είναι 62 και όλες οι πιθανές τιμές με τις οποίες μπορεί να αντικατασταθεί η διεύθυνση ενός συνδέσμου είναι 3844. Ίσως να μην υπάρχει ιστοσελίδα αυτή τη στιγμή στο Διαδίκτυο που να πλησιάζει έναν τέτοιο αριθμό από URLs. Στο παράδειγμα που παρουσιάσαμε παραπάνω το URL νούμερο 635 σε μια ιστοσελίδα θα κωδικοποιείται με το αναγνωριστικό AF.

3.2.4 Πειραματικές Μετρήσεις

Για να έχουμε μια καλύτερη εικόνα για το ποσοστό της μείωσης του μεγέθους των ιστοσελίδων που μπορούν να επιτύχουν οι μέθοδοι που εφαρμόσαμε χρησιμοποιήσαμε τα 87 από τα 100 δημοφιλέστερα sites σύμφωνα με το περιοδικό ZDNet [38]. Οι διευθύνσεις των sites είναι διαθέσιμες στο παράρτημα. Οι μετρήσεις που κάναμε ομαδοποιήθηκαν για να εμφανιστεί πια από τις επιμέρους μεθόδους που χρησιμοποιήθηκε συνεισφέρει περισσότερο στην μείωση του μεγέθους των ιστοσελίδων. Συνολικά το μέγεθος των 87 ιστοσελίδων που χρησιμοποιήθηκαν ήταν 1.959.029 bytes.

1. Στην πρώτη μέτρηση που έγινε μελετήσαμε την μείωση που επιτυγχάνεται όταν αφαιρέσουμε από τις ιστοσελίδες τους χαρακτήρες “ ”, “\t” και “\n”.

2. Στην δεύτερη μέτρηση και χρησιμοποιώντας πάλι τις αυθεντικές ιστοσελίδες αφαιρέσαμε τα σχόλια του κώδικα της HTML και
3. στην τρίτη μέτρηση αντικαταστήσαμε τις διευθύνσεις των URLs και των εικόνων με νέες διευθύνσεις 2 χαρακτήρων.
4. Τέλος εφαρμόσαμε όλες τις τεχνικές για να μελετήσουμε την μέγιστη μείωση που μπορούμε να επιτύχουμε.

Ο Πίνακας 7 παρουσιάζει τα αποτελέσματα των πρώτων μετρήσεων.

Συνολικό μέγεθος: 1.959.029 bytes

Τεχνική	Bytes	Διαφορά (Bytes)	Ποσοστό
“ ” “\t” “\n”	1.844.172	114.857	5.86%
Σχόλια HTML	1.864.590	94.439	4.82%
Διευθύνσεις	1.623.878	335.151	17.10%
Τα παραπάνω και Αφαίρεση “\” και alt=””	1.423.373	535.656	27.34%

Πίνακας 7 Μετρήσεις για την μείωση του μεγέθους που επιτυγχάνει ο HtmlMinimizer σε 87 δημοφιλής ιστοσελίδες (Βάθος 0)

Από τα πρώτα αυτά αποτελέσματα παρατηρούμε πως το 5.86% των ιστοσελίδων αποτελείται από περιττά κενά και χαρακτήρες αλλαγής γραμμών και το 4.82% αποτελείται από σχόλια στον HTML κώδικα. Κάνοντας την αντικατάσταση των URLs που περιέχουν αυτές οι ιστοσελίδες με νέα εξοικονομήσαμε 17.1% του συνολικού αρχικού μεγέθους. Εφαρμόζοντας όλες τις τεχνικές μαζί η εξοικονόμηση αυτή αυξήθηκε και έφτασε το 27.34%.

Σημείωση: ένα τμήμα του συνολικού ποσοστού όταν εφαρμόστηκαν όλες οι τεχνικές ανήκει στην αφαίρεση των quotes και των περιττών πεδίων alt=”” για αυτό και το άθροισμα των ποσοστών των μεθόδων δεν ταυτίζεται με το συνολικό.

Εφαρμόσαμε πάνω στα ήδη συμπιεσμένα από τον HtmlMinimizer αποτελέσματα την κλασική συμπίεση (gzip) και μελετήσαμε κατά πόσο μπορεί να αποδώσει πάνω σε αυτά.

Ο Πίνακας 8 παρουσιάζει την περαιτέρω βελτίωση των αποτελεσμάτων με την χρήση του gzip.

Συνολικό μέγεθος: 1.959.029 bytes

Με την εφαρμογή GZIP: 430.934 bytes

Τεχνική	HtmlMinimizer	HtmlMinimizer +(GZIP)	Ποσοστό επί των συμπιεσμένων
“ ” “\t” “\n”	1.844.172	410.872	4.65%
Σχόλια HTML	1.864.590	409.641	4.94%
Διευθύνσεις	1.623.878	391.553	9.13%
Τα παραπάνω και Αφαίρεση “\”” και alt=””	1.423.373	356.712	17.22%

Πίνακας 8 Η απόδοση του gzip πάνω στα ήδη επεξεργασμένα δεδομένα από τον HtmlMinimizer (Βάθος 0)

Παρατηρούμε πως η συμπίεση με το gzip συνεχίζει να αποδίδει κατά ένα μικρό ποσοστό στις επιμέρους τεχνικές και σε σημαντικό ποσοστό στο σύνολό τους. Το άθροισμα των ποσοστών των επιμέρους τεχνικών είναι σχεδόν ίσο με το συνολικό ποσοστό της συμπίεσης.

Στην συνέχεια χρησιμοποιώντας πάλι τις ίδιες ιστοσελίδες εφαρμόσαμε τις παραπάνω τεχνικές και πήραμε μετρήσεις για τις ίδιες αλλά και για τα “παιδιά” αυτών των ιστοσελίδων. Το συνολικό μέγεθος του δείγματος αυτή τη φορά ήταν 60.762.940 bytes.

Οι ανάλογες μετρήσεις φαίνονται στον επόμενο πίνακα.

Συνολικό μέγεθος: 60.762.940 bytes

Τεχνική	Bytes	Διαφορά (Bytes)	Ποσοστό
“ ” “\t” “\n”	56.808.494	3.954.446	6.50%
Σχόλια HTML	58.553.025	2.209.915	3.63%
Διευθύνσεις	50.544.480	10.218.460	16.81%
Συνολικά τα παραπάνω Αφαίρεση “\”” και alt=””	45.235.188	15.527.752	25.55%

Πίνακας 9 Μετρήσεις για την μείωση του μεγέθους που επιτυγχάνει ο HtmlMinimizer σε 87 δημοφιλή ιστοσελίδες και στις θυγατρικές τους (Βάθος 1)

Τα ποσοστά περαιτέρω συμπίεσης με την χρήση του gzip και για αυτό το πείραμα φαίνονται στη συνέχεια.

Συνολικό μέγεθος: 60.762.940 bytes Με την εφαρμογή GZIP: 12.740.784 bytes

Τεχνική	HtmlMinimizer	HtmlMinimizer +(GZIP)	Ποσοστό επί των συμπιεσμένων
“ ” “\t” “\n”	56.808.494	12.140.358	4.71%
Σχόλια HTML	58.553.025	12.225.201	4.04%
Διευθύνσεις	50.544.480	11.678.351	8.33%
Τα παραπάνω και Αφαίρεση “\”” και alt=””	45.235.188	10.815.605	15.11%

Πίνακας 10 Η απόδοση του gzip πάνω στα ήδη επεξεργασμένα δεδομένα από τον HtmlMinimizer (Βάθος 1)

Τέλος στις μετρήσεις συμπεριλάβαμε και τις θυγατρικές ιστοσελίδες σε βάθος 2 για να έχουμε ένα ακόμα μεγαλύτερο δείγμα της επίδοσης του HtmlMinimizer και εφαρμόσαμε και πάλι την μέθοδο gzip για να εμφανιστεί το ποσοστό της περαιτέρω συμπίεσης.

Οι δυο πίνακες που παρουσιάζουν τις τελευταίες αυτές μετρήσεις είναι ο Πίνακας 11 και ο Πίνακας 12.

Συνολικό μέγεθος: 199.563.859 bytes

Τεχνική	Bytes	Διαφορά (Bytes)	Ποσοστό
“ ” “\t” “\n”	188.444.998	11.118.861	5.57%
Σχόλια HTML	191.334.121	8.229.738	4.12%
Διευθύνσεις	171.221.578	28.342.281	14.20%
Συνολικά τα παραπάνω Αφαίρεση “\”” και alt=””	151.075.481	48.488.378	24.29%

Πίνακας 11 Μετρήσεις για την μείωση του μεγέθους που επιτυγχάνει ο HtmlMinimizer στις 87 δημοφιλέστερες ιστοσελίδες και στις θυγατρικές τους (Βάθος 2)

Συνολικό μέγεθος: 199.563.859 bytes Με την εφαρμογή GZIP: 40.216.108 bytes

Τεχνική	HtmlMinimizer	HtmlMinimizer +(GZIP)	Ποσοστό επί των συμπιεσμένων
“ ” “\t” “\n”	188.444.998	38.455.121	4.37%
Σχόλια HTML	191.334.121	37.842.417	5.90%
Διευθύνσεις	171.221.578	37.161.534	7.59%
Τα παραπάνω και Αφαίρεση “\”” και alt=””	151.075.481	33.498.515	16.70%

**Πίνακας 12 Η απόδοση του gzip πάνω στα ήδη επεξεργασμένα δεδομένα από τον HtmlMinimizer
(Βάθος 2)**

Το γενικό συμπέρασμα από τις μετρήσεις που υλοποιήθηκαν θα μπορούσε να είναι πως ο συνδυασμός της δικής μας μεθόδου και της κλασσικής συμπίεσης θα μπορούσαν να εφαρμοστούν ταυτόχρονα για πετύχουμε τη μέγιστη απόδοση. Επίσης, συνεργασία των παραπάνω μεθόδων με την μέθοδο του delta encoding ίσως να μπορούσε να αυξήσει ακόμα περισσότερο την συμπίεση αφού η βελτίωση που επιτυγχάνει αυτή η μέθοδος δεν συμπεριλαμβάνεται ούτε από τον HtmlMinimizer, ούτε και από το κλασσικό gzip. Θα πρέπει όμως ο HtmlMinimizer να είναι υλοποιημένος κατάλληλα ώστε να μπορεί να κάνει την επεξεργασία των ιστοσελίδων σε πολύ μικρό χρόνο. Η δική μας πρωτότυπη υλοποίηση επειδή είναι σε Java δεν θα έχει καλή απόδοση σε αυτό το τομέα, ειδικά αν πρόκειται να χρησιμοποιηθεί ως proxy για μεγάλο αριθμό χρηστών. Ωστόσο η απόδοση μπορεί να βελτιωθεί σημαντικά αν η υλοποίηση γίνει με μια ταχύτερη γλώσσα προγραμματισμού.

3.3 Το πρωτόκολλο CC/PP (Composite Capability/Preference Profiles)

Το Διαδίκτυο αυτή τη στιγμή διαθέτει τεράστια ποσά δεδομένων τα οποία έχουν αρχίσει εδώ και καιρό να είναι προσπελάσιμα από διάφορες συσκευές. Πολλές ιστοσελίδες στο Διαδίκτυο μέχρι σήμερα είχαν υλοποιηθεί με εξασφαλισμένη την προϋπόθεση πως ο χρήστης που θα τις ζητήσει θα διαθέτει έναν παραδοσιακό υπολογιστή με μεγάλη οθόνη και αρκετή υπολογιστική ισχύ. Το CC/PP [4] είναι ένα σχετικά νέο πρωτόκολλο το οποίο θα προσπαθήσει να δώσει στον server μια σαφή εικόνα για το πόσο ικανή είναι η συσκευή με την οποία ο χρήστης πλοηγείται στο Διαδίκτυο. Αυτό δημιουργεί μια βάση πάνω στην οποία εξυπηρετητής και εξυπηρετούμενος μπορούν να διαπραγματευτούν για τα δεδομένα που θα μεταφερθούν.

Το CC/PP έχει δημιουργηθεί από τον οργανισμό W3C και προάγεται από αυτόν. Υπάρχουν και άλλες ομάδες που προσπαθούν να υλοποιήσουν την ίδια λειτουργικότητα. Μια τέτοια ομάδα είναι η IETF Content Negotiation (CONNEG) [39]. Η δική τους υλοποίηση και το CC/PP δεν είναι συμβατές αλλά προσπάθειες γίνονται ώστε να υπάρξει συνεργασία μεταξύ αυτών των δυο υλοποιήσεων.

Το CC/PP είναι σχεδιασμένο για να μπορεί να συνεργαστεί με διάφορες συσκευές, από PDAs μέχρι κανονικούς υπολογιστές, φορητούς υπολογιστές, κινητά τηλέφωνα και τηλεοράσεις συνδεδεμένες στο Διαδίκτυο. Δίνει την δυνατότητα στους servers του Διαδικτύου να ενημερώνονται για τα χαρακτηριστικά της συσκευής του χρήστη από τον ίδιο κατασκευαστή του υλικού της συσκευής και τον κατασκευαστή του λογισμικού της συσκευής απευθείας από το Διαδίκτυο χωρίς μεγάλη επιβάρυνση της ίδιας της συσκευής. Αυτό το χαρακτηριστικό είναι αρκετά ανακουφιστικό ειδικά για τις κυψελοειδείς τεχνολογίες που κάνουν χρήση περιορισμένης χωρητικότητας (limited-bandwidth).

Το CC/PP παρέχει μια περιγραφή των δυνατοτήτων που έχει η εφαρμογή πλοήγησης του χρήστη και η συσκευή γενικότερα. Η πληροφορία αυτή θα είναι χρήσιμη στους servers του Διαδικτύου είτε σαν τμήμα του γνωστού HTTP πρωτοκόλλου, είτε σαν τμήμα κάποιου άλλου πρωτοκόλλου. Είναι θέμα του κάθε server του Διαδικτύου να αναλύσει αυτή την πληροφορία και να επιστρέψει τα κατάλληλα δεδομένα. Δυο από τις κύριες μεθόδους που μπορεί να χρησιμοποιήσει ένας server για να εξυπηρετήσει καλύτερα μια συσκευή είναι η επιλογή (Selection) των δεδομένων και η μετατροπή (Transformation) αυτών ώστε να συμβαδίζουν με τις απαιτήσεις της περιγραφής.

3.3.1 Επιλογή ή Μετατροπή των δεδομένων

Επιλογή είναι η διαδικασία σύμφωνα με την οποία ο server διαλέγει την κατάλληλη παρουσίαση των δεδομένων για συγκεκριμένη αίτηση από ένα σύνολο διαθέσιμων επιλογών. Κάποιος server μπορεί να έχει τρεις διαφορετικές εκδόσεις παρουσίασης της ίδιας ιστοσελίδας. Μπορεί να έχει μια πλούσια παρουσίαση σε XHTML (eXtensible HTML) με Java και αρκετό κώδικα σε scripts για εφαρμογές με δυνατότητες απεικόνισης γραφικών. Μπορεί να διαθέτει μια έκδοση με XHTML που αποτελείται μόνο από απλό κείμενο για εφαρμογές που δεν δέχονται γραφικά και τέλος την ίδια ιστοσελίδα σε έκδοση WML (Wireless Markup Language) για να μπορεί να εξυπηρετεί και κινητές συσκευές. Από τις δυνατότητες και τις προτιμήσεις που έχουν περιγραφεί στο CC/PP ο server θα διαλέξει όποια από αυτές νομίζει ότι ταιριάζει περισσότερο.

Από την άλλη πλευρά υπάρχει η τεχνική της μετατροπής. Στην περίπτωση αυτή δεν υπάρχει η δυνατότητα επιλογής από υπάρχουσες εκδόσεις αλλά η παρουσίαση και τα δεδομένα που θα αποσταλούν δημιουργούνται εκείνη τη στιγμή σύμφωνα πάντα με την περιγραφή του χρήστη. Τα δεδομένα σε αυτή την περίπτωση μπορούν να διατηρούνται σε XML (Extensible Markup Language) μορφή και να μετατρέπονται στην κατάλληλη γλώσσα (XHTML, WML κ.ά) πάντα κατά περίπτωση. Σε διαφορετική περίπτωση, μια εικόνα για παράδειγμα που είναι σε JPEG μορφή θα μετατρέπεται σε κάποια άλλη μορφή (π.χ GIF) και θα στέλνεται στον παραλήπτη.

3.3.2 Ο Apache web server και το CC/PP πρωτόκολλο

Το CC/PP πρωτόκολλο όπως αναφέρθηκε, θα πρέπει να είναι γνωστό σε clients, servers και ενδιάμεσους proxies. Οι υλοποιήσεις σε CC/PP όλο και πληθαίνουν. Πρόσφατα ενσωματώθηκε μια βιβλιοθήκη η DELI [12] στην εφαρμογή Cocoon η οποία θα γνωστοποιεί την CC/PP πληροφορία στο εργαλείο αυτό ώστε να δημιουργούνται δυναμικά κατάλληλα διαμορφωμένες σελίδες για την κάθε συσκευή. Το CC/PP πρωτόκολλο μας έδωσε την ευκαιρία να συμμετέχουμε στο Device Independence Workshop του W3C [8] με την αναφορά (A CC/PP aware Apache Web Server) που υπάρχει στο Παράρτημα Α

Στις επόμενες ενότητες θα περιγραφεί η υλοποίηση που έγινε ώστε να μπορέσει ο Apache server [14] να μπορεί να λαμβάνει CC/PP πληροφορία και ανάλογα να επιστρέφει κατάλληλα δεδομένα. Η μέθοδος που ακολουθήσαμε για την υλοποίηση ήταν να ενσωματώσουμε την πληροφορία του CC/PP μέσα στο HTTP όπως ακριβώς περιγράψαμε στην προηγούμενη ενότητα ακολουθώντας το HTTP-ext μοντέλο.

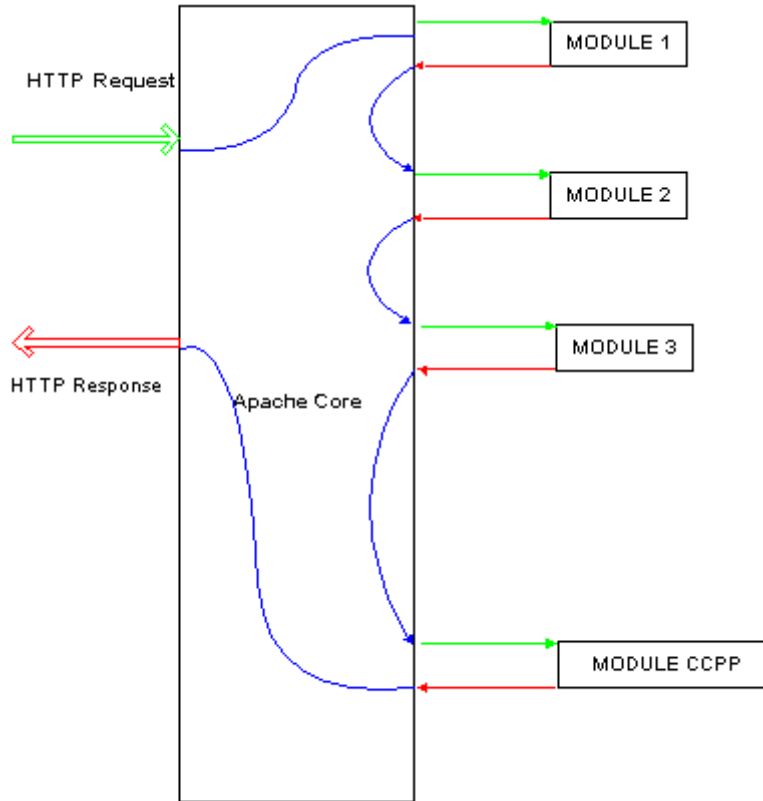
3.3.2.1 Ο Apache Web Server

Ο Apache web server είναι ο δημοφιλέστερος server αυτή τη στιγμή στο Διαδίκτυο. Το 54% των sites τον χρησιμοποιούν για να μπορέσουν να διαθέσουν τις ιστοσελίδες με τα δεδομένα τους στους χρήστες του Διαδικτύου [41]. Αυτό οφείλεται στο ότι είναι αρκετά απλός, γρήγορος και το λογισμικό του είναι διαθέσιμο στο ευρύ κοινό. Επίσης, η προγραμματιστική διεπαφή (API: Application Programming Interface) την οποία παρέχει ο πυρήνας που αποτελεί τον server είναι πολύ εύχρηστη. Οποιοσδήποτε θέλει να προσθέσει κάποια παραπάνω λειτουργικότητα σε αυτή που ήδη υπάρχει, μπορεί να το κάνει μέσω αυτού του API. Με το να είναι διαθέσιμος ο κώδικας πολλοί που τον επεξεργάστηκαν, τον επέκτειναν και συνεισέφεραν στην ανάπτυξή του. Όλα αυτά μας επηρέασαν και αποφασίσαμε να γράψουμε κώδικα για τον Apache ώστε να μπορεί να δέχεται και να επεξεργάζεται την CC/PP πληροφορία που θα στείλει μαζί με την αίτησή της κάποια συσκευή.

Το λογισμικό που μπορεί να προστεθεί στον Apache και δεν επηρεάζει τον κεντρικό πυρήνα του (core) ονομάζεται module και μπορεί να αφαιρεθεί και να προστεθεί δυναμικά από αυτόν. Το είδος της επεξεργασίας που του προστέθηκε πάνω στα δεδομένα που διατηρεί τον καθιστά αμέσως έναν transcoding server ή proxy, ανάλογα με το πως έχει ρυθμιστεί να λειτουργεί κάθε φορά.

3.3.2.2 CC/PP στον Server

Η πρώτη υλοποίηση που θα περιγραφεί έχει να κάνει με την ενσωμάτωση του CC/PP πρωτοκόλλου στον Apache όταν αυτός λειτουργεί σαν server. Ο τρόπος για να γίνει κάτι τέτοιο ήταν να γραφτεί κατάλληλο λογισμικό που μπορεί να επικοινωνήσει με τον πυρήνα του Apache και να πάρει πληροφορίες για τις αιτήσεις που φτάνουν σε αυτόν. Η αρχιτεκτονική του Apache φαίνεται στην Εικόνα 8

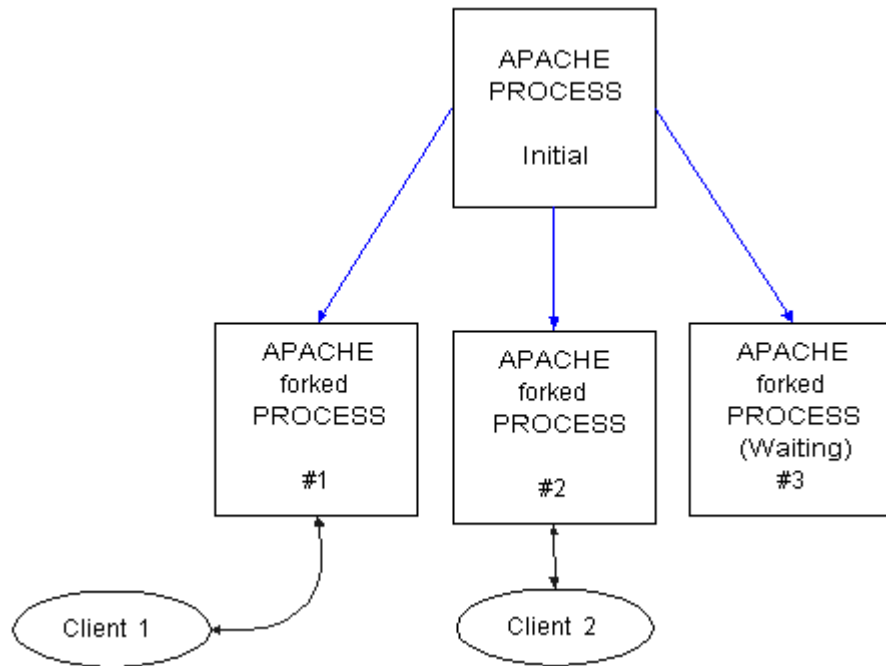


Εικόνα 8 Η αρχιτεκτονική του Apache server

Αποτελείται από τον πυρήνα και τα modules. Τα modules είναι εξειδικευμένα τμήματα λογισμικού που εκτελούν μια συγκεκριμένη εργασία χωρίς να μπορούν να επικοινωνήσουν μεταξύ τους ή να επηρεάζουν το ένα το άλλο. Ο μηχανισμός των modules εξασφαλίζει στον Apache εύκολη επεκτασιμότητα του συστήματος, μεγάλη ευελιξία με την προσθήκη νέων, ασφαλή επικοινωνία και προστασία των δεδομένων που ανταλλάσσονται. Αυτές οι δομές αναφέρονται ρητά στην διεπαφή (interface) που προσφέρουν και είναι κυρίως η δομή στην οποία αποθηκεύονται τα στοιχεία της αίτησης. Στην εικόνα φαίνεται η σειρά με την οποία εκτελούνται τα modules του Apache και το νέο module που προσθέσαμε για την διαχείριση του CC/PP. Κάθε module έχει τρεις επιλογές για κάθε εισερχόμενη αίτηση.

- Την επεξεργάζεται πλήρως όποτε και επιστρέφει στον πυρήνα του Apache χωρίς να περάσει την αίτηση στα επόμενα modules (return code=DONE).
- Την επεξεργάζεται μερικώς και την παραδίδει και στο επόμενο module (return CODE=OK)
- Δεν την επεξεργάζεται καθόλου και την παραδίδει στα επόμενα modules (return code=DECLINED)

Ο Apache όταν ξεκινά δημιουργεί αρχικά μια ομάδα διεργασιών κάνοντας χρήση του system call fork()³. Με αυτό τον τρόπο η αρχική διεργασία δημιουργεί κάποια αντίγραφα της. Αν αυτές οι πρώτες διεργασίες δεν φτάνουν για να εξυπηρετήσουν τις εισερχόμενες αιτήσεις τότε δημιουργούνται και νέες διεργασίες. Το μέγιστο πλήθος διεργασιών του Apache που μπορούν να τρέχουν ταυτόχρονα σε ένα μηχάνημα μπορεί να οριστεί από τον διαχειριστή του κατά την εκκίνηση του web server. Στην Εικόνα 9 φαίνεται ο τρόπος με τον οποίο δημιουργούνται οι Apache διεργασίες.



Εικόνα 9 Ο τρόπος με τον οποίο δημιουργούνται οι Apache διεργασίες από την αρχική

Τα μαύρα βέλη αναφέρονται σε HTTP συνδέσεις και τα γαλάζια βέλη παριστάνουν την δημιουργία διεργασιών με την χρήση fork. Κάθε μια από τις διεργασίες θα εκτελέσει με την σειρά τα modules που διαχειρίζεται ο Apache για την συγκεκριμένη αίτηση που εξυπηρετεί.

Το module που υλοποιήθηκε είχε σαν σκοπό να διαβάζει την πληροφορία που υπάρχει στις Profile και Profile-Diff επικεφαλίδες όπως αυτές ορίζονται στο πρωτόκολλο CC/PP. Σε περίπτωση που δεν υπάρχουν αυτές, η κανονική λειτουργία του Apache δεν επηρεάζεται καθόλου. Σε περίπτωση που υπάρχουν όμως, το module αυτό είναι ικανό να συνδεθεί σε οποιονδήποτε server υποδείξουν οι έμμεσες αναφορές (indirect references) που περιέχονται στην Profile επικεφαλίδα. Η σύνδεση γίνεται με την χρήση socket μέσα από τον Apache. Όταν μεταφερθούν τα δεδομένα τότε χρησιμοποιείται ο XML αναλυτής που υπάρχει διαθέσιμος και

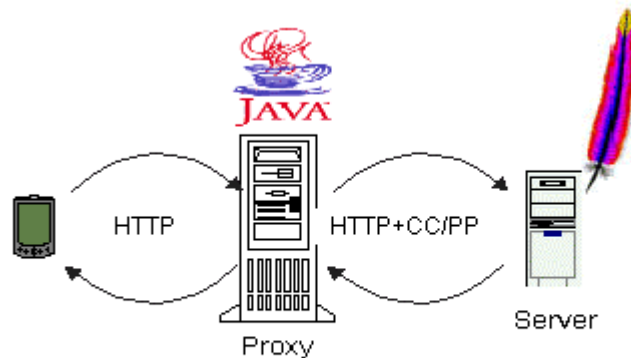
³ Η νέα έκδοση που ακόμα δεν έχει πάρει την τελική της μορφή υποστηρίζει threads

είναι ενσωματωμένος με τον Apache για να μπορέσει να εξαχθεί η πληροφορία. Ο αναλυτής (σε μορφή βιβλιοθήκης) ονομάζεται *expat*.

Χρησιμοποιήσαμε XML αναλυτή και όχι RDF αναλυτή γιατί ήταν άμεσα διαθέσιμος και μπορούσε να μας δώσει ακριβώς τα ίδια αποτελέσματα. Δεν υπήρχε ιδιαίτερο νόημα να ελέγξουμε αυστηρά το Profile αν ακολουθεί τους όρους και τις προδιαγραφές της RDF με έναν RDF validator και μετά να χρησιμοποιήσουμε έναν RDF αναλυτή για να εξαγάγουμε την πληροφορία. Αυτή είναι η μέθοδος που προτείνεται ότι πρέπει να ακολουθηθεί στο μέλλον, αν το CC/PP χρησιμοποιηθεί. Εδώ θα πρέπει να σημειωθεί πως η επεξεργασία της RDF είναι 5-20 φορές πιο αργή σε σχέση με την XML και υπάρχουν έντονες διαμάχες σε e-mail λίστες του W3C για το αν πρέπει να χρησιμοποιηθεί ο ένας ή ο άλλος τρόπος.

Από την στιγμή που ο Apache μπορούσε να επεξεργάζεται την CC/PP πληροφορία που δεχόταν, έπρεπε κάποιος να του στείλει αιτήσεις που να περιέχουν υποτιθέμενα Profiles συσκευών. Οι γνωστές εφαρμογές πλοήγησης Netscape και Internet Explorer δεν παρέχουν πρόσβαση στον κώδικα τους με αποτέλεσμα να μην μπορούμε να αλλάξουμε την δομή των αιτήσεων που στέλνουν και να προσθέσουμε τα επιπλέον πεδία στην επικεφαλίδα του HTTP. (Έχει ξεκινήσει όμως η μελέτη για το αν μπορούμε να προσθέσουμε πεδία στην αίτηση που στέλνουν τα εργαλεία χρησιμοποιώντας την διεπαφή που προσφέρουν).

Για το σκοπό αυτό χρησιμοποιήσαμε έναν ενδιάμεσο proxy που έκανε την εξής διαδικασία για κάθε αίτηση που δεχόταν. Διάβαζε όλη την HTTP επικεφαλίδα της αίτησης και πρόσθετε στο τέλος ένα Profile που του είχαμε υποδείξει. Με αυτό το τρόπο όλες οι αιτήσεις που φτάνανε στον Apache με το κατάλληλο CC/PP module διέθεταν την απαραίτητη πληροφορία. Τα συστατικά στοιχεία από τα οποία αποτελείται αυτό το σύστημα φαίνονται στην Εικόνα 10

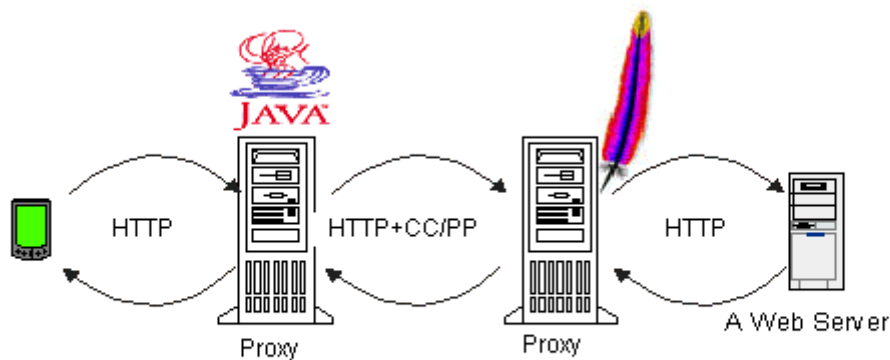


Εικόνα 10 Η μεταφορά της CC/PP πληροφορίας μέσω του Java Proxy

3.3.2.3 CC/PP στον Proxy

Ο Apache της προηγούμενης ενότητας δουλεύει σαν server. Μπορεί δηλαδή να επεξεργαστεί αιτήσεις που απευθύνονται αποκλειστικά και μόνο για τον συγκεκριμένο server. Όπως αναφέρθηκε όμως το CC/PP θα πρέπει να μπορεί να είναι γνωστό και στους ενδιάμεσους κόμβους του Διαδικτύου (proxies). Για να υλοποιήσουμε κάτι τέτοιο χρησιμοποιήσαμε και πάλι τον Apache αφού τον ρυθμίσαμε κατάλληλα κατά την εγκατάστασή του ώστε αυτός να συμπεριφέρεται πλέον ως proxy και όχι ως server. Για να το επιτύχουμε αυτό ενεργοποιήσαμε τμήματα λογισμικού (modules) που παρέχονται και μετατρέπουν τον Apache σε έναν caching proxy.

Για να μπορέσει αυτός ο proxy να διαβάσει και να επεξεργαστεί HTTP επικεφαλίδες που παρέχουν CC/PP πληροφορία καλέσαμε κατάλληλα τον κώδικα του module που χειρίζεται το CC/PP μέσα από το proxy module. Έχοντας διαμορφώσει τον Apache να λειτουργεί με αυτό τον τρόπο μπορούμε να επισκεφτούμε τις σελίδες του Διαδικτύου και ανάλογα με την πληροφορία που περιέχει το CC/PP να παίρνονται κάποιες αποφάσεις (επιλογή ή μετατροπή) μέσα στον proxy. Το σύστημα σε αυτή τη περίπτωση είναι όπως παρουσιάζεται στην Εικόνα 11



Εικόνα 11 Η μεταφορά της CC/PP πληροφορίας μέσω του Java Proxy και του Apache Proxy Server

3.3.3 Μετατροπή επιλεγμένων δεδομένων (Transcoding)

Για να μπορέσουμε να ελέγξουμε την λειτουργικότητα του CC/PP πρωτοκόλλου και πως πράγματι μπορεί να εξυπηρετήσει συσκευές με περιορισμένες απαιτήσεις, αφού αυτός είναι από τους βασικούς στόχους του, μετατρέψαμε τις δυο υλοποιήσεις που περιγράφηκαν στην προηγούμενη ενότητα σε transcoding εφαρμογές. Επιλέξαμε δηλαδή να μετατρέπουμε κάποια δεδομένα σύμφωνα με την περιγραφή του CC/PP. Τα δεδομένα που επιλέχθηκαν για αυτό το σκοπό είναι οι εικόνες μορφής GIF, τα αρχεία ήχου μορφής MP3 και τα αρχεία video μορφής MPEG. Τις εικόνες GIF τις μετατρέπουμε ως προς το μέγεθός τους, ενώ τα αρχεία ήχου και video ως προς τον ρυθμό στον οποίο έχει γίνει η κωδικοποίησή τους. Επιπλέον στο video μπορούμε να αφαιρούμε τον ήχο και να μικραίνουμε το μέγεθος του παραθύρου που χρειάζονται για να παρουσιαστούν.

3.3.3.1 Transcoding στον Apache server

Σε αυτή την περίπτωση τα δεδομένα υπόκεινται σε διάφορες μετατροπές μέσα στον ίδιο τον server ανάλογα με το Profile που στέλνει ή υποδεικνύει η συσκευή που κάνει την αίτηση. Την πρώτη φορά που θα ζητηθεί κάποιο αρχείο που ανήκει σε μια από τις παραπάνω μορφές και εφόσον μεταφερθεί CC/PP πληροφορία με κάποια περιγραφή για χαμηλή χωρητικότητα και μικρή οθόνη τότε τα δεδομένα μπορούν να μετατραπούν στην ίδια μορφή αλλά με λιγότερη πληροφορία. Λιγότερη πληροφορία σημαίνει πως το τελικό αποτέλεσμα δεν θα έχει το ίδιο μέγεθος με το αρχικό, είτε γιατί θα έχει χαμηλότερη ποιότητα, είτε μικρότερο μέγεθος εάν είναι εικόνα ή video. Την επόμενη φορά όμως που θα υπάρξει ανάλογη αίτηση και ανάλογη περιγραφή για το ίδιο αρχείο θα μεταδοθεί αμέσως το αρχείο το οποίο θα έχει ήδη μετατραπεί. Από αυτό φαίνεται καθαρά πως η πρώτη αίτηση θα καθυστερήσει αφού μεσολαβεί ο χρόνος της μετατροπής σε σχέση με τις επόμενες που θα εξυπηρετηθούν κανονικά και γρηγορότερα λόγω μικρότερου μεγέθους του νέου αρχείου. Το νέο αρχείο που δημιουργείται στον server έχει κατάλληλο όνομα που φανερώνει τα χαρακτηριστικά της μετατροπής που υπέστη για να μπορεί να είναι εύκολα αναγνωρίσιμο.

3.3.3.2 Transcoding στον Apache proxy

Παρόμοια λειτουργικότητα δώσαμε και στην υλοποίηση κατά την οποία ο Apache λειτουργεί σαν proxy. Σε αυτή την περίπτωση, αν στην αίτηση υπάρχει κατάλληλη πληροφορία CC/PP και το αρχείο που ζητείται από την συσκευή του χρήστη ανήκει σε μια από τις μορφές

που προαναφέραμε τότε θα μετατραπεί και θα αποθηκευτεί προσωρινά στον proxy server. Το όνομα που θα του δοθεί θα είναι τέτοιο ώστε να φανερώνει τα χαρακτηριστικά της μετατροπής όπως προηγουμένως αλλά και τον πραγματικό server από τον οποίο προέρχεται. Οι επόμενες αιτήσεις που θα ζητήσουν αρχείο που υπάρχει ήδη επεξεργασμένο και αποθηκευμένο στον proxy θα το λάβουν απευθείας από αυτόν.

3.3.3.3 Οι βιβλιοθήκες λογισμικού ImageMagick και FFmpeg

Για την επεξεργασία των αρχείων που μετατρέπονται στις περιπτώσεις των transcoding εφαρμογών, χρησιμοποιήθηκαν δυο εργαλεία με αρκετά μεγάλες δυνατότητες. Τα εργαλεία αυτά καλούνται από το module που χειρίζεται το CC/PP με την βοήθεια της προγραμματιστικής διεπαφής που προσφέρουν.

Το λογισμικό ImageMagick [36] χρησιμοποιήθηκε για την επεξεργασία των GIF εικόνων. Με την βοήθειά του μπορεί κάποιος να διαλέξει το νέο μέγεθος μιας εικόνας και να επηρεάσει την ποιότητα της εφαρμόζοντας διάφορα φίλτρα που του παρέχονται έτοιμα από την βιβλιοθήκη. Παρέχει προγραμματιστική διεπαφή σε C άλλα και σε Perl. Εμείς χρησιμοποιήσαμε αυτή της C μια και ο κώδικας από τον οποίο την καλούμε είναι και αυτός C.

Η βιβλιοθήκη λογισμικού που χρησιμοποιήθηκε στην περίπτωση των αρχείων με μορφή MP3 και MPEG (video) ονομάζεται FFmpeg [37]. Είναι ικανή να δημιουργεί από μια είσοδο πολλά αρχεία διαφορετικών μορφών το καθένα. Έχει αναπτυχθεί κυρίως για το λειτουργικό σύστημα Linux άλλα υπάρχει και μια έκδοση και για τα Windows χωρίς όμως γραφικό περιβάλλον. Στην περίπτωση των MP3 χρησιμοποιήσαμε το λογισμικό αυτό για να τα κωδικοποιήσουμε στην ίδια μορφή (MP3) με χαμηλότερο όμως ρυθμό μετάδοσης (bitrate), ενώ στην περίπτωση των MPEG αλλάζουμε τον ρυθμό μετάδοσης καθώς και το μέγεθος του παραθύρου του video. Επιπλέον, υπάρχουν επιλογές για να συμπεριλαμβάνεται ή όχι ο ήχος του video μέσα στον νέο αρχείο. Θα πρέπει να παρατηρήσουμε πως από τις βιβλιοθήκες που παρέχουν παρόμοια λειτουργικότητα και έχουμε χρησιμοποιήσει, η FFmpeg, είναι ίσως από τις γρηγορότερες σε χρόνο εκτέλεσης.

Επίσης θα πρέπει να αναφερθεί πως οι δυνατότητες αυτών των εργαλείων είναι αρκετά μεγαλύτερες αλλά εμείς επιλέξαμε αυτά τα χαρακτηριστικά τους για να επιδείξουμε την όλη διαδικασία που θα εκκινεί μια αίτηση με πληροφορία CC/PP.

3.3.3.4 Τα πρωτότυπα που υλοποιήθηκαν και ενδεικτικές μετρήσεις.

Δημιουργήσαμε μια HTML ιστοσελίδα με κάποια URLs να “δείχνουν” σε αρχεία εικόνων, ήχου και video. Κατόπιν, όλα αυτά τα αρχεία τα κάναμε διαθέσιμα στο Διαδίκτυο με την βοήθεια

του Apache (με το CC/PP) και τα προσπελάσαμε από έναν κοινό browser όπως είναι το Netscape 6.2. Κάποια από τα αποτελέσματα που πήραμε φαίνονται στις εικόνες που βρίσκονται στο Παράρτημα Β. Ενδεικτικά μπορούμε να αναφέρουμε κάποια αρχικά μεγέθη αρχείων που χρησιμοποιήθηκαν και τα τελικά τους μεγέθη μετά την μετατροπή. Για τις μετατροπές που έγιναν τα χαρακτηριστικά που μας ενδιέφεραν από το Profile που μεταφερόταν ήταν η χωρητικότητα του υποτιθέμενου συνδέσμου με τον οποίο συνδέεται η συσκευή στο Διαδίκτυο, το μέγεθος της οθόνης που διαθέτει και το αν μπορεί η συσκευή να παίζει ήχο ή όχι. Ανάλογα με τα χαρακτηριστικά αυτά ποικίλουν και τα νέα μεγέθη των νέων αρχείων που προκύπτουν. Ένα παράδειγμα CC/PP profile που χρησιμοποιήθηκε παρουσιάζεται και αυτό στο Παράρτημα Β.

	Initial Size (bytes)	Bit rate	Sound	Size	Final Size (bytes)
Video (.mpg) File: Fastpark.mpg	649.412	32 Kbps	No	100x100	43.012
Audio (.mp3) File: Fantasia.mp3	1.437.696	32 Kbps	Yes	N/A	359.026
Images (.gif) File: ngiann.gif	60.225	N/A	N/A	40x40	2.619

Πίνακας 13 Ενδεικτικά αποτελέσματα transcoding κάποιων αρχείων με συγκεκριμένο Profile και παραμέτρους

Όσον αφορά τις εικόνες στην έκδοση 40x40 είναι πολύ μικρές αλλά θέλαμε απλώς να πετύχουμε μια μετατροπή για να πάρουμε μια ιδέα του αποτελέσματος. Το πειραματικό περιβάλλον περιγράφεται στο Παράρτημα Β.

4 Συμπεράσματα της εργασίας και μελλοντικές επεκτάσεις

Καθώς ο παγκόσμιος ιστός συνεχώς μεγαλώνει σε μέγεθος, ταυτόχρονα αυξάνονται και τα δεδομένα τα οποία διαθέτει προς τους χρήστες του. Μέχρι πριν λίγο καιρό τα δεδομένα αυτά ήταν διαθέσιμα μόνο σε παραδοσιακές συσκευές που αποτελούνταν κυρίως από παραδοσιακούς υπολογιστές γραφείου. Οι υπολογιστικές συσκευές όμως έχουν εξελιχθεί και όλο και περισσότερες από αυτές ενσωματώνουν τα χαρακτηριστικά που απαιτούνται για να έχουν πρόσβαση στον παγκόσμιο ιστό. Η αρχή έγινε με τους φορητούς υπολογιστές, τα κινητά

τηλέφωνα, τα Pocket PCs ενώ μελλοντικά δεν αποκλείονται και οι “φορητοί” υπολογιστές (wearable computers).

Αρκετή από την πληροφορία που υπάρχει στο διαδίκτυο είχε οργανωθεί με το σκεπτικό πως ο χρήστης πάντα θα την προσπελαίνει από ένα υπολογιστή γραφείου. Λόγω του πλήθους όμως των νέων συσκευών τα τελευταία χρόνια και για λόγους καλύτερης οργάνωσης της πληροφορίας έχει γίνει ένας διαχωρισμός της παρουσίασης των δεδομένων από τα ίδια τα δεδομένα. Αυτό ξεκίνησε το 1998 με την εμφάνιση της γλώσσας αναπαράστασης πληροφορίας XML. Αυτό ήταν ένα πρώτο σημαντικό βήμα για να μπορέσει να υποδεχτεί το Διαδίκτυο πρόσβαση και από άλλες συσκευές με διαφορετικές και συνήθως ελάχιστες δυνατότητες (thin clients).

Για την διευκόλυνση αυτών των συσκευών άλλα και των ήδη υπάρχοντων έχουν εφαρμοστεί αρκετές τεχνικές που έχουν ελεγχθεί και αποδίδουν ικανοποιητικά κάτω από ορισμένες προϋποθέσεις (caching, prefetching). Στην τεχνική της ελαχιστοποίησης των ιστοσελίδων όπου αφαιρείται η περιττή πληροφορία προτείνουμε την αντικατάσταση των διαδικτυακών συνδέσμων (URLs) με νέα μικρότερου μεγέθους όπως στην δική μας παραλλαγή για να επιτευχθεί ακόμα μικρότερη μεταφορά δεδομένων σε συσκευή που δεν έχει την πολυτέλεια σύνδεσης μεγάλης χωρητικότητας. Οι επεκτάσεις που μπορούν και πρέπει να γίνουν σε περίπτωση που χρησιμοποιηθεί αυτή η τεχνική σε κάποιο πραγματικό σύστημα είναι η υλοποίηση της εφαρμογής σε κατάλληλη γλώσσα για να μπορεί να εκτελείται γρήγορα γιατί οι απαιτήσεις ενός proxy στο Διαδίκτυο είναι αρκετά μεγάλες και αυξάνονται δραματικά με τον αριθμό των χρηστών τους οποίους εξυπηρετεί. Επίσης, η επέκταση με την χρήση του BASE πεδίου της HTML που θα λύσει το πρόβλημα της αίτησης που αναφέρεται σε ανύπαρκτη διεύθυνση και περιγράφηκε στην σχετική ενότητα έχει ήδη ξεκινήσει.

Η χρήση της XML άνοιξε το δρόμο για τον διαχωρισμό της παρουσίασης των δεδομένων από τα ίδια τα δεδομένα (content and presentation). Για να μπορέσει όμως να εφαρμοστεί η κατάλληλη παρουσίαση πάνω στα κατάλληλα δεδομένα θα πρέπει ο κάτοχος αυτών να έχει κάποια πληροφορία για την συσκευή που τα ζητάει ώστε να εκτελέσει τις απαραίτητες ενέργειες. Κάποια πληροφορία στους servers μεταδίδεται στην HTTP επικεφαλίδα και συγκεκριμένα στο User-Agent πεδίο. Από την στιγμή όμως που οι συσκευές όλο και πληθαίνουν η πληροφορία αυτή πρέπει να οργανωθεί. Μια πρόταση προς αυτή την κατεύθυνση είναι το CC/PP πρωτόκολλο που περιγράφηκε και το οποίο προωθείται από τον οργανισμό W3C. Για να μπορέσει όμως το CC/PP να εδραιωθεί σαν πρότυπο παγκοσμίως θα πρέπει να αποσαφηνιστούν κάποια τμήματα του, όπως είναι η χρήση της RDF ή της XML. Επίσης θα πρέπει να οριστεί το λεξικό με βάση το οποίο θα χρησιμοποιείται το πρωτόκολλο και το αν θα ενσωματωθεί στο HTTP ή θα αποτελεί νέο πρωτόκολλο. Στην δεύτερη περίπτωση θα πρέπει να οριστούν ακόμα περισσότερες

λεπτομέρειες για το πλαίσιο επικοινωνίας μεταξύ server και χρήστη. Μια κατεύθυνση στην οποία θα μπορούσε να επεκταθεί αυτό που ήδη έχει υλοποιηθεί είναι μαζί με το transcoding που γίνεται αυτή τη στιγμή να υλοποιείται και διαφορετική παρουσίαση των στατικών δεδομένων. Τεχνολογίες που χρησιμοποιούνται για αυτό το σκοπό είναι συνήθως τα CSS (Cascading StyleSheets) πολλών επιπέδων και οι XSLT (eXtensible StyleSheet Language Transformations) “επεξεργαστές” οι οποίοι εφαρμόζουν διάφορα Style Sheets στα δεδομένα που πρόκειται να παρουσιαστούν για να δημιουργηθεί η τελική ιστοσελίδα. Με αυτό τον τρόπο θα μπορεί το σύστημα να παραδίνει τα νέα transcoded αρχεία μέσα σε κατάλληλες διαμορφωμένες σελίδες.

Τέλος σε μελλοντική έκδοση θα μπορούσαν να ενσωματωθούν τα συστήματα με το CC/PP και τον HtmlMinimizer για ακόμα μεγαλύτερη απόδοση. Το αν θα εφαρμοστεί η ελαχιστοποίηση σε κάποια ιστοσελίδα θα μπορούσε να υπάρχει σαν προτίμηση του χρήστη μέσα στο CC/PP profile του. Γενικότερα όμως θα πρέπει να επισημανθεί πως ένα υβριδικό μοντέλο επεξεργασίας των δεδομένων είναι καλύτερο όσον αφορά το πλήθος των πιθανών συνδυασμών των απαιτήσεων που θα μπορούσαν να δημιουργηθούν και την απόδοση του συστήματος. Για παράδειγμα ένα αρχείο που διαθέτει πολλά URLs θα έπρεπε να υποστεί επεξεργασία από τον HtmlMinimizer που υλοποιήθηκε ενώ ένα αρχείο με απλό κείμενο θα ήταν καλύτερα να μεταφερθεί συμπιεσμένο με άλλο λογισμικού τύπου gzip. Έτσι θα υπάρχει καλύτερη απόδοση σε μεγαλύτερο πλήθος περιπτώσεων και αλληλοσυμπλήρωση των διαφόρων τεχνικών μεταξύ τους.

5 Παράρτημα Α

A CC/PP aware Apache Web Server

Christos Papachristos and Evangelos Markatos
Institute of Computer Science
Foundation for Research and Technology – Hellas

1. Introduction

While current Internet is looking for standards that it can be based and grown upon them in a stable manner, different protocols have been suggested by several consortiums and groups. The Internet of the future will likely contain not only the standard PCs and portable computers but many other devices as well. We have already seen the mobile phones that are trying to access the world of Internet despite their limited capabilities in power and size.

In order for such mobile devices to access the Internet a new protocol must be used. The weaknesses mentioned, about the low power and limited size that such devices may have forced us to treat them in a different way depending on their capabilities. This can be explained by the following example.

When a PC (a client) is requesting a web server for a web document the web server will return that document. When another device with some different capabilities (mobile) is requesting the same document as in the previous case the interaction between the client and the server should not be the same. Suppose that the web document contains large images. In the first case, the computer has a screen capable of presenting such images. In the latter case however, the mobile phone has a much smaller screen incapable of handling those large images. Such devices need to be integrated with the current Internet in a way that they receive the best result depending on their capabilities. For that reason, we need a protocol that will take care of the interaction between clients and servers and make servers 'recognize' the kind and the capabilities of the client of each incoming request.

Such protocol, that could provide the web servers with the information of what kind of client performs a request each time, is the new CC/PP (Composite Capability/Preference Profiles) protocol. This protocol is suggested as a standard from the W3C group [WWW] (<http://www.w3c.org>).

2. Related Work & Possible Solutions

Currently there are not many implementations that can support the CC/PP protocol. A web server at the moment that is capable of handling HTTP requests containing CC/PP headers is the Jigsaw web server implemented by the W3C. Other implementations have been developed by the Keio University [KY] and the University of Wales [STL]. Researchers from Keio University have developed their own browser which supports the CC/PP protocol (called PANDA) and they make use of a CC/PP aware server proxy to communicate with the web server.

In the second case of University of Wales a server proxy and a browser proxy have been used in order to test the functionality of the CC/PP. The browser proxy was used to help current browsers and devices (without CC/PP) interact with the CC/PP compliant part of the system designed. The server proxy was in front of the web server containing the data requested, and it was responsible of requesting the data with traditional HTTP headers without CC/PP. The communication with the HTTP+CC/PP headers is held between these two proxies.

There is also another application developed in Monash University that can serve a mobile device with content read from a file [LT]. The system takes as input a document in MS excel format. It then applies some constraints, that specify the layout of the output document. Depending on the CC/PP attributes read from .ccpp file it produces a file that can be shown in a PalmOS emulator. The format of the file containing the CC/PP attributes is conformant to the W3C suggestions about the format of the Profile header. In this approach the CC/PP information is not sent by the client using a known protocol like HTTP. HTTP protocol may be involved in a newer version.

Four different approaches can be used in order to provide CC/PP functionality to a client:

1. The first approach is implemented with two intermediary proxies, a client proxy and server proxy. In this architecture CC/PP information is only transferred between these two proxies and they are responsible to perform the protocol translation in order the end-to-end (client through server) communication to be achieved (Figure 1). Having these two proxies between the corresponding client and the server may make the communication slow. There is also the issue that adding more intervening parts in the system is like adding more failure points to the round-trip request-response scenario and becomes less flexible in possible future changes.

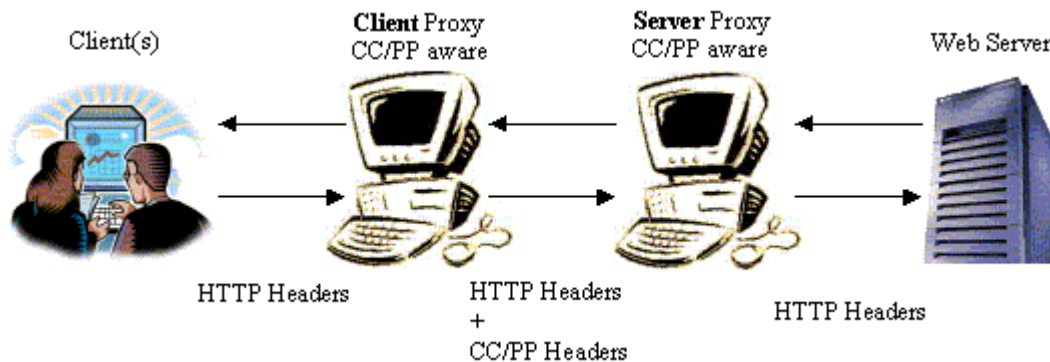


Figure 1: Architecture using two edge proxies supporting CC/PP communication between two hosts.

2. The second approach adopted from the Keio University includes one intermediary proxy, a server proxy. The PANDA browser implemented is CC/PP aware, so the client proxy for the translation between HTTP to HTTP+CC/PP and vice versa is not needed (Figure 2). In this approach the new browser implemented helped the researchers to test the functionality of the CC/PP protocol. Having only one proxy is a common case as this architecture is widely used in current Internet. The difference is that proxy's current usage is focused on serving clients with the cached documents so they are placed closer to the clients and away from the server in order to minimize the transfer delay of the documents. Locating a proxy away from the client may not have the same effects.

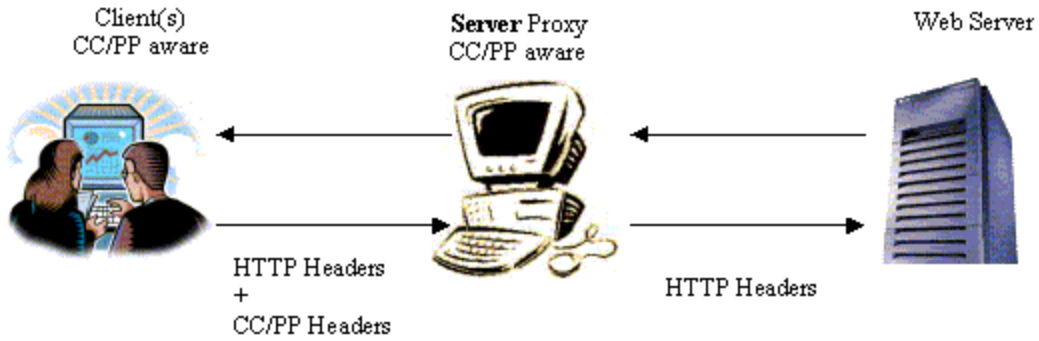


Figure 2: Communication between a client supporting CC/PP and a server that does not through a server proxy.

- The third approach is to implement a CC/PP aware web server communicating with an intermediary client proxy, as seen in Figure 3. That means, that a web server must be capable of handling HTTP with CC/PP headers. The client proxy is used to translate the HTTP requests of each client to new HTTP requests including CC/PP information.

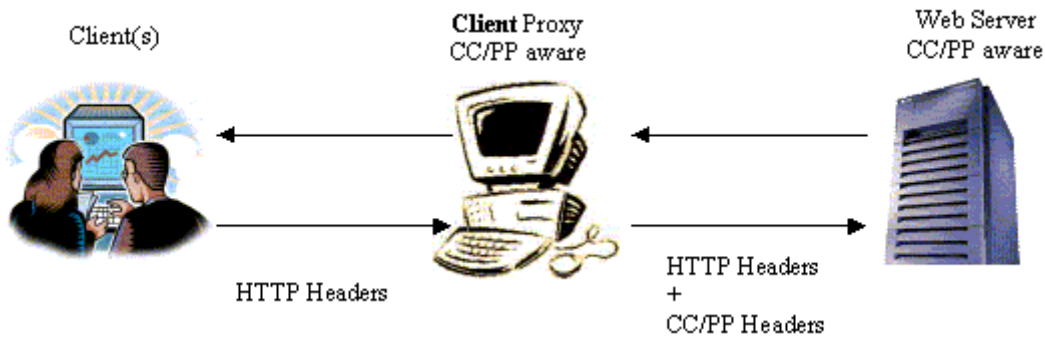


Figure 3: Communication between a web server supporting the CC/PP protocol with a client not supporting CC/PP. Usage of client proxy.

An alternative is to enhance the functionality of the client proxy in a client patch that will perform this translation as shown in Figure 4. This can give each client the opportunity to test if the CC/PP functionality provided by this new protocol offers any advantages in his/her daily navigation through the World Wide Web.

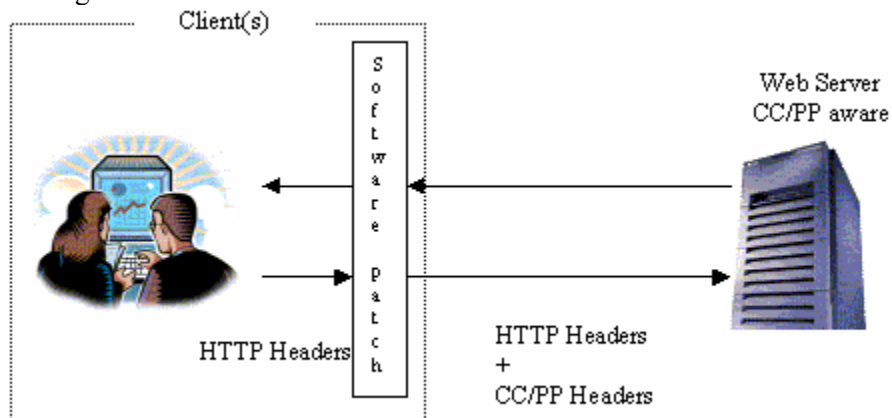


Figure 4: Communication between a configured Apache to support CC/PP and the usage of a software patch in the client side.

4. At last, there is the ideal approach of both server and client capable of sending and receiving CC/PP headers. Though is the best approach it is better to make one stable step each time and pass through some of the mentioned approaches (Figure 5).

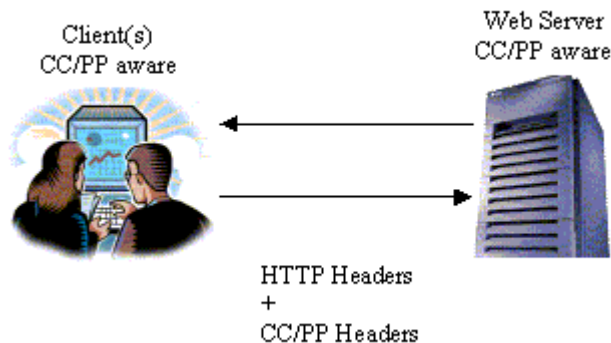


Figure 5: Ideal communication between two CC/PP compliant end hosts.

3. Functionality of proxies and software patch.

3.1 Client Proxy and Software Patch.

A client proxy as shown in the figures 1 and 3 communicates with the web client with the standard HTTP protocol. The functionality of the client proxy is to convert the standard HTTP headers sent by the web client to extended HTTP headers that contain CC/PP headers (HTTP headers + CC/PP headers). The new headers are then sent to the appropriate web server. If the web server is CC/PP aware then the headers are sent directly to that server. Otherwise the headers will be sent to a server proxy to make the appropriate translation.

3.2 Server Proxy.

A server proxy is used in implementations where the web server has not been transformed to support the CC/PP protocol. Its functionality is to:

- perform any necessary parsing in the headers (the CC/PP part)
- send the parsed headers to the appropriate web server (only HTTP headers)
- convert the returned results from the web server according to the CC/PP headers that where previously parsed
- send back the reply to the client. The client could be the browser or a client proxy.

A client proxy and a server proxy have similar functionality except that they communicate with different end-parts, the client and the server respectively.

4. Our approach

Our approach is based on the usage of a client proxy and is focused on integrating the server's proxy functionality inside a web server as a module. The whole system includes a client proxy, a web client (e.g iexplorer netscape) and the transformed server with the new module (Figure 6).

A well-known web server that is open source and is widely used is the Apache web server. These features made our choice easier between other existing web servers. The API (Application Programming Interface) provided from the Apache web server enables the implementation of new modules. We chose not to consider the issue of the client part as we wanted to focus on the server-side. Instead of trying to extend a well-known client with the CC/PP protocol, we decided to use a simple client proxy that could do the appropriate work of the transformation between original HTTP and extended HTTP (CC/PP headers included).

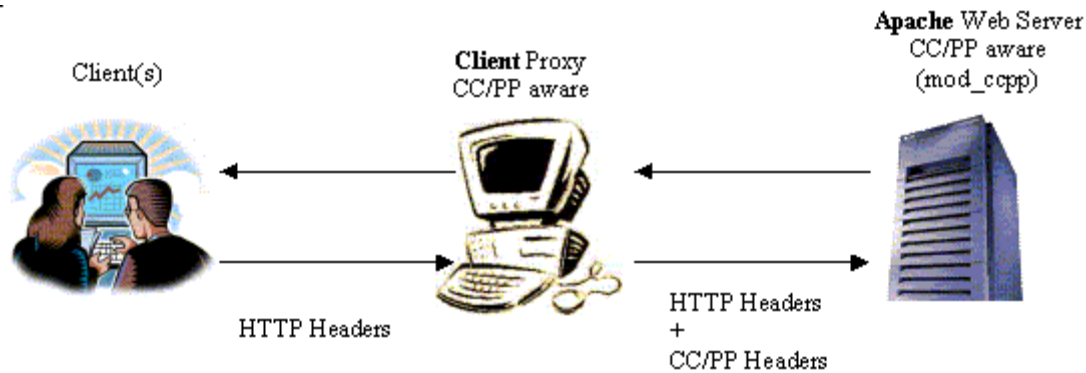


Figure 6: The communication between a CC/PP aware Apache server with a client proxy

The full functionality of a server proxy can be incorporated into the Apache web server. With regard to the client side, we developed a proxy that mediates between the Apache server and the web browser. In a future version we may try to transform the client proxy into a software patch (Figure 4).

Till now we have implemented a module for the Apache server that handles the CC/PP headers. This module is called `mod_ccpp` (`mod_ccpp.c`). It can be imported to the Apache like all the other extra modules with the usage of '`--activate-module`' derivative. This can be done in the initial configuration of the server. In order to test our module we have linked it with the [ImageMagick](#) (an image processing tool) library which can be used to transform images. ImageMagick library provides an API to transform images between formats and sizes. The generated web page is now dependant on the client's device characteristics sent in the CC/PP header.

Including that library inside the module makes the deployment of the module little more complicated. One automatic generated Makefile had to be altered in order to link the module with the ImageMagick library. Having installed the module, we were able to make requests to the Apache with various values for the CC/PP header fields. The server responded as expected in all cases.

As future work we may try to develop a client patch that could enable a browser (Mozilla) to send CC/PP headers and avoid the usage of the client proxy. Some measurements will be also held to determine the overhead of the CC/PP headers processing and the image transformation.

5. References

- [LT] Lawrence Teo, "Constraint-Based Document Layout for Mobile Computers", http://www.csse.monash.edu.au/projects/MobileComponents/projects/pda_doc_layout/report.html
- [WWW] World Wide Web Consortium, Composite Capabilities/Preference Profiles Working Group, <http://www.w3.org/Mobile/CCPP/>
- [KY] Kinuko Yasuda, "Implementation and Evaluation of Keio CC/PP Implementation", <http://yax.tom.sfc.keio.ac.jp/panda/slidemaker/0011ccpp/Overview.html>
- [STL] Stuart David Lewis, "Content Negotiation for Varying Web Enabled Devices", <http://www.ccpp.co.uk/>

6 Παράρτημα Β

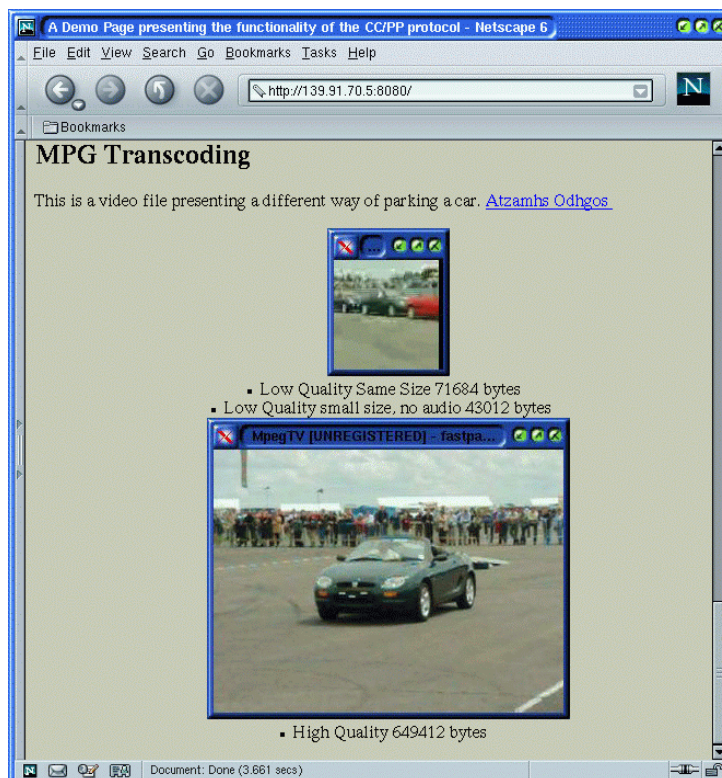
Πειραματικό περιβάλλον

Τα αποτελέσματα που παρουσιάζονται στην ενότητα είναι διαθέσιμα από το Διαδίκτυο αρκεί να ενεργοποιηθούν οι κατάλληλοι servers. Το Linux είναι το λειτουργικό σύστημα στο οποίο έγινε η ανάπτυξη των συστημάτων και στο οποίο εκτελούνται όλες οι εφαρμογές. Συγκεκριμένα χρησιμοποιήθηκε η 1.3.22 έκδοση του Apache με το 1.1.2 JServ module. Οι βιβλιοθήκες που χρησιμοποιήθηκαν για το transcoding είναι η ImageMagick5.4.2 και η FFmpeg0.4.5.

Ο Apache με το CC/PP module είναι στη διεύθυνση <http://julia.ics.forth.gr:8080>. Για να ενεργοποιηθεί όμως το CC/PP θα πρέπει ο browser να δείχνει στον proxy julia.ics.forth.gr στο port 22222. Σε περίπτωση που θέλουμε να δοκιμάσουμε το CC/PP στον Apache proxy τότε χρησιμοποιούμε στον browser τον proxy julia.ics.forth.gr στο port 33333.



Εικόνα 12 Αποτέλεσμα του transcoding σε μια GIF εικόνα. Η μετατροπή της σε μικρότερο μέγεθος



Εικόνα 13 Αποτέλεσμα του transcoding σε ένα MPG video. Το νέο μέγεθός του αν μικρύνουμε το παράθυρο

Τα ίδια αποτελέσματα θα δει οποιοσδήποτε προσπελάσει την ιστοσελίδα <http://julia.ics.forth.gr:8080> και κάνει στον browser του τις ρυθμίσεις που υποδεικνύονται.

Το CC/PP profile σύμφωνα με το οποίο γίνονται οι μετατροπές στα αρχεία είναι το παρακάτω.

Αρχικά ο proxy θα στείλει την CC/PP πληροφορία που δείχνει ο Πίνακας 14.

```
Profile:"http://julia.ics.forth.gr/index.html", "1-FSDF"
Profile-Diff-1:<?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/TR/1999/PR-rdf-syntax-19990105#"
      xmlns:PRF="http://www.w3.org/TR/WD-profile-vocabulary#">
<Description about="Hardware">
<Modifications>
<Description ScreenSize="100x100" />
</Modifications>
</Description>
</RDF>
```

Πίνακας 14 Το CC/PP profile που στέλνει ο Java proxy μαζί με τους HTTP headers στον Apache server

Όπως φαίνεται χρησιμοποιείται έμμεσος τρόπος για να βρει ο server ολόκληρο το CC/PP profile της συσκευής. Το αρχείο <http://julia.ics.forth.gr/index.html> περιέχει όλη την πληροφορία ενώ η συσκευή υποτίθεται πως δίνει μόνο μια διαφοροποίηση για το μέγεθος της οθόνης που διαθέτει.

Ο Πίνακας 15 παρουσιάζει πιο είναι το πραγματικό Profile της συσκευής και το οποίο το διαβάσει ο Apache μέσω του module που υλοποιήθηκε.

```
<?xml version="1.0" ?>
<RDF xmlns="http://www.w3.org/TR/1999/PR-rdf-syntax-19990105#"
xmlns:PRF="http://www.w3.org/TR/WD-profile-vocabulary#">
<Description about="Hardware">
<Defaults>
  <Description Vendor="Nokia" />
  <Description ScreenSize="100x100" />
  <Description Bandwidth="32Kbps" />
  <Description Memory="16mB" />
</Defaults>
<!--
Modifications>
  <Description PRF:Memory="32mB" />
</Modifications
-->
</Description>
<Description about="SoftWare">
<Defaults>
  <Description OSName="EPOC" />
  <Description OSVersion="5.3" />
</Defaults>
</Description>
<Description about="Browser">
<Defaults>
  <Description Name="Netscape" />
  <Description Version="6.2" />
  <Description Sound="No" />
</Defaults>
</Description>
</RDF>
```

Πίνακας 15 Ολόκληρο το CC/PP profile που θα χρειαστεί να διαβάσει ο Apache μετά την υπόδειξη της συσκευής ότι περιγράφεται από αυτό

7 Αναφορές και Βιβλιογραφία

- [1]. The HTTP Extension Framework, <http://www.w3.org/Protocols/HTTP/ietf-http-ext/>
- [2]. CC/PP exchange protocol based on HTTP Extension Framework, <http://www.w3.org/1999/06/NOTE-CCPPexchange-19990624>
- [3]. The HyperText Transfer Protocol (HTTP), <http://www.w3.org/Protocols>
- [4]. The CC/PP protocol, Composite Capabilities/Preference Profiles Working Group, <http://www.w3.org/Mobile/CCPP/>
- [5]. Composite Capability/Preference Profiles (CC/PP): Structure, W3C Working Draft 21 July 2000, <http://www.w3.org/TR/2000/WD-CCPP-struct-20000721>
- [6]. A user side framework for network negotiation, W3C Note 27 July 1999, <http://www.w3.org/TR/NOTE-CCPP/>
- [7]. Resource Description Framework (RDF) Model and Syntax Specification, <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>
- [8]. The W3C Workshop on Delivery Context, <http://www.w3.org/2001/12/2002-03-05-delivery-workshop.html>
- [9]. Kinuko Yasuda, “Implementation and Evaluation of Keio CC/PP Implementation”, <http://yax.tom.sfc.keio.ac.jp/panda/slidemaker/0011ccpp/Overview.html>
- [10]. Master Thesis of Stuart David Lewis, “Content Negotiation for Varying Web Enabled Devices”, <http://www.ccpp.co.uk/>
- [11]. Lawrence Teo, “Constraint-Based Document Layout for Mobile Computers”, http://www.csse.monash.edu.au/projects/MobileComponents/projects/pda_doc_layout/report.html
- [12]. Mark Butler, HPL-2001-260, “*DELI: A delivery context library for CC/PP and UAProf*”, <http://www-uk.hpl.hp.com/people/marbut/DeliUserGuideWEB.htm>
- [13]. The Device Independent Content Engine (DICE), <http://dice.ccpp.info/>
- [14]. The Apache Software Foundation <http://www.apache.org>
- [15]. The Apache Cocoon Project, <http://xml.apache.org/cocoon/>
- [16]. Gaurav Banga, Fred Douglass, Michael Radinovich, “*Optimistic Deltas for WWW Latency Reduction*”, 1997 USENIX Technical Conference.
- [17]. Jeffrey C. Mogul, Fred Douglass, Anja Balachander Krishnamurthy, “*Potential benefits of delta encoding and data compression for HTTP*”, SIGCOMM 1997
- [18]. A Mobile WWW-proxy (MOXY) <http://www.hut.fi/~aschantz/moxy/>

- [19]. An open XML-browser for exotic devices, <http://www.xsmiles.org>
- [20]. The Open Mobile Alliance, <http://www.wapforum.org>
- [21]. A Book search-engine, BestBookBuys, <http://www.bestbookbuys.com>
- [22]. Subhabrata Sen, Jennifer Rexford, and Don Towsley, “*Proxy prefix Caching for Multimedia Streams*”, INFOCOMM '99.
- [23]. Vladimir Korolev and Anupam Joshi, “*An End-End Approach to Wireless Web Access*”, In Proceedings of the International Workshop on Wireless Networks and Mobile Computing, April 2001.
- [24]. Thomas M. Kroeger, Darell D. E. Long, “*Exploring the Bounds of Web Latency Reduction from Caching and Prefetching*”, Proceedings of the USENIX Symposium on the Internet Technologies and Systems, Monterey California, December 1997
- [25]. Evangelos Markatos and Catherine E. Chronaki : “*A Top-10 Approach to Prefetching on the Web*”, In Proceedings of INET' 98 (The Internet Summit), Geneva, Switzerland, July 1998
- [26]. Fabian Breg, Aart Bik and Dennis Gannon, “*Exploiting Implicit Loop Parallelism using Multiple Multithreaded Servers in Java*” (Dec 1997).
- [27]. Evangelos Markatos and Catherine Chronaki, “*Caching, Prefetching and Coherence in the World Wide Web*”, ERCIM News No.25 - April 1996
- [28]. White Paper, “*IBM Transcoding Solution and Services*”,
http://www.research.ibm.com/networked_data_systems/transcoding/transcodef.pdf
- [29]. Hypertext Transfer Protocol -- HTTP/1.1,
<http://www.w3.org/Protocols/rfc2068/rfc2068>
- [30]. Extreme HTML Optimization, <http://webreference.com/authoring/languages/html/optimize/>
- [31]. The Java Tutorial, A practical guide for programmers,
<http://www.java.sun.com/docs/books/tutorial>
- [32]. The Java Servlet Technology, <http://java.sun.com/products/servlet/>
- [33]. The Apache Jserv Project, <http://java.apache.org/jserv/>
- [34]. The GZIP home page, <http://www.gzip.org>
- [35]. The Java Compiler Compiler. The Java Parser Generator,
http://www.webgain.com/products/java_cc/
- [36]. The ImageMagick library, Convert Edit and Compose Images,
<http://www.imagemagick.org>
- [37]. FFmpeg Streaming Multimedia System, <http://ffmpeg.sourceforge.net/>
- [38]. The ZDNet journal, <http://www.zdnet.com>

- [39]. Content Negotiation (CONNEG), <http://www.ietf.org/html.charters/OLD/conneg-charter.html>
- [40]. The MD5 Message-Digest Algorithm, <http://www.faqs.org/rfcs/rfc1321.html>
- [41]. Netcraft Web Server Survey, <http://www.netcraft.com/survey/>
- [42]. The Conceptual Architecture of the Apache Web Server,
http://www.math.uwaterloo.ca/~oadragoi/CS746G/a1/apache_conceptual_arch.html
- [43]. Dan Duchamp, “*Prefetching Hyperlinks*”, Proceedings of USITS '99. The 2nd USENIX Symposium on Internet Technologies and Systems.
- [44]. The HyperText Markup Language (HTML), <http://www.w3.org/MarkUp/>
- [45]. The Synchronized Multimedia Language (SMIL), <http://www.w3.org/AudioVideo/>
- [46]. Scalable Vector Graphics (SVG), W3C Recommendation 04 September 2001,
<http://www.w3.org/TR/SVG/>
- [47]. Extensible Markup Language (XML), <http://www.w3.org/XML/>
- [48]. The Jakarta site – Apache Project, <http://jakarta.apache.org/tomcat/>