ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Σημειώσεις για τα πρωτόκολλα στρώματος εφαρμογής: HTTP, FTP, E-mail, DNS

Διδάσκων
Αν. Καθηγητής Ι.Σ.Βενιέρης

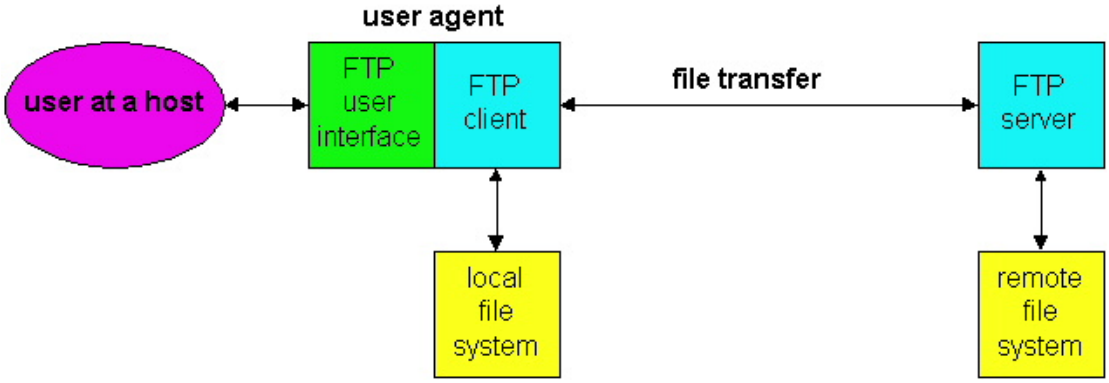
Αθήνα, Ιανουάριος 2003

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. Το πρωτόκολλο FTP 3
2. Το πρωτόκολλο DNS 6
2.1. Λειτουργία του DNS 7
2.2. DNS Μηνύματα 12
3. Ηλεκτρονικό Ταχυδρομείο 15
3.1. SMTP 16
3.2. Μορφή των μηνυμάτων – Πρωτόκολλο MIME 18
3.3. Πρωτόκολλα πρόσβασης στο Ηλεκτρονικό Ταχυδρομείο 22
3.3.1 POP3 24
3.3.2 IMAP 25
4. Το πρωτόκολλο HTTP 27
4.1. Μόνιμες και μη μόνιμες συνδέσεις 28
4.2. Μορφή των HTTP μηνυμάτων 30
4.3. Αλληλεπίδραση χρήστη και εξυπηρετητή 34
4.4. Υπό συνθήκη GET 35
4.5. Web cache 37

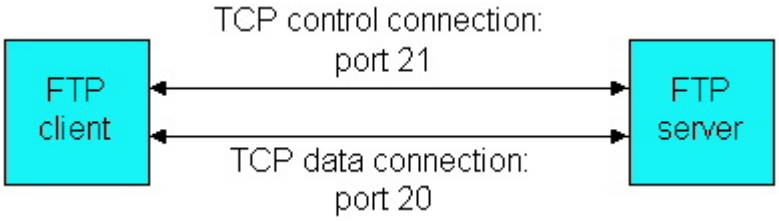
1. Το πρωτόκολλο FTP

Το **FTP (File Transfer Protocol)** είναι ένα πρωτόκολλο που χρησιμοποιείται για την μεταφορά αρχείων από έναν υπολογιστή του Διαδικτύου σε κάποιον άλλον. Το FTP ξεκίνησε πειραματικά το 1971 αλλά παραμένει ως τις μέρες μας εξαιρετικά δημοφιλές.
Ας υποθέσουμε ότι ένας χρήστης επιθυμεί να μεταφέρει ένα ή περισσότερα αρχεία από ή προς έναν άλλο απομακρυσμένο χρήστη. Για να μπορέσει ο χρήστης να έχει πρόσβαση στα αρχεία του απομακρυσμένου υπολογιστή, δηλαδή σε κάποιο *λογαριασμό (account)* του απομακρυσμένου υπολογιστή, πρέπει να δώσει ένα *αναγνωριστικό όνομα χρήστη (user name*) και έναν *κωδικό (password)*. Μετά την παροχή των παραπάνω πληροφοριών *πιστοποίησης (authentication)*, ο χρήστης μπορεί να μεταφέρει αρχεία από το σύστημα αρχείων του προς το απομακρυσμένο σύστημα αρχείων, και αντιστρόφως. Όπως φαίνεται και στο παρακάτω σχήμα ο χρήστης έρχεται σε επαφή με το FTP μέσω ενός αντιπροσώπου FTP.



Ο χρήστης πρώτα παρέχει το *όνομα του απομακρυσμένου υπολογιστή (remote host name)*, με αποτέλεσμα η FTP διαδικασία πελάτη στον τοπικό υπολογιστή να εγκαθιστά μία σύνδεση TCP με τον εξυπηρετητή FTP στον απομακρυσμένο υπολογιστή. Τότε ο χρήστης παρέχει το user name και το password, τα οποία στέλνονται μέσω της σύνδεσης TCP ως μέρος εντολών FTP. Μετά την πιστοποίηση του χρήστη από τον εξυπηρετητή, ο χρήστης μπορεί να αντιγράψει ή να μετακινήσει ένα ή περισσότερα αρχεία από το τοπικό του σύστημα αρχείων προς το απομακρυσμένο σύστημα αρχείων, και αντιστρόφως.
Το FTP χρησιμοποιεί δύο παράλληλες συνδέσεις TCP για την μεταφορά ενός αρχείου: μία **σύνδεση ελέγχου (control connection)** και μία **σύνδεση δεδομένων (data connection)**. Η σύνδεση ελέγχου χρησιμοποιείται για την μεταφορά πληροφοριών ελέγχου μεταξύ των δύο υπολογιστών, πληροφορίες όπως το όνομα χρήστη (user name), τον κωδικό, για την αλλαγή του απομακρυσμένου καταλόγου και εντολές για την ανάκτηση (get) ή καταχώρηση (put) αρχείων. Η σύνδεση δεδομένων χρησιμοποιείται για την πραγματική μεταφορά του αρχείου. Εξαιτίας της ύπαρξης δύο TCP συνδέσεων, λέμε ότι το FTP μεταφέρει την πληροφορία ελέγχου **εκτός ζώνης (out-of-band)**. Αντίθετα στα πρωτόκολλα που χρησιμοποιούν μόνο μία σύνδεση λέμε ότι η πληροφορία ελέγχου

μεταφέρεται *εντός ζώνης (in-band)*. Στο παρακάτω σχήμα φαίνονται οι δύο ξεχωριστές TCP συνδέσεις που χρησιμοποιεί το FTP.



Όταν ο χρήστης ξεκινά μία FTP σύνοδο με κάποιον απομακρυσμένο υπολογιστή, το FTP πρώτα εγκαθιστά μία **TCP** σύνδεση ελέγχου στην θύρα (port) **21** του FTP εξυπηρετητή. Ο FTP πελάτης στέλνει το αναγνωριστικό και τον κωδικό του χρήστη μέσω της σύνδεσης ελέγχου. Επίσης, μέσω της σύνδεσης ελέγχου ο FTP πελάτης στέλνει και εντολές για την αλλαγή του απομακρυσμένου καταλόγου. Όταν ο χρήστης ζητήσει μία μεταφορά αρχείου (από η προς τον απομακρυσμένο υπολογιστή) το FTP ανοίγει μία **TCP** σύνδεση δεδομένων στην θύρα **20** του FTP εξυπηρετητή. Μέσω αυτής της σύνδεσης δεδομένων στέλνεται μόνο ένα αρχείο και στη συνέχεια η σύνδεση δεδομένων κλείνει. Αν κατά τη διάρκεια αυτής της συνόδου ο χρήστης θέλει να μεταφέρει και άλλα αρχεία τότε ανοίγονται **ξεχωριστές συνδέσεις δεδομένων**, μία για κάθε αρχείο. Επομένως, στο FTP η σύνδεση ελέγχου παραμένει για όλη τη διάρκεια της συνόδου, ενώ χρησιμοποιείται μία ξεχωριστή σύνδεση δεδομένων για κάθε αρχείο που μεταφέρεται μεταξύ των δύο υπολογιστών.

Κατά την διάρκεια της συνόδου ο εξυπηρετητής κρατάει την *κατάσταση (state)* του χρήστη. Για κάθε σύνοδο έχουμε μία ξεχωριστή σύνδεση ελέγχου που συσχετίζεται με την λογαριασμό του χρήστη, και ο εξυπηρετητής κρατά το τρέχοντα κατάλογο του χρήστη στον λογαριασμό αυτό. Εξαιτίας της πληροφορίας για την κατάσταση των συνόδων των χρηστών έχουμε σημαντική μείωση στον αριθμό των χρηστών που μπορεί να εξυπηρετηθούν ταυτόχρονα, σε σχέση με άλλα πρωτόκολλα που δεν κρατάνε την κατάσταση του χρήστη, όπως το HTTP.

Τελειώνοντας την περιγραφή του FTP, θα αναφερθούμε στις **εντολές (commands)** και τις **αποκρίσεις (replies)** που χρησιμοποιεί. Οι εντολές από τον πελάτη προς τον εξυπηρετητή και οι αποκρίσεις από τον εξυπηρετητή προς τον πελάτη στέλνονται μέσω της TCP σύνδεσης ελέγχου σε **7-bit ASCII κωδικοποίηση**, και είναι αναγνώσιμες. Για τον διαχωρισμό των εντολών μεταξύ τους χρησιμοποιείται αλλαγή γραμμής και κάθε εντολή αποτελείται από τέσσερις κεφαλαίους ASCII χαρακτήρες και από κάποια προαιρετικά ορίσματα. Μερικές από τις πιο συχνά χρησιμοποιούμενες εντολές φαίνονται στον παρακάτω πίνακα.

Εντολή	Ορίσματα	Χρήση
USER	*username*	Αποστολή user name στον εξυπηρετητή
PASS	*password*	Αποστολή password στον εξυπηρετητή
LIST	-	Αίτηση για αποστολή όλων των ονομάτων των αρχείων του τρέχοντα καταλόγου (τα οποία στέλνονται μέσω μίας νέας TCP σύνδεσης δεδομένων)
RETR	*filename*	Ανάκτηση του αρχείου filename από τον τρέχοντα κατάλογο του απομακρυσμένου υπολογιστή
STOR	*filename*	Τοποθέτηση του αρχείου filename στον τρέχοντα κατάλογο του απομακρυσμένου υπολογιστή

Τυπικά υπάρχει μία προς μία αντιστοιχία μεταξύ των εντολών που δίνει ο χρήστης και των εντολών FTP που στέλνονται μέσω της σύνδεσης ελέγχου. Κάθε εντολή ακολουθείται από μία απόκριση, η οποία αποτελείται από έναν τριψήφιο ακέραιο αριθμό και ένα προαιρετικό μήνυμα. Μερικές τυπικές αποκρίσεις φαίνονται στον παρακάτω πίνακα.

Κωδικός απόκρισης	Προαιρετικό μήνυμα
331	User name OK, password required
125	Data connection already open; transfer starting
424	Can’t open data connection
452	Error writing file

2. Το πρωτόκολλο DNS

Κάθε μηχάνημα – host του Διαδικτύου μπορεί να αναγνωριστεί από την IP διεύθυνσή του, που είναι ένας δυαδικός αριθμός των 32 bits. Το ίδιο το δίκτυο (και συγκεκριμένα το στρώμα διαδικτύου) καταλαβαίνει μόνο τις IP διευθύνσεις. Οι άνθρωποι, όμως, μπορούν πιο εύκολα να θυμούνται ονόματα και όχι δυαδικά νούμερα. Για αυτό το λόγο τις περισσότερες φορές τα προγράμματα σπάνια απευθύνονται στους host του Διαδικτύου χρησιμοποιώντας την IP διεύθυνση, αλλά κάνουν χρήση συμβολικών ονομάτων, με την μορφή ακολουθιών ASCII χαρακτήρων. Για την αντιστοίχηση μεταξύ των δυαδικών διευθύνσεων και των διευθύνσεων σε μορφή ASCII χαρακτήρων χρησιμοποιείται ένα πρωτόκολλο του στρώματος εφαρμογών, το **DNS (Domain Name System)**.
Το DNS είναι μία κατανεμημένη βάση δεδομένων που υλοποιείται μέσω μίας ιεραρχίας εξυπηρετητών ονομάτων (name servers) και ένα πρωτόκολλο του στρώματος εφαρμογής που επιτρέπει στους host και στους εξυπηρετητές ονομάτων να επικοινωνούν για τους σκοπούς της υπηρεσίας μετάφρασης των διαφορετικών μορφών διευθύνσεων. Το πρωτόκολλο DNS χρησιμοποιεί το **UDP** και συγκεκριμένα την θύρα **53**.
Το DNS χρησιμοποιείται συχνά από άλλα πρωτόκολλα του στρώματος εφαρμογών, όπως το HTTP και το FTP, για την μετάφραση των διευθύνσεων που δίνουν οι χρήστες σε IP διευθύνσεις. Έστω ότι ένας χρήστης ζητά από τον browser να εμφανιστεί η σελίδα που έχει ως URL το http://www.ntua.gr/index.htm. Για να μπορέσει το μηχάνημα του χρήστη να στείλει μία HTTP αίτηση στον εξυπηρετητή Ιστού (Web server) www.ntua.gr, το μηχάνημα του χρήστη πρέπει να μάθει την IP διεύθυνση του www.ntua.gr. Αυτό συμβαίνει ως εξής: στο μηχάνημα του χρήστη, που λειτουργεί ως DNS πελάτης, ο browser αποσπά από το URL και περνά το www.ntua.gr στον DNS πελάτη. Ως μέρος της DNS **ερώτησης (query)** ο DNS πελάτης στέλνει το όνομα του host στον DNS εξυπηρετητή, από τον οποίο λαμβάνει μία **απάντηση (reply)** που περιλαμβάνει την IP διεύθυνση του host. Στη συνέχεια ο browser ανοίγει μία σύνδεση TCP με τον εξυπηρετητή HTTP που βρίσκεται στην συγκεκριμένη διεύθυνση IP. Όλα τα πακέτα IP (datagrams) που στέλνονται από τον πελάτη προς τον εξυπηρετητή για την συγκεκριμένη σύνδεση (δηλαδή όλες οι αιτήσεις HTTP) έχουν ως διεύθυνση προορισμού αυτή την διεύθυνση IP. Από το παράδειγμα αυτό γίνεται φανερό ότι το DNS εισάγει μία επιπλέον καθυστέρηση στις εφαρμογές του Διαδικτύου που χρησιμοποιούν το DNS. Για τη μείωση της καθυστέρησης αυτής η επιθυμητή διεύθυνση IP συνήθως αποθηκεύεται προσωρινά σε κάποιον κοντινό εξυπηρετητή ονομάτων, ο οποίος βοηθά στην μείωση του φορτίου στο ίδιο το Διαδίκτυο αλλά και στην μείωση της μέσης καθυστέρησης.
Όπως αναφέρθηκε και παραπάνω, το DNS ανήκει στο στρώμα εφαρμογών του Διαδικτύου. Σε σχέση όμως με τα άλλα πρωτόκολλα του στρώματος εφαρμογών, όπως το FTP και το HTTP, το DNS έχει μία σημαντική διαφορά: **ο χρήστης δεν μπορεί να έρθει σε απευθείας επαφή με το DNS**, αλλά αντίθετα το DNS παρέχει μία σημαντική λειτουργία του Διαδικτύου, την μετάφραση μεταξύ των ονομάτων των host και των διευθύνσεων IP, στις εφαρμογές των χρηστών και του άλλου λογισμικού του Διαδικτύου.
Αν και παραπάνω αναφερθήκαμε μόνο στη μετάφραση των ονομάτων των host σε διευθύνσεις IP, το DNS παρέχει και τις ακόλουθες σημαντικές υπηρεσίες:

- **Ψευδώνυμα (Alias) Host:** Ένας host με ένα πολύπλοκο όνομα μπορεί να έχει ένα ή περισσότερα εναλλακτικά ονόματα – ψευδώνυμα, τα οποία είναι συνήθως πιο εύκολα στην απομνημόνευση. Π.χ. ένας host με όνομα hellas.core.ntua.gr μπορεί να έχει ως ψευδώνυμα τα www.ntua.gr και ntua.gr. Στην περίπτωση αυτή το όνομα hellas.core.ntua.gr λέγεται κανονικό (canonical). Το DNS μπορεί να χρησιμοποιηθεί από μία εφαρμογή για την εύρεση του κανονικού ονόματος ή της IP διεύθυνσης, δοσμένου ενός ψευδωνύμου.
- **Ψευδώνυμα Εξυπηρετητών Ταχυδρομείου:** Συχνά και τα ονόματα των διευθύνσεων του ηλεκτρονικού ταχυδρομείου (e-mail) δίνονται σε μορφή εύκολη προς απομνημόνευση. Έστω ότι ο Nick έχει έναν λογαριασμό στο yahoo.gr και η e-mail διεύθυνσή του είναι nick@yahoo.gr. Παρόλα αυτά το όνομα του host του yahoo.gr είναι πιο πολύπλοκο από το απλό yahoo.gr (π.χ. μπορεί να είναι hellas.core.yahho.gr). Το DNS στην περίπτωση αυτή χρησιμοποιείται για την εύρεση του κανονικού ονόματος για το ψευδώνυμο του host, αλλά και για την εύρεση της IP διεύθυνσης του host. Το DNS επιτρέπει να έχουν το ίδιο ψευδώνυμο τόσο ο εξυπηρετητής Ταχυδρομείου όσο και ο εξυπηρετητής Ιστού.
- **Κατανομή φορτίου**: Πολλές τοποθεσίες του Διαδικτύου έχουν πολλούς εξυπηρετητές για τον ίδιο σκοπό, π.χ. την παροχή μίας ιστοσελίδας, εξαιτίας του μεγάλου όγκου της κίνησης από και προς αυτές. Στην περίπτωση αυτή κάθε εξυπηρετητής βρίσκεται σε διαφορετικό μηχάνημα και έχει διαφορετική IP διεύθυνση. Για αυτές τις περιπτώσεις στη βάση δεδομένων του DNS περιέχονται πολλές IP διευθύνσεις με το ίδιο κανονικό όνομα. Όταν κάποιος πελάτης DNS κάνει μία επερώτηση στον DNS εξυπηρετητή για κάποιον όνομα με πολλές διευθύνσεις IP, ο DNS εξυπηρετητής ανταποκρίνεται με όλες τις διευθύνσεις IP, αλλά περιστρέφοντάς κατά μία διεύθυνση την σειρά για κάθε αίτηση που γίνεται για το συγκεκριμένο όνομα. Επειδή συνήθως οι host χρησιμοποιούν την πρώτη διεύθυνση IP που βρίσκουν στην απόκριση DNS, η περιστροφή αυτή κατανέμει το φορτίο μεταξύ των διαφόρων εξυπηρετητών.

2.1. Λειτουργία του DNS

Παρακάτω θα δοθεί η περιγραφή της λειτουργίας του DNS και ειδικότερα η υπηρεσία μετάφρασης των διευθύνσεων. Ο πελάτης στέλνει ένα μήνυμα ερώτησης (query) στον εξυπηρετητή και μετά από κάποιο χρόνο (που μπορεί να είναι της τάξης των msec ή ακόμη και δευτερολέπτων) λαμβάνει ένα μήνυμα απάντησης (reply) το οποίο περιέχει την αντιστοιχία των διευθύνσεων. Επομένως, από την πλευρά του πελάτη το DNS φαίνεται να είναι σαν ένα μαύρο κουτί. Η υλοποίηση όμως της υπηρεσίας μετάφρασης είναι στην ουσία πολύ πιο πολύπλοκη, αφού ένας μεγάλος αριθμός από **εξυπηρετητές ονομάτων (name servers)** είναι κατανεμημένος σε ολόκληρο τον κόσμο και ένα πρωτόκολλο του στρώματος εφαρμογής περιγράφει τον τρόπο με τον οποίο επικοινωνούν οι host και οι εξυπηρετητές ονομάτων.
Μία απλή σχεδίαση του DNS θα είχε μόνο έναν εξυπηρετητή ονομάτων που θα περιείχε όλες τις αντιστοιχίες. Σε αυτό την σχεδίαση οι πελάτες απλά κατευθύνουν όλες τις επερωτήσεις στον εξυπηρετητή ονομάτων, ο οποίος αποκρίνεται απευθείας στους

πελάτες. Αν και αυτή η σχεδίαση είναι απλή δεν ανταποκρίνεται στις σημερινές απαιτήσεις του Διαδικτύου, εξαιτίας των παρακάτω μειονεκτημάτων.

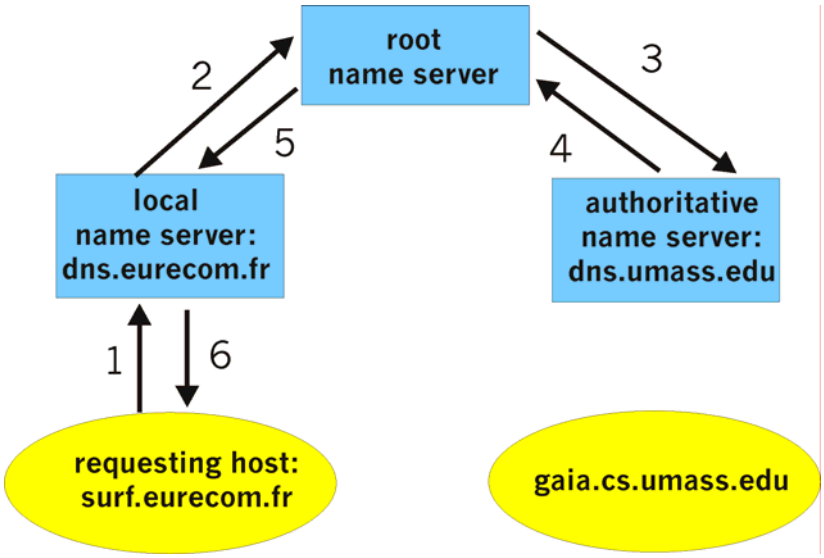
- Αν χρησιμοποιείται μόνο ένας εξυπηρετητής ονομάτων και πάθει κάποια βλάβη θα καταρρεύσει ολόκληρο το Διαδίκτυο.
- Ένας μόνο εξυπηρετητής ονομάτων πρέπει να χειρίζεται όλη την κίνηση του πρωτοκόλλου DNS, που είναι χιλιάδες επερωτήσεις το δευτερόλεπτο.
- Ένας μόνο εξυπηρετητής ονομάτων δεν μπορεί να είναι «κοντά» σε όλους τους πελάτες, με αποτέλεσμα η πλειοψηφία των επερωτήσεων να έχει μη αποδεκτή καθυστέρηση.
- Ένας μόνο εξυπηρετητής ονομάτων πρέπει να κρατάει εγγραφές για όλους τους host του Διαδικτύου, με αποτέλεσμα όχι μόνο η βάση δεδομένων να είναι τεράστια αλλά και να πρέπει να ανανεώνεται πολύ συχνά. Στην τελευταία περίπτωση εμπλέκονται και θέματα ασφάλειας, αφού σε έναν κόμβος του Διαδικτύου θα πρέπει να έχουν πρόσβαση όλοι άλλοι host του Διαδικτύου.

Για όλους τους παραπάνω λόγους το DNS σχεδιάστηκε να είναι *κατανεμημένο (distributed)*. Το DNS χρησιμοποιεί έναν μεγάλο αριθμό από εξυπηρετητές ονομάτων, οργανωμένους σε ιεραρχία και κατανεμημένους σε όλο το κόσμο. Κανένας εξυπηρετητής ονομάτων δεν περιέχει όλες τις αντιστοιχίες όλων των host του Διαδικτύου, αλλά οι αντιστοιχίες είναι κατανεμημένες στους διάφορους εξυπηρετητές ονομάτων. Με μία πρώτη προσέγγιση υπάρχουν τρεις τύποι εξυπηρετητών ονομάτων: **τοπικοί (local)**, **δρομολόγησης (root)** και **επίσημοι (authoritative)**. Οι τρεις τύποι αυτοί εξυπηρετητών ονομάτων επικοινωνούν μεταξύ τους αλλά και με τους host όπως περιγράφεται παρακάτω.

- **Τοπικοί Εξυπηρετητές Ονομάτων:** Κάθε παροχέας υπηρεσιών Διαδικτύου (Internet Service Provider – ISP), όπως ένα πανεπιστήμιο, έχει έναν τοπικό εξυπηρετητή ονομάτων, ο οποίος αποκαλείται και *προκαθορισμένος (default)*. Όταν ένας host κάνει μία DNS επερώτηση αυτή στέλνεται πρώτα στον τοπικό εξυπηρετητή ονομάτων του host. Συνήθως η IP διεύθυνση του εξυπηρετητή αυτού δίνεται από τον ίδιο το χρήστη στις ρυθμίσεις της πρόσβασης στο Διαδίκτυο. Ο τοπικός εξυπηρετητής ονομάτων συνήθως βρίσκεται πολύ κοντά στο host του χρήστη και καλύπτουν τις επερωτήσεις για άλλους host του ίδιου ISP, χωρίς την παρεμβολή κάποιου άλλου εξυπηρετητή ονομάτων.
- **Εξυπηρετητές Ονομάτων Δρομολόγησης:** Όταν ένας τοπικός εξυπηρετητής ονομάτων δεν περιέχει εγγραφή για το αιτούμενο όνομα host, τότε λειτουργεί ως ένας DNS πελάτης και κάνει επερώτηση σε έναν από τους εξυπηρετητές ονομάτων δρομολόγησης. Αν ο τελευταίος έχει μία εγγραφή για το όνομα του host την στέλνει ως απάντηση στον τοπικό εξυπηρετητή ονομάτων, και ακολούθως ο τοπικός εξυπηρετητής ονομάτων στέλνει την απάντηση στον αιτούντα host. Σε διαφορετική περίπτωση στέλνει την αίτηση στον επίσημο εξυπηρετητή ονομάτων που είναι υπεύθυνος για το συγκεκριμένο όνομα host, και του οποίου την διεύθυνση IP γνωρίζει εκ των προτέρων.
- **Επίσημοι Εξυπηρετητές Ονομάτων:** Κάθε host είναι εγγεγραμμένος σε κάποιον επίσημο εξυπηρετητή ονομάτων. Συνήθως ο επίσημος εξυπηρετητής ονομάτων είναι ο εξυπηρετητής ονομάτων του τοπικού ISP. Για μεγαλύτερη αξιοπιστία κάθε host εγγράφεται σε τουλάχιστον δύο επίσημους εξυπηρετητές ονομάτων.

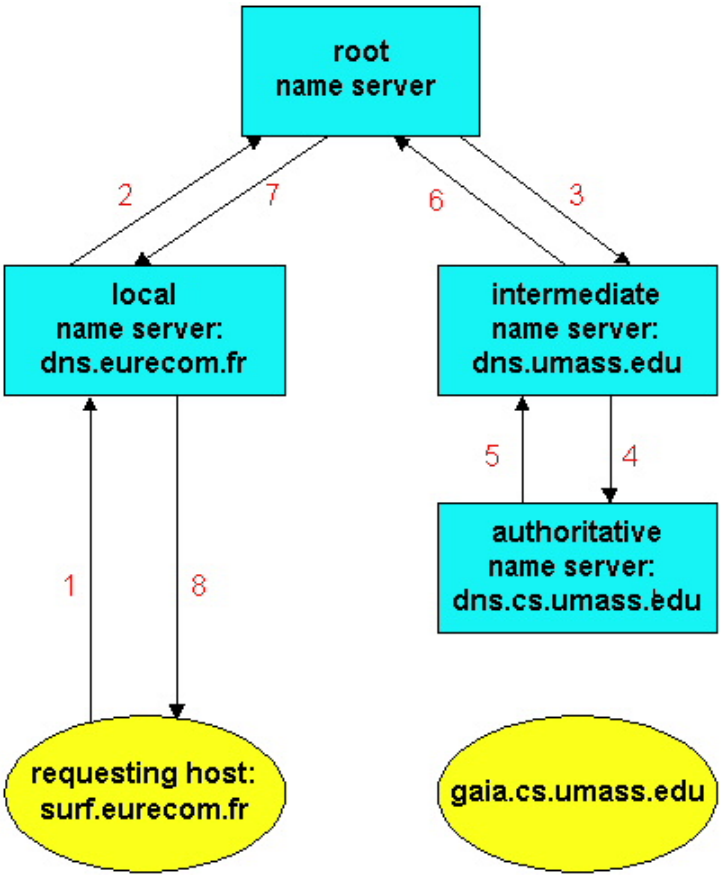
Κάθε επίσημος εξυπηρετητής ονομάτων έχει πάντοτε μία έγκυρη αντιστοιχία μεταξύ του ονόματος του host και της IP διεύθυνσής του, και όταν δεχθεί επερώτηση από κάποιον εξυπηρετητή ονομάτων δρομολόγησης ανταποκρίνεται με την εγγραφή αυτή. Στη συνέχεια ο εξυπηρετητής ονομάτων δρομολόγησης προωθεί την εγγραφή αυτή στον τοπικό εξυπηρετητή ονομάτων και ο τελευταίος με τη σειρά του την προωθεί στον αιτούντα host.

Ας εξετάσουμε ένα απλό παράδειγμα. Έστω ότι ο host surf.eurecom.fr θέλει την διεύθυνση IP του gaia.cs.umass.edu, και ότι ο τοπικός εξυπηρετητής ονομάτων του πρώτου host είναι ο dns.eurecom.fr και ο επίσημος εξυπηρετητής ονομάτων του δεύτερου ο dns.umass.edu. Στην αρχή ο host surf.eurecom.fr στέλνει μία ερώτηση DNS στον τοπικό εξυπηρετητή ονομάτων dns.eurecom.fr, η οποία περιέχει το όνομα του host για τον οποίο πρέπει να γίνει μετάφραση διευθύνσεων, που στο παράδειγμά μας είναι ο gaia.cs.umass.edu. Ο τοπικός εξυπηρετητής ονομάτων προωθεί την ερώτηση σε έναν εξυπηρετητή ονομάτων δρομολόγησης, ο οποίος με τη σειρά του την προωθεί στον επίσημο εξυπηρετητή ονομάτων της περιοχής umass.edu, που στο παράδειγμά μας είναι ο dns.umass.edu. Ο τελευταίος στέλνει την απάντηση πίσω στον αιτούντα host, μέσω του εξυπηρετητή ονομάτων δρομολόγησης και του τοπικού εξυπηρετητή ονομάτων. Στο παράδειγμα αυτό φαίνεται ότι για να βρεθεί η αντιστοιχία διευθύνσεων χρειάστηκαν να σταλθούν έξι DNS μηνύματα: τρεις ερωτήσεις και τρεις αποκρίσεις. Η σειρά των μηνυμάτων φαίνεται στο ακόλουθο σχήμα.



Στην παραπάνω ανάλυση υποτέθηκε ότι ο εξυπηρετητής ονομάτων δρομολόγησης γνωρίζει έναν επίσημο εξυπηρετητή ονομάτων για κάθε όνομα host. Η υπόθεση αυτή πολλές φορές είναι αναληθής, αφού για κάποιο δοσμένο όνομα host ο εξυπηρετητής ονομάτων δρομολόγησης μπορεί να μην γνωρίζει τον επίσημο εξυπηρετητή ονομάτων αλλά κάποιον **ενδιάμεσο (intermediate)** εξυπηρετητή ονομάτων ο οποίος με τη σειρά του γνωρίζει την διεύθυνση IP του επίσημου εξυπηρετητή ονομάτων για το συγκεκριμένο όνομα του host. Ας υποθέσουμε στο παραπάνω παράδειγμα ότι ο επίσημος εξυπηρετητής ονομάτων του gaia.cs.umass.edu είναι ο dns.cs.umass.edu και ότι ο dns.umass.edu γνωρίζει την IP διεύθυνση του dns.cs.umass.edu. Στην περίπτωση αυτή

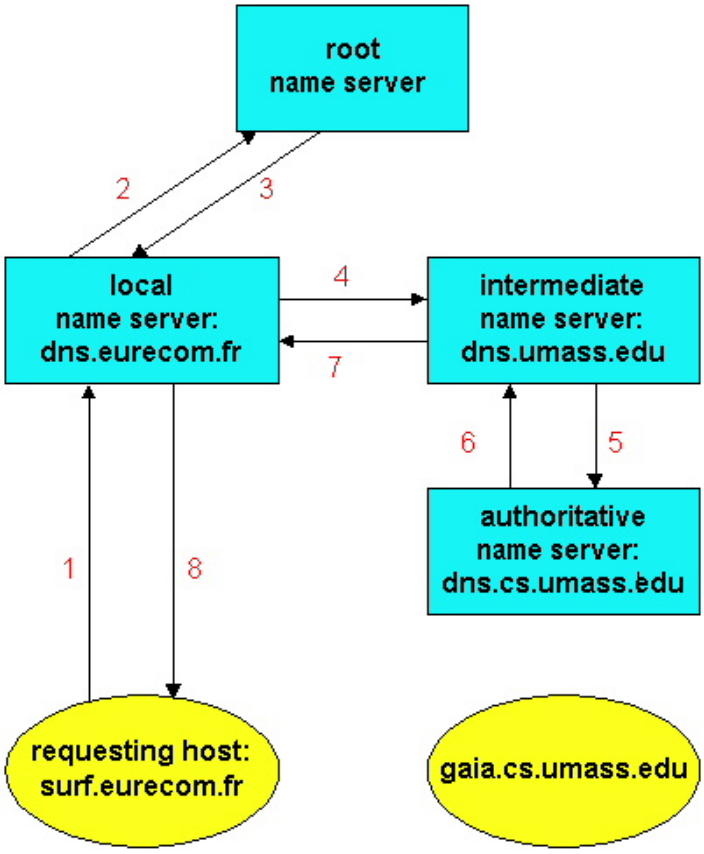
όταν ο εξυπηρετητής ονομάτων δρομολόγησης λάβει την επερώτηση θα την προωθήσει στον ενδιάμεσο εξυπηρετητή ονομάτων dns.umass.edu και αυτός με την σειρά του στον επίσημο εξυπηρετητή ονομάτων dsn.cs.umass.edu. Η ακολουθία των μηνυμάτων DNS φαίνεται στο παρακάτω σχήμα.



Στην τελευταία περίπτωση είδαμε ότι για την εύρεση της αντιστοιχίας των διευθύνσεων στάλθηκαν συνολικά οκτώ μηνύματα DNS. Ο αριθμός αυτός μπορεί να είναι ακόμη μεγαλύτερος αφού μπορεί να υπάρχουν περισσότεροι από ένας ενδιάμεσοι εξυπηρετητές ονομάτων στην αλυσίδα μεταξύ του εξυπηρετητή ονομάτων δρομολόγησης και του επίσημου εξυπηρετητή ονομάτων.

Στα παραπάνω παραδείγματα υποτέθηκε ότι όλες οι ερωτήσεις είναι **αναδρομικές (recursive)**. Όταν ο host Α κάνει μία αναδρομική ερώτηση στον εξυπηρετητή ονομάτων Β, τότε ο εξυπηρετητής ονομάτων Β κάνει την αιτούμενη αντιστοίχηση για λογαριασμό του Α και στη συνέχεια την προωθεί στον Α. Το πρωτόκολλο DNS, όμως, επιτρέπει και **iterative** ερωτήσεις σε κάθε βήμα της αλυσίδας μεταξύ του αιτούντα host και του επίσημου εξυπηρετητή ονομάτων. Όταν ο εξυπηρετητής ονομάτων Α κάνει μία iterative ερώτηση στον εξυπηρετητή ονομάτων Β, αν ο Β δεν έχει την αιτούμενη αντιστοιχία τότε στέλνει αμέσως μία DNS απόκριση στον Α, η οποία περιέχει την διεύθυνση IP του επόμενου εξυπηρετητή ονομάτων στην αλυσίδα, έστω τον Γ. Στην περίπτωση αυτή ο Α στέλνει απευθείας την ερώτηση στον Γ. Στην αλυσίδα των ερωτήσεων κάποιες ερωτήσεις μπορεί να είναι αναδρομικές ενώ κάποιες άλλες iterative. Συνήθως, όμως, όλες οι ερωτήσεις είναι αναδρομικές εκτός από την επερώτηση από τον τοπικό

εξυπηρετητή ονομάτων προς τον εξυπηρετητή ονομάτων δρομολόγησης, η οποία είναι iterative. Η τελευταία ερώτηση δεν είναι αναδρομική για την αποφυγή επιπλέον επιβάρυνσης του εξυπηρετητή ονομάτων δρομολόγησης, ο οποίος δέχεται τεράστιο αριθμό DNS επερωτήσεων.
Στο παρακάτω σχήμα φαίνεται το παράδειγμα που εξετάστηκε παραπάνω, αλλά στην περίπτωση αυτή γίνεται χρήση iterative ερώτησης από τον τοπικό εξυπηρετητή ονομάτων προς τον εξυπηρετητή ονομάτων δρομολόγησης.



Στην παραπάνω ανάλυση δεν αναφερθήκαμε καθόλου σε ένα σημαντικό χαρακτηριστικό του DNS: την προσωρινή αποθήκευση στην ενδιάμεση μνήμη, **cache**. Στην πραγματικότητα το DNS χρησιμοποιεί την cache για τη μείωση του φορτίου στο δίκτυο αλλά και για την μείωση της καθυστέρησης απόκρισης στις DNS επερωτήσεις. Όταν ένας εξυπηρετητής ονομάτων λαμβάνει μία αντιστοιχία για ένα όνομα host, την αποθηκεύει στην τοπική του μνήμη (π.χ. στην μνήμη RAM ή στον τοπικό δίσκο). Αν φτάσει στον εξυπηρετητή ονομάτων κάποια άλλη ερώτηση για ένα όνομα host που είναι καταχωρημένο στην προσωρινή μνήμη του εξυπηρετητή ονομάτων τότε ο τελευταίος παρέχει την διεύθυνση IP που είναι αποθηκευμένη στην μνήμη του. Κάθε εγγραφή της προσωρινής μνήμης μετά από κάποιο δεδομένο χρονικό διάστημα διαγράφεται, για να αντιμετωπιστεί και οι περίπτωση των host με μικρή διάρκεια ζωής.
Εγγραφές πόρων

Οι εξυπηρετητές ονομάτων που υλοποιούν την βάση δεδομένων του DNS αποθηκεύουν **εγγραφές πόρων (resource records)** για τις αντιστοιχίες μεταξύ των ονομάτων των host και των διευθύνσεων IP. Κάθε απόκριση DNS περιέχει μία ή περισσότερες εγγραφές πόρων.

Κάθε εγγραφή πόρων αποτελείται από την ακόλουθη τετράδα πεδίων

(Name, Value, Type, TTL)

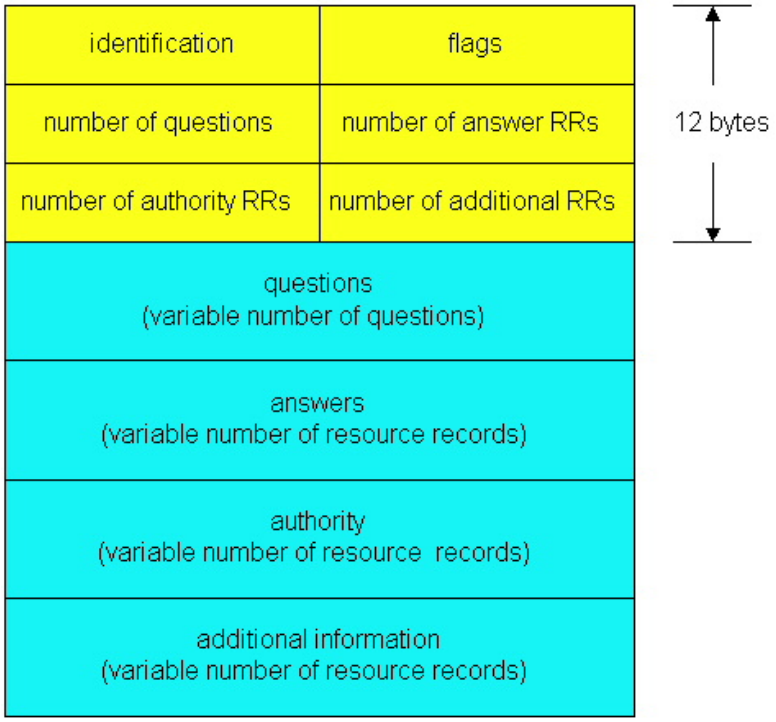
Το πεδίο TTL δίνει το χρόνο ζωής (time to live) της εγγραφής πόρων, μετά τη λήξη του οποίου η εγγραφή πρέπει να διαγραφεί από την προσωρινή μνήμη. Η σημασία των πεδίων Value και Name εξαρτώνται από τον τύπο Type της εγγραφής.

- Αν Type=**A** τότε το πεδίο Name είναι το όνομα του host και το πεδίο Value η διεύθυνση IP για το συγκεκριμένο όνομα host. Δηλαδή, οι εγγραφές τύπου A παρέχουν τη στάνταρ αντιστοιχία μεταξύ ονομάτων host και διευθύνσεων IP.
- Αν Type=**NS** τότε το πεδίο Name είναι το όνομα της περιοχής – domain (π.χ. ntua.gr) και το πεδίο Value το όνομα του εξυπηρετητή που γνωρίζει πως να ανακτά τις διευθύνσεις IP των host της ίδιας περιοχής. Η εγγραφή αυτή χρησιμοποιείται για τη δρομολόγηση των ερωτήσεων DNS μέσα στην ιεραρχία των δρομολογητών ονομάτων.
- Αν Type=**CNAME** τότε το πεδίο Value είναι το κανονικό όνομα για το ψευδώνυμο Name. Οι εγγραφές αυτές χρησιμεύουν στη δημιουργία ψευδωνύμων για τα ονόματα των host του Διαδικτύου.
- Αν Type=**MX** τότε το πεδίο Value είναι το όνομα ενός εξυπηρετητή ταχυδρομείου ο οποίος έχει ψευδώνυμο Name. Οι εγγραφές αυτές επιτρέπουν τα ονόματα των host των εξυπηρετητών ταχυδρομείου να έχουν απλά ψευδώνυμα.

Αν ένας εξυπηρετητής ονομάτων είναι επίσημος για κάποιον host τότε θα περιέχει μία εγγραφή με Τύπο (Type) A για το όνομα του host αυτού (ακόμη και αν ο εξυπηρετητής ονομάτων δεν είναι ο επίσημος για κάποιον host μπορεί να περιέχει μία εγγραφή στην προσωρινή του μνήμη). Σε περίπτωση που ένας εξυπηρετητής ονομάτων δεν είναι ο επίσημος για κάποιον host για τον οποίο περιέχει εγγραφή τότε πρέπει να περιέχει και μία εγγραφή με Τύπο NS για τον εξυπηρετητή ονομάτων της περιοχή του host αυτού. Επίσης, για τον εξυπηρετητή αυτό θα περιέχει και μία εγγραφή με Τύπο A για την αντιστοιχία του ονόματός του με την IP διεύθυνσή του.

2.2. DNS Μηνύματα

Το πρωτόκολλο DNS έχει δύο μόνο τύπους μηνυμάτων: τις ερωτήσεις (queries) και τις αποκρίσεις (replies). Οι δύο τύποι DNS μηνυμάτων έχουν την ίδια μορφή, η οποία φαίνεται στο ακόλουθο σχήμα.

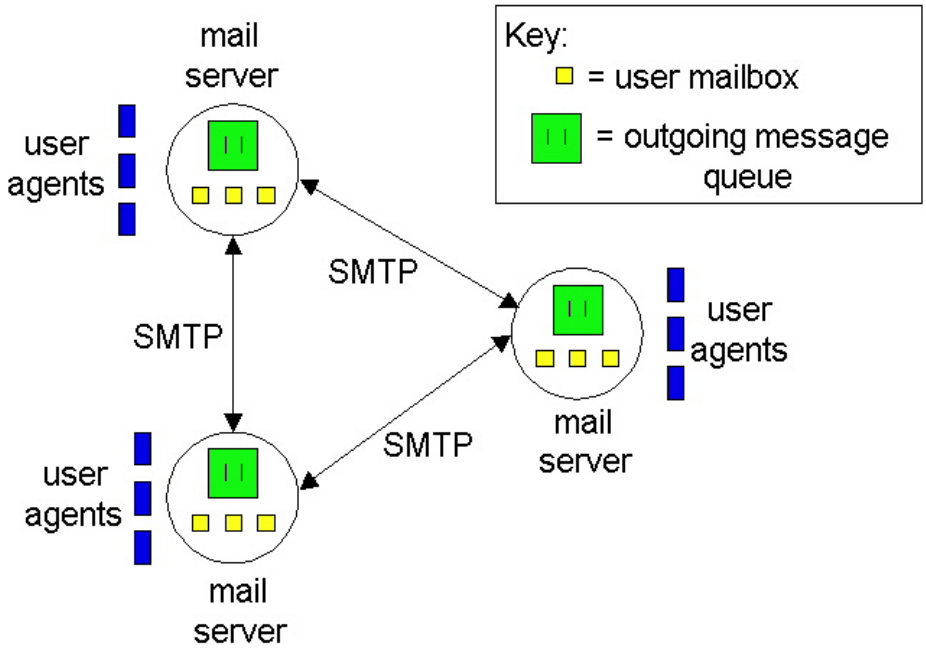


- Τα πρώτα 12 bytes είναι το *τμήμα επικεφαλίδας (header section)*, το οποίο περιέχει διάφορα πεδία. Το πεδίο *Identification*, που είναι 16 bit, αντιγράφεται από τις ερωτήσεις στις αντίστοιχες αποκρίσεις ώστε ο πελάτης που έστειλε την ερώτηση να γνωρίζει σε ποια ερώτηση αντιστοιχεί η συγκεκριμένη απόκριση. Στο πεδίο *flags* περιέχονται διάφορες σημαίες του 1 bit, όπως η σημαία query/reply που χρησιμοποιείται για την αναγνώριση του τύπου του DNS μηνύματος (το 0 αντιστοιχεί σε επερώτηση και το 1 σε απόκριση), η σημαία authoritative που έχει τιμή 1 αν ο εξυπηρετητής ονομάτων που στέλνει την απόκριση είναι ο επίσημος για το αιτούμενο όνομα host, η σημαία recursion-desired που τίθεται στην τιμή 1 αν ένας host θέλει να χρησιμοποιήσει αναδρομικές επερωτήσεις και η σημαία recursion available που τίθεται στην τιμή 1 στις αποκρίσεις για τις οποίες ο εξυπηρετητής ονομάτων έχει δυνατότητα υποστήριξης αναδρομής. Στο τμήμα τις επικεφαλίδας περιέχονται επίσης *τέσσερα πεδία των 16 bits* στα οποία δηλώνεται ο αριθμός των στοιχείων του καθενός από τα τέσσερα τμήματα που ακολουθούν το τμήμα της επικεφαλίδας.
- Το *τμήμα ερωτήσεων (question section)* περιέχει πληροφορία σχετικά με την επερώτηση που γίνεται. Το τμήμα αυτό περιλαμβάνει ένα πεδίο ονόματος που περιέχει το όνομα για το οποίο γίνεται η ερώτηση και ένα πεδίο στο οποίο δηλώνεται ο τύπος της επερώτησης (π.χ. αν θέλουμε την διεύθυνση IP ή το όνομα του εξυπηρετητή ταχυδρομείου).
- Στο *τμήμα απάντησης (answer section)* περιέχονται οι εγγραφές πόρων για το όνομα για το οποίο έγινε η ερώτηση. Μία απόκριση μπορεί να περιέχει περισσότερες από μία εγγραφές πόρων.
- Το *τμήμα πιστοποίησης (authority section)* που περιέχει εγγραφές από άλλους επίσημους εξυπηρετητές ονομάτων.

- Το *επιπλέον τμήμα (additional section)* περιέχει άλλες χρήσιμες εγγραφές. Π.χ. αν το τμήμα απάντησης είναι η απόκριση σε μία MX επερώτηση το τμήμα αυτό μπορεί να περιέχει το όνομα του host του εξυπηρετητή ταχυδρομείου που σχετίζεται με το ψευδώνυμο Name καθώς και την διεύθυνση IP του κανονικού ονόματος host του εξυπηρετητή ταχυδρομείου.

3. Ηλεκτρονικό Ταχυδρομείο

Το **ηλεκτρονικό ταχυδρομείο**, ή απλά **e-mail**, είναι σε χρήση για περισσότερο από δύο δεκαετίας. Μέχρι το 1990 χρησιμοποιούνταν σχεδόν αποκλειστικά από την ακαδημαϊκή κοινότητα, ενώ τα τελευταία χρόνια έχει γίνει ιδιαίτερα δημοφιλές σε όλους τους χρήστες του Διαδικτύου. Αυτό φαίνεται και από το μεγάλο πλήθος e-mails που στέλνονται καθημερινά, τα οποία συνολικά ξεπερνούν τον αριθμό των γραμμάτων που στέλνονται μέσω του παραδοσιακού ταχυδρομείου.
Στις μέρες μας τα πρωτόκολλα του ηλεκτρονικού ταχυδρομείου έχουν εξελιχθεί σε μεγάλο βαθμό (σε σχέση με το SMTP που επέτρεπε τη μεταφορά μόνο απλού κειμένου) ώστε τα e-mails μπορεί να περιέχουν όχι μόνο απλό κείμενο αλλά εικόνες, animation, ήχους και γενικά ότι σήμερα καλύπτει ο όρος πολυμέσα. Στο παρακάτω σχήμα βλέπουμε τη βασική αρχιτεκτονική που χρησιμοποιείται για την υπηρεσία του ηλεκτρονικού ταχυδρομείου.



Στο παραπάνω σχήμα βλέπουμε τα τρία βασικά στοιχεία της αρχιτεκτονικής του ηλεκτρονικού ταχυδρομείου: τους **αντιπροσώπους χρηστών (user agents)**, τους **εξυπηρετητές ταχυδρομείου (mail servers)** και το **πρωτόκολλο μεταφοράς** των ίδιων των μηνυμάτων **SMTP (Simple Mail Transfer Protocol)**. Οι αντιπρόσωποι χρηστών επιτρέπουν στους διάφορους χρήστες να συνθέσουν, να διαβάσουν και να στείλουν e-mail. Όταν ένας χρήστης στέλνει ένα e-mail αυτό φθάνει πρώτα σε κάποιον εξυπηρετητή ταχυδρομείου, όπου και αποθηκεύεται σε μία **εξερχόμενη ουρά μηνυμάτων (outgoing message queue)**. Όταν ο χρήστης θέλει να διαβάσει κάποιο μήνυμα τότε αυτό μεταφέρεται από το **mailbox** του χρήστη (το οποίο βρίσκεται στον εξυπηρετητή μηνυμάτων) στον υπολογιστή του χρήστη.

Το σύστημα e-mail επιτελεί βασικά τις ακόλουθες πέντε βασικές λειτουργίες.

- **Σύνθεση (Composition):** Αναφέρεται στη λειτουργία δημιουργίας μηνυμάτων και απαντήσεων. Αν και, όπως θα δούμε παρακάτω, τα μηνύματα e-mail αποτελούνται από απλό κείμενο, ο αντιπρόσωπος του χρήστη μπορεί να τον βοηθήσει στην συμπλήρωση κάποιων στοιχείων, π.χ. στην περίπτωση που θέλουμε να απαντήσουμε σε κάποιο μήνυμα ο αντιπρόσωπος χρήστη μπορεί να συμπληρώσει αυτόματα τη διεύθυνση αποστολής της απάντησης.
- **Μεταφορά (Transfer):** Αναφέρεται στην μεταφορά των μηνυμάτων από την πηγή προς τον προορισμό. Για το σκοπό αυτό εγκαθίσταται μία σύνδεση μεταξύ της πηγής και του προορισμού, συνήθως με χρήση και ενδιάμεσων κόμβων, με μία εντολή του χρήστη. Δηλαδή, από τη πλευρά του χρήστη αρκεί να δοθεί μία εντολή, π.χ. στείλε το μήνυμα, για να εκτελεστεί η ενέργεια αυτή αυτόματα από το σύστημα e-mail.
- **Αναφορά (Reporting):** Ο αποστελέας του μηνύματος ενημερώνεται για το τι έγινε με το μήνυμα, δηλαδή αν έφθασε στον τελικό του προορισμό, αν χάθηκε ή αν απορρίφθηκε από κάποιον ενδιάμεσο κόμβο. Σε πολλές περιπτώσεις η λειτουργία αυτή μπορεί να είναι πολύ χρήσιμη, π.χ. σε περιπτώσεις που θέλουμε να βεβαιωθούμε ότι ο παραλήπτης έλαβε το μήνυμα.
- **Παρουσίαση (Displaying):** Αναφέρεται στον τρόπο με τον οποίο παρουσιάζονται τα περιεχόμενα του μηνύματος στον χρήστη, για να μπορεί ο χρήστης να τα διαβάσει εύκολα.
- **Διευθέτηση (Disposition):** Αναφέρεται με το τι θα κάνει το μήνυμα ο παραλήπτης του, π.χ. μπορεί να το διαγράψει πριν ή μετά την ανάγνωσή του ή να το αποθηκεύσει.

Εκτός από τος παραπάνω βασικές λειτουργίες, ο αντιπρόσωπος του χρήστη μπορεί να έχει και κάποιες πιο εξειδικευμένες: μπορεί να έχει δυνατότητα **προώθησης (forwarding)** μηνυμάτων, παραμετροποίησης του mailbox, δημιουργίας **λίστας χρηστών (mailing list)** ή αποστολής του ίδιου μηνύματος προς πολλούς χρήστες ταυτόχρονα.

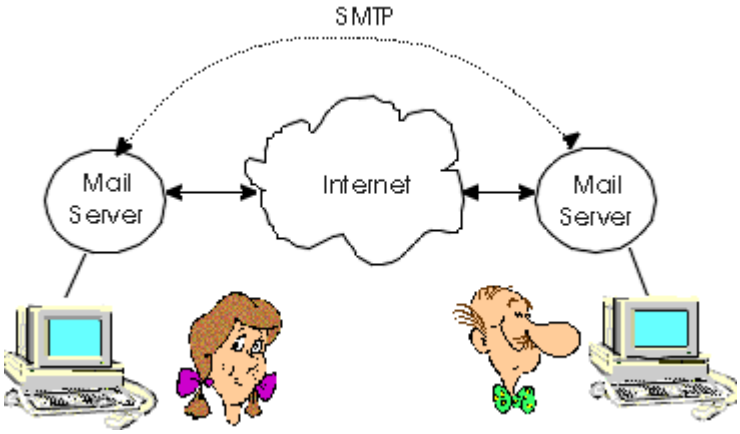
3.1. SMTP

Το **SMTP (Simple Mail Transfer Protocol)** είναι το πρωτόκολλο του στρώματος εφαρμογών που χρησιμοποιείται για την μεταφορά των μηνυμάτων του ηλεκτρονικού ταχυδρομείου. Το SMTP χρησιμοποιεί ως πρωτόκολλο του στρώματος μεταφοράς το **TCP** και συγκεκριμένα τη θύρα **25** και βασικά έχει δύο τμήματα: την **πλευρά του πελάτη (client side),** που είναι ο εξυπηρετητής ταχυδρομείου του αποστολέα και την **πλευρά του εξυπηρετητή (server side),** που είναι ο εξυπηρετητής ταχυδρομείου του παραλήπτη. Ο κάθε εξυπηρετητής ταχυδρομείου μπορεί να παίξει το ρόλο είτε του πελάτη είτε του εξυπηρετητή: όταν αποστέλλει ένα μήνυμα έχει το ρόλο του πελάτη, ενώ όταν δέχεται μηνύματα έχει το ρόλο του εξυπηρετητή.

Το SMTP είναι ένα πρωτόκολλο χαρακτήρων **ASCII των 7-bit**. Ο λόγος αυτής της επιλογής βρίσκεται πίσω στο 1982 όταν εκδόθηκε το πρωτόκολλο SMTP. Την εποχή εκείνη τα μηνύματα του ηλεκτρονικού ταχυδρομείου αποτελούνταν μόνο από απλό κείμενο και η επιλογή της κωδικοποίησης των χαρακτήρων με 7-bit ήταν λογική. Στις

μέρες μας όμως τα μηνύματα του ηλεκτρονικού ταχυδρομείου δεν περιλαμβάνουν μόνο απλό κείμενο αλλά και εικόνες, ήχους κτλ. τα οποία κωδικοποιούνται με 8-bit. Επομένως, για την μεταφορά τους χρειάζεται πρώτα η μετατροπή τους στα 7-bit, η αποστολή στον παραλήπτη σε μορφή 7-bit και η μετατροπή τους από τον παραλήπτη πίσω στην μορφή των 8 bit.
Θα εξετάσουμε το πρωτόκολλο SMTP με ένα παράδειγμα τυπικής αποστολής ενός μηνύματος ηλεκτρονικού ταχυδρομείου. Έστω ότι η Alice στέλνει ένα e-mail στον Bob. Τότε ακολουθείτε η ακόλουθη διαδικασία:

- Η Alice ξεκινάει το πρόγραμμα για την αποστολή e-mail, που στην ουσία είναι ένας αντιπρόσωπος χρήστη, στο οποίο παρέχει τη διεύθυνση e-mail του Bob και με το οποίο συνθέτει το μήνυμα. Τέλος, η Alice δίνει την εντολή για αποστολή του μηνύματος.
- Ο αντιπρόσωπος χρήστη της Alice στέλνει το μήνυμα στον αντιπρόσωπος ταχυδρομείου της, όπου μπαίνει σε στην εξερχόμενη ουρά αποστολής.
- Ο αντιπρόσωπος ταχυδρομείου της Alice βλέπει το μήνυμα στην ουρά και ενεργώντας ως πελάτης ανοίγει μία TCP σύνδεση με τον πράκτορα ταχυδρομείου του Bob στη θύρα 25, μέσω της οποίας στέλνονται τα SMTP μηνύματα.
- Μετά από κάποιου είδους χειραψίας (handshake), η οποία θα περιγραφεί αναλυτικά παρακάτω, στέλνεται το μήνυμα στον αντιπρόσωπο ταχυδρομείου του Bob.
- Η πλευρά εξυπηρετητή του εξυπηρετητή ταχυδρομείου του Bob λαμβάνει το μήνυμα και το τοποθετεί στο mailbox του Bob.
- Ο Bob χρησιμοποιώντας τον αντιπρόσωπο χρήστη διαβάζει το μήνυμα.



Μία σημαντική παρατήρηση: το μήνυμα δεν αποθηκεύεται σε κάποιον ενδιάμεσο εξυπηρετητή ταχυδρομείου του Διαδικτύου και για την αποστολή του μέσω του SMTP ανοίγεται μία απευθείας σύνδεση μεταξύ του εξυπηρετητή ταχυδρομείου του αποστολέα και του εξυπηρετητή ταχυδρομείου του παραλήπτη.
Ας δούμε αναλυτικά τη λειτουργία του SMTP. Στην αρχή ανοίγεται μία TCP σύνδεση στη θύρα 25 μεταξύ των εξυπηρετητών ταχυδρομείου του αποστολέα και του παραλήπτη. Στη συνέχεια πραγματοποιείτε μία μορφή χειραψίας: Στην αρχή οι δύο εξυπηρετητές ταχυδρομείου γνωστοποιούν ο ένας στον άλλον τη διεύθυνσή τους. Στη συνέχεια η πλευρά του πελάτη του εξυπηρετητή ταχυδρομείου του αποστολέα γνωστοποιεί στην πλευρά εξυπηρετητή του εξυπηρετητή ταχυδρομείου του παραλήπτη

τις e-mail διευθύνσεις του αποστολέα και του παραλήπτη. Στη συνέχεια στέλνεται το ίδιο το μήνυμα. Αν ο πελάτης έχει και να στείλει και άλλα μηνύματα προς τον ίδιο εξυπηρερτητή ταχυδρομείου τότε τα στέλνει μέσω της ίδιας TCP σύνδεσης. Μετά την αποστολή και του τελευταίου μηνύματος η πλευρά του πελάτη τερματίζει την σύνδεση TCP.
Παρακάτω φαίνεται ένα παράδειγμα αποστολής μεταξύ ενός πελάτη (c) και ενός εξυπηρετητή (S).

```
S: 220 yahoo.gr
C: HELO mail.gr
S: 250  Hello mail.gr, pleased to meet you
C: MAIL FROM: <alice@mail.gr>
S: 250 alice@mail.gr... Sender ok
C: RCPT TO: <bob@yahoo.gr>
S: 250 bob@yahoo.gr ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Test
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 yahoo.gr closing
```

Στο παραπάνω παράδειγμα ο πελάτης κάνει χρήση πέντε εντολών: **HELO**, **MAIL, FROM**, **RCPT TO**, **DATA**, **QUIT**, ενώ ο εξυπηρετητής απαντά με αντίστοιχες αποκρίσεις που περιέχουν κωδικούς κατάστασης και αντίστοιχα μηνύματα κατάστασης. Στην περίπτωση που θα αποσταλούν περισσότερα του ενός μηνύματα προς τον ίδιο εξυπηρετητή τότε η εντολή QUIT πραγματοποιείτε μετά την αποστολή και του τελευταίου μηνύματος και το κάθε μήνυμα ξεκινά από την εντολή HELO.
Τέλος θα αναφέρουμε και κάποια σημαντικά χαρακτηριστικά του SMTP που το κάνουν να διαφέρει από άλλα πρωτόκολλα του στρώματος εφαρμογών του Διαδικτύου. Το SMTP είναι ένα πρωτόκολλο με το οποίο ο πελάτης μπορεί να γράψει δεδομένα στον εξυπηρετητή, σε αντίθεση με την πλειονότητα των πρωτοκόλλων του στρώματος εφαρμογών (εξαίρεση αποτελεί το FTP). Επιπλέον, δεν δίνεται το μήκος του μηνύματος αλλά χρειάζεται η αποστολή μίας γραμμή κειμένου που να περιέχει μόνο μία τελεία για τη δήλωση του τέλους του μηνύματος. Τέλος, σε σχέση με το HTTP, παρά το γεγονός ότι ένα μήνυμα μπορεί να αποτελείται από πολλά διαφορετικά τμήματα στέλνεται ως ένα μόνο μήνυμα. Για το διαχωρισμό των διαφόρων τμημάτων γίνεται χρήση άλλων πρωτοκόλλων περιγραφής περιεχομένου.

3.2. Μορφή των μηνυμάτων – Πρωτόκολλο MIME

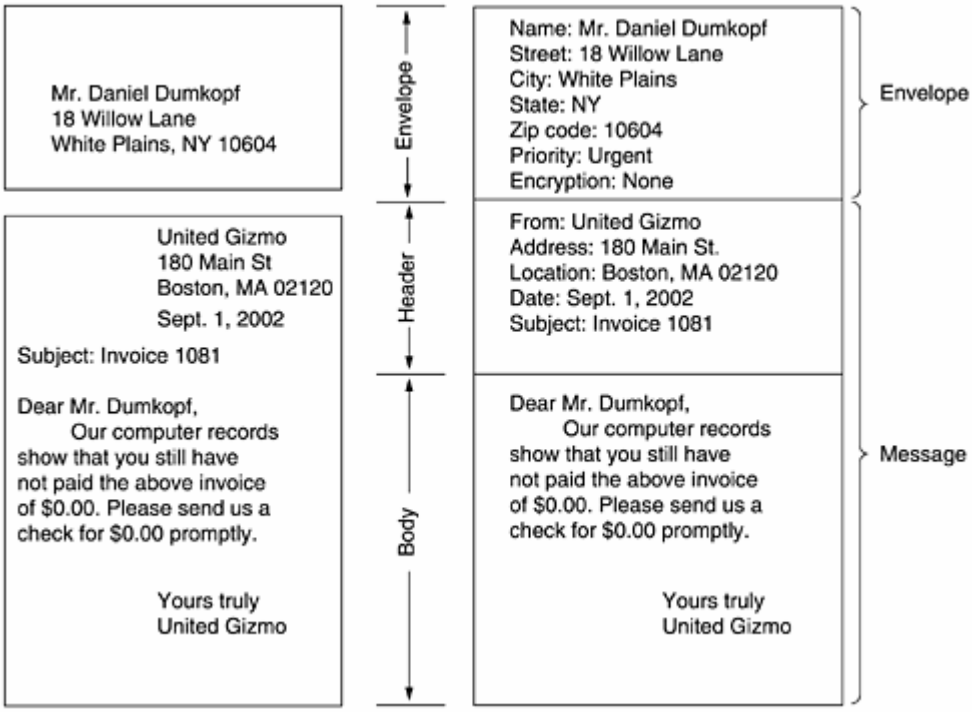
Παραπάνω αναφέρθηκε ότι το ηλεκτρονικό ταχυδρομείο χρησιμοποιεί κωδικοποίηση ASCII των 7-bit. Αν και η επιλογή αυτή ήταν ικανοποιητική στα πρώτα χρόνια της

εμφάνισης του ηλεκτρονικού ταχυδρομείου στις μέρες μας δεν είναι αρκετή για τους ακόλουθους λόγους:

- Τα μηνύματα μπορεί να είναι σε γλώσσες οι οποίες δεν χρησιμοποιούν μόνο το Λατινικό σετ χαρακτήρων ή ακόμη και σε γλώσσες που δεν χρησιμοποιούν καθόλου αλφάβητο (π.χ. τα Κινέζικα).
- Τα μηνύματα μπορεί να μην αποτελούνται μόνο από απλό κείμενο, αλλά να περιέχουν εικόνες, animation, βίντεο και γενικά κάθε είδους πολυμέσα.

Για την κάλυψη των παραπάνω αναγκών προτάθηκε ένα νέο πρωτόκολλο, το πρωτόκολλο Επεκτάσεων Ταχυδρομείου Διαδικτύου για Πολλαπλούς Σκοπούς **MIME (Multipurpose Internet Mail Extensions)**. Η βασική ιδέα του MIME είναι η συνέχιση της χρήσης της μορφής του σώματος των μηνυμάτων αλλά η προσθήκη δομής σε αυτήν καθώς και η παροχή κανόνων για την κωδικοποίηση μη ASCII χαρακτήρων. Επομένως το MIME αποτελεί μία επέκταση του **RFC 822** το οποίο περιέγραφε τη δομή του σώματος των μηνυμάτων.

Η βασική δομή ενός μηνύματος, σύμφωνα με το RFC 822, φαίνεται στο παρακάτω σχήμα. Στο σχήμα αυτό φαίνεται και η αντιστοιχία σε σχέση με ένα γράμμα του ταχυδρομείου.



Το RFC 822 χρησιμοποιεί γραμμές επικεφαλίδων που αποτελούνται από ένα αναγνωριστικό, ένα : και την αντίστοιχη τιμή του αναγνωριστικού. Κάθε επικεφαλίδα πρέπει να περιέχει μία γραμμή **From:** και μία γραμμή **To:** στις οποίες δηλώνονται αντίστοιχα ο αποστολέας και ο παραλήπτης του μηνύματος. Το μήνυμα μπορεί επίσης να περιλαμβάνει και μία γραμμή επικεφαλίδας **Subject:** για να δηλωθεί το θέμα του μηνύματος. Επομένως, η βασική δομή του τμήματος επικεφαλίδας του μηνύματος που δόθηκε κατά την περιγραφή του SMTP μπορεί να είναι η ακόλουθη.

```
From: alice@mail.gr
To: bob@yahoo.gr
Subject: test message.
```

Μετά την επικεφαλίδα του μηνύματος ακολουθεί μία κενή γραμμή και στη συνέχεια το **σώμα του μηνύματος (message body)**. Το μήνυμα τελειώνει με μία γραμμή που περιέχει μόνο μία τελεία.
Το MIME δίνει και τη δυνατότητα για την αποστολή μηνυμάτων που δεν περιέχουν μόνο απλό κείμενο κωδικοποιημένο σε 7-bit ASCII. Για την αποστολή αυτών των μηνυμάτων πρέπει να γίνει χρήση επιπλέον επικεφαλίδων. Δύο βασικές επικεφαλίδες είναι η **Content-Type:** και η **Content-Transfer-Encoding:** Η πρώτη επιτρέπει σε έναν χρήστη να κάνει την πρέπουσα ενέργεια ανάλογα με το περιεχόμενο του μηνύματος, ενώ η δεύτερη δηλώνει το είδος της κωδικοποίησης που έχει εφαρμοστεί στα περιεχόμενα του μηνύματος. Επομένως, όταν ένας αντιπρόσωπος χρήστη λάβει ένα μήνυμα που περιέχει τις παραπάνω δύο επικεφαλίδες πρώτα ελέγχει τα περιεχόμενα της επικεφαλίδας Content-Transfer-Encoding: για να αποκωδικοποιήσει τα δεδομένα και στη συνέχεια την content-Type: για να χρησιμοποιήσει την κατάλληλη ενεργεία ανάλογα με τον τύπο των περιεχομένων του μηνύματος.
Παρακάτω φαίνεται ένα παράδειγμα χρήσης του MIME για την αποστολή μίας εικόνας τύπου JPEG.

```
From: alice@mail.gr
To: bob@yahoo.gr
Subject: Picture.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.........................
......base64 encoded data

.
```

Στο παραπάνω μήνυμα βλέπουμε και τη χρήση της επικεφαλίδας **MIME-Version:** για την δήλωση της έκδοσης του MIME που χρησιμοποιήθηκε στο σώμα του μηνύματος. Το μήνυμα τελειώνει με μία γραμμή που περιέχει μόνο μία τελεία.
Υπάρχουν διάφοροι τύποι περιεχομένων που χρησιμοποιούνται από το MIME. Οι σημαντικότεροι από αυτούς είναι οι ακόλουθοι.

- **text:** Χρησιμοποιείται για να δηλώσει ότι το σώμα του μηνύματος περιέχει κείμενο. Μία υποκατηγορία του text είναι η text/plain στην οποία μπορεί να δωθεί ως παράμετρος η σύνολο χαρακτήρων που θα χρησιμοποιηθεί, π.χ. text/plain; charset="ISO-8859-1". Στην περίπτωση αυτή το κείμενο εμφανίζεται όπως είναι. Μία άλλη υποκατηγορία είναι η html, στην οποία το προς εμφάνιση κείμενο έχει τη μορφή HTML.

- **image:** Χρησιμοποιείται για την συμπερίληψη εικόνων και έχει δύο υποκατηγορίες, την image/gif και την image/jpeg ανάλογα αν η εικόνα είναι τύπου GIF ή JPEG, αντίστοιχα.
- **audio:** Χρησιμοποιείται για να δηλώσει ότι το περιεχόμενο είναι ένα αρχείο ήχου. Υποκατηγορίες είναι η βασική (8-bit mu-law κωδικοποίηση) και η 32Kadpcm.
- **video:** Χρησιμοποιείται για βίντεο και οι κύριες υποκατηγορίες του είναι οι mpeg και quicktime.
- **application:** Χρησιμοποιείται για δεδομένα που δεν ανήκουν σε καμιά από τις άλλες κατηγορίες. Μία χρήση της είναι για δεδομένα που θα επεξεργαστούν από κάποια εφαρμογή. Π.χ. για αρχεία τα οποία επεξεργάζονται από το Microsoft Word χρησιμοποιείται η υποκατηγορία application/msword.

Για μηνύματα που χρησιμοποιούνται συνδυασμοί από τα παραπάνω στοιχεία λέμε ότι είναι τύπου **multipart**. Όταν ένα μήνυμα περιέχει περισσότερα από ένα διαφορετικά στοιχεία τότε χρησιμοποιείται η γραμμή επικεφαλίδας **Content-Type=multipart/mixed**. Σε αυτή την περίπτωση ο παραλήπτης πρέπει να γνωρίζει που αρχίζει και που τελειώνει το κάθε τμήμα, πως κωδικοποιείται καθώς και τον τύπο του κάθε αντικειμένου. Ο παραπάνω διαχωρισμός γίνεται με την προσθήκη στο μήνυμα **χαρακτήρων συνόρων (boundary characters)** μεταξύ των αντικειμένων και με την συμπερίληψη σε κάθε αντικείμενο των επικεφαλίδων **Content-Type:** και **Content-transfer-Encoding:**
Παρακάτω φαίνεται ένα μήνυμα που αποτελείται από πολλά αντικείμενα.

```
From: alice@mail.gr
   To: bob@yahoo.gr
   Subject: Picture and explain
   MIME-Version: 1.0
   Content-Type: multipart/mixed; Boundary=StartOfNextPart
  --StartOfNextPart
   Dear Bob,
   The picture you requested.

   --StartOfNextPart
   Content-Transfer-Encoding: base64
   Content-Type: image/gif

   base64 encoded data .....
   .........................
   ......base64 encoded data
```

Από το παραπάνω παράδειγμα φαίνεται ότι στην επικεφαλίδα Content-Type: δηλώνεται ο τρόπος διαχωρισμού των διαφόρων τμημάτων του μηνύματος. Το διαχωριστικό πάντα ξεκινά με δύο παύλες και τελειώνει με αλλαγή γραμμής.
Όπως αναφέρθηκε και παραπάνω, ένα μήνυμα ηλεκτρονικού ταχυδρομείου αποτελείται από δύο τμήματα: την επικεφαλίδα και το σώμα του μηνύματος. Το σώμα του μηνύματος περιέχει τα δεδομένα και στην περίπτωση μηνύματος που αποτελείται από πολλά τμήματα περιέχει όλα τα τμήματα, κάνοντας χρήση των διαχωριστικών που

αναφέρθηκαν παραπάνω. Το τμήμα της επικεφαλίδας περιέχει επικεφαλίδες που ορίζονται τόσο στο RFC 822 όσο και επικεφαλίδες MIME. Όταν ο χρήστης λαμβάνει ένα μήνυμα αυτό περιέχει και κάποιες επιπλέον επικεφαλίδες, οι οποίες προστίθενται από τον SMTP εξυπηρετητή του παραλήπτη. Όλα τα λαμβανόμενα μηνύματα περιέχουν την επικεφαλίδα **Received:** στην κορυφή του μηνύματος, στην οποία δηλώνεται ο SMTP εξυπηρετητής του αποστολέα, ο SMTP εξυπηρετητής του παραλήπτη, καθώς και η χρονική στιγμή λήψης του μηνύματος από τον SMTP εξυπηρετητή του παραλήπτη. Ένα παράδειγμα μηνύματος που λαμβάνει ο παραλήπτης φαίνεται παρακάτω:

```
Received: from mail.gr by yahoo.gr ; 3 Jen 04 12:12:12 GMT
From: alice@mail.gr
To: bob@yahoo.gr
Subject: Picture.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
```

base64 encoded data .....
.....
.....base64 encoded data

Κάθε μήνυμα ηλεκτρονικού ταχυδρομείου που έχει ληφθεί, περιέχει τουλάχιστον μία επικεφαλίδα Received: περισσότερες από μία έχουμε στην περίπτωση που στη διαδρομή του μηνύματος περιέχονται και κάποιοι ενδιάμεσοι SMTP εξυπηρετητές, ο κάθε ένας από τους οποίους προσθέτει και μία επικεφαλίδα received: Αν στο παραπάνω παράδειγμα ο εξυπηρετητής mail.gr προωθούσε το ηλεκτρονικό ταχυδρομείο στον mail2004.gr και ο τελευταίος έστελνε το μήνυμα στον yahoo.gr, τότε ο παραλήπτης θα λάμβανε το μήνυμα με δύο επικεφαλίδες Received:

```
Received: from mail2004.gr by yahoo.gr ; 3 Jen 04 12:12:12 GMT
Received: from mail.gr by mail2004.gr ; 3 Jen 04 12:10:24 GMT
From: alice@mail.gr
To: bob@yahoo.gr
Subject: Picture.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
```

base64 encoded data .....
.....
.....base64 encoded data

3.3. Πρωτόκολλα πρόσβασης στο Ηλεκτρονικό Ταχυδρομείο

Τα πρώτα χρόνια χρήσης του Ηλεκτρονικού Ταχυδρομείου ο χρήστης για να πάρει τα μηνύματά του έπρεπε να συνδεθεί, μέσω telnet, στον εξυπηρετητή Ηλεκτρονικού Ταχυδρομείου. Με τον καιρό αναπτύχθηκαν **αντιπρόσωποι χρήστη (user agents)** οι οποίοι εκτελούνται στο μηχάνημα του χρήστη. Οι αντιπρόσωποι αυτοί παρείχαν ένα

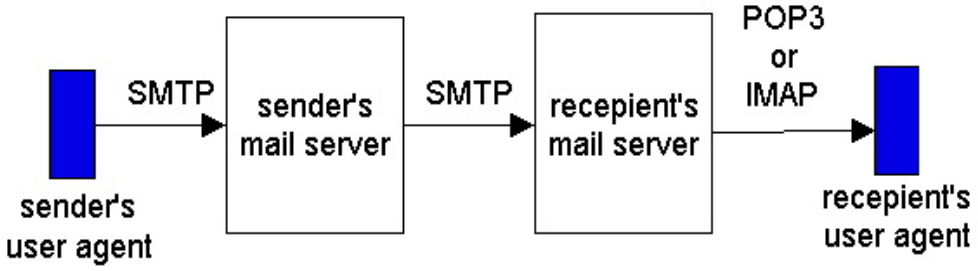
πλήθος από δυνατότητες στους χρήστες, όπως την προβολή αρχείων πολυμέσων και συνημμένων αρχείων. Οι πιο γνωστοί από τους αντιπροσώπους χρήστη είναι το Microsoft Outlook Express, το Eudora και το Netscape Messenger.

Ένας χρήστης που εκτελεί έναν αντιπρόσωπο χρήστη στο τοπικό του μηχάνημα είναι σαν να έχει έναν τοπικό εξυπηρετητή ταχυδρομείου. Στην παραπάνω προσέγγιση παρουσιάζεται ένα σημαντικό πρόβλημα: επειδή οι εξυπηρετητές ταχυδρομείου διαχειρίζονται τα mailboxes και τρέχουν τις πλευρές του εξυπηρετητή και του χρήστη του πρωτοκόλλου SMTP, οι αντιπρόσωποι χρήστη πρέπει να είναι συνεχώς συνδεδεμένοι με το Διαδίκτυο, για να μπορούν να λαμβάνουν τα μηνύματα που μπορεί να φθάσουν οποιαδήποτε χρονική στιγμή. Η συνεχής σύνδεση με το Διαδίκτυο δεν ανταποκρίνεται στην πλειοψηφία των χρηστών του Ηλεκτρονικού Ταχυδρομείου. Για την αποφυγή του παραπάνω προβλήματος, κάθε χρήστης διατηρεί ένα mailbox σε κάποιον μοιραζόμενο (shared) εξυπηρετητή ταχυδρομείου, δηλαδή σε έναν εξυπηρετητή ταχυδρομείου που πάντα είναι ενεργός και συνδεδεμένος στο Διαδίκτυο, ώστε να λαμβάνει τα μηνύματα των διαφόρων χρηστών του. Οι εξυπηρετητές ταχυδρομείου συνήθως ανήκουν στους ISP, δηλαδή στους παροχείς υπηρεσιών Διαδικτύου.

Για την επικοινωνία των πρακτόρων χρήστη και των μοιραζόμενων εξυπηρετητών ταχυδρομείου χρειάζεται ένα νέο πρωτόκολλο. Ας δούμε ένα παράδειγμα μεταφοράς ενός μηνύματος από τον αντιπρόσωπο χρήστη του αποστολέα (Alice) στον αντιπρόσωπο χρήστη του παραλήπτη (Bob). Υποθέτουμε ότι ο αντιπρόσωπος χρήστη του αποστολέα μπορεί να επικοινωνήσει απευθείας, μέσω του πρωτοκόλλου SMTP, με τον εξυπηρετητή ταχυδρομείου του Bob. Για τον σκοπό αυτό ο αντιπρόσωπος χρήστη δημιουργεί μία TCP σύνδεση, πραγματοποιεί την χειραψία του SMTP και στη συνέχεια μέσω της εντολής DATA αποστέλλει το μήνυμα. Μετά την αποστολή του μηνύματος η TCP σύνδεση τερματίζεται. Η παραπάνω προσέγγιση έχει ένα σημαντικό μειονέκτημα: στην περίπτωση που ο εξυπηρετητής ταχυδρομείου του Bob έχει πρόβλημα ο αποστολέας δεν μπορεί να γνωρίζει την πηγή του προβλήματος. Η προσέγγιση που χρησιμοποιείται συνήθως στην πράξη είναι η δημιουργία μίας TCP σύνδεσης με τον μοιραζόμενο εξυπηρετητή ταχυδρομείου του ίδιου του αποστολέα και η μεταφορά του μηνύματος, με χρήση του SMTP, πάνω από την TCP σύνδεση αυτή. Στη συνέχεια οι εξυπηρετητές ταχυδρομείου του αποστολέα και του παραλήπτη επικοινωνούν μεταξύ τους κάνοντας χρήση του πρωτοκόλλου SMTP. Σε περίπτωση αποτυχίας το μήνυμα μπορεί να αποθηκευτεί προσωρινά στον εξυπηρετητή του αποστολέα, ώστε να επαναληφθεί η αποστολή του αργότερα. Όταν το μήνυμα φθάσει στον εξυπηρετητή ταχυδρομείου του παραλήπτη αποθηκεύεται στο mailbox που αντιστοιχεί στον συγκεκριμένο παραλήπτη.

Για να μπορέσει ο παραλήπτης να πάρει το μήνυμα χρειάζεται η επικοινωνία του αντιπροσώπου χρήστη του με τον εξυπηρετητή ταχυδρομείου του. Για την επικοινωνία αυτή χρειάζεται ένα **πρωτόκολλο πρόσβασης ταχυδρομείου (mail access protocol)**. Τα πιο δημοφιλής πρωτόκολλα πρόσβασης ταχυδρομείου είναι το **POP3 (Post Office Protocol – Version 3)** και το **IMAP (Internet Mail Access Protocol)**.

Στο παρακάτω σχήμα φαίνεται η τυπική χρήση των διαφόρων πρωτοκόλλων για την μεταφορά του μηνύματος από τον αποστολέα στον παραλήπτη.



3.3.1 POP3

Το πρωτόκολλο **POP3**, που ορίζεται στο *RFC 1939*, είναι ένα απλό πρωτόκολλο μεταφοράς, με περιορισμένες δυνατότητες. Το πρωτόκολλο αυτό ξεκινά όταν ο αντιπρόσωπος χρήστη δημιουργεί μία **TCP** σύνδεση στη θύρα **110** του μοιραζόμενου εξυπηρετητή ταχυδρομείου και έχει τρεις φάσεις: *πιστοποίηση (authentication)*, *συναλλαγή (transaction)* και *ενημέρωση (update)*. Κατά την πρώτη φάση ο αντιπρόσωπος χρήστη στέλνει ένα όνομα χρήστη (username) και έναν κωδικό (password) στον εξυπηρετητή, τα οποία χρησιμοποιούνται από τον εξυπηρετητή για την πιστοποίηση του χρήστη. Κατά τη δεύτερη φάση, ο αντιπρόσωπος χρήστη μπορεί να σημειώσει τα μηνύματα προς διαγραφή ή να αφαιρέσει τα σημάδια διαγραφής από κάποια μηνύματα και να πάρει στατιστικά στοιχεία για τη χρήση του Ηλεκτρονικού Ταχυδρομείου. Κατά την τρίτη φάση, που εκτελείτε όταν ο χρήστης δώσει την εντολή **quit**, για τον τερματισμό του πρωτοκόλλου POP3, διαγράφονται τα μηνύματα που είναι σημειωμένα για διαγραφή.
Στη φάση της POP3 συναλλαγής, ο χρήστης δίνει εντολές στις οποίες ο εξυπηρετητής στέλνει αποκρίσεις (responses). Υπάρχουν δύο ειδών αποκρίσεων: **+OK** (που μερικές φορές ακολουθείται από δεδομένα από τον εξυπηρετητή προς τον πελάτη) για να δηλώσει ο εξυπηρετητής ότι η προηγουμένη εντολή ήταν έγκυρη, και **–ERR** για να δηλωθεί ότι η προηγούμενη εντολή είχε κάποιο λάθος.
Η φάση της πιστοποίησης έχει δύο κύριες εντολές: **user <username>** και **pass <password>**. Αν το password δεν αντιστοιχεί στο υπαρκτό username τότε ο εξυπηρετητής αποκρίνεται με ένα μήνυμα –ERR, διαφορετικά αποκρίνεται με ένα μήνυμα OK. Ακολουθεί ένα παράδειγμα επιτυχούς πιστοποίησης.

```
telnet mailServer 110
+OK POP3 server ready
user alice
+OK
pass hungry
+OK user successfully logged on
```

Ας δούμε με περισσότερη λεπτομέρεια τη φάση της συναλλαγής. Ο αντιπρόσωπος χρήστη συνήθως λειτουργεί σε έναν από τους παρακάτω τρόπους: **κατέβασμα και διαγραφή (download and delete)** ή **κατέβασμα και διατήρηση (download and keep)**. Κατά τον πρώτο τρόπο λειτουργίας ο αντιπρόσωπος χρήστη εκτελεί με τη σειρά τις εντολές **list**, **retr** και **dele**. Στο παρακάτω παράδειγμα φαίνεται ο παραπάνω τρόπος

λειτουργίας (με C συμβολίζεται ο πελάτης - αντιπρόσωπος χρήστη και με S ο εξυπηρετητής ταχυδρομείου), όταν ο χρήστης έχει δύο μηνύματα στο mailbox του.

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah ...
S: .................
S: ..........blah
S: .
C: dele 1
C: retr 2
S: blah blah ...
S: .................
S: ..........blah
S: .
C: dele 2
C:quit
S:+OK POP3 server signing off
```

Στην αρχή ο αντιπρόσωπος χρήστη ζητά να ανακτήσει το μέγεθος όλων των αποθηκευμένων μηνυμάτων. Στη συνέχεια ο αντιπρόσωπος χρήστη ανακτά και διαγράφει κάθε μήνυμα από τον εξυπηρετητή ταχυδρομείου. Κατά τον παραπάνω τρόπο λειτουργίας εκτός από τα μηνύματα της φάσης πιστοποίησης ανταλλάσσονται μόνο τέσσερις εντολές: list, retr, dele και quit. Η εντολή dele απλώς μαρκάρει τα μηνύματα που είναι προς διαγραφή. Τα μηνύματα αυτά διαγράφονται από το mailbox κατά την φάση της ενημέρωσης – εντολή quit.
Στην περίπτωση που ένας χρήστης θέλει να διαβάσει το ηλεκτρονικό του ταχυδρομείο από διαφορετικά μηχανήματα, τότε δεν μπορεί να χρησιμοποιήσει τον παραπάνω τρόπο λειτουργίας, αφού μετά την πρώτη ανάκτηση τα μηνύματα διαγράφονται. Στην περίπτωση αυτή ο χρήστης πρέπει να χρησιμοποιήσει τον δεύτερο τρόπο λειτουργίας που αναφέρθηκε παραπάνω, το κατέβασμα και τη διατήρηση. Σε αυτόν τον τρόπο λειτουργίας δεν χρησιμοποιείται η εντολή dele με αποτέλεσμα τα μηνύματα να διατηρούνται στο mailbox του χρήστη και μετά την ανάγνωσή τους.
Κατά την POP3 σύνοδο, ο εξυπηρετητής διατηρεί την κατάσταση του χρήστη. Συγκεκριμένα κρατά ποια μηνύματα έχουν σημειωθεί ότι πρέπει να διαγραφούν. Ο εξυπηρετητής όμως δεν είναι αναγκασμένος να κρατά την κατάσταση μεταξύ διαφόρων POP3 συνόδων – όλα τα μηνύματα είναι σημειωμένα στην αρχή κάθε συνόδου ως μηνύματα που δεν είναι προς διαγραφή. Αυτός ο τρόπος λειτουργίας απλοποιεί πολύ την υλοποίηση του POP3 εξυπηρετητή.

3.3.2 IMAP

Όταν ένας χρήστης κάνει χρήση του POP3 για την λήψη των ηλεκτρονικών μηνυμάτων του, δεν έχει καμία δυνατότητα για τη διαμόρφωση του τρόπου αποθήκευσης των μηνυμάτων αυτών στο mailbox. Σε πολλές όμως περιπτώσεις ο χρήστης θέλει τα μηνύματα που είναι αποθηκευμένα στο mailbox να έχουν κάποια διαμόρφωση αποθήκευσης, π.χ. να είναι οργανωμένα σε καταλόγους. Αυτό όμως δεν είναι δυνατό να γίνει όταν χρησιμοποιείται το πρωτόκολλο POP3.
Για τη λύση του παραπάνω, αλλά και άλλων προβλημάτων, ανακαλύφθηκε το πρωτόκολλο **IMAP (Internet Mail Access Protocol)**, το οποίο ορίζεται στο *RFC 1730*. Το πρωτόκολλο αυτό παρέχει πολλές επιπλέον δυνατότητες, σε σχέση με το POP3, αλλά ταυτόχρονα είναι και αρκετά πιο πολύπλοκο. Το IMAP παρέχει τη δυνατότητα οργάνωσης των μηνυμάτων στο mailbox σαν να ήταν αρχεία του τοπικού συστήματος: μπορεί να υπάρχουν στο mailbox περισσότεροι από ένας κατάλογοι και ο χρήστης μπορεί να μετακινεί μηνύματα μεταξύ τους. Επίσης, παρέχεται στον χρήστη η δυνατότητα για την αναζήτηση μηνυμάτων με βάση κριτήρια τα οποία ο ίδιος παρέχει. Ο IMAP εξυπηρετητής πρέπει να συντηρεί μία ιεραρχία καταλόγων για κάθε χρήστη, η οποία διατηρείτε ανάμεσα στις διάφορες IMAP συνόδους, με αποτέλεσμα να αυξάνεται σε σημαντικό βαθμό η πολυπλοκότητα του πρωτοκόλλου, σε σχέση με το POP3 το οποίο δεν διατηρεί κανένα στοιχείο κατάστασης μεταξύ των POP3 συνόδων.
Ένα άλλο σημαντικό χαρακτηριστικό του IMAP είναι η δυνατότητα λήψης από τον χρήστη ενός μόνο τμήματος από ένα MIME μήνυμα με πολλά τμήματα. Το χαρακτηριστικό αυτό είναι ιδιαίτερα χρήσιμο σε συνδέσεις με μικρό εύρος ζώνης (bandwidth) μεταξύ του πράκτορα χρήστη και του εξυπηρετητή ταχυδρομείου. Σε αυτή την περίπτωση ο χρήστης μπορεί να μη κατεβάσει μεγάλα τμήματα μηνυμάτων (π.χ. αρχεία βίντεο) ώστε να ελαττωθεί ο χρόνος απόκρισης.
Μία IMAP σύνοδος αποτελείται από τη φάση εγκατάστασης της σύνδεσης μεταξύ του πελάτη και του εξυπηρετητή ταχυδρομείου, από τη φάση χειραψίας και από τη φάση των συναλλαγών μεταξύ του πελάτη και του εξυπηρετητή. Κατά την τρίτη φάση ο πελάτης στέλνει μία εντολή στον IMAP εξυπηρετητή, ο οποίος αποκρίνεται με ένα μήνυμα ολοκλήρωσης (completion result response). Ο IMAP εξυπηρετητής βρίσκεται πάντα σε μία από τις τέσσερις ακόλουθες καταστάσεις. Στην **κατάσταση μη πιστοποίησης (non-authenticated state)**, στην οποία βρίσκεται κατά την έναρξη της συνόδου, ζητείται από τον χρήστη η παροχή ενός ονόματος χρήστη και ενός κωδικού για την πιστοποίησή του. Στην **κατάσταση πιστοποίησης (authenticated state)** ζητείτε από τον χρήστη η επιλογή ενός καταλόγου πριν από τις εντολές που έχουν σχέση με μηνύματα. Στην **κατάσταση επιλογής (selected state)** ο χρήστης μπορεί να δώσει εντολές που έχουν σχέση με μηνύματα. Τέλος, στην **κατάσταση εξόδου (logout state)** βρίσκεται ο εξυπηρετητής κατά τον τερματισμό της συνόδου. Ανάλογα με την κατάσταση που βρίσκεται ο IMAP εξυπηρετητής γίνονται αποδεκτές και αντίστοιχες εντολές του πρωτοκόλλου.

4. Το πρωτόκολλο HTTP

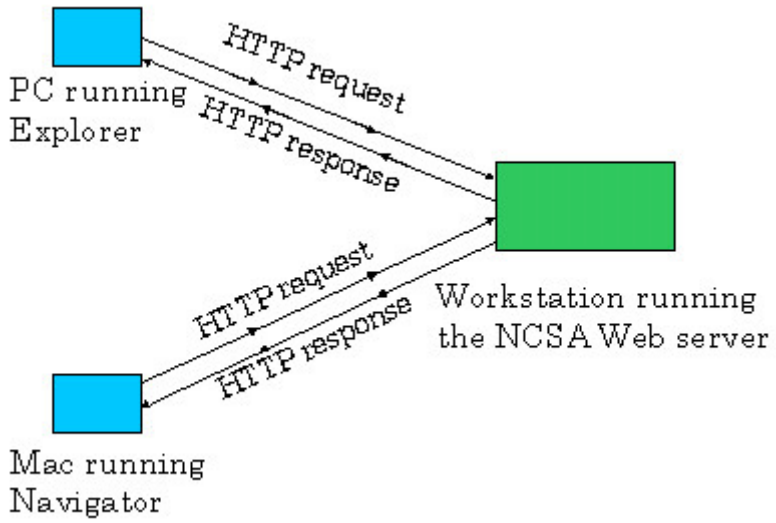
Στις αρχές τις δεκαετίας του 1980 το Διαδίκτυο χρησιμοποιούνταν κυρίως στην ακαδημαϊκή κοινότητα. Οι υπηρεσίες που παρείχε ήταν η πρόσβαση σε έναν απομακρυσμένο υπολογιστή, το ηλεκτρονικό ταχυδρομείο, η μεταφορά αρχείων και η αποστολή και λήψη νέων. Το 1989 στα εργαστήρια του Ευρωπαϊκού Κέντρου Πυρηνικών Ερευνών CERN προτάθηκε η ιδέα του **Παγκόσμιου Ιστού (World Wide Web – WWW)** ή απλά **Ιστού (Web)** από τον φυσικό Tim Berners – Lee ως μία αρχιτεκτονική για την επικοινωνία μεταξύ των διαφόρων ερευνητικών ομάδων του CERN. Η βασική ιδέα της αρχιτεκτονικής αυτής ήταν η διασύνδεση των εγγράφων που ήταν διασκορπισμένα στο Διαδίκτυο. Γρήγορα έγινε ιδιαίτερα δημοφιλές και γύρω του αναπτύχθηκε μία ολόκληρη βιομηχανία. Η δημοτικότητά του οφείλεται τόσο στο γραφικό περιβάλλον χρήσης του όσο και στην μεγάλη πληθώρα πληροφοριών που διαθέτει με αποτέλεσμα ο κάθε χρήστης να μπορεί να βρει αυτό που τον ενδιαφέρει με ελάχιστο κόστος.

Το **Πρωτόκολλο Μεταφοράς Υπερκειμένου HTTP (HyperText Transfer Protocol)** είναι η καρδιά του Ιστού. Το HTTP ανήκει στο στρώμα εφαρμογών του Διαδικτύου και υλοποιείτε ως δύο προγράμματα: ένα **πρόγραμμα πελάτη (client program)** και ένα **πρόγραμμα εξυπηρετητή (server program)**. Τα δύο αυτά προγράμματα εκτελούνται σε διαφορετικά μηχανήματα επικοινωνώντας μεταξύ τους ανταλλάσσοντας HTTP μηνύματα. Συγκεκριμένα το HTTP ορίζει τη δομή των μηνυμάτων αυτών καθώς και τον τρόπο ανταλλαγής τους ανάμεσα στον πελάτη και στον εξυπηρετητή.

Πριν περιγράψουμε αναλυτικά το πρωτόκολλο HTTP πρέπει να αναφερθούμε σε κάποια βασική ορολογία του Ιστού. Μία Ι**στοσελίδα (Web page)** αποτελείται από αντικείμενα. Με τον όρο **αντικείμενο (object)** εννοούμε ένα απλό αρχείο, όπως ένα αρχείο HTML, ένα αρχείο εικόνας ή ένα αρχείο βίντεο, το οποίο μπορεί να προσπελαστεί μέσω ενός URL. Οι περισσότερες Ιστοσελίδες αποτελούνται από ένα **βασικό αρχείο HTML** και διάφορα σχετικά αντικείμενα. Αν υποθέσουμε ότι έχουμε μία Ιστοσελίδα που περιέχει ένα αρχείο HTML και 3 αρχεία εικόνων τότε λέμε ότι η Ιστοσελίδα έχει 4 αντικείμενα. Το βασικό αρχείο HTML αναφέρεται στα άλλα αντικείμενα της σελίδας μέσω των URL των αντικειμένων. Κάθε URL αποτελείται από δύο τμήματα: το όνομα του υπολογιστή – host στον οποίον είναι αποθηκευμένο το αρχείο και το όνομα του μονοπατιού (path) του αντικειμένου. Π.χ. το URL www.ntua.gr/index.htm έχει ως όνομα host το www.ntua.gr και ως όνομα μονοπατιού το index.htm.

Ο **browser** είναι ο αντιπρόσωπος του Ιστού: απεικονίζει στον χρήστη τη ζητούμενη Ιστοσελίδα και παρέχει πληθώρα χαρακτηριστικών πλοήγησης και παραμετροποίησης. Επίσης, στους browser υλοποιείτε και η πλευρά του πελάτη του πρωτοκόλλου HTTP. Ένας **εξυπηρετητής Ιστού (Web server)** αποθηκεύει τα αντικείμενα της Ιστοσελίδας, το καθένα από τα οποία έχει ως διεύθυνση ένα URL. Στους εξυπηρετητές Ιστού υλοποιείτε και η πλευρά του εξυπηρετητή του πρωτοκόλλου HTTP.

Το HTTP ορίζει τον τρόπο με τον οποίο οι πελάτες του Ιστού (π.χ. οι browsers) ζητούν (request) Ιστοσελίδες από τους εξυπηρετητές του Ιστού (π.χ. τους Web servers) και πως οι εξυπηρετητές μεταφέρουν τις Ιστοσελίδες στους πελάτες. Η βασική ιδέα της του πρωτοκόλλου αυτού φαίνεται στο παρακάτω σχήμα.



Όταν ο χρήστης ζητά μία Ιστοσελίδα, ο browser στέλνει ένα μήνυμα **HTTP αίτησης (HTTP request)**, για τα διάφορα αντικείμενα της σελίδας, στον εξυπηρετητή. Ο εξυπηρετητής όταν λάβει το μήνυμα αυτό ανταποκρίνεται με μηνύματα **HTTP απόκρισης (HTTP response)** στα οποία περιέχονται τα αιτούμενα αντικείμενα. Μέχρι το 1997 όλοι οι browsers και όλοι οι εξυπηρετητές Ιστού υλοποιούσαν την έκδοση 1.0 του HTTP, που για συντομία γράφεται **HTTP/1.0**. Από το 1998 όμως άρχισαν να υποστηρίζουν και το **HTTP/1.1**, το οποίο είναι συμβατό με το HTTP/1.0. Δηλαδή, ένας browser που υποστηρίζει το HTTP/1.0 μπορεί να επικοινωνήσει με ένας εξυπηρετητή Ιστού που υποστηρίζει το HTTP/1.1 και, αντιστρόφως, ένας browser που υποστηρίζει το HTTP/1.1 μπορεί να επικοινωνήσει με έναν εξυπηρετητή Ιστού που υποστηρίζει το HTTP/1.0.

Τόσο το HTTP/1.0 όσο και το HTTP/1.1 χρησιμοποιούν το **TCP** ως πρωτόκολλο μεταφοράς. Αφού ο πελάτης εγκαταστήσει μία σύνδεση TCP με τον εξυπηρετητή αρχίζει την αποστολή μηνυμάτων – αιτήσεων προς αυτόν και τη λήψη μηνυμάτων – αποκρίσεων από αυτόν. Λόγω της χρήσης του TCP το HTTP δεν χρειάζεται να ασχοληθεί καθόλου με τη μεταφορά των δεδομένων. Το μόνο που πρέπει να κάνει είναι να στείλει τις αιτήσεις μέσω της TCP σύνδεσης και να περιμένει τις αποκρίσεις. Το TCP εγγυάται την αξιόπιστη μεταφορά των δεδομένων καθώς και τον έλεγχο της συμφόρησης.

Οι εξυπηρετητές του HTTP δεν κρατάνε καθόλου στοιχεία για την κατάσταση του πελάτη. Επομένως, αν ένας πελάτης στείλει μία αίτηση για ένα αρχείο δύο φορές, ο εξυπηρετητής θα του στείλει το αρχείο αυτό δύο φορές. Τα πρωτόκολλα που δεν κρατάνε καθόλου πληροφορία για την κατάσταση του πελάτη ονομάζονται stateless.

4.1. Μόνιμες και μη μόνιμες συνδέσεις

Το HTTP μπορεί να χρησιμοποιήσει τόσο **μόνιμες (persistent)** όσο και **μη μόνιμες (non-persistent)** συνδέσεις. Το HTTP/1.0 χρησιμοποιεί μη μόνιμες συνδέσεις ενώ το HTTP/1.1 χρησιμοποιεί κύρια μόνιμες συνδέσεις.
Θα εξετάσουμε πρώτα την περίπτωση των μη μόνιμων συνδέσεων, με ένα παράδειγμα. Έστω ότι θέλουμε να εμφανίσουμε στην οθόνη μία ιστοσελίδα που αποτελείται από το βασικό HTML αρχείο και άλλα τέσσερα αρχεία εικόνων. Έστω επίσης ότι το URL του βασικού HTML αρχείου είναι το www.ntua.gr/index.htm. Τότε πρέπει να ακολουθηθούν τα επόμενα βήματα για την λήψη της Ιστοσελίδας (αγνοούνται τυχόν DNS μηνύματα).

1. Ο HTTP πελάτης εγκαθιστά μία σύνδεση **TCP** στην **θύρα 80** του εξυπηρετητή HTTP. Η θύρα 80 είναι η προκαθορισμένη θύρα στην οποία ο εξυπηρετητής HTTP περιμένει για αφίξεις αιτήσεων HTTP από διάφορους πελάτες HTTP.
2. Στη συνέχεια ο πελάτης HTTP στέλνει ένα μήνυμα HTTP αίτησης ζητώντας το αρχείο index.htm.
3. Ο εξυπηρετητής στέλνει το αρχείο index.htm στον πελάτη μέσω ενός HTTP μηνύματος απόκρισης.
4. Ο HTTP εξυπηρετητής κλείνει την TCP σύνδεση (στην πραγματικότητα η TCP σύνδεση κλείνει όταν επιβεβαιωθεί η λήψη του αρχείου από τον πελάτη).
5. Ο browser λαμβάνει το αρχείο index.htm, κλείνει τη σύνδεση, και απεικονίζει τα περιεχόμενά του στην οθόνη του χρήστη. Επίσης ο browser αναγνωρίζει και τα URL των τεσσάρων αναφερόμενων εικόνων της Ιστοσελίδας.
6. Τα τέσσερα πρώτα βήματα επαναλαμβάνονται για καθένα από τα αναφερόμενα αντικείμενα (στην περίπτωσή μας εικόνες).

Από το παραπάνω παράδειγμα γίνεται φανερό γιατί ο τύπος αυτός της σύνδεσης ονομάζεται μη μόνιμος – μετά από την μεταφορά ενός αντικειμένου της Ιστοσελίδας η TCP σύνδεση τερματίζεται και για κάθε άλλο αντικείμενο της Ιστοσελίδας πρέπει να εγκατασταθεί μία καινούργια TCP σύνδεση. Στο παραπάνω παράδειγμα για την μεταφορά της Ιστοσελίδας χρειάζεται η εγκατάσταση συνολικά 5 TCP συνδέσεων.
Παρά το γεγονός ότι πρέπει να εγκατασταθούν τέσσερις διαφορετικές TCP συνδέσεις για τη μεταφορά των τεσσάρων εικόνων, οι σύγχρονοι browser υποστηρίζουν την **παράλληλη μεταφορά (parallel transfer)** των διαφόρων αντικειμένων. Σε αυτή την περίπτωση οι εικόνες μπορεί να μεταφερθούν παράλληλα με αποτέλεσμα τη μείωση του συνολικού χρόνου που χρειάζεται για τη ολόκληρης της σελίδας στον browser. Όταν εγκαθίστανται παράλληλες TCP συνδέσεις το συνολικό *εύρος ζώνης (bandwidth)* μοιράζεται ανάμεσα στις παράλληλες συνδέσεις.
Πριν να περιγράψουμε τις μόνιμες συνδέσεις θα κάνουμε ένα χοντρικό υπολογισμό του χρόνου που χρειάζεται για τη μεταφορά ολόκληρης της Ιστοσελίδας. Για τον σκοπό αυτό ορίζουμε ως **RTT (Round Trip Time)** το χρόνο που χρειάζεται ένα πακέτο να μεταφερθεί από τον εξυπηρετητή Ιστού στον browser και μετά πάλι πίσω. Ο χρόνος αυτός περιλαμβάνει την *καθυστέρηση διάδοσης (propagation delay)*, την *καθυστέρηση* του πακέτου *στις ουρές* των δρομολογητές *(queuing delay)* καθώς και την *καθυστέρηση επεξεργασίας του πακέτου (packet processing delay)*. Ας δούμε τι συμβαίνει όταν ο χρήστης ζητά μία Ιστοσελίδα από τον browser. Αρχικά ο browser εγκαθιστά μία TCP σύνδεση με τον εξυπηρετητή Ιστού. Για το σκοπό αυτό λαμβάνει χώρα μία *τριμερής χειραψία (three-way handshake)*: πρώτα ο πελάτης στέλνει ένα μικρό TCP segment στον εξυπηρετητή, στη συνέχεια ο εξυπηρετητής επιβεβαιώνει τη λήψη του παραπάνω

μηνύματος με ένα νέο TCP segment που στέλνει πίσω στον πελάτη και τέλος ο πελάτης επιβεβαιώνει τη λήψη του τελευταίου αυτού μηνύματος. Η τελευταία επιβεβαίωση μπορεί να είναι εμβόλιμη (piggy backing) στην HTTP αίτηση που κάνει ο πελάτης. Όταν η αίτηση αυτή φτάσει στον εξυπηρετητή αυτός αποκρίνεται με μία απόκριση HTTP που περιλαμβάνει το ζητούμενο αρχείο. Με μια πρώτη προσέγγιση ο συνολικός χρόνος είναι 2RTT, συν το χρόνο μετάδοσης του αρχείου από τον εξυπηρετητή, για κάθε αρχείο.
Από τα παραπάνω γίνεται φανερό ότι οι μη μόνιμες συνδέσεις παρουσιάζουν κάποια σημαντικά προβλήματα: πρώτον, για κάθε αντικείμενο της Ιστοσελίδας πρέπει να εγκατασταθεί μία ξεχωριστή TCP σύνδεση με αποτέλεσμα να χρειάζεται κάποιος χώρος αποθήκευσης για την κατάστασή της τόσο στην πλευρά του πελάτη όσο και στην πλευρά του εξυπηρετητή. Το πρόβλημα αυτό παρουσιάζεται κυρίως στους εξυπηρετητές που συνήθως πρέπει να είναι ικανοί να εξυπηρετήσουν πολλούς πελάτες ταυτόχρονα. Δεύτερον, το κάθε αντικείμενο της Ιστοσελίδας δέχεται καθυστέρηση 2RTT και, τρίτον, στο TCP έχουμε το χαρακτηριστικό της *αργής εκκίνησης (slow start)* που προσδίδει ακόμη μεγαλύτερη καθυστέρηση στη μεταφορά των αντικειμένων. Αν και τα μειονεκτήματα αυτά μπορεί να περιοριστούν αν γίνει χρήση παράλληλων συνδέσεων, το HTTP υποστηρίζει μία πιο αποδοτική τεχνική, τις μόνιμες συνδέσεις.
Στις **μόνιμες συνδέσεις** ο εξυπηρετητής TCP δεν τερματίζει τη σύνδεση αμέσως μετά την αποστολή της απόκρισης. Αντίθετα, την τερματίζει μετά από κάποιο συγκεκριμένο χρονικό διάστημα κατά το οποίο η σύνδεση είναι ανενεργή (δηλαδή δεν γίνεται μεταφορά δεδομένων). Μέσα στο χρονικό διάστημα αυτό ο browser μπορεί να ζητήσει την αποστολή και άλλων αντικειμένων από την ίδια Ιστοσελίδα αλλά και την αποστολή και άλλων Ιστοσελίδων από τον ίδιο εξυπηρετητή Ιστού.
Υπάρχουν δύο τύποι μόνιμων συνδέσεων: **με pipelining** και **χωρίς pipelining**. Στην πρώτη περίπτωση ο πελάτης στέλνει μία καινούργια αίτηση μόνο μετά τη λήψη της προηγούμενης απάντησης, οπότε το κάθε αντικείμενο της Ιστοσελίδας υπόκειται σε έναν επιπλέον χρόνο RTT. Ο χρόνος απόκρισης μπορεί να βελτιωθεί ακόμη περισσότερο αν χρησιμοποιηθούν μόνιμες συνδέσεις με pipelining. Επίσης η χρήση pipelining καλύπτει και ένα άλλο μειονέκτημα που έχει η απουσία pipelining. Το μειονέκτημα αυτό είναι ο χρόνος αναμονής της λήψης απόκρισης από τους πελάτες και τη λήψη αιτήσεων από τους εξυπηρετητές, κατά τον οποίο γίνεται σπατάλη πόρων που θα μπορούσαν να χρησιμοποιηθούν σε άλλους σκοπούς.
Η προκαθορισμένη επιλογή του HTTP/1.1 είναι η χρήση μόνιμων συνδέσεων με pipelining. Όταν ο HTTP πελάτης γνωρίζει το URL ενός αναφερόμενου αντικειμένου τότε στέλνει αμέσως αίτηση στον εξυπηρετητή Ιστού, για τη λήψη του. Στην περίπτωση αυτή ο χρόνος λήψης όλων των αναφερόμενων αντικειμένων μπορεί να μειωθεί ακόμα και στο επίπεδο του ενός RTT. Με τη μέθοδο αυτή επίσης εξαλείφεται και το μειονέκτημα της αργής εκκίνησης, αφού οι αιτήσεις (μετά την αρχική) στέλνονται με μεγαλύτερο ρυθμό από τον αρχικό ρυθμό της αργής εκκίνησης.

4.2. Μορφή των HTTP μηνυμάτων

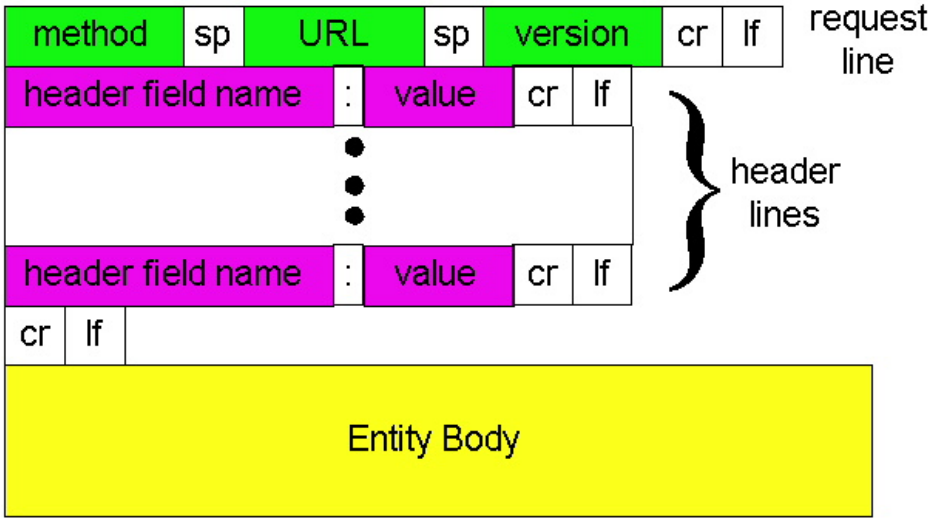
Το HTTP ορίζει μόνο δύο τύπους μηνυμάτων: τις HTTP αιτήσεις (requests) και τις HTTP αποκρίσεις (responses).

Ακολουθεί ένα παράδειγμα ενός μηνύματος HTTP αίτησης.

```
GET /lessons/index.htm HTTP/1.1
Connection: close
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language:gr

(extra carriage return, line feed)
```

Όπως φαίνεται και από το παραπάνω παράδειγμα οι HTTP αιτήσεις γράφονται με χαρακτήρες ASCII και μπορούν να διαβαστούν από τους ανθρώπους. Οι περισσότερες HTTP αιτήσεις αποτελούνται από 5 γραμμές κειμένου ακολουθούμενες από μία κενή γραμμή. Οι HTTP αιτήσεις περιέχουν τουλάχιστον μία γραμμή κειμένου, ενώ υπάρχουν και περιπτώσεις που αποτελούνται και από περισσότερες από 5 γραμμές κειμένου. Η πρώτη γραμμή κειμένου ονομάζεται **γραμμή αίτησης (request line)**, ενώ οι επόμενες **γραμμές επικεφαλίδας (header lines)**.
Η γραμμή αίτησης περιέχει τρία πεδία: το **πεδίο μεθόδου**, το **πεδίο URL** και το **πεδίο HTTP έκδοσης**. Το πεδίο μεθόδου μπορεί να έχει μία από τις ακόλουθες τιμές: **GET**, **POST** και **HEAD**. Η πιο συνηθισμένη μέθοδος στις HTTP αιτήσεις είναι η GET, με την οποία ζητείται από τον εξυπηρετητή η αποστολή του αρχείου που εμφανίζεται στο πεδίο URL. Στο URL δεν είναι απαραίτητο να υπάρχει και το όνομα του host, αφού ήδη έχει εγκατασταθεί μία σύνδεση με τον host αυτό (δηλαδή τον HTTP εξυπηρετητή). Τέλος, στο πεδίο έκδοσης HTTP περιγράφεται η έκδοση του HTTP που χρησιμοποιεί ο αιτών host. Στο παραπάνω παράδειγμα χρησιμοποιείται το HTTP/1.1.
Στη συνέχεια θα εξετάσουμε τις γραμμές επικεφαλίδας του παραπάνω παραδείγματος. Η γραμμή **Connection:close** λέει στον εξυπηρετητή να τερματιστεί η σύνδεση μετά την αποστολή του αρχείου, δηλαδή να μη γίνει χρήση μόνιμης σύνδεσης. Η επόμενη γραμμή που αρχίζει με **User-agent:** δηλώνει τον τύπο του browser του χρήστη. Στο παραπάνω παράδειγμα δηλώνεται ότι ο χρήστης χρησιμοποιεί τον browser Mozilla/4.0. Η γραμμή αυτή χρησιμοποιείται από τον εξυπηρετητή για την αποστολή διαφορετικών αντικειμένων ανάλογα με τον browser του χρήστη (τα οποία όμως έχουν το ίδιο URL). Η **Accept:** γραμμή δηλώνει τον τύπο των αντικειμένων που υποστηρίζει ο browser. Στο παραπάνω παράδειγμα δηλώνεται ότι ο browser υποστηρίζει αρχεία HTML και αρχεία εικόνων τύπου GIF και JPEG. Τέλος, η **Accept-language:** γραμμή δηλώνει την γλώσσα έκδοσης του αντικειμένου που επιθυμεί να λάβει ο browser. Αν ο εξυπηρετητής δεν έχει έκδοση του αντικειμένου για την γλώσσα αυτή τότε στέλνει την προκαθορισμένη έκδοση του αντικειμένου. Στο παραπάνω παράδειγμα αν ο εξυπηρετητής διαθέτει μία ελληνική έκδοση του αρχείου lessons/index.htm τότε πρέπει να τη στείλει, αλλιώς να στείλει την προκαθορισμένη έκδοση του αρχείου.
Η δομή του μηνύματος HTTP αίτησης φαίνεται στο παρακάτω σχήμα.

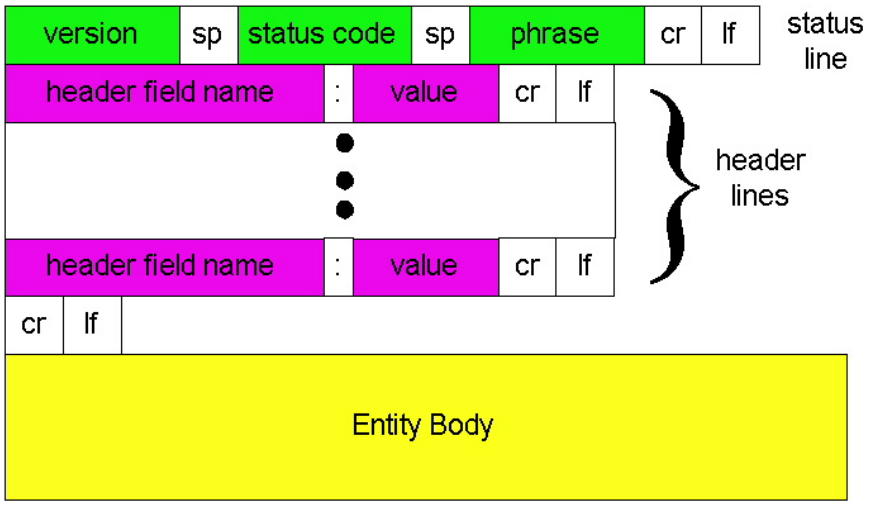


Από το παραπάνω σχήμα βλέπουμε ότι η γενική μορφή του μηνύματος ακολουθεί τη δομή του παραπάνω παραδείγματος. Το πεδίο **Entity Body** δεν χρησιμοποιείται για τη μέθοδο GET, χρησιμοποιείται όμως όταν γίνεται χρήση της μεθόδου POST. Ένα παράδειγμα χρήσης της μεθόδου POST είναι η αποστολή των δεδομένων που συμπλήρωσε ο χρήστης σε διάφορες φόρμες μίας Ιστοσελίδας. Στην περίπτωση αυτή τα δεδομένα αποστέλλονται στο πεδίο Entity Body και ανάλογα με τα δεδομένα αυτά ο χρήστης λαμβάνει μία κατάλληλη έκδοση της Ιστοσελίδας. Παρόμοια χρήση του πεδίου Entity Body γίνεται και στην περίπτωση χρήσης της μεθόδου HEAD. Η μέθοδος αυτή είναι χρήσιμη στον εντοπισμό σφαλμάτων (debugging) από τους developers των εξυπηρετητών Ιστού: όταν ο εξυπηρετητής λάβει αίτηση με χρήση της μεθόδου HEAD αποκρίνεται με ένα HTTP μήνυμα στο οποίο δεν περιλαμβάνεται το αρχείο.
Παρακάτω φαίνεται ένα παράδειγμα μηνύματος HTTP απόκρισης.

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html
data data data data data ...
```

Το **HTTP μήνυμα απόκρισης** αποτελείται από τρία μέρη: την **γραμμή κατάστασης (status line)**, τις **γραμμές επικεφαλίδας (header lines)** και το **τμήμα περιεχομένου (entity body)**. Το τμήμα περιεχομένου περιέχει τα δεδομένα του αιτούμενου αρχείου, τα οποία στο παραπάνω παράδειγμα φαίνονται ως data data data data data data … Η γραμμή κατάστασης περιέχει τρία πεδία: το πεδίο *HTTP έκδοσης*, το πεδίο *κωδικού κατάστασης* και το πεδίο του αντίστοιχου *μηνύματος κατάστασης*. Στο παραπάνω παράδειγμα δηλώνεται ότι γίνεται χρήση του HTTP/1.1 και ότι δεν παρουσιάστηκε κάποιο σφάλμα (κωδικός κατάστασης 200 και αντίστοιχο μήνυμα κατάστασης OK).
Στο παραπάνω παράδειγμα περιλαμβάνονται έξι γραμμές επικεφαλίδας. Η γραμμή **Connection:close** δηλώνει ότι μετά την αποστολή του μηνύματος αυτού η TCP σύνδεση

θα τερματιστεί. Η **Date:** γραμμή δηλώνει την ημερομηνία και την ώρα κατά την οποία δημιουργήθηκε και στάλθηκε η απόκριση, η **Server:** γραμμή δηλώνει τον τύπο του εξυπηρετητή Ιστού, η **Last-Modified:** δηλώνει την ημερομηνία και την ώρα κατά την οποία το αντικείμενο δημιουργήθηκε ή τροποποιήθηκε για τελευταία φορά, η γραμμή **Content-Length:** δηλώνει το μήκος των δεδομένων που αποστέλλονται (σε bytes), ενώ η γραμμή **Content-Type:** δηλώνει τον τύπο του αντικειμένου που περιέχεται στο τμήμα περιεχομένου του μηνύματος. Στο παραπάνω παράδειγμα δηλώνεται ότι χρησιμοποιείται ο Apache/1.3.0 (Unix) εξυπηρετητής Ιστού και ότι αποστέλλεται ένα αρχείο HTML μήκους 6821 bytes.
Σε περίπτωση που ο browser χρησιμοποιεί το HTTP/1.0 ακόμη και αν ο εξυπηρετητής Ιστού υποστηρίζει το HTTP/1.1, ο εξυπηρετητής Ιστού πρέπει να χρησιμοποιήσει μη μόνιμες συνδέσεις (δηλαδή στην απόκριση πρέπει να στείλει τη γραμμή Connection:close).
Η δομή των HTTP μηνυμάτων απόκρισης φαίνεται στο παρακάτω σχήμα.



Από το παραπάνω σχήμα βλέπουμε ότι η γενική μορφή του μηνύματος ακολουθεί τη δομή του παραπάνω παραδείγματος. Στον παρακάτω πίνακα φαίνονται οι διάφοροι **κωδικοί κατάστασης των HTTP αποκρίσεων** με την αντίστοιχη σημασία τους.

Κωδικός κατάστασης	**Μήνυμα κατάστασης**	**Επεξήγηση**
200	OK	Επιτυχής αίτηση. Η πληροφορία επιστέφεται στην απόκριση
301	Moved Permanently	Το αιτούμενο αντικείμενο δεν υπάρχει. Η νέα τοποθεσία βρίσκεται στην επικεφαλίδα Location: Το λογισμικό του πελάτη πρέπει αυτόματα να αναζητήσει το αρχείο στο νέο URL.
400	Bad Request	Ο εξυπηρετητής δεν μπορεί να αναγνωρίσει το μήνυμα της αίτησης
404	Not Found	Το αιτούμενο αντικείμενο δεν

		υπάρχει στον εξυπηρετητή
505	HTTP Version Not Supported	Η αιτούμενη HTTP έκδοση δεν υποστηρίζεται από τον εξυπηρετητή

Το HTTP υποστηρίζει περισσότερους τύπους γραμμών επικεφαλίδας από αυτούς που περιγράφηκαν παραπάνω. Όμως, ο αριθμός των επικεφαλίδων που θα συμπεριληφθεί σε μία HTTP αίτηση εξαρτάται από τον τύπο και την έκδοση του browser (π.χ. αν ένας browser υποστηρίζει μόνο το HTTP/1.0 δεν μπορεί να συμπεριλάβει επικεφαλίδες του HTTP/1.1) αλλά και από τις ρυθμίσεις του χρήστη. Το ίδιο συμβαίνει και για τους τύπους των επικεφαλίδων που συμπεριλαμβάνουν οι εξυπηρετητές Ιστού στις HTTP αποκρίσεις.

4.3. Αλληλεπίδραση χρήστη και εξυπηρετητή

Το πρωτόκολλο HTTP ανήκει στα **stateless** πρωτόκολλα, αφού δεν κρατά την κατάσταση των συνδέσεων με τους διάφορους χρήστες – πελάτες. Σε μερικές περιπτώσεις όμως είναι επιθυμητή η αναγνώριση των χρηστών από τον εξυπηρετητή, π.χ. αν ένα σύνολο υπηρεσιών προορίζεται μόνο για κάποια συγκεκριμένη ομάδα χρηστών. Για την αναγνώριση των χρηστών το HTTP προσφέρει δύο μηχανισμούς: την **πιστοποίηση (authentication)** και τα **cookies**.
Πολλές τοποθεσίες του Ιστού απαιτούν την παροχή ενός ονόματος και ενός κωδικού για την πρόσβαση των χρηστών σε ένα σύνολο υπηρεσιών. Η διαδικασία αυτή ονομάζεται **πιστοποίηση (authentication)**. Το HTTP παρέχει ειδικούς κωδικούς κατάστασης και επικεφαλίδες για το λόγο αυτό. Ας υποθέσουμε ότι ένας χρήστης θέλει να πάρει ένα αντικείμενο από τον εξυπηρετητή Ιστού, για το οποίο όμως απαιτείται πιστοποίηση. Τότε ακολουθείτε η παρακάτω σειρά ενεργειών.

1. Ο πελάτης στέλνει μία συνηθισμένη αίτηση στον εξυπηρετητή για την παροχή του αντικειμένου.
2. Ο εξυπηρετητής αποκρίνεται με ένα μήνυμα χωρίς περιεχόμενο και με κωδικό κατάστασης 401 Authentication Required. Σε αυτό το μήνυμα ο εξυπηρετητής περικλείει μία επικεφαλίδα WWW-Authentication, η οποία περιγράφει τις λεπτομέρειες της πιστοποίησης – συνήθως η παροχή από το χρήστη ενός ονόματος και ενός κωδικού.
3. Ο πελάτης λαμβάνει το παραπάνω μήνυμα και καλεί το χρήστη να εισάγει ένα όνομα και τον αντίστοιχο κωδικό. Στη συνέχεια ξαναστέλνει την αίτηση, αλλά αυτή τη φορά περιλαμβάνει και μία Authentication: επικεφαλίδα στην οποία περιλαμβάνονται το όνομα και ο κωδικός που έδωσε ο χρήστης.

Μετά τη λήψη του πρώτου αντικειμένου ο browser συνεχίζει να στέλνει το όνομα και τον κωδικό σε κάθε αίτηση προς τον παραπάνω εξυπηρετητή. (Αυτό συνήθως γίνεται μέχρις ότου ο χρήστης κλείσει τον browser). Αν και ο παραπάνω μηχανισμός δεν καλύπτει θέματα ασφάλειας, παρέχει μία στοιχειώδη μέθοδο πιστοποίησης των χρηστών από τον εξυπηρετητή Ιστού.

Η άλλη μέθοδος για την αναγνώριση των χρηστών από τους εξυπηρετητές Ιστού είναι τα **cookies**. Πολλές τοποθεσίες του Ιστού χρησιμοποιούν cookies ενώ κάποιες άλλες όχι. Ας εξετάσουμε ένα παράδειγμα στο οποίο γίνεται χρήση cookies. Όταν ο πελάτης στείλει για πρώτη φορά αίτηση στην τοποθεσία αυτή στο μήνυμα απόκρισης του εξυπηρετητή περιλαμβάνεται η επικεφαλίδα **Set-cookie:**, η οποία περιέχει έναν αναγνωριστικό αριθμό που δημιουργείτε από τον εξυπηρετητή Ιστού. Π.χ. η επικεφαλίδα αυτή μπορεί να είναι η `Set-cookie: 1678453.` Όταν ο HTTP πελάτης λάβει την απόκριση αυτή τότε προσθέτει μία γραμμή στο τοπικό αρχείο του χρήστη στο οποίο περιέχονται τα cookies. Η γραμμή αυτή συνήθως περιλαμβάνει το όνομα του host καθώς και τον αναγνωριστικό αριθμό του μηνύματος απόκρισης. Αν μετά από κάποιο χρονικό διάστημα ο πελάτης θέλει να ξαναστείλει μία αίτηση στον ίδιο εξυπηρετητή τότε ανακτά από το αρχείο με τα cookies τον αναγνωριστικό αριθμό και στο μήνυμα αίτησης περικλείει μία **Cookie:** επικεφαλίδα, με την οποία παρέχει στον εξυπηρετητή τον αναγνωριστικό αυτό αριθμό. Με τον τρόπο αυτό ο εξυπηρετητής δε χρειάζεται να γνωρίζει το όνομα και τον κωδικό του χρήστη, αλλά του αρκεί να γνωρίζει ότι ο χρήστης αυτός είναι ο ίδιος με αυτόν που είχε κάνει την πρώτη αίτηση. Οι εξυπηρετητές Ιστού μπορεί να χρησιμοποιήσουν τα cookies για διάφορους λόγους, όπως:

- Αν ο εξυπηρετητής χρειάζεται πιστοποίηση για το χρήστη αλλά δεν θέλει κάθε φορά να ζητάει από τον χρήστη την παροχή ονόματος και κωδικού, μπορεί να δημιουργήσει ένα cookie.
- Αν ο εξυπηρετητής χρειάζεται να θυμάται τις προτιμήσεις του χρήστη ώστε να αποστέλει περιεχόμενο ανάλογα με τις προτιμήσεις αυτές.
- Αν ο εξυπηρετητής θέλει να κρατά την κατάσταση για κάποιες από τις ενέργειες που έχει κάνει ο χρήστης, π.χ. σε μία Ιστοσελίδα on-line αγορών την ποσότητα των προϊόντων που θέλει να αγοράσει ο χρήστης.

Από την παραπάνω ανάλυση προκύπτει και το βασικό μειονέκτημα των cookies: ένας χρήστης αναγνωρίζεται ως διαφορετικός αν αλλάξει το μηχάνημα με το οποίο συνδέεται με την παραπάνω τοποθεσία του Ιστού.

4.4. Υπό συνθήκη GET

Με την αποθήκευση των προηγουμένων αντικειμένων που έχουν ληφθεί, μπορεί να μειωθεί ο χρόνος ανάκτησης των διαφόρων αντικειμένων καθώς και το φορτίο στο Διαδίκτυο. Η προσωρινή αυτή αποθήκευση των αντικειμένων συχνά εμφανίζεται στην πλευρά των πελατών του Ιστού.

Αν και η τεχνική της προσωρινής αποθήκευσης των αντικειμένων που έχουν ληφθεί μπορεί να μειώσει τον χρόνο ανάκτησης των Ιστοσελίδων παρουσιάζει ένα νέο πρόβλημα που πρέπει να αντιμετωπιστεί: το αντίγραφο της προσωρινής μνήμης μπορεί να μην είναι πλέον έγκυρο, αφού το αντικείμενο που είναι αποθηκευμένο στον εξυπηρετητή Ιστού μπορεί να έχει στο μεταξύ τροποποιηθεί. Το HTTP παρέχει ένα μηχανισμό που επιτρέπει στους πελάτες να μάθουν αν το αντικείμενο το οποίο έχουν προσωρινά αποθηκεύσει είναι έγκυρο ή όχι. Ο μηχανισμός αυτός αποκαλείτε **υπό συνθήκη GET (conditional GET)**. Μία HTTP αίτηση λέγεται ότι είναι υπό συνθήκη

GET αν χρησιμοποιεί τη μέθοδο GET και αν περιλαμβάνει μία **If-Modified-Since:** γραμμή επικεφαλίδας.
Θα δούμε τον παραπάνω μηχανισμό με ένα παράδειγμα. Στην αρχή ο browser ζητά από τον εξυπηρετητή Ιστού ένα αντικείμενο:

```
GET /fruit/kiwi.gif HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
```

Στη συνέχεια ο εξυπηρετητής Ιστού στέλνει πίσω στον browser το αντικείμενο αυτό:

```
HTTP/1.0 200 OK
Date: Wed, 12 Aug 1998 15:39:29
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24
Content-Type: image/gif
```
data data data data data ...

Όταν λάβει ο browser την παραπάνω απόκριση εμφανίζει το αντικείμενο, αλλά επιπλέον αποθηκεύει ένα αντίγραφό του στην προσωρινή μνήμη του. Παράλληλα στην προσωρινή μνήμη αποθηκεύει και την ημερομηνία και ώρα τελευταίας τροποποίησης που έλαβε στην παραπάνω απόκριση. Μετά από κάποιο χρονικό διάστημα, ο χρήστης ζητά πάλι την ίδια Ιστοσελίδα. Ο HTTP πελάτης όμως έχει στην προσωρινή μνήμη του μία έκδοση της σελίδας αυτής, αλλά δεν γνωρίζει αν είναι ακόμη έγκυρη. Για να μάθει αν το αντικείμενο αυτό είναι ακόμα έγκυρο στέλνει στον εξυπηρετητή μία υπό συνθήκη GET αίτηση:

```
GET /fruit/kiwi.gif HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
If-modified-since: Mon, 22 Jun 1998 09:23:24
```

Στην παραπάνω αίτηση η ημερομηνία και ώρα της επικεφαλίδας **If-Modified-Since:** είναι η ίδια με αυτή της προσωρινής μνήμης του, δηλαδή είναι η ίδια με αυτήν που έλαβε στην απόκριση από τον εξυπηρετητή. Αν το αντικείμενο δεν έχει τροποποιηθεί μετά την ημερομηνία αυτή ο εξυπηρετητής θα στείλει πίσω στον browser το ακόλουθο μήνυμα απόκρισης:

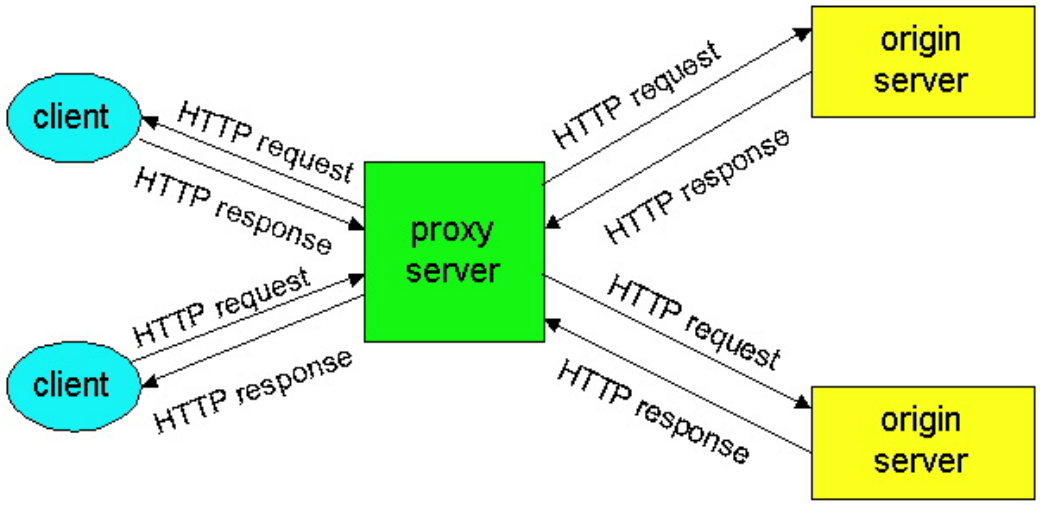
```
HTTP/1.0 304 Not Modified
Date: Wed, 19 Aug 1998 15:39:29
Server: Apache/1.3.0 (Unix)
```
(empty entity body)

Ο εξυπηρετητής, δηλαδή, στέλνει μία απόκριση με κωδικό κατάστασης **304 Not Modified** στην οποία δεν περιλαμβάνει το ίδιο το αντικείμενο. Στην περίπτωση αυτή ο browser μπορεί να χρησιμοποιήσει το αντικείμενο που έχει στην προσωρινή μνήμη του.

4.5. Web cache

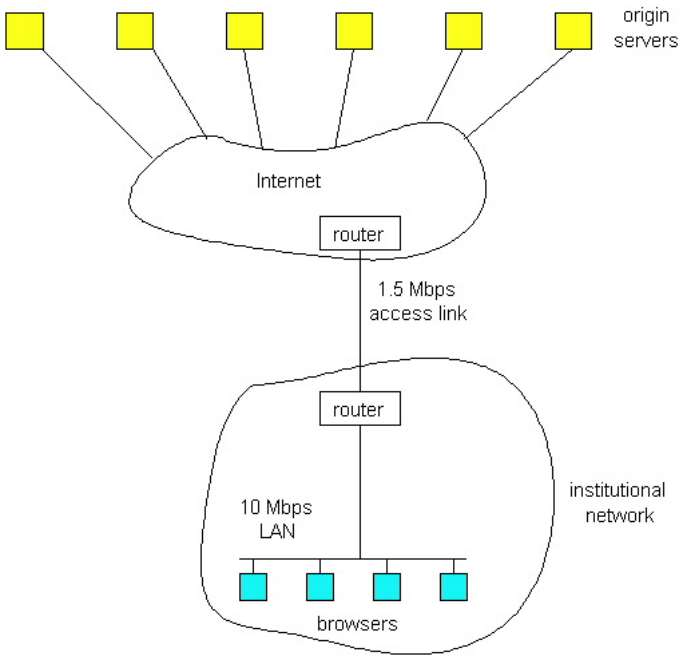
Η **Web cache**, ή διαφορετικά **πληρεξούσιος εξυπηρετητής (proxy server)**, είναι μία οντότητα του δικτύου η οποία ικανοποιεί τις HTTP αιτήσεις για λογαριασμό του πελάτη. Η Web cache έχει το δικό του χώρο αποθήκευσης στον δίσκο, στον οποίο αποθηκεύει όλα τα αντικείμενα που έλαβε πρόσφατα. Οι χρήστες μπορεί να παραμετροποιήθουν τον browser ώστε όλες οι αιτήσεις να απευθύνονται προς την Web cache.
Θα δείξουμε τη λειτουργία της Web cache με ένα παράδειγμα. Έστω ότι ο browser στέλνει μία αίτηση για το αντικείμενο www.ntua.gr/pyrf.gif.

- Ο browser εγκαθιστά μία σύνδεση TCP με τον πληρεξούσιο εξυπηρετητή και στέλνει μία αίτηση για το παραπάνω αντικείμενο στην Web cache.
- Η Web cache ελέγχει για να δει αν έχει ένα αντίγραφο του αντικειμένου αυτού στην μνήμη του. Αν η αναζήτηση είναι επιτυχής, προωθεί το αντικείμενο στον πελάτη μέσω μίας HTTP απόκρισης.
- Αν η Web cache δεν έχει το αιτούμενο αντικείμενο, ανοίγει μία σύνδεση TCP με τον αυθεντικό εξυπηρετητή, στο παράδειγμά μας τον www.ntua.gr, μέσω της οποίας στέλνει την αίτηση HTTP για το αντικείμενο. Όταν ο αυθεντικός εξυπηρετητής λάβει την αίτηση, στέλνει το αντικείμενο στην Web cache, μέσω μίας απόκρισης HTTP.
- Όταν η Web cache λάβει την παραπάνω απόκριση, αποθηκεύει ένα αντίγραφο του αντικειμένου και προωθεί την απόκριση στον πελάτη (μέσω της ήδη υπάρχουσας σύνδεσης TCP).



Από την παραπάνω ανάλυση φαίνεται ότι η Web cache είναι παράλληλα και πελάτης και εξυπηρετητής: όταν λαμβάνει αιτήσεις από τον πελάτη – browser και όταν του στέλνει αποκρίσεις είναι εξυπηρετητής, ενώ όταν στέλνει αιτήσεις ή λαμβάνει αποκρίσεις από τον αυθεντικό εξυπηρετητή Ιστού είναι πελάτης.
Η χρήση της η Web cache έχει τρία βασικά πλεονεκτήματα: πρώτον, μπορεί να μειωθεί ο χρόνος απόκρισης μίας αίτηση από τον πελάτη, ειδικά αν το σημείο συμφόρησης του δικτύου βρίσκεται μεταξύ της η Web cache και του αυθεντικού εξυπηρετητή, που είναι

και η συνηθισμένη περίπτωση. Σε αυτή την περίπτωση η η Web cache μπορεί να παράσχει το αντικείμενο πολύ γρήγορα στον πελάτη. Δεύτερον, μειώνει το φορτίο μεταξύ του ISP και του Διαδικτύου, με αποτέλεσμα να χρειάζονται γραμμές μικρότερου εύρους ζώνης σε σχέση με την περίπτωση που δε γίνεται χρήση της Web cache. Τρίτον, η Web cache μειώνει γενικά το φορτίο στο Διαδίκτυο, με αποτέλεσμα την αύξηση της απόδοσης του Δαιδικτύου, αφού η κίνηση του Ιστού αποτελεί το μεγαλύτερο τμήμα της συνολικής κίνησης του Διαδικτύου.
Στη συνέχεια θα εξετάσουμε ένα παράδειγμα στο οποίο φαίνονται τα πλεονεκτήματα που αναφέρθηκαν παραπάνω. Ας υποθέσουμε την τοπολογία δικτύου του ακόλουθου σχήματος.



Η τοπολογία αυτή αποτελείται από δύο δίκτυα: το Διαδίκτυο (Internet) και το δίκτυο ενός ινστιτούτου (institutional network). Το δίκτυο του ινστιτούτου είναι ένα τοπικό δίκτυο υψηλής ταχύτητας (10 Mbps LAN). Τα δύο δίκτυα συνδέονται μέσω γραμμής 1.5 Mbps. Ο αυθεντικός εξυπηρετητής (origin server) βρίσκεται στο Διαδίκτυο και μπορεί να βρίσκεται σε οποιοδήποτε μέρος του κόσμου. Επίσης υποθέτουμε ότι το μέσο μήκος των αντικειμένων είναι 100 Kbits και ότι ο μέσος ρυθμός αιτήσεων από τους browser του δικτύου του ινστιτούτου προς τον παραπάνω εξυπηρετητή είναι 15 το δευτερόλεπτο. Ο μέσος χρόνος από τη στιγμή που ο δρομολογητής του Διαδικτύου στέλνει μία HTTP αίτηση μέχρι τη στιγμή που λαμβάνει μία HTTP απάντηση από τον αυθεντικό εξυπηρετητή είναι 2 δευτερόλεπτα.
Ο συνολικός χρόνος απόκρισης, δηλαδή ο χρόνος από τη στιγμή κατά την οποία ένας host του τοπικού δικτύου στέλνει μία αίτηση μέχρι τη στιγμή που θα λάβει την απάντηση, είναι ίσος με τη καθυστέρηση στο τοπικό δίκτυο συν την καθυστέρηση μεταξύ των δύο δρομολογητών (του σχήματος) συν την καθυστέρηση για την ανάκτηση του αντικειμένου από τον δρομολογητή του Διαδικτύου. Η πυκνότητα κίνησης στο τοπικό δίκτυο είναι ίση με

$$(15 \text{ requests/sec})*(100 \text{ Kbits/request})/(10\text{Mbps}) = .15$$

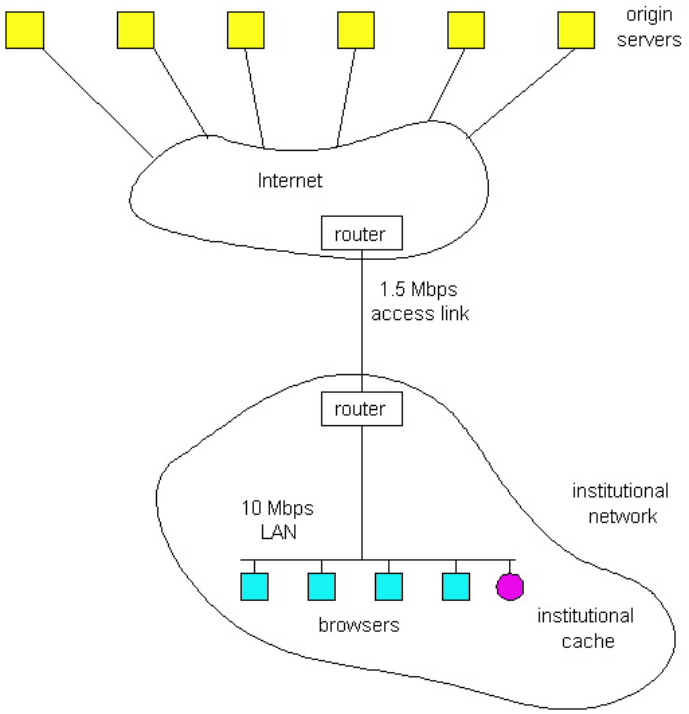
ενώ η πυκνότητα κίνησης μεταξύ των δύο δρομολογητών είναι ίση με

(15 requests/sec)(100 Kbits/request)/(1.5 Mbps) = 1*

Η παραπάνω πυκνότητες κίνησης οδηγούν σε καθυστέρηση κάποιων δεκάδων χιλιοστών του δευτερολέπτου στο τοπικό δίκτυο, αλλά οδηγεί σε μεγάλη καθυστέρηση μεταξύ των δρομολογητών, ακόμη και σε καθυστερήσεις λεπτών.

Για να αντιμετωπιστεί το παραπάνω πρόβλημα υπάρχουν δύο λύσεις. Η πρώτη είναι η αύξηση του εύρους ζώνης της γραμμής μεταξύ των δύο δρομολογητών, π.χ. στην την τιμή των 10 Mbps. Αυτό θα έχει ως αποτέλεσμα τη μείωση της πυκνότητας κίνησης μεταξύ των δύο δρομολογητών στην τιμή 0.15. Η συνολική καθυστέρηση στην περίπτωση αυτή θα είναι ελάχιστα μεγαλύτερη από 2 δευτερόλεπτα, που είναι η καθυστέρηση εντός του Διαδικτύου. Η λύση, όμως, αυτή έχει μεγάλο κόστος.

Μία δεύτερη λύση στο παραπάνω πρόβλημα είναι η εγκατάσταση Web cache στο τοπικό δίκτυο. Η λύση αυτή φαίνεται στο ακόλουθο σχήμα.



Στην περίπτωση αυτή υποθέτουμε ότι η Web cache έχει αποθηκευμένα το 40% τον αιτούμενων αντικειμένων. Οι αιτήσεις για τα αντικείμενα αυτά θα ικανοποιούνται σχεδόν άμεσα, έστω σε χρόνο 10 msec. Το υπόλοιπο 60% των αιτήσεων δεν ικανοποιούνται από την Web cache και στην περίπτωση αυτή η καθυστέρηση θα είναι ανάλογη της περίπτωσης μη χρήσης της Web cache (συν κάποιας καθυστέρησης που εισάγεται από τη χρήση της Web cache). Αν θέλουμε να υπολογίσουμε τη μέση καθυστέρηση αυτή θα δίνεται από τον τύπο

*0.4 * 0.01 + 0.6 * 2.01 = 1.206 δευτερόλεπτα*

ενώ το μέσο φορτίο μεταξύ των δύο δρομολογητών θα είναι

*0.6*15*0.1 = 0.9 Mbps*

Επομένως βλέπουμε ότι η χρήση της Web cache όχι μόνο βελτιώνει το χρόνο απόκρισης αλλά ταυτόχρονα μειώνει και το φορτίο προς το τμήμα του Διαδικτύου και γενικά το φορτίο στο ίδιο το Διαδίκτυο.