

1 Εισαγωγή

1.1 Προοίμιο

Το Διαδίκτυο αποτελεί μια τεχνολογική υποδομή που έχει επηρεάσει τον τρόπο ζωής σε ένα μεγάλο τμήμα του πλανήτη μας, είναι ο βασικός μοχλός της 4ης βιομηχανικής επανάστασης και πάνω στην υποδομή αυτή έχουν αναπτυχθεί εφαρμογές που διευκολύνουν την επικοινωνία, την εργασία, την εκπαίδευση, το εμπόριο, τις οικονομικές συναλλαγές, και πολλές άλλες πλευρές της σύγχρονης ζωής.

Η κατανόηση της αρχιτεκτονικής των σύγχρονων εφαρμογών του διαδικτύου είναι σημαντική για τους επιστήμονες των υπολογιστών και της πληροφορικής αλλά και για το ευρύ κοινό. Το βιβλίο αυτό απευθύνεται κύρια σε φοιτητές τμημάτων υπολογιστών και πληροφορικής, για να χρησιμοποιηθεί ως διδακτικό βοήθημα ενός εξαμηνιαίου μαθήματος, με τίτλο ``Εισαγωγή στον προγραμματισμό του Διαδικτύου" ή ``Εισαγωγή στον προγραμματισμό του Παγκόσμιου Ιστού". Στόχος είναι η εισαγωγή στις τρεις βασικές τεχνολογίες του Ιστού, HTML, CSS, JavaScript. Επιχειρείται επίσης εισαγωγή στην αρχιτεκτονική του διαδικτύου, ξεκινώντας από το μοντέλο πελάτη/εξυπηρετητή και το πρωτόκολλο HTTP. Επίσης στο πλαίσιο των μαθημάτων θα γίνει περιγραφή σύγχρονων τεχνολογιών που επιτρέπουν και διευκολύνουν την ανάπτυξη εφαρμογών, όπως η βιβλιοθήκη Bootstrap που επεκτείνει την CSS και τη JavaScript στην πλευρά του φυλλομετρητή, ενώ στην πλευρά του εξυπηρετητή, βιβλιοθήκες της JavaScript, όπως η express.js η express-handlebars.js η express-session.js, κλπ.

Ο φοιτητής που έχει μελετήσει τα θέματα του βιβλίου και έχει εκπονήσει τις σχετικές εργασίες, θα είναι σε θέση αφενός να κατανοεί τις τεχνολογίες που χρησιμοποιούνται στις διαδικτυακές εφαρμογές σήμερα και διέπουν γενικότερα τη λειτουργία του διαδικτύου, αφετέρου να είναι σε θέση να σχεδιάσει, να αναπτύξει, και να ελέγξει την ορθότητα λειτουργίας μιας σύνθετης διαδικτυακής εφαρμογής που περιλαμβάνει ένα σύνολο διασυνδεδεμένων ιστοσελίδων, που υποστηρίζει διαφορετικούς ρόλους χρηστών, και επιτρέπει τη μόνιμη αποθήκευση δεδομένων. Ο φοιτητής θα πρέπει να είναι σε θέση να κατανοεί τα θέματα ασφάλειας που σχετίζονται με αυτή την εφαρμογή και να ικανοποιεί τις απαιτήσεις ευχρηστίας και προσβασιμότητας.

Το βιβλίο οργανώνεται σε 15 κεφάλαια, το καθένα από τα οποία αντιστοιχεί, όπως περιγράφεται στη συνέχεια, σε ένα εβδομαδιαίο μάθημα. Κάθε κεφάλαιο συνοδεύεται ακόμη από προτάσεις για πρακτικές ασκήσεις που μπορούν να αποτελέσουν το εργαστηριακό σκέλος του μαθήματος.

Στο βιβλίο αυτό έχουμε ενσωματώσει την εμπειρία διδασκαλίας του αντικείμενου για πάνω από είκοσι χρόνια στο Πανεπιστήμιο Πατρών. Το αντικείμενο του μαθήματος άλλαξε ριζικά με την πάροδο του χρόνου αφού είμαστε υποχρεωμένοι να κάνουμε συνεχείς τροποποιήσεις ώστε να συμβαδίσουμε με τις τεχνολογικές εξελίξεις στο

1 Εισαγωγή

αντικείμενο του προγραμματισμού διαδικτυακών εφαρμογών, καθώς και να αντιμετωπίσουμε με όσο καλύτερο τρόπο, την πρόκληση εισαγωγής των φοιτητών μας στο σύντομο χρόνο ενός εξαμηνιαίου μαθήματος σε ένα αρκετά σύνθετο και απαιτητικό πεδίο. Τα τελευταία χρόνια, με την καθιέρωση της HTML5, CSS3 και JavaScript ES6 υπάρχει πλέον ένα ώριμο τεχνολογικά πεδίο το οποίο αποτελεί τη βάση για πιο σύνθετα εργαλεία. Το μάθημα εστιάζει σε αυτές ακριβώς τις τεχνολογίες. Επίσης μια καινοτομία, που έχουμε εισάγει στο σχετικό μάθημα και στο βιβλίο αυτό, είναι να ακολουθήσουμε πλήρως JavaScript προσέγγιση. Στο παρελθόν στις διδακτικές ενότητες που αφορούν τον εξυπηρετητή κάναμε εισαγωγή στην PHP, μία σύνθετη τεχνολογία με ευρεία διάδοση, όμως διαπιστώσαμε ότι η κάλυψη του αντικειμένου ήταν επιφανειακή, δεδομένου του περιορισμένου χρόνου. Συνεπώς αποφασίσαμε να διαθέσουμε περισσότερο χρόνο στην JavaScript δίνοντας βαρύτητα στην τεχνολογία αυτή που έχει ωριμάσει, και η χρήση της επεκτείνεται ραγδαία και στον εξυπηρετητή. Συνεπώς όπως θα διαπιστώσει ο αναγνώστης, προτείνονται κεφάλαια για προχωρημένα θέματα JavaScript, όπως ασύγχρονη λειτουργία και διαχείριση συμβάντων, ενώ διατίθεται επαρκής χρόνος για περιγραφή του Node.js και του πλαισίου Express.js στον εξυπηρετητή.

Σε ένα τυπικό εξαμηνιαίο μάθημα Ελληνικού πανεπιστημίου, προτείνουμε να διατεθεί 1 εβδομάδα για το πρωτόκολλο HTTP και την αρχιτεκτονική του διαδικτύου, 2 εβδομάδες για εισαγωγή στην HTML, 2 εβδομάδες για εισαγωγή στην CSS και το Bootstrap, 4 εβδομάδες για εισαγωγή στην JavaScript, και τέλος 4 εβδομάδες για εισαγωγή στον προγραμματισμό στον εξυπηρετητή με Node.js/ Express.js handlebars και άλλες βιβλιοθήκες της JavaScript που χρειάζονται για αυθεντικοποίηση του χρήστη, σύνδεση στη βάση δεδομένων, διαχείριση συνεδρίας, κλπ. Αναπόσπαστο τμήμα αυτής της προσέγγισης είναι το μάθημα να συνοδεύεται από πρακτικές εργασίες (εργαστηριακές ασκήσεις) κάθε εβδομάδα, και μια ομαδική εργασία που απαιτεί χρόνο περίπου 8 εβδομάδων. Η εργασία έχει τελικό στόχο την ανάπτυξη και διάθεση σε δημόσια ιστοσελίδα μια πλήρους διαδικτυακής εφαρμογής με πολλαπλούς ρόλους χρηστών, και χρήση όλων των τεχνολογιών που περιγράφονται στο μάθημα. Στο βιβλίο υπάρχει επίσης ένα κεφάλαιο για τον canvas (Κεφάλαιο 11), το οποίο είναι προαιρετικό και μπορεί να επιλεγεί αντί για εμβάθυνση σε θέματα βάσεων δεδομένων και αυθεντικοποίησης χρήστη (κεφάλαια 14 και 15) στην περίπτωση που ο διδάσκων επιλέγει να εστιάσει περισσότερο σε προγραμματισμό στην πλευρά του φυλλομετρητή.

Τα εργαλεία που απαιτούνται για την παρακολούθηση του μαθήματος και ιδιαίτερα την εκπόνηση των ασκήσεων και της εργασίας είναι όλα ελεύθερα διαθέσιμα, προτείνεται η χρήση του [VS Code](#) για ανάπτυξη του κώδικα ή εναλλακτικά κάποιου άλλου σύγχρονου εργαλείου ανάπτυξης κώδικα (sublime, atom, κλπ.), το περιβάλλον ανάπτυξης (developer tools) που είναι ενσωματωμένο στις πιο πρόσφατες εκδόσεις των φυλλομετρητών [Chrome](#) και [Firefox](#), το [github](#) για την αποθήκευση του κώδικα της εργασίας, και η πλατφόρμα [Heroku](#) για τη φιλοξενία της εφαρμογής.

Το μάθημα για το οποίο έχει γραφτεί το τρέχον εγχειρίδιο διδάσκεται στο 4ο ή 5ο έτος σπουδών, και προϋποθέτει να έχουν προηγηθεί μαθήματα όπως εισαγωγή στον προγραμματισμό (πχ Python και Java), εισαγωγή στους αλγόριθμους και δομές δεδομένων, εισαγωγή στις βάσεις δεδομένων (σχεσιακό μοντέλο). Επίσης χρήσιμο μάθημα είναι ένα μάθημα αλληλεπίδρασης ανθρώπου- υπολογιστή που εστιάζει σε κανόνες ευχρηστίας και προσβασιμότητας με ιδιαίτερη έμφαση σε σχεδίαση και ανάπτυξη διαδικτυακών εφαρμογών. Παρόλο που το

1 Εισαγωγή

μάθημα προϋποθέτει εξοικείωση με βασικές έννοιες της επιστήμης και τεχνολογίας υπολογιστών, προσπαθούμε να καλύψουμε τις απαιτήσεις και του αρχάριου αναγνώστη, με πολλά απλά εισαγωγικά παραδείγματα και ασκήσεις. Το αντικείμενο του μαθήματος εξάλλου διεγείρει την περιέργεια και του μη ειδικού.

Το βιβλίο αυτό συνοδεύουν μαθήματα στην πλατφόρμα mathesis.cup.gr με αντικείμενο την ανάπτυξη ιστοσελίδων, ενώ οι ασκήσεις και τα παραδείγματα που περιγράφονται στο βιβλίο είναι διαθέσιμα από την σχετική ιστοσελίδα του βιβλίου στο [github](https://github.com).

1.2 Εισαγωγή στην αρχιτεκτονική του διαδικτύου

Το διαδίκτυο ως τεχνολογική υποδομή κλείνει ζωή 40 χρόνων. Τα πρωτόκολλα TCP/IP ορίστηκαν στα μέσα της δεκαετίας του 1970, ενώ τα πρώτα μηνύματα ηλεκτρονικού ταχυδρομείου στάλθηκαν πριν από 50 χρόνια (1972). Ο παγκόσμιος ιστός, η εφαρμογή του διαδικτύου που συντέινε αποφασιστικά στη σημερινή του εξάπλωση και χρήση, έχει συμπληρώσει σχεδόν 30 χρόνια ζωής (1992, Tim Berners Lee). Συνεπώς υπάρχουν σήμερα γενιές, ιδιαίτερα στις δυτικές κοινωνίες που δεν έχουν γνωρίσει τον κόσμο χωρίς το διαδίκτυο.

Το διαδίκτυο στον ανεπτυγμένο κόσμο, που η πρόσβαση αγγίζει το 90% του πληθυσμού, θεωρείται βασική υποδομή όπως τα δίκτυα ενέργειας και τηλεπικοινωνίας. Η διάδοση στη χρήση του συνεχίζει να αυξάνει ραγδαία, αφού και ο αναπτυσσόμενος κόσμος ταχύτατα αυξάνει τη χρήση του. Μάλιστα σε συνδυασμό με την αυξημένη φορητότητα στις υπολογιστικές συσκευές των χρηστών και την ευρεία διάδοση ασύρματων δικτύων υψηλής ταχύτητας, η πρόσβαση σε διαδικτυακούς πόρους αποτελεί πλέον αναπόσπαστο τμήμα της καθημερινής μας ζωής, στο σπίτι, στη δουλειά στις κοινωνικές συναναστροφές.

Θα ξεκινήσουμε το εισαγωγικό αυτό κεφάλαιο με μια περιγραφή βασικών εννοιών της αρχιτεκτονικής του Διαδικτύου.

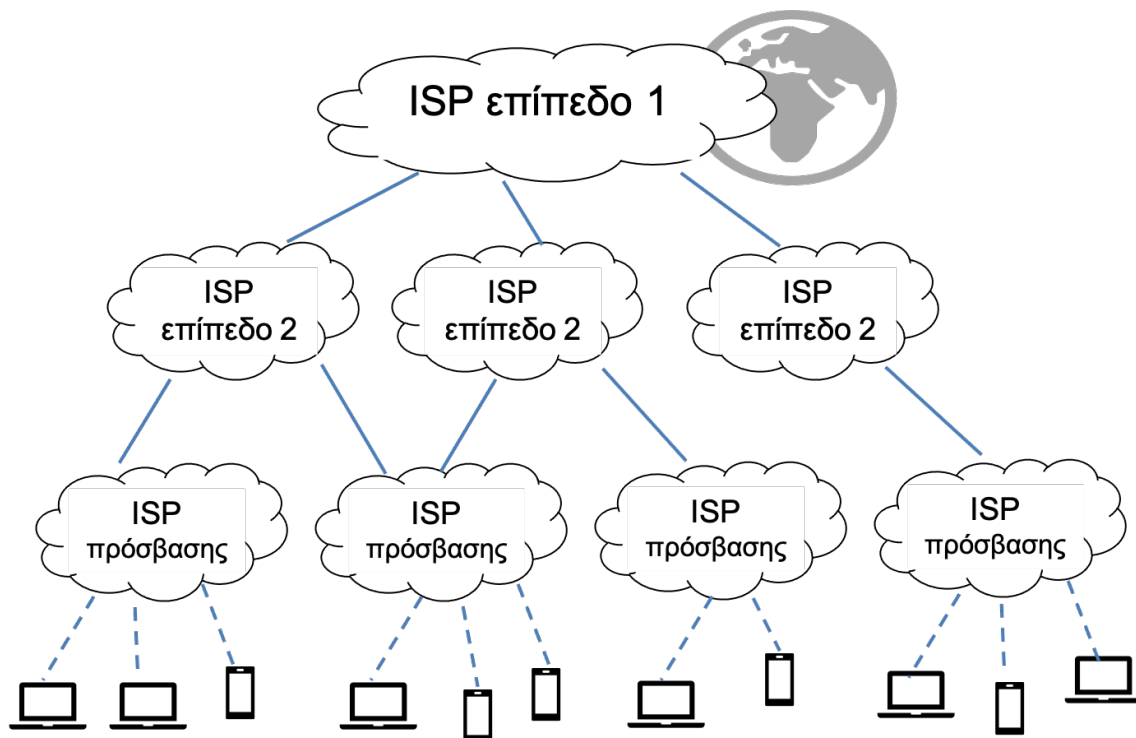
1.2.1 Η αρχιτεκτονική του διαδικτύου

Το **διαδίκτυο** είναι ένα σύνολο από διασυνδεδεμένα δίκτυα υπολογιστών, τα οποία χρησιμοποιούν τα πρότυπα πρωτόκολλα που είναι γνωστά ως **TCP/IP** (Πρωτόκολλο Ελέγχου Μετάδοσης, Transmission Control Protocol, TCP/ Πρωτόκολλο Διαδικτύου, Internet Protocol, IP). Στα πρωτόκολλα αυτά θα γίνει ιδιαίτερη αναφορά σε επόμενη ενότητα.

Μια σύντομη παρουσίαση της αρχιτεκτονικής του διαδικτύου θα πρέπει να κάνει αναφορά στα δίκτυα **Παροχέων Υπηρεσιών Διαδικτύου** (Internet Service Providers, ISP), τα οποία ορίζουν μια ιεραρχία, με καθένα από τα δίκτυα αυτά να έχει διαφορετικό ρόλο και τεχνικά χαρακτηριστικά, όπως η ταχύτητα μετάδοσης και ο όγκος πληροφορίας που διακινούν. Στην κορυφή αυτής της ιεραρχίας είναι ένα σύνολο από δίκτυα που συναποτελούν τον **κορμό του διαδικτύου** (Internet backbone), πρόκειται για σχετικά μικρό αριθμό δικτύων υψηλής ταχύτητας και χωρητικότητας, τα λεγόμενα δίκτυα ISP πρώτου επιπέδου (βλέπε εικόνα 1). Σε αυτά τα δίκτυα συνδέονται δίκτυα με πιο τοπικό χαρακτήρα και διαφορετικά χαρακτηριστικά όσον αφορά

1 Εισαγωγή

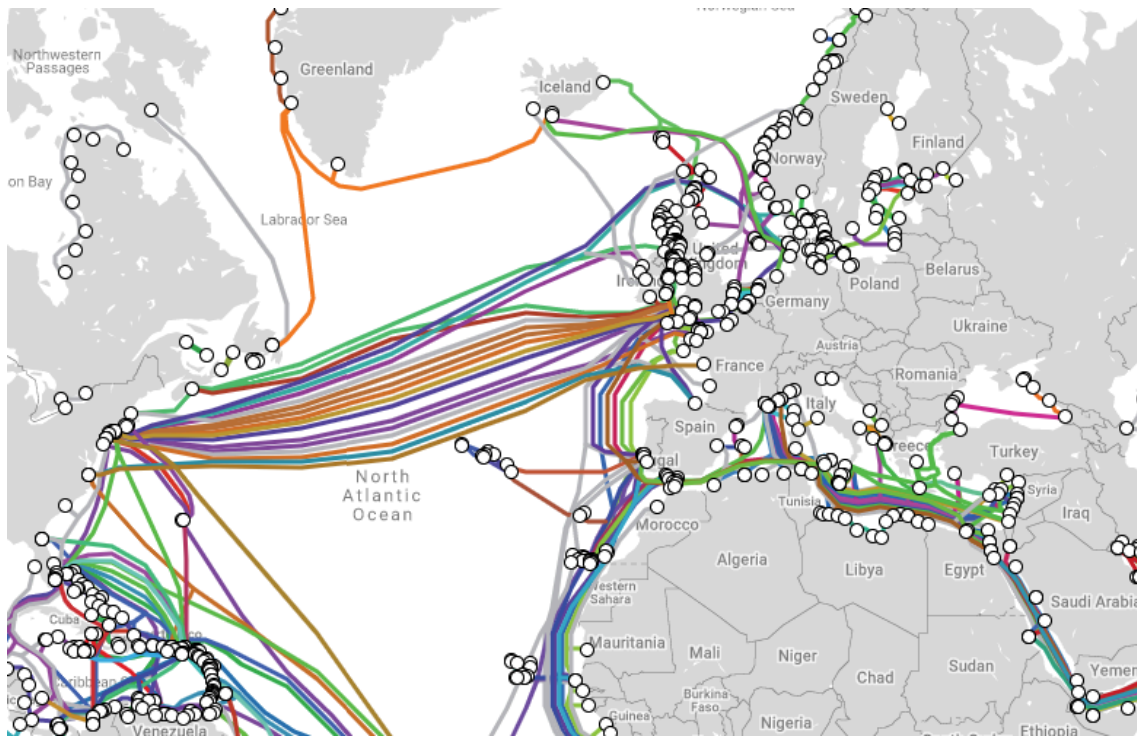
τις δυνατότητές τους (δίκτυα ISP επιπέδου 2). Τα δίκτυα των δύο αυτών κατηγοριών στην ουσία αποτελούν δίκτυα εξυπηρετητών (servers), οι οποίοι παρέχουν τη βασική τηλεπικοινωνιακή υποδομή του Διαδικτύου. Στους εξυπηρετητές αυτούς συνδέονται τα δίκτυα **πρόσβασης ISP** (access ISP), τα οποία παρέχουν υπηρεσίες πρόσβασης στο Διαδίκτυο σε μεμονομένους πελάτες (τερματικούς σταθμούς, hosts). Οι τερματικοί σταθμοί μάλιστα δεν είναι σε όλες τις περιπτώσεις κλασσικοί υπολογιστές, αλλά σήμερα μπορεί να είναι κάμερες ασφαλείας, οικιακές συσκευές, αυτοκίνητα, κλπ. που διαθέτουν σύνδεση στο διαδίκτυο.



Σχήμα 1.1: Αρχιτεκτονική του διαδικτύου ως ιεραρχία δικτύων ISP (παροχών υπηρεσιών διαδικτύου).

Η επικοινωνία μεταξύ δύο τερματικών σταθμών πιθανόν να εμπλέξει ενδιάμεσους εξυπηρετητές που ανήκουν σε διαφορετικά δίκτυα ISP. Το πρώτο πρόβλημα που παρουσιάζεται είναι πώς θα επικοινωνήσουν δύο συνδεδεμένοι υπολογιστές. Σημαντική παράμετρος είναι η ύπαρξη μοναδικών διευθύνσεων όλων των συνδεδεμένων τερματικών σταθμων. Στη συνέχεια θα αναφερθούμε στον τρόπο διευθυνσιοδότησης των υπολογιστών στο διαδίκτυο.

1 Εισαγωγή



Σχήμα 1.2: Χάρτης των υποβρύχιων καλωδίων που παρέχει η ιστοσελίδα

<https://www.submarinecablemap.com/>. Οι συνδέσεις αυτές που έχουν ποντιστεί στο βυθό της θάλασσας ή άλλες αντίστοιχες στην ξηρά, περιλαμβάνουν οπτικές ίνες μετάδοσης δεδομένων υψηλής ταχύτητας που συνδέονται με κόμβους που αποτελούν τον κορμό του διαδικτύου.

1.2.2 Διευθύνσεις υπολογιστών διαδικτύου

Το διαδίκτυο στηρίζεται σε ένα μηχανισμό μοναδικών διευθύνσεων των διασυνδεδεμένων συσκευών, είτε πρόκειται για εξυπηρετητές, είτε για τερματικούς σταθμούς. Αυτές οι διευθύνσεις είναι γνωστές ως διευθύνσεις **IP (Internet Protocol)**. Σύμφωνα με το πρότυπο **IPv4** μια διεύθυνση IP έχει μήκος 32 bit, που σημαίνει ότι ο συνολικός αριθμός διευθύνσεων με αυτό το πρότυπο δεν μπορούσε να είναι μεγαλύτερος από 2^{32} , λίγο περισσότερες από 4 δισεκατομμύρια διευθύνσεις.

Οι διευθύνσεις IPv4 απαρτίζονται από 4 τμήματα των 8 bit (το κάθε τμήμα παίρνει τιμές 0-255), πχ. 172.217.18.100. Τα τμήματα αυτά ορίζουν ιεραρχικά: την κατηγορία του δικτύου, τη διεύθυνση του δικτύου και υποδικτύου (ορίζεται από τη μάσκα υποδικτύου), και τη διεύθυνση του τερματικού σταθμού (host).

Η **μάσκα υπό-δικτύου** καθορίζει τον αριθμό bits που κατανέμονται στη διεύθυνση υποδικτύου και εκείνων που αφορούν τη διεύθυνση του υπολογιστή. Αν n bits αφορούν τη διεύθυνση του υπολογιστή, τότε $32-n$ bits διατίθενται για τη διεύθυνση του δικτύου και υπο-δικτύου.

1 Εισαγωγή

Με τη χρήση μάσκας υποδικτύου είναι δυνατόν σε κάποιο δίκτυο να ορίσουμε για τα διαφορετικά υπο-δίκτυα το πλήθος των υπολογιστών που αυτά περιλαμβάνουν, ανάλογα με τις ανάγκες του οργανισμού ιδιοκτήτη του δικτύου. Είναι φανερό από τα παραπάνω ότι τα τελευταία n bits μιας διεύθυνσης είναι η ταυτότητα του υπολογιστή, δηλαδή το πιο δεξιά τμήμα της διεύθυνσης.

Συνεπώς με τη μάσκα υποδικτύου καθορίζουμε ποιο τμήμα της διεύθυνσης IP είναι η διεύθυνση δικτύου (netid) και ποιο τμήμα είναι η διεύθυνση του υπολογιστή (hostid). Η μάσκα είναι ένας αριθμός 32 bit που έχει ψηφίο 1 για netid και 0 για hostid.

1.2.3 Ασκήσεις

Άσκηση 1: Ζητείται να βρεθεί το μέγεθος ενός υποδικτύου του οποίου η μάσκα είναι η 255.255.255.192.

Απάντηση: Η μάσκα σε δυαδική αναπαράσταση είναι:

11111111 11111111 11111111 11000000

Αυτό σημαίνει ότι οι διευθύνσεις των υπολογιστών του δικτύου αυτού έχουν μήκος 6 bit, άρα συνολικά το δίκτυο αυτό μπορεί να έχει $2^6 = 64$ υπολογιστές.

Άσκηση 2. Έστω υπολογιστής με διεύθυνση IP = 150.251.110.200 που ανήκει στο παραπάνω δίκτυο. Ποια είναι η ταυτότητα του υπολογιστή αυτού (hostid);

Απάντηση: Επειδή όπως είπαμε πιο πάνω τα τελευταία 6 bits της διεύθυνσης αφορούν την ταυτότητα του υπολογιστή, αρκεί να εξετάσουμε το τελευταίο byte της διεύθυνσης, $200_{10} = 11001000$, συνεπώς τα bits που αντιστοιχούν σε hostid είναι "001000" που είναι και η διεύθυνση του υπολογιστή.

Δεδομένης της μεγάλης ανάπτυξης του διαδικτύου και συνεπώς των απαιτήσεων για μοναδικές διευθύνσεις, είμαστε σε σταδιακή μετάβαση στην έκδοση 6 του πρωτοκόλλου (**IPv6**) που προβλέπει διευθύνσεις 128 bit, άρα μέχρι 2^{128} διευθύνσεις, ένας αριθμός συντριπτικά μεγαλύτερος από αυτόν του IPv4 (340 τρις τρις τρις διευθύνσεις).

Μια διεύθυνση IPv6 είναι της εξής μορφής:

2001:0db8:85a3:0000:0000:8a2e:0370:7334

Παρατηρούμε ότι ο αριθμός αναπαρίσταται από οκτώ (8) ομάδες δεκαεξαδικών ψηφίων, χωρισμένων με το σύμβολο ":".

Ο οργανισμός που συντονίζει τη λειτουργία του διαδικτύου και καθορίζει το εύρος συνεχόμενων διευθύνσεων που διατίθεται σε κάθε περιοχή του διαδικτύου είναι ο **ICANN**, Internet Corporation for Assigned Names and Numbers. Η διαχείριση των διευθύνσεων γίνεται σε συνεργασία με την Internet Assigned Numbers Authority (IANA), που συντηρεί τους σχετικούς καταλόγους.

1.2.4 Σύστημα ονομασίας περιοχών DNS

Δεδομένου ότι η αναπαράσταση διευθύνσεων των συσκευών που βρίσκονται συνδεδεμένες στο διαδίκτυο με δυαδικούς ή δεκαδικούς/δεκαεξαδικούς αριθμούς δεν είναι εύχρηστες για τους ανθρώπους, υπάρχει ένας εναλλακτικός μηχανισμός διευθυνσιοδότησης υπολογιστών στο διαδίκτυο, που στηρίζεται σε μια ιεραρχική συγκρότηση μνημονικών διευθύνσεων, οργανωμένων σε "περιοχές" (domains).

Στο πιο υψηλό επίπεδο της ιεραρχίας αυτής έχουμε συνήθως γεωγραφικές περιοχές, όπως η περιοχή ``gr" για διευθύνσεις στην Ελλάδα. Σε κάθε μια από τις περιοχές αυτές ορίζονται υποπεριοχές με ευθύνη του ιδιοκτήτη της περιοχής υψηλού επιπέδου, και κατοχυρώνονται μοναδικά ονόματα που εκπροσωπούν φορείς ιδιωτικούς ή δημόσιους, αλλά ακόμη και φυσικά πρόσωπα. Κάποιες διευθύνσεις, εύκολες στην απομνημόνευση, μπορεί να αποτελέσουν αντικείμενο εμπορικών δοσοληψιών όταν γίνονται διαθέσιμες.

Οι γεωγραφικές περιοχές που έχουν κατοχυρωθεί στο υψηλότερο επίπεδο (δύο χαρακτήρες ASCII, σύμφωνα με την προτυποποίηση ISO 3166-1) είναι 255, μεταξύ των οποίων και περιοχές όπως η .eu για την Ευρωπαϊκή Ένωση. Τα τελευταία χρόνια, δόθηκε η δυνατότητα χρήσης χαρακτήρων πέραν του ASCII στα ονόματα περιοχών, με αποτέλεσμα να έχουν κατοχυρωθεί 316 γεωγραφικές περιοχές (στοιχεία του 2020), μεταξύ των οποίων και η περιοχή ``.ελ", για την Ελλάδα.

Πέραν αυτών, υπάρχουν ακόμη οι επτά αρχικές περιοχές, εκ των οποίων οι τρεις είναι γενικής χρήσης: **.com** commercial (ιδιωτικές επιχειρήσεις), **.org** organization (οργανισμοί), **.net** network (δίκτυα), ενώ υπάρχουν τέσσερις ακόμη με περιορισμούς στη χρήση: **.int** (περιορισμένο σε διακρατικές ενώσεις), **.edu** education (κύρια αμερικανικά πανεπιστήμια), **.gov** government (αμερικανική κυβέρνηση), **.mil** (αμερικανικός στρατός).

Τα τελευταία χρόνια έχει αναπτυχθεί ένας μακρύς κατάλογος από περιοχές πρώτου επιπέδου δημόσια διαθέσιμες ή δεσμευμένες από επιχειρήσεις, για παράδειγμα η περιοχή **.bank** για τον τραπεζικό τομέα, **.museum** για μουσεία, κλπ.

Στην ελληνική περιοχή ``gr", επιτρέπεται η εισαγωγή νέων ονομάτων δεύτερου ή τρίτου επιπέδου, μετά από αίτηση στην αρμόδια αρχή. Στο δεύτερο επίπεδο υπάρχουν κατοχυρωμένα ονόματα, όπως το gov.gr για δημόσιες υπηρεσίες, sch.gr για το σχολικό δίκτυο και edu.gr για εκπαιδευτικούς οργανισμούς.

Μια διεύθυνση με το σύστημα αυτό, έχει το πιο ειδικό στοιχείο στο αριστερό άκρο, ενώ το πιο γενικό στο δεξί άκρο, αντίθετα από τις διευθύνσεις IP που είδαμε στην προηγούμενη ενότητα. Έτσι η διεύθυνση **www.ece.upatras.gr** ορίζει ότι στην περιοχή .gr έχει οριστεί η υποπεριοχή δεύτερου επιπέδου upatras (Πανεπιστήμιο Πατρών), στην περιοχή αυτή έχει οριστεί η περιοχή ece (Τμήμα Ηλεκτρολόγων και Τεχν. Υπολογιστών) στο πλαίσιο της οποίας υπάρχει ο εξυπηρετητής **www**.

Μια διεύθυνση με το σύστημα αυτό αντιστοιχεί σε μια ή περισσότερες διευθύνσεις IP. Για παράδειγμα, στο τερματικό (cmd στο λειτουργικό windows) μπορούμε να δώσουμε την εντολή:

```
$ ping www.ece.upatras.gr
64 bytes from 150.140.189.12: time=18.285 ms
```

1 Εισαγωγή

Σημειώνεται ότι η εντολή `ping` είναι ένα πρόγραμμα για έλεγχο διαθεσιμότητας ενός εξυπηρετητή, και μέτρησης του χρόνου απόκρισής του.

Από την απόκριση αντιλαμβανόμαστε ότι η συγκεκριμένη μνημονική διεύθυνση αντιστοιχεί σε διεύθυνση IP 150.140.189.12.

Καλείστε να χρησιμοποιήσετε την εντολή αυτή για να ελέγξετε τη διεύθυνση IP γνωστών διευθύνσεων εξυπηρετητών διαδικτύου.

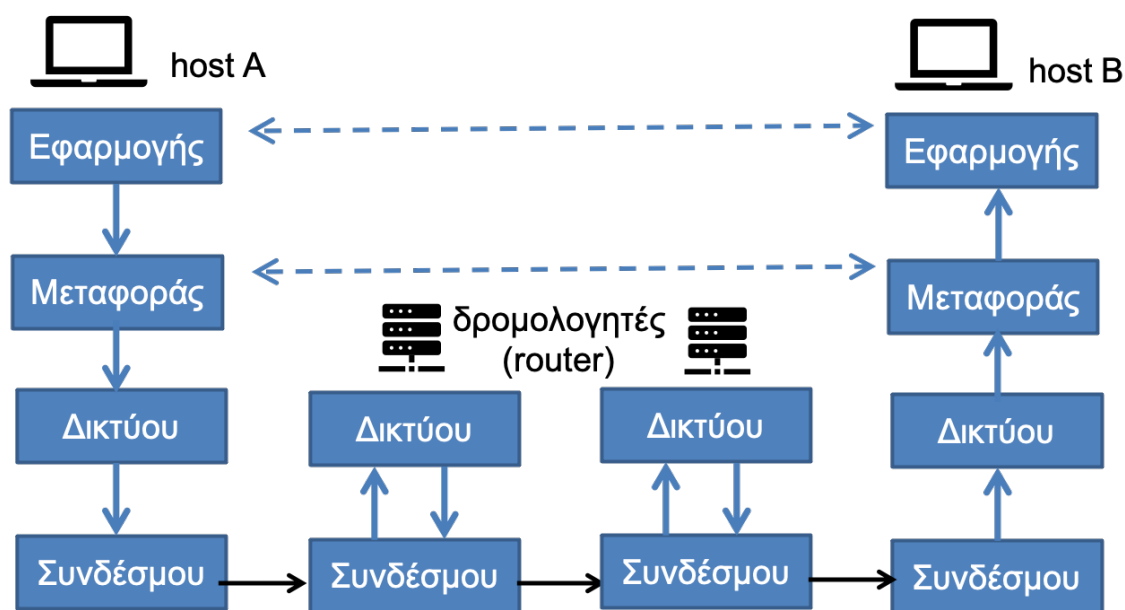
Το ερώτημα είναι πώς ένας υπολογιστής που θέλει να στείλει ένα μήνυμα σε ένα εξυπηρετητή (πχ. όταν πληκτρολογούμε `www.google.com` στον φυλλομετρητή μας), θα ξέρει σε ποια διεύθυνση IP να στείλει το μήνυμα.

Το διαδίκτυο διαθέτει μια υπηρεσία αντίστοιχη των τηλεφωνικών καταλόγων που διατηρεί καταλόγους αντιστοίχισης μνημονικών ονομάτων σε διευθύνσεις IP. Η υπηρεσία αυτή είναι κατανεμημένη ιεραρχικά και στηρίζεται στην ύπαρξη **εξυπηρετητών ονομάτων (name servers)** που διαθέτουν καταλόγους, αυτοί συντηρούνται συνήθως από τους εξυπηρετητές δικτύων ISP. Η αναζήτηση στους εξυπηρετητές ονομάτων λέγεται **DNS lookup**.

1.2.5 Πρωτόκολλα του διαδικτύου

Η επικοινωνία υπολογιστών απαιτεί κοινά πρωτόκολλα, δηλαδή συμβάσεις για τον τρόπο επικοινωνίας και τον τρόπο δόμησης της πληροφορίας που ανταλλάσσεται.

Το διαδίκτυο στηρίζεται σε ένα σύνολο πρωτοκόλλων οργανωμένων σε επίπεδα. Διακρίνονται συγκεκριμένα 4 επίπεδα, που φαίνονται στην εικόνα: τα επίπεδα Εφαρμογής, Μεταφοράς, Δικτύου και Συνδέσμου.



Σχήμα 1.3: Εικόνα 3. Διαστρωμάτωση επιπέδων πρωτοκόλλων του διαδικτύου.

Κάθε επίπεδο επικοινωνεί με εκείνο που βρίσκεται πιο πάνω ή πιο κάτω από αυτό, ανάλογα με τη φορά διάδοσης της πληροφορίας.

Αν υποθέσουμε ότι δύο υπολογιστές (host A και host B), επικοινωνούν. Συγκεκριμένα ο host A αποστέλει δεδομένα στον host B. Τα δεδομένα παράγονται από μια διαδικτυακή εφαρμογή στον A, η εφαρμογή (που βρίσκεται στο **επίπεδο εφαρμογής (application layer)**) είναι υπεύθυνη για επικοινωνία με τον χρήστη. Ανάλογα με την εφαρμογή χρησιμοποιούνται διαφορετικά πρωτόκολλα. Το πιο συνηθισμένο είναι το πρωτόκολλο HTTP για επικοινωνία της εφαρμογής του παγκόσμιου ιστού, ενώ υπάρχουν ακόμη το πρωτόκολλο SMTP για το ηλεκτρονικό ταχυδρομείο, VoIP για ομότιμη σύνδεση και αποστολή δεδομένων φωνής, κλπ. Συνήθως οι εφαρμογές του επιπέδου αυτού ακολουθούν το μοντέλο πελάτη/εξυπηρετητή. Το επίπεδο εφαρμογής μεταφέρει τα δεδομένα στο πιο κάτω επίπεδο που είναι το επίπεδο μεταφοράς για να επικοινωνήσει με την αντίστοιχη εφαρμογή στον υπολογιστή B.

Το **επίπεδο μεταφοράς (transport layer)** διαιρεί τα δεδομένα που πρέπει να αποσταλούν σε μικρά πακέτα, τα αριθμεί ώστε να είναι δυνατή η επανασύνθεση τους στον παραλήπτη και τα μεταφέρει στον παρακάτω επίπεδο δικτύου. Στο επίπεδο μεταφοράς έχουμε δύο πρωτόκολλα:

- Το **Πρωτόκολλο ελέγχου μετάδοσης TCP (Transmission Control Protocol)**, το οποίο εγκαθιστά μια σύνδεση με τον απομακρυσμένο υπολογιστή πριν την έναρξη αποστολής δεδομένων (ορίζεται η θύρα σύνδεσης), ενώ αν κάποιο μήνυμα δεν παραδοθεί επαναποστέλλεται.

1 Εισαγωγή

- Το **Πρωτόκολλο αυτοδύναμων πακέτων χρήστη (User Datagram Protocol, UDP)**, το οποίο είναι πιο γρήγορο από το TCP, αφού δεν εγκαθιστά σύνδεση και δεν κάνει έλεγχο παραλαβής των απεσταλμένων πακέτων. Ο κίνδυνος απώλειας πληροφορίας για κάποιες εφαρμογές, όπως αυτές ροής πολυμέσων και μετάδοσης φωνής δεν είναι απαγορευτικός, ενώ η γρήγορη μετάδοση είναι το κυρίως ζητούμενο.

Το επόμενο επίπεδο είναι το **Επίπεδο δικτύου (Internet Protocol IP)**, το οποίο παραλαμβάνει τα πακέτα που ετοίμασε το επίπεδο μεταφοράς και αναλαμβάνει να τα δρομολογήσει στον επόμενο υπολογιστή, ώστε να πρωτηθούν εγγύτερα στον προορισμό τους με χρήση πινάκων δρομολόγησης. Τα πακέτα είναι παντελώς ανεξάρτητα και πιθανόν να ακολουθήσουν διαφορετικές διαδρομές.

Όταν αποφασιστεί ο επόμενος υπολογιστής στον οποίο θα αποσταλεί το πακέτο, αυτό προωθείται στο πιο κάτω, **Επίπεδο συνδέσμου δεδομένων (data link layer)**, το οποίο αναλαμβάνει την αποστολή του πακέτου, ανάλογα με το δίκτυο στο οποίο είναι συνδεδεμένος ο υπολογιστής, Ethernet, WiFi, κλπ. Στους ενδιάμεσους υπολογιστές (δρομολογητές), το μήνυμα παραλαμβάνεται στο επίπεδο συνδέσμου, μεταφέρεται στο επίπεδο δικτύου και ελέγχεται ο πίνακας δρομολόγησης ώστε να αποφασιστεί ο υπολογιστής στον οποίο πρέπει να προωθηθεί το πακέτο. Τέλος στο σταθμό παραλήπτη, το μήνυμα μεταφέρεται αντίστροφα, από το επίπεδο συνδέσμου, προς το επίπεδο εφαρμογής.

1.2.6 Εφαρμογές του διαδικτύου

Το διαδίκτυο χρησιμοποιείται σήμερα από πολλές εφαρμογές, κάποιες από τις οποίες έχουν οριστεί από τα πρώτα χρόνια του διαδικτύου, κάποιες είναι πιο πρόσφατες. Η εφαρμογή που κυριαρχεί και έχει συμβάλει στη διάδοσή της χρήσης του είναι βεβαίως ο **παγκόσμιος ιστός (world wide web)** που όμως θα αποτελέσει αντικείμενο ξεχωριστών ενοτήτων στη συνέχεια.

Ας κάνουμε μια σύντομη επισκόπηση άλλων εφαρμογών εδώ.

Το **Ηλεκτρονικό ταχυδρομείο (email)** είναι μια από τις πιο παλιές αλλά παραμένει ακόμη μια πολύ δημοφιλής εφαρμογή του διαδικτύου. Η εφαρμογή αυτή, όπως και πολλές άλλες στο διαδίκτυο, ακολουθεί το μοντέλο πελάτη/εξυπηρετητή. Ο **Εξυπηρετητής αλληλογραφίας (email server)** λαμβάνει μηνύματα από τους χρήστες που έχουν εγγραφεί σε αυτόν, και τα προωθεί στους εξυπηρετητές των αποδεκτών του κάθε μηνύματος. Τα μηνύματα αποθηκεύονται στον εξυπηρετητή αυτόν και ο αποδέκτης μπορεί να συνδεθεί στον εξυπηρετητή του για να δει τα μηνύματα που έχουν έλθει για αυτόν.

Τα πρωτόκολλα που σχετίζονται με το email είναι:

Το **Απλό Πρωτόκολλο Μεταφοράς Ταχυδρομείου (Simple Mail Transfer Protocol, SMTP)**, το οποίο ευθύνεται για αποστολή ενός μηνύματος από εξυπηρετητή σε εξυπηρετητή, αλλά και για την αποστολή του μηνύματος από τον υπολογιστή του χρήστη στον εξυπηρετητή του.

Για πρόσβαση στην αλληλογραφία ενός χρήστη (mailbox), μπορεί να χρησιμοποιηθούν δύο πρωτόκολλα:

1 Εισαγωγή

- Το **Post Office Protocol version 3 (POP3)**, το πιο απλό από τα δύο, που απλά κατεβάζει το μήνυμα στον υπολογιστή του χρήστη.
- Το **Internet Mail Access Protocol (IMAP)**, το οποίο επιτρέπει στον χρήστη να αποθηκεύει και να χειρίζεται μηνύματα ταχυδρομείου στον εξυπηρετητή.

Ένα ακόμη πρωτόκολλο με μεγάλη χρήση στην εφαρμογή αυτή, αλλά και με γενικότερη χρήση για κωδικοποίηση πληροφορίας, είναι το **Multipurpose Internet Mail Extension (MIME)**. Το πρωτόκολλο αυτό αντιμετωπίζει το πρόβλημα που έχει το SMTP, που έχει σχεδιαστεί για μεταφορά μηνυμάτων με κωδικοποίηση ASCII και επιτρέπει την αποστολή πολυμέσων, κειμένου με κωδικοποίηση utf-8, κλπ.

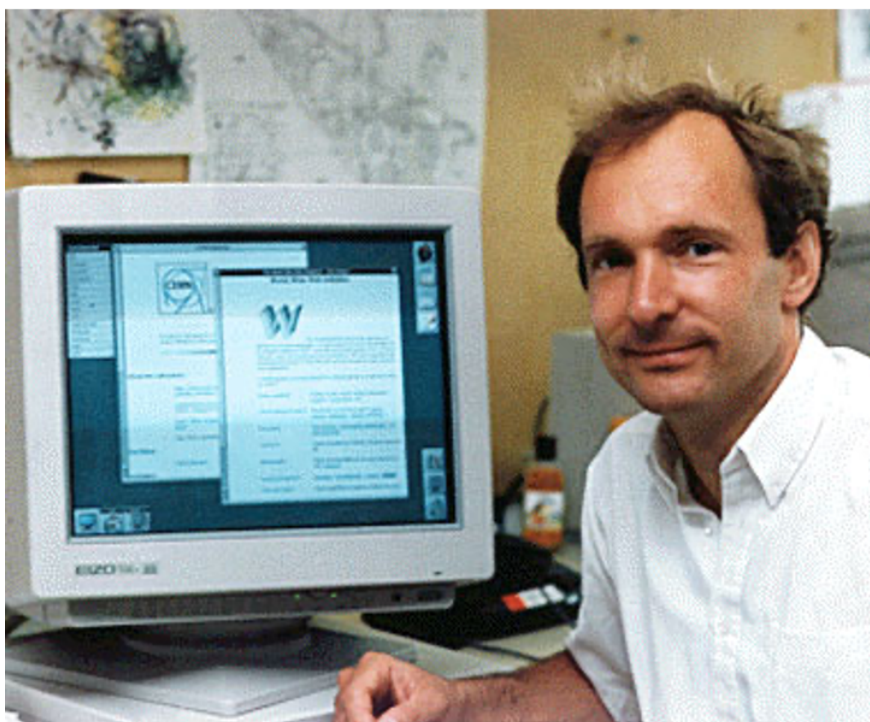
Το δεύτερο παράδειγμα εφαρμογών είναι πιο πρόσφατο και σχετίζεται με εφαρμογές επικοινωνίας, όπως το Skype, Viber, Whatsup, Messenger, κλπ. Όλες αυτές οι εφαρμογές χρησιμοποιούν την υποδομή του διαδικτύου για να επικοινωνήσουν (κλήση) με μια άλλη συνδεδεμένη συσκευή και στη συνέχεια να εγκαταστήσουν μια σύνδεση για την αποστολή μιας ροής δεδομένων ήχου ή/και βίντεο. Τα πρωτόκολλα που χρησιμοποιούν στηρίζονται στο **Voice over Internet Protocol (VoIP)** που επιτρέπει την χρήση του διαδικτύου για παροχή φωνητικής επικοινωνίας μέσω ομότιμης σύνδεσης (peer connection). Τα δεδομένα αποστέλλονται με χρήση του πρωτοκόλλου UDP του επιπέδου μεταφοράς, για αποστολή πακέτων φωνής/ βίντεο, ενός πρωτοκόλλου, το οποίο, όπως περιγράφηκε σε προηγούμενη ενότητα επιτυγχάνει μεγαλύτερη ταχύτητα στη μετάδοση δεδομένων από το συνηθισμένο TCP που χρησιμοποιείται στον παγκόσμιο ιστό, αλλά το οποίο δεν διασφαλίζει μη-απώλεια δεδομένων.

Το διαδίκτυο όπως γνωρίζουμε τείνει να υποκαταστήσει άλλα μέσα, όπως το ραδιόφωνο και η τηλεόραση, προς την κατεύθυνση αυτή συμβάλουν εφαρμογές **πολυεκπομπής (multicasting)** οι οποίες απαιτούν από τους δρομολογητές του διαδικτύου να κατευθύνουν ροές δεδομένων προς πολλαπλούς αποδέκτες, κάτι που η νέα έκδοση του πρωτοκόλλου IPv6 υποστηρίζει σε μεγάλο βαθμό.

1.3 Ο παγκόσμιος ιστός

Ο παγκόσμιος ιστός ως εφαρμογή του διαδικτύου δημιουργήθηκε στις αρχές της δεκαετίας του 1990. Στην εφεύρεση της τεχνολογίας αυτής συνέβαλε σε μεγάλο βαθμό ο βρετανός ερευνητής τότε του Κέντρου Θεωρητικής Φυσικής CERN στην Ελβετία, Tim Berners Lee (βλέπε φωτογραφία 2).

1 Εισαγωγή



Σχήμα 1.4: Ο Tim Berners Lee (CERN, 1992) μπροστά στον πρώτο φυλλομετρητή ιστού.

Ο ερευνητής αυτός του CERN πρότεινε το συνδυασμό της έννοιας του **υπερκειμένου (hypertext)** με το διαδίκτυο. Για το λόγο αυτό εισήγαγε τρεις τεχνολογίες που επιτρέπουν τη διασύνδεση κειμένων που βρίσκονται σε διάφορους εξυπηρετητές του διαδικτύου, μέσω **υπερσυνδέσμων (hyperlinks)**. Οι τεχνολογίες αυτές ήταν το Πρωτόκολλο μεταφοράς υπερκειμένων **Hyper Text Transfer Protocol HTTP**, η γλώσσα σύνταξης υπερκειμένων **HyperText Markup Language HTML** και ο μηχανισμός ονοματοδοσίας των πόρων του ιστού **Universal Resource Locator URL**. Με τις τεχνολογίες αυτές, οι οποίες έκτοτε αναπτύχθηκαν και αποτελούν και σήμερα βασικά συστατικά των εφαρμογών του διαδικτύου, θα ασχοληθούμε στη συνέχεια του βιβλίου. Βεβαίως σήμερα μια ιστοσελίδα δεν περιλαμβάνει μόνο κείμενο, αλλά και πολυμέσα, ενώ η έλευση της γλώσσας προγραμματισμού JavaScript το 1995, έδωσε στην αλληλεπίδραση με μια ιστοσελίδα πιο διαδραστικό χαρακτήρα.

Τι είναι ένα υπερκείμενο; Είναι ένα σύνολο από έγγραφα, το καθένα από τα οποία περιέχει ενσωματωμένους συνδέσμους προς άλλα έγγραφα. Η έννοια του υπερκειμένου ήταν γνωστή πολύ πριν το διαδίκτυο, όμως με το πρωτόκολλο HTTP έγινε δυνατή η διασύνδεση των εγγράφων τα οποία βρίσκονται σε διαφορετικούς υπολογιστές του διαδικτύου, δημιουργώντας στην ουσία ένα ενιαίο πληροφοριακό χώρο.

Η αλληλεπίδραση ενός χρήστη με ένα υπερκείμενο, περιλαμβάνει την επιλογή ενός υπερσυνδέσμου με το ποντίκι συνήθως, ενέργεια που οδηγεί αυτόματα, στην κλήση και φόρτωση της ιστοσελίδας-στόχου. Η ανάκτηση του εγγράφου αυτού γίνεται με χρήση του πρωτοκόλλου HTTP. Η διαδοχική φόρτωση διασυνδεδεμένων ``σελίδων"

1 Εισαγωγή

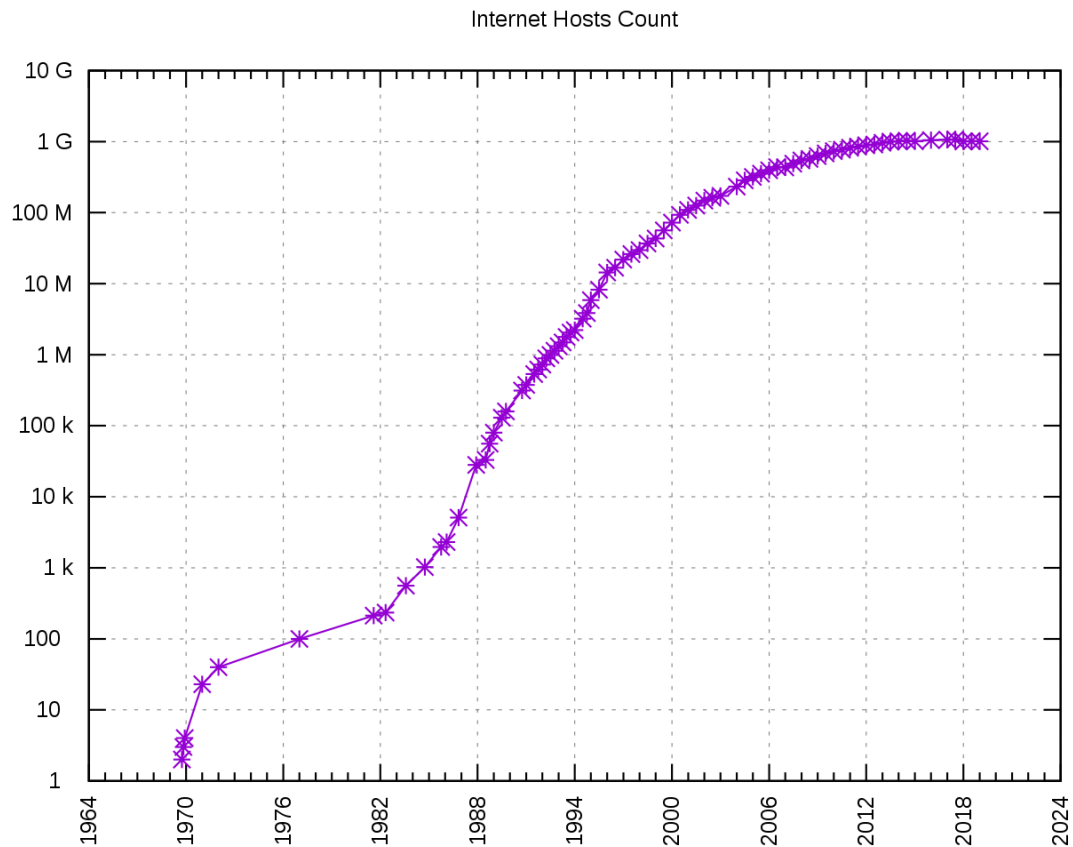
υπερκειμένου παράγει την αίσθηση της ``πλοήγησης" στον παγκόσμιο ιστό.

Η αρχιτεκτονική της εφαρμογής του ιστού είναι αυτή του πελάτη/εξυπηρετητή, όπως εξάλλου οι περισσότερες εφαρμογές του διαδικτύου. Ο ίδιος ο Tim Berners Lee ανέπτυξε τον πρώτο πελάτη της εφαρμογής, τον **φυλλομετρητή ιστού (web browser)**, ο πρώτος αυτός φυλλομετρητής ονομάστηκε WorldWideWeb. Είναι ένα λογισμικό που λειτουργεί στο επίπεδο εφαρμογών του διαδικτύου, και το οποίο με χρήση του πρωτοκόλλου HTTP αποστέλει αιτήματα προς ένα εξυπηρετητή, παραλαμβάνει την απόκριση, και παρουσιάζει την ιστοσελίδα που έχει παραλάβει προς τον χρήστη. Επίσης ο T.Berners-Lee ανέπτυξε τον πρώτο **εξυπηρετητή ιστού (web server)** ο οποίος ονομάστηκε CERN httpd. Ο εξυπηρετητής είναι το λογισμικό που δέχεται αιτήματα από τον πελάτη και αποστέλει περιεχόμενο, συνήθως ιστοσελίδες γραμμένες σε HTML, ενώ σήμερα ένας εξυπηρετητής μπορεί να αποστείλει πολλές διαφορετικές μορφές δεδομένων.

Στη γρήγορη διάδοση του παγκόσμιου ιστού συνέβαλε, το γεγονός ότι η ιδέα εξ αρχής διατέθηκε ελεύθερα, και άλλοι ερευνητές την ανέπτυξαν περαιτέρω. Η εφαρμογή του παγκόσμιου ιστού, συνέτεινε στην ραγδαία ανάπτυξη χρήσης του διαδικτύου. Σήμερα οι περισσότεροι χρήστες του διαδικτύου είναι χρήστες ουσιαστικά της εφαρμογής του παγκόσμιου ιστού, και για το λόγο αυτό οι όροι διαδίκτυο και ιστός συγχέονται.

Στην εικόνα φαίνεται η εκθετική αύξηση των διασυνδεδεμένων υπολογιστών στο διαδίκτυο, από το γράφημα προκύπτει ότι η περίοδος 1980-90 ήταν η περίοδος με την μεγαλύτερη αύξηση.

1 Εισαγωγή



Σχήμα 1.5: Πλήθος εξυπηρετητών διασυνδεδεμένων στο διαδίκτυο, περίοδος 1970-2020 (fig-1-Internet_Hosts_Count_log?).

1.3.1 Ο εξυπηρετητής ιστού

Ένας **εξυπηρετητής ιστού (web server)** είναι μια εφαρμογή που δέχεται αιτήματα μέσω του πρωτοκόλλου HTTP ή της ασφαλούς παραλλαγής του, HTTPS. Μόλις ένας πελάτης, τυπικά ένας φυλλομετρητής κάνει αίτημα για ένα πόρο προς τον εξυπηρετητή, το αίτημα θα πρέπει να ικανοποιηθεί αν ο πόρος είναι διαθέσιμος, ή να σταλεί ένα μήνυμα σφάλματος. Επίσης ο εξυπηρετητής παραλαμβάνει δεδομένα από τον χρήστη, αν αυτό επιτρέπεται και τα αποθηκεύει στον υπολογιστή του εξυπηρετητή.

Εργασίες που αναλαμβάνει ο εξυπηρετητής είναι η μετάφραση του URL path σε όνομα αρχείου στο σύστημα αρχείων του εξυπηρετητή, η εγκυροποίηση του χρήστη, η διαχείριση της ροής δεδομένων για να αποφεύγεται κορεσμός του διαδικτύου και να είναι δυνατή η εξυπηρέτηση πολλών πελατών.

Ας δούμε ένα παράδειγμα μετάφρασης URL path.

1 Εισαγωγή

Αν ο εξυπηρετητής λάβει ένα αίτημα για τον πόρο:

```
http://www.example.com/path/file.html
```

Ο εξυπηρετητής ιστού στο `www.example.com` θα προσαρτήσει τη δεδομένη διαδρομή στη διαδρομή του καταλόγου της ρίζας του ιστότοπου (Host). Σε ένα εξυπηρετητή Apache, αυτός είναι συνήθως `/home/www/website`. Το αποτέλεσμα είναι ο πόρος του τοπικού συστήματος αρχείων:

```
/home/www/www.example.com/path/file.html
```

να αποσταλεί ως απάντηση στο αίτημα.

Η ιστορία των εξυπηρετητών, όπως ήδη αναφέρθηκε, ξεκίνησε από τον **CERN HTTPd** (HTTP daemon). Μια βελτιωμένη έκδοση αναπτύχθηκε το 1993 στο NCSA στο Πανεπιστήμιο Illinois at Urbana--Champaign, με το όνομα **NCSA HTTPd**. Στο πλαίσιο αυτού του εξυπηρετητή δόθηκε η δυνατότητα μέσω Common Gateway Interface (CGI) διασύνδεσης με ένα εξωτερικό πρόγραμμα (γραμμένο σε οποιαδήποτε γλώσσα), και συνεπώς η δυναμική εξυπηρέτηση αιτημάτων ιστοσελίδων.

Όταν η ανάπτυξη αυτού του εξυπηρετητή σταμάτησε, ο κώδικας μεταφέρθηκε στο πρότζεκτ ανοικτού κώδικα **Apache** που σύντομα έγινε ο πιο δημοφιλής εξυπηρετητής και έχει διατηρήσει αυτή τη θέση σχεδόν ως τις μέρες μας. Το όνομα αναφέρεται στη ινδιάνικη φυλή των Απάτσι, φημισμένη για τις δεξιότητες τους στη στρατηγική του πολέμου και την ανεξάντλητη αντοχή τους, αλλά κάνει επίσης ένα χαριτωμένο λογοπαίγνιο για ένα εξυπηρετητή που έχει φτιαχτεί με διαδοχικά μπαλώματα (patches).

Σήμερα υπάρχουν αρκετές διαφορετικές επιλογές εξυπηρετητών, μεταξύ των πιο δημοφιλών είναι ο **Apache** που ήδη αναφέρθηκε, ο **Nginx** που σύμφωνα με κάποιες **στατιστικές** είναι ο πιο δημοφιλής σήμερα εξυπηρετητής, ο **IIS** της εταιρείας Microsoft, κλπ.

Μια εφαρμογή εξυπηρέτησης αιτημάτων ιστού, όπως θα δούμε στη συνέχεια, μπορεί να γραφτεί σε οποιαδήποτε γλώσσα προγραμματισμού. Στα κεφάλαια 12-15 θα δούμε βήμα προς βήμα την ανάπτυξη του εξυπηρετητή μιας εφαρμογής ιστού με χρήση της γλώσσας JavaScript. Εκεί θα χρησιμοποιήσουμε το περιβάλλον Node.js και στη συνέχεια το πλαίσιο Express.js. Στο Node.js μπορούμε να ενσωματώσουμε ένα εξυπηρετητή ιστού με χρήση του module `http` ως εξής:

```
var http = require('http');
http.createServer(function (req, res) {
  res.write('καλή σας μέρα!');
  res.end();
}).listen(8080); //ο εξυπηρετητής αποκρίνεται στη θύρα 8080
```

Ένας εξυπηρετητής ιστού περιέχεται επίσης στο module `Express.js` που θα χρησιμοποιήσουμε στο παράδειγμά μας στο κεφάλαιο 13.

1 Εισαγωγή

Τέλος ένα ακόμη παράδειγμα είναι ο εξυπηρετητής που περιέχεται στο πλαίσιο Flask για ανάπτυξη εφαρμογών ιστού με χρήση της Python.

Ένα παράδειγμα:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return '<h1>καλή σας μέρα!</h1>'
app.run(debug=True) # ο εξυπηρετητής αποκρίνεται στη θύρα 5000
```

Στην περίπτωση αυτή η εφαρμογή χρησιμοποιεί τον εξυπηρετητή που είναι ενσωματωμένος στο πλαίσιο Flask και είναι επαρκής για εξυπηρέτηση αιτημάτων κατά την ανάπτυξη της εφαρμογής (ένα αίτημα κάθε φορά). Όμως όταν η εφαρμογή διατεθεί σε παραγωγική χρήση ένας άλλος εξυπηρετητής πρέπει να χρησιμοποιηθεί όπως ο [Gunicorn](#) ή ο [Nginx](#) ή [Apache](#) μέσω της διεπαφής `mod_wsgi`.

Μια συζήτηση για το θέμα αυτό μπορείτε να αναζητήσετε στο [reddit.com](#)

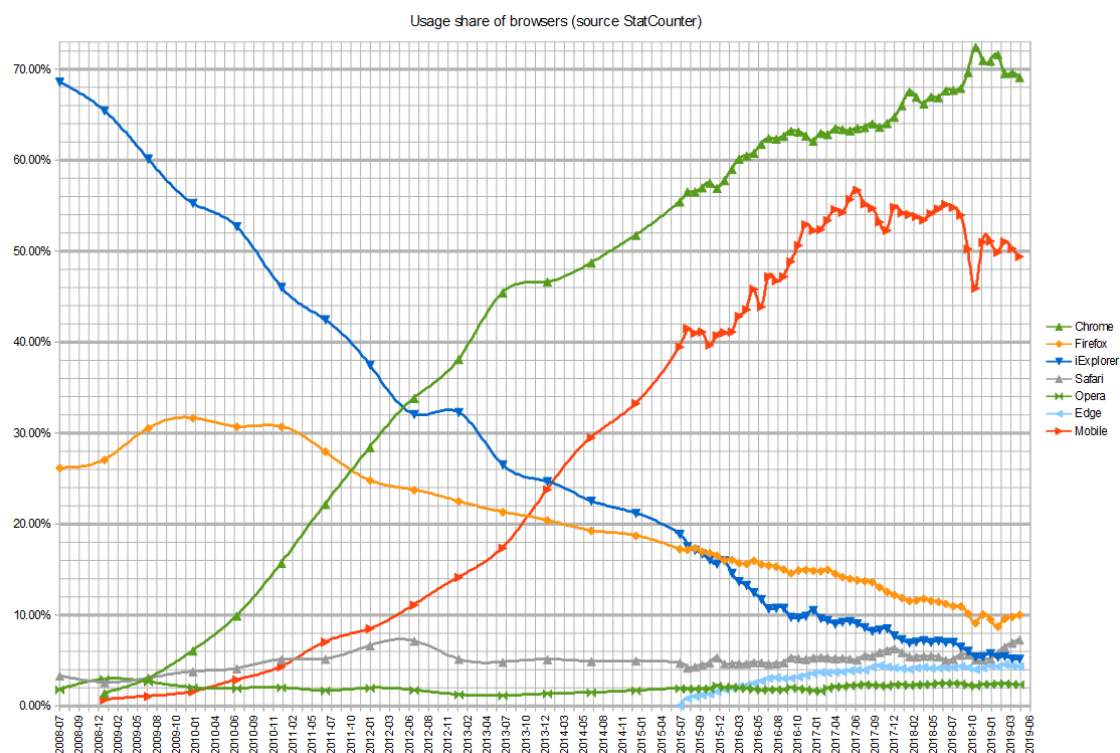
1.3.2 Ο φυλλομετρητής ιστού

Στην ενότητα αυτή θα δούμε τα κύρια χαρακτηριστικά των φυλλομετρητών ιστού (web browsers).

Ιστορικά ο πρώτος φυλλομετρητής δημιουργήθηκε από τον ίδιο τον εφευρέτη του ιστού T. Berners-Lee το 1990 και ονομάστηκε **WorldWideWeb**. Στο Ερευνητικό Κέντρο NCSA του Πανεπιστημίου Illinois, ο Marc Andreessen τότε νέος ερευνητής, πήρε την ιδέα του πρώτου φυλλομετρητή, και την εξέλιξε κύρια με δυνατότητες γραφικής διεπαφής για να δημιουργηθεί ο **NCSA Mosaic**, ο πρώτος φυλλομετρητής με σημαντική διάδοση. Ο ίδιος ο Andreessen σύντομα άφησε το πανεπιστήμιο για να δημιουργήσει μια εταιρία που εξέλιξε περαιτέρω την ιδέα για να δημιουργήσει τον φυλλομετρητή **Netscape Navigator** που υπήρξε ο πρώτος φυλλομετρητής με τεράστια διάδοση, και συνεισφορά στην ανάπτυξη της τεχνολογίας των φυλλομετρητών και των εφαρμογών διαδικτύου γενικότερα. Για παράδειγμα η γλώσσα προγραμματισμού JavaScript, που θα μάς απασχολήσει σε σημαντικό βαθμό στο βιβλίο αυτό, προέρχεται από την ομάδα του Netscape (1995). Ο φυλλομετρητής αυτός το 1995 είχε το 90% των χρηστών του διαδικτύου, όμως έχασε στον λεγόμενο ``πρώτο πόλεμο των φυλλομετρητών'', από τον ανταγωνισμό του **Internet Explorer** της εταιρίας Microsoft, και έφτασε το 2006 να χρησιμοποιείται μόνο από το 1% των χρηστών. Θα πρέπει να σημειωθεί ότι ο Netscape Navigator διατέθηκε στη συνέχεια δημόσια, και αποτέλεσε τη βάση του κώδικα του οργανισμού Mozilla που στη συνέχεια ανέπτυξε τον φυλλομετρητή **Firefox** με βάση τη μηχανή σχεδίασης ιστοσελίδας Gecko.

Η διαμάχη μεταξύ των κατασκευαστών φυλλομετρητών ακολούθησε και την εξέλιξη της τεχνολογίας, και ιδιαίτερα τα νέα πρότυπα HTML5, CSS3 καθώς και την εκτεταμένη χρήση φορητών συσκευών επηρέασε τις επιλογές των χρηστών.

1 Εισαγωγή



Ο ``δεύτερος πόλεμος των φυλλομετρητών" που αφορά την περίοδο από τα μέσα της δεκαετίας του 2000 μέχρι σήμερα, είδε την πτώση του Internet Explorer από το 95% της αγοράς που κατείχε το 2003 σε κάτω από 1% σήμερα. Την ίδια περίοδο αναπτύχθηκε ο **Mozilla Firefox** και στη συνέχεια ο **Google Chrome** ο οποίος σήμερα είναι ο πιο δημοφιλής φυλλομετρητής με πάνω από το 70% των χρηστών.

Ο Chrome χρησιμοποιεί τη μηχανή παρουσίασης της ιστοσελίδας **WebKit**, που είχε αναπτυχθεί στον φυλλομετρητή Safari. Επίσης χρησιμοποιεί τη μηχανή εκτέλεσης κώδικα JavaScript **V8**, που επίσης είναι η μηχανή JavaScript που τρέχει στο περιβάλλον Node.js

Οι φυλλομετρητές των φορητών συσκευών (έξυπνα τηλέφωνα και ταμπλέτες) έχουν σημαντικές διαφορές από αυτούς των επιτραπέζιων υπολογιστών, ενώ η πρόσβαση στο διαδίκτυο μέσω των συσκευών αυτών αυξάνει συνεχώς.

Θα πρέπει να σημειώσουμε ότι σήμερα, οι κύριοι φυλλομετρητές είναι συμβατοί με τις πιο σημαντικές προδιαγραφές των προτύπων HTML5, CSS3, ES6. Όμως παραμένει ένας αριθμός χρηστών σε παλαιότερες συσκευές με εκδόσεις φυλλομετρητών που δεν υποστηρίζουν τα πρότυπα. Ο σχεδιαστής μιας διαδικτυακής εφαρμογής βρίσκεται πάντα μπροστά στο δίλημμα να χρησιμοποιήσει τις πιο τελευταίες δυνατότητες της τεχνολογίας, μη επιτρέποντας σε ένα μικρό ποσοστό των χρηστών να έχει πρόσβαση στην εφαρμογή του, ή να παραμείνει σε παλαιότερες τεχνολογίες; Η ιστοσελίδα <https://caniuse.com/> επιτρέπει να ελέγξουμε τη διάδοση μιας συγκεκριμένης τεχνολογίας σε διαφορετικές εκδόσεις φυλλομετρητών.

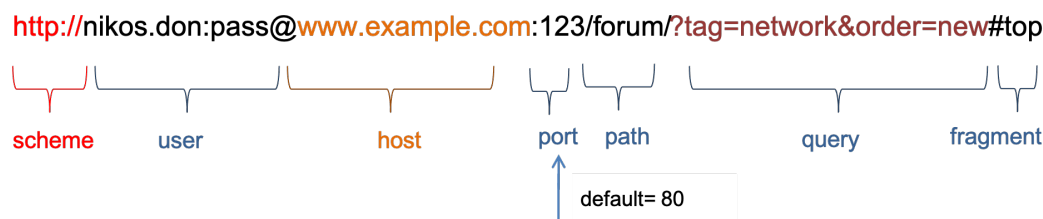
1.3.3 Οι διευθύνσεις URL

Ως τώρα έχουμε διερευνήσει τη βασική λειτουργία και τις τρέχουσες τεχνολογίες των εξυπηρετητών ιστού και των φυλλομετρητών ιστού. Ωστόσο η επικοινωνία μεταξύ των δύο αυτών λογισμικών δεν θα ήταν δυνατή αν δεν είχε οριστεί ένα πρωτόκολλο μοναδικής ονομασίας ενός διαδικτυακού πόρου, όπου πόρος ορίζεται ένα έγγραφο HTML, μια εικόνα, ή ακόμη και μια παράγραφος σε ένα έγγραφο, ώστε να μπορέσει ο πελάτης να ζητήσει τον πόρο από τον εξυπηρετητή.

Η μοναδική αυτή ονομασία ενός πόρου ονομάζεται **Ενιαίος εντοπιστής πόρων (Uniform Resource Locator, URL)**.

Το όνομα αυτό περιέχει πληροφορίες για τον εξυπηρετητή στον οποίο είναι αποθηκευμένος ο πόρος (εδώ θα χρησιμοποιήσουμε τα μνημονικά ονόματα των υπολογιστών που ορίζονται από το σύστημα ονομασίας περιοχών DNS), το πρωτόκολλο μέσω του οποίου θα ανακτήσουμε τον πόρο, τη διαδρομή μέσα στο σύστημα αρχείων του εξυπηρετητή που έχει αποθηκευτεί ο πόρος, ο χρήστης ο οποίος ζητάει πρόσβαση, η θύρα επικοινωνίας με την εφαρμογή, πιθανές μεταβλητές και οι τιμές που τους έχουν δοθεί ως αποτέλεσμα συμπλήρωσης μιας φόρμας και τέλος το επί μέρους στοιχείο του εγγράφου (παράγραφος).

Ας δούμε ένα παράδειγμα:



Σχήμα 1.6: Παράδειγμα URL.

Στο παράδειγμα αυτό βλέπουμε ένα πλήρες όνομα πόρου:

- `http://` ορίζει ότι το πρωτόκολλο ανάκτησης του πόρου είναι το HTTP
- `nikos.don:pass` το δεύτερο στοιχείο καθορίζει ότι ο χρήστης που ζητάει τον πόρο είναι ο `nikos.don` με μυστικό κωδικό `pass`
- `www.example.com` είναι το μνημονικό όνομα του εξυπηρετητή στον οποίο διατίθεται ο πόρος
- `123` είναι ο αριθμός της θύρας στην οποία αποκρίνεται ο εξυπηρετητής, συνήθως για τις εφαρμογές του ιστού, η θύρα είναι η θύρα 80, οπότε αν δεν έχει καθοριστεί θεωρείται η προκαθορισμένη αυτή τιμή.
- `/forum/` αυτή είναι η διαδρομή στο σύστημα αρχείων του εξυπηρετητή που θα αναζητηθεί ο πόρος, αν όπως στο παράδειγμα δεν υπάρχει όνομα αρχείου, αλλά μόνο φάκελος, τότε ο εξυπηρετητής κάνει υπόθεση για προκαθορισμένα ονόματα αρχείων, όπως `index.html`
- `?tag=network&order=new` με το τμήμα αυτό του URL περνάμε ζευγάρια μεταβλητών/τιμών στον εξυπηρετητή. στο παράδειγμα έχουμε ότι `tag = "network"` και `order = "new"`

1 Εισαγωγή

- #top το στοιχείο αυτό ορίζει ότι ζητάμε το στοιχείο με ταυτότητα #top του πόρου. Συνήθως οι φυλλομετρητές σε αυτές τις περιπτώσεις κάνουν κύληση της σελίδας ώστε να τεθεί στην αρχή της σελίδας το στοιχείο με αυτή την ταυτότητα.

1.3.4 Ειδικά σύμβολα και κωδικοποίηση

Κάποιοι χαρακτήρες όπως '&', '+', '%' και ο κενός χαρακτήρας (space) αν υπάρχουν σε ονόματα πόρων ή διαδρομών κωδικοποιούνται ως ``%xx" όπου xx είναι η τιμή ASCII σε δεκαεξαδική μορφή, π.χ. '&' = ``%26", κενό = ``%20". Οι παράμετροι που περιλαμβάνονται στο query string δίδονται ως λίστα παραμέτρων/τιμών που διαχωρίζονται με τον χαρακτήρα '&'

var1=value1&var2=value2&var3=value3

1.3.5 Απλοποίηση URL

Στους περισσότερους φυλλομετρητές, αν παραληφθεί το πρωτόκολλο υπονοείται το http:// ή το https://. Επίσης αν παραληφθεί η διαδρομή (path), υπονοείται τη ρίζα του καταλόγου, και μέσα σε αυτόν ο προκαθορισμένος πόρος index.html, οπότε είναι συχνή πρακτική τα URL να περιέχουν μόνο το μνημονικό όνομα του εξυπηρετητή, π.χ. google.com.

1.4 Το πρωτόκολλο HTTP

Με βάση την γενική συζήτηση που προηγήθηκε, προκύπτει η μεγάλη σημασία του πρωτοκόλλου HTTP για τις εφαρμογές του παγκόσμιου ιστού. Στην ενότητα αυτή θα δούμε τα κύρια χαρακτηριστικά του πρωτοκόλλου αυτού.

Το πρωτόκολλο **HyperText Transfer Protocol (HTTP)** είναι πρωτόκολλο επιπέδου εφαρμογών του διαδικτύου, που καθορίζει τον τρόπο που επικοινωνεί ένας πελάτης με τον εξυπηρετητή σε μια εφαρμογή του παγκόσμιου ιστού.

Οι εκδόσεις του HTTP είναι: HTTP 0.9 (1991), HTTP 1.0 (1996), HTTP 1.1 (1997), HTTP 2.0 (2015). Η πιο διαδεδομένη έκδοση είναι η HTTP 1.1 ενώ η HTTP 2.0 επιφέρει σημαντικές αλλαγές, όπως η κωδικοποίηση των μηνυμάτων σε δυαδική μορφή.

Τα κύρια χαρακτηριστικά του είναι:

- Ακολουθεί την αρχιτεκτονική Πελάτη- Εξυπηρετητή, συνεπώς υπάρχουν δύο είδη μηνυμάτων: αιτήματα και αποκρίσεις.
- Χρησιμοποιεί τις υπηρεσίες του πρωτοκόλλου TCP του επιπέδου μεταφοράς, ή μιας σύνδεσης TCP κρυπτογραφημένης με το πρωτόκολλο Transport Layer Security (TLS), για την περίπτωση του HTTPS.
- Πρωτόκολλο ανταλλαγής δεδομένων σε μορφή κειμένου μέχρι την έκδοση 1.1

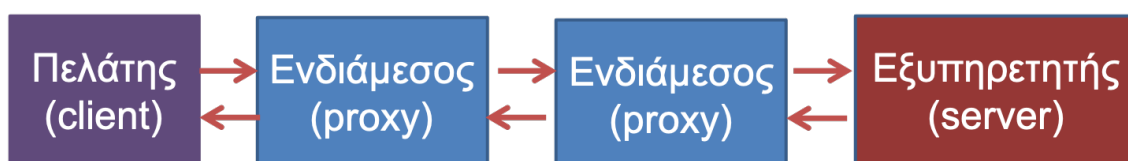
1 Εισαγωγή

- Από την έκδοση HTTP 1.1 επιτρέπονται επίμονες συνδέσεις TCP για πολλαπλές μεταφορές δεδομένων στην ίδια σύνδεση.
- Το πρωτόκολλο δεν θυμάται την προηγούμενη επικοινωνία (Stateless Protocol). Αυτή η ιδιότητα μπορεί να δημιουργήσει πρόβλημα σε μια εφαρμογή τύπου eshop στην οποία απαιτείται συνέχεια, πχ για το περιεχόμενο του καλαθιού αγορών. Για να αντιμετωπιστεί το πρόβλημα αυτό χρησιμοποιούνται cookies.
- Από την έκδοση HTTP 1.1 επιτρέπει Proxy caching, δηλαδή τη χρήση διαμεσολαβητών για προσωρινή αποθήκευση περιεχομένου.
- Επιτρέπει τη διαπραγμάτευση του περιεχομένου (content negotiation), ως προς τα προτιμώμενα χαρακτηριστικά που θα έχει ο πόρος (γλώσσα, κωδικοποίηση, κλπ.).

1.4.1 Διαχείριση ενδιάμεσων μνημών

Στην ανταλλαγή μηνυμάτων HTTP μεταξύ πελάτη και εξυπηρετητή μπορεί να παρεμβληθούν ενδιάμεσοι υπολογιστές οι οποίοι μπορεί να παίξουν διάφορους ρόλους.

Μια εικόνα αυτής της αρχιτεκτονικής φαίνεται στη συνέχεια



Σχήμα 1.7: Σύνδεση πελάτη-εξυπηρετητή με παρεμβολή ενδιάμεσων.

Παραδείγματα αυτών των ενδιάμεσων είναι:

- **προσωρινή μνήμη** είτε δημόσια ή ιδιωτική, όπως η προσωρινή μνήμη του φυλλομετρητή
- **φιλτράρισμα**, όπως η ανίχνευση ιών ή γονικοί έλεγχοι
- **εξισορρόπηση φορτίου**, για να επιτρέπεται σε διαφορετικούς υπολογιστές του εξυπηρετητή να εξυπηρετούν τα διαφορετικά αιτήματα
- **έλεγχος ταυτότητας** για έλεγχο πρόσβασης σε διαφορετικούς πόρους
- **καταγραφή** ώστε να αποθηκεύονται αποθήκευση ιστορικές πληροφορίες.

Μέσω των μηνυμάτων HTTP μπορεί να ελεγχθεί η λειτουργία της επικοινωνίας πελάτη/εξυπηρετητή.

Μερικές από τις παραμέτρους που μπορεί να ελεγχθούν μέσω μηνυμάτων HTTP είναι:

- ενδιάμεση αποθήκευση (caching)
- ταυτοποίηση του χρήστη
- διαχείριση συνεδρίας με cookies
- διαχείριση του περιορισμού same origin

1.4.2 Η ροή επικοινωνίας HTTP

Ας παρακολουθήσουμε την επικοινωνία του φυλλομετρητή μας με ένα εξυπηρετητή, στον οποίο στέλνει ένα αίτημα εξυπηρέτησης, πχ να ζητήσει τον πόρο: <http://www.upatras.gr/el> που είναι η κεντρική ιστοσελίδα του Πανεπιστημίου Πατρών στην Ελληνική γλώσσα.

Θα γίνουν τα εξής βήματα:

- Θα ανοίξει μια **σύνδεση TCP** μεταξύ των δύο υπολογιστών. Όπως θα δούμε στη συνέχεια η διαχείριση της σύνδεσης αυτής μπορεί να γίνει με διαφορετικούς τρόπους.
- Θα σταλεί το **αίτημα HTTP**. Το μήνυμα στην έκδοση HTTP 1.1 είναι αναγνώσιμο από τον χρήστη, ενώ από την έκδοση HTTP 2.0 αν και το περιεχόμενο παραμένει το ίδιο, το μήνυμα κωδικοποιείται σε πλαίσια δυαδικών δεδομένων (frames). Το μήνυμα για το παράδειγμά μας είναι:

```
GET /el HTTP/1.1
Host: www.upatras.gr
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en;q=0.9,el-GR;q=0.8
```

- Η απόκριση στο μήνυμά μας είναι:

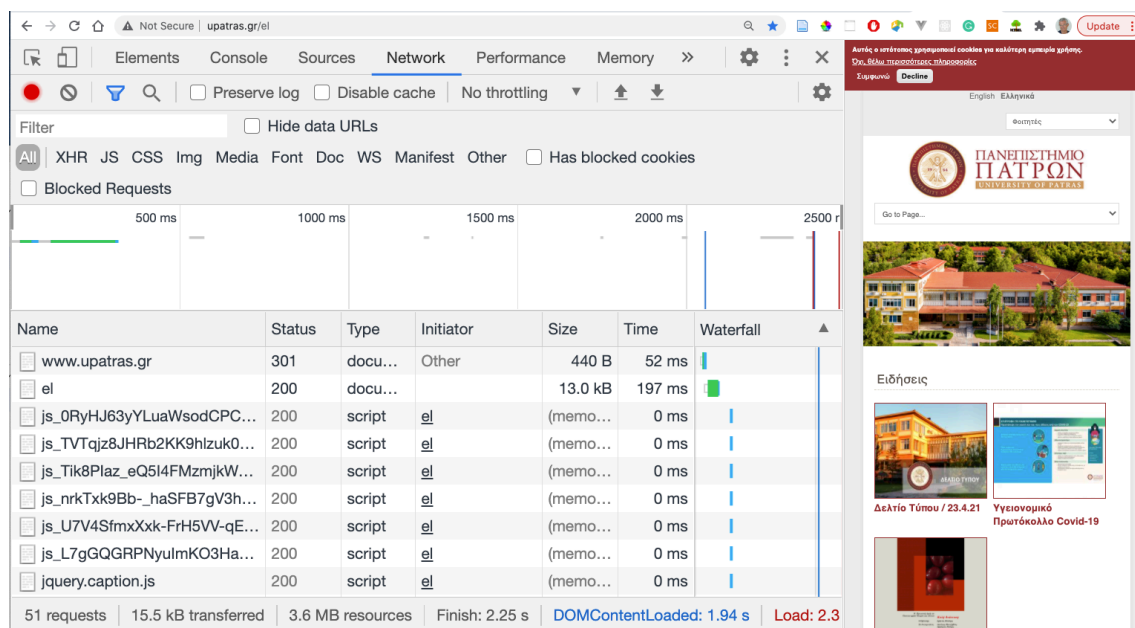
```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Wed, 28 Apr 2021 19:00:28 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Cache-Control: no-cache, must-revalidate
X-Content-Type-Options: nosniff
Content-Language: el
X-Frame-Options: SAMEORIGIN
X-Generator: Drupal 7 (http://drupal.org)
Link: <http://www.upatras.gr/el>
X-XSS-Protection: 1; mode=block
Content-Encoding: gzip
```

1 Εισαγωγή

<!DOCTYPE html> το υπόλοιπο της σελίδας

- Η σύνδεση αυτή θα χρησιμοποιηθεί και για τους άλλους πόρους της ιστοσελίδας.

Για να παρατηρήσουμε τη διαδικασία αυτή και να δούμε τα σχετικά μηνύματα HTTP αρκεί να ανοίξουμε τα εργαλεία σχεδιαστή (επιλογή inspect) στον φυλλομετρητή Chrome ή Firefox και να επιλέξουμε την επιλογή network. Θα παρατηρήσουμε ότι για τη συγκεκριμένη ιστοσελίδα στάλθηκαν 51 αιτήματα HTTP και η συνολική διάρκεια που η σύνδεση παρέμεινε ανοικτή ήταν 2,25 sec.



Σχήμα 1.8: Στα αριστερά φαίνεται η κατάσταση των αιτημάτων HTTP όπως εμφανίζονται στην καρτέλα Network των εργαλείων σχεδιαστή του Chrome. Ενώ στα δεξιά εμφανίζεται η ιστοσελίδα. Αν επιλέξουμε ένα από τα αιτήματα, μπορούμε να δούμε το raw HTTP μήνυμα (αίτημα και απόκριση).

1.4.3 Αιτήματα εξυπηρέτησης HTTP

Ο πρώτος τύπος μηνυμάτων που θα δούμε είναι τα αιτήματα (requests) HTTP.

Ένα αίτημα περιλαμβάνει:

- Τη **μέθοδο HTTP**, όπως η GET, POST ή HEAD, η οποία καθορίζει τη λειτουργία που επιθυμεί ο πελάτης. Συνήθως, ένας πελάτης θέλει να ανακτήσει έναν πόρο (χρησιμοποιώντας τη GET) ή να αποστείλει το περιεχόμενο μιας φόρμας HTML (χρησιμοποιώντας την POST).

1 Εισαγωγή

- Τη **διαδρομή του προς ανάκτηση πόρου**. Αυτή είναι η διεύθυνση URL του πόρου, αφού έχουν αφαιρεθεί τα στοιχεία που είναι προφανή όπως για παράδειγμα το πρωτόκολλο (`http://`), το μνημονικό όνομα του εξυπηρετητή (`www.upatras.gr`) ή η θύρα TCP (80).
- Την **έκδοση του πρωτοκόλλου HTTP**.
- Ακολουθούν οι **κεφαλίδες** του μηνύματος, εκ των οποίων η μόνη που είναι υποχρεωτική είναι η κεφαλίδα `Host` που αφορά τη διεύθυνση του εξυπηρετητή. Υπάρχουν ακόμη προαιρετικές κεφαλίδες που μεταφέρουν επιπλέον πληροφορίες στον εξυπηρετητή.
- Ακολουθεί το **σώμα του μηνύματος**, το οποίο για ορισμένες μεθόδους όπως η `POST` περιέχει τον πόρο που αποστέλλεται (όπως γίνεται και στα μηνύματα απόκρισης).

Οι διαφορετικές μέθοδοι (τύποι) αιτημάτων είναι:

- **GET**: ανάκτηση πόρου (95% των αιτημάτων που διακινούνται είναι αυτού του τύπου).
- **HEAD**: ανάκτηση μεταδεδομένων της κεφαλίδας (πχ., `mod time`)
- **POST**: υποβολή δεδομένων φόρμας στο σώμα του μηνύματος
- **PUT**: αποθήκευση επισυναπτόμενου εγγράφου ως URI (νέο στο HTTP/ 1.1)
- **DELETE**: διαγραφή πόρου (*resource*) (νέο στο HTTP/ 1.1)
- **TRACE**: `http ``echo"` για debugging (νέο στο HTTP/ 1.1)
- **CONNECT**: χρήση από proxies (νέο στο HTTP/ 1.1)
- **OPTIONS**: επιλογές του proxy (νέο στο HTTP/ 1.1)

Η δομή του μηνύματος αίτησης HTTP είναι η εξής:



Σχήμα 1.9: Η δομή του μηνύματος αίτησης HTTP.

1.4.3.1 Κεφαλίδες μηνυμάτων αίτησης

Οι κεφαλίδες ενός μηνύματος αίτησης είναι ζεύγη ιδιότητα:τιμή και καταλαμβάνουν η κάθε μια μια ξεχωριστή γραμμή. Δεν υπάρχει διάκριση μεταξύ κεφαλαίων-πεζών χαρακτήρων.

Οι σπουδαιότερες κεφαλίδες είναι:

Host: Δείχνει το μνημονικό όνομα του εξυπηρετητή και της θύρας σύνδεσης TCP, όπως αυτά ορίζονται στο URL κλήσης του πελάτη. Είναι υποχρεωτική κεφαλίδα στην έκδοση HTTP/1.1 .

1 Εισαγωγή

Accept: Οι κεφαλίδες αυτού του τύπου καθορίζουν επιθυμητούς και αποδεκτούς τύπους δεδομένων από τον πελάτη. Οι διαφορετικοί τύποι διαχωρίζονται με κόμματα. Συχνά ακολουθούνται από έναν αριθμό μεταξύ 0 και 1, που ορίζει τη σχετική βαρύτητα του τύπου. Όταν δεν ορίζεται βαρύτητα, τότε θεωρείται η τιμή 1. για παράδειγμα:

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Υπάρχουν διάφοροι τύποι κεφαλίδων accept, **Accept-Encoding** (κωδικοποίηση), **Accept-Language** (γλώσσα), **Accept** (τύποι δεδομένων, κωδικοποίηση MIME, πχ. text/html)

Connection: keep-alive (για επίμονες συνδέσεις), ή close (για βραχύβιες συνδέσεις).

Content-Length: xxxx, η κεφαλίδα αυτή χρησιμοποιείται μόνο σε αιτήσεις POST. Ο αριθμός xxxx εκφράζει το μέγεθος των αποστέλλομενων δεδομένων του μηνύματος σε bytes (δηλ. χωρίς τις κεφαλίδες). Απαιτείται στο HTTP 1.1 για να καθορίζονται τα όρια τερματισμού και έναρξης κάθε ξεχωριστής αίτησης μέσα από μια επίμονη σύνδεση.

If-Modified-Since: <timestamp> Η κεφαλίδα αυτή δηλώνει ότι ο Πελάτης διαθέτει ήδη ένα αντίγραφο της σελίδας με χρονική σήμανση τροποποίησης timestamp και ζητάει από τον Εξυπηρετητή να επιστρέψει την αιτούμενη σελίδα μόνο εφόσον αυτή έχει τροποποιηθεί μεταγενέστερα της δηλωμένης χρονικής στιγμής.

User-Agent: περιγραφή φυλλομετρητή Η κεφαλίδα αυτή περιέχει του περιβάλλοντος του πελάτη (φυλλομετρητή) που κάνει την αίτηση. Δεν χρησιμοποιείται πάντοτε, ενώ το περιεχόμενο της δεν μπορεί να θεωρείται αξιόπιστο.

1.4.4 Μήνυμα απόκρισης HTTP

Ο εξυπηρετητής αποκρίνεται σε ένα αίτημα με ένα μήνυμα απόκρισης. Η δομή του μηνύματος αυτού είναι η εξής:

Η αρχική γραμμή της απόκρισης ονομάζεται **γραμμή κατάστασης**.

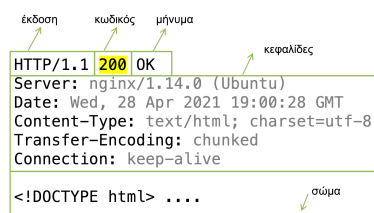
Η αρχική γραμμή αποτελείται από τα εξής τρία στοιχεία χωρισμένα με κενά:

- **Έκδοση πρωτοκόλλου HTTP** (π.χ. HTTP/1.1)
- **Κωδικός κατάστασης απάντησης**(Status code)
- **Μήνυμα περιγραφής** του κωδικού κατάστασης (reason phrase).

Ακολουθούν οι **γραμμές κεφαλίδων** οι οποίες είναι προαιρετικές. Στη συνέχεια ακολουθεί μια κενή γραμμή δύο χαρακτήρων (CRLF). CRLF = \r\n (χαρακτήρας ASCII 13 ακολουθούμενος από τον χαρακτήρα ASCII 10), και μετά το (προαιρετικό) σώμα του μηνύματος (π.χ. ένα αρχείο που μεταφέρεται στον φυλλομετρητή, δεδομένα αναζήτησης ή αποτελέσματα αναζήτησης).

Ακολουθεί το παράδειγμα του μηνύματος απόκρισης της προηγούμενης ενότητας.

1 Εισαγωγή



Σχήμα 1.10: Παράδειγμα μηνύματος απόκρισης HTTP που θα μπορούσε να στείλει ο εξυπηρετητής για να απαντήσει στο προηγούμενο μήνυμα.

Οι δυνατές τιμές που μπορεί να πάρουν οι κωδικοί κατάστασης, που είναι ακέραιοι αριθμοί τριών χαρακτήρων, εκ των οποίων ο πρώτος καθορίζει την κατηγορία της απάντησης, είναι:

- **1xx:** μήνυμα πληροφορίας
- **2xx:** κατάσταση επιτυχούς επεξεργασίας
- **3xx:** προώθηση του μηνύματος σε διαφορετικό παραλήπτη
- **4xx:** σφάλμα προερχόμενο από τον πελάτη
- **5xx:** σφάλμα προερχόμενο από τον εξυπηρετητή.

Πιο ειδικά μερικά από τα πιο συνήθη μηνύματα που μπορεί να επιστρέψει ένας εξυπηρετητής είναι:

- 100 Continue
- 200 OK
- 301 Moved Permanently
- 302 Moved Temporarily
- 304 Not Modified
- 404 Not Found
- 403 Forbidden
- 503 Service Unavailable
- 500 Internal Server Error

1.4.4.1 Κεφαλίδες μηνυμάτων απόκρισης

Ένα μήνυμα απόκρισης μπορεί να περιέχει κεφαλίδες γενικού σκοπού ή ως ειδική επεξήγηση της απόκρισης.

Αν εξετάσουμε ως παράδειγμα την απόκριση της προηγούμενης ενότητας, σχολιάζοντας τις κεφαλίδες:

- **Server:** nginx/1.14.0 (Ubuntu)

Η κεφαλίδα αυτή παρέχει πληροφορία για το λογισμικό του εξυπηρετητή ιστού με τη μορφή `Server: product | comment`, αν η απόκριση διαβιβαστεί μέσω ενός proxy αυτή η πληροφορία δεν τροποποιείται.

1 Εισαγωγή

- Date: Wed, 28 Apr 2021 19:00:28 GMT

Η κεφαλίδα αυτή περιέχει πληροφορίες για την ημέρα και ώρα κατά την οποία γεννήθηκε το μήνυμα

- Content-Type: text/html; charset=utf-8

Η κεφαλίδα αυτή παρέχει πληροφορία για τον τύπο δεδομένων του πόρου. Περιέχει τον τύπο MIME των δεδομένων, το πρότυπο κωδικοποίησης των χαρακτήρων charset, ενώ μπορεί να περιέχει ακόμη πληροφορίες για περιπτώσεις πολλαπλών ομάδων δεδομένων τις συμβολοσειρές οριοθέτησης τους: Content-Type: multipart/form-data; boundary=συμβολοσειρά

- Content-Encoding: gzip

Η κεφαλίδα αυτή συνδυάζεται με την κεφαλίδα Content-type και την κεφαλίδα Transfer-encoding που ακολουθεί για να ορίσει τον τρόπο συμπίεσης της πληροφορίας που στέλνεται. Αφορά ιδιαίτερα πολυμέσα, εικόνες, ήχο, κλπ. Βοηθάει τον πελάτη να αποσυμπίεσει την πληροφορία που λαμβάνει σύμφωνα με τον τύπο MIME των δεδομένων. Τιμές που μπορεί να πάρει αυτή η κεφαλίδα είναι gzip, compress, deflate, br ή και συνδυασμός τους.

- Content-Language: el

Η κεφαλίδα αυτή ορίζει το ακροατήριο για το οποίο προορίζεται η σελίδα. Οι κωδικοί των γλωσσών ορίζονται σύμφωνα με το πρότυπο ISO 639-1 που περιλαμβάνει συχνά και την υποκατηγορία σε μια γλώσσα, πχ ``en-CA" περιγράφει Αγγλικά για τον Καναδά. Περισσότερες από μια γλωσσικές ομάδες μπορεί να οριστούν χωρισμένες με κόμματα.

- Transfer-Encoding: chunked

Η κεφαλίδα αυτή καθορίζει τον τρόπο μετάδοσης της πληροφορίας σε τεμάχια. Συχνά συνοδεύεται από την κεφαλίδα Content-Encoding: που αφορά τον τρόπο συμπίεσης της πληροφορίας που στέλνεται σε τεμάχια. Συνήθίζεται για την αποστολή μεγάλου όγκου πληροφορίας. Στην περίπτωση αυτή παραλείπεται η κεφαλίδα Content-Length: η οποία καθορίζει το μέγεθος της πληροφορίας που επιστρέφει. Η αποστολή των τεμαχίων τερματίζει με την αποστολή ενός άδειου τεμαχίου. Θα πρέπει να σημειωθεί ότι αυτή η επιλογή δεν έχει νόημα στο πρωτόκολλο HTTP/2.0 όπου ο τεμαχισμός και κωδικοποίηση της πληροφορίας ακολουθεί μια άλλη λογική όπως θα εξηγηθεί στη συνέχεια.

- Connection: keep-alive

Η κεφαλίδα αυτή καθορίζει ότι πρόκειται για ``επίμονη σύνδεση" και συνεπώς σε θέση να ικανοποιήσει και άλλα αιτήματα μετά την απόκριση αυτή, σε επόμενη ενότητα θα γίνει πιο λεπτομερής περιγραφή των τύπων σύνδεσης. Άλλη εναλλακτική τιμή που θα μπορούσε να πάρει η κεφαλίδα είναι close που υποδεικνύει βραχύβια σύνδεση.

- Expires: Sun, 19 Nov 1978 05:00:00 GMT

Η κεφαλίδα αυτή χρησιμοποιείται για να οριστεί η ημερομηνία μετά την οποία η πληροφορία θεωρείται ότι παύει να ισχύει. Η συγκεκριμένη τιμή είναι στο παρελθόν, συνεπώς η κεφαλίδα αγνοείται.

1 Εισαγωγή

- Cache-Control: no-cache, must-revalidate

Η κεφαλίδα αυτή καθορίζει την πολιτική ανανέωσης του περιεχομένου σε ενδιάμεσες μνήμες. Η παράμετρος no-cache καθορίζει ότι η πληροφορία μπορεί να αποθηκευτεί σε ενδιάμεσες μνήμες αλλά θα πρέπει να ελεγχθεί με τους μηχανισμούς που διαθέτουν οι ενδιάμεσες μνήμες για την εγκυρότητά της. Αυτό ορίζεται από τη δεύτερη παράμετρο, must-revalidate. Ανάλογα με το περιεχόμενο που αποστέλει ο εξυπηρετητής μπορεί να ορισθούν διαφορετικές τιμές για τον έλεγχο ενδιάμεσων μνημών. Για παράδειγμα no-store σημαίνει ότι η πληροφορία δεν πρέπει να αποθηκευτεί σε ενδιάμεση μνήμη, ενώ η δεύτερη παράμετρος αν πάρει την τιμή immutable, δηλώνει ότι η πληροφορία δεν αλλάζει με τον χρόνο.

- Link: <<http://www.upatras.gr/el>>

Η κεφαλίδα αυτή επιτρέπει στον εξυπηρετητή να συνδέσει τον πόρο που περιέχεται στην απόκριση σε κάποιο σύνδεσμο, ο οποίος περιέχει μεταδεδομένα για τον πόρο, συχνά συνοδεύει πολυμέσα, όπως βίντεο ή εικόνες και περιλαμβάνει πληροφορίες όπως Creative Commons, οδηγίες κλπ. Η τιμή της κεφαλίδας αυτής πρέπει να περιλαμβάνεται σε ετικέτες < > και μπορεί να συνοδεύεται από παραμέτρους όπως rel=meta/preconnect κλπ.

Τέλος στην απόκριση υπάρχουν μια σειρά από κεφαλίδες που αρχίζουν με τον χαρακτήρα ``X-`, μεταξύ των οποίων η κεφαλίδα X-Generator: Drupal 7 (<http://drupal.org>), που υποδεικνύει το λογισμικό που έχει παράξει την απόκριση. Οι κεφαλίδες αυτές όμως δεν είναι στάνταρ και βαίνουν προς κατάργηση, όπως περιγράφεται στο σχετικό [λήμμα της ιστοσελίδας MDN Web Docs: HTTP headers](#).

1.4.5 Διαχείριση σύνδεσης σε αιτήματα HTTP

Ένα πρόβλημα είναι πώς θα διαχειριστούμε τη σύνδεση TCP που θα πρέπει να ανοίξει για την προώθηση ενός αιτήματος HTTP.

Θα πρέπει να σημειωθεί ότι η αποκατάσταση σύνδεσης TCP είναι μια χρονοβόρα διαδικασία γιατί απαιτεί ανταλλαγή τριών μηνυμάτων (SYN, SYN/ACK, ACK) μεταξύ των δύο υπολογιστών. Μια αρχική προσέγγιση, που ήταν η τυπική συμπεριφορά στο πρωτόκολλο HTTP/ 1.0 ήταν να αποκαταστήσουμε μια **βαρχύχρονη σύνδεση (short lived connection)**, όπου έπρεπε για κάθε αίτημα πόρου να ανοίγει νέα σύνδεση TCP η οποία έμενε ανοικτή μέχρι την παραλαβή απάντησης μεταξύ πελάτη και εξυπηρετητή και στη συνέχεια έκλεινε.

Από την έκδοση HTTP 1.1 η προκαθορισμένη συμπεριφορά είναι να εγκαθίσταται μια **επίμονη σύνδεση (persistent connection)**, η οποία παραμένει ενεργή για αρκετό χρόνο ώστε να καταστεί εφικτή η αποστολή όλων των αρχείων που συγκροτούν την ιστοσελίδα (εικόνες, αρχεία css, js, κλπ.). Με την παράμετρο Keep-Alive της κεφαλίδας ενός αιτήματος HTTP μπορούμε να ορίσουμε τη διάρκεια της επίμονης σύνδεσης.

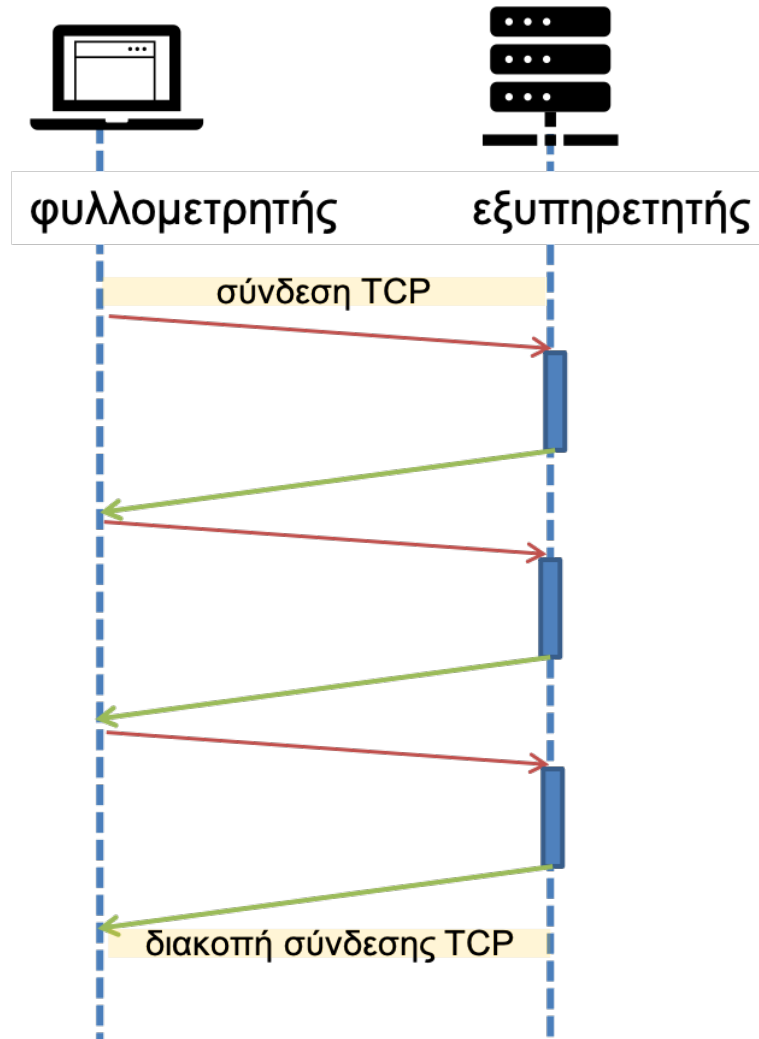
Για παράδειγμα με τη δήλωση:

Keep-Alive: timeout=5, max=100

1 Εισαγωγή

ορίζουμε ότι η σύνδεση θα μείνει ενεργή για 5sec και στο πλαίσιο της να επιτρέψει μέχρι 100 αιτήματα για πόρους.

Στο παρακάτω σχήμα φαίνεται αυτός ο τύπος σύνδεσης.



Σχήμα 1.11: Επίμονη σύνδεση.

Αυτός ο τρόπος διαχείρισης της σύνδεσης είναι πιο αποδοτικός από τον προηγούμενο της βραχύχρονης σύνδεσης.

Πρέπει να σημειωθεί ότι υπάρχει μια ακόμη, εναλλακτική μέθοδος της **διεκπαιρέωσης (pipelining)** σύμφωνα με την οποία ο πελάτης στέλνει ανεξάρτητες μεταξύ τους αιτήσεις, διαδοχικά μέσα από το επίμονο κανάλι επικοινωνίας, χωρίς να περιμένει πρώτα τον εξυπηρετητή να απαντήσει στην προηγούμενη αίτηση. Σε αυτή τη

1 Εισαγωγή

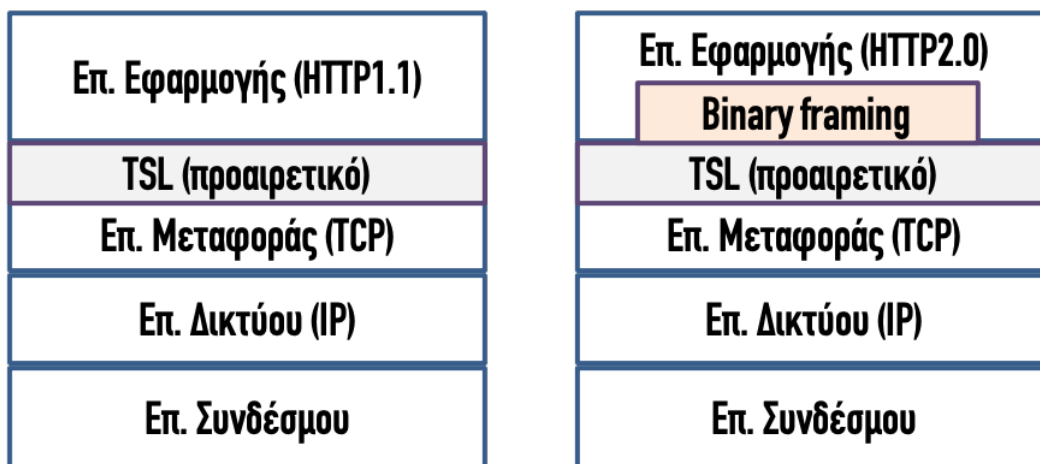
διαδικασία, εξαιρείται η αίτηση και απάντηση για τον πόρο του εγγράφου HTML, καθώς αυτός ορίζει ποιοι θα είναι οι επόμενοι πόροι που θα ζητήσει ο φυλλομετρητής. Η μέθοδος αυτή όμως μπορεί να προκαλέσει θέματα σε περίπτωση που κάποιο από τα παράλληλα αιτήματα αποτύχει, και για τον λόγο αυτό δεν είναι η προκαθορισμένη συμπεριφορά σε κανένα φυλλομετρητή.

Μια εναλλακτική τεχνική είναι η **πολυπλεξία των αιτημάτων** που έχει εισάγει η έκδοση HTTP/ 2.0, που θα περιγραφεί στη συνέχεια.

1.4.6 Πολυπλεξία μηνυμάτων με την HTTP 2.0

Το νέο πρωτόκολλο HTTP/2.0 δεν αλλάζει τη σημασιολογία του HTTP/1.1 καθώς και τις μεθόδους του πρωτοκόλλου.

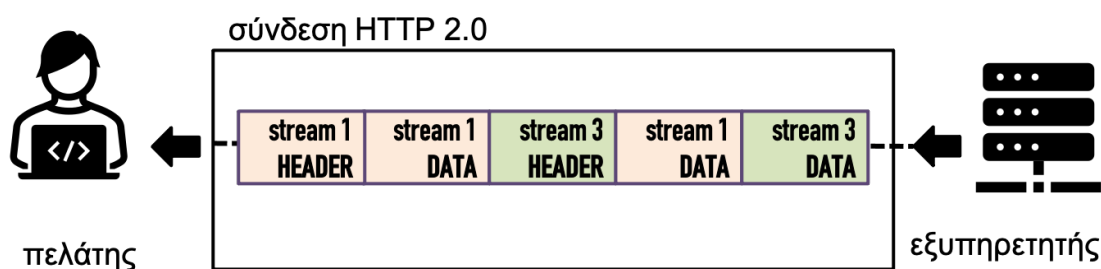
Όμως αλλάζει τον τρόπο που κωδικοποιούνται και μεταδίδονται τα μηνύματα, ουσιαστικά βελτιώνοντας την απόδοση στην μετάδοση μεγάλου όγκου πληροφορίας που περιέχεται στις σύγχρονες ιστοσελίδες. Η διαστρωμάτωση των πρωτοκόλλων με το HTTP/1.1 και το HTTP/2.0 φαίνονται στην παρακάτω εικόνα.



Σχήμα 1.12: Διαστρωμάτωση εφαρμογής ιστού, αριστερά: HTTP 1.1, δεξιά HTTP 2.0.

Η μόνη διαφορά ότι παρεμβάλεται μια διαδικασία κωδικοποίησης και τεμαχισμού του μηνύματος σε μια σειρά από δυαδικά πλαίσια (binary frames), πριν τα δεδομένα μεταφερθούν στο πιο χαμηλότερο επίπεδο μεταφοράς.

Κάθε μήνυμα κωδικοποιείται και χωρίζεται σε binary frames, ξεχωριστά για το τμήμα της κεφαλίδας και του σώματος δεδομένων. Τα πλαίσια αυτά δυαδικών δεδομένων στέλνονται ανεξάρτητα το ένα από το άλλο, πολλά δε μηνύματα μπορούν να περιπλεχθούν στο ίδιο κανάλι.



Σχήμα 1.13: Ροή δυαδικών πλαισίων δεδομένων στο πρωτόκολλο HTTP 2.0.

1.4.7 Διαπραγμάτευση περιεχομένου

Στο πλαίσιο ενός αιτήματος HTTP μπορεί να γίνει διαπραγμάτευση του περιεχομένου ενός αιτήματος. Ο πελάτης και ο εξυπηρετητής μπορούν να συμφωνήσουν π.χ. στην αποστολή συμπιεσμένης μορφής gzip ενός εγγράφου υπερκειμένου ή άλλου πόρου, εφόσον και οι δύο το υποστηρίζουν.

Στην περίπτωση αυτή, η συμπίεση γίνεται αδιαφανώς μέσα από το κανάλι επικοινωνίας, ενώ το περιεχόμενο του πόρου εμφανίζεται στον τελικό χρήστη ως ασυμπίεστο. Επίσης άλλα σημεία για τα οποία μπορεί να γίνει διαπραγμάτευση είναι η κωδικοσελίδα χαρακτήρων εγγράφου, ο τύπος εικόνων του εγγράφου, η εξυπηρέτηση της αίτησης, μόνο εφόσον ο πόρος έχει τροποποιηθεί, κλπ.

Οι ιδιότητες της κεφαλίδας του μηνύματος που χρησιμοποιούνται για τη διαπραγμάτευση αυτή είναι:

- Accept: maintype/subtype όπου ορίζονται οι προτιμήσεις ως προς τον τύπο περιεχομένου με χρήση κωδικοποίησης MIME, πχ text/html, image/jpeg, application/octet-stream
- Accept-Encoding, ορίζει τον τύπο κωδικοποίησης συμπίεσης πχ. Accept-Encoding: br, gzip;q=0.8
- Accept-Language, ορίζει την προτιμώμενη γλώσσα, μάλιστα ορίζοντας την ποιότητα: Accept-Language: fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5

1.4.8 Διαχείριση Cookies

Cookies ορίζονται ως δεδομένα, συνήθως μικρού μεγέθους, τα οποία ο εξυπηρετητής αποστέλει στον φυλλομετρητή για προσωρινή αποθήκευση ώστε να χρησιμοποιηθούν στη συνέχεια. Τα cookies χρησιμεύουν για να εξασφαλίσουν τη διατήρηση της συνέχειας σε διαφορετικά αιτήματα από τον ίδιο φυλλομετρητή, για παράδειγμα σε μια εφαρμογή ηλεκτρονικού καταστήματος. Το πρόβλημα που προσπαθεί να αντιμετωπίσει η χρήση των cookies είναι ότι το πρωτόκολλο HTTP δεν ``θυμάται`` την προηγούμενη επικοινωνία (stateless protocol), άρα κάθε νέο αίτημα από ένα φυλλομετρητή δεν μπορεί να συνδεθεί με την ως τότε επικοινωνία. Ο τρόπος να ληθεί το πρόβλημα είναι το cookie που αρχικά στάλθηκε από τον εξυπηρετητή να επισυναφθεί σε κάθε νέο αίτημα του φυλλομετρητή ως απόδειξη της ταυτότητάς του. Άλλες χρήσεις της τεχνικής αυτής είναι για

1 Εισαγωγή

προσωποποίηση της πληροφορίας, αφού στα cookies μπορεί να αποθηκευτούν οι προτιμήσεις του συγκεκριμένου χρήστη, ενώ μπορεί ακόμη να χρησιμοποιηθούν για καταγραφή της συμπεριφοράς του χρήστη, κάτι που μπορεί να παραβιάζει την αρχή προστασίας των προσωπικών δεδομένων. Για τον λόγο αυτό η Ευρωπαϊκή Ένωση στη νομοθεσία για προσωπικά δεδομένα (GDPR) απαιτεί την ενημέρωση του χρήστη κάθε φορά που χρησιμοποιείται η τεχνολογία αυτή.

1.4.8.1 Δημιουργία cookies

Η δημιουργία cookies από τον Εξυπηρετητή γίνεται με χρήση της κεφαλίδας Set-Cookie στην απόκριση του εξυπηρετητή.

Στο παρακάτω παράδειγμα η κεφαλίδα του μηνύματος κοινοποιεί στον πελάτη ότι θα πρέπει να αποθηκευτούν δύο cookies, το καθένα με τιμή μια συμβολοσειρά

```
HTTP/2.0 200 OK
Content-Type: text/html
Set-Cookie: cookie1="τραγανιστό μπισκότο"
Set-Cookie: cookie2="μουστοκούλουρο"
```

Στη συνέχεια σε κάθε νέο αίτημα του πελάτη προς τον εξυπηρετητή, αποστέλλονται τα δύο cookies.

```
GET /sample_page.html HTTP/2.0
Host: www.example.org
Cookie: cookie1="τραγανιστό μπισκότο"; cookie2="μουστοκούλουρο"
```

Η παράμετρος Expires ενός cookie ορίζει τη διάρκεια ζωής του. Για παράδειγμα:

```
Set-Cookie: cookie2="μουστοκούλουρο"; Expires=Thu, 31 Oct 2021 07:28:00 GMT;
```

Επίσης η παράμετρος Secure επιβάλλει την πρόσβαση στο cookie μέσω ασφαλούς σύνδεσης https και όχι απλής http. Τέλος η παράμετρος HttpOnly ορίζει ότι στο cookie η πρόσβαση γίνεται μόνο μέσω της HTTP και όχι μέσω της JavaScript.

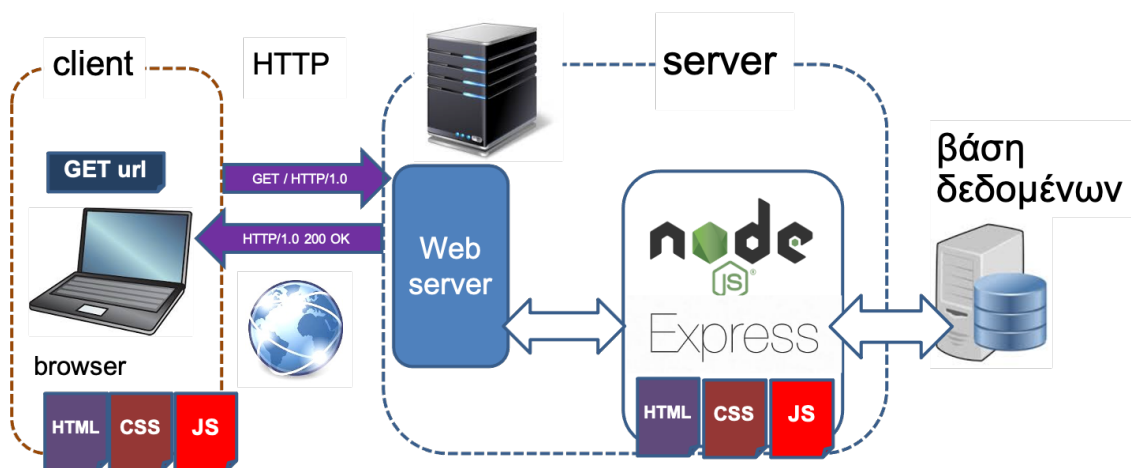
Τα cookies μπορούν να δημιουργηθούν μέσω της ιδιότητας document.cookie από την JavaScript ως εξής:

```
document.cookie = "cookie2=μουστοκούλουρο";
```

Τα cookies που δημιουργούνται με αυτόν τον τρόπο δεν μπορούν να δεχτούν ως παράμετρο την HttpOnly.

1.5 Η αρχιτεκτονική των τεχνολογιών του βιβλίου

Με βάση τη γενική επισκόπηση της ιστορίας και της εξέλιξης του διαδικτύου που έγινε σε αυτό το πρώτο κεφάλαιο, που δόθηκε ιδιαίτερη έμφαση στην εφαρμογή του παγκόσμιου ιστού και το πρωτόκολλο που διέπει την επικοινωνία πελάτη/εξυπηρετητή στο πλαίσιο του, το HTTP, θα περιγράψουμε στη συνέχεια το αντικείμενο του βιβλίου αυτού



Σχήμα 1.14: Αρχιτεκτονική μιας τυπικής εφαρμογής που θα αναπτύξουμε στο πλαίσιο του βιβλίου.

Για την περιγραφή των επόμενων κεφαλαίων θα χρησιμοποιήσουμε την παραπάνω εικόνα της αρχιτεκτονικής μιας διαδικτυακής εφαρμογής. Μιας ιστοσελίδας που θα αναπτύξουμε.

Η εφαρμογή μας θα έχει δύο μέρη: ένα μέρος που θα κατεβαίνει στον φυλλομετρητή (στον πελάτη), όταν ο χρήστης καλέσει την ιστοσελίδα μας, και ένα μέρος που θα εκτελείται στον εξυπηρετητή, για να αποκριθεί στο αίτημα. Η ιστοσελίδα που θα σταλεί στον φυλλομετρητή θα αποτελείται από ένα συνδυασμό αρχείων HTML, CSS και JavaScript. Στον εξυπηρετητή θα τρέχει μια εφαρμογή JavaScript (στο περιβάλλον Node.js) που σε συνδυασμό με ένα Εξυπηρετητή ιστού ικανοποιεί τα αιτήματα του χρήστη της ιστοσελίδας.

Αρχικά θα δούμε στα κεφάλαια 2 και 3 πώς να περιγράψουμε τη δομή και το περιεχόμενο μιας ιστοσελίδας με τη γλώσσα HTML5.

Στην συνέχεια, στα κεφάλαια 4 και 5 θα δούμε πώς θα ορίσουμε την εμφάνιση του περιεχομένου που καθορίσαμε με την HTML, χρησιμοποιώντας τη γλώσσα CSS3.

Στο κεφάλαιο 6 θα δούμε πώς η παρουσίαση του περιεχομένου θα γίνει πιο απλά με χρήση μιας βιβλιοθήκης παρουσίασης περιεχομένου που είναι η Bootstrap.

Στα κεφάλαια 7-10 θα δούμε με αυξανόμενη λεπτομέρεια τη σύνταξη και χρήση της γλώσσας προγραμματισμού JavaScript, την οποία θα χρησιμοποιήσουμε στην αρχή σε συνδυασμό με τις τεχνολογίες που είδαμε στα

1 Εισαγωγή

πρώτα 6 κεφάλαια για να δώσουμε μεγαλύτερη διαδραστικότητα στην ιστοσελίδα μας όταν αυτή κατέβει στον φυλλομετρητή.

Στη συνέχεια στο κεφάλαιο 11 θα δούμε πώς θα χρησιμοποιήσουμε την JavaScript για να δημιουργήσουμε μια γραφική διεπαφή στο χώρο δύο διαστάσεων που ορίζει το στοιχείο canvas.

Από το κεφάλαιο 12 και εξής θα μελετήσουμε τις τεχνολογίες που απαιτούνται για να εξυπηρετήσουμε τα αιτήματα από τον φυλλομετρητή. Θα εισαχθούμε στο περιβάλλον Node.js που είναι το περιβάλλον εκτέλεσης της JavaScript έξω από τον φυλλομετρητή. Στη συνέχεια θα δούμε το πλαίσιο Express.js (Κεφάλαιο 13) που μάς επιτρέπει να δρομολογήσουμε αιτήματα σε ένα εξυπηρετητή και σε συνδυασμό με τη μηχανή template handlebars να ετοιμάσουμε το περιεχόμενο των ιστοσελίδων που θα στείλουμε στον πελάτη. Στο κεφάλαιο 14 θα δούμε πώς ο εξυπηρετητής μας μπορεί να επικοινωνήσει με μια βάση δεδομένων για μόνιμη αποθήκευση δεδομένων της εφαρμογής μας και ανάκτηση δεδομένων που ζητάει ο χρήστης. Τέλος στο κεφάλαιο 15 θα δούμε πώς θα διαχειριστούμε τη συνεδρία του χρήστη και θα ελέγξουμε την ταυτότητά του.

Κλείνοντας αυτή την επισκόπηση μπορούμε να παρατηρήσουμε ότι στο τέλος αυτής της διαδρομής θα έχουμε μια πλήρη εικόνα πώς δομείται και λειτουργεί μια σύγχρονη διαδικτυακή εφαρμογή, με κατανόηση θεμάτων εμφάνισης περιεχομένου, διαχείρισης της διάδρασης με τον χρήστη, ασφάλειας και αποθήκευσης δεδομένων, συνδυάζοντας σύγχρονες τεχνολογίες και έχοντας ένα ισχυρό υπόβαθρο για περαιτέρω ανάπτυξη δεξιοτήτων και γνώσεων της λειτουργίας του διαδικτύου και των εφαρμογών του.

1.6 Άσκηση

Ακολουθεί ένα πρόγραμμα Python που μάς επιτρέπει να εισάγουμε μια διεύθυνση URL και μάς επιστρέφει την κεφαλίδα της απόκρισης HTTP που λαμβάνουμε. Χρησιμοποιεί τη βιβλιοθήκη requests.

```
# πρόγραμμα που ζητάει ένα url και επιστρέφει την κεφαλίδα της απόκρισης HTTP
import requests
```

```
while True:
    url = input('url:')
    if url == '': break
    if not url.startswith('http'):
        url = 'http://' + url
    try:
        with requests.get(url) as response:
            print("\nRESPONSE STATUS: ", response.status_code)
            print("RESPONSE HEADER")
            for key, value in response.headers.items():
                print("{:30s} {}".format(key, value))
    except:
        print('error opening', url)
```

1 Εισαγωγή

ο κώδικας Ένα παράδειγμα χρήσης:

```
url:http://hci.ece.upatras.gr/
```

```
RESPONSE STATUS: 200
```

```
RESPONSE HEADER
```

Date	Sat, 27 Mar 2021 12:30:29 GMT
Server	Apache/2.4.18 (Ubuntu)
Set-Cookie	language=en; expires=Sun, 27-Mar-2022 12:30:29 GMT; Max-Age=31536000; path=
Link	<https://hci.ece.upatras.gr/wp-json/>; rel="https://api.w.org/"
Vary	Accept-Encoding
Content-Encoding	gzip
Content-Length	9848
Keep-Alive	timeout=5, max=100
Connection	Keep-Alive
Content-Type	text/html; charset=UTF-8

Πειραματιστείτε με το πρόγραμμα αυτό στέλνοντας μηνύματα σε διάφορες διευθύνσεις του διαδικτύου, δέστε προσεκτικά την κεφαλίδα της απόκρισης που πήρατε, μπορείτε να βγάλετε συμπεράσματα για το τι τύπου είναι ο εξυπηρετητής που σάς απάντησε (είναι η κεφαλίδα Server), τι τύπου είναι η απόκριση, (Content-Type), αν επισυνάπτονται αρχεία cookies και τι διάρκεια ζωής έχουν (Set-Cookie), κλπ.)

Επίσης σκεφτείτε με ποιον τρόπο θα μπορούσατε να χρονομετρήσετε το συνολικό χρόνο από την αποστολή του αιτήματος μέχρι την παραλαβή της απόκρισης (χρήση της `time.process_time()`)

1.7 Σύνοψη

Στο πρώτο αυτό εισαγωγικό κεφάλαιο εστίασαμε στα κύρια χαρακτηριστικά του διαδικτύου με ιδιαίτερη έμφαση στον παγκόσμιο ιστό, και το πρωτόκολλο επικοινωνίας υπολογιστών που τον αφορά το HTTP. Με βάση τη γενική αυτή εικόνα δόθηκε μια συνοπτική περιγραφή των επόμενων κεφαλαίων που θα μάς βοηθήσουν να αναπτύξουμε μια σύνθετη διαδικτυακή εφαρμογή. Αρχή και βασική τεχνολογία κάθε διαδικτυακής εφαρμογής είναι είναι η γλώσσα σήμανσης υπερκειμένου (Hyper Text Markup Language, HTML) που αποτελεί αντικείμενο του επόμενου κεφαλαίου.