



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS



ECE_ΓΚ802 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΔΙΑΔΙΚΤΥΟΥ - 8^ο εξάμηνο

HCI202 Introduction to Web Technologies (Master in HCI)

Θεωρία: Ν. Αβούρης, Χ. Σιντόρης, Χ. Φείδας
Εργαστήριο: Χ. Σιντόρης

Πρόγραμμα – (2021-2022)

- **Θεωρία**
 - 3 ώρες: Παρασκευή 11-14 μμ / ΗΛ8
- **Εργαστήριο** (για το ΓΚ802)
 - 2 ώρες: Δευτέρα 10-12 / ΚΥΠΕΣ-II

Τρόπος αξιολόγησης

(ΓΚ802)

- Εργαστήριο 40% (ασκήσεις 20% + 20% τελική εξέταση),
- Ομαδική εργασία –πρότζεκτ 30%,
- Τελική εξέταση θεωρίας 30%

(HCI202)

- Ατομική εργασία-πρότζεκτ 50%,
- Άσκησεις 20%
- Τελική εξέταση θεωρίας (30%)

μαθησιακά αποτελέσματα	<p>Με την ολοκλήρωση αυτού του μαθήματος ο φοιτητής θα πρέπει να γνωρίζει τις βασικές αρχές της αρχιτεκτονικής του διαδικτύου και τις τεχνολογίες ανάπτυξης σύγχρονων διαδικτυακών εφαρμογών, καθώς και να είναι σε θέση να επιλέξει τις κατάλληλες τεχνολογίες για τη σχεδίαση διαδικτυακών εφαρμογών για επίλυση σύνθετων προβλημάτων.</p>
Δεξιότητες	<p>Με την ολοκλήρωση του μαθήματος ο φοιτητής αναμένεται να έχει αποκτήσει λειτουργική γνώση τεχνολογιών HTML, CSS, JavaScript, και προγραμματισμού στην πλευρά του εξυπηρετητή. Με βάση τις δεξιότητες αυτές να είναι σε θέση να σχεδιάσει, να αναπτύξει, και να ελέγχει την ορθότητα λειτουργίας μιας διαδικτυακής εφαρμογής που υλοποιείται με τις τεχνολογίες αυτές.</p>
Προαπαιτήσεις	<p>Γλώσσες Προγραμματισμού (πχ Java, Python), Εισαγωγή στα Δίκτυα Υπολογιστών.</p>
Περιεχόμενα (ύλη) του μαθήματος	<p>Σκοπός του μαθήματος είναι η μελέτη της αρχιτεκτονικής και δομής του Διαδικτύου, των βασικών εργαλείων προγραμματισμού εφαρμογών διαδικτύου, τόσο στην πλευρά του χρήστη όσο και του εξυπηρετητή.</p> <ol style="list-style-type: none"> Εισαγωγή στο διαδίκτυο, αρχιτεκτονική, πρωτόκολλα {1 εβδ.] Προγραμματισμός στην πλευρά του πελάτη: (HTML5) - Stylesheets (CSS3) - Bootstrap [2 εβδομάδες] JavaScript [3 εβδομάδες] Προγραμματισμός στην πλευρά του εξυπηρετητή με javascript (node.js, express.js) [4 εβδομάδες]

πρόσθετο διαδικτυακό υλικό

Ακόμη συστήνεται η εγγραφή και παρακολούθηση διαδικτυακών μαθημάτων στην πλατφόρμα mathesis.cup.gr:

- Εισαγωγή στην ανάπτυξη ιστοσελίδων
 - εισαγωγή στην HTML, CSS, JavaScript
- Προχωρημένα θέματα ανάπτυξης ιστοσελίδων
 - φόρμες, grid, flex, bootstrap, canvas, fetch, JS objects

Εγχειρίδιο μαθήματος

Εγχειρίδιο:

<https://eclass.upatras.gr/modules/document/file.php/EE767/ΓΚ802-Εγχειρίδιο-2021-2022.pdf>



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας
Υπολογιστών

Μάθημα: ECE_ΓΚ802
Προγραμματισμός Διαδικτύου

Εγχειρίδιο Μαθήματος

... είπαν για το μάθημα

Θα θέλα να πω πως αν και άριστα οργανωμένα το μάθημα οι περισσότεροι συμφοιτητές μου(μαζί και γω) συναντήσαμε δυσκολία στην αφομοίωση του server side και ειδικά του node και express που είχαν πολλές νέες μεθόδους σε συνδυασμό με εγγενείς δυσκολίες πχ ασύγχρονο κτλ .Πιστεύω θα έπρεπε να γίνει μια ανακατανομή του χρόνου του μαθήματος και να αφιερώνονται περισσότερες διδακτικές ώρες σε αυτό το κομμάτι σε σχέση την πιο εύκολα κατανοητή html.

Το μάθημα έχει μεγάλη ύλη και έχει πολλές νέες τεχνολογίες προς τον φοιτητή. Θεωρώ ότι το παιχνίδι ερωτήσεων δεν χρειάζεται γιατί χάνεται χρόνος που μπορούσαν να εξηγηθούν κάποιες λεπτομέρειες. Επίσης , θα μπορούσε να μειωθεί ο αριθμός των εργαστηρίων ειδικά , στο front end. Κατά τα άλλα , θεωρώ ότι είναι από τα πιο οργανωμένα μαθήματα της σχολής και ένα μάθημα που ανοίγει διεξόδους για την μετέπειτα πορεία του φοιτητή.

Τεχνολογίες μαθήματος

Θα αναπτύξουμε εφαρμογές διαδικτύου με τρεις σχετιζόμενες τεχνολογίες: HTML5, CSS3 και JavaScript.

HTML



JS

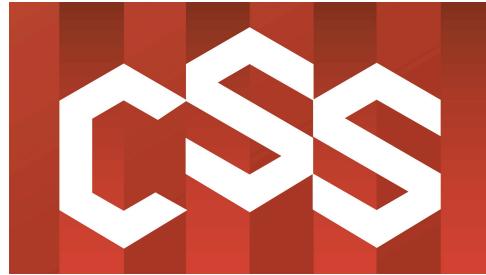
html5



Η HTML5 είναι γλώσσα επισημείωσης (markup language) που ορίζει τη δομή **και αντικείμενα** μιας ιστοσελίδας: παραγράφους κειμένου, κεφαλίδες, εικόνες, κλπ.

CSS3

Η CSS3 είναι μια γλώσσα μορφοποίησης που ορίζει τον τρόπο παρουσίασης του περιεχομένου της ιστοσελίδας, δηλαδή το στυλ παρουσίασης των στοιχείων της (χρώματα, γραμματοσειρές, θέσεις στοιχείων, κλπ.)

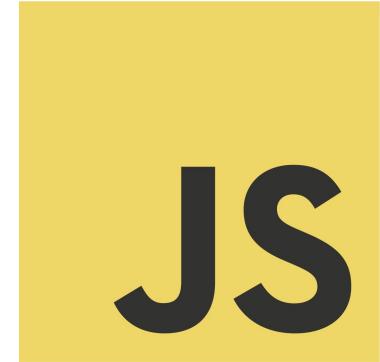


Επίσης θα δούμε το CSS/HTML framework **Bootstrap**



JavaScript

Η JavaScript ή 'JS', είναι μια γλώσσα προγραμματισμού που εκτελείται στον browser και επιτρέπει διάδραση με τον χρήστη, μετατρέποντας τη σελίδα σε ζωντανή εφαρμογή.



Στον εξυπηρετητή θα χρησιμοποιήσουμε επίσης JavaScript (το περιβάλλον εκτέλεσης JS node.js)



Backend framework

Πλαίσιο ανάπτυξης εφαρμογών
διαδικτύου στο περιβάλλον node.js:
express.js



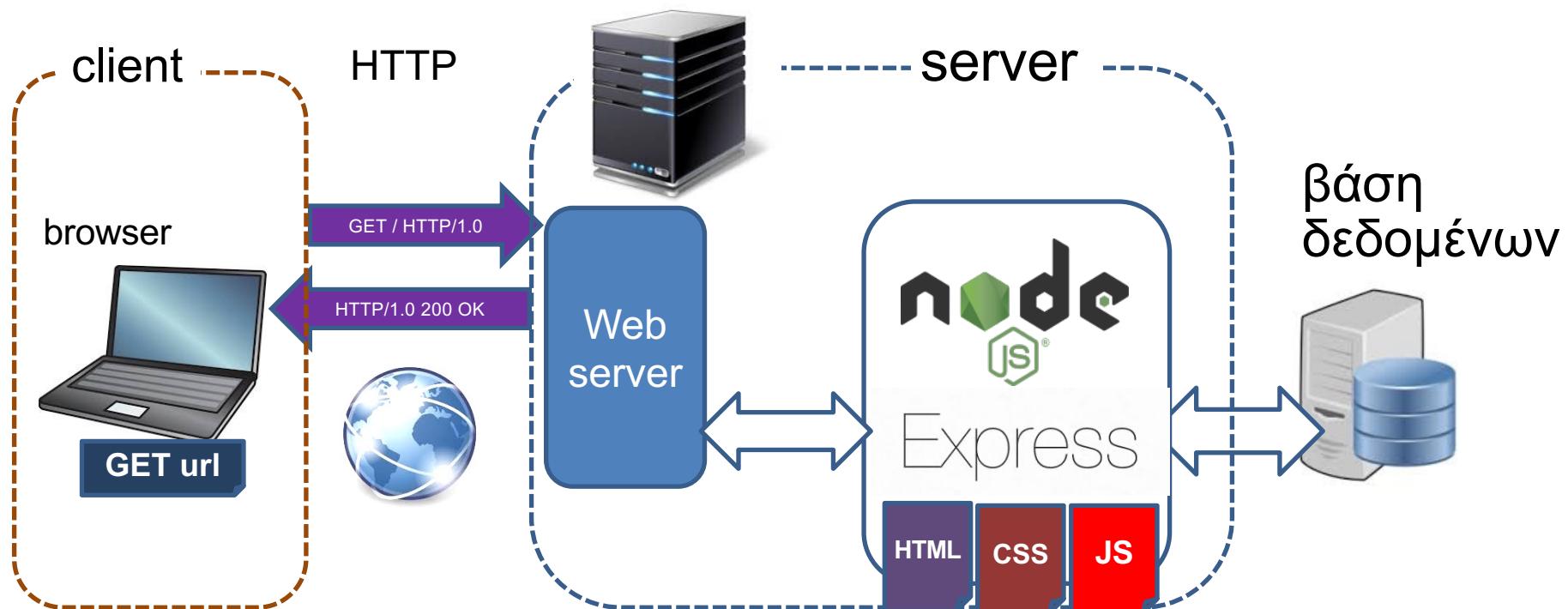
Με πρόσθετα:
express-session.js
express-handlebars.js

handlebars



Express

αρχιτεκτονική των τεχνολογιών του μαθήματος



Full stack web development

HTML

the Client

CSS

(user)

Bootstrap

JavaScript

<HTTP> - web server -----

Node.js

the Server

Express.js

Handlebars

Session.js

MySQL/Mongo

the Data

Εναλλακτικές επιλογές

HTML

CSS

Bootstrap

JavaScript

<HTTP>

django

 **spring**[®]

MySQL/Postgress/Mongo ...



Front-end
frameworks



Back-end
frameworks
(Python,
Java, PHP)

Εισαγωγή: Ιστορία - το
διαδίκτυο και ο
παγκόσμιος ιστός

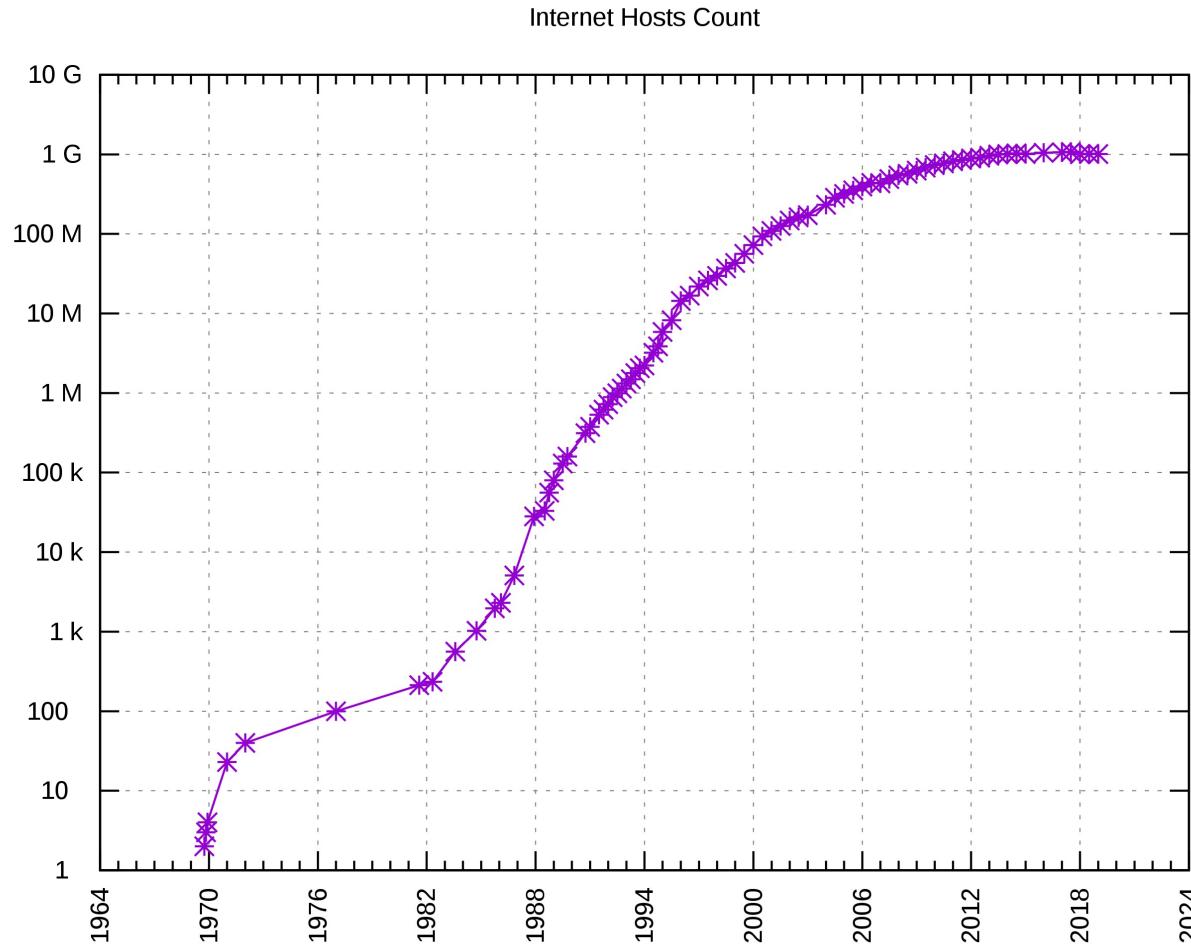
Ιστορία

- 60's: πρώτο Packet Switching Network δίκτυο ARPANET
- 72 Internet Mail
- 74 TCP/IP (V. Cerf, B. Kahn)
- 84 Το Internet ξεπερνά τους **1.000 Hosts**
- 84 Εισαγωγή του DNS
- 87 Το Internet ξεπερνά τους **10.000 Hosts**
- 92 Το Internet ξεπερνά τους **1 εκατομμύριο Hosts**
- 92 Ο **ιστός (Web)** σχεδιάστηκε από τον **Tim Berners-Lee** (CERN) HTML, URI, HTTP
- 93 Ο φυλλομετρητής MOSAIC (M. Andreesen)
- 94 Πρώτη έκδοση της CSS (H.K. Lie)
- 95 Πρώτη έκδοση της JavaScript (B. Eich)
- 08 HTML – έκδοση 5
- 09 node.js (R.Dhal)
- 10 express.js (TJ Holowaychuk)

Ιστορία: Μερικά Ονόματα ...

- Joseph C.R. **Licklider** (ARPA/IPTO η ιδέα του δικτύου ARPANET),
- Bob **Taylor** (δημιουργός του ARPANET)
- Vint **Cerf**, Bob **Kahn** (ορίζουν τα TCP/IP)
- Tim **Berners-Lee** (παγκόσμιος ιστός)
- Marc **Andreessen** (γραφικός φυλλομετρητής MOSAIC, NETSCAPE)
- Brendan **Eich** (JavaScript, Mozilla Project)

Εκθετική αύξηση αριθμού hosts



Το πρώτο Web Site

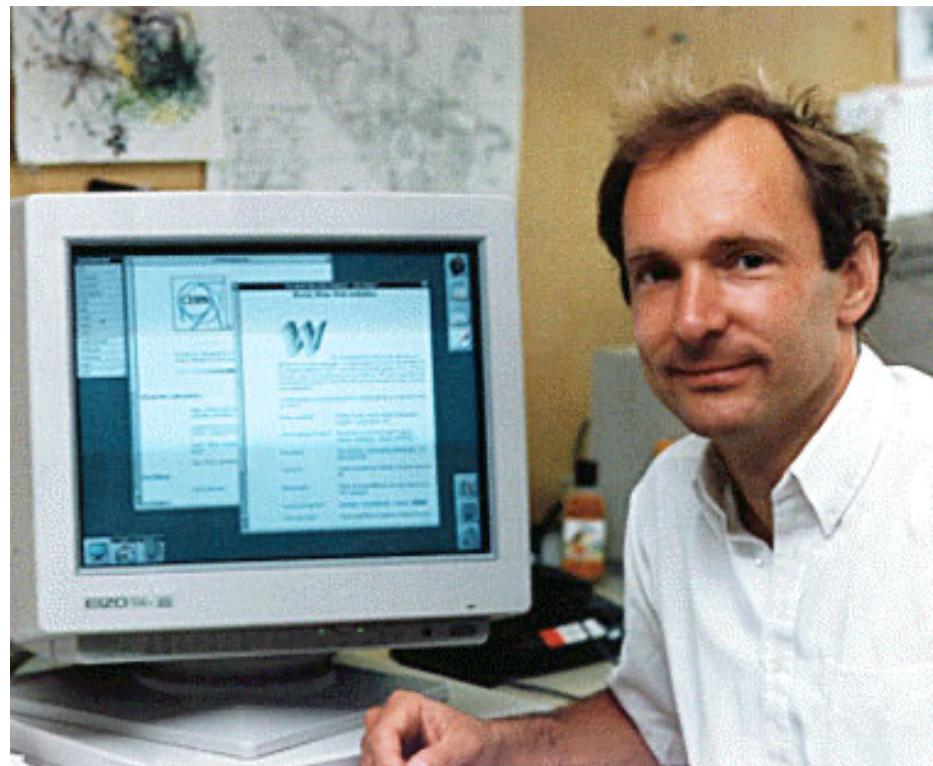
The screenshot displays a window titled "CERN Entry Point" which contains several links:

- [Help](#)
- [Phone book](#)
- [Computer center](#)
- [News](#)
- [See also](#)

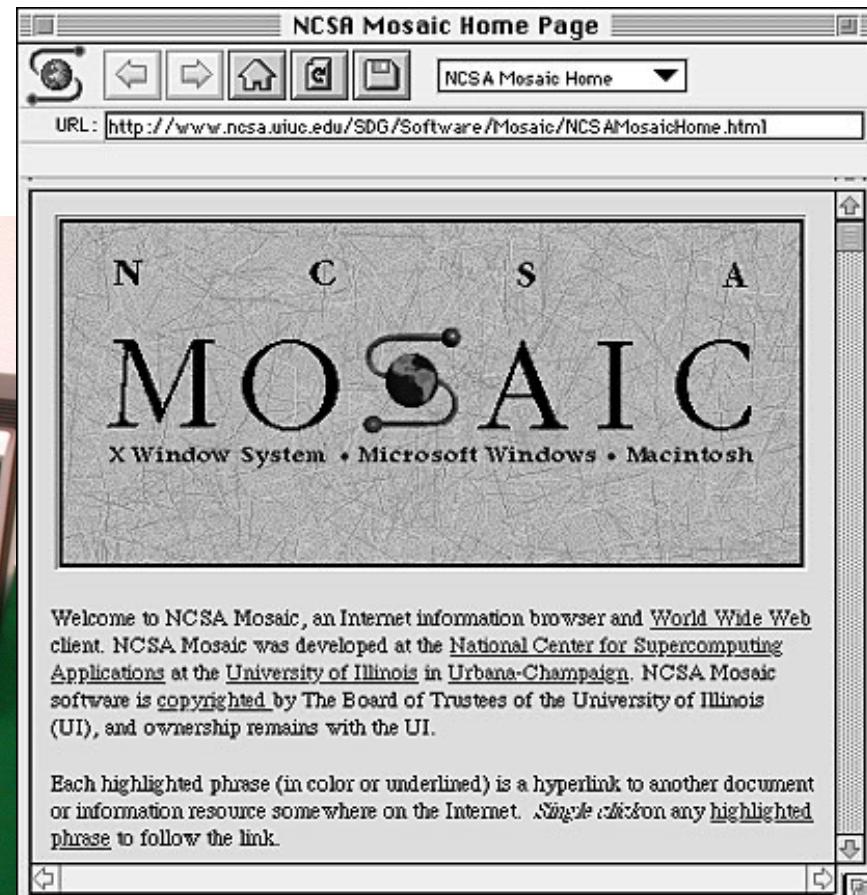
Under the "See also" section, there is a link to "High-energy SUBJECT". A red arrow points from this link to a separate window titled "High-Energy Physics Information" which has the heading "High Energy Physics".

Below the "High-Energy Physics Information" window, another red arrow points to a window titled "SLAC SPIRES". This window shows a search results list for "find author Kellner".

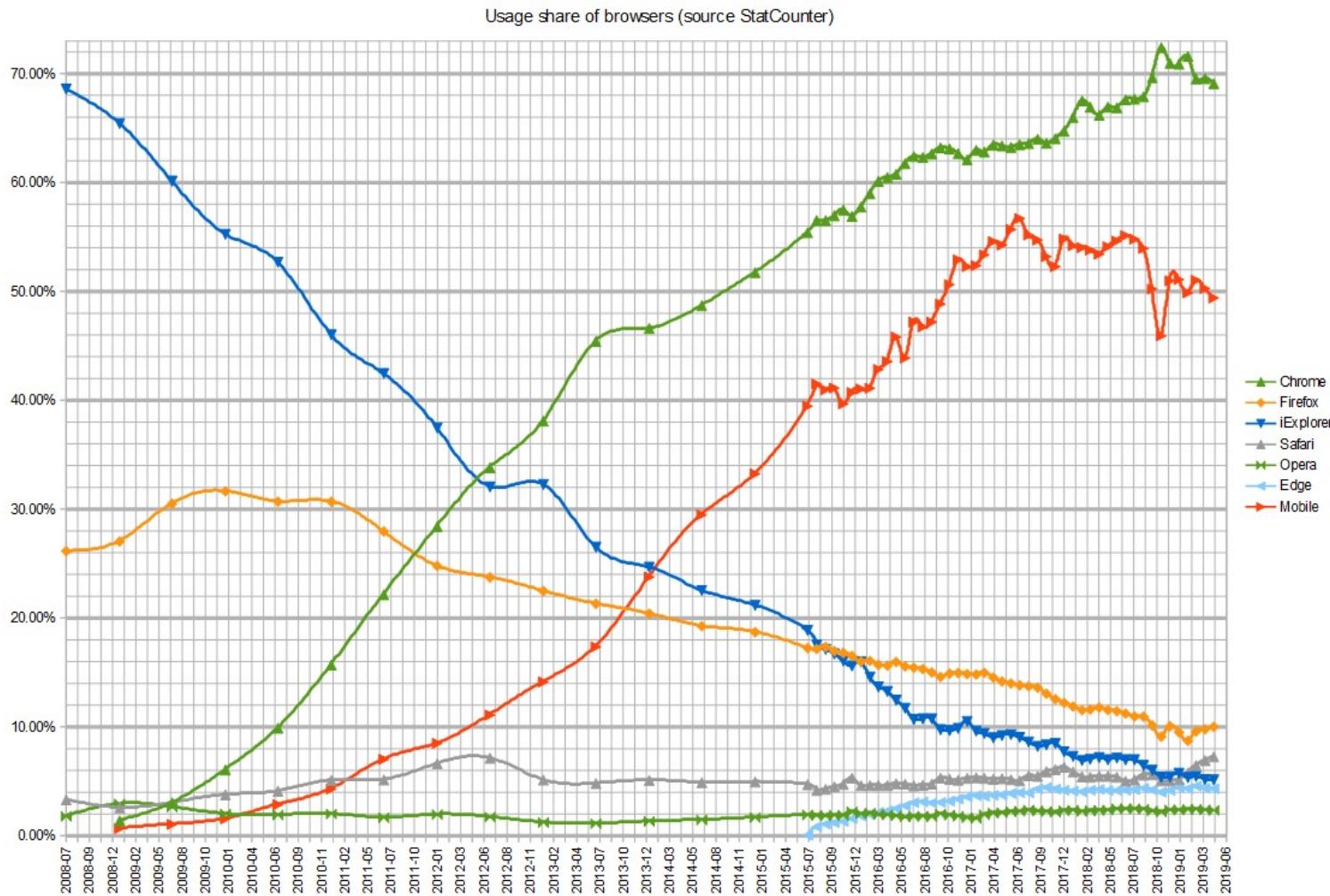
Rank	Author	Title
28	G. Kellner	DEVELOPMENT OF SOFTWARE FOR ALEPH USING STRUCTURED TECHNIQUES. COMPUT. PHYS. COMMUN. 45 (1987) 229-243.
29	M. Arneodo, et al.	SHADOWING IN DEEP INELASTIC MUON SCATTERING FROM NUCLEAR TARGETS. Phys.Lett. 211B:493,1988.
30	M. Arneodo, et al.	CHARGE STRUCTURE OF THE HADRONIC FINAL STATE IN DEEP INELASTIC MUON - NUCLEON SCATTERING. Z.Phys.C48:347,1988.
31	B. Carpenter, et al.	THE MUSCLE REPORT: THE COMPUTING NEEDS OF THE LEP EXPERIMENTS.



Πρώτος Web Server και ο πρώτος Browser



The browsers' wars



Usage share of web browsers according to StatCounter till 2019-05

Ερωτήσεις

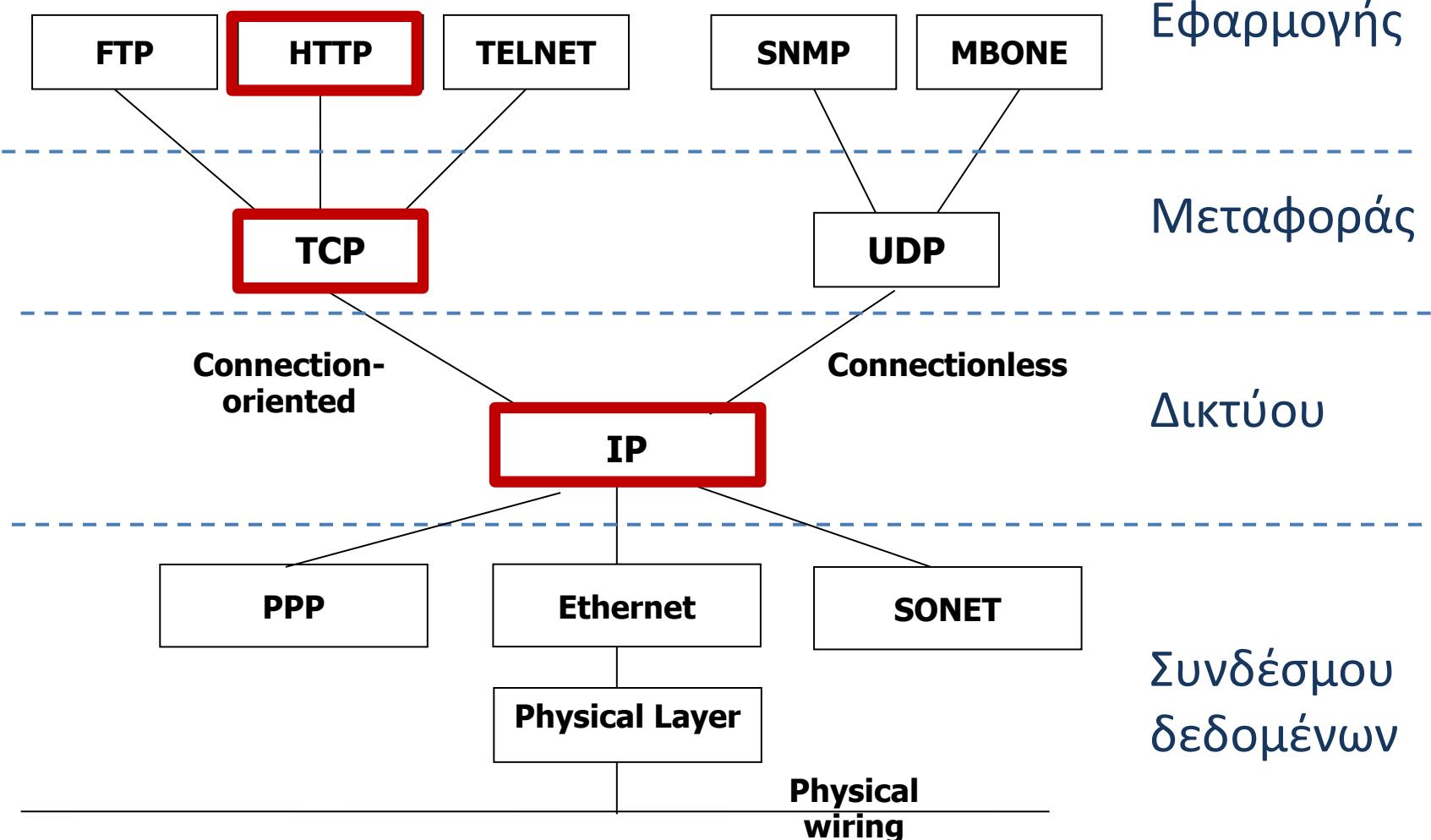
- Τι είναι τα RFC, ποιο το RFC για το email, FTP
- Ανακαλύψτε περισσότερες λεπτομέρειες για τη ζωή των εφευρετών του διαδικτύου (π.χ. πηγή K. Hafner, M. Lyon, 96)
- Πώς μετράμε τον αριθμό hosts (πηγή ;)
- Ποια η σχέση παγκόσμιου ιστού και διαδικτύου;

ασκήσεις

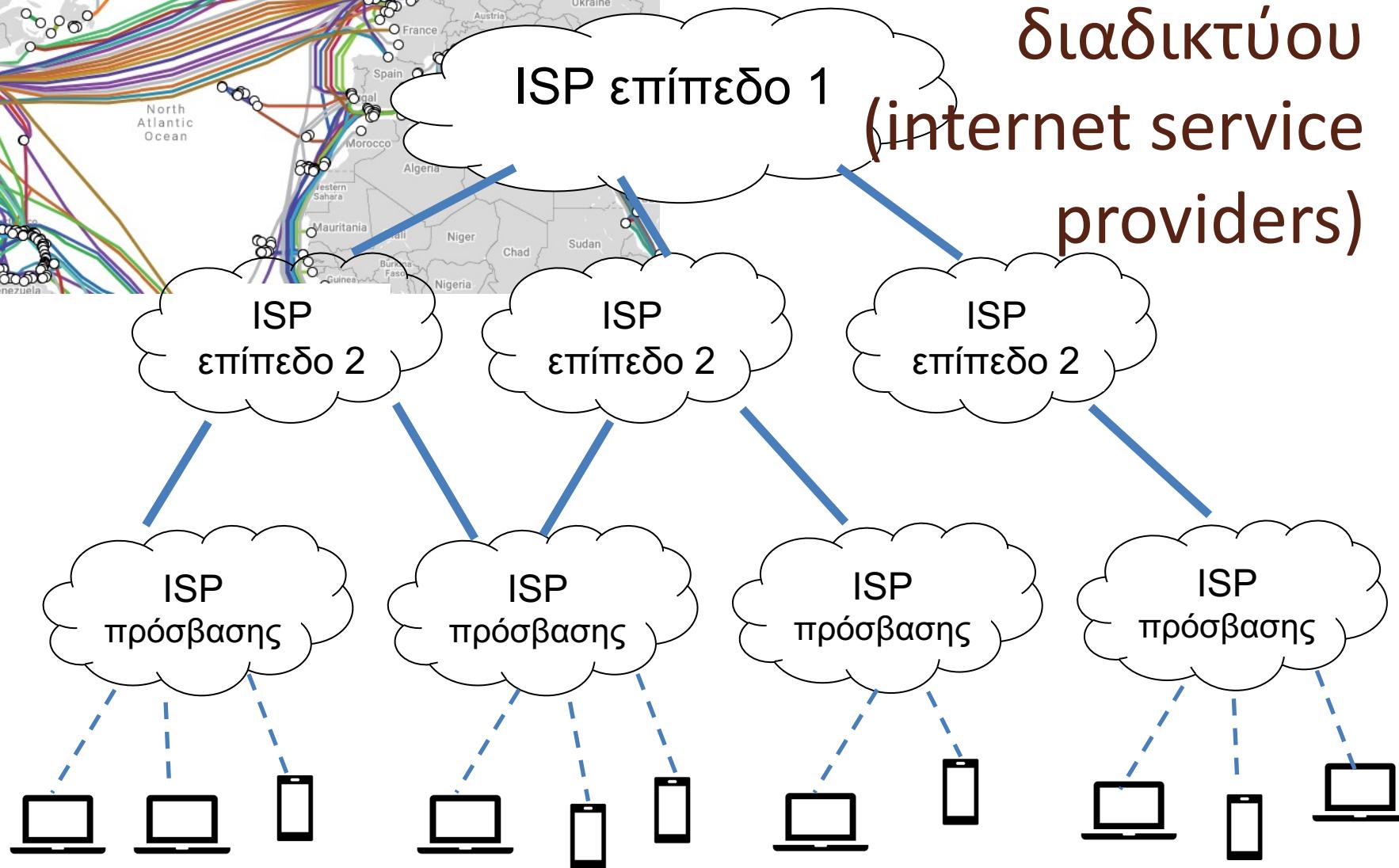
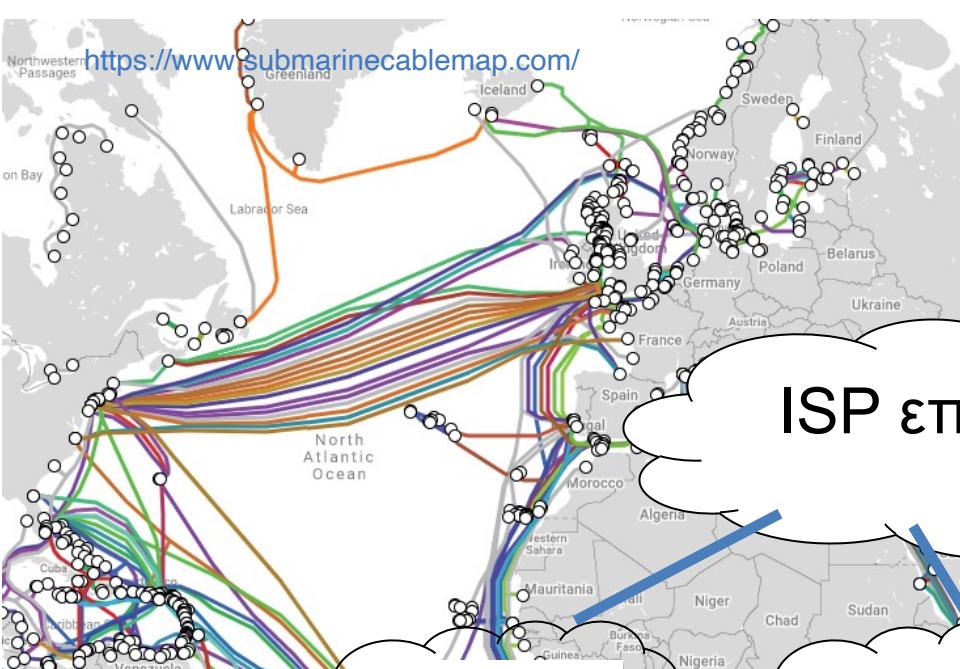
- Να βρείτε το μέγεθος του διαδικτύου (εξηγήστε τις μονάδες μέτρησης, την πηγή σας και τον Τρόπο μέτρησης)
- Μελετήστε το άρθρο *Web Science an Interdisciplinary Approach* και απαντήστε στα εξής ερωτήματα
 - B.1 Ποιοι κατά τη γνώμη σας είναι οι κυριότεροι λόγοι που ειναι δυνατή η χρήση του όρου web science
 - B.2 Ποια τα χαρακτηριστικά της επιστημονικής αυτής περιοχής
 - B.3 Πώς σχετίζεται με άλλες επιστημονικές περιοχές

Το διαδίκτυο:
αρχιτεκτονική και
πρωτόκολλα

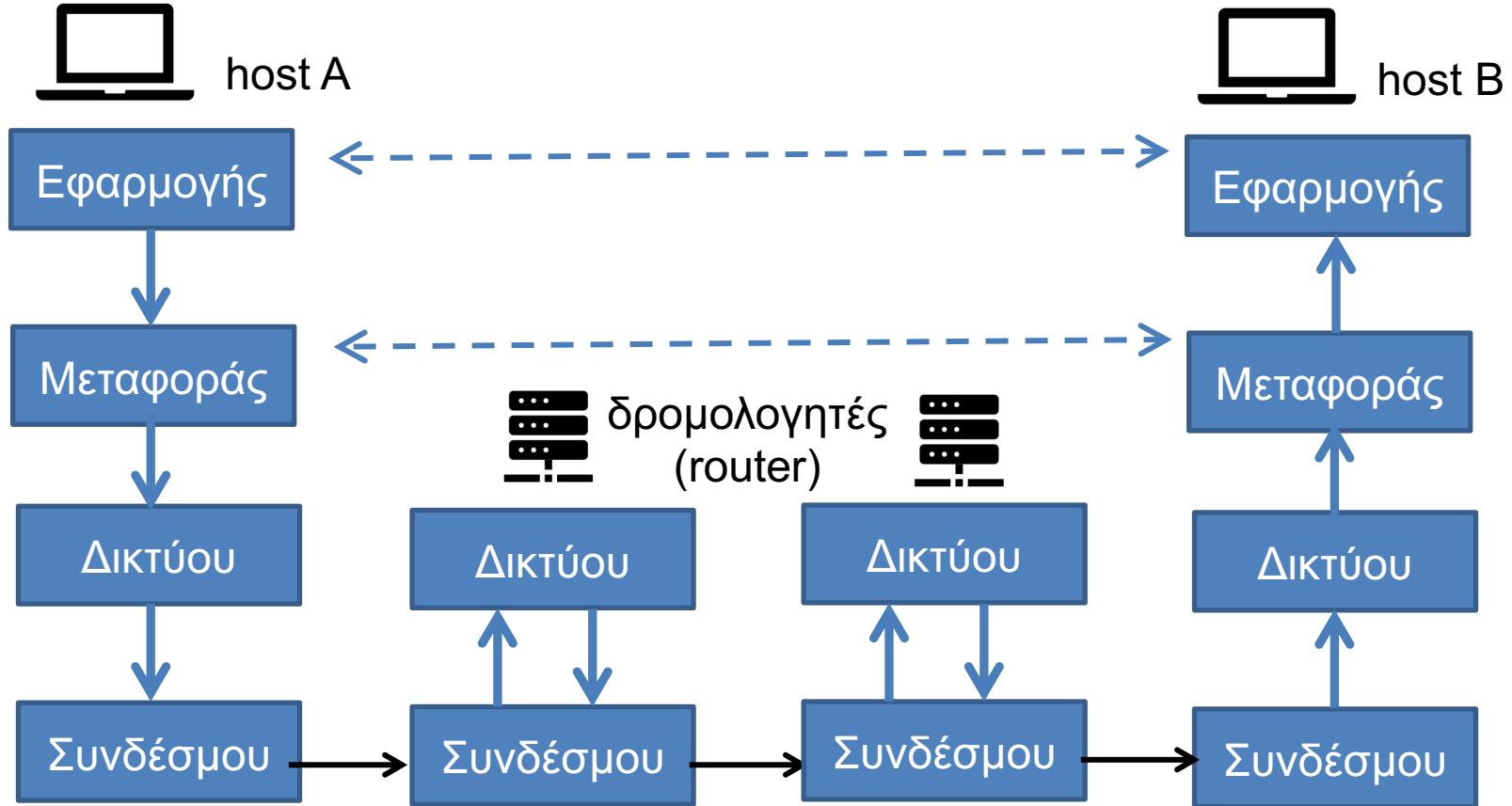
Πρωτόκολλα διαδικτύου



Αρχιτεκτονική δικτύων παρόχων διαδικτύου (internet service providers)



Δρομολόγηση δεδομένων στο διαδίκτυο



Μετάδοση δεδομένων στο διαδίκτυο

Internet Layers

Application Layer
(FTP, HTTP, SMTP)

DATA

Transport Layer
(TCP, UDP)

TCP Header DATA

Network Layer
(IP, ICMP)

IP Header TCP Header DATA

Network Access Layer
(Ethernet, FDDI, Token Ring)

Ethernet
Header

IP Header

TCP Header

DATA

Ethernet
trailer

IP datagram header

bit offset	0–3	4–7	8–13	14–15	16–18	19–31		
0	Version	Header Length	Differentiated Services Code Point	Explicit Congestion Notification		Total Length		
32	Identification			Flags	Fragment Offset			
64	Time to Live	Protocol			Header Checksum			
96	Source IP Address							
128	Destination IP Address							
160	Options (if Header Length > 5)							
160 or 192+	Data							

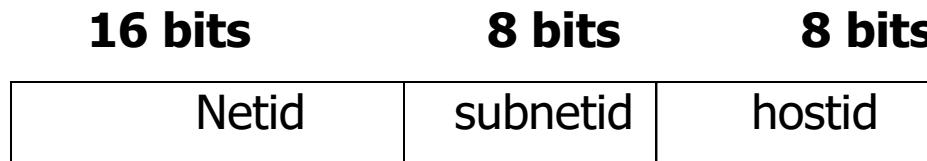
Διευθύνσεις IP (v.4)

- Μοναδικές διευθύνσεις κάθε κόμβου του διαδικτύου (**32 bit**)
- Δίνονται από το InterNIC
- Απαρτίζονται από 4 τμήματα των 8 bit (τιμές 0-255) που ορίζουν ιεραρχικά: κατηγορία δικτύου, διεύθυνση δικτύου, μηχανή
- Υπάρχουν 3 κατηγορίες IP δικτύων (A,B,C)

4.294.967.296 (2^{32}) πιθανές μοναδικές διευθύνσεις

Subnet Masks

- Η μάσκα υποδικτύου είναι μηχανισμός μέσω του οποίου μπορούμε να καθορίσουμε 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.



Subnet Mask: 1111111111111111 11111111 00000000
= 255.255.255.0

IPv6

- Η τρέχουσα έκδοση του πρωτοκόλλου IP (έκδοση 4) παρουσιάζει προβλήματα, με κύριο αυτό του μικρού αριθμού δυνατών διευθύνσεων
- Νέα έκδοση IPv6 έχει μήκος διευθύνσεων **128 bits**
- Γρηγορότερη επεξεργασία επικεφαλίδων που έχουν λιγότερα πεδία
- Μηχανισμοί πιστοποίησης και ασφαλείας
- Μηχανισμοί χαρακτηρισμού πακέτων για εφαρμογές πραγματικού χρόνου.

Άσκηση

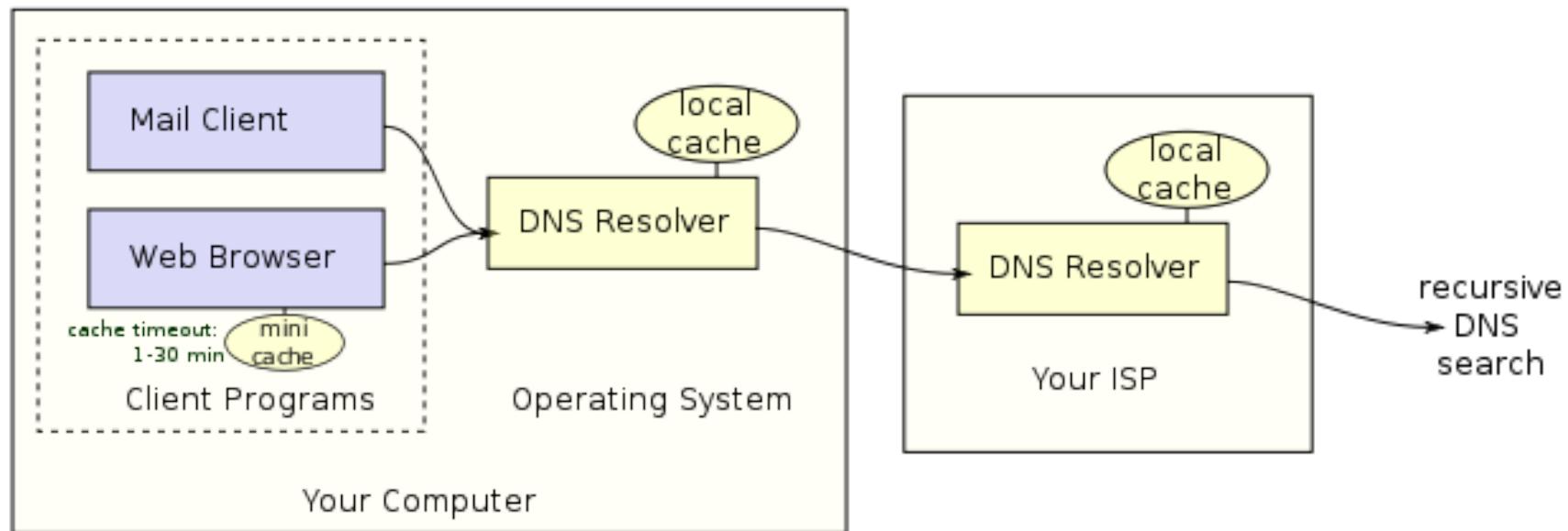
- Έστω ότι η διεύθυνση μας είναι 140.252.1.1 και ότι η μάσκα υποδικτύου είναι 255.255.255.0. Να βρείτε τη σχέση μας με τους εξής υπολογιστές:
 - (α) 140.252.4.5
 - (β) 140.252.1.22
 - (γ) 192.43.235.6

DNS Domain Name System

Σύστημα ονοματοδοσίας διαδικτύου

- Μηχανισμός μετασχηματισμού των διευθύνσεων IP σε ονόματα υπολογιστών
- Από **150.140.130.190** σε **www.upatras.gr**
>ping www.upatras.gr
>nsllookup 150.140.130.190
- Στις διευθύνσεις το πιο ειδικό τμήμα της διεύθυνσης βρίσκεται στο αριστερό άκρο
- Μηχανισμοί δρομολόγησης (δυναμική ή στατική), πίνακες δρομολόγησης - Πρωτόκολλο ανίχνευσης σφαλμάτων (Internet Control Message Protocol, ICMP)

Επίλυση αιτημάτων DNS στον πελάτη



TCP (Transmission Control Protocol

Πρωτόκολλο Ελέγχου Μεταφοράς)

- Οι κύριοι στόχοι του πρωτοκόλλου TCP είναι να επιβεβαιώνεται η αξιόπιστη αποστολή και λήψη δεδομένων, να μεταφέρονται τα δεδομένα χωρίς λάθη και με σωστή σειρά μεταξύ του στρώματος δικτύου (network layer) και του στρώματος εφαρμογής (application layer)
- Οι περισσότερες σύγχρονες υπηρεσίες στο Διαδίκτυο βασίζονται στο TCP. πχ το SMTP (port 25), το παλαιότερο (και μη-ασφαλές) Telnet (port 23), το FTP το HTTP (port 80), ο Παγκόσμιος Ιστός.

UDP & TCP

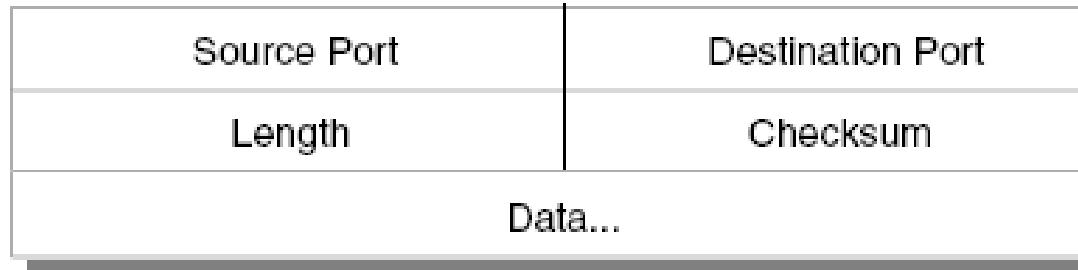


Figure 4-2 UDP: Datagram format

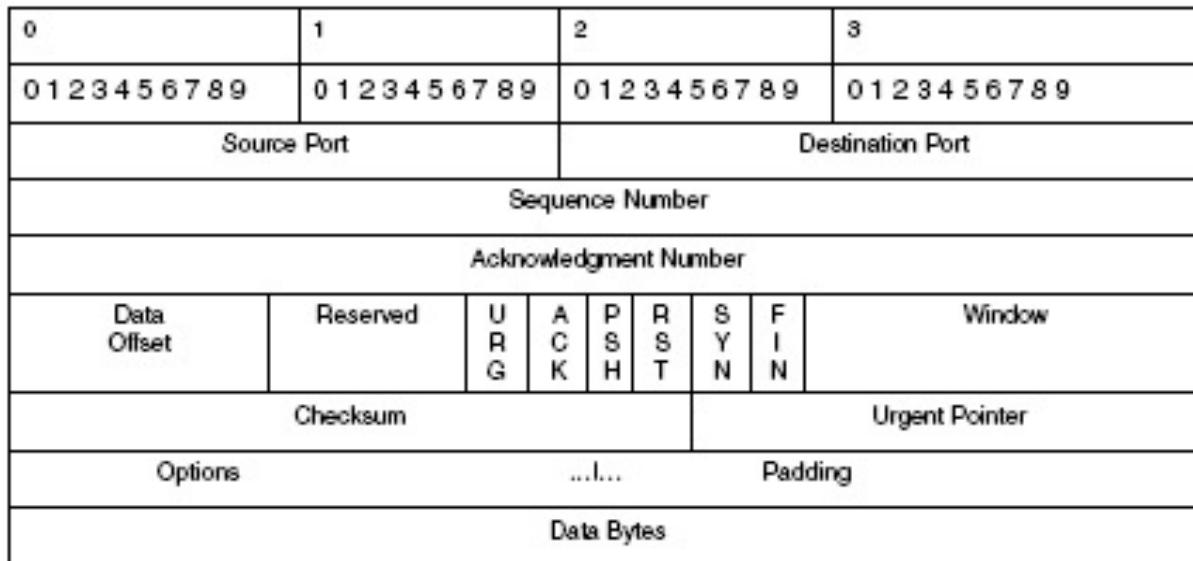


Figure 4-10 TCP: Segment format

TCP packet

TCP επικεφαλίδα

+	Bits 0 - 3	4 - 9	10 - 15	16 - 31
0		Source Port Θύρα Προέλευσης		Destination Port Θύρα Προορισμού
32			Sequence Number Αριθμός ακολουθίας	
64			Acknowledgment Number Αριθμός επιβεβαίωσης	
96	Data Offset	Reserved	Flags Σημαίες	Window Παράθυρο
128		Checksum Άθροισμα ελέγχου		Urgent Pointer Επείγοντα δεδομένα
160			Options Επιλογές (προαιρετικές)	
160/192+			Data Δεδομένα	

Λειτουργία αποστολής πακέτων TCP

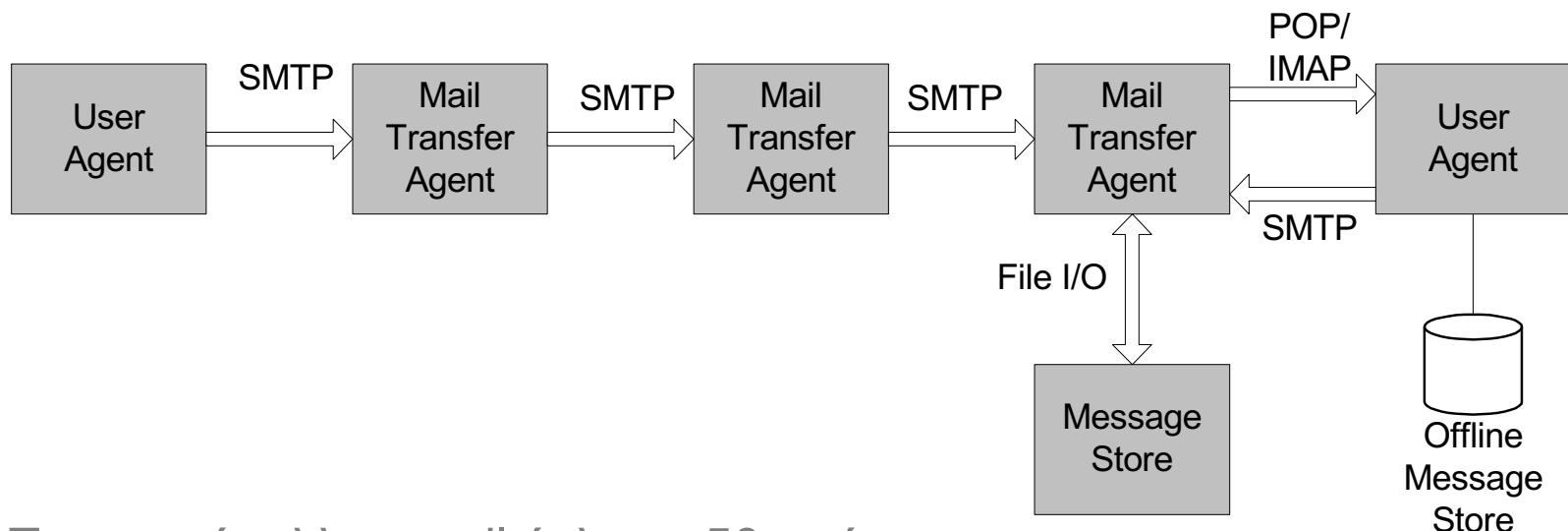
Το πρωτόκολλο ελέγχου μεταφοράς (TCP) είναι connection oriented, δηλαδή η μεταφορά δεδομένων γίνεται μέσω σύνδεσης, η οποία οριοθετείται από ένα σήμα έναρξης και ένα σήμα τέλους ή διακοπής.

Έναρξη - Τριμερής χειραψία / 3-way handshake

Πριν να προσπαθήσει ένα πρόγραμμα-πελάτης (client) να συνδεθεί με έναν εξυπηρετητή, ο server πρέπει πρώτα να δεσμεύσει μια θύρα και να την ανοίξει ώστε να δέχεται συνδέσεις: αυτό καλείται passive open. Όταν γίνει αυτό, ο client μπορεί να αρχίσει τη σύνδεση (active open). Για να γίνει μια σύνδεση, γίνεται μια "χειραψία" ανάμεσα στα συμμετέχοντα μέρη, το λεγόμενο **three-way handshake**.

email

- SMTP (rfc 821,822),
- MIME (rfc 2045, 2049),
- POP, IMAP

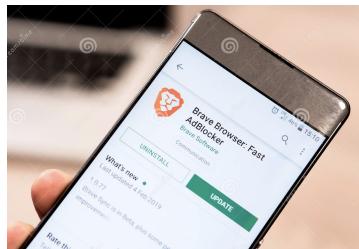


Το πρωτόκολλο email έκλεισε 50 χρόνια

ο παγκόσμιος ιστός

HTTP - URL

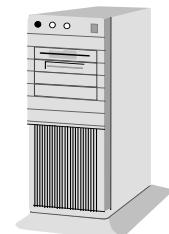
Ο Παγκόσμιος ιστός (WWW), 1992...



CLIENT
(φυλλο-
μετρητής)

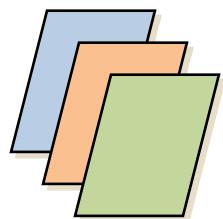


URL (ενιαίο πρωτόκολλο
διευθύνσεων πόρων)



WEB
SERVER

HTTP (πρωτόκολλο
μετάδοσης)



HTML/CSS/JS
(μορφοποίηση
περιεχομένου)

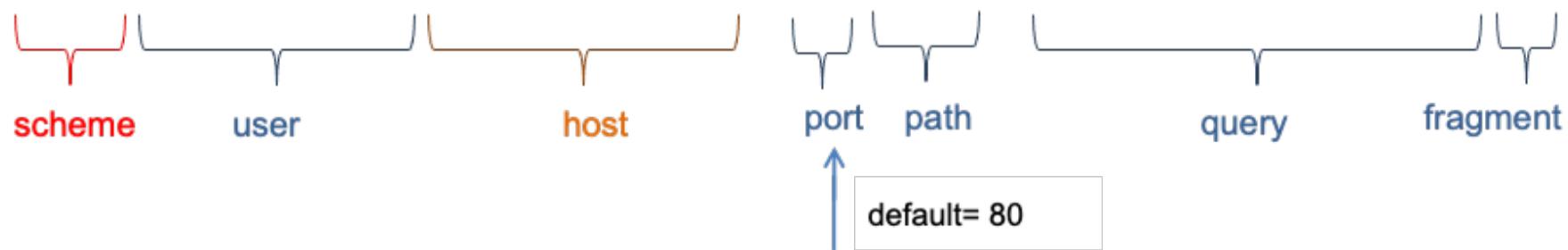
To web έκλεισε 30 χρόνια

Διευθύνσεις εντοπισμού πόρων URL

- Ένας **πόρος** είναι ένα τμήμα πληροφορίας που μπορεί να αναγνωριστεί μονοσήμαντα από ένα προσδιορισμό του, ονομαζόμενο **URL (Universal Resource Locator, Παγκόσμιος Εντοπιστής του Πόρου)** εναλλακτικός όρος URI / UR identifier
- Ένας πόρος μπορεί να είναι:
 - Ένα αρχείο (π.χ. έγγραφο υπερκειμένου HTML)
 - Εικόνες και πρόσθετα στοιχεία, κλπ.
- Μια ιστοσελίδα περιέχει ένα συνδυασμό πόρων

Δομή ενός URL

`http://nikos.don:pass@www.example.com:123/forum/?tag=network&order=new#top`

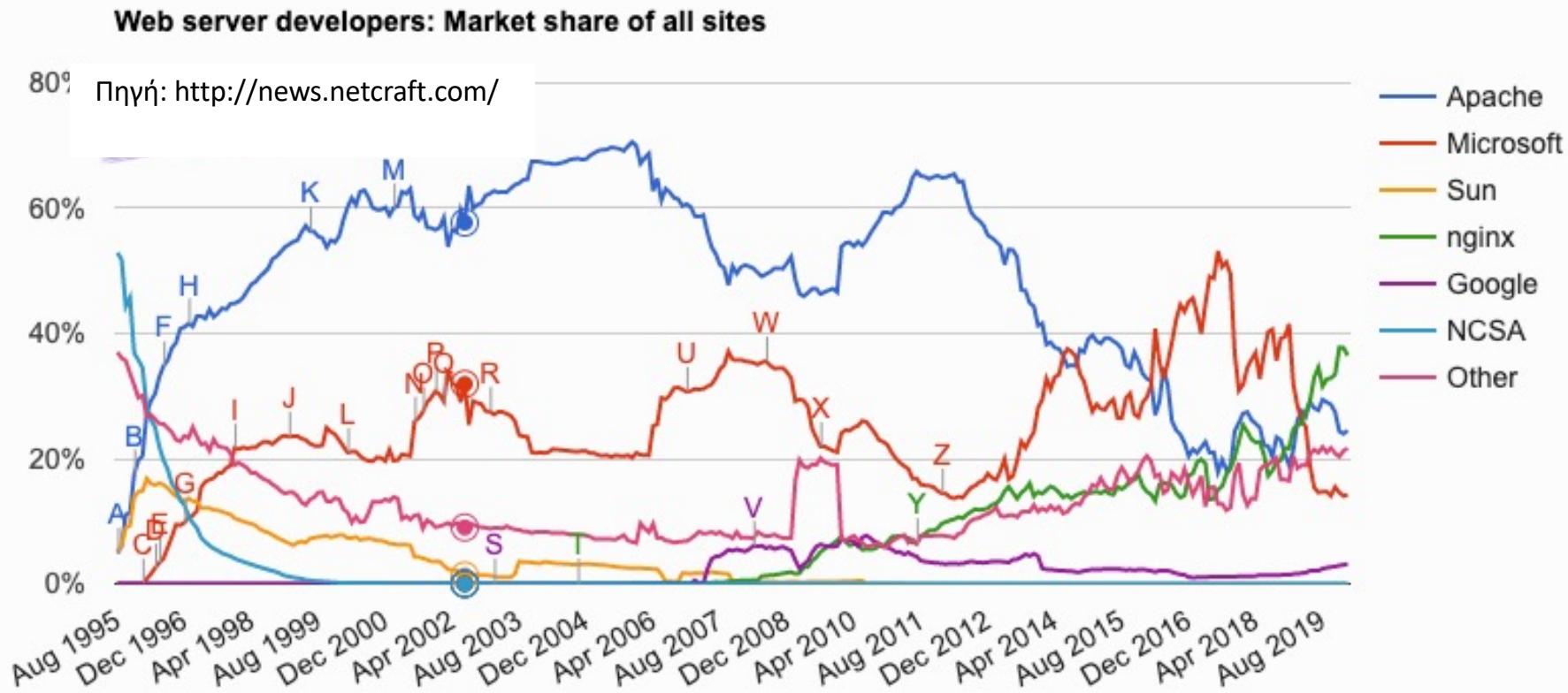


Μορφοποίηση URLs

- Χαρακτήρες όπως ‘&,’ ‘+,’ ‘%’ και ο κενός χαρακτήρας (space) κωδικοποιούνται ως "%xx" όπου xx είναι η τιμή ASCII σε δεκαεξαδική μορφή, π.χ. ‘&’ = "%26"
- Οι παράμετροι δίδονται ως λίστα παραμέτρων και τιμών που ενώνονται και διαχωρίζονται με το ‘&’
 - var1=value1&var2=value2&var3=value3

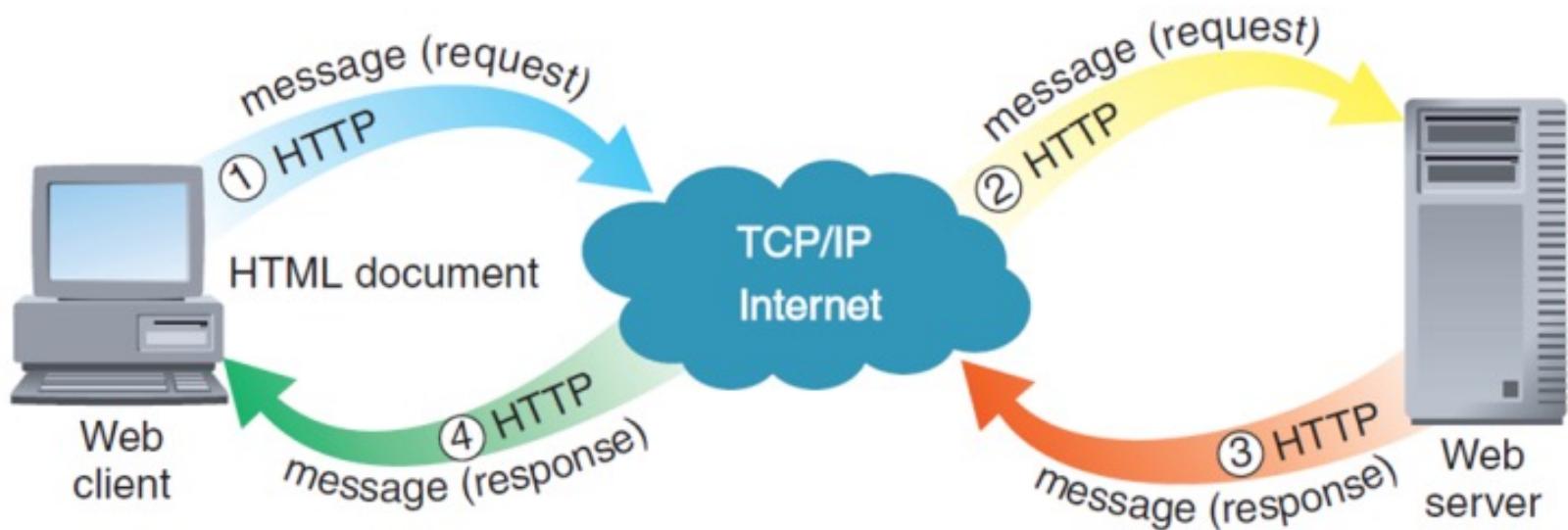
Web Servers

- Ο Web Server είναι ένα λογισμικό που εξυπηρετεί αιτήματα HTTP



το πρωτόκολλο http

Το πρωτόκολλο HTTP



Web client → HTTP REQUEST → TCP/IP internet → HTTP REQUEST — > Web server

Web server→ HTTP RESPONSE → TCP/IP internet → HTTP RESPONSE — > Web Client

HyperText Transfer Protocol

- Ελληνική Απόδοση: **Πρωτόκολλο
Μεταφοράς Υπερκειμένου**
- Υπερκείμενο;
 - Ένα σύνολο από κείμενα με ενσωματωμένους συνδέσμους σε άλλα κείμενα (υπερσυνδέσμους)
 - Η Έννοια του υπερκειμένου ήταν γνωστή πολύ πριν το διαδίκτυο, όμως με το πρωτόκολλο HTTP έγινε δυνατή η διασύνδεση των κειμένων από απόσταση μέσω του διαδικτύου

HTTP κύρια χαρακτηριστικά

Πρωτόκολλο επικοινωνίας μεταξύ client and server

Πρωτόκολλο επιπέδου εφαρμογών του διαδικτύου

- Client/Server model:

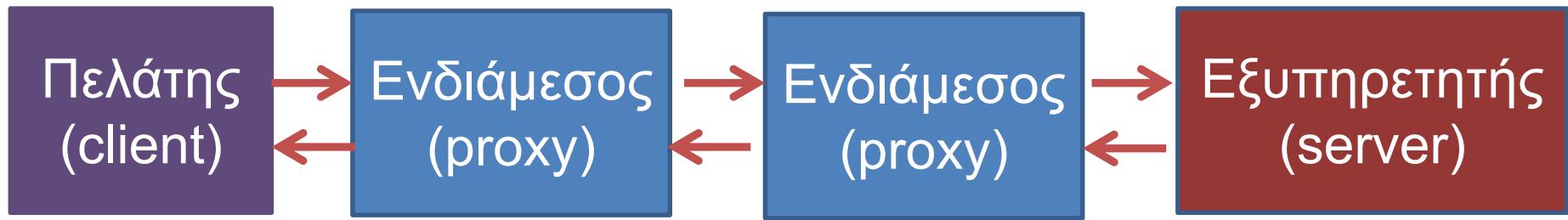
- Client: browser στέλνει αίτημα, λαμβάνει περιεχόμενο και το παρουσιάζει
- Server: λαμβάνει αιτήματα και ανταποκρίνεται

- Εκδόσεις

- **HTTP 1.0 (RFC 1945, 1996)**
- **HTTP 1.1 (RFC 2616, 1999)**
- **HTTP 2.0 (RFC 7540, 2015)** 50% χρήσης σήμερα
- **HTTP 3.0 (draft, 2021)**, 5% χρήση σήμερα

Χαρακτηριστικά του HTTP

- **Character-based protocol** – Πρωτόκολλο επικοινωνίας με κείμενο/χαρακτήρες
- **Persistent connections** (στο HTTP 1.1) – Επίμονη σύνδεση TCP για πολλαπλές μεταφορές δεδομένων
- **Stateless Protocol** – Μη-μνημόνευση προηγούμενης επικοινωνίας
- **Proxy caching** (βελτιωμένο στο HTTP 1.1) – Χρήση διαμεσολαβητών για προσωρινή αποθήκευση περιεχομένου
- **Content negotiation** – Διαπραγμάτευση περιεχομένου



HTTP 1.0

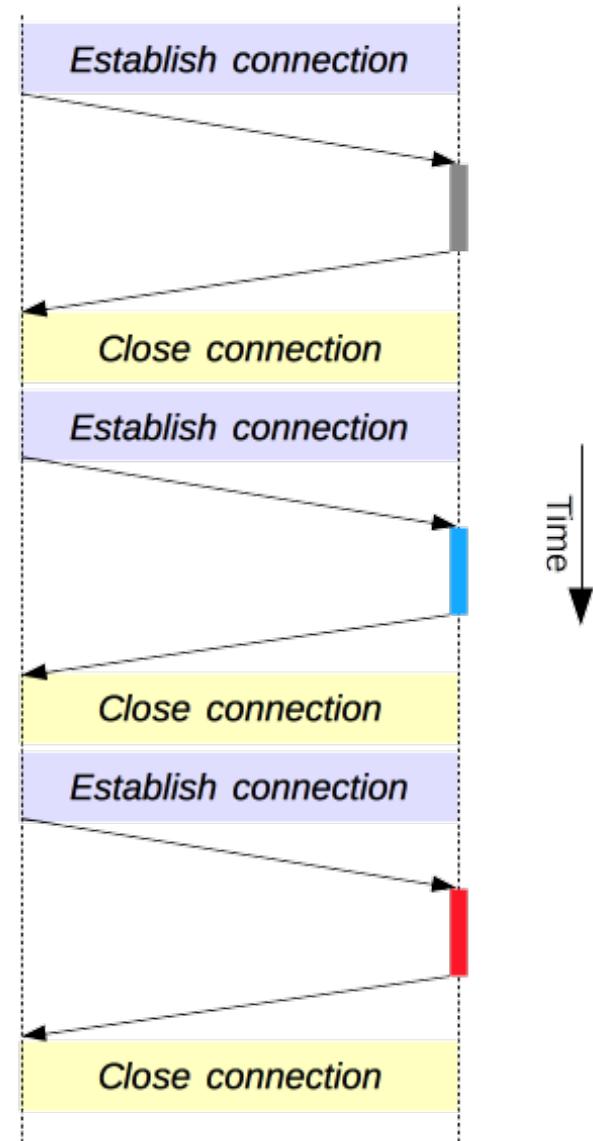
Client-Server Session

Μια βασική συνεδρία στο HTTP 1.0 έχει τις εξής φάσεις:

1. O Client εκκινεί μια σύνδεση TCP.
2. O Client στέλνει το αίτημα (request).
3. O Server αποστέλλει μια απάντηση (response).
4. O Server κλείνει τη σύνδεση TCP.

Αυτό επαναλαμβάνεται για όσα αιτήματα προκύψουν

Client Server



HTTP 1.1

Client-Server Session

Το HTTP 1.1 πρόσθεσε νέου τύπου συνδέσεις

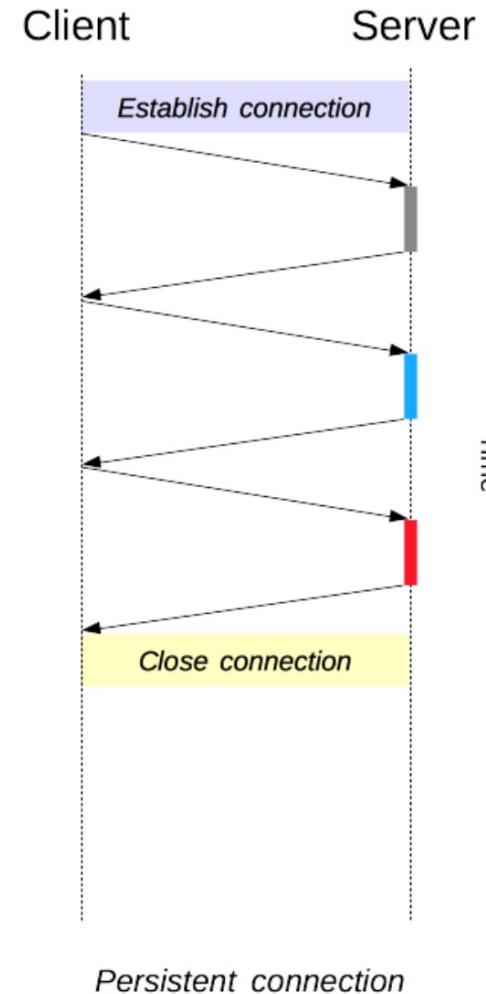
- Επίμονες συνδέσεις
- Τεχνική διεκπαιρέωσης

Persistent connections

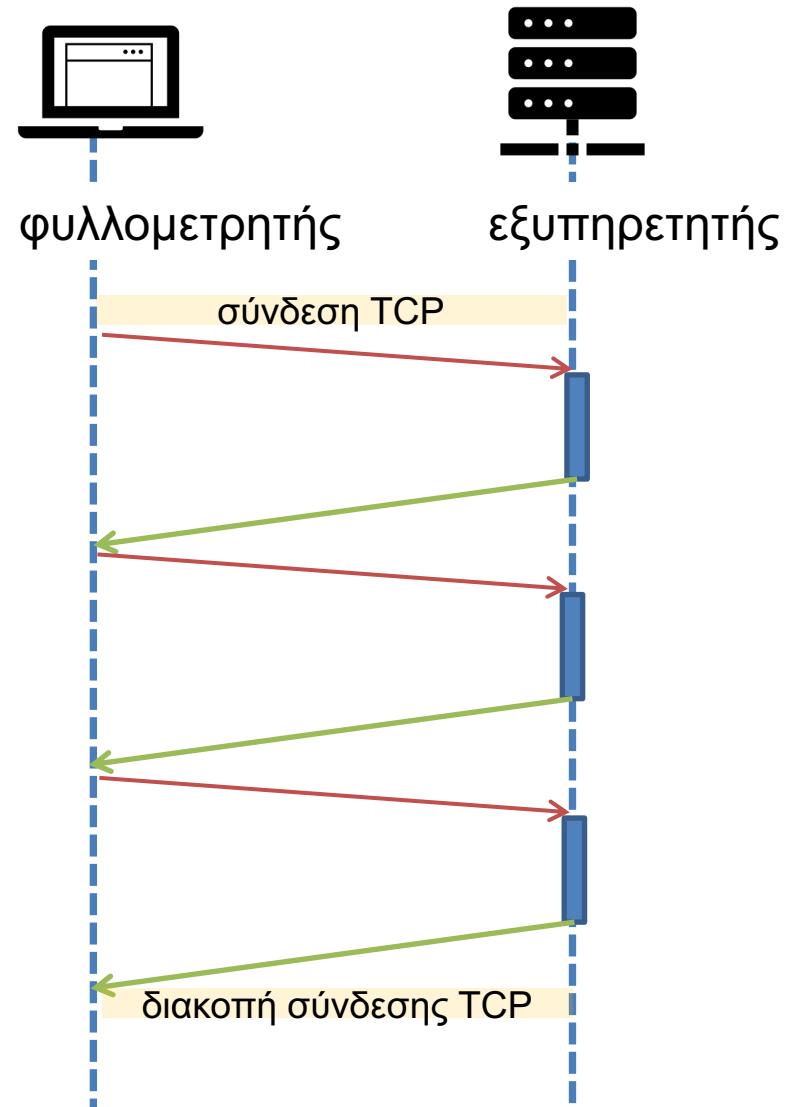
Επίμονες συνδέσεις στο HTTP/1.1

Αν μια σελίδα έχει 10 εικόνες, τότε απαιτούνται 11 HTTP/1.0 συνεδρίες για να εμφανιστεί ολόκληρη η σελίδα στον browser. Κάθε συνεδρία απαιτεί την έναρξη μιας νέας σύνδεσης TCP (αργή έναρξη).

Στο HTTP/1.1, μια **επίμονη σύνδεση TCP** είναι αρκετή. Απαιτείται λιγότερος χρόνος για να εμφανιστεί ολόκληρη η σελίδα καθώς μειώνετο το απαιτούμενο εύρος ζώνης και ο χρόνος απόκρισης. Επιπλέον μειώνεται και ο χρόνος που απασχολείται ο επεξεργαστής για την εγκαθίδρυση και τερματισμό των συνδέσεων.

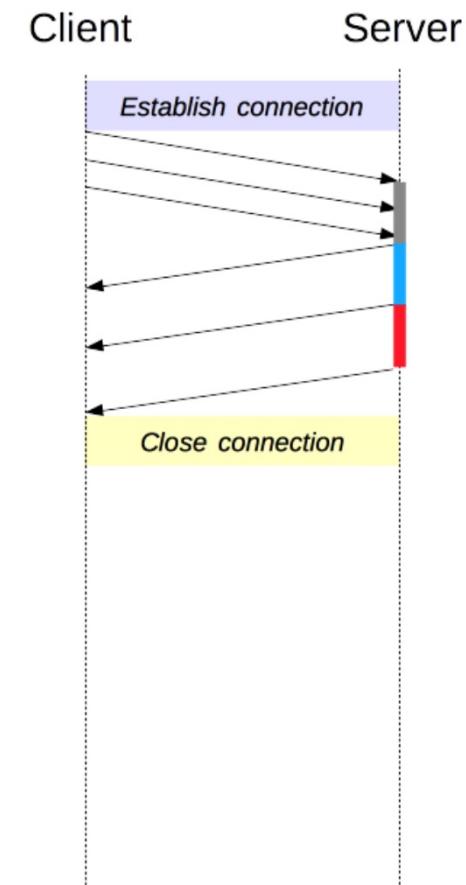


Επίμονες συνδέσεις HTTP/1.1



HTTP Pipelining

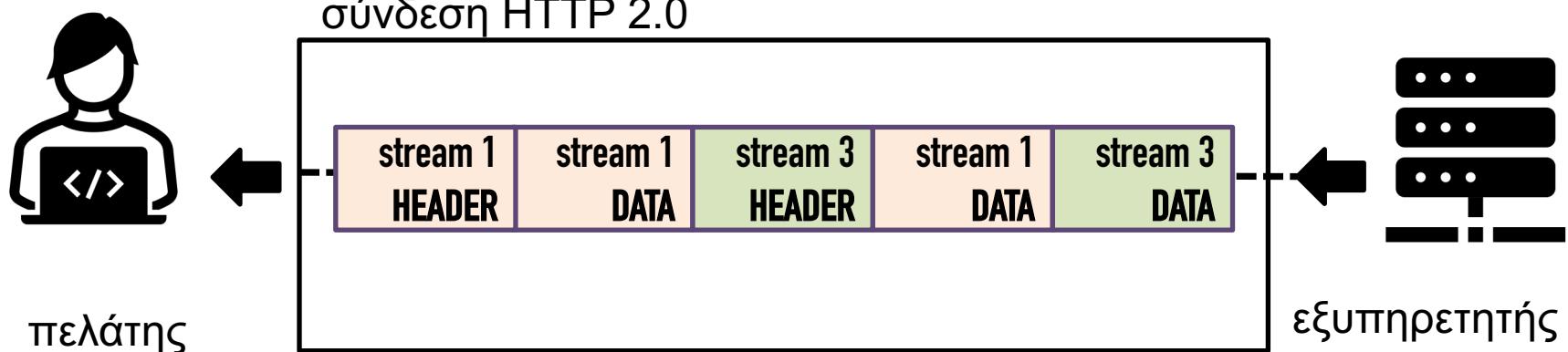
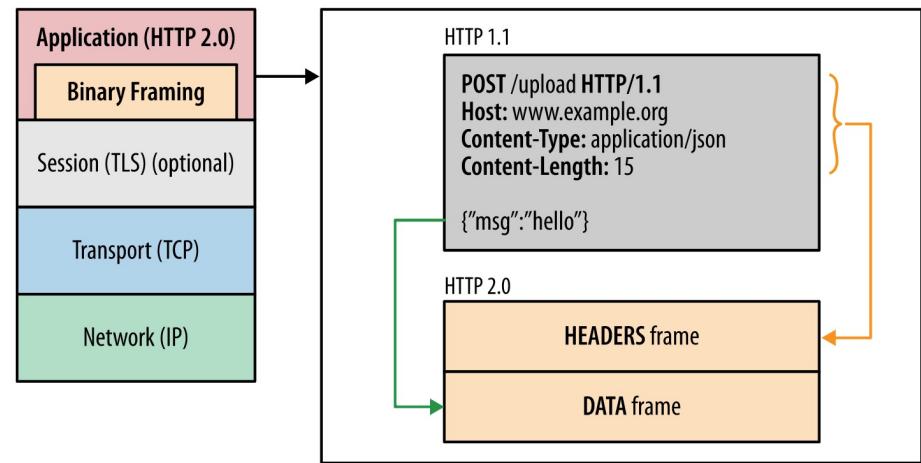
- Με την **τεχνική διεκπεραίωσης** (Pipelining), ο Client στέλνει ανεξάρτητες μεταξύ τους αιτήσεις, διαδοχικά μέσα από το επίμονο κανάλι επικοινωνίας, χωρίς να περιμένει πρώτα τον Server να απαντήσει στην προηγούμενη αίτηση.
- Εξαιρείται η αίτηση-απάντηση για τον πόρο του εγγράφου, καθώς αυτός ορίζει ποιοι θα είναι οι επόμενοι πόροι που θα ζητήσει ο Client.



HTTP Pipelining

HTTP/2 data frame multiplexing

- Το νέο πρωτόκολλο HTTP/2 δεν αλλάζει το API και τη σημασιολογία του HTTP/1.1 όμως αλλάζει τον τρόπο που κωδικοποιούνται και μεταδίδονται τα μηνύματα
- Κάθε μήνυμα **κωδικοποιείται και χωρίζεται σε binary frames**



HTTP: Stateless Protocol

- Το HTTP είναι πρωτόκολλο χωρίς ιστορικό της προηγούμενης κατάστασης:
 - Όταν ένας Server έχει μεταφέρει τα αιτούμενα δεδομένα στον Client, ο Server δεν θυμάται τι συνέβη (ακόμα και αν η σύνδεση είναι επίμονη).
- Υπάρχουν διάφοροι μηχανισμοί μνημόνευσης κατάστασης στη πλευρά του Server (πχ χρήση cookies).

Διαπραγμάτευση Περιεχομένου

- Οι Client και Server μπορούν να διαπραγματευτούν και να συμφωνήσουν μετάδοση δεδομένων.
- π.χ. συμφωνούν στην αποστολή συμπιεσμένης μορφής gzip ενός εγγράφου, 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.
- Άλλα θέματα για διαπραγμάτευση :
 - Κωδικοσελίδα χαρακτήρων εγγράφου.
 - Τύπος εικόνων του εγγράφου.
 - Γλωσσική έκδοση του περιεχομένου του εγγράφου.
 - Εξυπηρέτηση της αίτησης, μόνο εφόσον ο πόρος έχει τροποποιηθεί.

Multipurpose Internet Mail Extensions (MIME)

**Πρωτόκολλο για μετάδοση διαφορετικών τύπων δεδομένων,
αρχικά ως επέκταση του πρωτόκολλου ανταλλαγής email**

maintype/subtype: για καθιερωμένους τύπους δεδομένων, π.χ. text/html, image/jpeg, application/octet-stream

maintype/x-subtype: για μη καθιερωμένους τύπους δεδομένων, π.χ. application/x-gzip, image/x-icon, application/x-java-class

application	Δεδομένα Εφαρμογής
audio	Δεδομένα Ήχου
image	Δεδομένα Εικόνας
message	Δεδομένα Μηνύματος
model	Δεδομένα Μοντελοποίησης
multipart	Δεδομένα Πολλαπλών Τύπων
text	Δεδομένα Κειμένου
video	Δεδομένα Κινούμενης Εικόνας

Καταχωρημένοι Τύποι MIME

Παράδειγμα τύπων MIME

application/force-download	Οδηγία εξαναγκασμένης μεταφοράς αρχείου
application/download	Οδηγία εξαναγκασμένης μεταφοράς αρχείου
application/javascript	Αρχείο ή κείμενο με κώδικα JavaScript
application/octet-stream	Αρχείο με καθαρά δυαδικά δεδομένα
application/pdf	Αρχείο Portable Document Format, PDF
application/vnd.ms-excel	Αρχείο του Microsoft Excel
application/x-gzip	Αρχείο συμπιεσμένης δομής αρχείων gzip
application/x-www-form-urlencoded	Δεδομένα φόρμας HTML κωδικοποιημένα κατά URL
image/gif	Αρχείο ψηφιακής εικόνας gif
image/jpeg	Αρχείο ψηφιακής εικόνας jpg
image/png	Αρχείο ψηφιακής εικόνας png
message/rfc822	Δεδομένα ηλεκτρονικού ταχυδρομείου
multipart/form-data	Πολλαπλά δεδομένα από φόρμα HTML
text/csv	Αρχείο με τιμές διαχωρισμένες με comma (;)
text/html	Αρχείο εγγράφου HTML
text/plain	Αρχείο απλού κειμένου (txt)
text/xml	Αρχείο εγγράφου XML
video/mpeg	Αρχείο video κωδικοποιημένο κατά MPEG
video/quicktime	Αρχείο video κωδικοποιημένο κατά QuickTime

Αίτημα HTTP

Αίτημα HTTP (HTTP request)

- Μέθοδος αίτησης (Request Method),
 - GET/ HEAD/ POST
- Ταυτότητα αιτούμενου πόρου
- Έκδοση πρωτοκόλλου επικοινωνίας (π.χ. HTTP/1.1)
- Κεφαλίδες πληροφοριών αίτησης
- Ένα μήνυμα σε μορφή MIME με πληροφορίες από τον αιτούντα

Μέθοδοι HTTP Request

- **GET**: ανάκτηση αρχείου (95% αιτημάτων)
- **HEAD**: ανάκτηση μεταδεδομένων (πχ., mod time)
- **POST**: υποβολή δεδομένων φόρμας
- **PUT**: αποθήκευση επισυναπτόμενου εγγράφου ως URI (νέο στο 1.1)
- **DELETE**: διαγραφή πόρου (resource) (νέο :1.1)
- **LINK/UNLINK**: μόνο στο 1.0, όχι πια στο 1.1
- **TRACE**: http “echo” for debugging (νέο στο 1.1)
- **CONNECT**: χρήση από proxies (1.1)
- **OPTIONS**: επιλογές του proxy (1.1)

Δομή Αίτησης HTTP

Μέθοδος	sp	URL	sp	Έκδοση	cr	lf
Κεφαλίδα	sp	Τιμή	cr	lf	...	
Κεφαλίδα	sp	Τιμή	cr	lf	...	
cr	lf					

Γραμμές Κεφαλίδων

Κείμενο Μηνύματος
(σε POST αιτήσεις μόνο)

Σημείωση: cr lf =
"\r\n" (ASCII 13
ακολουθούμενο από
ASCII 10)

Η αρχική γραμμή η οποία περιέχει τη μέθοδο της αίτησης, το URL του ζητούμενου πόρου, καθώς και την έκδοση του HTTP πρωτοκόλλου που χρησιμοποιείται.

Καμία, μια ή περισσότερες γραμμές header.

Μια κενή διαχωριστική γραμμή (CRLF).

Ένα προαιρετικό μήνυμα (π.χ. ένα αρχείο που μεταφέρουμε στο server ή δεδομένα κάποιας φόρμας).

Παράδειγμα αίτησης HTTP



Κεφαλίδες - Headers

- HTTP 1.0 ορίζει **16 κεφαλίδες (headers)**
- HTTP 1.1 ορίζει **46 κεφαλίδες**
 - Καμιά δεν είναι υποχρεωτική, πλην της HOST όταν το αίτημα απευθύνεται σε web server.
 - Μια αίτηση που αποστέλλεται σε έναν proxy δεν χρειάζεται να περιλαμβάνει κεφαλίδες.
 - Μια απάντηση από Server δεν χρειάζεται να περιέχει κεφαλίδες.

Αίτηση GET: η πιο συνηθισμένη μέθοδος της HTTP

- Χρησιμοποιείται για την αίτηση μεταφοράς αρχείου από τον Server ή εκτέλεσης προγράμματος σε αυτόν.
- Το URI περιέχει τον πόρο που ζητήθηκε.
- Η πιο συχνά χρησιμοποιούμενη αίτηση.
- Δεν περιέχει μήνυμα αλλά το URI μπορεί να περιέχει παραμέτρους προς τον πόρο του Web Server (αναφερόμενες ως μεταβλητές τύπου GET).

`http://www.upatras.gr/welcome.html`

`GET /welcome.html HTTP/1.0[CRLF]
HOST: www.upatras.gr[CRLF]
[CRLF]`

Παραδείγματα Αιτήσεων HTTP

```
GET /index.php HTTP/1.0[CRLF]  
[CRLF]
```

```
GET /index.php HTTP/1.1[CRLF]  
Host: www.upatras.gr:80[CRLF]  
Accept: image/gif, image/jpeg[CRLF]  
User-Agent: Mozilla/4.0[CRLF]  
Connection: Keep-Alive[CRLF]  
[CRLF]
```

Αίτηση HEAD

- Σύνταξη όμοια με την GET
- Επιστρέφει πληροφορίες κεφαλίδας για το ζητούμενο URL χωρίς να επιστρέφει και το περιεχόμενο αυτού
 - Χρήσιμο για την ανάκτηση πληροφοριών σχετικών με το URL χωρίς να απαιτείται η μεταφορά του ίδιου του πόρου (απαιτεί λιγότερο χρόνο και μεταφορά μικρότερου όγκου δεδομένων).
 - Οι clients μπορούν να μεταφέρουν πληροφορίες μόνο όταν αυτές έχουν διαπιστωμένα ανανεωθεί.
 - Χρησιμοποιείται από τους Proxy Servers για τον έλεγχο ανάγκης ανανέωσης τοπικά αποθηκευμένων σελίδων.

Αίτηση POST

Χρησιμοποιείται για την αποστολή δεδομένων ή κωδικοποιημένων αρχείων προς τον Web Server.

- Συχνά χρησιμοποιούμενη για αποστολή στοιχείων από φόρμες.
- Επίσης για μεταφόρτωση αρχείων από client προς server (uploads).
- Τα δεδομένα που συμπληρώθηκαν από το χρήστη στη φόρμα ή τα περιεχόμενα του αρχείου, μετατρέπονται σε κατάλληλη μορφή, εισάγονται στο σώμα μηνύματος της αίτησης και αποστέλλονται προς το πρόγραμμα επεξεργασίας αυτών στον Web Server.
- Απαιτούνται επιπλέον headers για να περιγραφεί το μήνυμα, όπως οι **Content-Type** και **Content-Length**.
- Το αιτούμενο URL είναι η διεύθυνση πόρου ενός προγράμματος που διαχειρίζεται τα αποστελλόμενα δεδομένα.
- Η απάντηση HTTP είναι συνήθως η έξοδος του προγράμματος αυτού και όχι ένα στατικό αρχείο.

Παραδείγματα Αιτήσεων HTTP

```
POST /receiver.php HTTP/1.1[CRLF]
```

```
Host: www.upatras.gr:80[CRLF]
```

```
Content-Type: application/x-www-form-
    urlencoded; charset=UTF-8[CRLF]
```

```
Content-Length: 50[CRLF]
```

```
[CRLF]
```

```
ModuleCode=243607&FileID=2207&rndval=128647
```

```
6333676[CRLF]
```

Κεφαλίδες Μηνυμάτων Αίτησης στο HTTP/1.0 και HTTP/1.1

Κεφαλίδες Αίτησης

- Οι πιο κοινές κεφαλίδες μηνυμάτων αίτησης του HTTP/1.1 αναλύονται στη συνέχεια:
 - Host
 - Accept
 - Accept-Encoding
 - Authorization
 - Connection
 - Cookie
 - If-Modified-Since
 - Referer
 - User-Agent

Κεφαλίδα Αίτησης Host

- Host:
 - Δείχνει το όνομα του συνδεόμενου Server και της πόρτας σύνδεσης TCP, όπως αυτά ορίζονται στο URL του Client.
 - **Απαραίτητο** να οριστεί σε αιτήσεις από το HTTP 1.1, ειδάλλως οδηγεί σε επιστρεφόμενη απάντηση ένδειξης σφάλματος.
 - Απαιτείται εξαιτίας της προώθησης αιτήσεων σε διαφορετικούς Servers και σε Web Servers που έχουν πολλαπλά ονόματα εξυπηρετητή.

Κεφαλίδες Αίτησης Accept

Οι αναφερόμενες κεφαλίδες ορίζουν επιθυμητούς και αποδεκτούς τύπους δεδομένων από τον Browser.

Οι διαφορετικοί τύποι διαχωρίζονται με κόμμα. Συχνά ακολουθούνται από έναν αριθμό μεταξύ 0 και 1, που ορίζει το σχετικό βάρος του τύπου. Όταν δεν ορίζεται βάρος, τότε θεωρείται η τιμή 1.

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

- **Accept:**
 - Ορίζει τους τύπους MIME που μπορεί να χειριστεί ο Client (π.χ. text/html, image/gif).
- **Accept-Encoding:**
 - οι τύποι κωδικοποίησης που αποδέχεται ο Client (π.χ. συμπίεση gzip).
- **Accept-Charset:**
 - οι υποστηριζόμενες κωδικοσελίδες περιεχομένου (π.χ. ISO-8859-7).
- **Accept-Language:**
 - οι υποστηριζόμενες γλώσσες περιεχομένου (π.χ. en, en-us, gr, de).

Κεφαλίδα Αίτησης Authorization

- Authorization:
 - Αναγνώριση και ταυτοποίηση χρήστη για εξουσιοδότηση πρόσβασης σε προστατευμένες από κωδικό σελίδες.
- Αντί για εξουσιοδότηση μέσω HTTP authorization, μπορούμε να χρησιμοποιήσουμε μηχανισμούς όπως:
 - Φόρμες HTML για αποστολή ονόματος και κωδικού χρήστη (βλέπετε αίτηση POST).
 - Αποθήκευση ονόματος και κωδικού χρήστη μέσω μηχανισμού διατήρησης κατάστασης σύνδεσης (session management).

Κεφαλίδα Αίτησης Connection

- **Connection: keep-alive**
 - Δείχνει ότι ο φυλλομετρητής μπορεί να διαχειριστεί επίμονες συνδέσεις (persistent connections).
 - Εξ ορισμού κατάσταση σύνδεσης στο HTTP 1.1.
 - Σε μια **επίμονη σύνδεση**, ο Client και ο Server χρησιμοποιούν την ίδια σύνδεση TCP για μια σειρά αιτήσεων και απαντήσεων.
- **Connection: close**
 - Δείχνει ότι η σύνδεση TCP θα κλείσει μόλις ο Client λάβει μια απάντηση.
 - Εξ ορισμού συμπεριφορά στο HTTP 1.0.

Κεφαλίδα Αίτησης Content-Length

- **Content-Length: xxxx**
 - Η κεφαλίδα αυτή χρησιμοποιείται μόνο σε αιτήσεις POST.
 - Ο αριθμός **XXXX** εκφράζει το μέγεθος των αποστελλόμενων 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.

Κεφαλίδα Αίτησης If-Modified-Since

- **If-Modified-Since:** <timestamp>
 - Η κεφαλίδα αυτή δείχνει ότι ο Client διαθέτει ήδη ένα αντίγραφο της σελίδας με χρονική σήμανση timestamp και ζητάει από τον Server να στείλει την αιτούμενη σελίδα μόνο εφόσον αυτή έχει τροποποιηθεί μεταγενέστερα της δηλωμένης χρονικής στιγμής.
 - Η χρονική σήμανση είναι της μορφής GMT, όπως φαίνεται παρακάτω:

```
Last-Modified: Thu, 31 Dec 2020 23:59:59 GMT
```

Κεφαλίδα Αίτησης Referer

- Referer: <URL>
 - Περιέχει το URL από το οποίο ο Client αναφέρεται στο αιτούμενο URL του Server (δηλ. το URL της σελίδας στην οποία ανήκει ο υπερσύνδεσμος που οδήγησε στην αίτηση αυτή).
 - Χρησιμοποιείται για έλεγχο της κίνησης πληροφοριών.
 - Καταγράφεται από τους Web Servers για στατιστική ανάλυση και μπορεί να χρησιμοποιηθεί για την απόδοση τροποποιημένου δυναμικού περιεχομένου.
 - Ορθογραφικά λανθασμένη (σωστή: Referrer) για λόγους διευκόλυνσης στη γραφή.

Κεφαλίδα Αίτησης User-Agent

- User-Agent: xxxx
 - Η συμβολοσειρά **xxxx** περιέχει κείμενο που αναγνωρίζει τον browser του Client που κάνει την αίτηση.
 - Δεν χρησιμοποιείται πάντοτε, ενώ το περιεχόμενο της δεν μπορεί να θεωρείται αξιόπιστο.

```
User-Agent: Mozilla/5.0 (Windows; U; Windows NT  
5.1; en-US; rv:1.7.12) Gecko/20050915  
Firefox/1.0.7
```

Άσκηση: Αναγνωρίστε το αίτημα HTTP

GET / HTTP/1.1

Host: www.upatras.gr

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Accept-Encoding: gzip, deflate

Accept-Language: en-GB,en-US;q=0.9,en;q=0.8

Connection: keep-alive

Cookie: cookie-agreed=2; has_js=1; _ga=GA1.2.1167840985.1613743305; SAPWP_active=1; _gid=GA1.2.551254727.1614007388

Host: www.upatras.gr

Upgrade-Insecure-Requests: 1

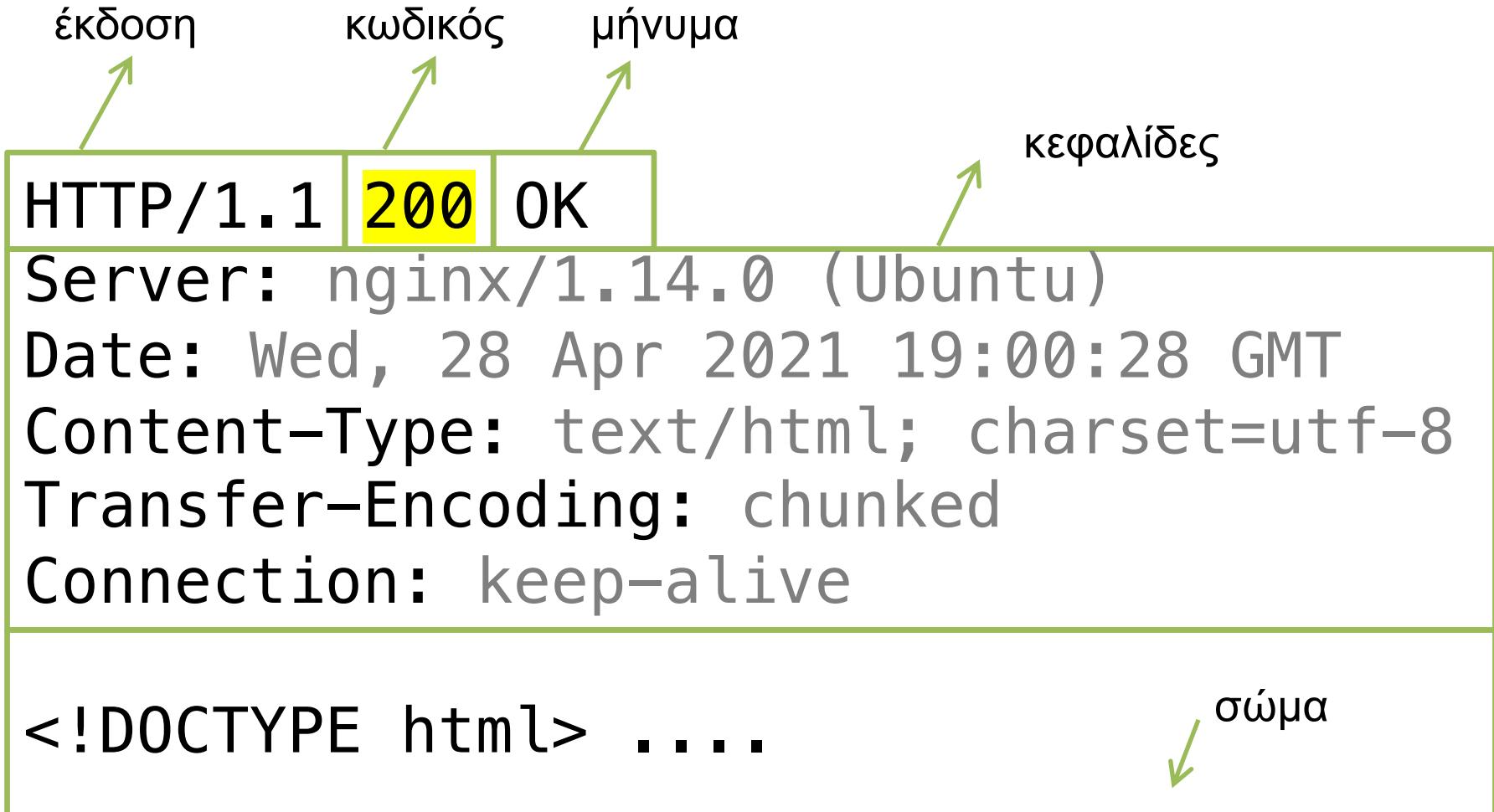
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6)

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150

Safari/537.36

Απόκριση HTTP

Παράδειγμα απόκρισης HTTP



Περιγραφή απόκρισης HTTP

- Η αρχική γραμμή της απάντησης ονομάζεται **γραμμή κατάστασης**
- Η αρχική γραμμή αποτελείται από:
 - Έκδοση πρωτοκόλλου HTTP (π.χ. HTTP/1.1)
 - Κωδικός κατάστασης απάντησης (**Status code**)
 - Φράση περιγραφής του κωδικού κατάστασης (**reason phrase**)

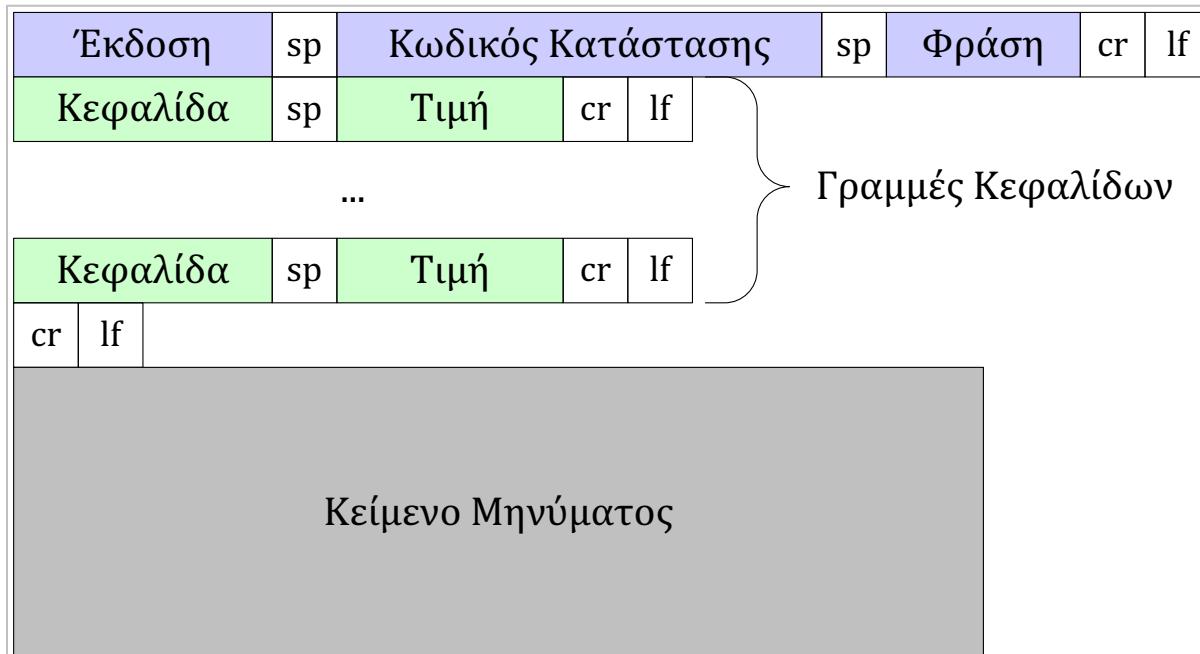
Δομή απόκρισης στο HTTP

Η αρχική γραμμή περιέχει τον κωδικό κατάστασης της επεξεργασίας της αίτησης.

Καμία, μια ή περισσότερες γραμμές header.

Μια κενή γραμμή (CRLF). CRLF = "\r\n" (ASCII 13 ακολουθούμενο από ASCII 10)

Ένα προαιρετικό μήνυμα (π.χ. ένα αρχείο που μεταφέρεται στον Browser, δεδομένα αναζήτησης ή αποτελέσματα αναζήτησης).



Περιγραφή Κωδικών Κατάστασης

- Ο κωδικός κατάστασης είναι ένας ακέραιος αριθμός τριών δεκαδικών ψηφίων
- Το πρώτο ψηφίο καθορίζει την κατηγορία της απάντησης:
 - **1xx**: μήνυμα πληροφορίας
 - **2xx**: κατάσταση επιτυχούς επεξεργασίας
 - **3xx**: προώθηση του Client σε δι?φορετικό URL
 - **4xx**: σφάλμα προερχόμενο από τον Client
 - **5xx**: σφάλμα προερχόμενο από τον Server

Κωδικοί κατάστασης 1xx

- 100 Continue
- Επιτρέπει στον Client να καταλάβει αν ο Server είναι διατεθειμένος να αποδεχτεί την αίτηση, προτού ο Client στείλει το μήνυμα της αίτησης
- Χρησιμοποιείται προκειμένου να ελέγξει ο Client αν μπορεί ο Server :
 - να δεχθεί ένα μεγάλο σε μέγεθος δεδομένων μήνυμα POST,
 - αν ο Server μπορεί να δεχθεί τον τύπο των αποστελλόμενων δεδομένων της αίτησης POST
 - αν απαιτείται εξουσιοδοτημένη πρόσβαση,
- Η αίτηση του Client θα πρέπει να περιέχει την κεφαλίδα: Expect: 100 (Continue)

Κωδικοί κατάστασης 2xx

- 200 Success
 - Η αίτηση έγινε δεκτή, η επεξεργασία της ολοκληρώθηκε επιτυχώς και τα αποτελέσματα ακολουθούν.
- 202 Accepted
 - Η αίτηση έγινε δεκτή και η επεξεργασία δεν έχει ακόμα ολοκληρωθεί.
- 204 No Content
 - Ο Browser θα πρέπει να συνεχίσει να δείχνει το προηγούμενο περιεχόμενο.
- 205 Reset Content
 - Δεν υπάρχει νέο περιεχόμενο, αλλά ο Browser πρέπει να καθαρίσει την εμφανιζόμενη σελίδα.
 - Χρησιμοποιείται προκειμένου να αναγκάσει τον Browser να καθαρίσει τα πεδία της φόρμας δεδομένων που εμφανίζει.
 - Ορίζεται στο HTTP 1.1.

Κωδικοί κατάστασης 3xx

- **301 Moved Permanently**
 - Το νέο URL ορίζεται από την κεφαλίδα Location.
 - Οι Browsers πρέπει αυτόματα να ακολουθούν τον υπερσύνδεσμο προς το νέο URL, σε πλήρη αντικατάσταση του αρχικού URL.
- **302 Moved Temporarily**
 - Στο HTTP 1.1 ορίζεται ως "Found", αλλά το "Moved Temporarily" συνεχίζει να χρησιμοποιείται.
 - Παρόμοια με το 301, με τη διαφορά ότι το URL που ορίζεται από την κεφαλίδα Location είναι προσωρινό και η επόμενη αίτηση για το αρχικό URL μπορεί να μην αντιμετωπιστεί με εκτροπή σε άλλο URL.
 - Οι περισσότεροι Browsers χειρίζονται τον κωδικό 301 κατ' όμοιο τρόπο με τον κωδικό 302.

Κωδικοί κατάστασης 3xx (συν.)

- 303 See Other
 - Παρόμοια με τα 301 και 302, με τη διαφορά πως αν η αρχική αίτηση ήταν τύπου POST, η νέα αίτηση στο προωθημένο έγγραφο πρέπει να γίνει με αίτηση τύπου GET.
 - Εισάχθηκε στο HTTP 1.1.
- 304 Not Modified
 - Απάντηση σε αίτηση που χρησιμοποιεί την κεφαλίδα If-Modified-Since όταν η σελίδα δεν έχει τροποποιηθεί μεταγενέστερα της ορισμένης χρονικής στιγμής.
 - Αν η σελίδα έχει τροποποιηθεί, τότε η νέα σελίδα πρέπει να επιστραφεί με έναν κωδικό κατάσταση 200 OK.

Κωδικοί κατάστασης 3xx (συν.)

- 307 Temporary Redirect
 - Το νέο URL δίδεται από την κεφαλίδα Location.
 - Μονάχα αιτήσεις τύπου GET, αλλά όχι POST θα πρέπει να ακολουθούν το νέο URL.
 - Στον κωδικό κατάστασης 303 (See Other), τόσο οι αιτήσεις τύπου GET όσο και οι POST ακολουθούν το νέο URL.
 - Εισάχθηκε στο HTTP 1.1.

Κωδικοί κατάστασης 4xx

- 400 Bad Request
 - Συντακτικό λάθος στην αίτηση.
- 401 Unauthorized
- 403 Forbidden
 - Άρνηση πρόσβασης στον Server να προσπελάσει τη ζητούμενη σελίδα.
- 404 Not Found
 - Το URL αναφέρεται σε πόρο που δεν υπάρχει στον Server. Συχνό σφάλμα όταν ο χρήστης έχει πληκτρολογήσει λάθος όνομα αρχείου ή το αρχείο έχει διαγραφεί από τον Server.

Κωδικοί κατάστασης 4xx (συν.)

- 405 Method Not Allowed
- 410 Gone
- 411 Length Required
- 413 Request Entity Too Large
- 414 Request URI Too Long
- 415 Unsupported Media Type
- 417 Expectation Failed
 - Αν ο Server δεχθεί μια αίτηση με την κεφαλίδα Expect να έχει τιμή 100, σημαίνει πως ο Client ζητά να μάθει αν μπορεί να στείλει ένα επισυναπτόμενο έγγραφο σε επόμενη αίτηση.
 - Ο Server απαντά είτε με μήνυμα λάθους 417 ή 100 Continue.

Κωδικοί κατάστασης 5xx

- 500 Internal Server Error
 - Εσωτερικό σφάλμα στη λειτουργία του Server, ο λόγος πρέπει να αναζητηθεί από το διαχειριστή.
- 501 Not Implemented
- 502 Bad Gateway
- 503 Service Unavailable
 - Η απάντηση μπορεί να περιέχει μια κεφαλίδα Retry-After για να πει στον Client πότε μπορεί να ξαναστείλει την αίτηση του.
- 505 HTTP Version Not Supported
 - εισήχθη στο HTTP 1.1.

**Κεφαλίδες μηνυμάτων
απόκρισης
στο HTTP/1.0 και
HTTP/1.1**

Κεφαλίδες απόκρισης

- Ορισμός τοποθεσίας προώθησης.
- Καθορισμός cookies.
- Καθορισμός ημερομηνίας τροποποίησης της σελίδας.
- Καθοδήγηση του Browser να επαναφορτώσει τη σελίδα μετά από καθορισμένο χρονικό διάστημα.
- Απόδοση του μεγέθους του εγγράφου ώστε να χρησιμοποιηθούν επίμονες συνδέσεις (keep-alive).
- Καθορισμός του τύπου του εγγράφου που δημιουργήθηκε.
- Καθορισμός της γλώσσας και της κωδικοσελίδας εμφάνισης του εγγράφου.
- Ενεργοποίηση και έλεγχος της προσωρινής αποθήκευσης των πόρων σε Proxy Server ή τον τοπικό σκληρό δίσκο.

Κεφαλίδα Απάντησης Allow

- Allow:
 - Ορίζει τις μεθόδους αίτησης (GET, POST, κ.α.) που επιτρέπονται για το αιτούμενο URL.
 - Απαιτείται για τις απαντήσεις τύπου 405 (Method Not Allowed).

Κεφαλίδα Απάντησης Connection

- Connection:
 - Η τιμή **close** πληροφορεί τον Client πως δεν μπορεί να χρησιμοποιήσει επίμονες TCP συνδέσεις.
 - Στο HTTP 1.1 η χρήση επίμονων συνδέσεων είναι η εξ ορισμού λειτουργία.

Κεφαλίδες Απάντησης Content

- Content-Encoding:
 - Δηλώνει την κωδικοποίηση της σελίδας.
- Content-Language:
 - Δηλώνει τη γλώσσα που είναι γραμμένη η σελίδα.
 - Η τιμή της θα πρέπει να είναι κάποιος από τους καθιερωμένους κωδικούς γλώσσας: **en, en-us, gr, de**.
- Content-Length:
 - Δηλώνει το πλήθος των bytes στην απάντηση.
 - Απαιτείται αν χρησιμοποιούνται επίμονες συνδέσεις προκειμένου να καθορίζονται τα όρια των απαντήσεων.
- Content-Type:
 - Δηλώνει τον τύπο MIME του εγγράφου της απάντησης.

Κεφαλίδα Απάντησης **Expires**

- **Expires:**
 - Δηλώνει τη χρονική στιγμή πέρα από την οποία, ο πόρος θα πρέπει να θεωρείται παρωχημένος και κατά συνέπεια δεν θα πρέπει να αποθηκεύεται σε προσωρινή μνήμη.
 - Χρησιμοποιείται σε πόρους, όπου το περιεχόμενο τους είναι έγκυρο για μικρό χρονικό διάστημα.
 - Για το επιθυμητό αποτέλεσμα θα πρέπει να γίνεται συνδυασμένη χρήση με τις κεφαλίδες απάντησης **Cache-Control**.

Κεφαλίδα Απάντησης **Last-Modified**

- **Last-Modified:**
 - Δηλώνει τη χρονική στιγμή τελευταίας αλλαγής του πόρου.
 - Η τιμή της κεφαλίδας (ημερομηνία και ώρα), μπορεί να χρησιμοποιηθεί σε μεταγενέστερες ερωτήσεις για τον πόρο αυτό, κάνοντας χρήση της κεφαλίδας ερώτησης **If-Modified-Since**, με σκοπό την μεταφορά μόνο κάποιας ενημερωμένης έκδοσης του πόρου.

Κεφαλίδα Απάντησης Location

- Location:
 - Δηλώνει μια νέα διεύθυνση πόρου, την οποία θα προσπελάσει ο Browser.
 - Χρησιμοποιείται για ανακατευθύνσεις σύμφωνα με κάποια εσωτερική κατάσταση του Server ή παραμέτρους εκτέλεσης προγράμματος.
 - Περιέχεται σε όλες τις απαντήσεις τύπου 3xx.

Άλλες Κεφαλίδες Απάντησης

- **Server :**

- Περιέχει την υπογραφή του Web Server, η οποία αναγνωρίζει τον κατασκευαστή και άλλες ενεργές λειτουργίες.

Server: nginx/1.14.0 (Ubuntu)

Άσκηση: Αναγνωρίστε την απόκριση HTTP

HTTP/1.1 301 Moved Permanently

Server: nginx/1.14.0 (Ubuntu)

Date: Thu, 25 Feb 2021 12:27:22 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

Location: http://www.upatras.gr/el

X-Frame-Options: SAMEORIGIN

X-Content-Type-Options: nosniff

X-XSS-Protection: 1; mode=block

Ειδικές Κεφαλίδες Αιτήσεων/Απαντήσεων στο HTTP

Κεφαλίδες Απάντησης. Cache-Control

Η τιμή **no-cache** αποτρέπει την προσωρινή αποθήκευση του πόρου σε κοινή (Proxy) ή ιδιωτική (Browser) ενδιάμεση μνήμη.

Η τιμή **private** αποτρέπει την αποθήκευση του πόρου σε μνήμη κοινής χρήσης αλλά μπορεί να την επιτρέψει στην ιδιωτική μνήμη του Proxy ή του Client που αιτήθηκε τη σελίδα.

Η τιμή **public** ορίζει τον πόρο ως ελεύθερο για προσωρινή αποθήκευση από κοινόχρηστες ενδιάμεσες μνήμες (Proxy ή Browser).

Διαχείριση cookies

Τα cookies είναι μικρά αρχεία που αποθηκεύονται στον browser και αποστέλλονται πίσω στον εξυπηρετητή που τα έχει δημιουργήσει.

Αντιμετωπίζουν το πρόβλημα της διαχείρισης της ιστορίας σύνδεσης

Επιτρέπουν την προσωποποίηση της πληροφορίας που εμφανίζεται στον χρήστη

Καταγράφουν τη συμπεριφορά του χρήστη

Χρήση των Κεφαλίδων Cookie και Set-Cookie



Ο εξυπηρετητής στέλνει ένα cookie

Ο φυλλομετρητής το επιστρέφει δηλώνοντας την ταυτότητά του

Κεφαλίδα απόκρισης Set-Cookie

- **Set-Cookie:**
 - Η κεφαλίδα αυτή καθορίζει ένα cookie που σχετίζεται με την εφαρμογή του Web Server.
 - Το cookie έχει διάφορα πεδία.
 - Κάθε cookie απαιτεί ξεχωριστή κεφαλίδα.
 - Η κεφαλίδα αυτή δεν είναι μέρος του HTTP 1.1 αλλά χρησιμοποιείται ευρέως.

**Set-Cookie: name=value; expires= value; path=value;
domain= value; secure**

Κεφαλίδα Αίτησης Cookie

- **Cookie:**
 - Περιέχει cookies τα οποία έχουν αποσταλεί νωρίτερα από τον Server στον Client.

Άσκηση HTTP

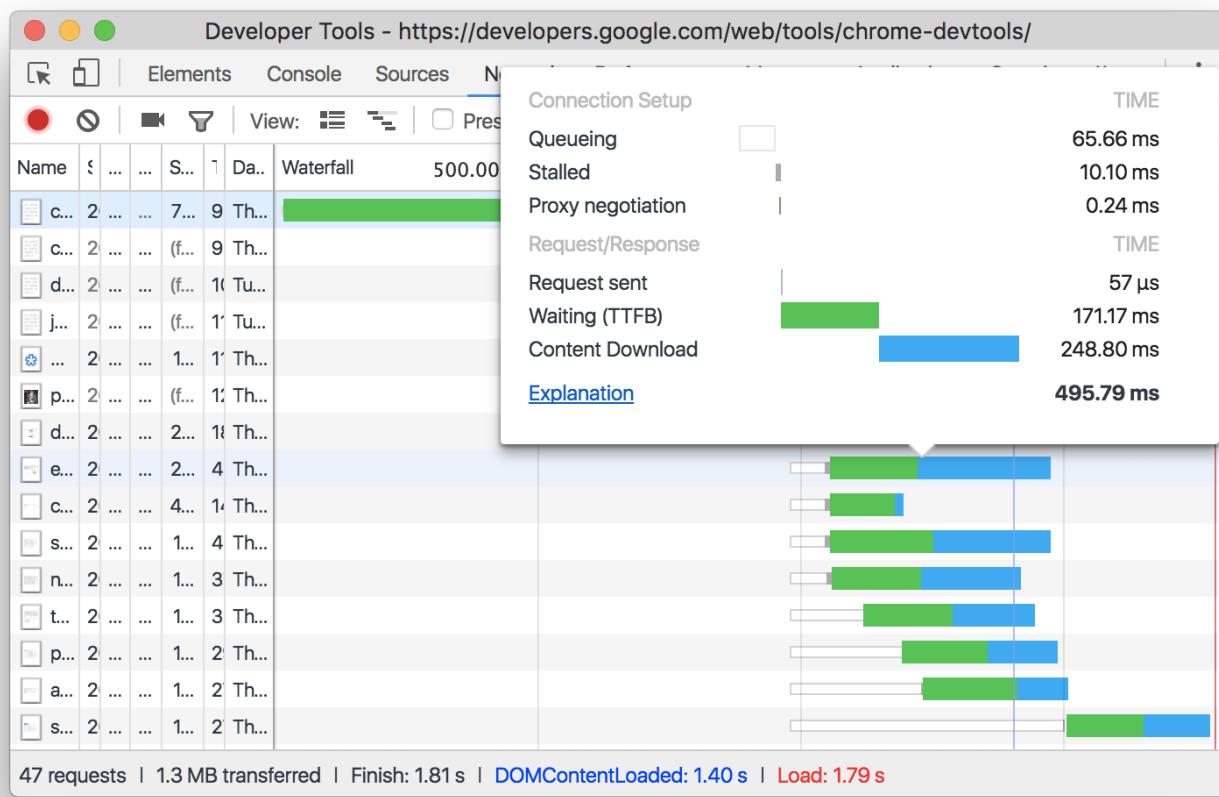
Χρήση του βασικού προγράμματος **telnet** για σύνδεση με έναν Web Server στη θύρα 80.

```
nmac:~ nma$ telnet www.upatras.gr 80
Trying 150.140.130.190...
Connected to gondor.upatras.gr.
Escape character is '^]'.
GET / HTTP/1.1
HOST: www.upatras.gr

HTTP/1.1 301 Moved Permanently
Server: nginx/1.14.0 (Ubuntu)
Date: Thu, 20 Feb 2020 13:52:44 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
Location: http://www.upatras.gr/el
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
```

Άσκηση HTTP

- Μπείτε στο chrome / inspector στην καρτέλα network και παρατηρήστε μηνύματα HTTP που σχετίζονται με φόρτωση της σελίδας www.upatras.gr



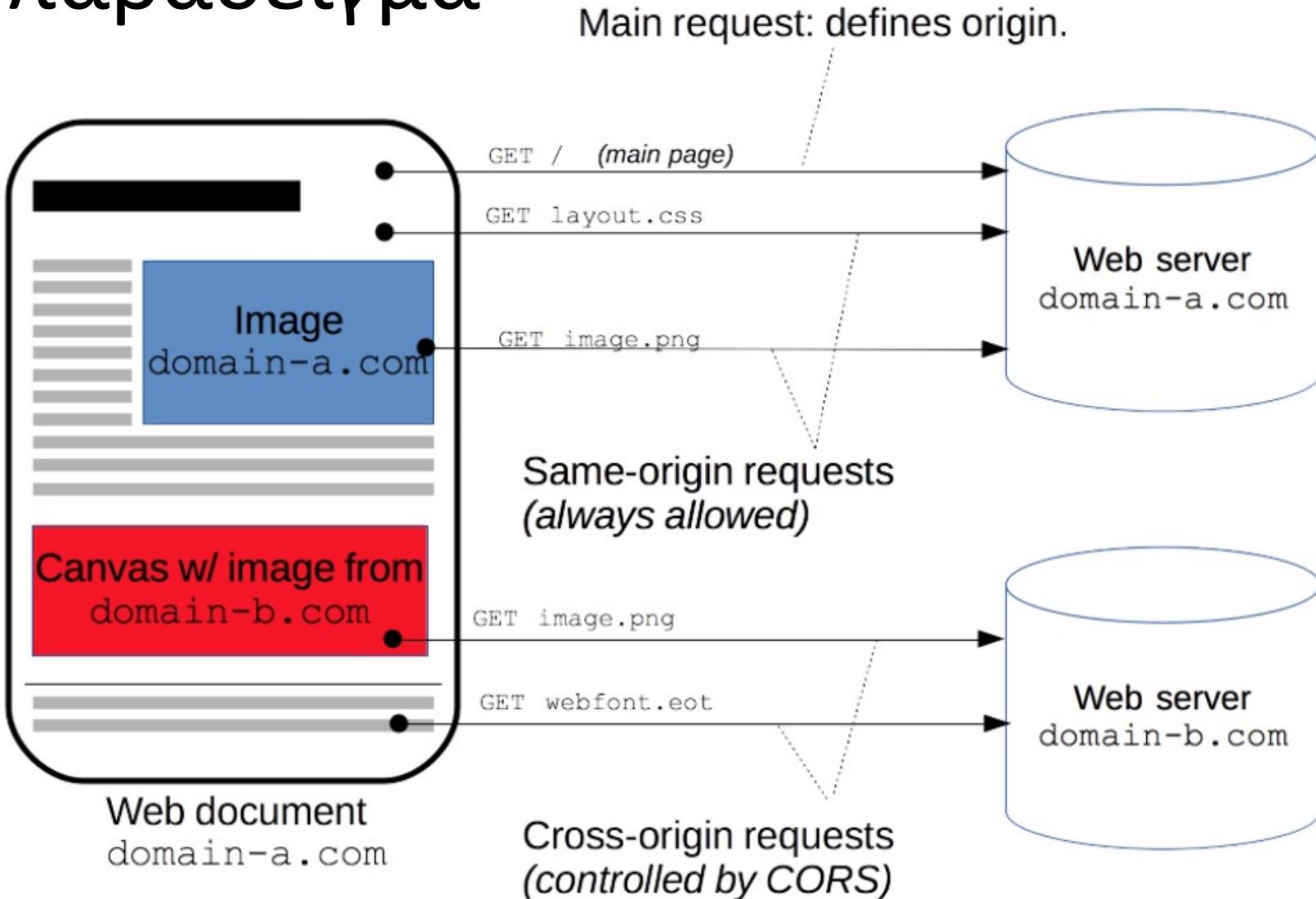
CORS

Cross-origin resource sharing μοίρασμα πόρων διαφορετικής προέλευσης

To **Cross-Origin Resource Sharing (CORS)** είναι ένας μηχανισμός που βασίζεται στις κεφαλίδες του HTTP και ο οποίος επιτρέπει σε έναν εξυπηρετητή να υποδεικνύει τυχόν άλλες διευθύνσεις, πρωτόκολλα ή θύρες, πέραν της δικής του από τις οποίες ένας φυλλομετρητής επιτρέπεται να φορτώσει πόρους.

Για λόγους ασφαλείας, οι φυλλομετρητές περιορίζουν τα αιτήματα HTTP διαφορετικής προέλευσης τα οποία ξεκινούν για παράδειγμα από ένα πρόγραμμα JavaScript μιας ιστοσελίδας.

παράδειγμα



Περιπτώσεις που χρειαζόμαστε πόρους διαφορετικής προέλευσης

- όταν επιθυμούμε να ανακτήσουμε πληροφορίες με fetch API.
- όταν θέλουμε να ανακτήσουμε web safe fonts
- όταν θέλουμε να ανακτήσουμε εικόνες και άλλα πολυμέσα από τρίτους
- όταν θέλουμε σχήματα ή άλλους πόρους

πώς να αποφύγουμε το σφάλμα

[No 'Access-Control-Allow-Origin' header is present on the requested resource—when trying to get data from a REST API](#)

Με χρήση CORS proxy για να αποφύγουμε το πρόβλημα “No Access-Control-Allow-Origin header”

Πώς να αποφύγουμε το CORS
<https://stackoverflow.com/questions/43871637/no-access-control-allow-origin-header-is-present-on-the-requested-resource-whe/43881141#43881141>

will it CORS?

Εργαλείο για να ελέγξουμε αν οι πόροι που θα χρησιμοποιήσουμε θα έχουν πρόβλημα CORS

<https://httptoolkit.tech/will-it-cors/>

εργαστήριο #1

HTTP

Requests

Βιβλιοθήκη υψηλού επιπέδου της python για διαχείριση http requests

πειραματιστείτε με διάφορες
περιπτώσεις αιτημάτων και
αποκρίσεων

<http://docs.python-requests.org>

requests

```
>>> import requests
```

Ανάκτηση ιστοσελίδας με GET με τη βιβλιοθήκη requests

```
>>> r = requests.get('https://api.github.com/events')
```

Η μέθοδος αυτή επιστρέφει ένα αντικείμενο [Response](#) .

Αντίστοιχα αίτημα POST για αποστολή δεδομένων:

```
>>> r = requests.post('https://httpbin.org/post', data = {'key': 'value'})
```

Μια αιτήση PUT της HTTP :

```
>>> r = requests.put('https://httpbin.org/put', data = {'key': 'value'})
```

O header:

```
>>> r.headers
```

Το περιεχόμενο της απόκρισης:

```
>>> r.text
```

μέρος α- οι κεφαλίδες των αποκρίσεων

Σε αρχείο python με όνομα **lab1-1.py** γράψτε κώδικα που:
ζητάει από τον χρήστη ένα URL και επιστρέφει τις κεφαλίδες
(headers) της απόκρισης HTTP (αν έχετε πρόβλημα με την
python, δείτε τον κώδικα της ex1.py)

τροποποιήστε τον κώδικα ώστε να απαντάει για τη URL που
έδωσε ο χρήστης με τις εξής πληροφορίες:

(α) ποιο είναι το λογισμικό που χρησιμοποιεί ο εξυπηρετητής για να
απαντήσει στο αίτημα (o web server);

(β) αν η σελίδα χρησιμοποιεί cookies

Για να δείτε ποιες είναι οι κεφαλίδες που δίνουν τις
πληροφορίες που σας χρειάζονται, ανατρέξτε στις διαφάνειες
της διάλεξης στο [eclass](#) ή στην τεκμηρίωση στο [MDN Web Docs](#).

μέρος β

Η σελίδα <http://ds.upatras.gr/> περιέχει μια φόρμα που δέχεται ένα από τα εξής στοιχεία ενός υπαλλήλου του Πανεπιστημίου: Επίθετο, email ή τηλέφωνο, και επιστρέφει όλα τα στοιχεία του.

Σημείωση: η φόρμα αποστέλει τα στοιχεία με τη μέθοδο **POST** της **HTTP**.

Σε αρχείο **python** με όνομα **lab1-2.py** γράψτε κώδικα που: ζητάει από τον χρήστη το επώνυμο ενός υπαλλήλου και απαντάει με το μέιλ του



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Ονομαστικός
Τηλεφωνικός
Κατάλογος



Η εφαρμογή παρέχει τη δυνατότητα αναζήτησης των πληροφοριών επακοινωνίας Πανεπιστημίου Πατρών. Η αναζήτηση γίνεται με βάση το **επώνυμο**, τη **διαχειρούμενη έδρα** ή το **τηλέφωνο**. Το κλειδί αναζήτησης που θα δοθεί από το μέγεθος **τουλάχιστον τέσσερις (4)** αλφαριθμητικούς χαρακτήρες για το επώνυμο.

Ο τηλεφωνικός κατάλογος των Κεντρικών Υπηρεσιών του Πανεπιστημίου Πατρών [PDF](#).

Επώνυμο:

Email:

Τηλέφωνο:

Παπαδοπούλου Ιωάννα
Τμήμα Γεωλογίας (Διοικητικό προσωπικό)
Τηλ: +302610997925
Email: gianna@upatras.gr
[\(vCard\)](#)

Παπαδόπουλος Γεώργιος
Τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών (Ομότιμος Καθηγητής)
Τηλ: +302610996423 +302610997284 +302610996817
Email: papadopoulos@ece.upatras.gr
[\(vCard\)](#)

μέρος β

Τροποποιήστε τον κώδικα ώστε να τυπώνει επίσης το τηλέφωνο (ή τα τηλέφωνα) του υπαλλήλου, ενώ επίσης να μας γνωρίζει το μέγεθος της απάντησης HTTP.

Δείτε τις οδηγίες για το [πώς μπορείτε να περάσετε παραμέτρους](#) σε αιτήματα HTTP με την Python.

Η βιβλιοθήκη requests θα σας επιστρέψει όλη την ιστοσελίδα της απάντησης, εσείς όμως χρειάζεστε μόνο κάποιες πληροφορίες (μέιλ και τηλέφωνα), οπότε θα χρειαστεί να εξάγετε αυτή την πληροφορία. Προτείνετε να ανατρέξετε σε regular expressions με τη βιβλιοθήκη [re](#) της python.

Εναλλακτική βιβλιοθήκη: urllib.request

- Αποτελεί τμήμα της βασικής διανομής
- Απαιτεί κωδικοποίηση/αποκωδικοποίηση των δεδομένων (για περίπτωση μηνύματος POST)

```
import urllib.request
with urllib.request.urlopen('https://www.upatras.gr')
as response:
    print("\nRESPONSE HEADER")
    for key, value in response.headers.items():
        print(f'{key:30s} {value}')
```

<https://docs.python.org/3/howto/urllib2.html>

Αίτημα POST αποστολής δεδομένων

```
data = urllib.parse.urlencode({'surname' : surname}).encode()
with urllib.request.urlopen(url, data=data) as response:
    the_page = response.read().decode('utf-8')
```