



# Το πρωτόκολλο HTTP/ 1.0

# Το πρωτόκολλο HTTP

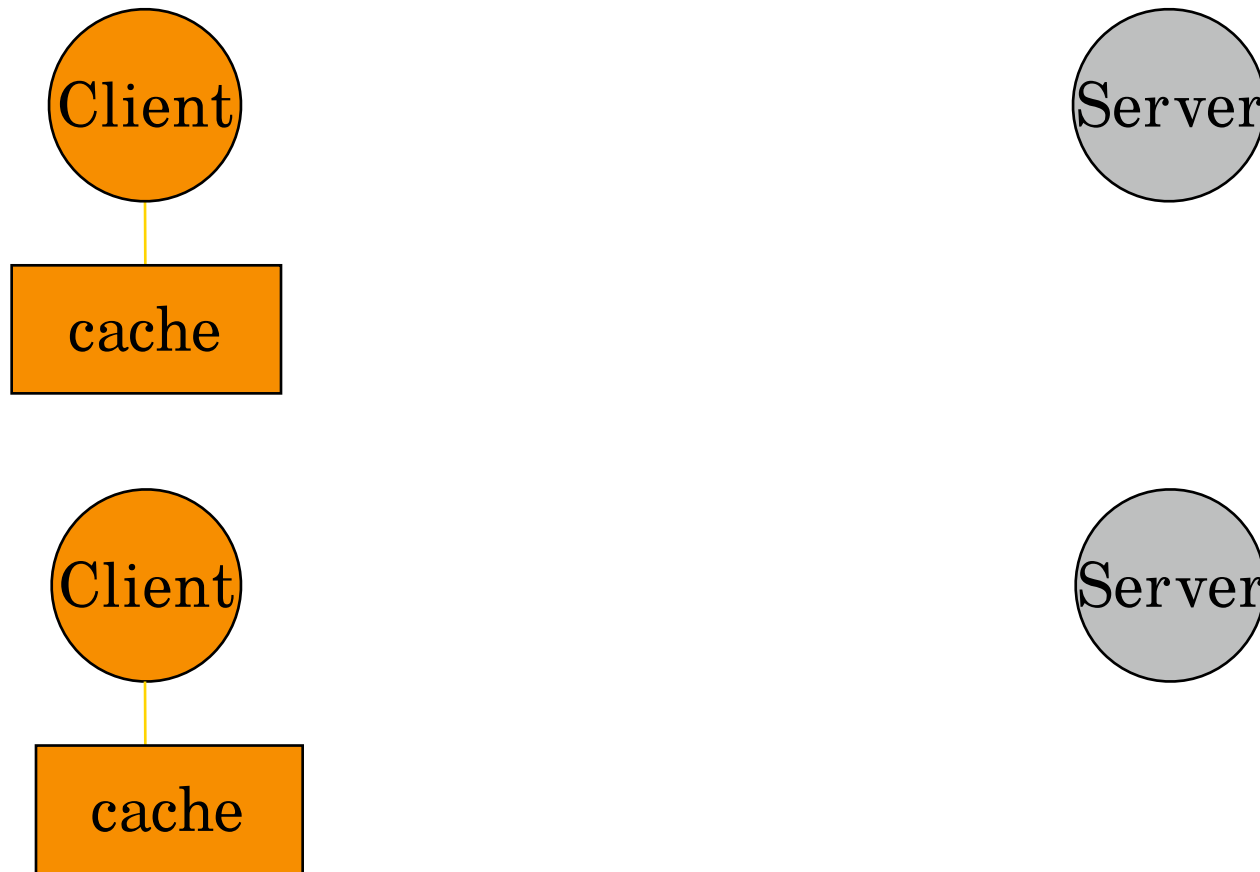


- Το πρωτόκολλο HTTP χρησιμοποιείται για την επικοινωνία φυλλομετρητών (browsers) με Εξυπηρετητές Ιστού (Web Servers).
- Καθορίζει μια απλή “συνομιλία” ανάμεσα σε κάποιο πρόγραμμα-πελάτη Ιστού και σε κάποιο πρόγραμμα εξυπηρετητή Ιστού. Στο πλαίσιο της συνομιλίας αυτής, ο πελάτης αποστέλλει μιά “αίτηση” στον διαθέτη και περιμένει κάποια απάντηση στην αίτηση αυτή.
- Το HTTP καθορίζει τον τρόπο με τον οποίο πρέπει να διατυπωθεί η αίτηση και η απάντηση.
- Το HTTP **δεν** καθορίζει πώς θα επιτευχθεί η δικτυακή σύνδεση πελάτη-διαθέτη και πώς θα γίνει η διαχείριση της επικοινωνίας (ποιός φροντίζει για την επικοινωνία;)
- HTTP 1.0, RFC 1945

# Βασική Αρχιτεκτονική WWW



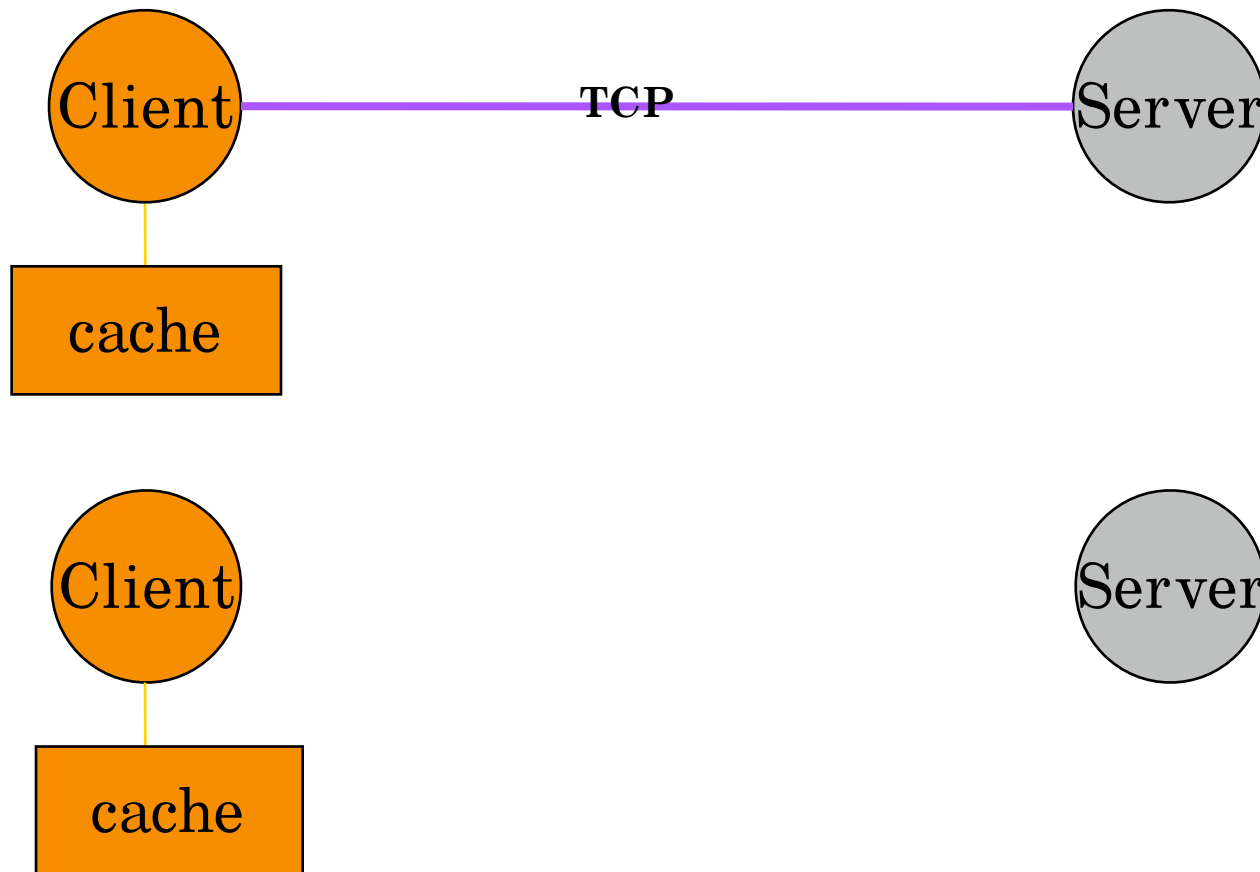
- Μοντέλο HTTP 1.0 (έχει υποκατασταθεί από το HTTP 1.1):



# Βασική Αρχιτεκτονική WWW



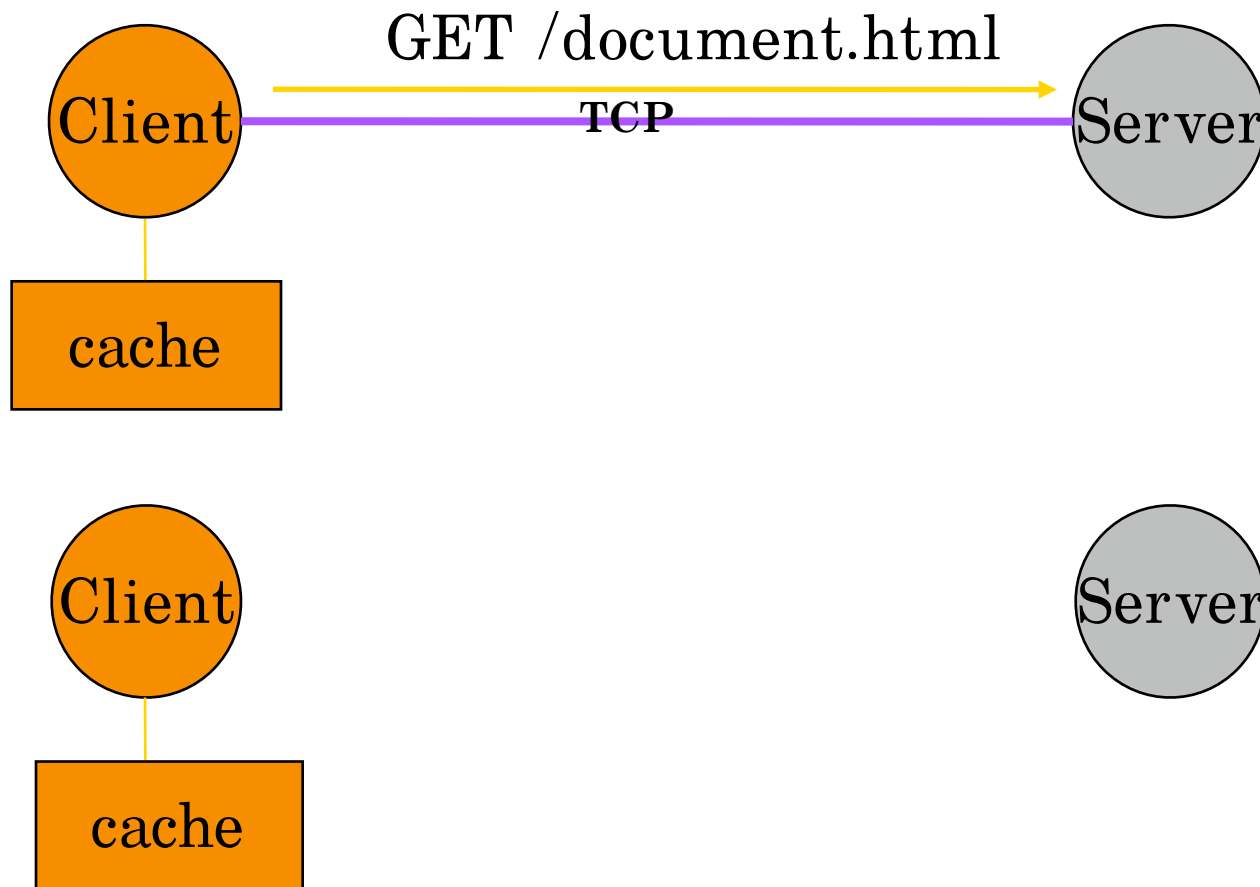
- Μοντέλο HTTP 1.0 (έχει υποκατασταθεί από το HTTP 1.1):



# Βασική Αρχιτεκτονική WWW



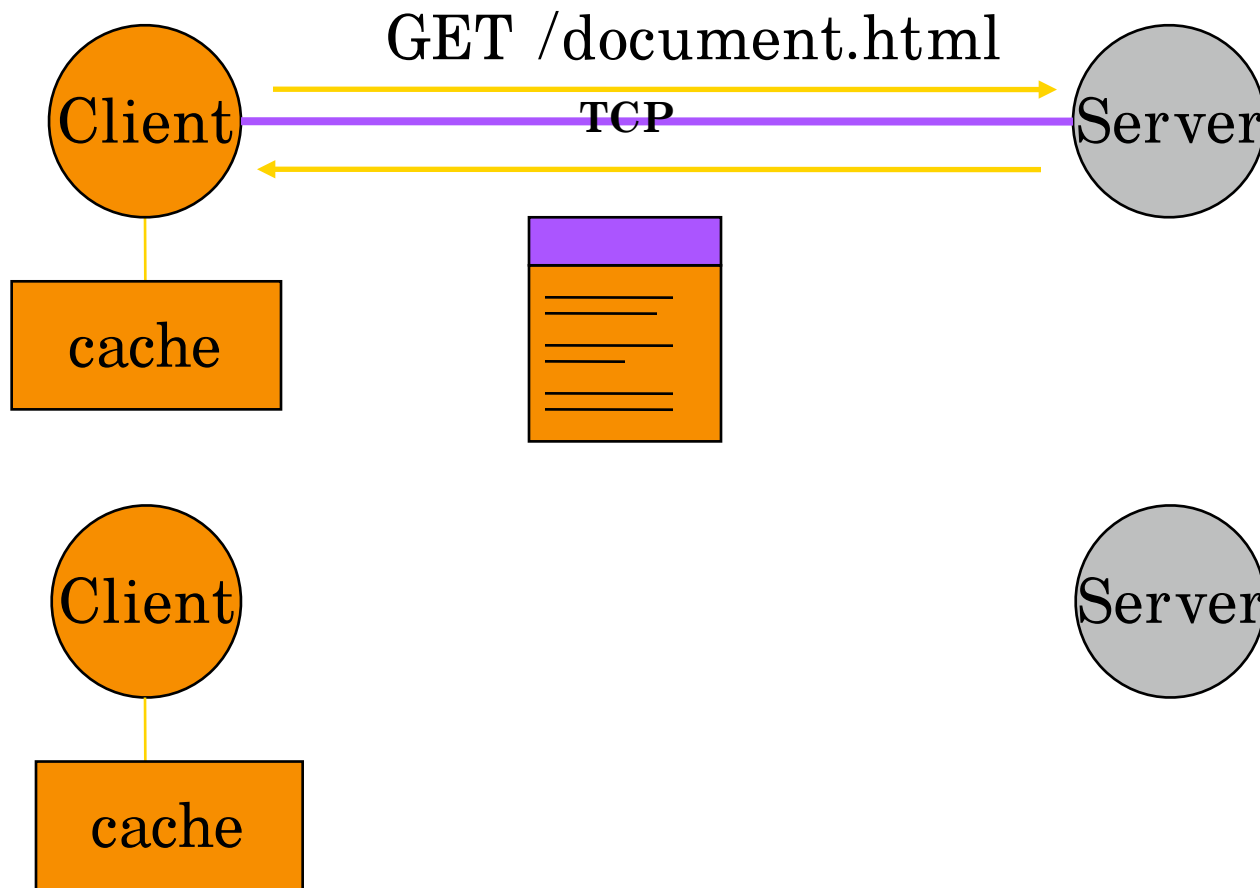
- Μοντέλο HTTP 1.0 (έχει υποκατασταθεί από το HTTP 1.1):



# Βασική Αρχιτεκτονική WWW



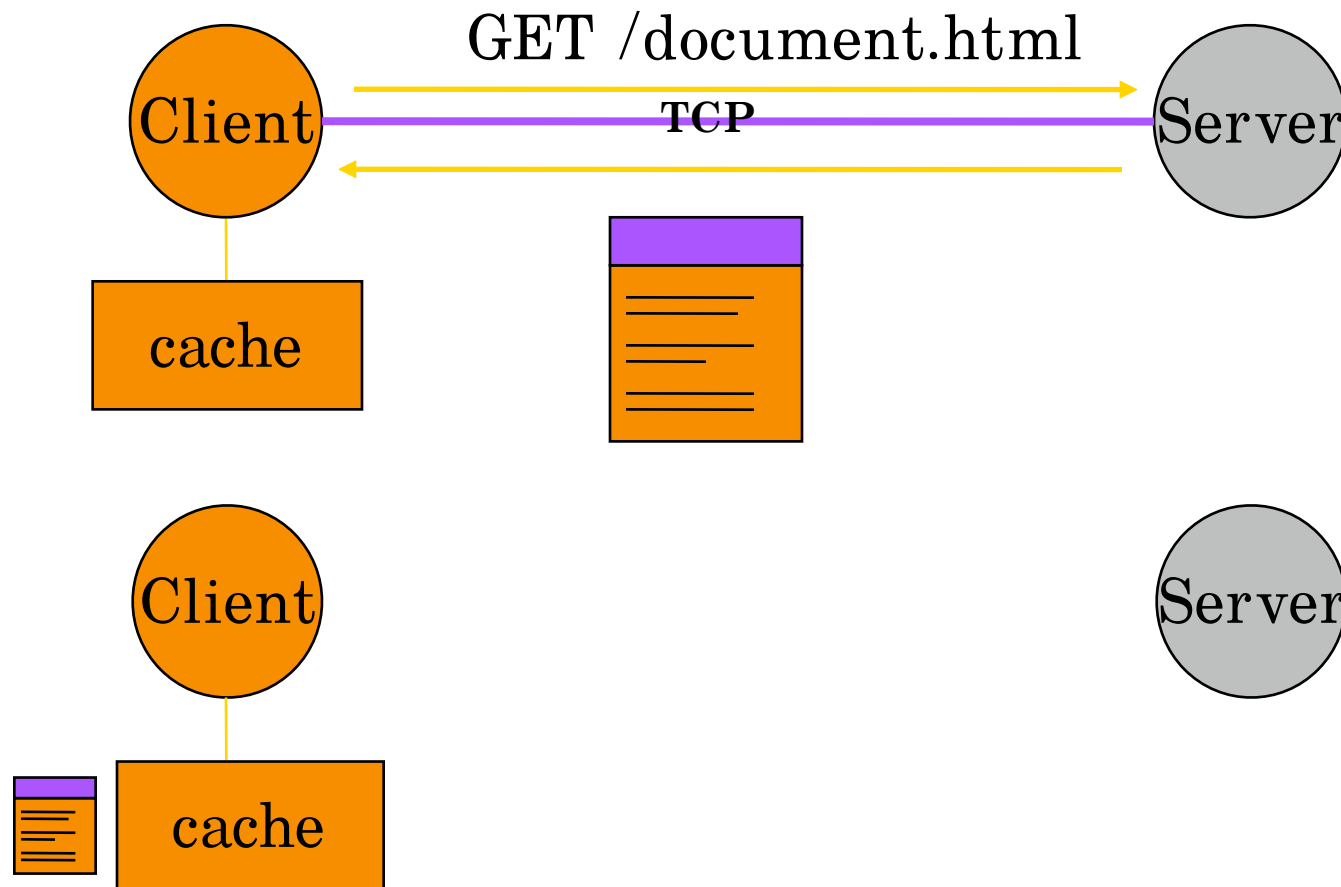
- Μοντέλο HTTP 1.0 (έχει υποκατασταθεί από το HTTP 1.1):



# Βασική Αρχιτεκτονική WWW



- Μοντέλο HTTP 1.0 (έχει υποκατασταθεί από το HTTP 1.1):



# Τι είναι το HTTP;



- **HyperText Transfer Protocol:**
  - Πρωτόκολλο αιτήματος-απάντησης.
  - Πρωτόκολλο εφαρμογών (όπως τα ftp και telnet).
  - Επιλήσμον (σε αντίθεση με τα ftp και telnet).
- **Ποιά η διαφορά HTTP και WWW;**
  - WWW: HTTP + HTML + URI
  - WWW: browsers, servers, proxies, gateways
- **Πως χρησιμοποιείται το HTTP;**
  - Για την μεταφορά πληροφοριών σε πολλαπλούς μορφότυπους, γλώσσες και συμβολοσειρές.
  - Η σύνταξη των μηνυμάτων HTTP βασίζεται στο MIME.
  - Τα περιεχόμενα των μηνυμάτων HTTP δεν αφορούν το πρωτόκολλο.



# HTTP: Ιστορική αναδρομή



3/1990	Το CERN προτείνει το Web
1/1992	Προδιαγραφή HTTP/0.9
12/1992	Πρόταση για πρόσθεση του MIME στο HTTP
3/1993	Πρώτο προσχέδιο HTTP/1.0
6/1993	HTML (προδιαγραφή 1.0)
10/1993	Προδιαγραφή URL
11/1993	Δεύτερο προσχέδιο HTTP/1.0
3/1994	Υιοθέτηση URI στο WWW
5/1996	HTTP/1.0 RFC 1945
1/1997	Εισήγηση HTTP/1.1, RFC 2068
6/1999	Προσχέδιο προδιαγραφής HTTP/1.1, RFC 2616
2001	HTTP/1.1 επίσημη προδιαγραφή

# Βασικές ιδιότητες του HTTP



- **Παγκόσμιο URI:** το HTTP βασίζεται στον μηχανισμό διευθυνσιοδότησης πληροφοριακών πόρων που παρέχεται από το URI.
- **Ανταλλαγή αιτήματος-απάντησης:** τα αιτήματα αποστέλλονται από συστήματα πελατών και προκαλούν απαντήσεις από τους εξυπηρετητές. Δοσοληψίες HTTP δεν ενεργοποιούνται από τους εξυπηρετητές.
- **Επιλήσμον (statelessness):**
  - Πελάτες και εξυπηρετητές δεν διατηρούν στοιχεία για την αλληλεπίδρασή τους μεταξύ διαδοχικών ανταλλαγών αιτημάτων-απαντήσεων (λόγοι επίδοσης).
  - Στην πράξη διατηρούν πληροφορίες κατάστασης, αλλά αυτό δεν αφορά το πρωτόκολλο.
- **Μεταδεδομένα πληροφοριακών πόρων:**
  - Πληροφορίες σχετικές με τους πόρους του WWW συχνά συμπεριλαμβάνονται στα μηνύματα του HTTP.
  - Π.χ. το μέγεθος ενός αρχείου, ο μορφότυπος του περιεχομένου του, η ημερομηνία τελευταίας ενημέρωσής του κ.ο.κ.

# Ορολογία HTTP



- HTTP Μήνυματα (messages)
- Οντότητες (entities)
- Πόροι (resources)
- Διεκπεραιωτής χρήστη (user agent)

# Μηνύματα HTTP



- Σειρά οκταδικών ψηφίων που στέλνονται μέσω TCP/IP.
- Διακρίνονται σε *αιτήματα (requests)* και *απαντήσεις (replies)*.
- Μηνύματα αιτημάτων αρχίζουν με *γραμμή αιτήματος*.
- Μηνύματα απαντήσεων αρχίζουν με *γραμμή κατάστασης*.
- Κάθε μήνυμα στο HTTP περιλαμβάνει μηδέν ή περισσότερες επικεφαλίδες, οι οποίες διαχωρίζονται από το προαιρετικό σώμα του μηνύματος με **CR** ή **LF**.
- Συντακτική δομή αιτημάτων:
  - Request-Line**
  - General/Request/Entity Header(s)**
  - CRLF**
  - Optional Message Body**

# Αίτημα πελάτη HTTP (παράδειγμα)



**GET /motd HTTP/1.0**

**Date: Wed, 22 Mar 2000 08:09:01 GMT**

**Pragma: No-cache**

**From: mdd@ucy.ac.cy**

**User-Agent: Mozilla/4.03**

**CRLF**

- Γραμμή αιτήματος
- Γενικές επικεφαλίδες: μπορούν να υπάρχουν σε μηνύματα αιτήματος και απάντησης.
- Επικεφαλίδες αιτήματος

# Αίτημα πελάτη HTTP (παράδειγμα)



**POST /motd HTTP/1.0**

**Date: Wed, 22 Mar 2000 08:09:01 GMT**

**Pragma: No-cache**

**From: mdd@ucy.ac.cy**

**User-Agent: Mozilla/4.03**

**Content-Length: 23**

**Allow: GET, HEAD, PUT**

**CRLF**

**Welcome to Comer's VAX**

- **Επικεφαλίδες οντότητας**
  - Content-Length: καθορίζει το μήκος της οντότητας που μεταφέρει το μήνυμα.
  - Allow: καθορίζει τις εντολές που μπορούν να εκτελεσθούν στον πόρο motd

# Απαντήσεις HTTP



- **Συντακτική δομή απαντήσεων HTTP:**
  - Status-Line**
  - General/Request/Entity Header(s)**
  - CRLF**
  - Optional Message Body**
- **Π.χ.:**
  - HTTP/1.0 200 OK**
  - Date: Tue, 4 Jul, 1996, 19:57:05 GMT**
  - Server: Netscape-Enterprise/3.5.1**
  - Content-length: 23**
  - CRLF**
  - Welcome to Comer's VAX**

# Οντότητες HTTP



- Καθορίζονται ως οι πόροι που περικλείονται σε μηνύματα αιτημάτων ή αποκρίσεων HTTP.
- Αποτελούνται από:
  - Επικεφαλίδες οντότητας
  - Προαιρετικό περιεχόμενο οντότητας (entity body)
- Αποτελούν το πιο σημαντικό τμήμα ενός μηνύματος HTTP, αφού μεταφέρουν το περιεχόμενο του μηνύματος.



# Διεκπεραιωτής Χρήστη (User Agent)



- Αποτελεί το σύστημα πελάτη (client) το οποίο *αρχικοποιεί* ένα αίτημα προς κάποιον εξυπηρετητή.
- Μπορεί να είναι:
  - Φυλλομετρητής
  - Αράχνη (crawler)
  - Άλλο σύστημα (προσωπικοί διεκπεραιωτές, απομνημονευτές Ιστού κλπ)
- Παραδείγματα UA:
  - Mozilla/3.01 (X11; SunOS 5.5 sun4m)
  - Mozillas/2.0 (MSIE 2.1; AOL 3.0; Mac)

# Μέθοδοι HTTP/1.0



- Οι μέθοδοι αιτημάτων του HTTP προσδιορίζουν τη δράση που πρέπει να εκτελεσθεί στον πληροφοριακό πόρο ο οποίος καθορίζεται από το URI της γραμμής αιτήματος.
- Το HTTP/1.0 καθορίζει μόνο τρεις μεθόδους:
  - GET
  - HEAD
  - POST
- Ορισμένες εκδόσεις πελατών και εξυπηρετητών υλοποιούν και άλλες μεθόδους, όπως PUT, DELETE, LINK και UNLINK.
- Ο εξυπηρετητής που δέχεται το αίτημα, εκτελεί την περιγραφόμενη δράση.

# Μέθοδοι HTTP/1.0: GET



# Μέθοδοι HTTP/1.0: GET



## GET

- Αναζήτησε και επέστρεψε πόρο που καθορίζεται από το URI
- Το URI μπορεί να περιλαμβάνει και ορίσματα (arguments) για την υποβολή δεδομένων εισροής σε πρόγραμμα πύλης (gateway):  
`GET /foo/bar.html?x=bar&bam=baz`
- Ένα αίτημα GET μπορεί να συνοδεύεται από **τροποποιητική επικεφαλίδα**, η οποία να το μετατρέπει σε conditional GET. Π.χ.:  
`GET /foo/bar.html HTTP/1.0`  
`If-Modified-Since: Sun, 12 Nov 2000 11:12:23 GMT`

# Μέθοδοι HTTP/1.0: HEAD



- Παρόμοια με την GET, αλλά ζητά την επιστροφή μόνο μεταδεδομένων για κάποιον πόρο. Ζητάει από έναν εξυπηρετητή να στείλει τις επικεφαλίδες απάντησης για κάποιο αρχείο στο διαθέτη , αλλά όχι και τα περιεχόμενα του αρχείου.
- Χρησιμοποιείται από προγράμματα τα οποία εκτελούν αυτόματες αναζητήσεις για κείμενα στον Ιστό.
- Η μέθοδος αυτή είναι ένας γρήγορος τρόπος για να βλέπει κανείς αν ένα αρχείο ή πρόγραμμα gateway είναι πράγματι παρόν στον Ιστό, για την εύρεση γενικών πληροφοριών σχετικά με το αρχείο (τον MIME τύπο του, την ημερομηνία αλλαγής του κλπ).

# Παράδειγμα αιτήματος HEAD



**HEAD /foo.html HTTP/1.0**

**HTTP/1.0 200 OK**

**Content-Length: 3219**

**Last-Modified: Sun, 12 Nov 2000 11:12:23 GMT**

**Content-Type: text/html**

# Μέθοδοι HTTP/1.0: POST



- Σε αντίθεση με τις GET και HEAD, οι οποίες ζητούν την ανάκτηση πληροφοριών, η POST χρησιμοποιείται για την ενημέρωση ενός πόρου ή την παροχή εισροής σε κάποιο πρόγραμμα επεξεργασίας στον εξυπηρετητή.
- Το «σώμα» μηνυμάτων POST περιλαμβάνει τα δεδομένα που στέλνονται από τον πελάτη στον εξυπηρετητή.
- Δεν είναι ούτε ασφαλής μέθοδος, αφού μπορεί να αλλάξει τα περιεχόμενα ενός πληροφοριακού πόρου, ούτε μηδενοδύναμη.

```
POST /foo/bar.cfm HTTP/1.0
Content-Length: 143
<entity body>
```

-- υποχρεωτική χρήση εδώ

```
POST /search.cgi HTTP/1.0
Content-Length: 34
CRLF
Query iktinos
Db greek-architecture
```

# Μέθοδοι HTTP/1.0



## PUT, DELETE, LINK, UNLINK

- not really used - big security issues if not careful
- Δεν συμπεριλαμβάνονται στην προδιαγραφή HTTP/1.0, εκτός από το PUT, το οποίο αναφέρεται σε παράρτημα του RFC 1945



# Επικεφαλίδες HTTP/1.0



- Αποτελούνται από χαρακτήρες ASCII, περιλαμβάνοντας κάποιο όνομα και συνήθως μια τιμή, που ανατίθεται στο όνομα.
- Παίζουν σημαντικό ρόλο στο HTTP, καθώς αποτελούν το βασικό μηχανισμό για τροποποίηση της διαχείρισης των αιτημάτων του. Επιτρέπεται η εισαγωγή νέων επικεφαλίδων, αυθαίρετου μεγέθους.
- Χρήσεις:
  - Παροχή μεταδεδομένων για κάποιο πόρο.
  - Παραμετροποίηση αιτημάτων ή απαντήσεων.
  - Καθορισμός αν μια απάντηση μπορεί να απομνημονευθεί, ή πως μπορεί να αποκωδικοποιηθεί.
  - Επέκταση του πρωτοκόλλου

# Χρήση επικεφαλίδων στο HTTP/1.0



- Ένα μήνυμα HTTP μπορεί να έχει αυθαίρετο αριθμό επικεφαλίδων – καθεμιά από τις οποίες ακολουθείται από CR και LF χαρακτήρες για διαχωρισμό της με τις επόμενες.
- Οι περισσότερες επικεφαλίδες είναι προαιρετικής χρήσης. Δομοστοιχεία του Ιστού μπορούν να αγνοούν τις προαιρετικές επικεφαλίδες.
- Ορισμένες επικεφαλίδες είναι υποχρεωτικές για κάποια μηνύματα αιτήματος-απόκρισης.
- Συντακτικό επικεφαλίδων:

**<όνομα>: <τιμή> (,<τιμή>)\***

**Π.χ.           Date: Thu, 23 Dec 1999 08:12:31 GMT**  
**Accept-Language: de-CH, en-US**

# Ιεραρχία επικεφαλίδων HTTP



- **Γενικές:** χρήση σε αιτήματα και απαντήσεις
- **Αιτημάτων:**
  - χρησιμοποιούνται σε αιτήματα για εκδήλωση προτίμησης στη μορφή της απάντησης, για παροχή επιπρόσθετης πληροφορίας με το αίτημα ή για καθορισμό κάποιου περιορισμού στον εξυπηρετητή που θα εκτελέσει το αίτημα.
- **Απαντήσεων:**
  - χρήση σε μηνύματα αποκρίσεων για παροχή περαιτέρω πληροφοριών ή μεταπληροφοριών προς τον πελάτη, ή για έγερση αιτήματος για περισσότερες πληροφορίες από τον πελάτη.
- **Οντότητας:** παρέχουν πληροφορίες για την οντότητα την οποία αφορά το μήνυμα.
- Η σειρά με την οποία χρησιμοποιούνται οι επικεφαλίδες σε κάποιο μήνυμα δεν είναι σημαντική – συνήθως μπαίνουν πρώτα οι γενικές επικεφαλίδες, ακολουθούν οι επικεφαλίδες αιτημάτων ή απαντήσεων και στο τέλος έπονται οι επικεφαλίδες οντοτήτων. 22

# Γενικές Επικεφαλίδες



- Αφορούν στο μήνυμα HTTP και όχι στο περιεχόμενό του (δηλ. την οντότητα που μεταφέρει).
- Το HTTP/1.0 καθορίζει δύο μόνο γενικές επικεφαλίδες:
  - **Date:** προσδιορίζει την ημερομηνία δημιουργίας του μηνύματος.
  - **Pragma:** επιτρέπει την αποστολή οδηγιών προς τον παραλήπτη του μηνύματος. Οι οδηγίες αυτές αποτελούν ένα τρόπο για να ζητείται από δομοστοιχεία τού Ιστού να συμπεριφέρονται με συγκεκριμένο τρόπο, ενώ διαχειρίζονται ένα αίτημα ή μια απόκριση.
  - Το πρωτόκολλο δεν επιβάλλει την υπακοή στις οδηγίες αυτές, αλλά απλώς απαιτεί την μετάδοσή τους.
  - Το HTTP/1.0 καθορίζει μόνο ένα **pragma: no-cache**

# Επικεφαλίδες Αιτημάτων



- **Authorization:** Χρησιμοποιείται από τον διεκπεραιωτή-πελάτη για την υποβολή των διαπιστευτηρίων (credentials) που χρησιμοποιούνται για την πρόσβαση σε κάποιο πόρο. Π.χ.:
  - **Authorization: Basic YXZpYZRpS29IDizM1NA==**
  - Η λέξη Basic αναφέρεται σε βασικό σχήμα ταυτοποίησης με ταυτότητα χρήστη (userid) και κωδικό (password), κωδικοποιημένα σε Base64 μορφή.
- **From:** επιτρέπει στον χρήστη να παρέχει το ηλε/τ του, για την ταυτοποίησή του (κυρίως για χρήστες-αράχνες).

# Επικεφαλίδες Αιτημάτων (συνέχεια)



- **If-Modified-Since:** παράδειγμα επικεφαλίδας συνθήκης, η οποία επηρεάζει τον τρόπο με τον οποίο θα τύχει επεξεργασίας ένα αίτημα.
  - Π.χ.:  
**GET /foo.html HTTP/1.0**  
**If-Modified-Since: Sun, 21 May 2000 07:00:25 GMT**
- **Referer:** επιτρέπει στον πελάτη να συμπεριλάβει σε ένα αίτημα το URI του πόρου από τον οποίο ελήφθη το αιτούμενο URI του συγκεκριμένου μηνύματος.
  - Π.χ.:  
**GET /Disasters/worldglance.html HTTP/1.0**  
**Referer: http://www.cnn.com**
    - Συνέπειες: κίνδυνοι για την αποκάλυψη ιδιωτικών δεδομένων.

# Η χρήση της τοπικής cache



- Στην περίπτωση που το ζητούμενο αρχείο δεν έχει τροποποιηθεί μετά την ημερομηνία που καθορίζει ο πελάτης, ο εξυπηρετητής επιστρέφει το ακόλουθο:

**HTTP/1.0 304 Not Modified**

**Date: Thu, 03 Aug 1996 16:04:09 GMT**

**Server: NCSA/1..3**

**MIME-version: 1.0**

- Τότε ο φυλομετρητής-πελάτης μπορεί να χρησιμοποιήσει την “κρυμμένη” έκδοση του αρχείου που αναζήτησε.

# Επικεφαλίδες Αιτημάτων (συνέχεια)



- **User-Agent:** παρέχει πληροφορίες για την έκδοση του φυλλομετρητή που χρησιμοποιεί το σύστημα πελάτη, σε τι λειτουργικό σύστημα τρέχει κοκ.
- Π.χ.:  
**User-Agent: Mozilla/4.03 (Macintosh; I; 68K, Nav)**



# Επικεφαλίδες Απαντήσεων



- **Location:** για παραπομπή ενός αιτήματος σε άλλο URI.
- **Server:** καθορίζει τον τύπο του χρησιμοποιούμενου εξυπηρετητή.
- **WWW-Authenticate:** χρησιμοποιείται για την έγερση από τον εξυπηρετητή απαίτησης προς τον πελάτη για αποστολή διαπιστευτηρίων με σκοπό την παροχή πρόσβασης σε ελεγχόμενο πόρο. Χρησιμοποιείται μαζί με απάντηση **404 Unauthorized**. Π.χ.:  
**WWW-Authenticate: Basic realm="ChaseChem"**

# Επικεφαλίδες Απαντήσεων (συνέχεια)



# Επικεφαλίδες Απαντήσεων (συνέχεια)



- Χρησιμοποιούνται, χωρίς να ορίζονται στο πρωτόκολλο:

# Επικεφαλίδες Απαντήσεων (συνέχεια)



- Χρησιμοποιούνται, χωρίς να ορίζονται στο πρωτόκολλο:
  - **Retry-After: (date) or (seconds)**
  - **[Set-]Cookie:**  
**Part\_Number="Rocket\_Launcher\_0001";**  
**Version="1"; Path="/acme"**
  - **Title: (title)**

# Επικεφαλίδες Οντοτήτων



- **Allow:** χρησιμοποιείται για να προσδιορίσει τις έγκυρες μεθόδους που μπορούν να εφαρμοσθούν σε κάποιον πόρο.
  - Π.χ. αν κάποιος πηγαίος εξυπηρετητής λάβει ένα αίτημα για μη εξουσιοδοτημένη μέθοδο, μπορεί να απαντήσει ενημερώνοντας τον πελάτη για τις μεθόδους που είναι αποδεκτές για το συγκεκριμένο αίτημα.
  - Σε μήνυμα αιτήματος PUT, με το **Allow** ο πελάτης μπορεί να καθορίσει τι μεθόδους επιτρέπει να εκτελούνται στον πόρο που δημιουργεί:  
**PUT /foo.html HTTP/1.0**  
**Allow: HEAD, GET, PUT**
- **Content-Type:** προσδιορίζει τον μορφότυπο περιεχόμενου (media type) της οντότητας. Π.χ.:  
**POST /chat/chatroom.cgi HTTP/1.0**  
**User-Agent: Mozilla/3.0C**  
**Content-Type: application/x-www-form-urlencoded**

# Επικεφαλίδες Οντοτήτων (συνέχεια)



- **Content-Encoding:** προσδιορίζει πως έχει κωδικοποιηθεί το περιεχόμενο της οντότητας και πως θα πρέπει να αποκωδικοποιηθεί για να πάρουμε την οντότητα στο μορφότυπο που καθορίζεται από το **Content-Type**. Π.χ.: **Content-Encoding: x-gzip**
- **Content-Length:**
  - Προσδιορίζει το μέγεθος της μεταφερόμενης οντότητας σε χαρακτήρες.
  - Χρησιμοποιείται για να επικυρώνουμε αν μια οντότητα έχει φθάσει στην ολότητά της, και σαν επικύρωση ότι μια απομνημονευμένη οντότητα δεν διαφέρει από την οντότητα στην τρέχουσα μορφή της.
  - Αποφεύγεται η χρήση της με την μεταφορά δυναμικού περιεχομένου, για λόγους επίδοσης.

# Επικεφαλίδες Οντοτήτων (συνέχεια)



- **Expires:** χρησιμοποιείται για τον προσδιορισμό της χρονικής στιγμής μετά από την οποία ένας πόρος μπορεί να θεωρηθεί ως λήξας. Π.χ.:  
**HTTP/1.0 200 OK**  
**Server: Microsoft-IIS/4.0**  
**Date: Mon, 04 Dec 2000 18:16:45 GMT**  
**Expires: Tue, 05 Dec 2000 16:00:00 GMT**
- **Last-Modified:** καθορίζει τη χρονική στιγμή της τελευταίας ενημέρωσης του μεταφερόμενου πόρου.

# Κλάσεις Απαντήσεων HTTP/1.0



## Οικογένεια κωδικών με πέντε «τύπους»:

- 1xx: informational
- 2xx: successful, e.g. 200 OK
- 3xx: redirection
  - 301 Moved Permanently
  - 304 Not Modified
- 4xx: Client Error
  - 400 Bad Request
  - 401 Unauthorized
  - 403 Forbidden
  - 404 Not Found
- 5xx: Server Error
  - 501 Not Implemented      503 Service Unavailable



# Η ιστορία ενός κλίκ



- Υποθέσατε:
  - Είστε στο: [www.yahoo.com/index.html](http://www.yahoo.com/index.html)
  - Επιλέγετε το: [www.grippy.org/mattmarg/](http://www.grippy.org/mattmarg/)
- Ο πλοηγός χρησιμοποιεί το DNS => IP addr για [www.grippy.org](http://www.grippy.org)
- Ανοίγει σύνδεση TCP με την IP διεύθυνση του [www.grippy.org](http://www.grippy.org)
- Στέλνει το ακόλουθο αίτημα:

**get /mattmarg/ HTTP/1.0**

**User-Agent: Mozilla/2.0 (Macintosh; I; PPC)**

**Accept: text/html; \*/\***

**Cookie: name = value**

**Referer: http://www.yahoo.com/index.html**

**Host: www.grippy.org**

**Expires: ...**

**If-modified-since: ...**

Request

Request  
Headers

# Απόκριση HTTP



	Status
<b>HTTP/1.0 200 Found</b>	Response
<b>Date: Mon, 10 Feb 1997 23:48:22 GMT</b>	1st header
<b>Server: Apache/1.1.1 HotWired/1.0</b>	
<b>Content-type: text/html</b>	<i>Image/jpeg, ...</i>
<b>Last-Modified: Tues, 11 Feb 1999 22:45:55 GMT</b>	

- One click => several responses
- HTTP1.0: new TCP connection for each page element
- HTTP1.1: KeepAlive - several requests/connection

# Σύνδεση HTTP 1.0



- Μία σύνδεση HTTP αποτελείται από τα ακόλουθα **τέσσερα** στάδια:
  1. **Έναρξη της σύνδεσης:** ο πελάτης επικοινωνεί με τον εξυπηρετητή σε ένα URL και σε αριθμό πύλης (port number), τα οποία καθορίζονται από το URL (κατά συνθήκη τιμή του αριθμού πύλης το 80).
  2. **Πραγματοποίηση της σύνδεσης:** ο πελάτης στέλνει ένα μήνυμα στον εξυπηρετητή, ζητώντας εξυπηρέτηση και καθορίζοντας την HTTP-μέθοδο της δοσοληψίας (συνήθως **GET** ή **POST**).
  3. **Αποστολή απάντησης:** επικεφαλίδα (περιγραφή κατάστασης της δοσοληψίας, περιγραφή δεδομένων) και δεδομένα.
  4. **Κλείσιμο της σύνδεσης.**

# Ενσωμάτωση Εικόνων



- Ένα υπερκείμενο στον Ιστό αποθηκεύεται συνήθως σε περισσότερα του ενός αρχεία (ίσως και κόμβους): το κείμενο σε ένα αρχείο και κάθε εικόνα σε διαφορετικό αρχείο.
- Ωστόσο, όταν επιλέξουμε να ανοίξουμε ένα αρχείο, στο παράθυρο του φυλλομετρητή μας εμφανίζεται ολόκληρη η πληροφορία μαζί, μορφοποιημένη με τον κατάλληλο τρόπο.
- Η αναζήτηση και εμφάνιση ολόκληρης της πληροφορίας γίνεται υπ' ευθύνη του πελάτη-φυλλομετρητή (πώς;). Ο διαθέτης «βλέπει» απλά μιά σειρά από αιτήματα, πρώτα για το αρχείο HTML και στη συνέχεια για κάθε ενσωματωμένη εικόνα.
- Η αυτόματη αναζήτηση των ενσωματωμένων εικόνων, ωστόσο, έχει συνέπειες για τον διαθέτη (υπερφόρτωση).<sup>37</sup>



- **Access Control (Έλεγχος Πρόσβασης)**

- Για πολλές εφαρμογές του Ιστού είναι χρήσιμο να υπάρχει έλεγχος πρόσβασης σε συγκεκριμένα αντικείμενα.
- Η υλοποίηση του ελέγχου πρόσβασης προϋποθέτει την επέκταση των εξυπηρετητών ΠΠΠ με τρεις μηχανισμούς:
  - Μηχανισμό καθορισμού του ποιός έχει δικαίωμα πρόσβασης σε ποιές πληροφορίες.
  - Μηχανισμό προσδιορισμού ταυτότητας πελάτη (authentication - επαλήθευση).
  - Μηχανισμό επιβολής των κανόνων.

- **Πρόσβαση σε άλλα πρωτόκολλα: FTP, Gopher, WAIS**

- Γίνεται με την ευθύνη του προγράμματος πελάτη και χωρίς διαμεσολάβηση εξυπηρετητών Ιστού ή του πρωτοκόλλου HTTP.

# Ταυτόχρονη Εξυπηρέτηση Πολλών Αιτημάτων



- Οι εξυπηρετητές Ιστού είναι σχεδιασμένοι ώστε να μπορούν να εξυπηρετήσουν ταυτόχρονα, πολλές ταυτόχρονες αιτήσεις HTTP.
- Αυτό επιτυγχάνεται με διάφορους τρόπους:
  - Με κλωνοποίηση ενός νέου αντιγράφου του httpd προγράμματος για κάθε νέα αίτηση. Κάθε κλώνος του httpd διαχειρίζεται ακριβώς μιά αίτηση.
  - Με χρήση πολυνηματικής εκτέλεσης (multithreading): το πρόγραμμα httpd έχει πολλά νήματα εκτέλεσης. Κάθε νήμα διαχειρίζεται μια αίτηση. Το πρόγραμμα δίνει τον έλεγχο σε άλλο νήμα όταν το εκτελούμενο νήμα πρέπει να περιμένει κάποια εξυπηρέτηση από το Λειτουργικό Σύστημα.
  - Με κατανομή της εργασίας σε πολλά υποβοηθητικά προγράμματα: το httpd απλά δέχεται αιτήσεις και τις κατανέμει σε άλλα προγράμματα για περαιτέρω εξυπηρέτηση.

# Πολλαπλοί Εξυπηρετητές σε έναν κόμβο



- Σε έναν κόμβο του Διαδικτύου μπορούν να τρέχουν περισσότεροι του ενός εξυπηρετητές Ιστού, εφόσον χρησιμοποιούν διαφορετική πύλη επικοινωνίας (port).
- Η κατά συνθήκη πύλη επικοινωνίας μέσω HTTP είναι η 80, ενώ πύλες διαθέσιμες στους χρήστες αριθμούν από 1025 και πάνω. Π.χ.:  
`http://www.server.org/inter.html`  
`http://www.server.org:8080/inter.html`  
`http://www.server.org:8081/inter.html`
- Ορισμένες φορές είναι επιθυμητό να τρέχουν περισσότεροι του ενός εξυπηρετητές σε έναν κόμβο, αλλά κάθε διαθέτης να έχει διαφορετικό όνομα κόμβου (host name).
- Αυτό γίνεται με την υποστήριξη του Λειτουργικού Συστήματος στην παροχή *Virtual Host Support* (υποστήριξη εικονικού κόμβου) και την αντιστοίχιση περισσοτέρων της μίας διευθύνσεων IP στο ίδιο μηχάνημα.

# Τα μειονεκτήματα του HTTP 1.0



- Απλό μοντέλο αιτημάτων/απαντήσεων, το οποίο οδηγεί σε προβλήματα:
  - Επίδοσης και Απόδοσης (performance & efficiency)
  - Υλοποίησης εφαρμογών που απαιτούν διαδοχικές αλληλεπιδράσεις πελάτη-εξυπηρετητή.
  - Πολύ φτωχά μοντέλα συνέπειας κρυφής μνήμης (cache consistency)
  - Δυσκολία συνύπαρξης εξυπηρετητών με διαφορετικά ονόματα πεδίου, πάνω στην ίδια διεύθυνση IP. Αυτό που είναι δυνατόν, είναι η χρήση *virtual hosts* – δηλαδή η ύπαρξη περισσότερων εξυπηρετητών *WWW* στην ίδια μηχανή, όπου ο κάθε εξυπηρετητής έχει διαφορετικό όνομα πεδίου και *IP address*.
  - Δυσκολία υπολογισμού μεγέθους περιεχομένου (content-lengths)
  - Διακοπή συνόδου ισοδυναμεί με απώλεια δεδομένων – δεν υπάρχει η δυνατότητα τεμαχισμού της ανακτούμενης πληροφορίας (chunking)



# Επίμονο HTTP (Persistent HTTP)



- Ορισμένα από τα προβλήματα του HTTP/1.0 αντιμετωπίσθηκαν με την υιοθέτηση του Επίμονου HTTP – Persistent HTTP (P-HTTP), σύμφωνα με το οποίο, μετά από μια δοσοληψία HTTP, η αντίστοιχη TCP σύνδεση δεν διακόπτεται, ούτως ώστε τυχόντα αιτήματα που έπονται από τον πελάτη προς τον διαθέτη να εξυπηρετηθούν μέσα από αυτή.
- Στο HTTP/1.0 με χρήση επικεφαλίδας: Connection: Keep-Alive
- Τι πετυχαίνει το P-HTTP προσφέρει βελτιώσεις:
  - Κυρίως στην επίδοση του εξυπηρετητή παρά στο δίκτυο.
  - Επιτρέπει την διασωλήνωση (pipelining) πολλών αιτημάτων στην ίδια σύνοδο TCP
- Δυσκολίες:
  - how does a client know when document is returned?
  - when does the connection get dropped?
    - idle timeouts on server side
    - client drops connections
    - server needs to reclaim resources