



Το πρωτόκολλο HTTP/1.1



Τα προβλήματα του HTTP/1.0

- Ελειπής υποστήριξη εναποθήκευσης (διάρκεια, τοποθεσία κλπ).
- Ανάκτηση ολόκληρου πόρου, ακόμη και αν χρειάζεται μόνο μικρό απόσπασμά του, και αδυναμία να συνεχιστούν διακεκομμένες μεταφορές πόρων.
- Κακή χρήση του TCP για σύντομα μηνύματα αποκρίσεων, τα οποία είναι ο κανόνας στον Ιστό.
- Απουσία εγγύησης για πλήρη παραλαβή δυναμικών πόρων.
- Προβλήματα επεκτασιμότητας και αδυναμία ενημέρωσης για την ύπαρξη διαμεσολαβητών.
- Έλλειψη διευθύνσεων IP.
- Αδυναμία υιοθέτησης προτιμήσεων των πελατών και των εξυπηρετητών για κατάλληλη διαρρύθμιση αιτημάτων και αποκρίσεων.
- Χαμηλά επίπεδα ασφάλειας.
- Ασάφεια στους κανόνες λειτουργίας με διαμεσολαβητές και απομνημονευτές.
- Διάφορα προβλήματα με μεθόδους, επικεφαλίδες και κωδικούς αποκρίσεων.



Νέες ιδέες στο HTTP/1.1

- Μηχανισμός hop-by-hop (βήμα προς βήμα)
- Κωδικοποίηση μεταφοράς (transfer encoding)
- Εικονική στέγαση (virtual hosting)
- Βελτιστοποιημένη χρήση εύρους ζώνης (bandwidth optimization)
- Σημασιολογική διαφάνεια στην εναποθήκευση
- Υποστήριξη για εναλλακτικές κωδικοποιήσεις ενός πόρου



Μηχανισμός βήμα-προς-βήμα

- Το HTTP/1.0 υιοθετεί μια απ' **άκρου-εις-άκρη** προσέγγιση για την μετάδοση μηνυμάτων, ανάλογη με την εικόνα που έχει ο χρήστης του Ιστού: απευθείας σύνδεση με τον πηγαίο εξυπηρετητή.
- Ωστόσο: μεταξύ πελάτη και εξυπηρετητή συνήθως διαμεσολαβούν πολλοί ενδιάμεσοι εξυπηρετητές, οι οποίοι μπορεί να χρησιμοποιούν διαφορετικές (βελτιστοποιημένες) μεθόδους επικοινωνίας.
- Έτσι, δύο ενδιάμεσοι διαμεσολαβητές, πρέπει να έχουν τη δυνατότητα να εισαγάγουν τα κατάλληλα μεταδεδομένα στα μηνύματα HTTP που ανταλλάσσουν, ώστε να μπορούν να βελτιστοποιήσουν την μεταξύ τους επικοινωνία.
- Βήμα-βήμα επικεφαλίδες χρησιμοποιούνται στο HTTP/1.1 γι' αυτό τον σκοπό. Ένας διαμεσολαβητής που παραλαμβάνει επικεφαλίδες β-β τις **διαγράφει** πριν προωθήσει το μήνυμα παρακάτω. Επίσης, μπορεί να προσθέσει κάποιες επικεφαλίδες β-β που να αφορούν στην επικοινωνία του με την επόμενη στάση του μηνύματος.

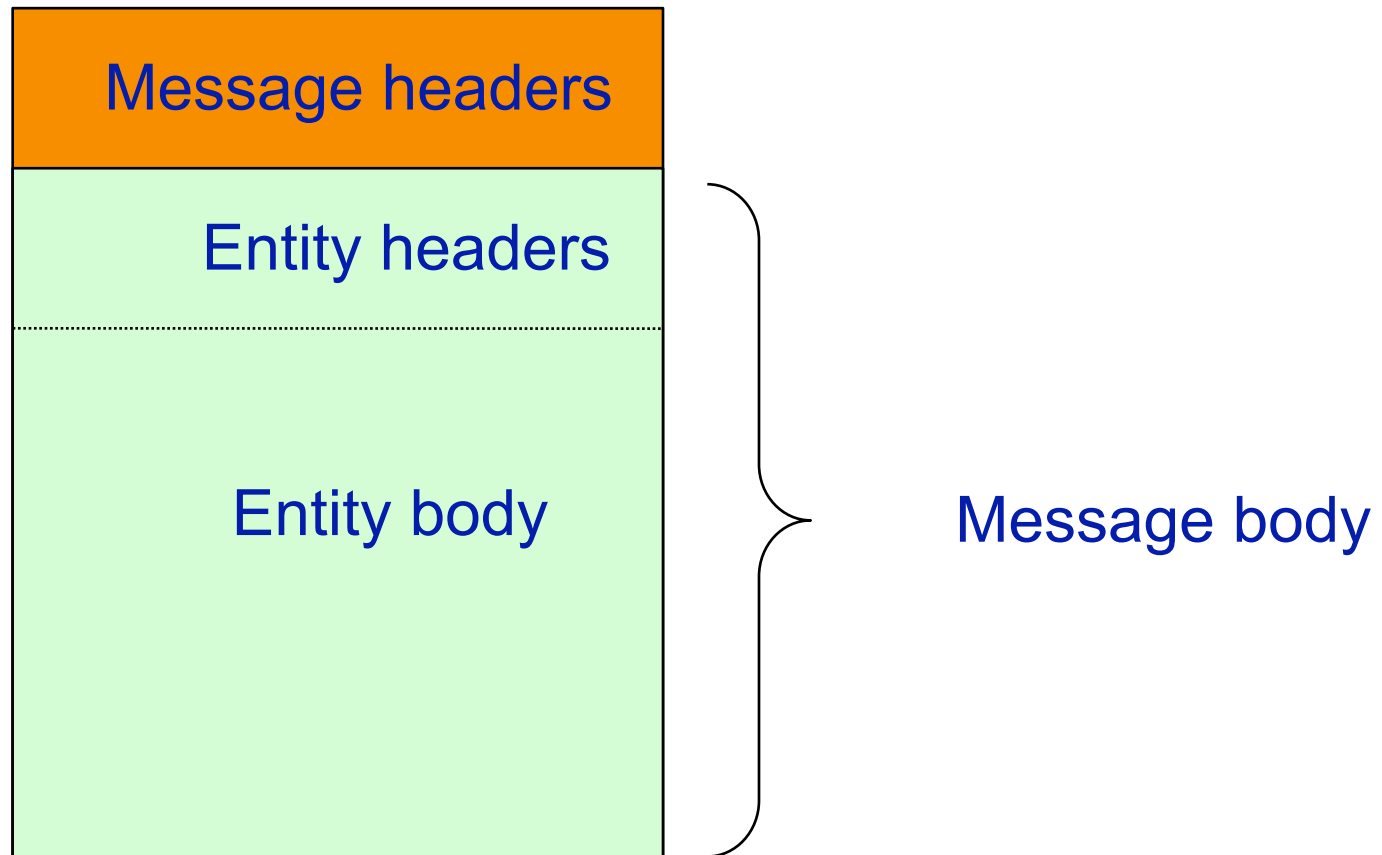


Κωδικοποίηση Περιεχομένου vs Μεταφοράς

- Βασικές έννοιες του HTTP/1.1:
 - **Μήνυμα (message)**: μονάδα επικοινωνίας στο HTTP, που αποτελείται από επικεφαλίδες και (προαιρετικό) σώμα.
 - **Οντότητα (entity)**: το περιεχόμενο το οποίο μεταφέρεται από το μήνυμα και το οποίο διαχωρίζεται σε:
 - επικεφαλίδες οντότητας (entity header) και
 - σώμα οντότητας (entity body)
 - Το σώμα μηνύματος (message body) είναι το ίδιο με το σώμα της οντότητας, εκτός αν στο μήνυμα χρησιμοποιείται κωδικοποίηση μεταφοράς (transfer encoding).



HTTP Message





Content encoding - κωδικοποίηση περιεχομένου

- Primarily used to allow a document to be compressed or otherwise usefully transformed *without losing the identity of its underlying media type and without loss of information*.
 - Frequently, the entity is stored in coded form, transmitted directly, and only decoded by the recipient.
 - Content coding values indicate an encoding transformation that has been or can be applied to an entity.
 - Π.χ. Content-Encoding: gzip
- If content-coding of an entity is **not** "identity" (no transformation whatsoever), then response **MUST** include a Content-Encoding entity-header that lists the content-coding(s) used.
- If the content-coding of an entity in a request message is not acceptable to the origin server, the server **SHOULD** respond with a status code of 415 (Unsupported Media Type).
- If multiple encodings have been applied to an entity, the content codings **MUST** be listed in the order in which they were applied.



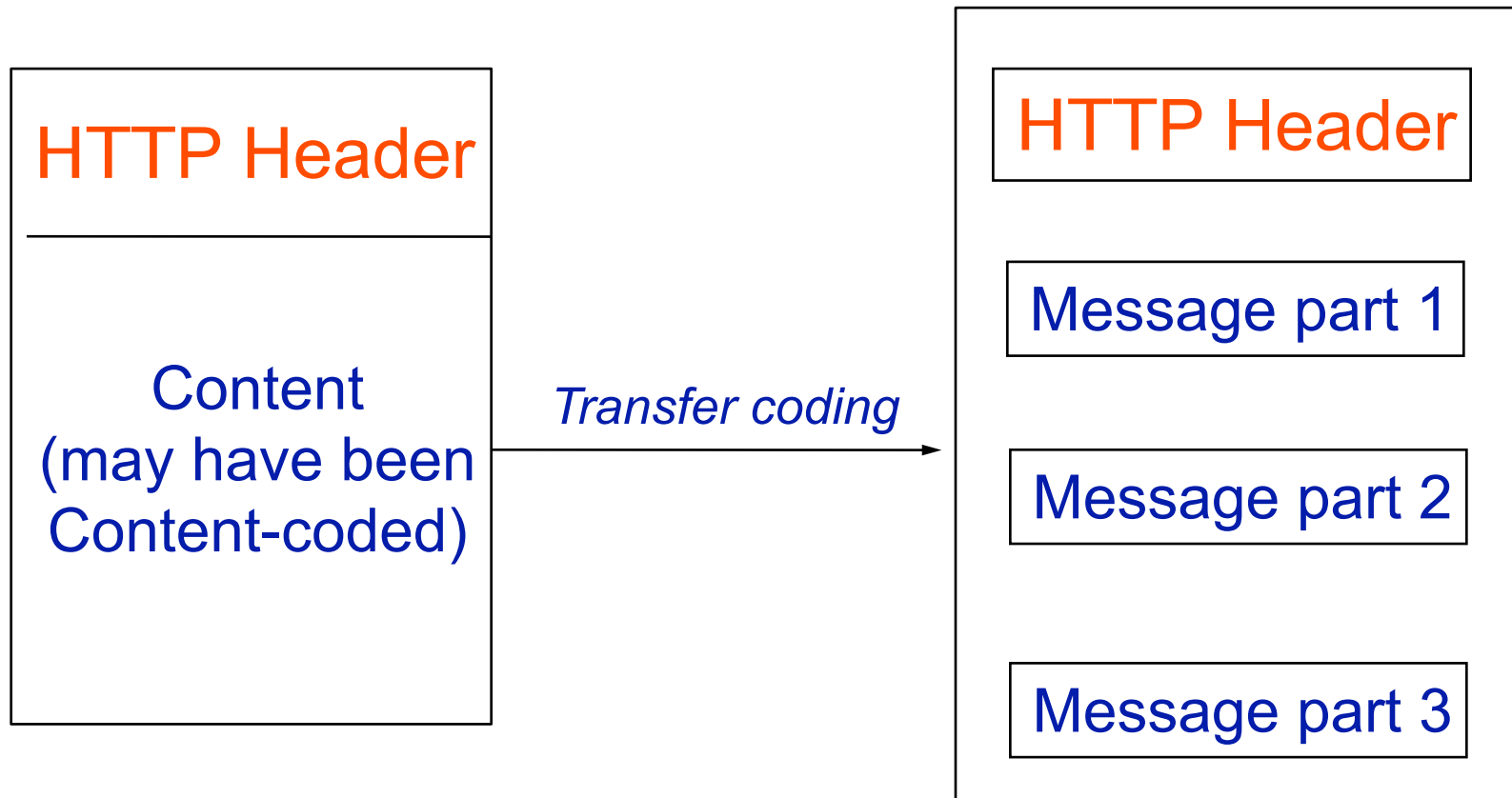
Κωδικοποίηση μεταφοράς (transfer encoding)

- Μετασχηματισμός κωδικοποίησης που έχει, δύναται ή απαιτείται να εφαρμοσθεί στο σώμα μεταφερόμενης οντότητας για την ασφαλή μεταφορά της μέσω του δικτύου.
 - In HTTP, the only unsafe characteristic of message-bodies is the difficulty in determining the exact body length or the desire to encrypt data over a shared transport.
 - Whenever a transfer-coding is applied to a message-body, the set of transfer-codings MUST include "*chunked*", unless the message is terminated by closing the connection.
 - When the "*chunked*" transfer-coding is used, it MUST be the last transfer-coding applied to the message-body.
- Ο μετασχηματισμός κωδικοποίησης είναι ιδιότητα του μηνύματος και όχι της αρχικής οντότητας (πόρος Ιστού).
- Οι επικεφαλίδες που σηματοδοτούν την κωδικοποίηση μεταφοράς (transfer encoding) είναι οι: **TE** και **Transfer-Encoding**, οι οποίες είναι επικεφαλίδες βήμα-βήμα.

—



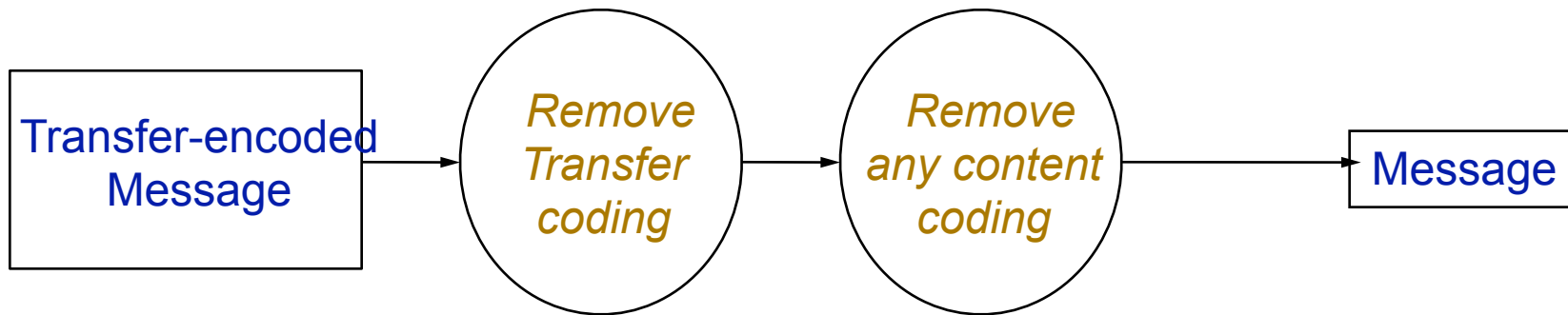
Κωδικοποίηση Μεταφοράς Μηνύματος



- Κωδικοποίηση ανεξάρτητη της σημασιολογίας του περιεχομένου



Αποκωδικοποίηση Μεταφοράς Μηνύματος





Δομή μηνυμάτων στο HTTP/1.1

- Γενική «γραμματική» μηνυμάτων HTTP/1.1:

generic-message =

start-line

*(message-header)

CRLF

[message-body]

message-header = general-header | entity-header | request-header | response-header

start-line = request-line | status-line



Μηνύματα Αιτημάτων HTTP/1.1

- Το μήνυμα αιτήματος είναι πάντοτε το πρώτο μήνυμα σε μια αλληλεπίδραση HTTP. Αποστέλλεται από το σύστημα πελάτη στον εξυπηρετητή, αφού έχει πρώτα εγκαθιδρυθεί η διασύνδεσή τους.

request-message =

request-line

***(general-header | request-header | entity-header)**

CRLF

[message-body]

request-line =

method <space> request-URI <space> HTTP-version

CRLF



Μηνύματα Αποκρίσεων HTTP/1.1

response =

status-line

***(general-header | response-header | entity-header)**

CRLF

[message-body]

status-line =

**HTTP-version <space> status-code <space>
reason-phrase CRLF**



Μέθοδοι Αιτημάτων HTTP/1.1

- Μέθοδοι HTTP: GET, HEAD, POST
- PUT, DELETE, LINK, UNLINK: εμφανίζονται σε μερικές υλοποιήσεις του HTTP/1.0 και περιγράφονται στο παράρτημα του RFC 1945.
- Στο HTTP/1.1:
 - **GET, HEAD, POST**
 - PUT, DELETE – ενσωματώθηκαν
 - OPTIONS, TRACE, CONNECT – προστέθηκαν
 - LINK, UNLINK – αφαιρέθηκαν
- **MUST-level** requirement: μόνο για τις **GET** και **HEAD**. Το HTTP/1.1 καθιστά όλες τις μεθόδους πλέον των **GET** και **HEAD** προαιρετικές, όσον αφορά την υλοποίησή τους από εξυπηρετητές HTTP.



Μέθοδοι Αιτημάτων HTTP/1.1: GET

- **GET:** για την ανάκτηση ενός πληροφοριακού πόρου (URI).
- Αν ακολουθεί στην επικεφαλίδα το **If-Match, If-Modified-Since, If-None-Match, If-Range, If-Unmodified-Since**, η μέθοδος καθίσταται μέθοδος ανάκτησης *υπό συνθήκη* (conditional get). Η χρήση στην επικεφαλίδα του **Range** μετατρέπει την μέθοδο σε μερική ανάκτηση.

GET /Stuff/Funny/silly.html HTTP/1.1

User-Agent: NCSA Mosaic for the X Window System/2.5

Accept: text/plain

Accept: text/html

Accept: application/postscript

Accept: image/gif

Accept: */*



Μέθοδοι Αιτημάτων HTTP/1.1: PUT

- **PUT:**

- Για αποθήκευση μιας οντότητας που στέλνει ο πελάτης προς τον εξυπηρετητή κάτω από URI το οποίο καθορίζεται στο **request-URI**.
- Ο εξυπηρετητής ελέγχει τα δικαιώματα πρόσβασης του πελάτη πριν εφαρμόσει τη μέθοδο PUT στον καθορισμένο πόρο. Εξετάζει επίσης αν ο πόρος προϋπάρχει ή αν θα πρέπει να δημιουργηθεί.
- Αν η νέα οντότητα δημιουργηθεί στον εξυπηρετητή, αυτός επιστρέφει στον πελάτη μήνυμα **201 (created)**.
- Αν γίνει ενημέρωση μιας υπάρχουσας οντότητας στον εξυπηρετητή, αυτός επιστρέφει μήνυμα **200 (ok)**. Αν στην απόκριση προς τον πελάτη δεν συμπεριλαμβάνεται κάποια οντότητα, ο κωδικός κατάστασης της απόκρισης είναι **204 (no content)**.
- Διαμεσολαβητές οι οποίοι δέχονται μηνύματα PUT, πρέπει να ελέγχουν αν έχουν απομνημονεύσει τον μεταφερόμενο πληροφοριακό πόρο – σε μια τέτοια περίπτωση θα πρέπει να σημειώσουν τον πόρο αυτό ως λήξαντα.



Μέθοδοι Αιτημάτων HTTP/1.1: PUT

PUT /foobar/foo.html HTTP/1.1

User-Agent: Mozilla/2.0

Authorization: Basic ZHeadFuaYow29PinWU=

Content-Type: text/html

Content-Length: 433

Host: bar.com

...

<433 bytes of possibly new foo.html>



Μέθοδοι Αιτημάτων HTTP/1.1: HEAD, POST, DELETE

- **HEAD:**

- Στο HTTP/1.1 μπορεί να χρησιμοποιηθεί και με επικεφαλίδες συνθήκης (σε αντίθεση με το HTTP/1.0)

- **POST:**

- Για υποβολή πληροφοριών από τον πελάτη προς κάποια οντότητα (συνήθως πρόγραμμα πύλης), η οποία καθορίζεται στο URI.
- Αν οι πληροφορίες φθάσουν στον εξυπηρετητή και το αίτημα ικανοποιηθεί με την αποστολή προς τον πελάτη μιας οντότητας, η απόκριση έχει κωδικό κατάστασης **200 (οκ)**.
- Αν στην απόκριση προς τον πελάτη δεν συμπεριλαμβάνεται κάποια οντότητα, ο κωδικός κατάστασης της απόκρισης είναι **204 (no content)**.

- **DELETE:**

- Ένας πελάτης μπορεί να ζητήσει την διαγραφή ενός πόρου από τον πηγαίο διαθέτη μέσω της DELETE. Θετική απάντηση του εξυπηρετητή σημαίνει ότι αυτός σκοπεύει να διαγράψει τον πόρο (όχι όμως απαραίτητως άμεσα).



Μέθοδοι Αιτημάτων HTTP/1.1: CONNECT

- Δεν υπήρχε στην αρχική προδιαγραφή του HTTP/1.1 αλλά δεσμεύθηκε αργότερα για μεταγενέστερη προσθήκη.
- CONNECT αιτήματα εγείρονται από ένα διαμεσολαβητή προς κάποιον εξυπηρετητή (όχι από πελάτη)
 - Π.χ. στην περίπτωση που μεταξύ πελάτη και εξυπηρετητή παρεμβάλλεται SSL accelerator, όπου ο accelerator:
 - πραγματοποιεί τη διαπραγμάτευση με τον πελάτη και το κρυπτογραφικό πρωτόκολλο
 - εγκαθιδρύει μια σήραγγα με τον εξυπηρετητή, με την CONNECT - μέσω της σήραγγας προωθεί στον εξυπηρετητή τα αποκρυπτογραφημένα αιτήματα του πελάτη



Γενικές Επικεφαλίδες Μηνυμάτων HTTP/1.1

- **Γενικές Επικεφαλίδες (general header):**
Μπορούν να αφορούν εξίσου και στις δύο κατηγορίες μηνυμάτων:
 - Αιτήματα HTTP
 - Αποκρίσεις HTTP
- Δεν αφορούν στην οντότητα που μεταφέρεται από το μήνυμα.



Γενικές Επικεφαλίδες

- **Date:** Προσδιορίζει την ημερομηνία δημιουργίας του μηνύματος, δηλ. της οντότητας που μεταφέρεται από το μήνυμα. Η χρήση της σε μηνύματα αιτημάτων πελατών είναι προαιρετική.
- **Pragma:**
 - Για τον καθορισμό implementation-specific οδηγιών προς οποιονδήποτε παραλήπτη στην διαδρομή πελάτη-εξυπηρετητή.
 - **Pragma: no-cache**
 - Ισοδυναμεί με **Cache-control: no-cache** για οπισθοδρομική συμβατότητα (Backward compatibility) με HTTP/1.0.
- **Cache-Control:** Προσδιορίζει διάφορες οδηγίες (directives) προς τα συστήματα εναποθήκευσης (caching) αναφορικά με το πώς αυτά θα διαχειριστούν το σχετικό μήνυμα.



Γενικές Επικεφαλίδες (συνέχεια)

- **Connection:**
 - Επιτρέπει στον αποστολέα να καθορίσει την τιμή ορισμένων παραμέτρων που αφορούν μια συγκεκριμένη σύνδεση στη βήμα-προς-βήμα διαδρομή από τον πελάτη στον διαθέτη.
 - Επίσης, χρησιμοποιείται για διαχείριση μιας σύνδεσης, σύμφωνα και με τη προτέρα χρήση της στο HTTP/1.0 (βλ. **Connection: Keep-Alive**) για την επιβολή επιμονής (persistence). Έτσι, στο HTTP/1.1, χρησιμοποιείται από εξυπηρετητές για να δοθεί η εντολή **Connection: Close** κλεισίματος μιας «επίμονης» σύνδεσης, μετά την λήψη του αντίστοιχου μηνύματος.
- **Transfer-Encoding:** transformation to message body
- **Trailer:** list of headers at end of message
- **Upgrade:** upgrading to other protocols
- **Via:** learning about intermediate servers
- **Warning:** error-notification



Επικεφαλίδες αιτημάτων

- Επιτρέπουν στον πελάτη να περάσει πληροφορίες για το αίτημά του ή για τα χαρακτηριστικά του προς τον διαθέτη.
- Λειτουργούν σαν τροποποιητές αιτημάτων, καθώς χρησιμοποιούνται από τα συστήματα πελάτη για να περνούν επιπρόσθετες πληροφορίες σχετικές με τα αιτήματά τους προς τους διαθέτες.
- Το HTTP/1.1 έχει 19 επικεφαλίδες μηνυμάτων, 14 παραπάνω από τις επικεφαλίδες του 1.0 (HTTP/1.0: **Authorization, From, If-Modified-Since, Referer, User-Agent**).
- Η επικεφαλίδα **Authorization** έχει επεκταθεί σημαντικά στο HTTP/1.1. Υπάρχει ξεχωριστό RFC, το οποίο περιγράφει **Basic** και **Digest Authentication**.



Κατηγορίες επικεφαλίδων αιτημάτων

- Προτιμήσεις απόκρισης: γλώσσα, συμβολοσειρά (response preference):
 - Accept, Accept-Charset, Accept-Encoding, Accept-Language, TE (preferred transfer encoding)
- Πληροφορίες οι οποίες αποστέλλονται με το αίτημα: πληροφορίες ταυτοποίησης χρήστη, κλπ (information sent with request):
 - Authorization, From, Referer, User-agent, Proxy-Authorization.
- Υπό συνθήκη αιτήματα: μεταβολή ενός αιτήματος σε υπό συνθήκη αίτημα (conditional requests):
 - If-Modified-Since, If-Match, If-Non-Match, If-Unmodified-Since, If-Range.
- Περιορισμός του εξυπηρετητή: προτρέπουν τον εξυπηρετητή να συμπεριφερθεί κατά κάποιο συγκριμένο τρόπο (constraint on server):
 - Expect (expected server behavior), Host, Max-Forwards, Range.



Επικεφαλίδες Αιτημάτων

- **Accept:** καθορίζει τους μορφότευπους των μέσων (media) που είναι αποδεκτοί σε μιά απόκριση διαθέτη.
- **Accept-Charset:** προτιμητέα συμβολοσειρά.
- **Accept-Encoding:** προτιμητέες κωδικοποιήσεις περιεχομένου.
- **Accept-Language:** προτιμητέα γλώσσας.



Επικεφαλίδες Αιτημάτων (συνέχεια)

- **Authorization:** Χρησιμοποιείται από τα συστήματα-πελάτη για αποστολή πληροφοριών που χρειάζονται για την ταυτοποίηση (authentication) του πελάτη στον διαθέτη. Συνήθως στέλνεται μετά την λήψη από τον πελάτη μηνύματος **401 (unauthorized)**.
- **From:** χρησιμεύει για την δήλωση της διεύθυνσης του ηλε/τ του χρήστη που χρησιμοποιεί το σύστημα-πελάτη.
- **Referer:** Χρησιμεύει από τον πελάτη για ενημέρωση του εξυπηρετητή σχετικά με το URI του πόρου εκ του οποίου προέκυψε το αίτημα προς τον εξυπηρετητή.
- **User-Agent:** Τι είδους φυλλομετρητής κάνει την αίτηση.
- **Proxy-Authorization:** Εξουσιοδότηση ενός πελάτη προς τον διαμεσολαβητή.



Επικεφαλίδες Αιτημάτων (συνέχεια)

- **If-Modified-Since, If-Unmodified-Since:** χρησιμεύει για την τροποποίηση ενός αιτήματος GET σε αίτημα υπό συνθήκη. Εξυπηρετεί στην περίπτωση τοπικής εναποθήκευσης (στον πελάτη ή σε διαμεσολαβητή) ιστιακών πόρων. Αποκρίσεις επιτυχίας: **304 (not modified)** , **200 (ok)**
- **If-Match, If-None-Match:** Έλεγχος με βάση ετικέτες οντότητας (entity tags).
- **If-Range:** χρησιμοποιείται σε συνδυασμό με την επικεφαλίδα **Range** στην περίπτωση που κάποιος πελάτης επιδιώκει να μεταφορτώσει ένα τμήμα κάποιου πόρου, στην περίπτωση που υπόλοιπα τμήματα που διαθέτει είναι επίκαιρα.



Επικεφαλίδες Αιτημάτων (συνέχεια)

- **Host:** Καθορίζει τον διαδικτυακό κόμβο και τον αριθμό πύλης του αιτούμενου πληροφοριακού πόρου, όπως αυτά λαμβάνονται από το URL που καθόρισε ο χρήστης. Είναι η μόνη επικεφαλίδα που βρίσκεται **υποχρεωτικά** σε κάθε αίτημα HTTP/1.1.
 - Συμπεριλαμβάνεται στα αιτήματα ώστε να είναι δυνατόν ο πηγαίος εξυπηρετητής να επιλύσει τον αποδέκτη του αιτήματος, στην περίπτωση ύπαρξης πολλαπλών κόμβων ΠΠΠ στην ίδια διεύθυνση IP (με χρήση ψευδωνύμων).
 - Απουσία του πεδίου **Host** συνεπάγεται απόκριση **400 (bad request)**.
- **Expect**
- **Max-Forwards:** επιτρέπει σε ένα πελάτη να καθορίσει τον μέγιστο αριθμό προωθήσεων αιτήματός του από διαμεσολαβητές (χρησιμοποιείται με την TRACE, OPTION)
- **Range:** αίτημα για ένα εύρος χαρακτήρων της οντότητας



Επικεφαλίδες Απόκρισης

- **Επικεφαλίδες Απόκρισης (response-header):**
 - Χρησιμοποιούνται για να περάσουν πληροφορίες σχετικά με την απόκριση σε αίτημα, οι οποίες δεν «χωρούν» στο status line.
 - Δεν αφορούν στο «σώμα» της απάντησης.
 - Στο HTTP/1.1 έχουν αποσαφηνιστεί οι ρόλοι των επικεφαλίδων Location, Server και WWW-Authenticate και έχουν προστεθεί έξι νέες επικεφαλίδες:
 - Επαναπροώθηση: Location
 - Πληροφοριακές: Server, Retry-After, Accept-Ranges
 - Ασφάλειας: WWW-Authenticate, Proxy-Authenticate
 - Σχετικές με εναποθήκευση: Etag, Age, Vary



Επικεφαλίδες Απόκρισης (συνέχεια)

- **Server:** Παρέχει πληροφορίες σχετικά με τον εξυπηρετητή (τι είδους, ποιά έκδοση κλπ).
- **Accept-Ranges:** Προσδιορίζει τις μονάδες εύρους που μπορούν να χρησιμοποιούνται σε αιτήματα **Range**. Στο HTTP/1.1 μπορεί να είναι είτε **bytes** είτε **none**.
- **Location:** Περιλαμβάνεται σε απόκριση **201 (created)** αιτήματος δημιουργίας νέου πόρου, και αναφέρει το νεοδημιούργητο URI. Στην περίπτωση απόκρισης **3xx** (redirection), προσδιορίζει το URI το οποίο θα πρέπει να χρησιμοποιήσει ο πελάτης.
- **Retry-After:** Σε συνδυασμό με αποκρίσεις **503 (service unavailable)**



Επικεφαλίδες Οντότητας (entity headers)

- **Επικεφαλίδες Οντότητας (entity header):** Αφορούν στην οντότητα που μεταφέρεται από το μήνυμα, ή στον πληροφοριακό πόρο τον οποίο αφορούσε το αντίστοιχο αίτημα HTTP (αν δεν μεταφέρεται οντότητα με το μήνυμα).
- Βασικά πρόκειται για μεταπληροφορίες.
- Μη αναγνωρίσιμες επικεφαλίδες οντότητας αγνοούνται από τους παραλήπτες τους. Οι διαμεσολαβητές πρέπει να προωθούν τέτοιες επικεφαλίδες, ακόμη κι αν δεν τις αναγνωρίζουν.
- Πέρα από τις έξι επικεφαλίδες οντότητας του HTTP/1.0 (Allow, Content-Encoding, Content-Length, Content-Type, Expires, Last-Modified), το HTTP/1.1 προσθέτει τέσσερις νέες επικεφαλίδες: Content-Language, Content-Location, Content-MD5, Content-Range.



Επικεφαλίδες Οντότητας (συνέχεια)

- **Content-Length:**

- Καθορίζει το μέγεθος της μεταφερόμενης οντότητας σε χαρακτήρες. Είναι σημαντική η χρήση του για την επιβεβαίωση της πλήρους παραλαβής μιας οντότητας.
- Χρησιμοποιείται σε μηνύματα **αιτημάτων και αποκρίσεων**.
- Χωρίς χρήση της επικεφαλίδας Content-Length, ένα σύστημα πελάτη θα πρέπει να κλείσει τη σύνδεσή του με τον εξυπηρετητή για να σηματοδοτήσει την λήξη αποστολής ενός αιτήματος.
- Η χρήση του σε «δυναμικές» απαντήσεις συνήθως παραλείπεται.

- **Expires:** Χρησιμοποιείται για να καθορίσει την χρονική στιγμή ύστερα από την οποία μια οντότητα θεωρείται λήξασα. Η χρονική στιγμή κωδικοποιείται σύμφωνα με το RFC 1123.

- **Last-Modified:**

- Προσδιορίζει την χρονική στιγμή στην οποία ο εξυπηρετητής θεωρεί ότι τροποποιήθηκε για τελευταία φορά το περιεχόμενο της οντότητας.
- Πρέπει να συμπεριλαμβάνεται στην απόκριση των εξυπηρετητών Ιστού, όποτε είναι αυτό δυνατό.



Επικεφαλίδες βήμα προς βήμα

- Το HTTP/1.1 εισαγάγει την επικεφαλίδα β-β **Connection**, για να δηλώσει τον κατάλογο των επικεφαλίδων β-β που εισάγονται σε ένα μήνυμα και αφορούν στον επόμενο σταθμό του μηνύματος **μόνο**.
- Π.χ. η δήλωση **Connection: header1, header2**, υποδηλοί ότι όταν ένας εξυπηρετητής λάβει μήνυμα με αυτή, θα πρέπει να διαγράψει τις επικεφαλίδες **header1** και **header2** από το μήνυμα, πριν το επαναδρομολογήσει σε κάποιον άλλο εξυπηρετητή ή ενδιάμεσο.
- Επικεφαλίδες β-β είναι οι:
 - **Keep-Alive**
 - **Proxy-Authenticate**
 - **Proxy-Authorization**
 - **Trailers**
 - **TE**
 - **Transfer-Encoding**
 - **Upgrade**
- Οι υπόλοιπες καθορισμένες επικεφαλίδες του HTTP θεωρούνται απ' άκρου εις άκρο και δεν μπορούν να θεωρηθούν β-β ακόμη κι αν δηλωθούν με το **Connection**.



Κωδικοί Κατάστασης HTTP/1.1

- Σαράντα ένας κωδικοί κατάστασης στο HTTP/1.1
- Οικογένεια κωδικών με πέντε «τύπους»:
- 1xx: **Informational**. Σηματοδοτούν επιτυχή λήψη τού αιτήματος από τον εξυπηρετητή και συνεχιζόμενη επεξεργασία του.
 - 100 Continue
 - 101 Switching Protocols
- 2xx: **Success class**
 - 200 OK
 - 201 Created
 - 202 Accepted
 - 203 Non-Authoritative Information (metadata not definitive)
 - 204 No Content
 - 205 Reset Content (user view change)
 - 206 Partial Content (successful range request)



Κωδικοί Κατάστασης HTTP/1.1

- **3xx: Redirection.** Καθορίζουν εναλλακτικές δράσεις στις οποίες πρέπει να προβεί ο πελάτης για την ολοκλήρωση του αιτήματός του.
 - 300 Multiple Choices
 - 301 Moved Permanently
 - 302 Found
 - 303 See Other
 - 304 Not Modified (revalidation response)
 - 305 Use Proxy (request again via proxy)
 - 306 (unused)
 - 307 Temporary Redirect (temporary location of URI)



Κωδικοί Κατάστασης HTTP/1.1

- 4xx: **Client Error.** Το σύστημα πελάτη έχει κάνει κάποιο σφάλμα στο αίτημά του, το οποίο δεν μπορεί να εξυπηρετηθεί.
- Στο HTTP/1.1 ορίζονται 14 νέοι κωδικοί κατάστασης σφάλματος πελάτη, οι οποίοι μπορούν να ταξινομηθούν σε πέντε υποκατηγορίες:
- HTTP/1.0 error codes
 - 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found
- Clarification of HTTP/1.0 codes:
 - 405 Method Not Allowed, 407 Proxy Authentication Required
 - 408 Request Timeout, 410 Gone
- Για διαπραγμάτευση περιεχομένου στο HTTP/1.1
 - 406 Not Acceptable, 415 Unsupported Media Type, 413 Request Entity Too Large
- Length Related
 - 411 Length Required, 414 Request-URI Too Long



Κωδικοί Κατάστασης HTTP/1.1

- 5xx: Server error. Επιστρέφεται όταν ο εξυπηρετητής δεν μπορεί να διαχειριστεί ένα εισερχόμενο αίτημα ή όταν βρίσκεται σε κατάσταση σφάλματος:
 - 500 Internal Server Error (unexpected server condition)
 - 501 Not Implemented (functionality lacking)
 - 502 Bad Gateway (upstream server error)
 - 503 Service Unavailable (temporary inability to handle request)
 - 504 Gateway Timeout (upstream server's delay in response)
 - 505 HTTP Version Not Supported



Εναποθήκευση στο HTTP/1.1



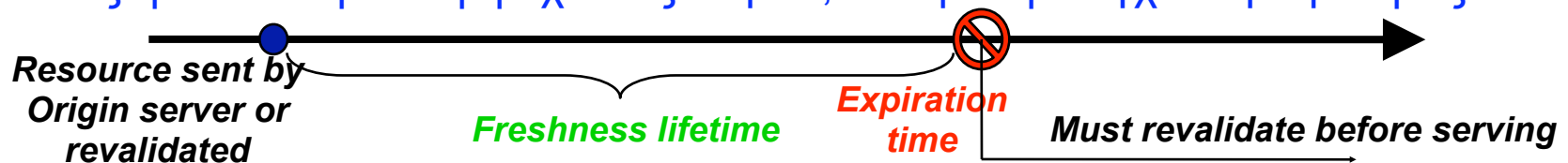
Εναποθήκευση στο HTTP/1.1

- Το HTTP/1.1 καθορίζει μιας σειρά από σημασιολογικούς κανόνες σχετικούς με την εναποθήκευση σε κρύπτες (cache):
 - Εναποθηκευσιμότητα (cacheability)
 - Επικύρωση (validation)
 - Συνέπεια αποκρίσεων (coherence of cached responses).
- Το HTTP/1.1 δεν ασχολείται με θέματα αντικατάστασης (cache replacement).
- Στόχος HTTP/1.1: **σημασιολογική διαφάνεια (semantic transparency)** της εναποθήκευσης: *μια κρυφή μνήμη θα πρέπει να μπορεί να επιστρέφει μια απομνημονευμένη απόκριση χωρίς να υπάρχει ο φόβος ότι ο πηγαίος εξυπηρετητής θα επέστρεφε μια διαφορετική απόκριση.*



Ορολογία HTTP/1.1 για εναποθήκευση

- **Age:** ηλικία μιας απόκρισης, από την στιγμή τής αποστολής της από τον πηγαίο εξυπηρετητή ή από την επανεπικύρωσή της από την cache (σε sec).
- **Expiration time:** χρονική στιγμή μετά από την οποία η cache *πρέπει* να επικυρώσει και πάλι μια οντότητα, πριν την επιστρέψει σε απάντηση (σε sec). Η χρονική στιγμή λήξης ορίζεται από τον πηγαίο εξυπηρετητή (σε αντίθετη περίπτωση, η κρύπτη μπορεί να ορίσει με ευρετικό τρόπο την χρονική στιγμή λήξης).
- **Freshness lifetime & staleness:** χρονικό διάστημα από την δημιουργία μιας απάντησης μέχρι την στιγμή της λήξης (expiration).
- **Validity:** εγκυρότητα ενός αποθηκευμένου πόρου.
- **Cacheability:** ιδιότητα ιστιακού πόρου που καθορίζει αν ο πόρος μπορεί να εναποθηκευθεί σε κάποια κρύπτη ή όχι.
- **Cache maintenance:** διαχείριση της cache σημαίνει την αντιμετώπιση θεμάτων όπως η εναποθήκευση ή όχι ενός πόρου, απόφαση ελέγχου εγκυρότητας κλπ.





Απομνημόνευση στο HTTP/1.0

- Οδηγία αιτημάτων: **Pragma: no-cache**
- Υπό συνθήκη GET: **If-modified-Since**
- Επικεφαλίδα απάντησης: **Expires**
- **Cache busting**: κατάχρηση της επικεφαλίδας Expires ώστε να επιβάλλεται η άμεση λήξη ενός πόρου και συνεπώς να αποτρέπεται η εναποθήκευσή του σε κάποια κρύπτη.



Αρχές εναποθήκευσης στο HTTP/1.1

- Διαχωρισμός του τι απομνημονεύεται από το αν μια εναποθηκευμένη απάντηση μπορεί να χρησιμοποιηθεί με ασφάλεια: ο ρόλος του πρωτοκόλλου είναι να διασφαλίζει την ορθότητα και όχι να καθορίζει θέματα πολιτικής (π.χ. την διάρκεια μιας απομνημονευμένης οντότητας).
- Διασφάλιση της ορθότητας ακόμη και με κάποιο κόστος.
- Δυνατότητα των εξυπηρετητών να παράσχουν περισσότερες πληροφορίες σχετικά με την απομνημονευσιμότητα των πόρων, και παροχή μεγαλύτερου ελέγχου στους χρήστες σχετικά με το τι μπορεί να απομνημονευθεί.
- Αποδέσμευση από τις *απόλυτες* χρονοσφραγίδες και την ανάγκη συγχρονισμού των ρολογιών πελατών και εξυπηρετητών.
- Υποστήριξη της εναποθήκευσης διαπραγματευόμενων αποκρίσεων.
- Ευρωστία: τα δομοστοιχεία λογισμικού πρέπει να είναι “φιλελεύθερα” στο τι μηνύματα δέχονται και συντηρητικά στο τι μηνύματα στέλνουν.



Αρχές εναποθήκευσης στο HTTP/1.1

- Το HTTP/1.1 καθορίζει κανόνες για τον συνδυασμό επικεφαλίδων ώστε να παρέχονται ορθές απαντήσεις από τις κρύπτες.
- Π.χ., μόλις μια κρύπτη λάβει μια απόκριση από έναν πηγαίο εξυπηρετητή, θα πρέπει να αποφασίσει τι απόκριση να στείλει στον πελάτη:
 - Αν η απόκριση του Π/Ε είναι 200 OK, η κρύπτη απομνημονεύει την απόκριση και την προωθεί.
 - Αν η απόκριση είναι 304 Not Modified, η κρύπτη μπορεί να στείλει στον πελάτη την αποθηκευμένη απόκριση που διαθέτει.
 - Ωστόσο η κρύπτη *δεν επιτρέπεται* να χρησιμοποιήσει τις επικεφαλίδες απ-άκρου-εις άκρο της αποθηκευμένης απόκρισης, αλλά *πρέπει* να χρησιμοποιήσει τις νέες επικεφαλίδες που λαμβάνει από τον πηγαίο εξυπηρετητή (ο εξυπηρετητής μπορεί να έχει αλλάξει τις τιμές του Expires και του Cache-Control)..



Επικεφαλίδες Εναποθήκευση

- Το HTTP/1.1 προσθέτει 4 νέες επικεφαλίδες που αφορούν σε εναποθήκευση:
 - Age
 - Cache-Control
 - Etag
 - Vary
- **Age:** χρησιμοποιείται από τον εξυπηρετητή για να προσδιορίσει την ηλικία του ιστιακού πόρου, ορισμένη είτε σαν το χρονικό διάστημα από τη στιγμή δημιουργίας του ιστιακού πόρου στον εξυπηρετητή, ή σαν το χρονικό διάστημα από τη στιγμή επικύρωσης τού πόρου από τον εξυπηρετητή εναποθήκευσης.



Cache-Control σε μηνύματα αιτήματος

- **Γενική επικεφαλίδα** που χρησιμεύει για διακίνηση οδηγιών σχετικά με την απομνημόνευση πόρων. Οι ενδιαμέσοι εξυπηρετητές απαιτείται όπως αναμεταδίδουν την επικεφαλίδα αυτή, και όλες οι κρυφές μνήμες πρέπει να εκτελούν τις οδηγίες που διακινεί.
- Οδηγίες που καθορίζονται από την Cache-Control σε μηνύματα **αιτημάτων**:
 - **no-cache**: Απαγορεύει την επιστροφή εναποθηκευμένου πόρου (end-to-end reload)
 - **only-if-cached**: Επιβάλλει απάντηση μόνο από κάποια cache.
 - **no-store**: Αποτρέπει την εναποθήκευση αιτημάτων και αποκρίσεων (για λόγους, π.χ., ασφαλείας πληροφοριών).
 - **max-age**: Ο πελάτης μπορεί να επιβάλλει διαφορετική ημερομηνία λήξης ενός πόρου, από αυτήν που έχει καθοριστεί μέσω του μηχανισμού Expires – π.χ. **max-age:0** επιβάλλει επικύρωση από-άκρου-εις-άκρον.
 - **max-stale**: Ο πελάτης αποδέχεται απαντήσεις που μπορεί να έχουν λήξει, επιβάλλει ωστόσο ένα άνω όριο στην υπέρβαση του χρόνου λήξης. Η χρήση της Max-stale χωρίς όρισμα προσδιορίζει ότι ο πελάτης είναι πρόθυμος να αποδειχθεί οποιαδήποτε απόκριση.
 - **no-transform**: Εμποδίζει τους διαμεσολαβητές από του να μετασχηματίζουν το μορφότυπο (media type) πόρων που διακινούνται μέσω HTTP.



Cache-Control σε μηνύματα αποκρίσεων

- **public:** ο εξυπηρετητής προσδιορίζει ότι η συγκεκριμένη απόκριση μπορεί να απομνημονευθεί - ο προσδιορισμός αυτός έχει προτεραιότητα έναντι άλλων πολιτικών που μπορεί να χρησιμοποιούνται από τις κρύπτες.
- **private:** σημαίνει ότι το μήνυμα αφορά έναν μόνο χρήστη και ΔΕΝ ΠΡΕΠΕΙ να απομνημονευτεί σε κάποια κοινόχρηστη κρύπτη.
- **No-cache:** βασικός στόχος είναι να εξαναγκάζει τις κρύπτες να ελέγχουν την εγκυρότητα του σχετικού πόρου με τον πηγαίο εξυπηρετητή. Επομένως, η απόκριση μπορεί να απομνημονευθεί αλλά η κρύπτη πρέπει να ελέγξει την εγκυρότητα του πόρου πριν προωθήσει την απόκριση.
 - Μπορεί να χρησιμοποιηθεί και σε συνδυασμό με ονόματα επικεφαλίδων:

HTTP/1.1 200 OK
Server: Apache/1.2.6
Date: Thu, 24 Feb 2004 18:13:36 GMT
Content-Type: text/html
Location: http://foo.com/zzz.gif
Cache-Control: no-cache=Location

Το περιεχόμενο της απόκρισης μπορεί να επαναχρησιμοποιηθεί αν απομνημονευθεί, αλλά όχι η επικεφαλίδα Location



Επεκτάσεις της Cache-control

- Γίνονται με χρήση cache-extension tokens:
 - **Informational extensions** (δεν επιβάλλουν αλλαγή στη συμπεριφορά της κρύπτης) MAY be added without changing the semantics of other directives.
 - **Behavioral extensions** (λειτουργούν σαν τροποποιητές των υπαρχόντων οδηγιών κρύπτης - cache directives).
 - Both the **new directive and the standard directive are supplied**, such that applications which do not understand the new directive will default to the behavior specified by the standard directive.
 - In this way, *extensions to the cache-control directives can be made without requiring changes to the base protocol*.
 - For example, consider a hypothetical new response directive called **community** which acts as **a modifier to the private directive**. We define this new directive to mean that, in addition to any non-shared cache, **any cache which is shared only by members of the community named within its value may cache the response**. An origin server wishing to allow the UCY community to use an otherwise private response in their shared cache(s) could do so by including

Cache-Control: private, community="UCY"



Cache-Control σε μηνύματα αποκρίσεων

- **no-store:** χρησιμοποιείται για να αποτρέπει την απομνημόνευση του μηνύματος αιτήματος και του μηνύματος απόκρισης.
- **no-transform:** επιβάλλει την μη μετατροπή μίας απόκρισης από κάποια κρύπτη (π.χ. για τον περιορισμό της ευκρίνειας μιας εικόνας).
- **must-revalidate:** επιτρέπει στον πηγαίο εξυπηρετητή να προσδιορίσει ότι για κάποια απόκριση η κρύπτη πρέπει να αποφύγει να την σερβίρει ληγμένη σε κάποιον πελάτη.
 - Σε μια τέτοια περίπτωση η κρύπτη θα πρέπει να επαναβεβαιώσει την εγκυρότητα της απόκρισης (και αν δεν τα καταφέρει, να στείλει μήνυμα λάθους, πχ 504 Gateway Timeout).
 - Η must-revalidate επιβάλλει την επαναβεβαίωση **άμα τη λήξη μιας αποθηκευμένης απόκρισης**, ενώ η no-cache επιβάλλει την επαναβεβαίωση κάθε φορά που σχετικό αίτημα φθάνει στην κρύπτη (ανεξαρτήτως λήξης ή όχι).
- **proxy-revalidate:** ίδια με την must-revalidate μόνο που δεν αφορά σε μη κοινόχρηστες κρύπτες (προσωπικές κρύπτες).
- **max-age:** προτιμώμενος τρόπος για τον καθορισμό του χρόνου λήξης μιας οντότητας - ακυρώνει τον χρόνο λήξης που ορίζεται με το Expires



Επικεφαλίδα ETag

- Έχει εισαχθεί για την μεταφορά συμπαγούς κλειδιού για διάκριση ανάμεσα σε διαφορετικές εκδόσεις του ίδιου ιστιακού πόρου, οι οποίες μπορεί να συνυπάρχουν στο διαδίκτυο.
- Κάθε διαφορετική έκδοση έχει διαφορετικό κλειδί.
- Η χρήση του Etag αποσυνδέει τον έλεγχο εγκυρότητας στις κρύπτες από τον μηχανισμό λήξης και το πρόβλημα του συγχρονισμού των ρολογιών.
- Χρήση:
HTTP/1.1 200 OK
Date: Sun, 26 Apr 2000 18:12:00 GMT
Server: Apache/1.2.6 Red Hat
Last-Modified: Fri, 24 Apr 2000 09:00:00 GMT
ETag: cc678-12d12-663888909
Content-Length: 15044
Content-Type: text/html

<contents of foo.html>
- Χρησιμοποιείται σε συνδυασμό με επικεφαλίδες **If-Match**, **If-None-Match**.



Επικεφαλίδα Vary

- Η εισαγωγή και αναζήτηση ενός πόρου σε μια κρύπτη στο HTTP/1.0 γίνεται με χρήση του URI ως κλειδιού αποθήκευσης του πόρου.
- Ωστόσο, η συνύπαρξη διαφορετικό “εκδοχών” του ίδιου πόρου καθιστούν το URI ανεπαρκές σαν κλειδί, καθώς η *διαπραγμάτευση περιεχομένου* επιτρέπει στην κρύπτη να επιλέγει την απόκριση που θα στείλει με βάση τις επιλογές του πελάτη.
- Η επικεφαλίδα Vary χρησιμοποιείται για να δηλωθούν οι επικεφαλίδες αιτήματος οι οποίες πρέπει να χρησιμοποιούνται για την επιλογή της κατάλληλης εκδοχής ενός πόρου.



Παράδειγμα Vary

HTTP/1.1 200 OK

Date:...

Server:...

Last-Modified:...

Etag: ...

Content-Length: 15044

Content-Type: text/html

Vary: Accept-Language

....

A cache caching this response must also store the **value in the request headers listed in the vary response header**.

If a subsequent request made to the cache is about to yield a cached response, the cache must ensure that all the headers in that request that are listed in the Vary header are identical to what were in the original request.



Βελτιστοποίηση Εύρους Ζώνης στο HTTP/1.1



Βελτιστοποίηση χρησιμοποιούμενου Εύρους Ζώνης

- Το χρησιμοποιούμενο από το HTTP εύρος ζώνης έχει αυξηθεί δραματικά λόγω:
 - Αύξησης του μεγέθους των πληροφοριακών πόρων.
 - Ενσωμάτωσης εικόνων σε κείμενα.
 - Παρουσία περισσότερων χρηστών στον Ιστό, με μεγαλύτερη χωρητικότητα δικτυακών συνδέσεων.
 - Η χρήση από το Netscape του παράλληλου ανοίγματος πολλών συνδέσεων για παράλληλη πρόσκτηση πόρων.
- Βελτιστοποιήσεις που προτάθηκαν από το HTTP/1.1:
 - Αν δεν χρειάζεται να μεταφερθεί μια ολόκληρη οντότητα, το πρωτόκολλο πρέπει να επιτρέπει τον καθορισμό τμημάτων της για μεταφορά (**range request mechanism**).
 - Αν ο αποστολέας γνωρίζει εκ των προτέρων ότι ο παραλήπτης δεν μπορεί να διαχειριστεί το σώμα του μηνύματος, καλύτερα να μην το αποστείλει καθόλου. Ανταλλαγή πληροφοριών ελέγχου από το πρωτόκολλο, διευκολύνει προς αυτή την κατεύθυνση (**expect/continue mechanism**).
 - Δυνατότητα μετασχηματισμού συμπίεσης μιας οντότητας πριν αυτή σταλθεί στο δίκτυο, και ανακατασκευής της άμα τη παραλήψη (**compression**).



Αίτημα Διαστημάτων (range request)

- Μια από τις κινητήριες δυνάμεις πίσω από τα αιτήματα διαστημάτων στο HTTP υπήρξη ο μορφότυπος PDF της Adobe, ο οποίος προσδιορίζει σε ένα έγγραφο τους αρχικούς και καταληκτικούς χαρακτήρες (bytes) της κάθε σελίδας.
- Έτσι, ένας πελάτης ιστού, έχει τη δυνατότητα να εγκαθιδρύσει πολλαπλές συνδέσεις και να ανακτήσει παραλλήλως διαφορετικές σελίδες ενός PDF αρχείου, πριν τις χρησιμοποιήσει για να ανακατασκευάσει το αρχείο (χρήση παραλληλίας για λόγους **επίδοσης**, δηλ. μικρότερης υστέρησης στην εμφάνιση του αρχείου).
- Το HTTP/1.1 επιτρέπει μέσω του μηχανισμού **αιτήματος διαστημάτων** τη δυνατότητα προσδιορισμού από έναν πελάτη **τμημάτων ενός πληροφοριακού πόρου** τα οποία ο πελάτης θέλει να αποκτήσει.



Αίτημα διαστήματος (range request)

- Επιτρέπει την διακίνηση μέσω HTTP/1.1 τμήματος ενός (μεγάλου) ιστιακού πόρου, για λόγους εξοικονόμησης εύρους ζώνης:

```
GET bigfile.html HTTP/1.1
```

```
Host: www.whatinfo.com
```

```
Range: 2000-3999
```

```
HTTP/1.1 206 Partial Content
```

```
Date: Thu, 10 Feb 2001 20:02:00 GMT
```

```
Last-Modified: Wed, 9 Feb 2000 14:58:08 GMT
```

```
Content-Range: bytes 2000-3999/100000
```

```
Content-Length: 2000
```

```
Content-Type: text/html
```



Αίτημα διαστήματος (συνέχεια)

GET bigfile.html HTTP/1.1

Host: www.whatinfo.com

Range: 0-100, 2000-2400, 9600-

HTTP/1.1 206 Partial Content

Date: Thu, 10 Feb 2001 20:02:00 GMT

Server: Apache/1.2.6 Red Hat

Last-Modified: Wed, 9 Feb 2000 14:58:08 GMT

Etag: cc9878-12d12-66394036

Content-Type: multipart/byteranges; boundary=----ROPE----

----ROPE----

Content-Type: text/html

Υποδηλώνει ότι η
απόκριση δεν
συμπεριλαμβάνει
ολόκληρο τον πόρο



Αίτημα διαστήματος (συνέχεια)

- Αν τα αιτήματα διαστήματος αφορούν σε διαστήματα περιεχομένου που δεν υπάρχουν, τότε ο εξυπηρετητής επιστρέφει:
 - 416 Requested Range Not Satisfiable
- Ένας εξυπηρετητής μπορεί να δηλώσει αν δέχεται ή όχι αιτήματα διαστήματος ως εξής:
 - Accept-Ranges: bytes
 - Accept-Ranges: none (σε τέτοια περίπτωση μπορεί να απαντήσει σε αίτημα διαστήματος με ολόκληρο τον πόρο)



Αίτημα διαστήματος υπό συνθήκη

- Τι συμβαίνει αν ζητηθεί από μια κρύπτη ένα τμήμα ενός πληροφοριακού πόρου, η ολότητα του οποίου **δεν** βρίσκεται στην κρύπτη;
 - Αν ο πληροφοριακός πόρος δεν έχει αλλάξει στον πηγαίο εξυπηρετητή, αρκεί να μεταφέρουμε από αυτόν μόνο το ζητούμενο τμήμα.
 - Αν ο πληροφοριακός πόρος έχει αλλάξει, θα πρέπει να μεταφέρουμε ολόκληρο τον πόρο.
 - Για να αποφύγουμε τα δύο μηνύματα στην πρώτη περίπτωση, το HTTP/1.1 χρησιμοποιεί την επικεφαλίδα If-Range.



Αίτημα διαστήματος υπό συνθήκη

```
GET bigfile.html HTTP/1.1
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Range: bytes=1300-1500
If-Range: cc678-12d12-66394036
User-Agent: Mozilla
Host: www.notonline.com
```

- Η επικεφαλίδα If-Range επιτρέπει τη σύγκριση τού **entity-tag**, το οποίο μεταφέρει (cc678-12d12-66394036), έναντι του entity-tag το οποίο αποθηκεύεται στον πηγαίο εξυπηρετητή, για διαπίστωση τού εάν ο πληροφοριακός πόρος έχει μεταβληθεί.
- Αν το κλειδί είναι το ίδιο, τότε ο εξυπηρετητής θα στείλει τα ζητούμενα 201 bytes, αλλιώς όλο τον πόρο.



Αίτημα διαστήματος υπό συνθήκη

```
GET /ads/game03.gif?0CYES HTTP/1.1  
Host: www.notonline.com  
Range: bytes=8353-  
If-Unmodified-Since: Tue, 26 Oct 2002 10:54:20 GMT
```

- Στην περίπτωση που ο πόρος δεν έχει αλλάξει, η απάντηση είναι ως εξής:

```
HTTP/1.1 206 Partial content  
Server: Microsoft-IIS/3.0  
Date: Tue, 26 Oct 2002 10:54:20 GMT  
Content-Type: image/gif  
Last-Modified: Tue, 26 Oct 2002 9:54:20 GMT  
Content-Length: 14255  
Content-Range: bytes 8353-22607/22608
```



Μηχανισμός Δήλωσης Προσδοκιών

Expect/Continue



Μηχανισμός expect/continue

- Υπάρχει ανάγκη μηχανισμού μέσω του οποίου ένας πελάτης θα μπορεί να ανακαλύπτει τους πιθανούς περιορισμούς ενός εξυπηρετητή πριν υποβάλλει προς αυτό ένα (μεγάλο) αίτημα:
Expect
- Η επικεφαλίδα Expect σχεδιάστηκε καταρχήν για να διευκολύνει τη διαχείριση αιτημάτων PUT και POST με μεγάλα σώματα.
- Σε αίτημα με Expect, ο εξυπηρετητής απαντά με:
 - **100 Continue** αν μπορεί να χειριστεί το αίτημα
 - **413 Request Entity Too Large** αν το αίτημα είναι πολύ μεγάλο
 - **403 Forbidden** αν ο πελάτης δεν δικαιούται να κάνει αυτό το αίτημα
 - **417 Expectation Failed** αν ο εξυπηρετητής δεν ανέμενε το



Μηχανισμός expect/continue

POST /secure.txt HTTP/1.1
Content-Length: 23248
Authorization: Basic ZGufjdfJK-09=
Expect: 100-Continue

HTTP/1.1 100 Continue

HTTP/1.1 417 Expectation Failed



Μηχανισμός expect/continue

- Ο *μηχανισμός expect/continue* είναι βήμα προς βήμα.
- Η προσδοκία αφορά μόνο το επόμενο βήμα. Αν ο εξυπηρετητής του επομένου βήματος δεν μπορεί να ικανοποιήσει την προσδοκία, πρέπει να απαντήσει με μήνυμα **417 Expectation Failed**.
- Ωστόσο, η *επικεφαλίδα* αιτήματος **Expect** είναι απ' άκρου εις άκρη.
- Στην περίπτωση δηλαδή που ένας ενδιάμεσος πρέπει να επαναπροωθήσει κάποιο μήνυμα **Expect** (επειδή δεν μπορεί να το εξυπηρετήσει τοπικά), η επικεφαλίδα **Expect** *πρέπει* να σταλεί με το επαναπροωθούμενο μήνυμα.



Συμπύεση



Ανάγκη συμπίεσης στο HTTP

- Δεν αφορά εικόνες gif, jpeg (γιατί;)
- Δεν αφορά σε επικεφαλίδες – το HTTP δεν πραγματοποιεί συμπίεσεις επικεφαλίδων (σημαντικό σε μικρά μηνύματα πάνω από ασύρματους διαύλους).
- Αφορά σε μεγάλες οντότητες που μεταφέρονται από αποκρίσεις και αιτήματα HTTP (ascii, ps, pdf κλπ).
- Ο μηχανισμός **Content-Encoding** του HTTP/1.0 υποστηρίζει μόνο κωδικοποίηση απ' άκρου εις άκρη. Έτσι, δύο διαμεσολαβητές δεν μπορούν να χρησιμοποιήσουν το δικό τους πρωτόκολλο συμπίεσης ή να διαπραγματευτούν πρωτόκολλο συμπίεσης την στιγμή της αποστολής ενός μηνύματος.



Κωδικοποίηση μεταφοράς

- Ένας πελάτης μπορεί να συμπεριλάβει στα αιτήματά του επικεφαλίδα για να δηλώσει τι είδους κωδικοποίηση μεταφοράς προτιμά:
 - TE: `vdcomp;q=0.9, compress;q=0.1`
- Ο εξυπηρετητής που δέχεται τέτοιο αίτημα θα επιλέξει αλγόριθμο κωδικοποίησης τον οποίο να διαθέτει και να αντιστοιχεί στην υψηλότερη παράμετρο ποιότητας που δηλώνεται στο αίτημα.



Διαχείριση Συνδέσεων HTTP

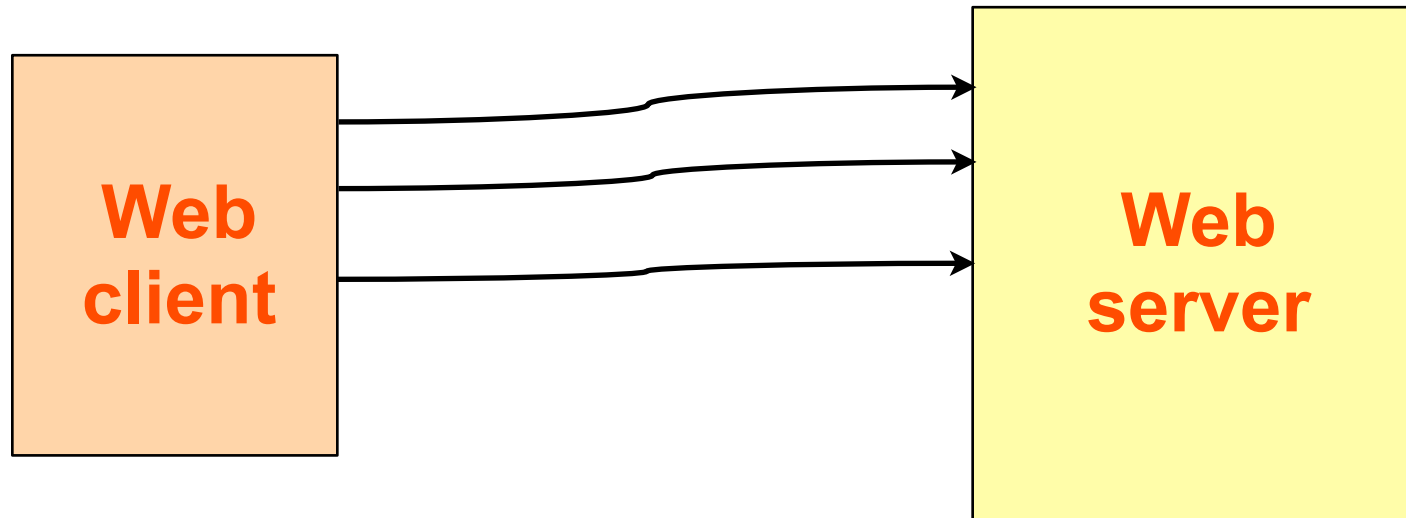


Διαχείριση Συνδέσεων στο HTTP/1.1

- Το HTTP χρησιμοποιεί συνδέσεις του TCP, οι οποίες όμως δεν είναι βελτιστοποιημένες για τις «σύντομες» δοσοληψίες του HTTP.
 - Η χρήση του TCP επιβάλλει την υλοποίηση μιας τριπλής χειραψίας (three-way handshake) για την εγκαθίδρυση μιας σύνδεσης και άλλα 4 πακέτα για το κλείσιμο της σύνδεσης.
 - Ένα μήνυμα HTTP συνήθως «χωράει» σε 10 πακέτα – άρα για ένα τυπικό μήνυμα HTTP έχουμε σπατάλη 7 πακέτων στα 17.
- Επιπλέον, τυπικά η ανάκτηση ενός ιστιακού πόρου συνεπάγεται πλέον πολλές δοσοληψίες HTTP για την ανάκτηση των ενσωματωμένων εικόνων. Για την επιτάχυνση της διαδικασίας αυτής, ορισμένοι φυλλομετρητές του HTTP/1.0 χρησιμοποίησαν *παράλληλες συνδέσεις HTTP*.



Παράλληλες TCP συνδέσεις





Επαναχρησιμοποίηση TCP συνδέσεων

- Μείωση του συνολικού κόστους των TCP συνδέσεων (λιγότερα μηνύματα για εγκαθίδρυση και διακοπή)
- Μείωση υστέρησης με αποφυγή πολλών TCP slow-starts.
- Αποφυγή της σπατάλης εύρους ζώνης και μείωση του συνολικού συνωστισμού.
- Προτεινόμενες λύσεις στο HTTP/1.1:
 - Πολλαπλά αιτήματα αποστέλλονται μέσα από μια *επίμονη σύνδεση*, χωρίς ο πελάτης να χρειάζεται να περιμένει απαντήσεις από τον εξυπηρετητή (pipelining - διασωλήνωση).
 - Μείωση του συνολικού αριθμού συνδέσεων αν οι διαμεσολαβητές χρησιμοποιούν επίμονες συνδέσεις.
 - Η επίμονη σύνδεση καθίσταται ο καθιερωμένος τρόπος σύνδεσης στο HTTP/1.1



Επίμονες Συνδέσεις

- Βασίζεται στην προσέγγιση του Επίμονου HTTP – Persistent HTTP (P-HTTP), σύμφωνα με το οποίο, μετά από μια δοσοληψία HTTP, η αντίστοιχη TCP σύνδεση δεν διακόπτεται, ούτως ώστε τυχόν αιτήματα που έπονται από τον πελάτη προς τον διαθέτη να εξυπηρετηθούν μέσα από αυτή.
- Βασικό θέμα: πελάτης και εξυπηρετητής πρέπει να γνωρίζουν τότε ένα μήνυμα έχει ολοκληρωθεί → κάθε μήνυμα πρέπει να περιλαμβάνει επικεφαλίδα τύπου Content-Length για προσδιορισμό του μήκους του μηνύματος.
- Πλεονεκτήματα: οικονομία πόρων, διασωλήνωση αιτημάτων, λιγότερα πακέτα, μεγαλύτερη αντοχή σε ασυμβατότητες HTTP.



Επίμονες Συνδέσεις στο HTTP/1.1

- Στο HTTP/1.1 όλες οι συνδέσεις που γίνονται, θεωρείται ότι είναι επίμονες. Έτσι, κατά συνθήκη οι εξυπηρετητές HTTP/1.1 κρατούν ανοικτές τις συνδέσεις, εκτός αν οι διαχειριστές τους κάνουν διαφορετική ρύθμιση.
- Εν γένει, το HTTP/1.1 δεν ασχολείται με το πότε πρέπει να κλείσει μια επίμονη σύνδεση. Ωστόσο, επιτρέπει τη χρήση της επικεφαλίδας:

Connection: close

για να επιτρέψει σε πελάτες και εξυπηρετητές να σηματοδοτούν το κλείσιμο μιας σύνδεσης, μετά το πέρας της δοσοληψίας τους.



Διασωλήνωση στο HTTP/1.1 (pipelining)

- Μέσα από μια επίμονη σύνδεση, ένας πελάτης HTTP/1.1 μπορεί να στείλει πολλά αιτήματα προς τον εξυπηρετητή.
- Αν όμως χρειάζεται να περιμένει την απόκριση ενός αιτήματος για να στείλει το επόμενο αίτημα, υπάρχει επιπλέον καθυστέρηση.
- Η αντιμετώπιση του προβλήματος αυτού γίνεται με χρήση **διασωλήνωσης** (pipelining):
 - Ο πελάτης στέλνει αιτήματα χωρίς να περιμένει την ολοκλήρωση των αποκρίσεων τους, με την υπόθεση ότι τα αιτήματά του εξυπηρετούνται με την κατάλληλη σειρά
 - Το πρωτόκολλο επιβάλλει ότι μόνο *μηδενοδύναμες* (idempotent) μέθοδοι μπορούν να χρησιμοποιηθούν σε σειρά, μέσα σε μια διασωλήνωση.
 - Idempotent methods: the side-effects of $N > 0$ identical requests is the same as for a single request. The methods GET, HEAD, PUT and DELETE share this property. Also, the methods OPTIONS and TRACE SHOULD NOT have side effects, and so are inherently idempotent.



Κλείσιμο επίμονων συνδέσεων

- Στην απόφαση για το πότε ένας εξυπηρετητής πρέπει να κλείσει μια επίμονη σύνδεση, υπάρχουν πολλοί (αντικρουόμενοι) παράγοντες που πρέπει να ληφθούν υπόψη:
 - Ο εξυπηρετητής θέλει να εξυπηρετήσει διαφορετικούς πελάτες. Κρατώντας ανοικτή μια σύνδεση, περιορίζει τους διαθέσιμους πόρους του για αιτήματα από νέους πελάτες.
 - Η διακοπή της σύνδεσης ενός πελάτη, μπορεί να συνεπάγεται την έναρξη νέας σύνδεσης από τον ίδιο πελάτη, με αναίτιο κόστος.
 - Ο εξυπηρετητής μπορεί να αποφασίσει να κλείσει μια σύνδεση όταν ο πελάτης υπερβεί ένα ανώτατο χρονικό όριο σύνδεσης (ή ένα ανώτατο χρονικό όριο ανενεργού σύνδεσης).
 - Ο εξυπηρετητής μπορεί να διατηρεί συνδέσεις με διαμεσολαβητές για μεγαλύτερο χρονικό διάστημα απ' ότι με απλούς πελάτες.



Κλείσιμο επίμονων συνδέσεων

- Προτάσεις για προσθήσεις επικεφαλίδων στο HTTP/1.1:
 - **timeout**: μέγιστη χρονική διάρκεια μιας σύνδεσης
 - **max**: μέγιστος αριθμός εξυπηρετούμενων αιτημάτων σε μια επίμονη σύνδεση
- Οι προτάσεις αυτές έχουν μειονεκτήματα, γι' αυτό και δεν υιοθετήθηκαν στο HTTP/1.1



Μετάδοση Μηνυμάτων

- Πώς αποφασίζουμε για το πότε έχει ολοκληρωθεί η αποστολή ενός μηνύματος στο HTTP;
 - Στο 1.0 με το κλείσιμο της σύνδεσης.
 - Αυτό δεν είναι δυνατόν στο HTTP/1.1 (γιατί;)
 - Αν μεταφέρεται στατικό περιεχόμενο, με χρήση της επικεφαλίδας **Content-Length**.
 - Αν έχουμε δυναμικό περιεχόμενο, ο υπολογισμός του **Content-Length** εισάγει ένα στοιχείο καθυστέρησης στις απαντήσεις (γιατί;).
- Η λύση που προτείνει το HTTP/1.1 είναι με την εισαγωγή της *κατατεμαχισμένης* κωδικοποίησης μεταφοράς (chunked encoding).



Κατατεμαχισμένη μεταφορά

- Μηχανισμός κωδικοποίησης μεταφοράς (transfer-coding) που επιτρέπει:
 - Στον αποστολέα να τεμαχίσει το σώμα ενός μηνύματος σε τεμάχια (chunks) αυθαίρετου μεγέθους, τα οποία αποστέλλονται ανεξάρτητα.
 - Κάθε τεμαχίου προηγείται αριθμός που ισούται με τους χαρακτήρες του τεμαχίου, κωδικοποιημένος σε HEX μορφή.
 - Ο αποστολέας τοποθετεί ένα τεμάχιο μηδενικού μεγέθους στο τέλος του μηνύματος, ώστε να σηματοδοτήσει προς τον παραλήπτη την ολοκλήρωση της αποστολής του μεταφερόμενου περιεχομένου.
- Η κωδικοποίηση chunked του HTTP δεν επηρεάζει το πώς μεταφέρονται δεδομένα από τη διαστρωμάτωση μεταφοράς (TCP)
- Η κατατεμαχισμένη μεταφορά μπορεί να χρησιμοποιηθεί και σε αιτήματα HTTP και σε αποκρίσεις HTTP.



Κατατεμαχισμένη μεταφορά

HTTP/1.1 200 OK

Server: Apache/1.2.7-dev

Date: Tue, 07 Jul 1998 18:21:41 GMT

Connection: Keep-Alive

Transfer-Encoding: chunked

Content-Type: text/html

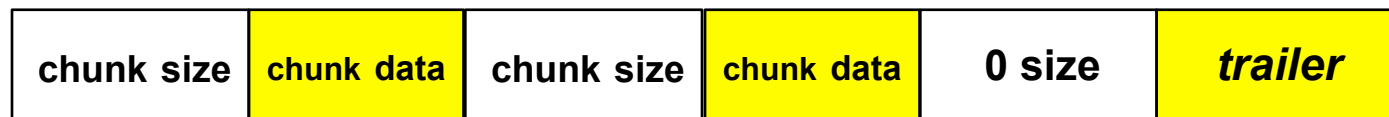
691

<... 1681 bytes of chunk data ...>

76

<... 118 bytes of chunk data ...>

0



Προαιρετική κατάληξη: περιέχει επικεφαλίδα οντότητας, όπως πχ Content-MD5 δηλ. «αποτύπωμα» της οντότητας σε MD5



Επικεφαλίδα Ουράς (trailer header)

- Στην περίπτωση χρήσης κωδικοποίησης κατατεμαχισμού, υπάρχει η δυνατότητα στο τέλος του μηνύματος να τοποθετηθεί μια “ουρά” (trailer), η οποία να μεταφέρει επιπλέον επικεφαλίδες και τιμές.
- Η ύπαρξη της ουράς σηματοδοτείται με την χρήση της επικεφαλίδας trailer στην αρχή του μηνύματος.
- Με την επικεφαλίδα TE μπορεί κάποιος αποστολέας να υποδείξει ότι αποδέχεται μηνύματα με ουρές:

```
GET /foo.html HTTP/1.1  
Host: www.checkt.com  
TE: trailers
```

```
HTTP/1.1 200 OK  
Trailer: Splinfo  
Transfer-encoding: chunked  
CRLF  
691  
    <1681 bytes >  
0  
Splinfo: vol=7; pe= “u4,9998,5466”  
CRLF
```




Υποστήριξη Επεκτασιμότητας



Τρόποι υποστήριξης επεκτασιμότητας στο HTTP/1.1

- Δυνατότητα διερεύνησης των δυνατοτήτων ενός εξυπηρετητή πριν την αποστολή ενός αιτήματος προς αυτόν.
- Δυνατότητα ελέγχου του μηνύματος που παραδίδεται σε έναν εξυπηρετητή με την αποστολή ενός αιτήματος από κάποιον πελάτη.
- Εξερεύνηση των διαμεσολαβητών που υπεισέρχονται στην αλληλεπίδραση μεταξύ ενός πελάτη κι ενός εξυπηρετητή.
- Δυνατότητα αναβάθμισης της επικοινωνίας σε κάποιο άλλο πρωτόκολλο.



Μέθοδοι Αιτημάτων HTTP/1.1: OPTIONS

- **OPTIONS:** Επιτρέπει σε έναν πελάτη να λάβει πληροφορίες σχετικά με τις δυνατότητες εξυπηρετητών ή διαμεσολαβητών.
- Π.χ:
 - Ποιές μεθόδους αποδέχεται ο εξυπηρετητής.
 - Αν υπάρχουν συγκεκριμένες απαιτήσεις συνδυασμένες με κάποιο πόρο.
 - Ποιές είναι οι δυνατότητες ενός διαμεσολαβητή στο μονοπάτι μεταξύ του πελάτη και ενός εξυπηρετητή.



Μέθοδοι Αιτημάτων HTTP/1.1: OPTIONS

- Για να ελέγξει ένας πελάτης ποιές μεθόδους υποστηρίζει ένας εξυπηρετητής :

Πελάτης: **OPTIONS * HTTP/1.1**
 Host: foo.com

Εξυπηρετητής: **HTTP/1.1 200 OK**
 Allow: HEAD, GET, POST, TRACE, OPTIONS

(αν αντί “” τοποθετήσαμε κάποιο URI, θα ζητούσαμε πληροφορίες για τις μεθόδους που υποστηρίζονται στο συγκεκριμένο URI)*

- Για να διασφαλίσουμε την έγερση ενός τέτοιου αιτήματος σε κάποιο διαμεσολαβητή, χρησιμοποιείται το Max-Forwards.

OPTIONS /bar HTTP/1.1
Host: foo.com
User-Agent: Mozilla/2.0
Max-Forwards: 1



Μέθοδοι Αιτημάτων HTTP/1.1: TRACE

- Η TRACE επιτρέπει σε έναν πελάτη να λαμβάνει τα περιεχόμενα του μηνύματος που ο ίδιος έστειλε σε κάποιον παραλήπτη (μηχανισμός loopback).

```
TRACE /bar HTTP/1.1
Host: foo.com
User-agent: Mozilla/2.0
```

```
HTTP/1.1 200 OK
Content-type: message/
http
```

```
TRACE /bar HTTP/1.1
Host: foo.com
User-agent: Mozilla/2.0
```



Επικεφαλίδα Via

- Στο HTTP/1.1, όλοι οι ενδιάμεσοι (proxies) και οι πύλες (gateways) είναι **υποχρεωμένοι** να προσθέτουν πληροφορίες ταυτοποίησής για τον εαυτό τους και τον εξυπηρετητή από τον οποίο παραλαμβάνουν μηνύματα.
- Αυτό γίνεται μέσω του μηχανισμού Via.
- Οι διαμεσολαβητές είναι υποχρεωμένοι να επικολλήσουν:
 - το **όνομα πεδίου** τους
 - την **έκδοση του πρωτοκόλλου** του εξυπηρετητή από τον οποίο προήλθε το μήνυμα που λαμβάνουν

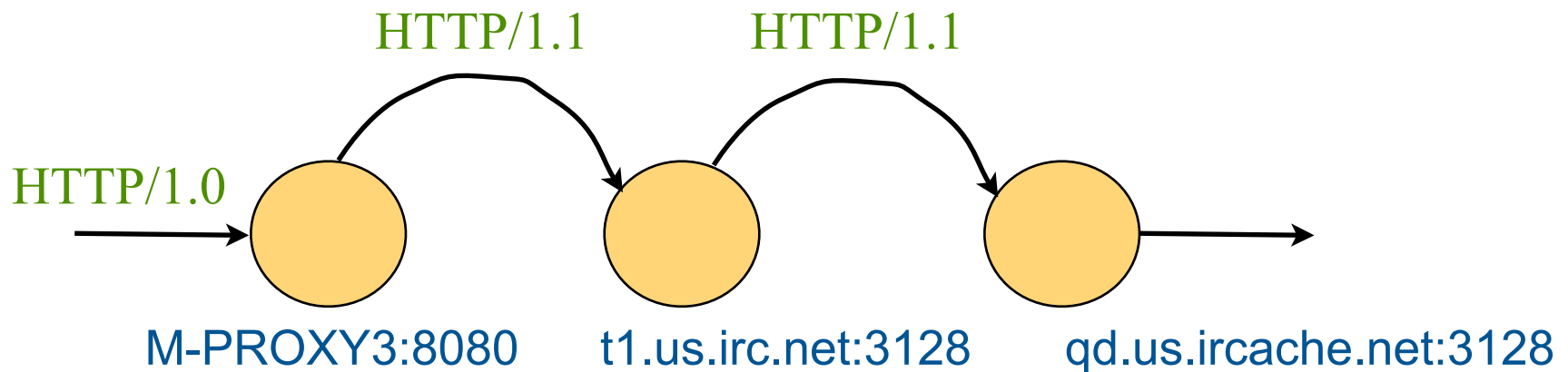
στο περιεχόμενο της γενικής επικεφαλίδας Via, πριν προωθήσουν το μήνυμα στον τελικό παραλήπτη.

- Έτσι, ο τελικός παραλήπτης έχει τη δυνατότητα να γνωρίζει για τις στάσεις του μηνύματος στον δρόμο του από την αφετηρία σε αυτόν.



Παράδειγμα χρήσης Via

Via: 1.0 M-PROXY3:8080 (Squid/2.1.PATCH2),
1.1 t1.us.irc.net:3128 (Squid/2.3.DEVEL1),
1.1 qd.us.ircache.net:3128 (Squid/2.3.DEVEL1)





TRACE και VIA

- Η μέθοδος TRACE μπορεί να χρησιμοποιηθεί σε συνδυασμό με:
 - την επικεφαλίδα VIA
 - την επικεφαλίδα MAX-FORWARDS

ώστε να μπορούμε να ανακαλύψουμε το μονοπάτι που ακολουθεί ένα HTTP μήνυμα από έναν πελάτη σε κάποιο εξυπηρετητή.



Αναβάθμιση πρωτοκόλλου

- Η β-β επικεφαλίδα Upgrade επιτρέπει στον αποστολέα ενός μηνύματος να καθορίσει εναλλακτικά πρωτόκολλα επικοινωνίας, για συνέχιση της επικοινωνίας πάνω από την ίδια σύνοδο μεταφοράς.
- Π.χ.:

```
GET http://foo.com/priv HTTP/1.1  
Host: www.goo.org  
Upgrade: SafeBank/1.0  
Connection: Upgrade
```

```
HTTP/1.1 Switching Protocols  
Upgrade: SafeBank/1.0  
Connection: Upgrade  
<response for client's request>
```



Καθορισμός ονόματος πεδίου

- Υποστηρίζει το πεδίο επικεφαλίδας **Host** για την υποστήριξη πολλαπλών εικονικών κόμβων μη βασισμένων στο IP (non-IP based virtual hosts). Η επικεφαλίδα Host χρησιμοποιείται από έναν πελάτη για να καθορίσει σε ποιόν κόμβο στέλνεται ένα αίτημα HTTP. Αιτήματα χωρίς επικεφαλίδα Host λαμβάνουν ως απάντηση μήνυμα λάθους.
 - **Host: www.ninja.com**
 - Οι πελάτες ΠΡΕΠΕΙ να στέλνουν αυτή την επικεφαλίδα.
 - Υλοποιείται σχεδόν σε όλους τους πελάτες HTTP/1.0, HTTP/1.1
- **Αποδοχή απόλυτων URI σε αιτήματα:** το HTTP 1.0 δέχεται απόλυτα URI μόνο όταν το αίτημα στέλνεται προς κάποιον διαμεσολαβητή-proxy (γιατί;).



Διαπραγμάτευση περιεχομένου



Διαπραγμάτευση Περιεχομένου στο HTTP/1.1

- Προκύπτει σε περιπτώσεις όπου διαφορετικές εκδοχές του ίδιου πληροφοριακού πόρου υπάρχουν σε κάποιον διαθέτη. Κανονικά, οποιαδήποτε αναφορά (URI) παραπέμπει σε αυτόν τον πόρο, θα πρέπει να αναφέρεται σε όλες τις εκδοχές του.
- Κατηγορίες εκδοχών:
 - Εξαρτώμενες από την γλώσσα.
 - Εξαρτώμενες από την ποιότητα: π.χ. η ίδια εικόνα αλλά με χαμηλή ευκρίνεια παρέχεται σε χρήστες με αργές συνδέσεις.
 - Εξαρτώμενες από την δυνατότητα αποκωδικοποίησης πληροφοριών του πελάτη.
- Το HTTP/1.1 προσδιορίζει διαδικασίες *διαπραγμάτευσης περιεχομένου* για χρήση από έναν πελάτη και έναν εξυπηρετητή ώστε να προσδιοριστεί ποια εκδοχή ενός αιτούμενου πόρου θα χρησιμοποιηθεί για την ικανοποίηση συγκεκριμένου αιτήματος.
- Διαπραγμάτευση περιεχομένου:
 - *Καθοδηγούμενη από τον εξυπηρετητή (server-driven)*: ο εξ. είναι υπεύθυνος για να επιλέξει την συγκεκριμένη αναπαράσταση ενός πόρου
 - *Καθοδηγούμενη από τον διεκπεραιωτή (agent-driven)*: η ευθύνη της επιλογής μετακινείται προς τον πελάτη.



Διαπραγμάτευση καθοδηγούμενη από εξυπηρετητή

- Ο εξυπηρετητής χρησιμοποιεί για ν' αποφασίσει:
 - Διαθέσιμες αναπαραστάσεις του περιεχομένου
 - Στοιχεία από επικεφαλίδες αιτημάτων HTTP (πχ Accept, Accept-Charset, Accept-Encoding, Accept-Language, User-Agent).
 - Άλλες πληροφορίες (πχ διεύθυνση IP του πελάτη).
- Μειονεκτήματα:
 - Περιορισμένη γνώση εξ. για πελάτες
 - Χαμηλή αποδοτικότητα (αποθήκευση πολλών εκδόχων)
 - Περίπλοκη υλοποίηση εξ.
 - Άρση απομνημόνευσης λόγω δυναμικής δημιουργίας απόκρισης



Διαπραγμάτευση καθοδηγούμενη από πελάτη

- Ο πηγαίος εξυπηρετητής αποκρίνεται στον πελάτη με μια λίστα από όλες τις διαθέσιμες αναπαραστάσεις. Ο μόνος τρόπος για να γίνει αυτό στην τρέχουσα έκδοση του HTTP είναι με μήνυμα **300 (multiple choices)**.
- Ο πελάτης επιλέγει την επιθυμητή και αποστέλλει στον εξ. το αντίστοιχο αίτημα.
- Ωστόσο, καθώς η προδιαγραφή του HTTP δεν προσδιορίζει την μορφή της λίστας των διαθέσιμων αναπαραστάσεων, η επιλογή από τον πελάτη δεν μπορεί να γίνει αυτόματα (χρειάζεται επέμβαση του χρήστη).



Επικεφαλίδες Οντότητας

- **Allow:** Χρησιμεύει για τον καθορισμό των μεθόδων HTTP που υποστηρίζονται για κάποιο συγκεκριμένο πόρο. Υποχρεωτική η χρήση του στις αποκρίσεις εξυπηρετητών με κωδικό κατάστασης: **405 Method not allowed**.
- **Content-Language:** Προσδιορισμός γλώσσας περιεχομένου.
- **Content-Type:** Περιγράφει το μέσο κωδικοποίησης (media type) της μεταφερόμενης οντότητας. Π.χ: **Content-type: text/html** Η οντότητα αυτή, ωστόσο, μπορεί να μεταφέρεται σε διαφορετική κωδικοποίηση, λόγω επιπλέον κωδικοποιήσεων (οπότε χρησιμοποιείται το Content-Encoding).
- **Content-Encoding:** Περιγράφει την κωδικοποίηση που έχει γίνει στο περιεχόμενο της μεταφερόμενης οντότητας. Π.χ **Content-Encoding: x-gzip**



Επικεφαλίδες Οντότητας (συνέχεια)

- **Content-Range:**

- Αν η δοσοληψία HTTP αφορά σε μεταφορά τμήματος μιας οντότητας, η επικεφαλίδα Content-Range χρησιμεύει για τον καθορισμό της θέσης και του μεγέθους του μεταφερόμενου τμήματος σε σχέση με ολόκληρο το αρχείο της οντότητας.
- **Range: bytes=300-304,601-993**
- Αυτό το χαρακτηριστικό είναι ιδιαίτερα χρήσιμο όταν «κατεβάζουμε» μια πολύ μεγάλη οντότητα, διακόπτουμε την φόρτωση και θέλουμε αργότερα να συνεχίσουμε από το σημείο στο οποίο είχαμε σταματήσει (όπως στο FTP recovery)

- **Content-Location:**

- Αν η οντότητα την οποία ζητάει ο πελάτης βρίσκεται σε διαφορετική τοποθεσία από το αιτούμενο URI, ο εξυπηρετητής καθορίζει την τοποθεσία αυτή με την επικεφαλίδα Content-Location.
- Χρησιμεύει όταν ο ίδιος πληροφοριακός πόρος υλοποιείται σαν περισσότερες από μια οντότητες, οι οποίες βρίσκονται σε περισσότερες της μιας τοποθεσίες (π.χ. διαφορετικές μεταφράσεις της ίδιας ιστοσελίδας) – οπότε ο εξυπηρετητής επιλέγει για την απόκρισή του μία από τις οντότητες.
- Δεν καθορίζει μια οντότητα-υποκατάστατο για το αιτούμενο URI, αλλά μπορεί να χρησιμοποιηθεί σε μελλοντικά αιτήματα του πελάτη.

- **ETag:**

- Περιλαμβάνει την επιγραφή που χρησιμοποιείται για την επικύρωση της οντότητας.



Οι διαφορές HTTP/1.0 και HTTP/1.1

- **Καθορισμός πλήρους ονόματος πεδίου σε κάθε αίτημα.**
- **Νέες μέθοδοι αιτημάτων** (επέκταση λειτουργικότητας): DELETE, OPTIONS, PUT, TRACE.
- **Υποστήριξη της μεταφοράς τμήματος οντότητας:** συστήματα πελάτων μπορούν να καθορίσουν συγκεκριμένο απόσπασμα χαρακτήρων μίας οντότητας, το οποίο επιθυμούν να παραλάβουν από τον διαθέτη.
- **Διαπραγμάτευση περιεχομένου:** πελάτης και εξυπηρετητής μπορούν να «διαπραγματευθούν» για την επιλογή μίας από ένα σύνολο διαφορετικών αναπαραστάσεων του ίδιου πληροφοριακού πόρου. Διαφορετικές αναπαραστάσεις προσδιορίζονται από την χρησιμοποιούμενη γλώσσα, την ποιότητα της πληροφορίας, την κωδικοποίησή της κλπ.



Οι διαφορές HTTP/1.0 και HTTP/1.1 (συνέχεια)

- **Ασφαλέστερο σχήμα ταυτοποίησης και εξουσιοδότησης (authentication).**
- **Πιο προχωρημένο μοντέλο υποστήριξης απομνημόνευσης.**
 - **Age: <seconds, date>**
 - expiration from caches
 - **Cache-control: <command>**
 - Χαρακτηρίζει αρχεία σαν ιδιωτικά και αποτρέπει την αποθήκευσή τους σε λανθάνουσες μνήμες