ΕΠΙΠΕΔΟ ΕΦΑΡΜΟΓΗΣ
DNS, HTTP & SMTP

Τμήμα Πληροφορικής, Σχολή Θετικών Επιστημών,
Παν/μιο Θεσσαλίας

Εισαγωγή

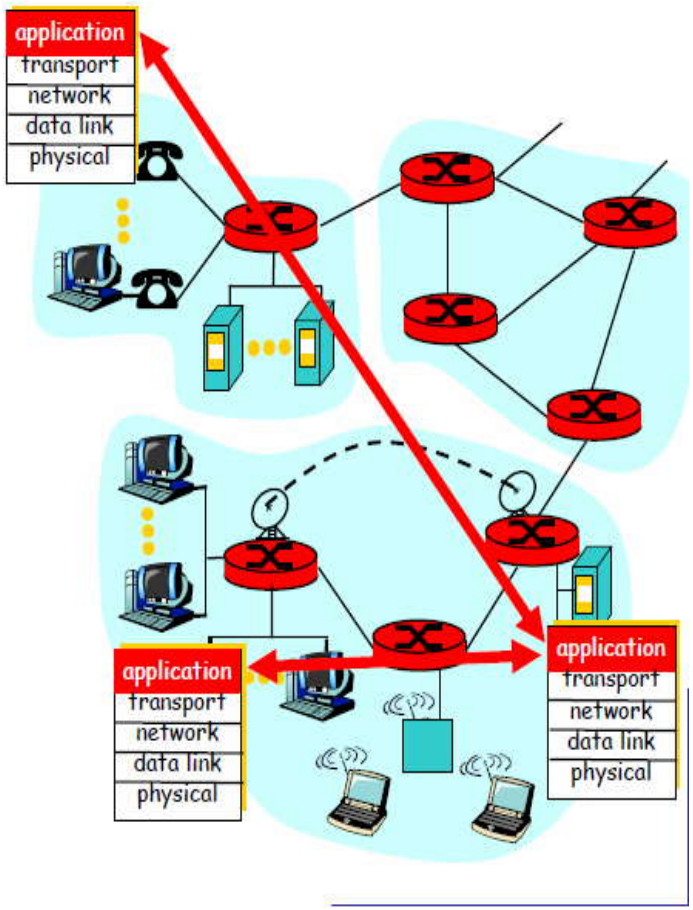
- Διεργασία: πρόγραμμα που εκτελείται σε ένα σύστημα
- Μέσα στο ίδιο σύστημα, δύο διεργασίες επικοινωνούν χρησιμοποιώντας δια-διεργασιακή επικοινωνία (IPC, ορίζεται από το ΛΣ)
- Διεργασίες που εκτελούνται σε διαφορετικά συστήματα επικοινωνούν με ένα πρωτόκολλο επιπέδου εφαρμογής

user agent: παρέχει τη διεπαφή του χρήστη με το δίκτυο

- Υλοποιεί τη διεπαφή του χρήστη και το πρωτόκολλο επιπέδου εφαρμογής
  - ✓ Web: browser
  - ✓ E-mail: mail reader
  - ✓ streaming audio/video: media player

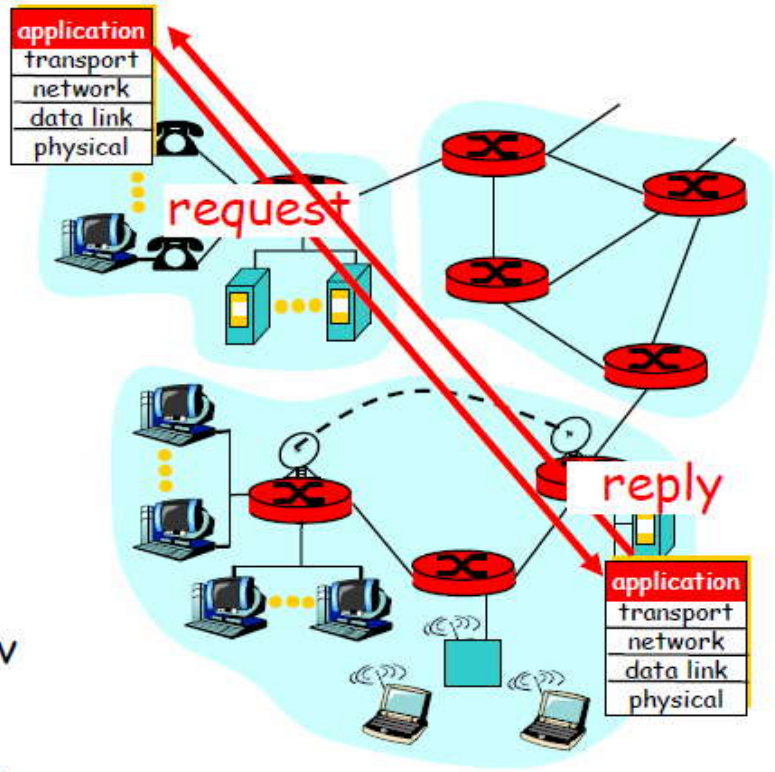
Εφαρμογές και πρωτόκολλα επιπέδου εφαρμογής

- Εφαρμογή: κατανεμημένες διεργασίες που επικοινωνούν
  - ✓ π.χ., e-mail, Web, διαμοιρασμός αρχείων P2P, ανταλλαγή μηνυμάτων
  - ✓ εκτελούνται σε τελικά συστήματα (hosts)
  - ✓ ανταλλάσσουν μηνύματα για την υλοποίηση της εφαρμογής
- Πρωτόκολλα επιπέδου εφαρμογής
  - ✓ ένα "τμήμα" μιας εφαρμογής
  - ✓ ορίζει τα μηνύματα που ανταλλάσσονται από τις εφαρμογές και τις ενέργειες που γίνονται
  - ✓ χρησιμοποιεί υπηρεσίες επικοινωνιών που παρέχονται από τα πρωτόκολλα του υφιστάμενου επιπέδου (TCP, UDP)



Μοντέλο πελάτη–εξυπηρετητή

Η τυπική δικτυακή εφαρμογή έχει δύο τμήματα: *πελάτη* και *εξυπηρετητή*



Πελάτης:

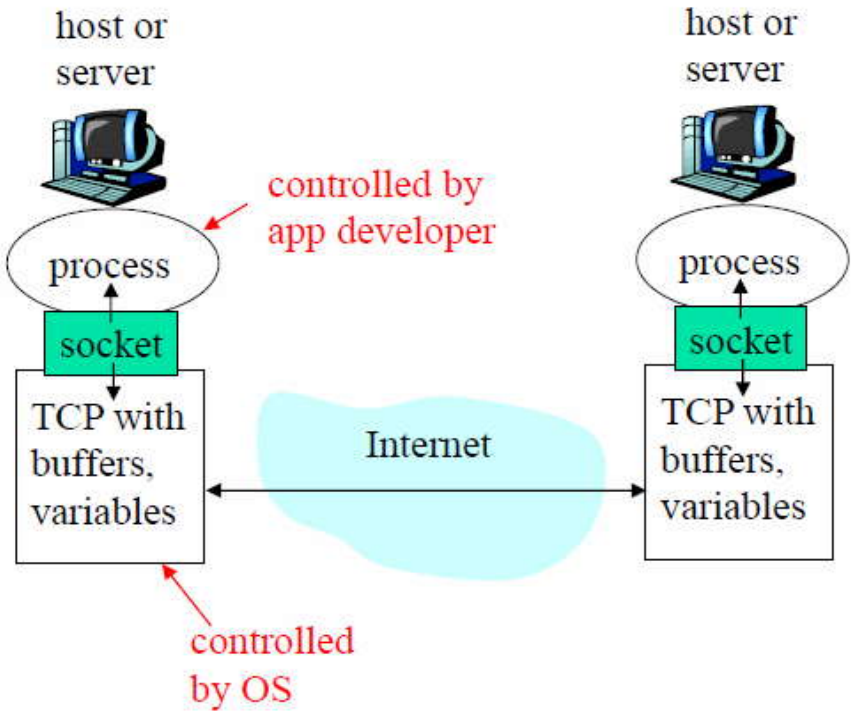
- Ξεκινάει την επικοινωνία με τον εξυπηρετητή
- Συνήθως ζητά κάποια από τις υπηρεσίες του εξυπηρετητή
- Web: ο πελάτης υλοποιείται στον browser

Εξυπηρετητής:

- Παρέχει την υπηρεσία που ζητήθηκε από τον πελάτη
- π.χ., ο εξυπηρετητής Web στέλνει τη σελίδα που ζητήθηκε

Επικοινωνία διεργασιών μέσω δικτύου

- Η διεργασία στέλνει/λαμβάνει μηνύματα προς/από την υποδοχή (socket) της
- Η υποδοχή ανάλογη της πόρτας:
  - ✓ ο αποστολέας τοποθετεί τα μηνύματα έξω από την πόρτα
  - ✓ η διεργασία αποστολέας υποθέτει την ύπαρξη μηχανισμού μεταφοράς που θα μεταφέρει το μήνυμα στην πόρτα της διεργασίας παραλήπτη



- API: (1) επιλογή πρωτοκόλλου μεταφοράς; (2) δυνατότητα καθορισμού τιμών σε παραμέτρους (περισσότερα στη συνέχεια...)

Αναγνώριση διεργασιών

- Για να μπορεί να λάβει μηνύματα μια διεργασία, πρέπει να υπάρχει τρόπος προσδιορισμού της
- Κάθε σύστημα έχει μια μοναδική διεύθυνση IP 32 bit
- Ερώτηση: Αρκεί η διεύθυνση IP ενός συστήματος για τον προσδιορισμό μιας διεργασίας;
- Απάντηση: όχι, πολλές διεργασίες μπορεί να εκτελούνται στο ίδιο σύστημα
- Ο τρόπος προσδιορισμού μιας διεργασίας στο Διαδίκτυο περιλαμβάνει τόσο τη διεύθυνση IP όσο και τους αριθμούς θυρών (port numbers) που λαμβάνουν οι διεργασίες από το σύστημα
- Παραδείγματα αριθμών θυρών:
  - ✓ HTTP server: 80
  - ✓ mail server: 25

(Περισσότερα στη συνέχεια...)

Τι Υπηρεσία μεταφοράς χρειάζεται μια εφαρμογή;

Απώλεια δεδομένων

- Ορισμένες εφαρμογές (π.χ., ήχος) αντέχουν απώλειες
- Άλλες εφαρμογές (π.χ., μεταφορά αρχείων, telnet) απαιτούν 100% αξιόπιστη μεταφορά δεδομένων

Χρονισμός

- Ορισμένες εφαρμογές (π.χ., διαδικτυακή τηλεφωνία, διαδραστικά παιχνίδια) απαιτούν χαμηλή καθυστέρηση

Εύρος Ζώνης

- Ορισμένες εφαρμογές (π.χ., πολυμέσα) απαιτούν μια ελάχιστη ποσότητα εύρους ζώνης να είναι διαθέσιμη
- Άλλες εφαρμογές κάνουν χρήση όσου εύρους ζώνης διαθέτουν

Απαιτήσεις των εφαρμογών από την υπηρεσία μεταφοράς

Εφαρμογή	Απώλεια	Εύρος Ζώνης	Ευαισθησία στο Χρόνο
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5kbps-1Mbps video:10kbps-5Mbps	yes, 100's msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few kbps up	yes, 100's msec
instant messaging	no loss	elastic	yes and no

DNS: Domain Name System

- Άνθρωποι: χρησιμοποιούν πολλούς τρόπους προσδιορισμού:
  - ✓ Αριθμό ταυτότητας, ον/μο
- Συστήματα και δρομολογητές στο Διαδίκτυο χρησιμοποιούν:
  - ✓ διευθύνσεις IP (32 bit) – ιεραρχικές, προσδιορίζουν ένα δίκτυο στο Δίκτυο των Δικτύων
  - ✓ ονόματα, π.χ., www.teilam.gr – χρησιμοποιούνται από ανθρώπους
- Ερώτηση: αντιστοίχιση μεταξύ διεύθυνσης IP και ονόματος;

Σύστημα Ονομάτων Περιοχών:

- Κατανεμημένη βάση δεδομένων που υλοποιείται ως ιεραρχία πολλών εξυπηρετητών ονομάτων (name servers)
- Πρωτόκολλο επιπέδου εφαρμογής επιτρέπει σε συστήματα και εξυπηρετητές ονομάτων να επικοινωνούν για να γίνεται μετάφραση από όνομα σε διεύθυνση
  - ✓ σημείωση: βασική λειτουργία του Internet, που υλοποιείται ως πρωτόκολλο επιπέδου εφαρμογής

Αρχείο hosts.txt

- Στο ARPANET κάθε κόμβος, σε αυτό το αρχείο, κρατούσε τις αντιστοιχίσεις ονομάτων με διευθύνσεις
- Κάθε προσθήκη νέου κόμβου, συνεπαγόταν την ενημέρωση ενός κεντρικού αντιγράφου του αρχείου και στη συνέχεια τη μεταφορά του με FTP σε όλους τους κόμβους του δικτύου
- Για μερικές εκατοντάδες κόμβους δουλεύει, μετά (Internet) τι γίνεται;
- Η απάντηση είναι το Σύστημα Ονομάτων Περιοχών (Domain Name System, RFCs 1034 και 1035)

Συστατικά της Υπηρεσίας DNS

- Χώρος ονομάτων περιοχών και εγγραφές πόρων (Domain name space and resource records)
  - ✓ καθορίζει τον ιεραρχικό χώρο ονομάτων και τα δεδομένα που σχετίζονται με τους πόρους που τηρούνται μέσα σε αυτόν
  - ✓ ερωτήματα για κόμβους εξάγουν πληροφορία από τις εγγραφές
- Εξυπηρετητές ονομάτων (Name servers)
  - ✓ εξυπηρετητές που κρατούν την πληροφορία για τη δομή του χώρου ονομάτων και τα δεδομένα των πόρων που περιέχουν
- Resolvers
  - ✓ προγράμματα που ανακτούν πληροφορία από τους εξυπηρετητές ονομάτων (clients)

Υπηρεσίες του DNS

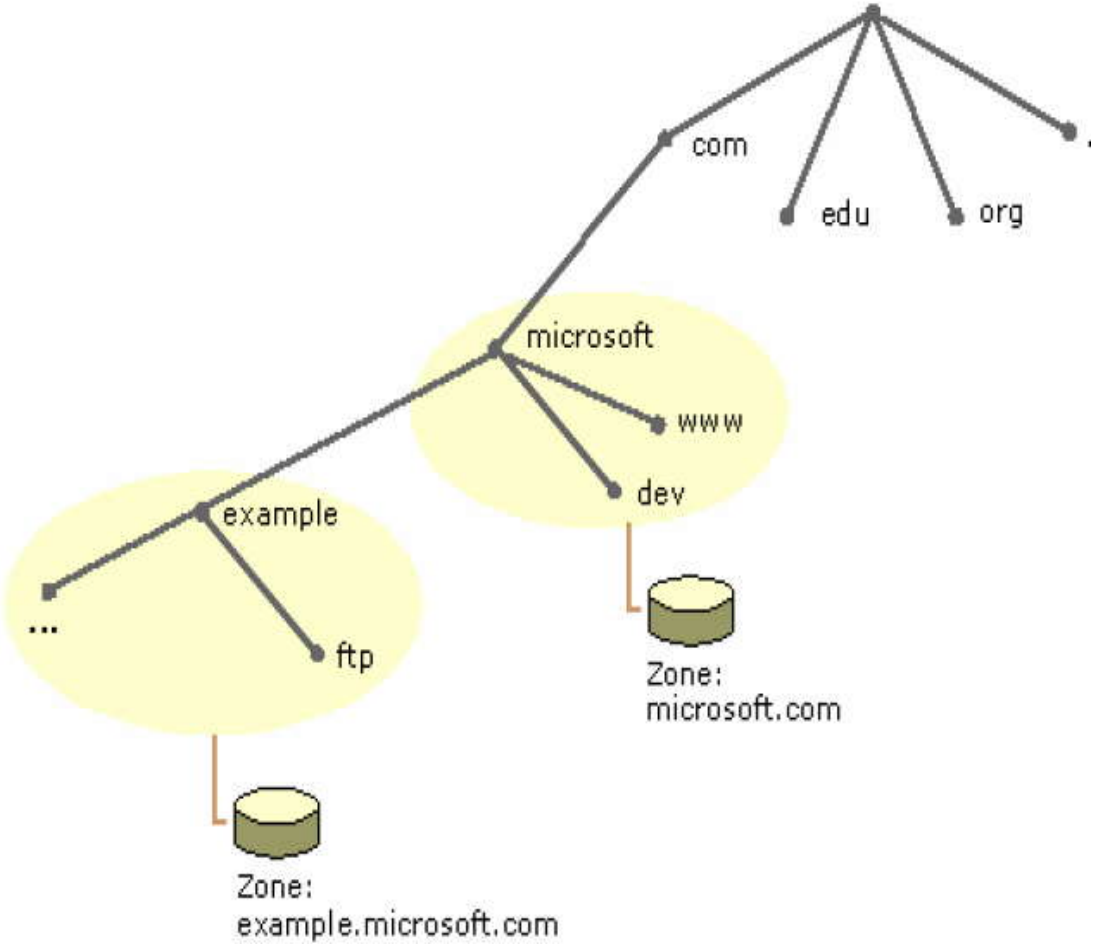
- Ανάκτηση διευθύνσεων IP από ονόματα (και αντίστροφα)
- Πληροφορίες για τους πόρους του κόμβου (υλικό, ΛΣ, πρωτόκολλα και υπηρεσίες που λειτουργούν)
- Τήρηση ψευδωνύμων για υπολογιστικά συστήματα (host aliasing)
- Τήρηση ψευδωνύμων για εξυπηρετητές ηλεκτρονικού ταχυδρομείου (mail server aliasing)
- Κατανομή φόρτου (load distribution) μεταξύ εξυπηρετητών που τρέχουν αντίγραφα μιας υπηρεσίας

Ο Χώρος Ονομάτων Περιοχών (Domain Name Space) (Ι)

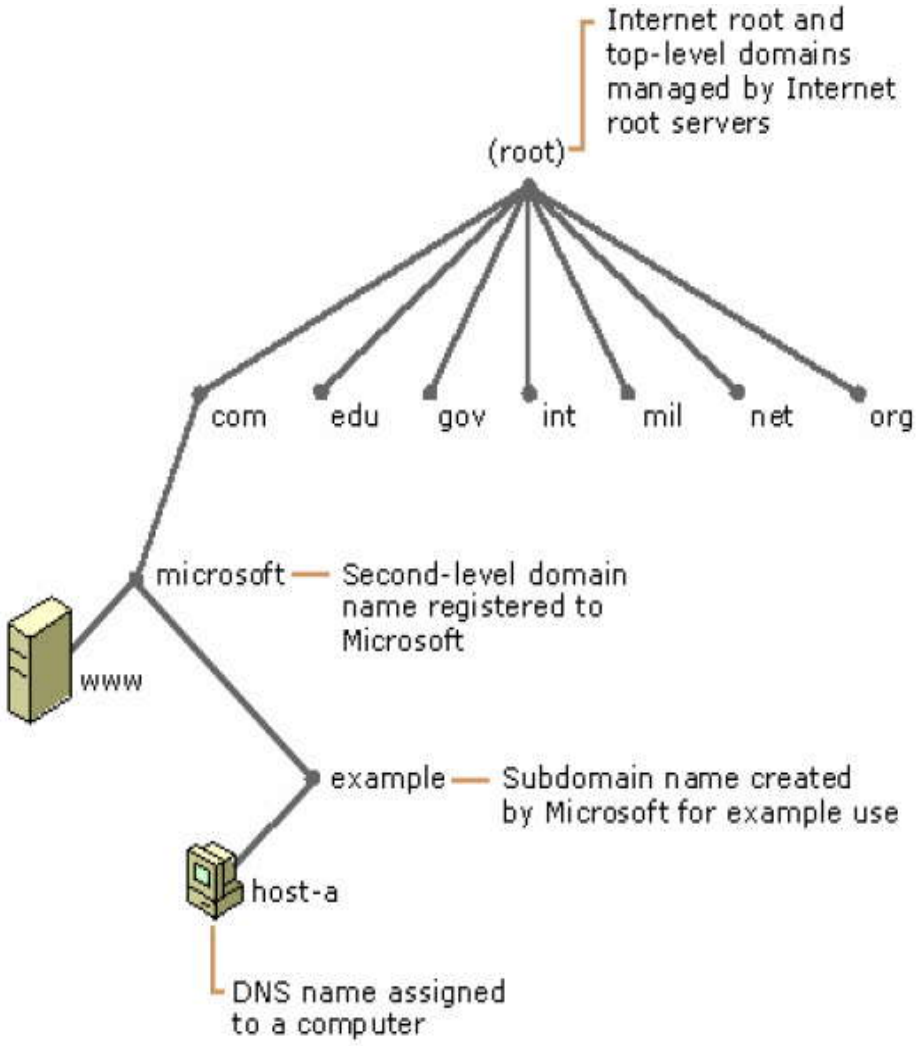
- Ο χώρος ονομάτων περιοχών είναι δομημένος ως ιεραρχική δενδρική δομή
- Η περιοχή (domain) ξεκινάει από κάποιο κόμβο του δένδρου και περιλαμβάνει όλους τους κόμβους που βρίσκονται κάτω από αυτόν
- Ζώνη (Zone): Πρόκειται για τμήμα του συνολικού συστήματος ονομάτων περιοχών για το οποίο έχει εκχωρηθεί η ευθύνη διοίκησης (delegation of authority)
- Μεταφορά ζώνης (zone transfer) είναι η δημιουργία αντιγράφων και ο συγχρονισμός των δεδομένων μιας ζώνης μεταξύ των εξυπηρετητών ονομάτων της ζώνης

Ζώνες και Περιοχές

- Οι ζώνες είναι διοικητικές περιοχές που μπορεί να δημιουργηθούν «κάτω» από ένα όνομα περιοχής
- Στο σχήμα η αρχική περιοχή microsoft.com χωρίζεται σε δύο ζώνες (microsoft.com και example.microsoft.com)



Ο Χώρος Ονομάτων Περιοχών



- Δενδρική δομή ονομάτων
- Top Level Domains: γενικού σκοπού, γεωγραφικά
- Second level domains: (μεταβίβαση αρμοδιοτήτων)
- Fully Qualified Domain Name (FQDN): η ακολουθία των ονομάτων από τον κόμβο ως τη ρίζα

Ο Χώρος Ονομάτων Περιοχών

- Η διαχείριση των περισσότερων περιοχών υψηλού επιπέδου μεταβιβάζεται σε κατάλληλους φορείς ή οργανισμούς (delegation of authority) από την ICANN (Internet Corporation for Assigned Names and Numbers)
  - ✓ η ICANN λειτουργεί την IANA (Internet Assigned Numbers and Authority) και,
  - ✓ είναι υπεύθυνη για την τήρηση της ζώνης ρίζας του DNS (DNS root zone)

Εξυπηρετητές Ονομάτων DNS

Γιατί όχι κεντρικοποιημένο Σύστημα Ονομάτων Περιοχών;

- Ένα σημείο αστοχίας (single point of failure)
- Ένταση κίνησης
- Απομακρυσμένη και κεντρικοποιημένη βάση δεδομένων
- Συντήρηση

Δεν λειτουργεί καλά για μεγάλους όγκους πληροφορίας!

Εξυπηρετητές Ονομάτων DNS

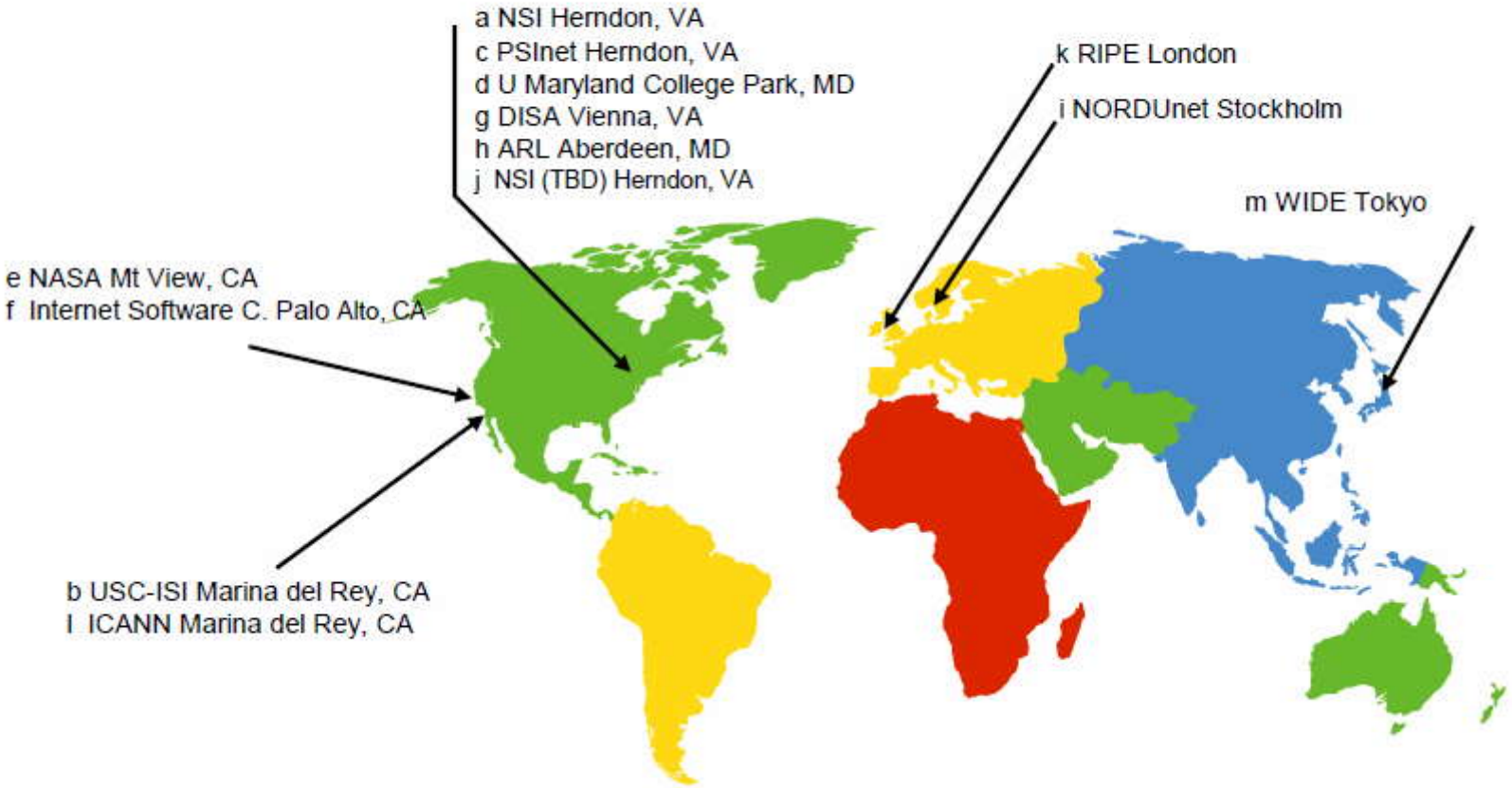
- Κανένας εξυπηρετητής δεν έχει όλες τις αντιστοιχίσεις IP και ονόματος
- Τοπικός εξυπηρετητής ονομάτων (ΕΟ):
  - ✓ κάθε πάροχος υπηρεσιών Internet έχει έναν τοπικό εξυπηρετητή ονομάτων
  - ✓ κάθε ερώτημα DNS από ένα σύστημα πηγαίνει πρώτα στον τοπικό εξυπηρετητή ονομάτων
- Έγκυρος (authoritative) εξυπηρετητής ονομάτων:
  - ✓ για ένα σύστημα: αποθηκεύει την IP και το όνομά του
  - ✓ μπορεί να κάνει αντιστοιχίσεις από όνομα σε διεύθυνση για το όνομα αυτού του συστήματος

DNS: κομβικοί εξυπηρετητές ονομάτων

- Ο τοπικός εξυπηρετητής ονομάτων που δεν μπορεί να αντιστοιχίσει το όνομα σε διεύθυνση επικοινωνεί με τον κομβικό εξυπηρετητή ονομάτων (root name server)
- Ο κομβικός εξυπηρετητής ονομάτων:
  - ✓ επικοινωνεί με τον έγκυρο εξυπηρετητή ονομάτων αν δεν γνωρίζει την αντιστοίχιση του ονόματος σε διεύθυνση
  - ✓ λαμβάνει την αντιστοίχιση
  - ✓ επιστρέφει την αντιστοίχιση στον τοπικό εξυπηρετητή ονομάτων

DNS: κομβικοί εξυπηρετητές ονομάτων

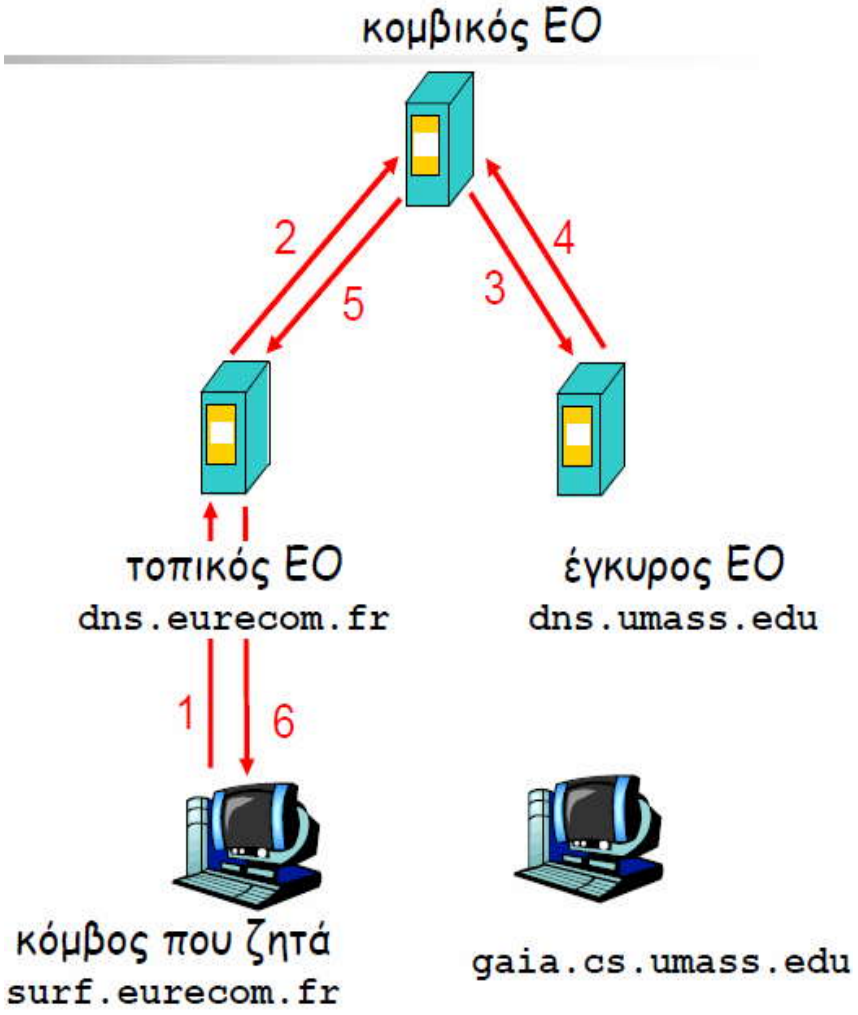
Υπάρχουν 13 κομβικοί εξυπηρετητές ονομάτων σε ολόκληρο τον κόσμο



Απλό παράδειγμα DNS (I)

Ο κόμβος **surf.eurecom.fr** θέλει τη διεύθυνση IP του **gaia.cs.umass.edu**

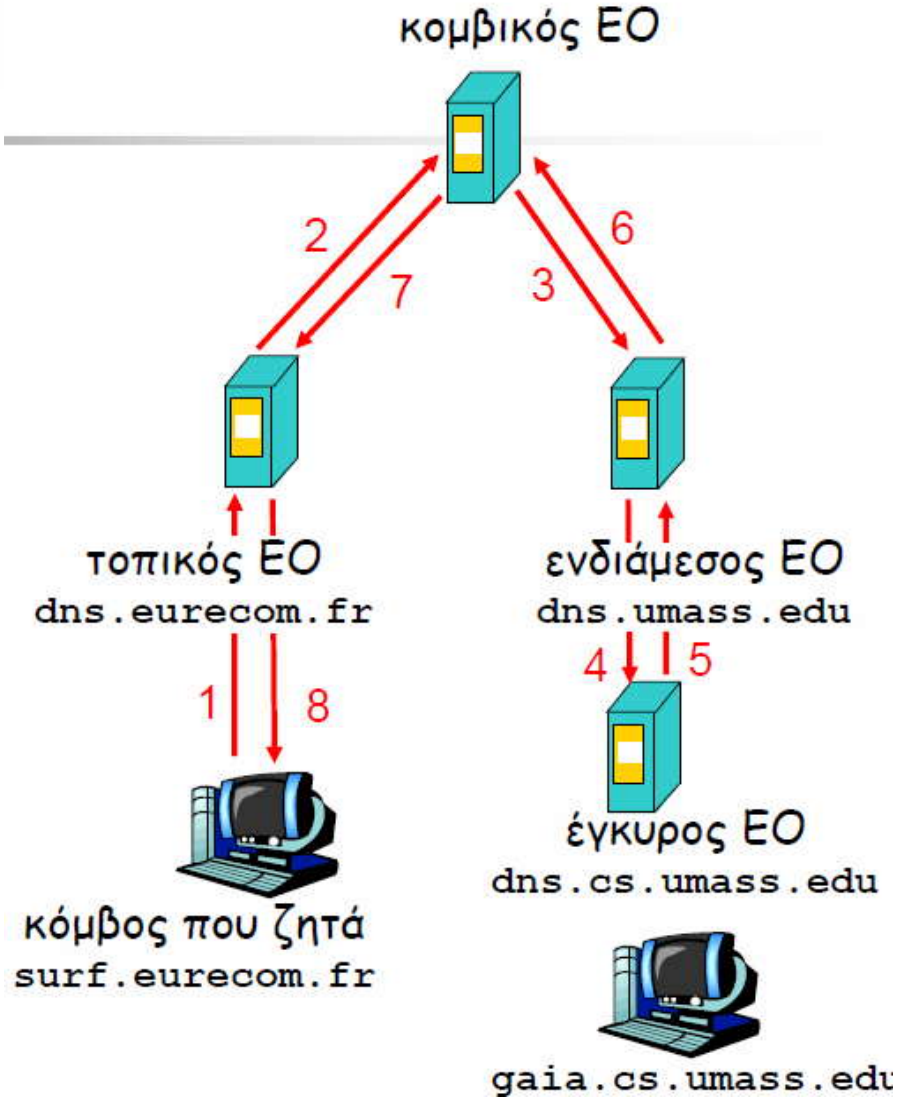
1. επικοινωνεί με τον τοπικό ΕΟ, **dns.eurecom.fr**
2. ο **dns.eurecom.fr** επικοινωνεί με τον κομβικό ΕΟ, αν κάτι τέτοιο απαιτείται
3. ο κομβικός ΕΟ επικοινωνεί με τον έγκυρο ΕΟ, **dns.umass.edu**, αν κάτι τέτοιο απαιτείται



Απλό παράδειγμα DNS (II)

κομβικός ΕΟ:

- Μπορεί να μη γνωρίζει τον έγκυρο ΕΟ
- Μπορεί να γνωρίζει τον ενδιάμεσο ΕΟ: με ποιον πρέπει να επικοινωνήσει για να βρει τον έγκυρο ΕΟ
- Αναδρομικό ερώτημα (recursive query)



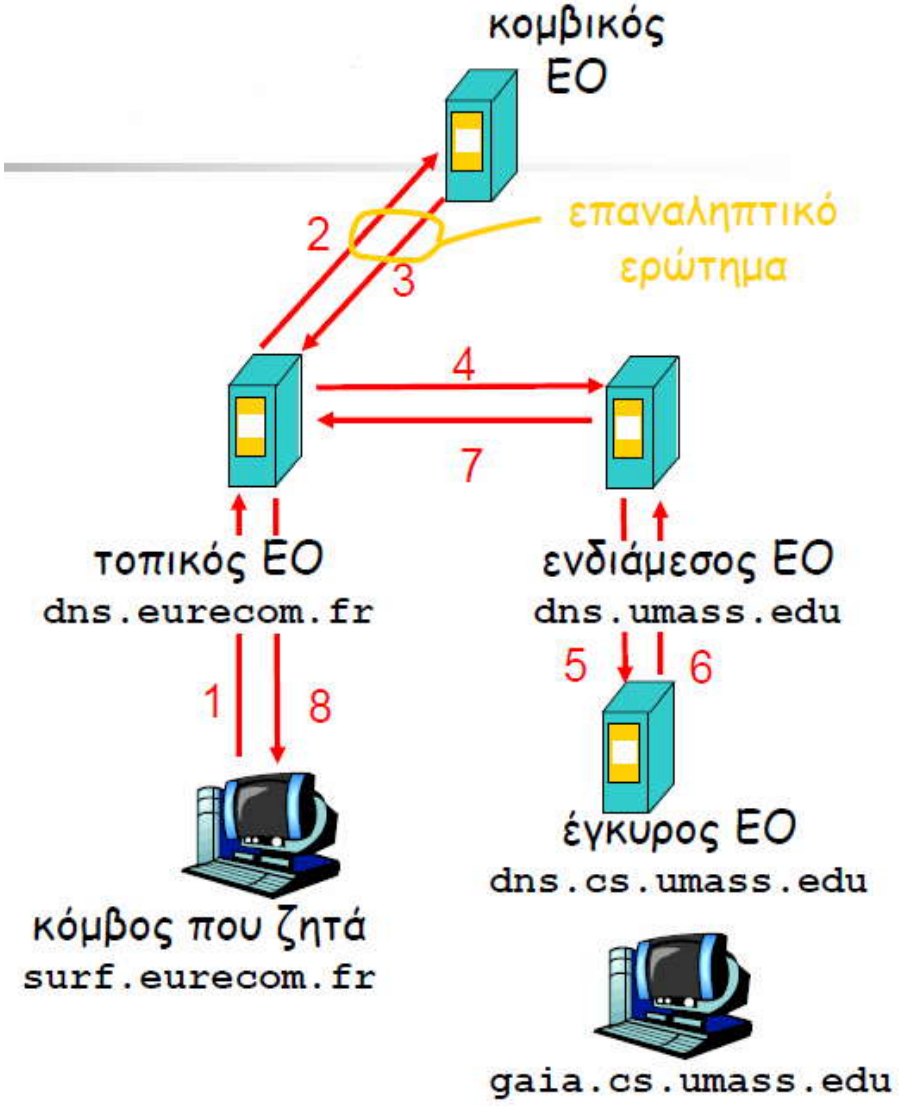
DNS: Τύποι ερωτημάτων

Αναδρομικό ερώτημα:

- Ο ΕΟ με τον οποίο επικοινωνούμε, αναλαμβάνει την αντιστοίχιση του ονόματος
- Τι γίνεται σε περιπτώσεις μεγάλου φόρτου;

Επαναληπτικό ερώτημα:

- Ο ΕΟ με τον οποίο ερχόμαστε σε επαφή μας παραπέμπει σε άλλο ΕΟ
- "δε γνωρίζω αυτό το όνομα, ρώτησε αυτόν τον ΕΟ"



Τοπική αποθήκευση στο DNS

- Όταν (οποιοσδήποτε) εξυπηρετητής ονομάτων μάθει την αντιστοίχιση, την αποθηκεύει τοπικά (caching)
  - ✓ οι εγγραφές που έχουν αποθηκευτεί τοπικά λήγουν (timeout) μετά από συγκεκριμένο χρονικό διάστημα
- Μηχανισμοί ενημέρωσης/ειδοποίησης σχεδιάζονται από την IETF
  - ✓ RFC 2136

Οι εγγραφές στο DNS

DNS: κατανεμημένη ΒΔ που αποθηκεύει εγγραφές πόρων (resource records - RR)

μορφή RR: `(name ttl class type type-specific-data)`

- name: το όνομα του κόμβου στο αρχείο ζώνης στον οποίο ανήκει αυτή η εγγραφή
- ttl: τιμή 32 bit. Η τιμή *ttl* (time to live) είναι σε δευτερόλεπτα. Τιμή μηδέν στο *ttl* σημαίνει ότι τα δεδομένα δεν πρέπει να αποθηκευθούν προσωρινά

Οι εγγραφές στο DNS

μορφή RR: `( name ttl class type type-specific-data )`

- class: μια τιμή 16 bit που καθορίζει την οικογένεια πρωτοκόλλων (η συνήθης τιμή είναι IN, που σημαίνει Internet protocol)
- type: Ο τύπος της εγγραφής πόρου που καθορίζει τις τιμές του πεδίου *type-specific-data* (δες παρακάτω για τιμές του πεδίου type)
- type-specific-data (ή data): τα περιεχόμενα κάθε εγγραφής καθορίζονται από τις τιμές *type* και *class*

Οι εγγραφές στο DNS

μορφή RR: `(name ttl class type type-specific-data )`

Παραδείγματα τιμών που μπορεί να πάρει το type:

- type=A
  - ✓ name είναι το όνομα του κόμβου
  - ✓ data είναι η διεύθυνση IP
- type=NS
  - ✓ name είναι όνομα περιοχής (π.χ. foo.com)
  - ✓ data είναι η διεύθυνση IP του έγκυρου ΕΟ για αυτή την περιοχή
- type=CNAME
  - ✓ name είναι ψευδώνυμο για κάποιο πραγματικό ("canonical") όνομα. Π.χ. το `www.ibm.com` είναι πραγματικά το `servereast.backup2.ibm.com`
  - ✓ data είναι το πραγματικό όνομα
- type=MX
  - ✓ data είναι το όνομα του mailserver που σχετίζεται με το name

Παράδειγμα εγγραφών ΕΟ ΤΕΙ Λαμίας

```
        IN      MX      10 mail.teilam.gr.
        IN      MX      20 nebula.noc.teilam.gr.

noc             IN      NS      uranus.teilam.gr.
noc             IN      NS      nebula.noc.teilam.gr.

uranus          IN      A       195.130.78.202
archon          IN      A       195.130.78.213
mail            IN      A       195.130.78.204

elabs           IN      CNAME   cipher.noc.teilam.gr.
webmail         IN      CNAME   cipher.noc.teilam.gr.

www             IN      CNAME   ccserv
ccserv          IN      A       195.130.78.68
```

Δυναμικό DNS (Dynamic DNS – DDNS)

- Απαιτείται το DNS να υποστηρίζει δυναμική ενημέρωση των εγγραφών ενός κόμβου (A και PTR)

Εξισορρόπηση Φόρτου: Ανακατεύθυνση Πελάτη

- Ο εξυπηρετητής DNS του πελάτη επιστρέφει μια λίστα διευθύνσεων IP που τηρεί ο εξυπηρετητής DNS του δικτυακού τόπου (εξισορρόπηση φόρτου – load balancing)
- Ο πελάτης (που έκανε την αίτηση στον εξυπηρετητή DNS) επιλέγει μια διεύθυνση από αυτές που βρίσκονται στη λίστα

Εξισορρόπηση Φόρτου: Ανακατεύθυνση Πελάτη

- Πλεονεκτήματα: ο πελάτης επιλέγει ανάλογα με διάφορα κριτήρια που μπορεί να θέσει ο ίδιος (μονοπάτι δρομολόγησης, χρόνο απόκρισης, επίπεδο συμφόρησης)
- Μειονεκτήματα:
  - ✓ ο πελάτης έχει τον έλεγχο της επιλογής (συνεπώς μπορεί να πάρει απόφαση μη βέλτιστη για το δίκτυο)
  - ✓ απαιτείται τροποποίηση στο πρωτόκολλο του DNS (πελάτη και εξυπηρετητή)

Ανακατεύθυνση από το DNS του Δικτυακού Τόπου: Συζήτηση

- Ο εξυπηρετητής DNS επιστρέφει κυκλικά τις διευθύνσεις IP εξυπηρετητών που τρέχουν την ίδια υπηρεσία
- Εναλλακτικά, μπορεί να επιστρέφει την IP μιας συσκευής (switch) που αναλαμβάνει την εξισορρόπηση φόρτου
- Πλεονεκτήματα:
  - ✓ προσαρμόζεται καλά στις γεωγραφικές θέσεις των πελατών
  - ✓ υλοποιείται στον παροχέα
- Μειονεκτήματα:
  - ✓ caching των αποκρίσεων DNS από τον εξυπηρετητή DNS του πελάτη (λύση: μείωση TTL, αύξηση όμως των ερωτημάτων DNS)
  - ✓ αν δε χρησιμοποιείται switch, δεν ανταποκρίνεται εύκολα σε αλλαγές στην κατάσταση του εξυπηρετητή

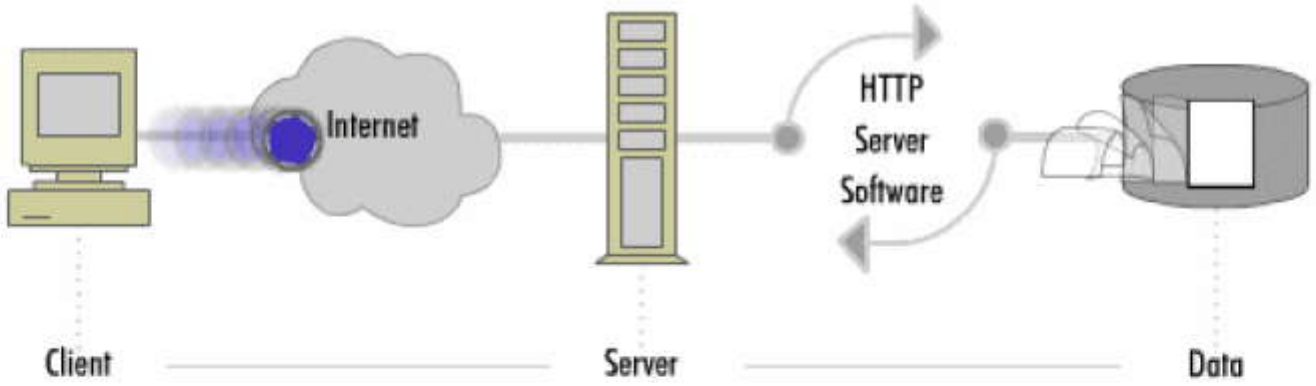
Το πρωτόκολλο HTTP (HyperText Transfer Protocol)

- Το Web περιγράφεται ως ένα καθολικό, αλληλεπιδραστικό, δυναμικό, κατανεμημένο πληροφοριακό σύστημα υπερμέσων, που τρέχει πάνω από το Internet
- Πρόκειται για την πρώτη γενιά μιας νέας τάξης κατανεμημένων συστημάτων, η οποία θα συνεχίσει να εξελίσσεται με γοργούς ρυθμούς τα επόμενα χρόνια
- Στη συνέχεια θα περιγράψουμε αναλυτικά την αρχιτεκτονική του πρωτοκόλλου HTTP

Εισαγωγή

- Το HTTP ορίζει τα αιτήματα που ένας πελάτης μπορεί να στείλει σε έναν εξυπηρετητή και τις αντίστοιχες αποκρίσεις του εξυπηρετητή
- Κάθε αίτημα περιέχει ένα URL (=συμβολοσειρά) που προσδιορίζει μια συνιστώσα (Web component) ή ένα στατικό αντικείμενο (π.χ. σελίδα HTML ή αρχείο εικόνας)
- Η προσέγγιση αίτηση/απόκριση (request/response) σημαίνει ότι ο εξυπηρετητής τρέχει συνεχώς περιμένοντας να δεχθεί αίτημα για εξυπηρέτηση από κάποιο πελάτη

Οι λειτουργίες του HTTP



- Ο εξυπηρετητής αναμένει διαρκώς για εισερχόμενες αιτήσεις στη θύρα (port) 80
- Όταν δεχθεί αιτήσεις δημιουργεί διεργασίες/νήματα για την εξυπηρέτησή τους (δηλ. για να αποστείλουν τις σελίδες που ζητούνται από τους πελάτες)

Χαρακτηριστικά του HTTP

- Το HTTP είναι ένα αμνήμων (stateless) πρωτόκολλο αίτησης/απόκρισης
- Δεν τηρείται πληροφορία στον εξυπηρετητή σχετικά με την εξελισσόμενη αλληλεπίδρασή του με τον πελάτη
- Αυτό διευκολύνει τη δουλειά ενός web server διότι μπορεί να «ξεχνάει» οτιδήποτε αφορά μια αίτηση μόλις την ικανοποιήσει
- Είναι ιδιαίτερα βολικό όταν υπάρχουν εκατομμύρια αιτήσεις. Είναι ένας από τους λόγους για τους οποίους το web κλιμακώνει σε πολύ μεγάλους όγκους πληροφορίας

Χαρακτηριστικά του HTTP

- Το γεγονός ότι το HTTP δεν τηρεί πληροφορίες κατάστασης καθιστά δυσκολότερη την ανάπτυξη εφαρμογών που εξαρτώνται από σειρά αιτήσεων και αποκρίσεων

- Για παράδειγμα, η φόρμα παραγγελίας ενός ηλεκτρονικού καταστήματος που αποτελείται από πολλές σελίδες, απαιτεί ειδική επεξεργασία στον πελάτη και τον εξυπηρετητή προκειμένου να τηρείται η κατάσταση κάθε τμήματος της φόρμας

Αρχιτεκτονική του HTTP

- Η τυπική αρχιτεκτονική του Web είναι σχεδιασμένη έτσι ώστε ένας πελάτης να μπορεί να αλληλεπιδράσει με έναν εξυπηρετητή μέσω ενός δικτύου TCP/IP
- Πελάτης και εξυπηρετητής επικοινωνούν χρησιμοποιώντας το πρωτόκολλο HTTP
- Τα υπόλοιπα επίπεδα (μεταφοράς, δικτύου, ζεύξης δεδομένων) χειρίζονται τις λεπτομέρειες μεταφοράς, όπως το να διασφαλιστεί ότι τα πακέτα δεδομένων θα φτάσουν σωστά στον προορισμό τους

Αρχιτεκτονική του HTTP

- Το HTTP προσδιορίζει τους κανόνες για τη «διατύπωση» των αιτήσεων και αποκρίσεων
- Η αίτηση `GET` στο HTTP είναι ο απλούστερος τρόπος δημιουργίας συνόδου μεταφοράς πληροφορίας
- Στην απλούστερή της εκδοχή η αίτηση `GET` απλά προσδιορίζει ένα αρχείο που θα μεταφερθεί (π.χ. `GET /jcb/home.html HTTP/1.0`)
- Δε δημιουργεί σύνδεση, διότι το TCP δημιουργεί τη δικτυακή σύνδεση για χάρη του HTTP

Λειτουργία του HTTP

- Μόλις λάβει μια αίτηση εντοπίζει τη συνιστώσα που περιγράφεται από το URL και την αντιγράφει στη σύνδεση δικτύου, μέσω της οποίας το TCP θα τη μεταφέρει στο browser
- Όταν ο πελάτης λάβει τη σελίδα για την οποία είχε κάνει αίτηση, διερμηνεύει την HTML και τον υπόλοιπο ενσωματωμένο κώδικα και απεικονίζει το αρχείο (σελίδα), όσο καλύτερα μπορεί, στον υπολογιστή του πελάτη
- Ας υποθέσουμε ότι κάποιος χρήστης πληκτρολογεί στον πελάτη που διαθέτει: `www.sun.com/httptest.html`

Δημιουργία σύνδεσης TCP

- Αρχικά θα πρέπει να δημιουργηθεί μια «σύνδεση» με το web server που «ακούει» στη διεύθυνση www.sun.com
- Αυτή είναι δουλειά του TCP. Αν δεν έχει γίνει πρόσφατα αναφορά σε αυτή τη διεύθυνση θα πρέπει να ρωτηθεί ο εξυπηρετητής ονομάτων της περιοχής του πελάτη
- Πρώτον, το HTTP απαιτεί μια αξιόπιστη σύνδεση μεταξύ του πελάτη και του εξυπηρετητή (άρα TCP)
- Δεύτερον, ο πελάτης σας στέλνει μια αίτηση στον εξυπηρετητή web της SUN που δείχνει κάπως έτσι:

Αποστολή μηνύματος αίτησης

- Πρόκειται για την αίτηση GET που λέει στον εξυπηρετητή το αρχείο που πρέπει να ανακτηθεί:

```
GET httptest.html HTTP/1.0
```

- Στη συνέχεια ενημερώνει τον εξυπηρετητή να μη στείλει το κείμενο αν δεν έχει τροποποιηθεί πρόσφατα:

```
If-modified-since: Wed, 19 Jul, 1998,
12:59:59 GMT
```

- Ο πελάτης αυτό-προσδιορίζεται ως μια έκδοση του Netscape Navigator που τρέχει σε Solaris 7:

```
User-agent: Mozilla 4.05 for Sol 7
```

Αποστολή μηνύματος αίτησης

- Αυτή η ομάδα των εντολών Accept λέει στον εξυπηρετητή τι είδους μορφές αρχείων μπορεί να χειριστεί ο πελάτης:

```
Accept: text/plain Accept: text/html
Accept: image/gif Accept: image/jpeg
```

- Υποθέτοντας ότι όλα δουλεύουν σωστά, ο HTTP server της SUN θα στείλει απόκριση στον πελάτη σας

Απόκριση εξυπηρετητή

- Αυτή είναι μια πιθανή απόκριση του εξυπηρετητή HTTP όταν όλα δουλεύουν σωστά:

```
HTTP/1.0 Status 200 Document follows
```

- Αυτο-προσδιορίζει το είδος και την έκδοσή του:

```
Server: NCSA/1.5
```

- Παρέχει την ημερομηνία και ώρα της απόκρισης:

```
Date: Tue, 20 Jul, 1998 14:00:00 GMT
```

Απόκριση εξυπηρετητή

- Ο εξυπηρετητής λέει στον πελάτη τι είδους κείμενο να περιμένει:

```
Content-type: text/html
```

- Προσδιορίζει το μέγεθος του κειμένου:

```
Content-length: 313
```

- Προσδιορίζει το είδος της κωδικοποίησης που υπάρχει για αυτό το κείμενο:

```
Content-encoding: none
```

Απόκριση εξυπηρετητή

- Η ακόλουθη πρόταση λέει στον πελάτη πότε τροποποιήθηκε για τελευταία φορά το αρχείο στον εξυπηρετητή:

```
Last-modified: Thu, 20 Jul 1998 00:00:00
GMT
```

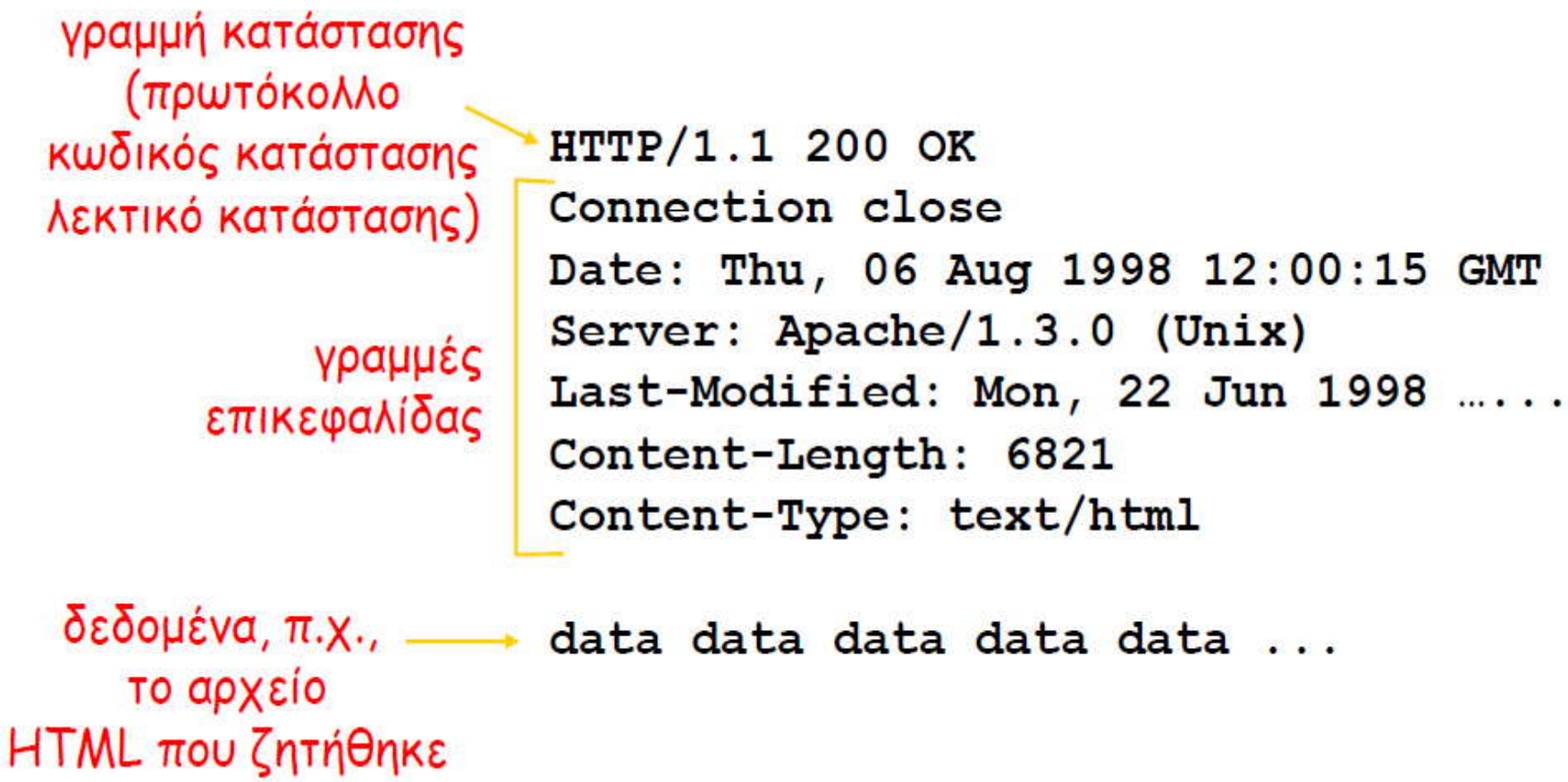
- Η κενή γραμμή που ακολουθεί τις παραπάνω δηλώσεις είναι σημαντική για το διαχωρισμό της επικεφαλίδας του HTTP (όλες τις παραπάνω γραμμές) από το ίδιο το κείμενο (όλες οι γραμμές που ακολουθούν)

Απόκριση εξυπηρετητή

- Στη συνέχεια αποστέλλεται στον πελάτη το κείμενο HTML:

```
<html><body><h2>Hello HTTP</h2>
<center><p>This page has been transmitted
to you from the Sun web site in Broomfield,
Colorado, USA.
<p>The page makes reference to one image
and not much else.  It's solely for
teaching about HTTP.
<p><img src="content:/images/sun-logo.gif"
alt="SunLogo">
</center></body></html>
```

Απόκριση εξυπηρετητή



Διερμηνεία και εμφάνιση δεδομένων σελίδας

- Μόλις η απόκριση του HTTP ληφθεί πλήρως από τον browser, η σύνδεση είτε κλείνει είτε παραμένει ανενεργή
- Τη στιγμή αυτή ο εξυπηρετητής θεωρεί ότι τελείωσε με τη σελίδα και συνεχίζει να ακούει στη θύρα 80 για αιτήσεις, «ξεχνώντας» οτιδήποτε αφορούσε την προηγούμενη σύνδεση
- Στη συνέχεια, ο browser «πιάνει δουλειά» αναλύοντας και διερμηνεύοντας την εισερχόμενη σελίδα HTML

Εργασία του πελάτη

- Καθαρίζει το παράθυρο κειμένου και ανιχνεύει την ετικέτα κειμένου <h1> προκειμένου να την εμφανίσει με τον κατάλληλο τρόπο
- Ανιχνεύει την ετικέτα παραγράφου <p> και γνωρίζει που τελειώνει η προηγούμενη παράγραφος και αρχίζει η επόμενη
- Όταν ανιχνεύσει την ετικέτα <img> αμέσως ξεκινάει το επόμενο βήμα

Μεταφορά ενσωματωμένων αρχείων

- Υποθέτοντας ότι η αυτόματη φόρτωση εικόνων είναι ενεργοποιημένη, ο πελάτης ανιχνεύει την ανάγκη για μεταφορά και του αρχείου που περιγράφεται στην ετικέτα `img`
- Ο πελάτης καλεί το TCP για να δημιουργήσει μια νέα σύνδεση (αν έχει κλείσει την αρχική, περισσότερα στη συνέχεια)
- Μόλις εγκατασταθεί η σύνδεση, ο πελάτης στέλνει μια δεύτερη αίτηση `GET` στον εξυπηρετητή HTTP της SUN (παρόμοια με την πρώτη)

Αποστολή μηνύματος αίτησης

- Η αίτηση GET ενημερώνει για το αρχείο που πρέπει να ανακτηθεί:

```
GET images/sun-logo.gif HTTP/1.0
```

- Ενημερώνει να μην σταλεί το αρχείο αν δεν έχει τροποποιηθεί πρόσφατα:

```
If-modified-since: Mon, 11 Jun, 1998, 12:59:59 GMT
```

- Ο πελάτης είναι ο Netscape Navigator που τρέχει σε Windows 95:

```
User-agent: Mozilla 4.04 for Win 95
```

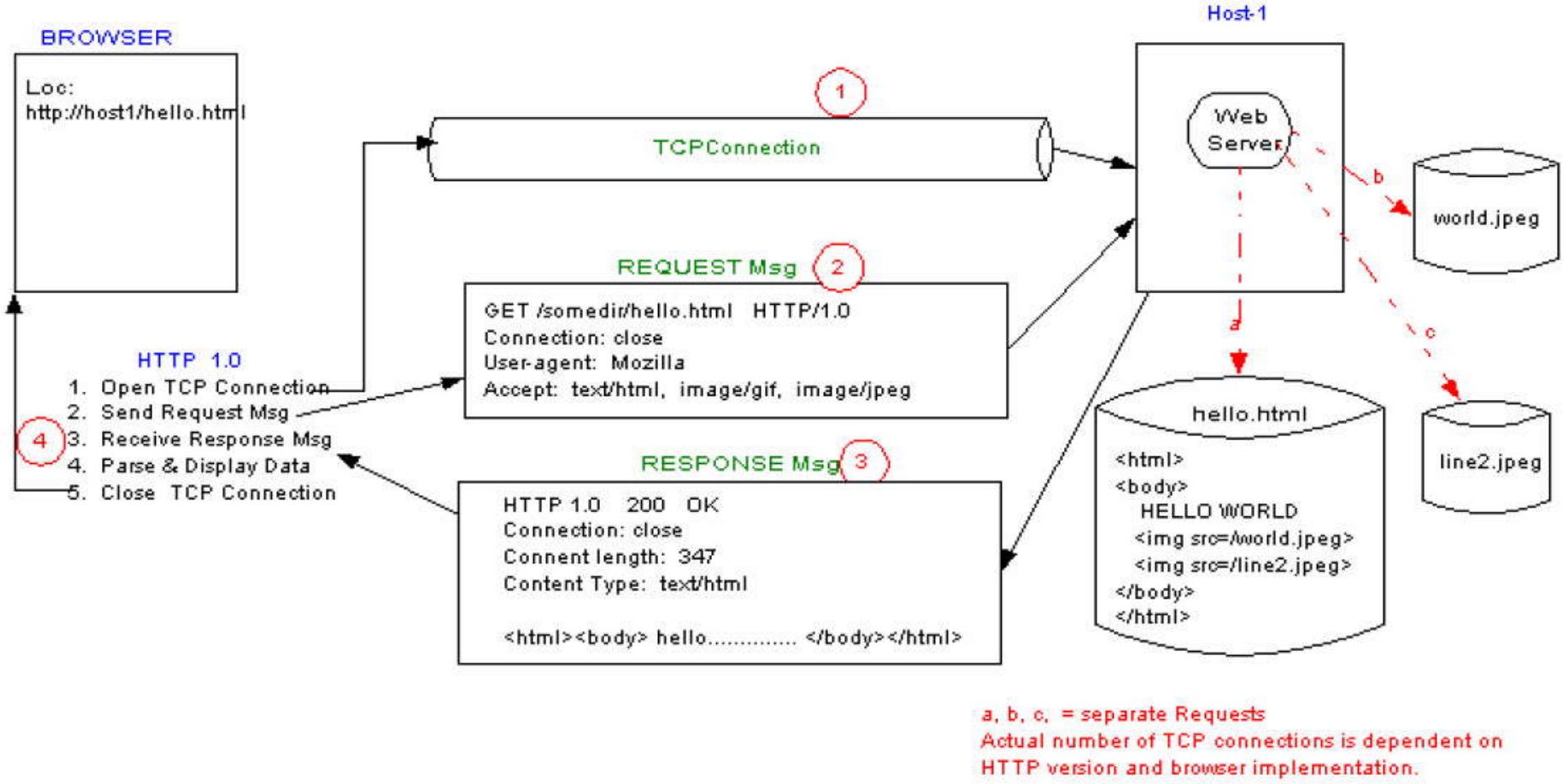
- Η ομάδα των εντολών accept λέει στον εξυπηρετητή τι μορφής αρχεία μπορεί να χειριστεί ο πελάτης:

```
Accept: text/plain Accept: text/html Accept:
image/gif Accept: image/jpeg
```

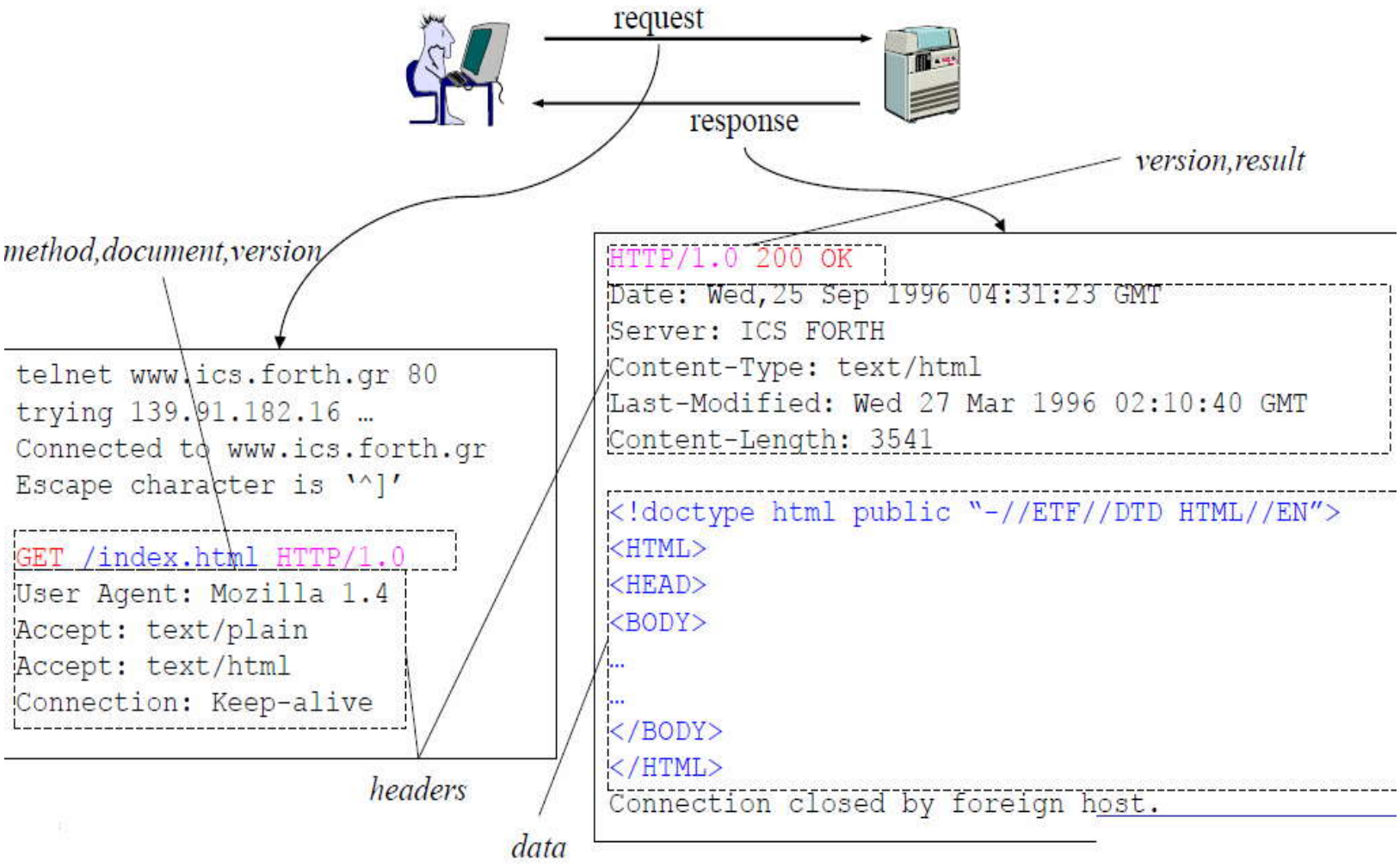
Μεταφορά της εικόνας

- Αυτή η εντολή `GET` είναι παρόμοια με την πρώτη αλλά για ένα διαφορετικό αρχείο (URL)
- Αυτή η εντολή ζητάει την εικόνα που είναι ενσωματωμένη στη σελίδα
- Υποθέτοντας ότι η εικόνα αυτή υπάρχει, ο εξυπηρετητής θα απαντήσει με τρόπο με αυτόν που απάντησε στο βήμα 3 παραπάνω
- Η μόνη διαφορά είναι ότι η επικεφαλίδα `Content-type` θα δείχνει τον κατάλληλο τύπο MIME για το αρχείο

Αρχιτεκτονική του HTTP



Αρχιτεκτονική του HTTP



MIME

- Το MIME (Multipurpose Internet Mail Extensions) είναι μια τυποποίηση κωδικοποίησης για τη μεταφορά τμημάτων πολυμέσων στο mail του Internet
- Όταν ένας πελάτης ανακτά ένα αρχείο ο εξυπηρετητής παρέχει τον τύπο MIME του αρχείου
- Ο πελάτης χρησιμοποιεί τον τύπο MIME για να διαπιστώσει αν υποστηρίζεται από τις ενσωματωμένες δυνατότητες του προγράμματος

MIME

- Τα μηνύματα MIME είναι όπως τα άλλα μηνύματα με επιπλέον πληροφορίες που αφορούν τον τύπο των δεδομένων που ακολουθούν στο σώμα του μηνύματος
- Ο τύπος ενός μηνύματος MIME περιγράφεται από ένα content type/subtype, επιτρέποντας τα αντικείμενα πολυμέσων να ενσωματώνονται σε μηνύματα άμεσα:
  - ✓ text/ascii, text/html
  - ✓ image/gif, audio/wav
  - ✓ application/doc, application/ps

Γιατί χρειάζεται το MIME;

- Επεκτασιμότητα τύπων δεδομένων: νέα ζεύγη content types/subtypes μπορούν να χρησιμοποιηθούν χωρίς να αφορούν τις υπάρχουσες εφαρμογές
- Οι browsers χειρίζονται τα αντικείμενα αναλόγως με τον τύπο τους
- Οι browsers μπορούν να χειριστούν ένα περιορισμένο τύπο δεδομένων
- Τα αντικείμενα που σχετίζονται με συγκεκριμένη εφαρμογή προωθούνται αυτόματα στα αντίστοιχα προγράμματα (registered plug-ins)

Η Εντολή POST του HTTP

- Χρησιμοποιείται εξίσου συχνά με την GET για την αποστολή δεδομένων σε εξυπηρετητή
- Οι φόρμες είναι ο πιο κοινός τύπος δεδομένων που αποστέλλονται με την POST
- Στην POST τα δεδομένα ακολουθούν την επικεφαλίδα της αίτησης
- Στην επικεφαλίδα της αίτησης υπάρχει το είδος του περιεχομένου και το μέγεθος των δεδομένων που αποστέλλονται
- Η εντολή GET απλοποιεί τη δουλειά του εξυπηρετητή, βάζει όμως όριο στο μέγεθος των δεδομένων που στέλνονται, σε αντίθεση με την POST που δε βάζει όριο

Ολοκλήρωση της μεταφοράς των στοιχείων της σελίδας…

- Τέλος ο browser εμφανίζει το μήνυμα "Document done" στη ράβδο κατάστασης και εμφανίζει τη σελίδα, μόλις ολοκληρωθούν τα ακόλουθα βήματα:
  - ✓ Η σελίδα έχει πλήρως αναλυθεί (parsed)
  - ✓ Η HTML έχει απεικονισθεί και έχουν εμφανιστεί όλες οι εικόνες
  - ✓ Εσωτερικά τμήματα άλλων τύπων δεδομένων έχουν εμφανιστεί στην οθόνη
  - ✓ Η σύνδεση TCP κλείσει

Τυπικό σενάριο HTTP

- Σύνδεση με τον τοπικό εξυπηρετητή DNS της περιοχής
- Μετατροπή του ονόματος σε διεύθυνση IP μέσω της Υπηρεσίας Ονομάτων Περιοχής
- Δημιουργία σύνδεσης TCP/IP με τον απομακρυσμένο κόμβο/εξυπηρετητή
- Κατάτμηση του μηνύματος σε πακέτα
- Δρομολόγηση των πακέτων μέσω του Διαδικτύου
- Αίτηση HTTP `GET/POST`

Τυπικό σενάριο HTTP

- Απόκριση του HTTP
- Το TCP/IP ρυθμίζει τα θέματα ελέγχου ροής και ελέγχου λαθών
- Ανάλυση της σελίδας στον πελάτη
- Επιπρόσθετες αιτήσεις HTTP `GET/POST` εφόσον αυτό είναι απαραίτητο
- Επιπρόσθετες αποκρίσεις του εξυπηρετητή HTTP στις αιτήσεις που δέχθηκε

Persistent HTTP (HTTP 1.1)

- Στην έκδοση 1.1 του HTTP, αντί να χρησιμοποιείται μια σύνδεση TCP για κάθε μεταφορά, υιοθετήθηκε η προσέγγιση της παραμένουσας ή μόνιμης σύνδεσης (persistent connection)
- Η ίδια σύνδεση παραμένει κατά τη διάρκεια πολλών αιτήσεων και αποκρίσεων (pipelining of requests)
- Πλεονέκτημα: λιγότερος φόρτος λόγω περιορισμού του πλήθους των συνδέσεων και μειονέκτημα η ανάγκη για προσδιορισμό αρχής και τέλους κάθε στοιχείου

Χρόνοι αιτήσεων και απόκρισης (Ι)

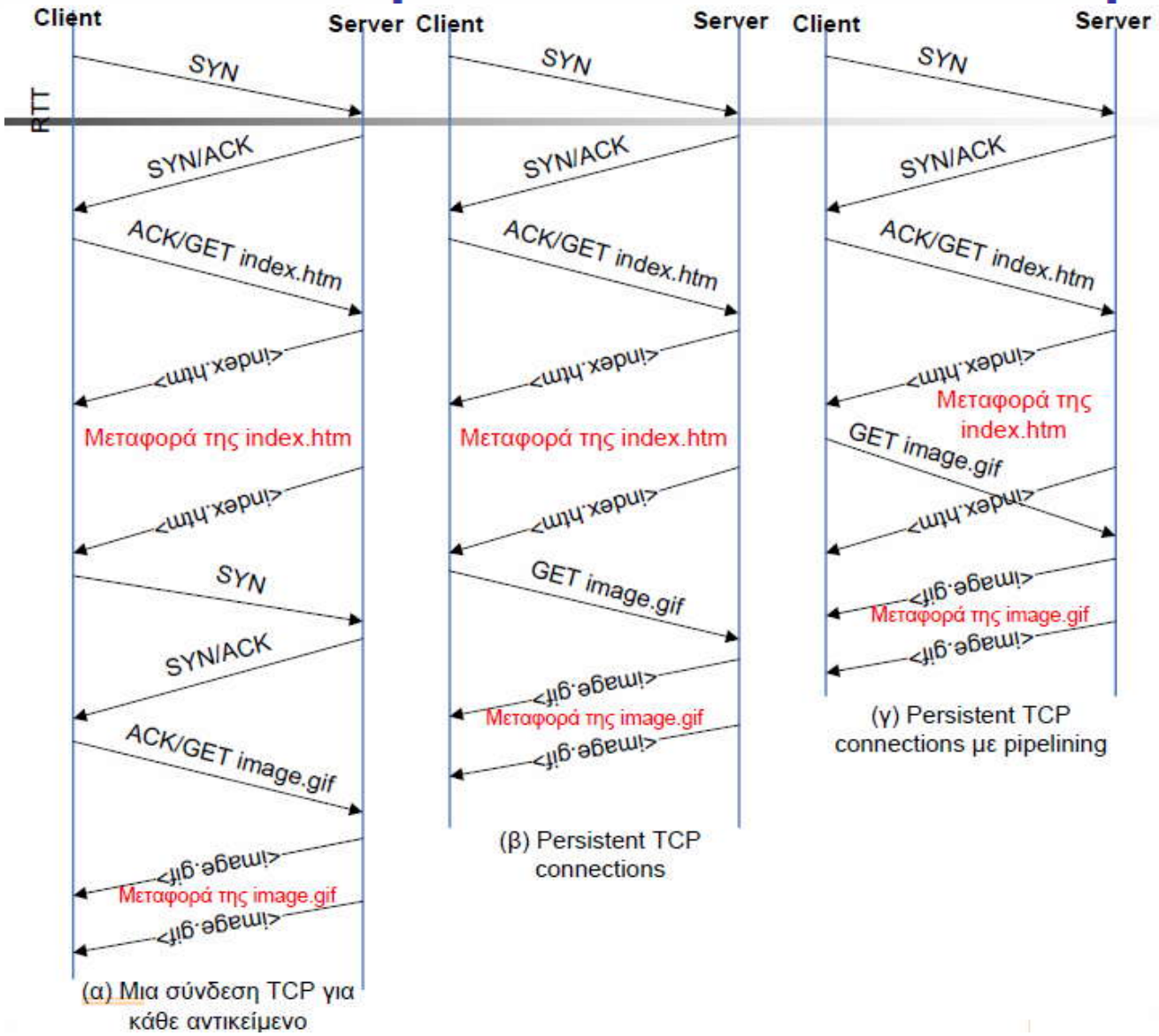
- Ορισμός του Round Trip Time (RTT): χρόνος για να ταξιδέψει ένα μικρό πακέτο από τον πελάτη στον εξυπηρετητή και πίσω

Χρόνος απόκρισης:

- ένα RTT για την αρχικοποίηση της σύνδεσης TCP
- ένα RTT για την αίτηση HTTP και να επιστρέψουν τα πρώτα bytes της απόκρισης HTTP
- χρόνος μετάδοσης αρχείου

Σύνολο = 2*RTT+χρόνος μετάδοσης αρχείου

Χρόνοι αιτήσεων και απόκρισης (II)



Persistent HTTP

Nonpersistent HTTP:

- απαιτεί 2 RTTs/αντικείμενο
- το ΛΣ πρέπει να αναθέσει πόρους του host για κάθε σύνδεση TCP
- οι browsers συνήθως ανοίγουν παράλληλες συνδέσεις TCP για να αποκτήσουν τα αντικείμενα ☹

Persistent HTTP

- ο εξυπηρετητής αφήνει τη σύνδεση ανοικτή μετά την απάντηση
- τα επακόλουθα μηνύματα HTTP μεταξύ του ίδιου client/server στέλνονται πάνω από τη σύνδεση

Persistent χωρίς pipelining:

- ο πελάτης δημιουργεί νέες αιτήσεις μόνο όταν έχει ληφθεί η προηγούμενη απόκριση
- ένα RTT για κάθε αντικείμενο

Persistent με pipelining:

- default στο HTTP/1.1
- ο πελάτης στέλνει αιτήσεις μόλις «συναντήσει» ένα αντικείμενο
- σχεδόν ένα RTT για όλα τα αναφερόμενα αντικείμενα (+ φόρτος στο server να διατηρήσει τη σειρά)

Αιτήσεις στο HTTP

- Μια αίτηση HTTP αποτελείται από μια μέθοδο αίτησης, ένα URL, πεδία επικεφαλίδας και ένα σώμα αίτησης. Το HTTP 1.1 ορίζει τις ακόλουθες μεθόδους αίτησης:
  - ✓ `GET` – Ανάκτηση του πόρου που προσδιορίζεται από το URL
  - ✓ `HEAD` – Επιστρέφει τις επικεφαλίδες που προσδιορίζονται από το URL (όμοια με τη GET αλλά χωρίς τα δεδομένα)
  - ✓ `POST` – Αποστολή Δεδομένων απεριόριστου μήκους στο web server
  - ✓ `PUT` – Αποθήκευση πόρου κάτω από το URL

Αιτήσεις στο HTTP

- ✓ DELETE – Απομάκρυνση του πόρου που προσδιορίζεται από το URL
- ✓ OPTIONS – Επιστρέφει τις μεθόδους HTTP που υποστηρίζει ο εξυπηρετητής
- ✓ TRACE – Επιστροφή των πεδίων επικεφαλίδας που αποστέλλονται με την αίτηση TRACE

- Η έκδοση 1.0 του HTTP περιείχε μόνο τις μεθόδους GET, POST και HEAD

Αποκρίσεις στο HTTP

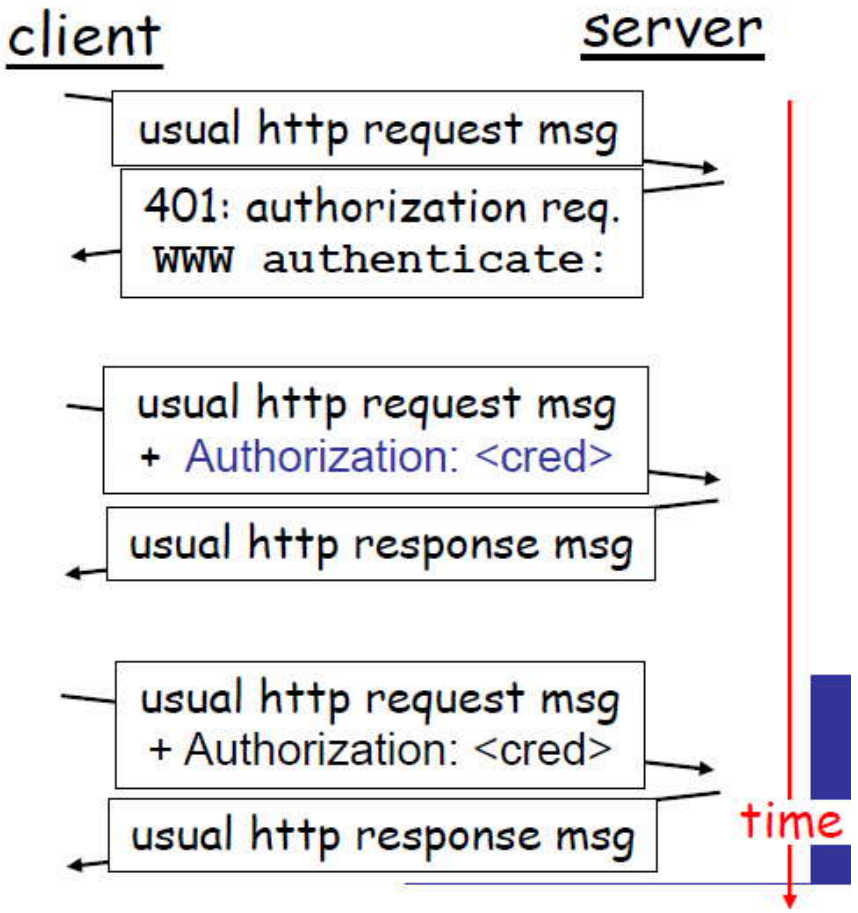
- Μια απόκριση στο HTTP περιέχει ένα κωδικό αποτελέσματος, πεδία επικεφαλίδας και ένα σώμα
- Το πρωτόκολλο HTTP περιμένει τον κωδικό αποτελέσματος και όλα τα πεδία επικεφαλίδας να επιστραφούν πριν από οποιοδήποτε σώμα
- Η γραμμή κατάστασης καταγράφει την έκδοση του πρωτοκόλλου, έναν αριθμητικό κωδικό κατάστασης και μια φράση κειμένου για την κατάσταση
- Ο κωδικός κατάστασης έχει τρία δεκαδικά ψηφία

Αποκρίσεις στο HTTP

- Ορισμένοι συνηθισμένοι κωδικοί κατάστασης είναι:
  - ✓ 404 – ο ζητούμενος πόρος δεν είναι διαθέσιμος
  - ✓ 401 – απαιτείται έλεγχος πρόσβασης
  - ✓ 301 – ο πόρος έχει ένα νέο URL που θα πρέπει να χρησιμοποιηθεί. Το νέο URL επιστρέφεται στο πεδίο Location της απόκρισης
  - ✓ 304 – ο ζητούμενος πόρος δεν έχει τροποποιηθεί μετά από την ημερομηνία που προσδιορίζει στην επικεφαλίδα του ο πελάτης
  - ✓ 500 – εσωτερικό λάθος στον εξυπηρετητή HTTP
  - ✓ 503 – εξυπηρετητής προσωρινά ανίκανος να ικανοποιήσει αίτηση λόγω υπερφόρτωσης

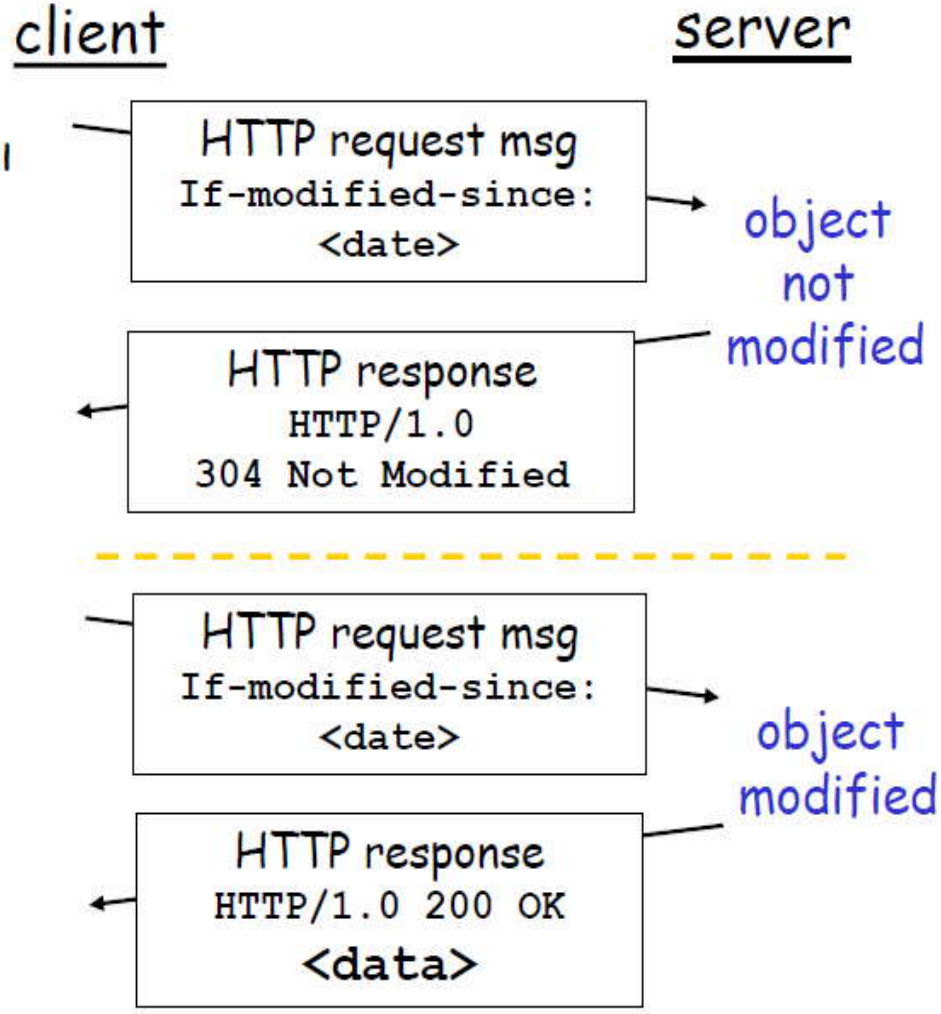
Έλεγχος πρόσβασης στο HTTP

- πιστοποίηση: έλεγχος πρόσβασης στο περιεχόμενο του εξυπηρετητή
- πιστοποιητικά: συνήθως όνομα χρήστη, κωδικός πρόσβασης
- αμνήμων: ο πελάτης πρέπει να παρέχει τα πιστοποιητικά σε κάθε αίτηση
  - ✓ επικεφαλίδα `Authorization:` σε κάθε αίτηση
  - ✓ αν δεν υπάρχει επικεφαλίδα `Authorization:` ο εξυπηρετητής αρνείται την πρόσβαση στέλνοντας την επικεφαλίδα `WWW authenticate:` στην απόκριση



GET υπό συνθήκη: τοπική αποθήκευση στον πελάτη

- στόχος: η μη αποστολή του αντικειμένου αν ο πελάτης διαθέτει ενημερωμένο αντίγραφο
- πελάτης: προσδιορίζει την ημερομηνία του αποθηκευμένου αντίγραφου στην αίτηση HTTP
  - ✓ `If-modified-since: <date>`
- εξυπηρετητής: η απόκριση δεν περιέχει αντικείμενο αν το αποθηκευμένο αντίγραφο δεν έχει τροποποιηθεί:
  - ✓ `HTTP/1.0 304 Not Modified`



Υποστήριξη ενδιαμέσων (Proxies)

- Ρυθμίζονται όλοι οι πελάτες ενός δικτύου να στέλνουν τις αιτήσεις τους στον ενδιάμεσο εξυπηρετητή (proxy server)
- Την πρώτη φορά που κάποιος χρήστης του δικτύου προσπελάσει μια σελίδα, αυτή:
  - ✓ ζητείται, εκ μέρους του χρήστη, από τον εξυπηρετητή που την κατέχει μέσω του ενδιάμεσου εξυπηρετητή,
  - ✓ αποθηκεύεται τοπικά στον ενδιάμεσο εξυπηρετητή (cache), και
  - ✓ στη συνέχεια μεταφέρεται στον πελάτη

Υποστήριξη ενδιαμέσων (ΙΙ)

- Στην επόμενη προσπέλαση στην ίδια σελίδα, η σελίδα αυτή θα μεταφερθεί από την προσωρινή μνήμη του ενδιάμεσου εξυπηρετητή στον πελάτη (εφόσον δεν έχει αλλάξει), αυξάνοντας έτσι την απόδοση, μειώνοντας το χρόνο αναμονής και την κίνηση στο Διαδίκτυο
- Σε πολλές περιπτώσεις δημιουργείται ιεραρχία από ενδιάμεσους οι οποίοι «διαδίδουν» προς τις μεγαλύτερες βαθμίδες της ιεραρχίας τις αιτήσεις των χρηστών περιμένοντας κάποιος από την ιεραρχία να κατέχει τη σελίδα

HTTP Cookies

- Πρόκειται για αξιοποίηση της επεκτασιμότητας του HTTP για να επιτρέψει σε προγράμματα που τρέχουν στο server (π.χ. προγράμματα PhP) να αποθηκεύουν και να ανακτούν πληροφορία στον πελάτη
- Αυτό επιτρέπει την τήρηση πληροφοριών κατάστασης σχετικών με την εφαρμογή στον πελάτη, έτσι ώστε οι προσπελάσεις να φαίνονται ότι σχετίζονται με τη σύνοδο μιας εφαρμογής
- Ορίζουν επιπλέον πληροφορίες επικεφαλίδας μέσω των οποίων οι servers προσαρτούν πληροφορίες κατάστασης

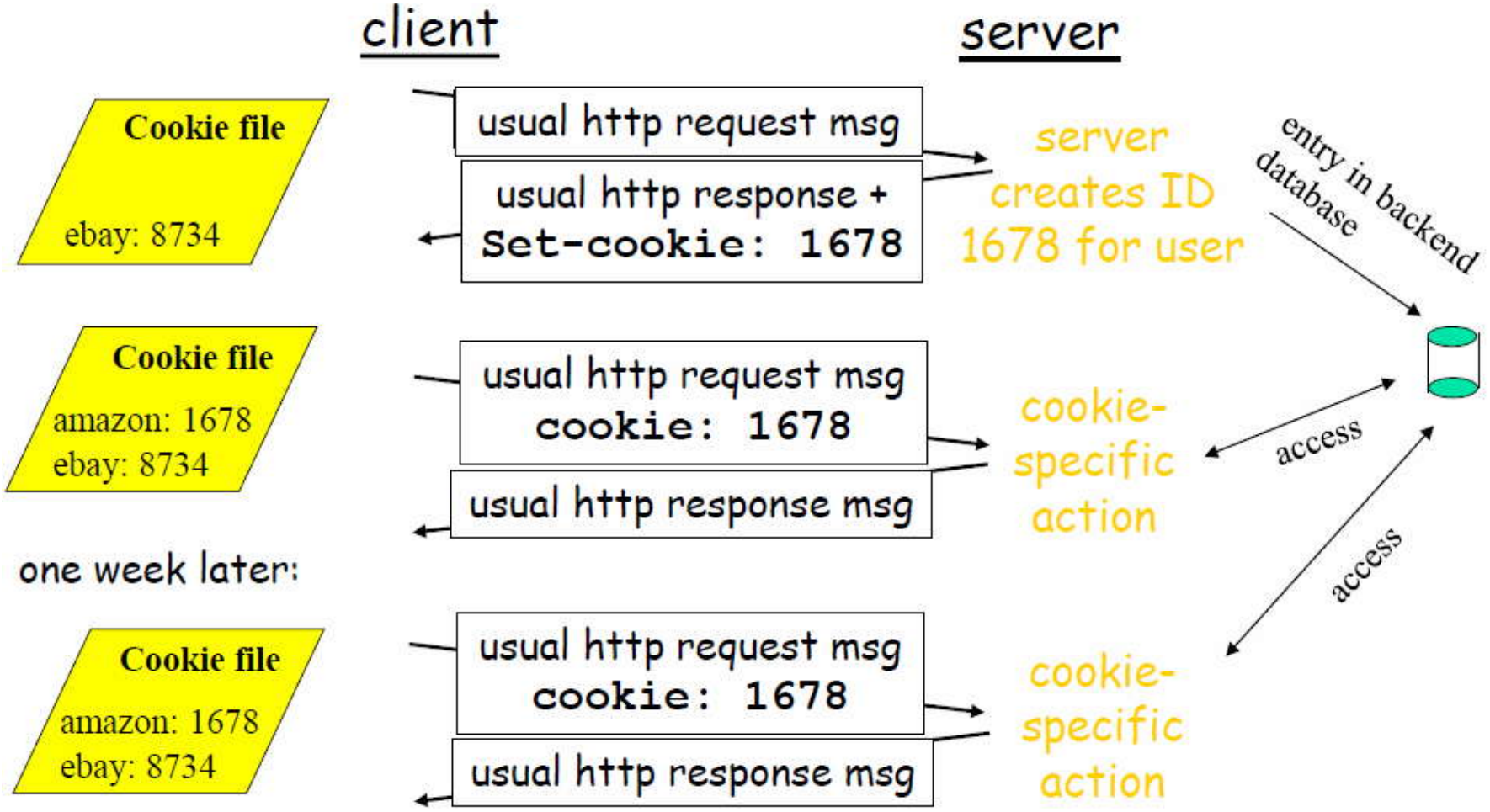
HTTP Cookies

- Η πληροφορία των cookies σχετίζεται με συγκεκριμένα URL (για τα οποία είναι έγκυρη)
- Όταν ζητηθεί αυτό το URL ο πελάτης περιλαμβάνει την πληροφορία του cookie στην αίτησή του
- Τα cookies έχουν και αυτά ημερομηνία λήξεως, μετά την οποία δεν αποστέλλονται με την αίτηση του πελάτη
- Η ασφαλής αποστολή τους γίνεται όπως και στις περιπτώσεις των «κανονικών» δεδομένων (π.χ. με χρήση του HTTPS και του SSL)

HTTP Cookies

- Δημιουργείται επικεφαλίδα `cookie`: στο μήνυμα αίτησης του HTTP
- Το ίδιο και στην επικεφαλίδα απόκρισης του HTTP
- Το αρχείο του cookie (συνήθως κάποιο id σχετικό με το δικτυακό τόπο που το έστειλε και/ή πληροφορίες για το χρήστη) φυλάσσεται στο χρήστη και το χειρίζεται ο browser
- Στον εξυπηρετητή χρησιμοποιείται σε back-end βάση δεδομένων

Παράδειγμα χρήσης Cookies



HTTPS – Ασφαλές HTTP

- Ο πελάτης αναγνωρίζει το `https` ως μια αίτηση χρήσης ασφαλών υποδοχών (sockets) για το πέρασμα των δεδομένων
- Το SSL (Secure Sockets Layer) επιτρέπει την αποστολή κρυπτογραφημένων δεδομένων μεταξύ πελάτη-εξυπηρετητή
- Πλεονέκτημα αποτελεί το γεγονός ότι το HTTP παραμένει το ίδιο και αλλάζει το επίπεδο μεταφοράς

Ο Πελάτης της υπηρεσίας HTTP

- Υπάρχουν διάφορες ονομασίες για αυτού του είδους τις προσθήκες (helper applications, plug-ins, Xtras, link libraries, cartridges, components, objects, classes, κ.λπ.)
- Η έννοια πίσω από όλα αυτά είναι απλή: όπου υπάρχει μια τυποποίηση, τρίτοι κατασκευαστές μπορούν να παρέχουν «εξαρτήματα» που προσαρμόζονται εύκολα στην υλοποίηση της τυποποίησης του κατασκευαστή

Plug-ins

- Πρόκειται για τμήματα προγραμμάτων που επεκτείνουν τις δυνατότητες του browser
- Το plug-in API επιτρέπει σε άλλους κατασκευαστές να επεκτείνουν το browser με έμφυτη υποστήριξη για νέους τύπους δεδομένων
- Πρόκειται για δυναμικά τμήματα κώδικα, που συμπληρώνουν αρχιτεκτονικές όπως το OLE και γλώσσες προγραμματισμού ανεξάρτητες από πλατφόρμα, όπως είναι η Java

Το Plug-in API

- Παρέχει ομοιόμορφη υποστήριξη για νέους τύπους δεδομένων στους χρήστες ενός browser
- Παρέχει τη μέγιστη δυνατή ευελιξία για τη συγγραφή plug-ins
- Είναι λειτουργικά ισοδύναμο σε όλες τις πλατφόρμες
- Μπορεί να αλληλεπιδράσει με κάποιο παράθυρο που αποτελεί μέρος της ιεραρχίας του browser
- Μπορεί να ανακτήσει δεδομένα από το δίκτυο μέσω URL και να δημιουργήσει ή να καταναλώσει δεδομένα

Βοηθητικές εφαρμογές εναντίον Plug-ins (I)

- Οι βοηθητικές εφαρμογές (helper applications) τρέχουν ως ξεχωριστά προγράμματα, ενώ τα plug-ins λειτουργούν μέσα από τον εξυπηρετητή
- Πρόκειται βασικά για μια εφαρμογή η οποία που καταλαβαίνει και μεταφράζει αρχεία που δεν μπορεί να χειριστεί ο πελάτης
- Τα plug-ins λειτουργούν ουσιαστικά ως επεκτάσεις στον εκάστοτε πελάτη και χειρίζονται συγκεκριμένους τύπους δεδομένων MIME και επεκτάσεις αρχείων

Βοηθητικές εφαρμογές εναντίον Plug-ins (II)

- Με τις βοηθητικές εφαρμογές μπορεί να γίνει multitasking μεταξύ μιας τέτοιας εφαρμογής και του παράθυρου του πελάτη
- Τα plug-ins είναι δυναμικά κομμάτια κώδικα και εξαρτώνται από την πλατφόρμα (platform specific)
- Η ολοκλήρωση plug-ins είναι μια διαδικασία διαφανής στους χρήστες, ενώ δεν απαιτείται διαμόρφωση των plug-ins

Αρχιτεκτονική του browser

- Σε γενικές γραμμές ένας browser έχει περισσότερο πολύπλοκη αρχιτεκτονική από έναν εξυπηρετητή
- Ο browser χειρίζεται τις λεπτομέρειες πρόσβασης και απεικόνισης των σελίδων
- Αποτελείται από πολλά συνεργαζόμενα κομμάτια λογισμικού: πελάτες, διερμηνευτές και έναν ελεγκτή που τα διαχειρίζεται
- Ο πελάτης τηρεί συσχετισμούς θέσεων στην οθόνη και ενεργών τμημάτων στο κείμενο HTML

Προαιρετικοί πελάτες

- Ένας browser μπορεί να περιέχει κομμάτια λογισμικού για την υλοποίηση επιπλέον λειτουργιών (FTP, E-mail)
- Χωρίς ο χρήστης σαφώς να καλέσει το λογισμικό αυτό, ο browser καλεί την υπηρεσία αυτόματα, κρύβοντας τις λεπτομέρειες από το χρήστη
- Το τμήμα του URL που αναφέρεται στο πρωτόκολλο χρησιμοποιείται για την επιλογή του πελάτη που θα αναλάβει την ανάκτηση του αρχείου

Τοπική αποθήκευση (Caching) στους browsers

- Το Web έχει διαφορετικό μοντέλο κίνησης πληροφορίας από τις υπόλοιπες εφαρμογές
- Οι browsers χρησιμοποιούν cache για την αύξηση της απόδοσης πρόσβασης σε σελίδες του web
- Αντίγραφο κάθε σελίδας που ανακτάται τοποθετείται σε μια cache στον τοπικό δίσκο και ο browser απευθύνεται στον κάτοχο της σελίδας μόνο αν δεν την βρει στο δίσκο
- Υπέρ: η ταχύτατη ανάκτηση. Κατά :η απώλεια χώρου στο δίσκο, η μεταβλητότητα στις επισκέψεις σε σελίδες

Υποστήριξη του HTTP για τοπική αποθήκευση

- Ο εξυπηρετητής μπορεί να ορίσει χρόνο ζωής της σελίδας στην cache, ενώ ο πελάτης μπορεί να ζητήσει για μια αίτησή του να μη χρησιμοποιηθεί η cache
- Η λειτουργία `HEAD` του HTTP βοηθάει στη διαχείριση της cache, διότι μέσω αυτής μπορούμε να ελέγξουμε την εγκυρότητα της έκδοσης της σελίδας που έχει η cache
- Ο πελάτης στέλνει μια αίτηση `HEAD` για τη σελίδα που επιθυμεί και αν η απάντηση δείξει ότι το αντίγραφο της cache είναι έγκυρο χρησιμοποιείται

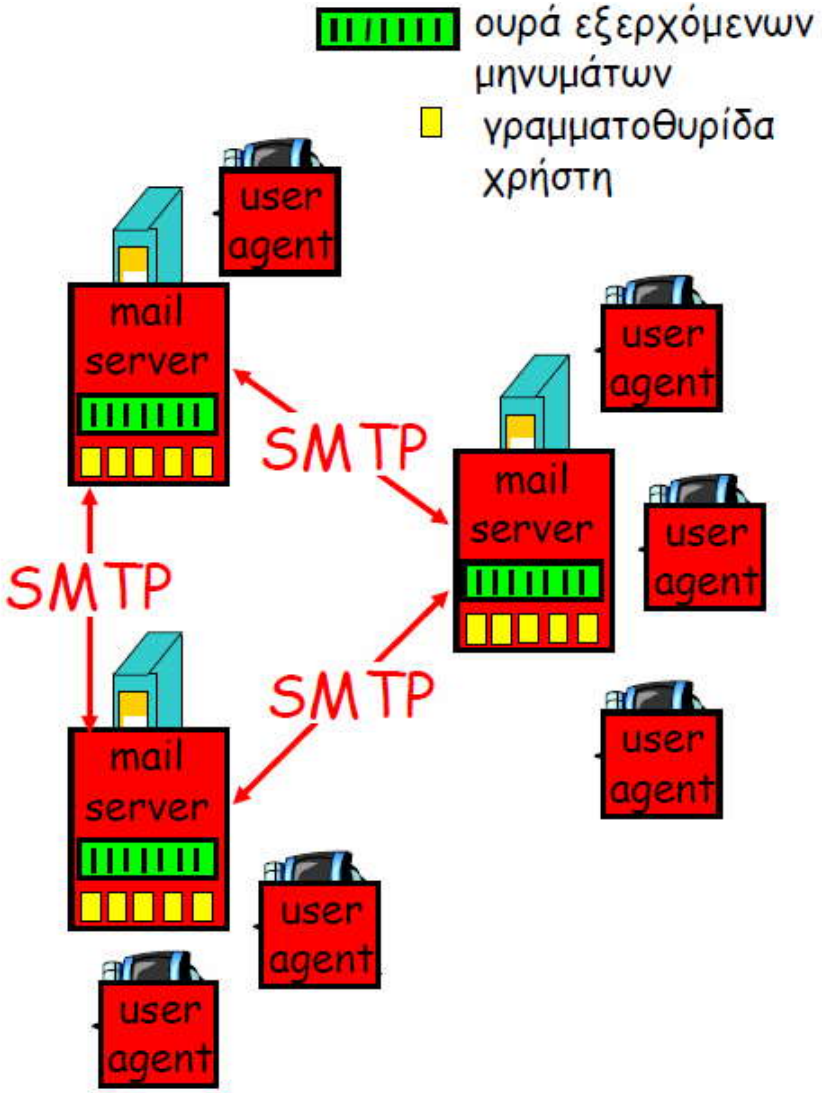
Ηλεκτρονικό Ταχυδρομείο

Τρία δομικά στοιχεία:

- Πράκτορες χρήστη (πελάτες)
- Εξυπηρετητές μηνυμάτων
- simple mail transfer protocol: SMTP

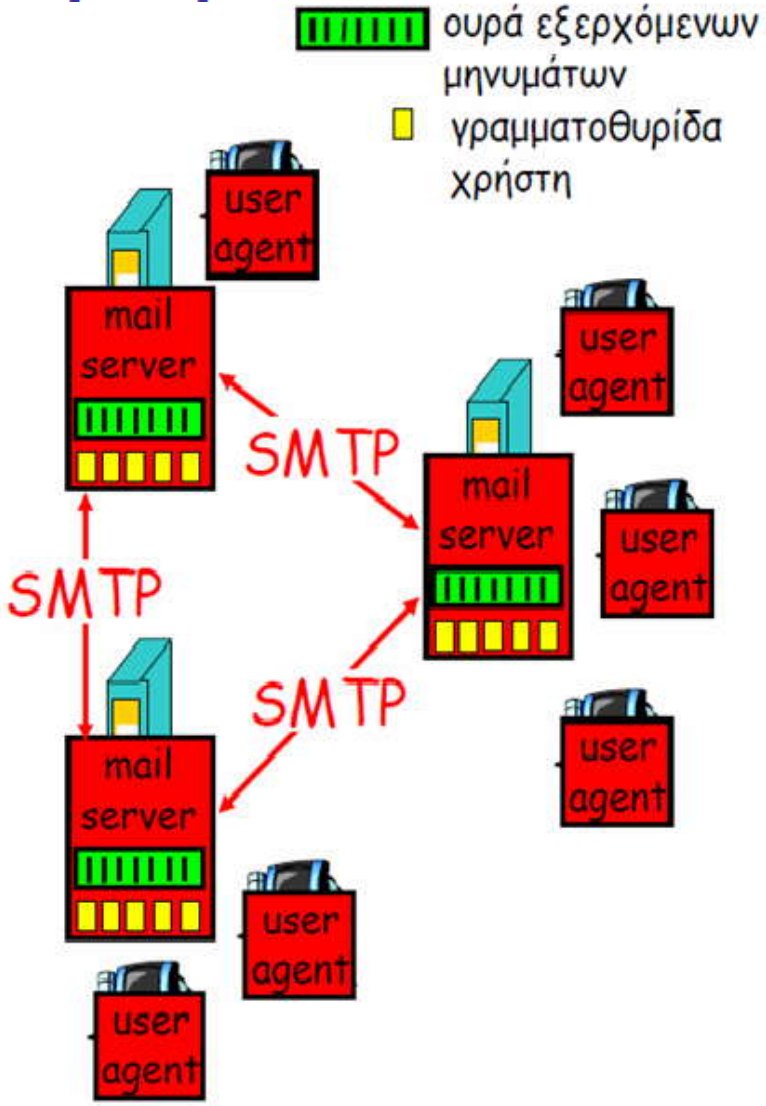
Πράκτορας χρήστη (user agent - UA)

- Σύνθεση, επεξεργασία και ανάγνωση μηνυμάτων ηλεκτρονικού ταχυδρομείου
- π.χ., pine, Outlook, elm, thunderbird
- Εξερχόμενα και εισερχόμενα μηνύματα αποθηκεύονται στον εξυπηρετητή



Ηλεκτρονικό Ταχυδρομείο

- Εξυπηρετητές μηνυμάτων
- Η γραμματοθυρίδα (mailbox) περιέχει εισερχόμενα μηνύματα για κάποιο χρήστη
- Ουρά εξερχόμενων μηνυμάτων για τα μηνύματα που θα αποσταλούν
- Μεταξύ εξυπηρετητών για την αποστολή e-mail χρησιμοποιείται το SMTP
  - ✓ πελάτης: ο αποστολέας εξυπηρετητής
  - ✓ «εξυπηρετητής»: ο παραλήπτης



Electronic Mail: SMTP [RFC 2821]

- Χρησιμοποιεί TCP για την αξιόπιστη μεταφορά μηνυμάτων (e-mails) από τον πελάτη στον εξυπηρετητή, στη θύρα 25
- Ο αποστολέας στέλνει άμεσα στον παραλήπτη εξυπηρετητή
- Τρεις φάσεις μεταφοράς
  - ✓ χειραψία (χαιρετισμός)
  - ✓ μεταφορά μηνυμάτων
  - ✓ τερματισμός
- Αλληλεπίδραση εντολής/απόκρισης
  - ✓ εντολές: κείμενο ASCII
  - ✓ απόκριση: κωδικός κατάστασης και φράση
- Τα μηνύματα πρέπει να είναι σε ASCII 7-bit

Σενάριο

1) Η Μαρία χρησιμοποιεί UA για να συνθέσει μήνυμα «προς» yannis@somewhere.gr

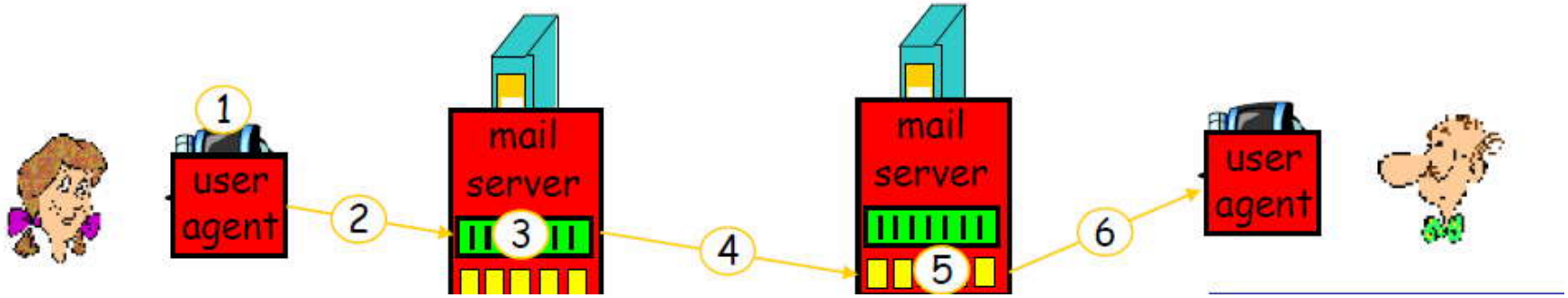
2) Ο UA της Μαρίας στέλνει το μήνυμα στο mail server της· το μήνυμα μπαίνει στην ουρά εξερχόμενων μηνυμάτων

3) Ο πελάτης SMTP ανοίγει σύνδεση TCP με το mail server του Γιάννη

4) Ο πελάτης SMTP στέλνει το μήνυμα της Μαρίας πάνω από τη σύνδεση TCP

5) Ο mail server του Γιάννη τοποθετεί το μήνυμα στη γραμματοθυρίδα του

6) Ο Γιάννης διαβάζει το μήνυμα με το δικό του UA



SMTP

- Το SMTP χρησιμοποιεί παραμένουσες συνδέσεις
- Το SMTP απαιτεί το μήνυμα (επικεφαλίδα & σώμα) να είναι 7-bit ASCII
- Ο εξυπηρετητής SMTP με το CRLF.CRLF προσδιορίζει το τέλος του μηνύματος

Σύγκριση με το HTTP:

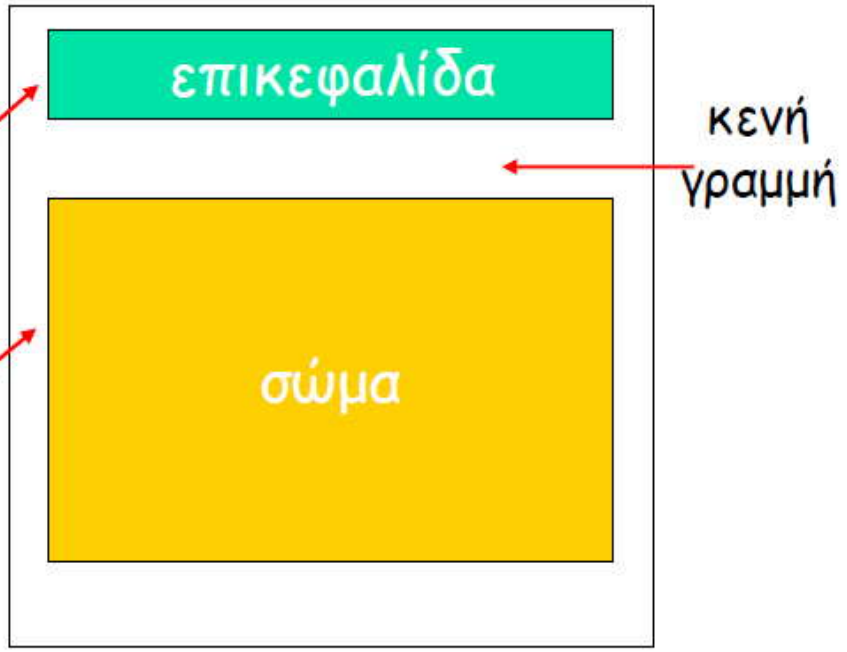
- HTTP: pull, SMTP: push
- Και τα δύο έχουν ASCII κωδικούς κατάστασης, εντολές και αποκρίσεις
- HTTP: κάθε αντικείμενο ενσωματώνεται στο δικό του μήνυμα απόκρισης
- SMTP: πολλά αντικείμενα στέλνονται σε ένα μήνυμα με πολλά μέρη

Μορφή μηνύματος ηλεκτρονικού ταχυδρομείου

SMTP: πρωτόκολλο για την ανταλλαγή μηνυμάτων ηλεκτρονικού ταχυδρομείου

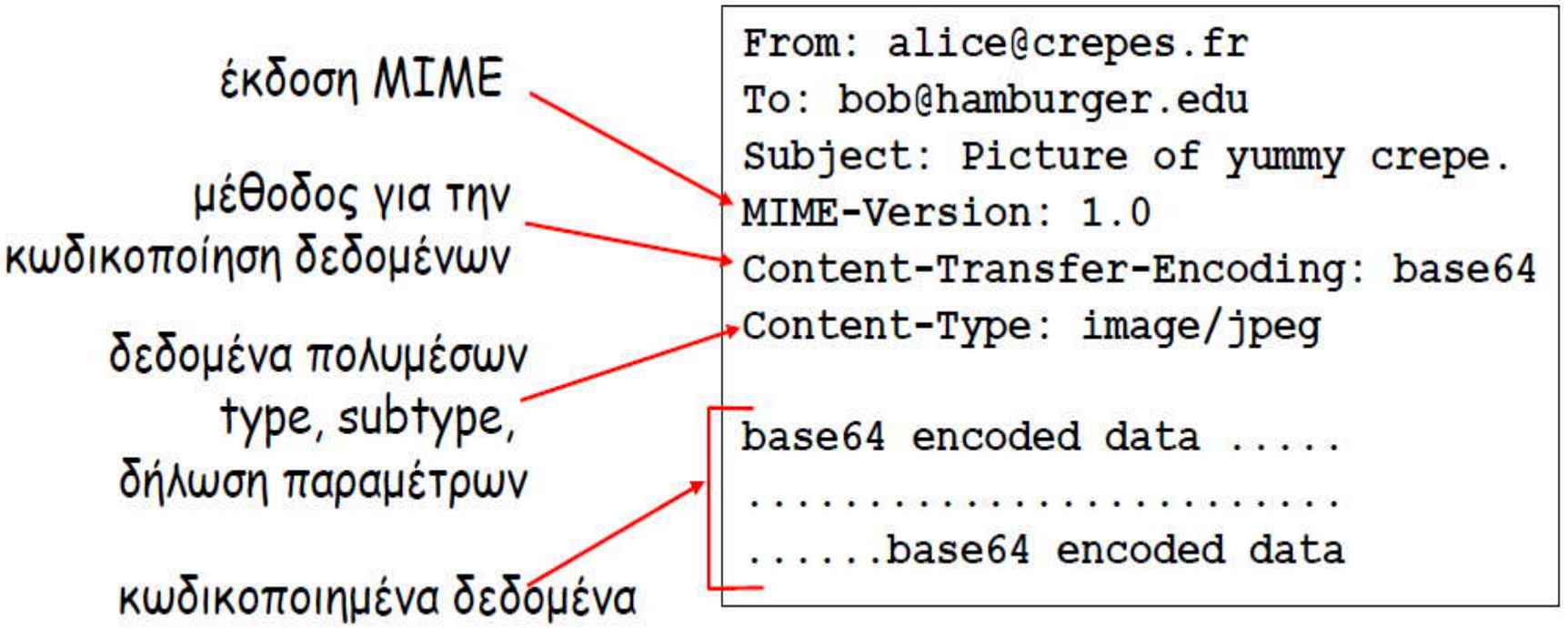
RFC 822: πρότυπο για τη μορφή μηνύματος κειμένου:

- Γραμμές επικεφαλίδας, π.χ.,
  - ✓ To:
  - ✓ From:
  - ✓ Subject:
  - ✓ διαφορετικές από τις εντολές SMTP!
- Σώμα
  - ✓ το "μήνυμα", μόνο χαρακτήρες ASCII



Μορφή μηνύματος: επεκτάσεις πολυμέσων

- MIME: multimedia mail extension, RFC 2045, 2056
- Επιπλέον γραμμές στην επικεφαλίδα του μηνύματος δηλώνουν τον τύπο περιεχομένου MIME



Τύποι MIME

`Content-Type: type/subtype; parameters`

Κείμενο (text)

- παραδείγματα από subtypes: **`plain, html`**

Εικόνα (image)

- παραδείγματα από subtypes: **`jpeg, gif`**

Ήχος (audio)

- παραδείγματα από subtypes: **`basic`** (κωδικοποίηση 8-bit mu-law), **`32kadpcm`** (κωδικοποίηση32 kbps)

Video

- παραδείγματα από subtypes: **`mpeg, quicktime`**

Εφαρμογή (application)

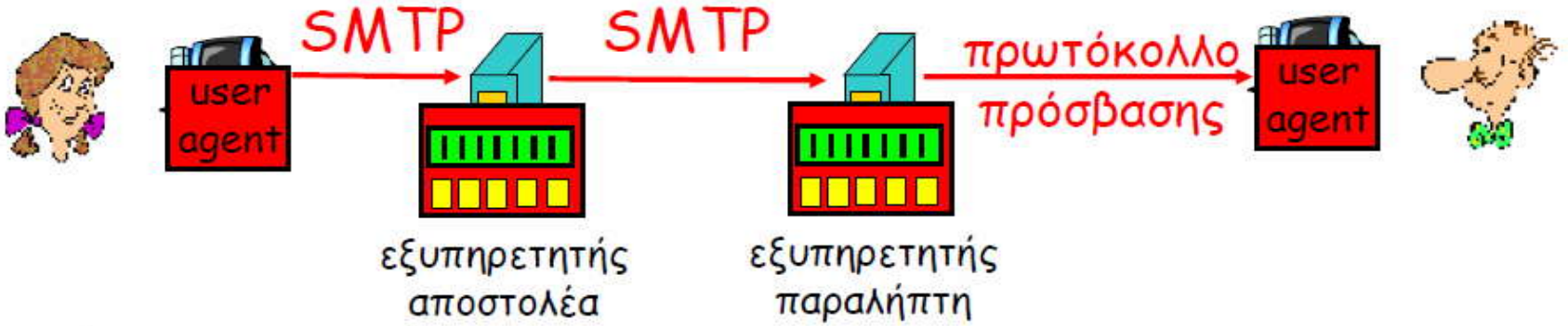
- άλλα δεδομένα που πρέπει να επεξεργαστεί ο αναγνώστης προκειμένου να είναι «ορατά»
- παραδείγματα: **`msword, octet-stream`**

Μήνυμα με πολλά μέρη (multipart)

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=StartOfNextPart

--StartOfNextPart
Dear Bob, Please find a picture of a crepe.
--StartOfNextPart
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data .....
.........................
......base64 encoded data
--StartOfNextPart
Do you want the recipe?
```

Πρωτόκολλα πρόσβασης σε μηνύματα



- SMTP: παράδοση/αποθήκευση στον εξυπηρετητή του παραλήπτη
- Πρωτόκολλο πρόσβασης στα μηνύματα: ανάκτηση από τον εξυπηρετητή
  - ✓ POP: Post Office Protocol [RFC 1939]
    - έγκριση (πελάτης <--> εξυπηρετητής) και κατέβασμα
  - ✓ IMAP: Internet Mail Access Protocol [RFC 1730]
    - περισσότερα χαρακτηριστικά (πιο πολύπλοκο)
    - χειρισμός μηνυμάτων που είναι αποθηκευμένα σε εξυπηρετητή
  - ✓ HTTP: Hotmail , Yahoo! Mail, κ.λπ.

Πρωτόκολλο POP3

Φάση έγκρισης:

- Εντολές πελάτη:
  - ✓ **user:** δήλωση username
  - ✓ **pass:** password
- Αποκρίσεις εξυπηρετητή:
  - ✓ **+OK**
  - ✓ **-ERR**

Φάση δοσοληψίας, πελάτης:

- **list:** κατάλογος αριθμών μηνυμάτων
- **retr:** ανάκτηση μηνύματος με αριθμό
- **dele:** διαγραφή
- **quit**

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

POP3 και IMAP

Μερικά ακόμα για το POP3

- Το προηγούμενο παράδειγμα χρησιμοποιεί τον τρόπο «κατέβασμα και διαγραφή»
- Ο Γιάννης δεν μπορεί να ξαναδιαβάσει το μήνυμα αν αλλάξει πελάτη
- "Κατέβασμα και κράτημα": αντίγραφα μηνυμάτων σε διαφορετικούς πελάτες
- Το POP3 είναι αμνήμων

IMAP

- Κρατάει όλα τα μηνύματα σε ένα μέρος: τον εξυπηρετητή
- Επιτρέπει στο χρήστη την οργάνωση των μηνυμάτων σε φακέλους
- IMAP τηρεί πληροφορίες κατάστασης μεταξύ συνόδων (με μνήμη):
  - ✓ ονόματα φακέλων και αντιστοιχίσεις μεταξύ μηνυμάτων και ονόματος φακέλου