



ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

Στρώμα εφαρμογής



- Βασικές αρχές και θέματα υλοποίησης για τις εφαρμογές δικτύου
 - Αρχιτεκτονικές εφαρμογών
 - Απαιτήσεις εφαρμογών για την υπηρεσία μεταφοράς
- Μερικές δημοφιλείς εφαρμογές και πρωτόκολλα του στρώματος εφαρμογής
 - Web και HTTP
 - P2P διανομή αρχείων

Περιεχόμενα

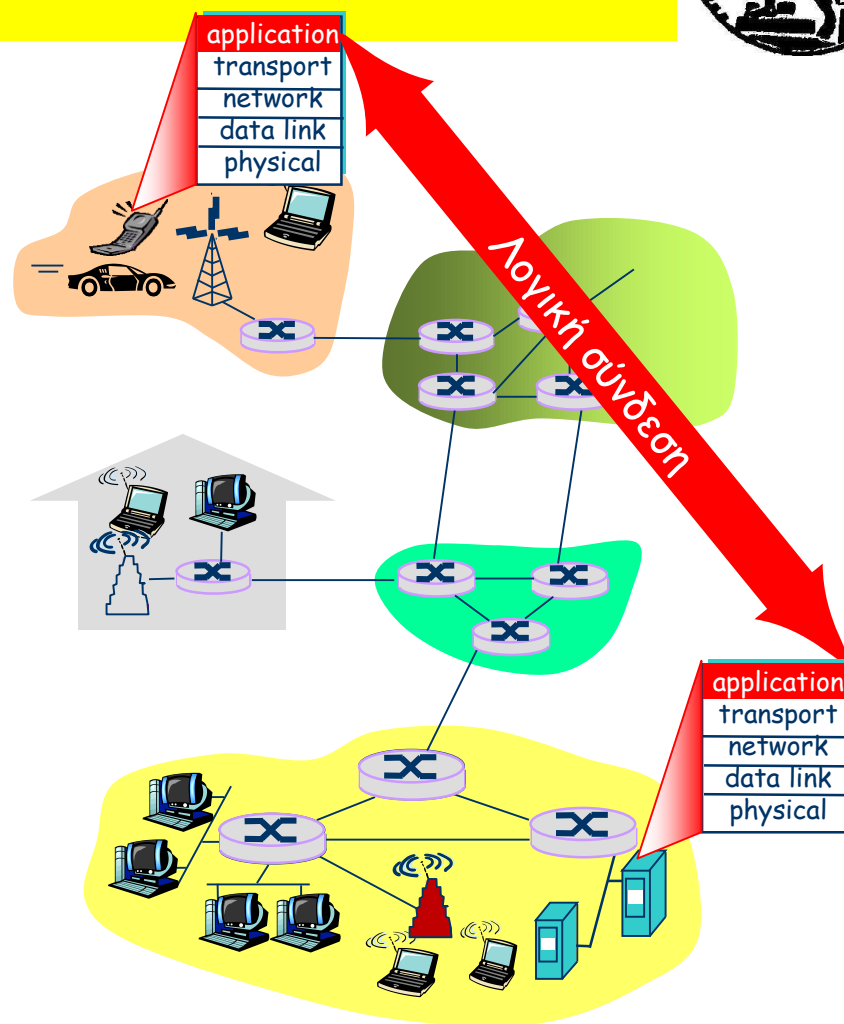


- Γενικά για τις εφαρμογές δικτύου
- Αρχιτεκτονικές εφαρμογών δικτύου
 - client-server (πελάτη-εξυπηρετητή)
 - peer-to-peer (ομοτίμων)
- Επικοινωνία διαδικασιών
 - Διεπαφές στρώματος εφαρμογής
 - Υπηρεσίες δικτύου που απαιτούνται για τις εφαρμογές
- Web and HTTP
- Εφαρμογές P2P
 - Διανομή αρχείων
 - Internet telephony

Στρώμα εφαρμογής



- Το στρώμα εφαρμογής παρέχει υπηρεσίες στον χρήστη.
- Η επικοινωνία παρέχεται μέσω μιας λογικής σύνδεσης, δηλαδή τα δύο στρώματα εφαρμογής θεωρούν ότι υπάρχει μια νοητή άμεση σύνδεση μεταξύ τους μέσω της οποίας μπορούν να στείλουν και να λάβουν μηνύματα.



Εφαρμογές δικτύου



Μερικές εφαρμογές δικτύου

- e-mail
- web
- instant messaging
- remote login
- P2P file sharing
- multi-user network games
- streaming stored video clips (π.χ. YouTube)
- voice over IP (π.χ. Skype)
- real-time video conferencing
- distributed computing
- ...
- ...

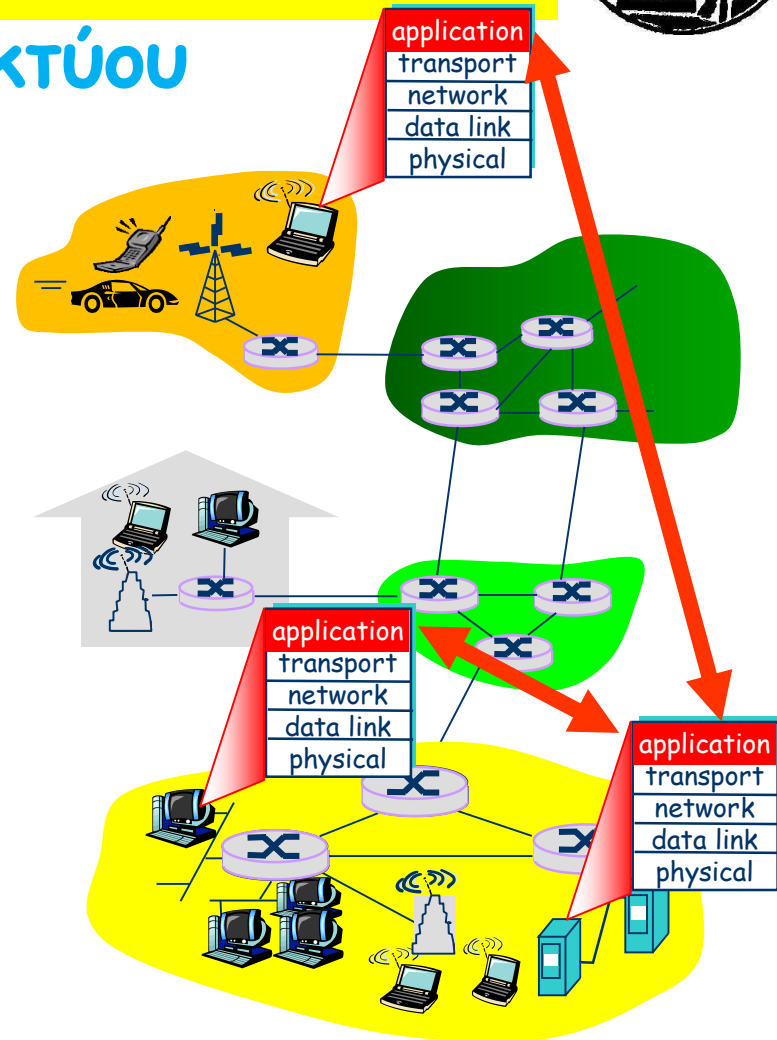
Εφαρμογές δικτύου



Δημιουργία εφαρμογής δικτύου

Γράφονται προγράμματα που:

- Τρέχουν σε (διαφορετικά) τερματικά συστήματα,
- Επικοινωνούν μεταξύ τους πάνω από το δίκτυο
 - π.χ., το software του web server επικοινωνεί με το software του browser.
- Οι συσκευές του δικτύου κορμού δεν τρέχουν εφαρμογές χρήστη.
- Οι εφαρμογές στα τερματικά συστήματα επιτρέπουν την ταχεία ανάπτυξη των εφαρμογών και τη διάδοσή τους.



Αρχιτεκτονικές εφαρμογών δικτύου



- Η αρχιτεκτονική μιας εφαρμογής είναι διαφορετική από την αρχιτεκτονική δικτύου.
- Για τον δημιουργό μιας εφαρμογής, η αρχιτεκτονική δικτύου είναι σταθερή και παρέχει συγκεκριμένες υπηρεσίες στις εφαρμογές.
- Η αρχιτεκτονική εφαρμογής σχεδιάζεται από τον δημιουργό της εφαρμογής και δείχνει πώς είναι δομημένη η εφαρμογή πάνω στα διάφορα τερματικά.
- Κατά την επιλογή αρχιτεκτονικής εφαρμογής, αυτός που σχεδιάζει την εφαρμογή θα επιλέξει, πιθανότατα, μια από τις δύο επικρατούσες αρχιτεκτονικές: **client-server** και **peer-to-peer (P2P)**.

Αρχιτεκτονικές εφαρμογών δικτύου



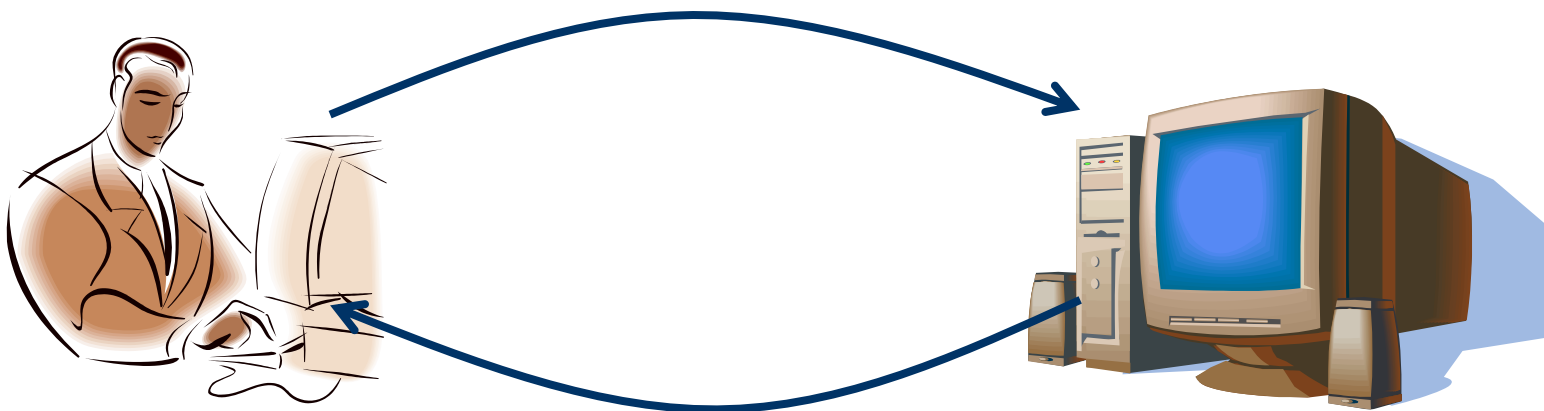
- **Client-server**: υπάρχει πάντα ένας ενεργοποιημένος host (server) που εξυπηρετεί αιτήσεις από πολλούς άλλους host (clients).
- **Peer-to-peer (P2P)**: αξιοποιεί την άμεση επικοινωνία μεταξύ ζευγών περιστασιακά συνδεδεμένων host, που ονομάζονται **ομότιμοι** (peers).
- **Υβριδική**: συνδυάζει στοιχεία και client-server και P2P.

Αρχιτεκτονικές εφαρμογών δικτύου



Αρχιτεκτονική Client-server

- **Πρόγραμμα Client**
 - Τρέχει σε host
 - Ζητάει υπηρεσία
 - π.χ., Web browser
 - **Πρόγραμμα Server**
 - Τρέχει σε host
 - Παρέχει υπηρεσία
 - π.χ., Web server
- `GET /index.html`



"Site under construction"

Αρχιτεκτονικές εφαρμογών δικτύου



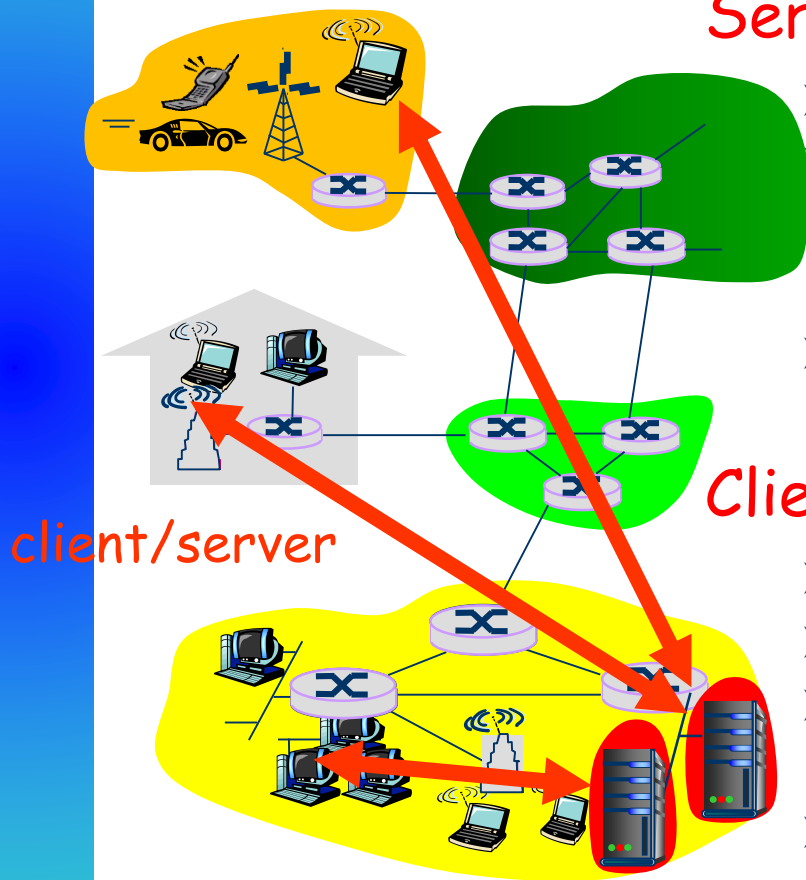
Αρχιτεκτονική Client-server

Server:

- Πάντα ενεργοποιημένος host
- Μόνιμη διεύθυνση IP.
- Δέχεται αιτήσεις από πολλούς client hosts.
- Ομάδα από servers (server farm) για κλιμάκωση.

Clients:

- Επικοινωνούν με τον server.
- Μπορεί να συνδέονται περιστασιακά.
- Μπορεί να έχουν δυναμικές διευθύνσεις IP.
- Δεν επικοινωνούν απευθείας με άλλους clients.
- Πρέπει να ξέρουν τη διεύθυνση του server.

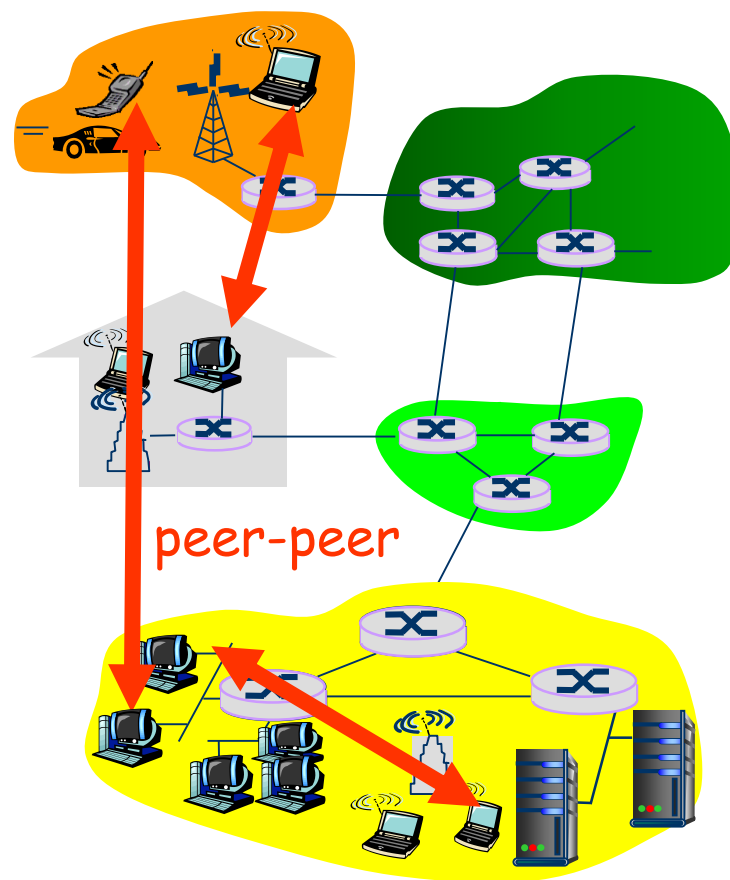


Αρχιτεκτονικές εφαρμογών δικτύου



Αμιγής αρχιτεκτονική P2P

- Όχι πάντα ενεργοποιημένος server.
- Τυχαία τερματικά συστήματα επικοινωνούν απευθείας.
- Ομότιμοι ζητούν υπηρεσία από άλλους ομότιμους, παρέχουν υπηρεσία σε άλλους ομότιμους.
 - Αυτοκλιμάκωση - νέοι ομότιμοι φέρουν νέα χωρητικότητα, καθώς και νέα αιτήματα εξυπηρέτησης.
- Οι ομότιμοι συνδέονται περιστασιακά και αλλάζουν διευθύνσεις IP.
 - Περίπλοκη διαχείριση.



Αρχιτεκτονικές εφαρμογών δικτύου



Υβριδική Client-server και P2P

Skype

- Εφαρμογή voice-over-IP P2P.
- Κεντρικός server: βρίσκει τη διεύθυνση του απόμακρου μέρους .
- Σύνδεση client-client: άμεση (όχι μέσω server).

Instant messaging

- Το chatting μεταξύ δύο χρηστών είναι P2P.
- Κεντρική υπηρεσία: ανίχνευση παρουσίας client/ εντοπισμός.
- Ο χρήστης εγγράφει την IP διεύθυνσή του όταν είναι online.
- Ο χρήστης επικοινωνεί με τον κεντρικό server για να βρει διευθύνσεις IP φίλων.

Επικοινωνία διαδικασιών



Διαδικασία: πρόγραμμα που τρέχει σε κάποιον host.

- Στον ίδιο host, δύο διαδικασίες επικοινωνούν χρησιμοποιώντας **inter-process επικοινωνία** (καθοριζόμενη από το OS).
- Διαδικασίες σε διαφορετικούς host επικοινωνούν με **ανταλλαγή μηνυμάτων**.

Διαδικασία client:

διαδικασία που αρχίζει την επικοινωνία.

Διαδικασία server:

διαδικασία που αναμένει την επαφή.

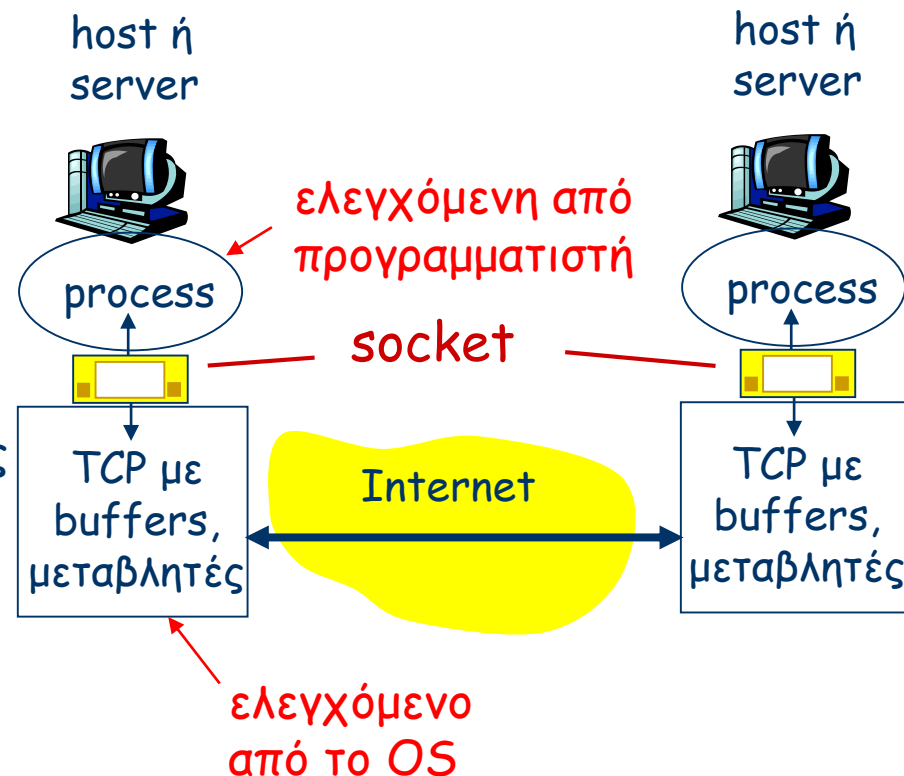
- **Σημείωση:** εφαρμογές με αρχιτεκτονικές P2P έχουν και διαδικασίες client και διαδικασίες server.

Επικοινωνία διαδικασιών



- Μια διαδικασία στέλνει / λαμβάνει μηνύματα προς/από το δίκτυο μέσω μιας software διεπαφής, που καλείται **υποδοχή (socket)** ή **API (Application Programmer's Interface)**

- Η αποστέλλουσα διαδικασία:
 - Ωθεί το μήνυμα προς την υποδοχή της.
 - Βασίζεται στην υποδομή μεταφοράς που βρίσκεται στην άλλη πλευρά της υποδοχής και μεταφέρει το μήνυμα στην υποδοχή της διαδικασίας λήψης.



- API: (1) επιλογή του πρωτοκόλλου μεταφοράς, (2) δυνατότητα ρύθμισης λίγων παραμέτρων (π.χ. max buffer size και MSS)

Επικοινωνία διαδικασιών



Διευθυνσιοδότηση διαδικασιών

- Για να λάβει μηνύματα, η διαδικασία πρέπει να έχει *ταυτότητα*.
- Ο host έχει μοναδική διεύθυνση IP 32-bit.
- Δεν αρκεί η διεύθυνση IP του host στον οποίο τρέχει η διαδικασία για να προσδιορίσει τη διαδικασία.
- Η *ταυτότητα* μιας διαδικασίας περιλαμβάνει και τη *διεύθυνση IP* και τον *αριθμό θύρας* που σχετίζεται με τη διαδικασία στον host.
- Παραδείγματα αριθμών θυρών:
 - HTTP server: 80
 - Mail server: 25
- Για να σταλεί μήνυμα HTTP στον web server edu-dy.cn.ntua.gr:
 - διεύθυνση IP: 147.102.40.9
 - αριθμός θύρας: 80

Ποια υπηρεσία μεταφοράς χρειάζεται για μια εφαρμογή;



Απώλειες δεδομένων

- Μερικές εφαρμογές (π.χ., audio) μπορεί να ανέχονται μερικές απώλειες.
- Άλλες εφαρμογές (π.χ., file transfer, telnet) απαιτούν 100% αξιόπιστη μεταφορά δεδομένων.

Καθυστέρηση

- Μερικές εφαρμογές (π.χ., Internet telephony, interactive games) απαιτούν μικρή καθυστέρηση για να είναι "αποτελεσματικές".

Εύρος ζώνης

- Μερικές εφαρμογές (π.χ., multimedia) απαιτούν κάποιο ελάχιστο εύρος ζώνης για να είναι "αποτελεσματικές".
- Άλλες εφαρμογές ("ελαστικές") χρησιμοποιούν οποιοδήποτε εύρος ζώνης είναι διαθέσιμο.

Απαιτήσεις γνωστών εφαρμογών για την υπηρεσία μεταφοράς



Οι απαιτήσεις των διαφόρων εφαρμογών, όσο αφορά τις απώλειες δεδομένων, τις καθυστερήσεις και το εύρος ζώνης ποικίλουν.

Εφαρμογή	Απώλειες	Εύρος ζώνης	Ευαισθησία στην καθυστέρηση
file transfer	όχι	ευέλικτο	όχι
e-mail	όχι	ευέλικτο	όχι
Web documents	όχι	ευέλικτο	όχι
real-time audio/ Video	ανεκτές	audio: 5kbps-1Mbps video:10kbps-5Mbps	ναι, εκατοντάδες ms
stored audio/video	ανεκτές	όπως ανωτέρω	ναι, λίγα sec
interactive games	ανεκτές	λίγα kbps - 10kbps	ναι, εκατοντ. ms
instant messaging	όχι	ευέλικτο	ναι και όχι

Υπηρεσίες μεταφοράς στο Internet



Υπηρεσία TCP:

- *Με σύνδεση:* απαιτείται εγκατάσταση σύνδεσης μεταξύ των διαδικασιών client και server.
- *Αξιόπιστη μεταφορά* μεταξύ διαδικασίας εκπομπής και διαδικασίας λήψης.
- *Έλεγχος ροής:* ο πομπός δεν πλημμυρίζει τον δέκτη.
- *Έλεγχος συμφόρησης:* εμποδίζει τον πομπό όταν το δίκτυο είναι υπερφορτωμένο.
- *Δεν παρέχει:* χρονική εγγύηση και εξασφάλιση ελάχιστου εύρους ζώνης.

Υπηρεσία UDP:

- Αναξιόπιστη μεταφορά δεδομένων μεταξύ διαδικασίας εκπομπής και διαδικασίας λήψης.
- Δεν παρέχει: εγκατάσταση σύνδεσης, αξιοπιστία, έλεγχο ροής, έλεγχο συμφόρησης, χρονική εγγύηση ή εξασφάλιση εύρους ζώνης.

Εφαρμογές στο Internet



Πρωτόκολλα εφαρμογής και μεταφοράς

Εφαρμογή	Πρωτόκολλο στρ. εφαρμογής	Πρωτόκολλο στρ. μεταφοράς
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (π.χ., YouTube), RTP	TCP ή UDP
Internet telephony	SIP, RTP ή propri- etary (π.χ., Skype)	τυπικά UDP



Πρωτόκολλα στρώματος εφαρμογής

Το πρωτόκολλο στρώματος εφαρμογής είναι ένα μέρος της εφαρμογής.

➤ Web

- Πρότυπο για τη μορφή των αρχείων: HTML
- Web browsers (π.χ. Mozilla)
- Web servers (π.χ. Apache)
- Πρωτόκολλο στρώματος εφαρμογής: HTTP

➤ E-mail

- Mail servers που έχουν mail boxes
- Mail readers
- Πρότυπο που ορίζει τη δομή του μηνύματος e-mail
- Πρωτόκολλο στρώματος εφαρμογής: SMTP

Εφαρμογές στο Internet



Πρωτόκολλα στρώματος εφαρμογής

- Ένα πρωτόκολλο στρώματος εφαρμογής ορίζει:
 - Τύπους ανταλλασσόμενων μηνυμάτων
 - π.χ., request, response
 - Συντακτικό μηνυμάτων
 - ποια πεδία στο μήνυμα και πώς τα πεδία περιγράφονται
 - Σημασιολογία των μηνυμάτων
 - σημασία της πληροφορίας στα διάφορα πεδία
 - Κανόνες για το πότε και πώς οι διαδικασίες στέλνουν και απαντούν σε μηνύματα.
- Public-domain protocols: ορίζονται στα RFC
 - επιτρέπουν διαλειτουργία
 - π.χ., HTTP, SMTP
- Proprietary protocols:
 - π.χ., Skype

Πρωτόκολλα στρώματος εφαρμογής



Web και HTTP

Βασικοί ορισμοί

- Μια ιστοσελίδα αποτελείται από ένα **βασικό αρχείο HTML** που περιέχει αρκετά **αναφερόμενα αντικείμενα**.
- Αντικείμενο μπορεί να είναι ένα αρχείο HTML, μια εικόνα JPEG, αρχείο audio,...
- Κάθε αντικείμενο διευθυνσιοδοτείται με ένα **URL (Uniform Resource Locator)**. Το URL έχει δύο μέρη:
 - όνομα host
 - όνομα διαδρομής

`www.someschool.edu/someDept/pic.gif`

όνομα host

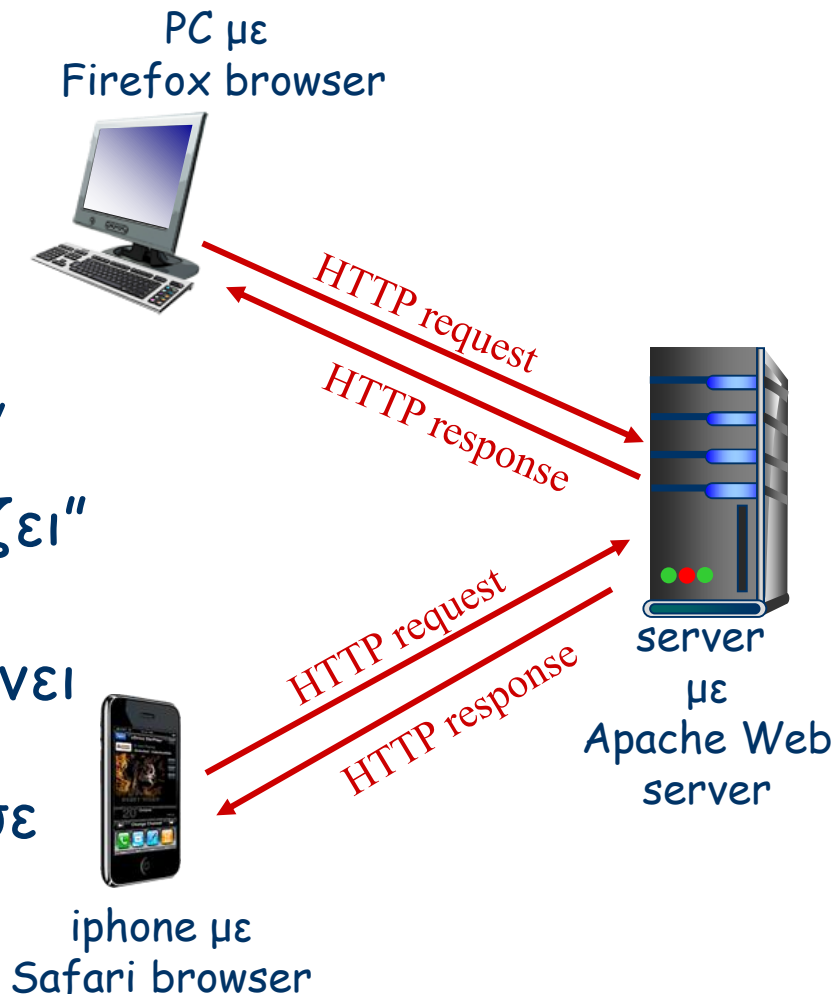
όνομα διαδρομής

HTTP



HTTP: HyperText Transfer Protocol

- Πρωτόκολλο στρώματος εφαρμογής του Web.
- Μοντέλο client/server:
 - **client**: browser που ζητάει, λαμβάνει (με HTTP πρωτόκολλο) και "απεικονίζει" Web αντικείμενα.
 - **server**: ο Web server στέλνει αντικείμενα (με HTTP πρωτόκολλο) απαντώντας σε αιτήσεις.
- HTTP 1.0: RFC 1945
- HTTP 1.1: RFC 2068



HTTP



Χρησιμοποιεί TCP:

- Ο server αναμένει αιτήσεις από τους clients.
- Ο client ξεκινά σύνδεση TCP (δημιουργεί υποδοχή) προς τη θύρα 80 του server.
- Ο server αποδέχεται τη σύνδεση TCP από τον client.
- Ο client στέλνει αίτηση για κάποιο αντικείμενο.
- Ο server του στέλνει το αντικείμενο.
- Η σύνδεση TCP κλείνει.

Το HTTP είναι "ακαταστατικό"

- Ο server δεν κρατάει πληροφορίες για προηγούμενες αιτήσεις του client.

Τα πρωτόκολλα που διατηρούν "κατάσταση" είναι πολύπλοκα!

- Πρέπει να διατηρείται πληροφορία για το παρελθόν (κατάσταση).
- Αν ο server/client χαλάσει, οι εικόνες του για την κατάσταση μπορεί να είναι ασύμβατες και πρέπει να ξαναγίνουν συμβατές.



Συνδέσεις HTTP

Μη επίμονο HTTP

- Το πολύ ένα αντικείμενο στέλνεται πάνω από μια σύνδεση TCP.
- Το HTTP/1.0 χρησιμοποιεί μη επίμονο HTTP.

Επίμονο HTTP

- Πολλά αντικείμενα μπορεί να σταλούν πάνω από την ίδια σύνδεση TCP μεταξύ client και server.
- Το HTTP/1.1 χρησιμοποιεί επίμονο HTTP στον default τρόπο λειτουργίας.

HTTP



Μη επίμονο HTTP

(περιέχει κείμενο,
αναφορές για 10
εικόνες jpeg)

Υποθέστε ότι ο χρήστης εισάγει το URL

`www.someSchool.edu/someDepartment/home.index`

1a. Ο HTTP client ξεκινά μια σύνδεση TCP προς τον HTTP server στο `www.someSchool.edu`, port 80

1b. Ο HTTP server, στον host `www.someSchool.edu`, αναμένει για σύνδεση TCP στην θύρα 80, "αποδέχεται" τη σύνδεση ειδοποιώντας τον client.

2. Ο HTTP client στέλνει HTTP *request message* (που περιέχει το URL) στην υποδοχή της σύνδεσης TCP. Το μήνυμα δείχνει ότι ο client θέλει το αντικείμενο `someDepartment/home.index`

3. Ο HTTP server λαμβάνει το μήνυμα αίτησης, σχηματίζει ένα *response message* που περιέχει το αντικείμενο που ζητήθηκε και στέλνει το μήνυμα στην υποδοχή του.

χρόνος



Μη επίμονο HTTP

4. Ο HTTP server κλείνει τη σύνδεση TCP.

5. Ο HTTP client λαμβάνει το μήνυμα απάντησης που περιέχει το αρχείο html, απεικονίζει το html. Αναλύοντας το αρχείο html, βρίσκει αναφορές για 10 αντικείμενα jpeg.

6. Τα βήματα 1-5 επαναλαμβάνονται για κάθε ένα από τα 10 αντικείμενα jpeg.

χρόνος



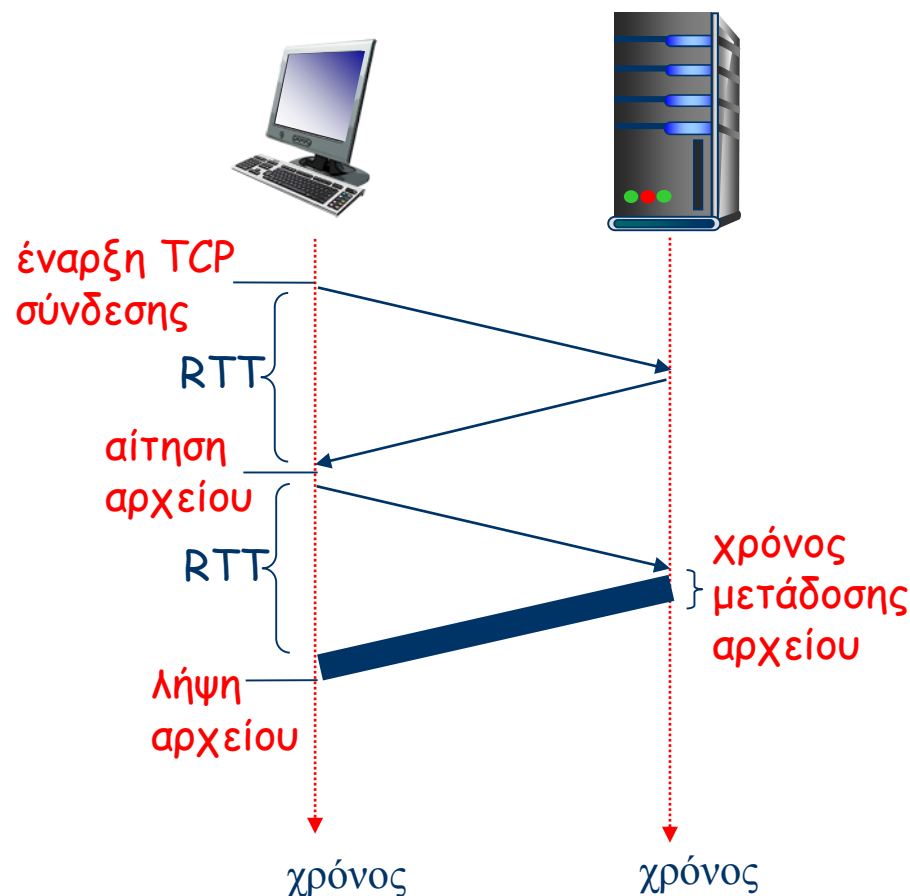
Μη επίμονο HTTP: χρόνος απόκρισης

Ορισμός του RTT: χρόνος για να πάει ένα μικρό πακέτο από τον client στον server και πίσω.

Το RTT περιλαμβάνει χρόνους διάδοσης και χρόνους αναμονής και επεξεργασίας στους ενδιάμεσους δρομολογητές.

Χρόνος απόκρισης:

- Ένα RTT για την έναρξη της σύνδεσης TCP.
- Ένα RTT για την αίτηση HTTP και την επιστροφή των πρώτων λίγων byte της HTTP απάντησης.
- Χρόνος μετάδοσης αρχείου.



$$\text{total} = 2\text{RTT} + \text{χρόνος μετάδοσης αρχείου}$$



Επίμονο HTTP

Μη επίμονο HTTP:

- Απαιτεί 2 RTT ανά object.
- Overhead στο OS για κάθε σύνδεση TCP.
- Οι browser ανοίγουν συχνά παράλληλες συνδέσεις TCP για να φέρουν αναφερόμενα objects.

Επίμονο HTTP:

- Ο server αφήνει ανοικτή τη σύνδεση μετά την αποστολή της απάντησης.
- Διαδοχικά μηνύματα HTTP μεταξύ των ίδιων client/server στέλνονται πάνω από την ανοικτή σύνδεση.

Επίμονο χωρίς συνεχή παροχή:

- Ο client κάνει νέα αίτηση μόνο όταν ληφθεί η προηγούμενη απάντηση.
- Ένα RTT για κάθε αναφερόμενο object.

Επίμονο με συνεχή παροχή:

- Ο client στέλνει αιτήσεις μόλις συναντήσει ένα αναφερόμενο object.
- Κατ' ελάχιστον ένα RTT για όλα τα αναφερόμενα objects.
- Default στο HTTP/1.1



Μήνυμα αίτησης HTTP

- Δύο τύποι μηνυμάτων HTTP: *request, response*
- **HTTP request:**
 - ASCII (μορφή αναγνώσιμη από ανθρώπους)

γραμμή αίτησης

μέθοδος URL πρωτόκολλο

GET /somedir/page.html HTTP/1.0

Host: www.someschool.edu

Connection: close ← διαχείριση σύνδεσης

User-agent: Mozilla/4.0

Accept: text/html, image/gif, image/jpeg

Accept-language: en

Carriage return,
line feed → (extra carriage return, line feed)

δηλώνει το τέλος
του μηνύματος

διαπραγμάτευση περιεχομένου

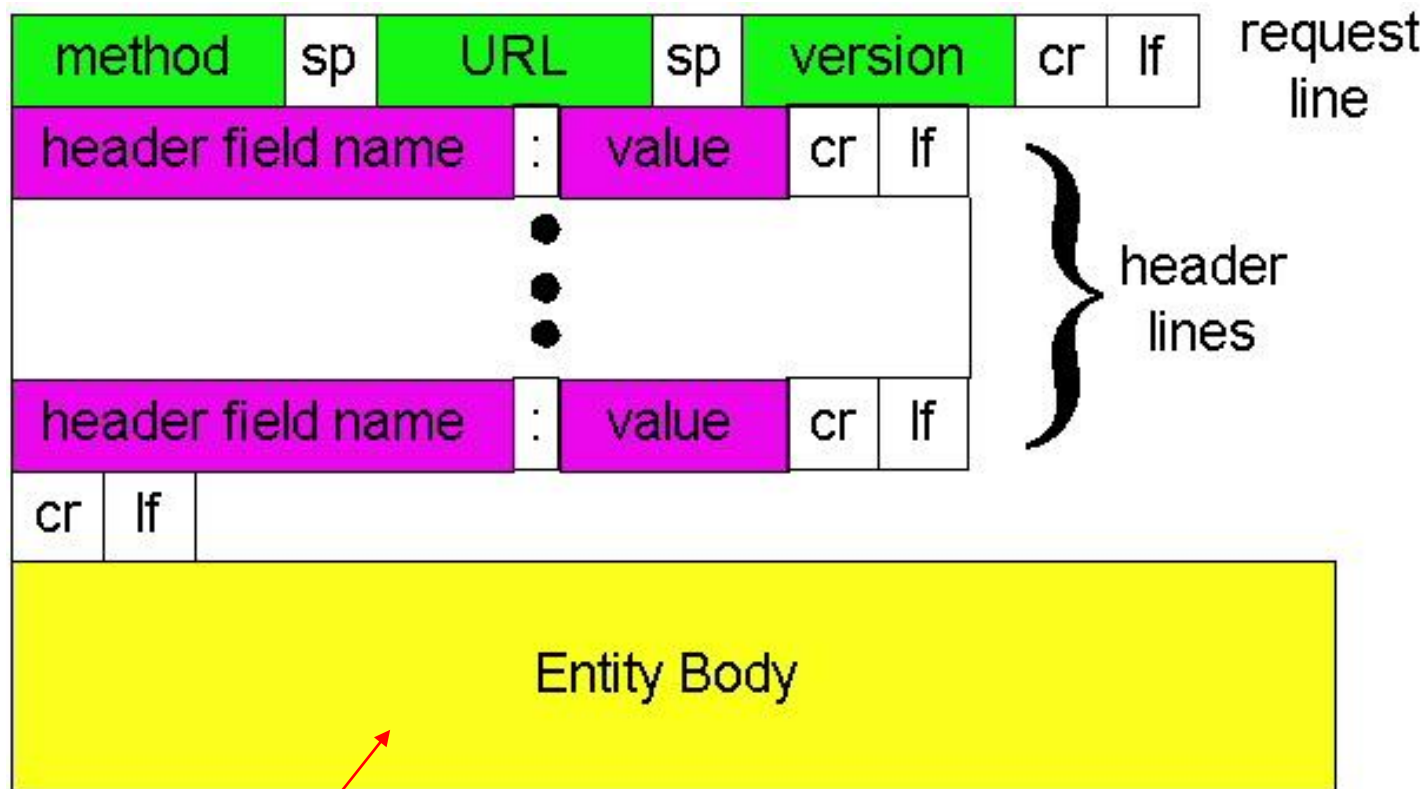
γραμμές
επικεφαλίδας

Δίκτυα υπολογιστών

HTTP



HTTP request: γενική μορφή



άδειο στο GET, χρησιμοποιείται στο POST



Αίτηση με φόρμα εισόδου

Μέθοδος Post:

- Η Web page περιέχει συχνά φόρμα εισόδου.
- Τα δεδομένα εισόδου ανεβάζονται στον server μέσα στο entity body.

Μέθοδος GET:

- Τα δεδομένα της φόρμας εισόδου μπαίνουν στο πεδίο URL της γραμμής αίτησης μετά το (?).
- Υπάρχει περιορισμός στο πλήθος των δεδομένων, που εξαρτάται από τον browser.

`www.somesite.com/animalsearch?monkeys&banana`



Άλλες μέθοδοι αίτησης

Μέθοδος HEAD:

- Παρόμοια με τη GET.
- Όταν ο server λαμβάνει αίτηση με τη μέθοδο HEAD απαντάει αφήνοντας εκτός το ζητούμενο αντικείμενο.
- Χρησιμοποιείται κυρίως για debugging.

Μέθοδος PUT:

- Επιτρέπει στον χρήστη να ανεβάσει ένα αντικείμενο σε συγκεκριμένη διαδρομή (directory) ενός server.
- Χρησιμοποιείται επίσης από εφαρμογές που χρειάζεται να ανεβάσουν αντικείμενα σε Web servers.

Μέθοδος DELETE:

- Επιτρέπει στον χρήστη ή σε μια εφαρμογή να απαλείψει ένα αντικείμενο από έναν Web server.



Τύποι μεθόδων στο HTTP

HTTP/1.0

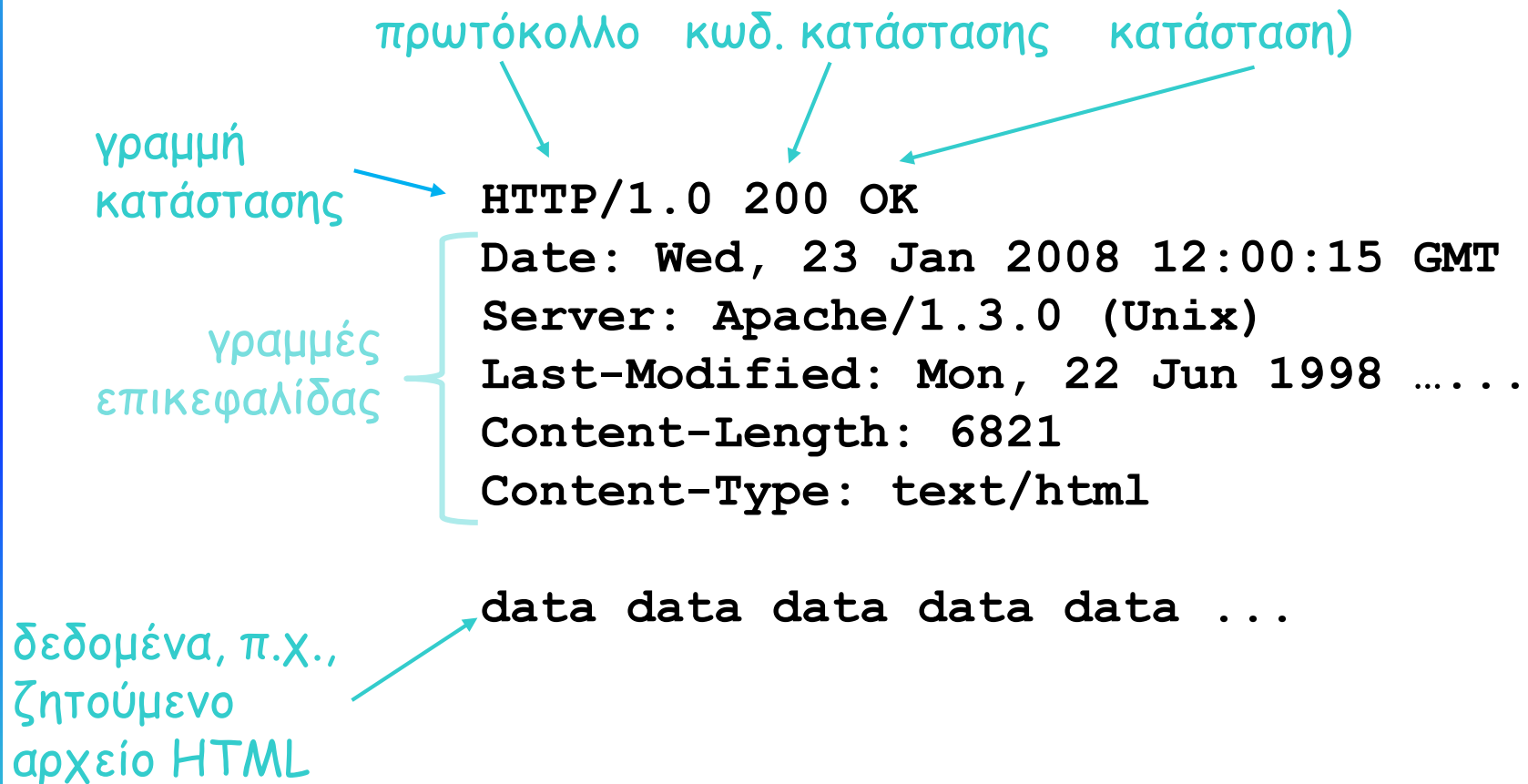
- GET
- POST
- HEAD
 - Ζητά από τον server να μην συμπεριλάβει στην απάντηση το αιτούμενο object.

HTTP/1.1

- GET, POST, HEAD
- PUT
 - Τοποθετεί αρχείο στο entity body σε διαδρομή που καθορίζεται σε πεδίο URL.
- DELETE
 - Απαλείφει το αρχείο που ορίζεται στο πεδίο URL.

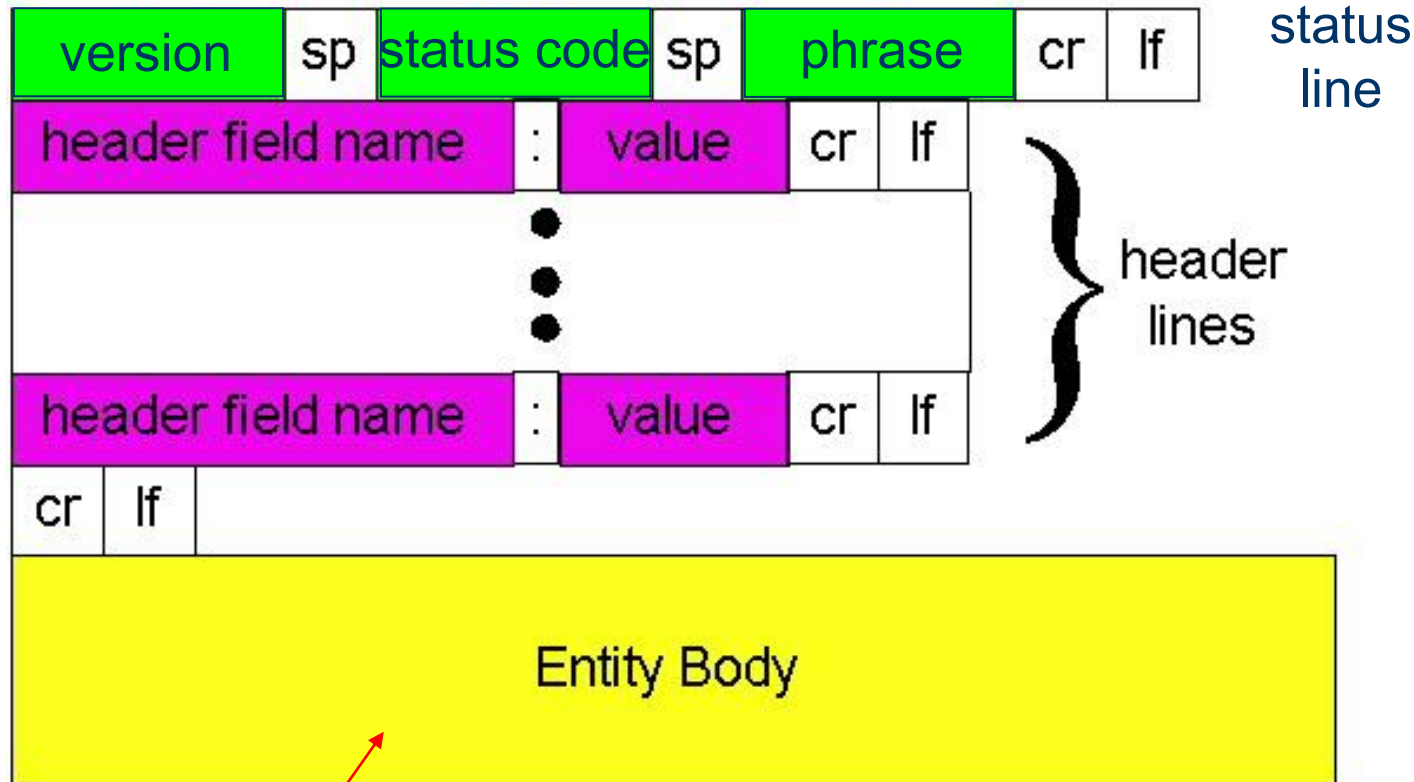


HTTP response





HTTP response: γενική μορφή



περιέχει το ζητηθέν αντικείμενο



HTTP response: κωδικοί κατάστασης

Εμφανίζονται στη γραμμή κατάστασης του μηνύματος απάντησης server-> client.

Παραδείγματα κωδικών:

200 OK

- Η αίτηση πέτυχε, το ζητηθέν αντικείμενο ακολουθεί μέσα σ' αυτό το μήνυμα.

301 Moved Permanently

- Το ζητηθέν αντικείμενο μετακινήθηκε, η νέα θέση καθορίζεται παρακάτω σ' αυτό το μήνυμα (Location:).

400 Bad Request

- Το μήνυμα αίτησης δεν έγινε κατανοητό από τον server.

404 Not Found

- Το ζητηθέν αντικείμενο δεν βρέθηκε σ' αυτόν τον server.

505 HTTP Version Not Supported



Κατάσταση user-server: cookies

Πολλές μεγάλες ιστοθέσεις χρησιμοποιούν cookies.

4 μέρη:

- 1) Γραμμή επικεφαλίδας cookie στο μήνυμα HTTP *response*.
- 2) Γραμμή επικεφαλίδας cookie στο μήνυμα HTTP *request*.
- 3) Αρχείο cookie διατηρούμενο στον host του χρήστη και υφιστάμενο διαχείριση από τον browser του χρήστη
- 4) Back-end database στο Web site

Παράδειγμα:

- Ο χρήστης Χ κάνει πάντα πρόσβαση στο Internet από το PC του.
- Επισκέπτεται το Amazon για πρώτη φορά.
- Όταν η αρχική αίτηση HTTP φθάσει στον Amazon server, ο server δημιουργεί:
 - μοναδική ID
 - εγγραφή στην backend database για την ID.

HTTP



Cookies: διατήρηση κατάστασης

client



server



cookie file



ebay 8734
amazon 1678

σύννηθες http request msg

Q Amazon server
δημιουργεί ID
1678 για τον χρήστη εγγραφή

σύννηθες http response msg
Set-cookie: 1678

σύννηθες http request msg
cookie: 1678

δράση
ειδική για
cookie

πρόσβαση



backend
database

σύννηθες http response msg

μετά μία βδομάδα:



ebay 8734
amazon 1678

σύννηθες http request msg
cookie: 1678

δράση
ειδική για
cookie

πρόσβαση

σύννηθες http response msg



Cookies

Τι μπορεί να μεταφέρουν:

- εξουσιοδότηση
- κάρτες αγοράς
- συστάσεις
- Web e-mail

Ιδιωτικό απόρρητο:

- Τα cookies επιτρέπουν στα sites να μάθουν πολλά για τους χρήστες.
- Οι χρήστες μπορεί να δίνουν όνομα και e-mail στα sites.

Πώς διατηρείται η "κατάσταση":

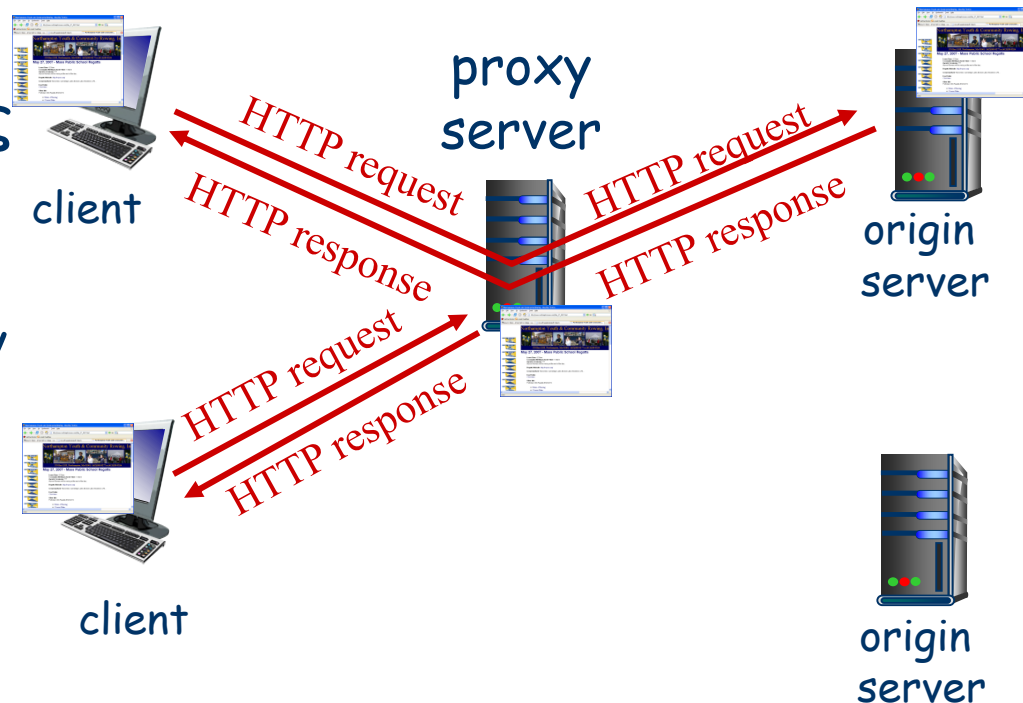
- Τα άκρα του πρωτοκόλλου: διατηρούν την κατάσταση στον πομπό/δέκτη για πολλαπλές transactions.
- Cookies: μηνύματα http μεταφέρουν την κατάσταση.



Web cache (proxy server)

Στόχος: ικανοποίηση της αίτησης του client χωρίς την ανάμιξη του αρχικού server.

- Ο χρήστης θέτει στον browser: Web accesses via cache.
- Ο browser στέλνει όλες τις αιτήσεις HTTP στην cache:
 - Αν υπάρχει το object στην cache, η cache επιστρέφει το object.
 - Αλλιώς, η cache ζητά το object από τον αρχικό server και στη συνέχεια επιστρέφει το object στον client.





Web cache

- Η cache λειτουργεί και ως client και ως server.
- Τυπικά η cache εγκαθίσταται από τον ISP (πανεπιστήμιο, εταιρία, οικιακό ISP).

Γιατί Web caching;

- Περιορίζει τον χρόνο απόκρισης στην αίτηση του client.
- Περιορίζει την κίνηση στη ζεύξη πρόσβασης ενός ιδρύματος.
- Internet με μεγάλη πυκνότητα από cache: δίνει τη δυνατότητα σε "φτωχούς" παρόχους περιεχομένου να παραδίδουν περιεχόμενο με αποτελεσματικό τρόπο (αλλά το ίδιο κάνει και το P2P file sharing).



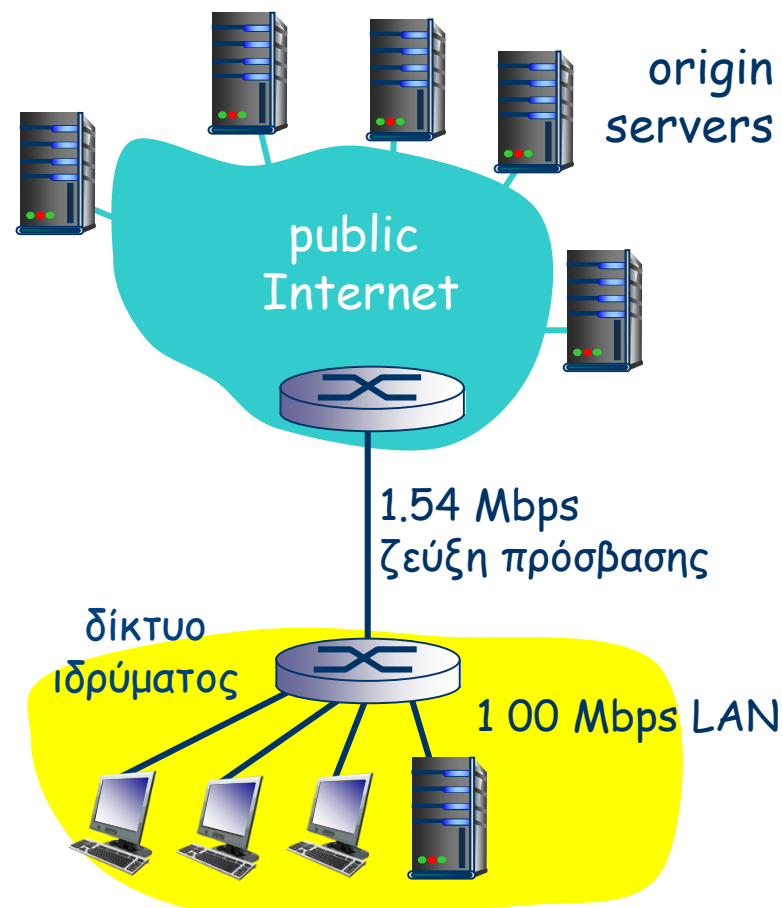
Υποθέσεις

- Μέσο μέγεθος object: 100kbit
- Μέσος ρυθμός αιτήσεων από τους browser του ιδρύματος προς τους origin servers: 15 αιτήσεις/sec
- Μέσος ρυθμός μετάδοσης προς τους browsers: 1.5 Mbps
- RTT από τον δρομολογητή του μέχρι οποιονδήποτε origin server = 2 sec

Συνέπειες

- Χρησιμοποίηση στο LAN = 1.5%
- Χρησιμοποίηση στη ζεύξη πρόσβασης = 99%
- Συνολική καθυστέρηση = καθυστέρηση Internet + καθυστέρηση πρόσβασης + καθυστέρηση LAN =
= 2 sec + minutes + microseconds

Παράδειγμα Caching





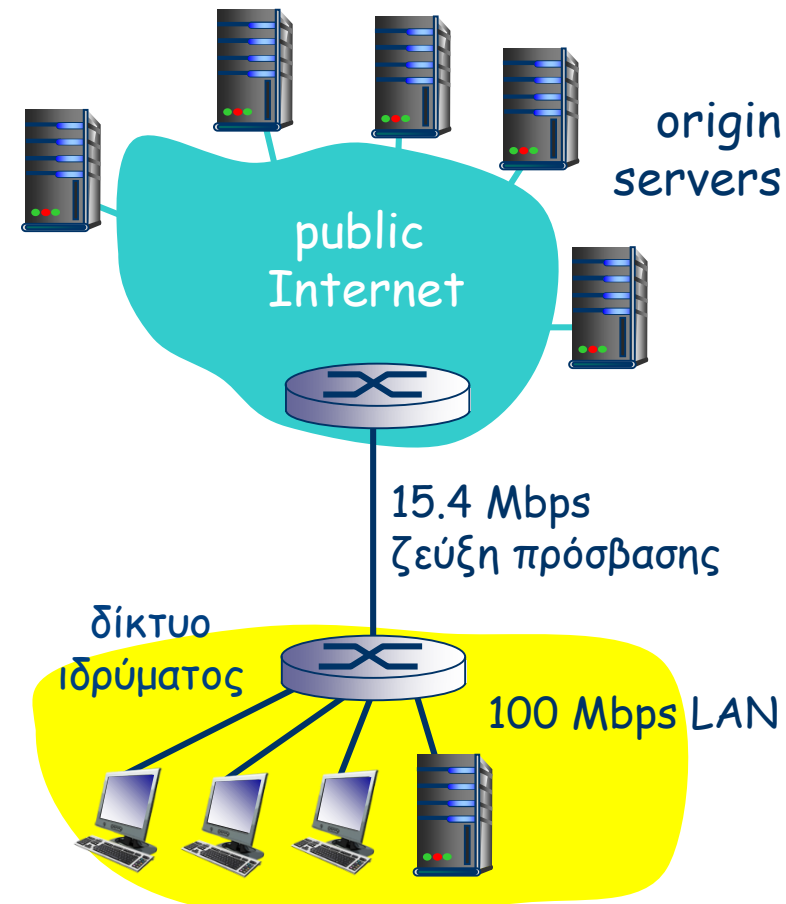
Παράδειγμα Caching

Εφικτή λύση

- Αύξηση του εύρους ζώνης της ζεύξης πρόσβασης, έστω, στα 15.4 Mbps

Συνέπεια

- Χρησιμοποίηση στο LAN = 1.5%
- Χρησιμοποίηση στη ζεύξη πρόσβασης = 9.9%
- Συνολική καθυστέρηση = καθυστέρηση Internet + καθυστέρηση πρόσβασης + καθυστέρηση LAN
= 2 sec + msec + μsec
- Συνήθως μια δαπανηρή αναβάθμιση.





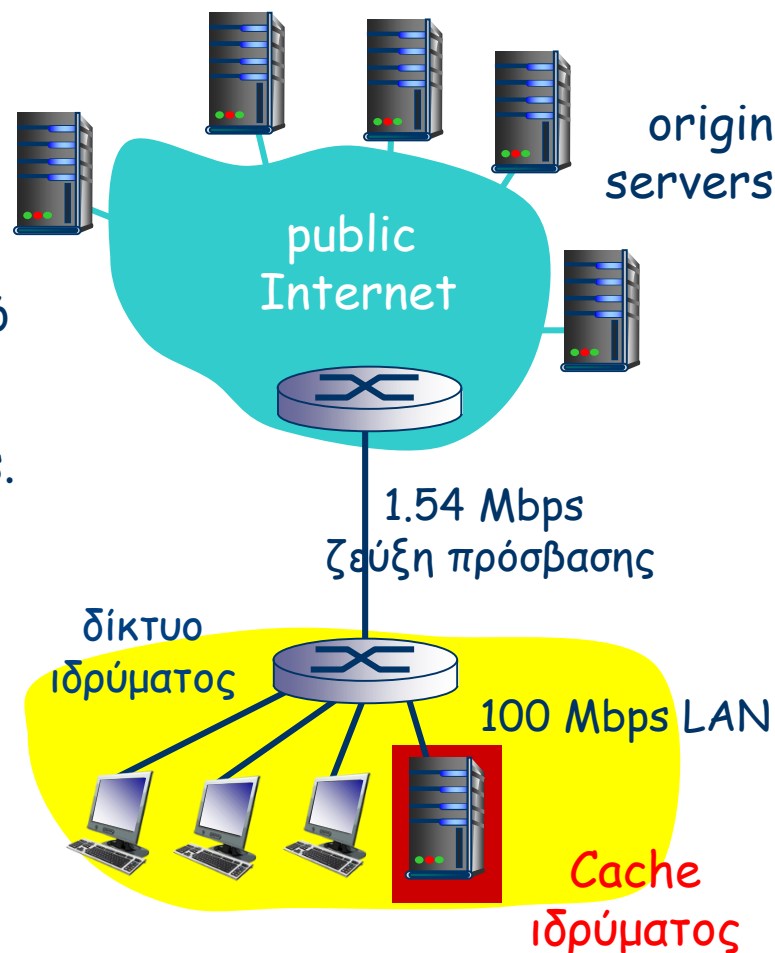
Παράδειγμα Caching

Εφικτή λύση: εγκατάσταση cache

- Έστω ότι ο ρυθμός επιτυχίας είναι 0.4

Συνέπεια

- 40% των αιτήσεων θα ικανοποιούνται σχεδόν αμέσως.
- 60% των αιτήσεων θα ικανοποιούνται από τους αρχικούς server.
- Ρυθμός δεδομένων προς browsers στη ζεύξη πρόσβασης: $0.6 * 1.5 \text{ Mbps} = 0.9 \text{ Mbps}$.
- Η χρησιμοποίηση της γραμμής πρόσβασης περιορίζεται στο $0.9 / 1.54 = 0.58$ με αποτέλεσμα αμελητέες καθυστερήσεις (έστω 10 msec).
- Συνολική μέση καθυστέρηση = καθυστέρηση Internet + καθυστέρηση πρόσβασης + καθυστέρηση LAN = $0.6 * (2.01) \text{ sec} + 0.4 * 0.01 \text{ sec} < 1.22 \text{ sec}$



HTTP



Δυναμικό GET

client



server

- **Στόχος:** να μην αποσταλλεί ένα object, αν η cache έχει αποθηκευμένη ενημερωμένη έκδοσή του.

- cache: προσδιορίζει την ημερομηνία του αποθηκευμένου αντιγράφου στην αίτηση HTTP:

If-modified-since: <date>

- server: η απάντησή του δεν περιέχει το object αν το αντίγραφο που βρίσκεται στην cache είναι ενημερωμένο:

HTTP/1.0 304 Not Modified

HTTP request
If-modified-since: <date>

object
μη
μεταβληθέν
πριν
<date>

HTTP response
**HTTP/1.0
304 Not Modified**

HTTP request
If-modified-since: <date>

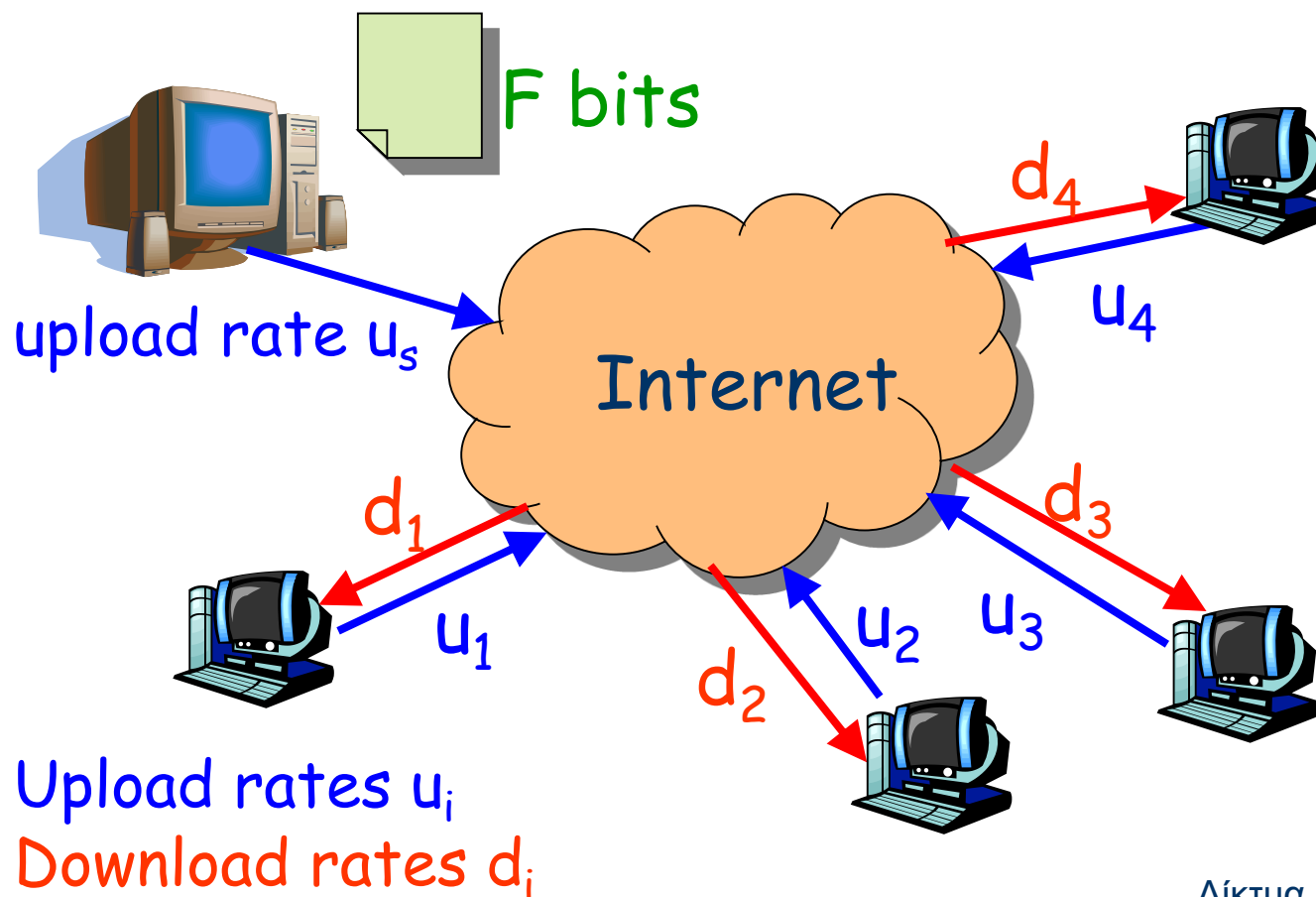
object
μεταβληθέν
μετά
<date>

HTTP response
**HTTP/1.0 200 OK
<data>**

Διανομή μεγάλου αρχείου από Server



Ερώτηση: Πόσος χρόνος απαιτείται για να διανεμηθεί ένα μεγάλο αρχείο εξ αρχής από έναν server σε N άλλους υπολογιστές;



Διανομή μεγάλου αρχείου από Server



- Μετάδοση από τον Server σε N δέκτες:
 - Ο Server πρέπει να μεταδώσει NF bits.
 - Απαιτείται, τουλάχιστον, χρόνος NF / u_s
- Λήψη δεδομένων:
 - Ο πιο αργός δέκτης λαμβάνει με ρυθμό $d_{min} = \min_i \{d_i\}$
 - Απαιτείται, τουλάχιστον, χρόνος F / d_{min}

$$\left. \begin{array}{l} \text{Χρόνος για τη διανομή} \\ \text{του } F \text{ σε } N \text{ clients με τη} \\ \text{μέθοδο client/server} \end{array} \right\} = D_{cs} \geq \max \{ NF / u_s, F / d_{min} \}$$

αυξάνει γραμμικά με το N
(για μεγάλο N)

Διανομή μεγάλου αρχείου από Server



Επιτάχυνση της διανομής μεγάλου αρχείου

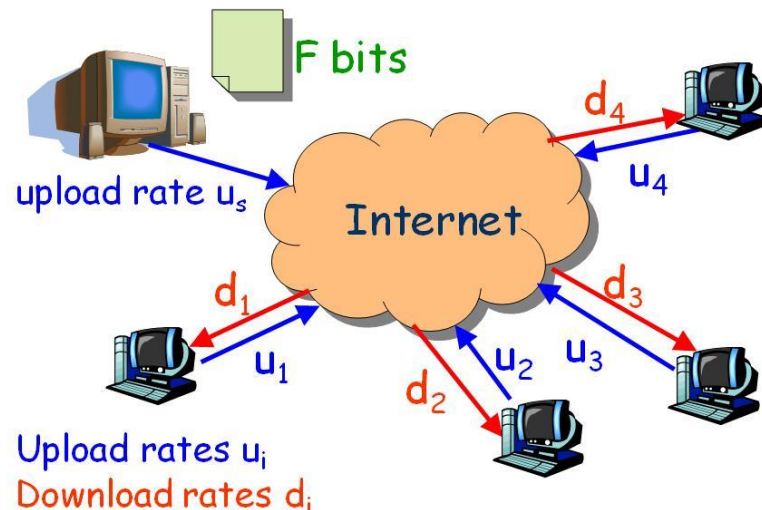
- Αύξηση του ρυθμού upload του server
 - Μεγαλύτερο εύρος ζώνης στη ζεύξη, για έναν server.
 - Για πολλαπλούς servers, μεγαλύτερο εύρος ζώνης στη ζεύξη καθενός.
 - Απαιτείται αναβάθμιση της υποδομής.
- Εναλλακτική λύση: να βοηθούν οι δέκτες στη διανομή του αρχείου
 - Peer-to-peer file sharing (P2P).
 - Οι δέκτες λαμβάνουν ένα αντίγραφο των δεδομένων.
 - Στη συνέχεια το αναδιανέμουν σε άλλους δέκτες.
 - Περιορίζεται έτσι ο φόρτος στον server.

Διανομή μεγάλου αρχείου από Server



Διανομή μεγάλου αρχείου P2P

- Ο server πρέπει να ανεβάσει ένα αντίγραφο σε χρόνο F/u_s
- Ο client i χρειάζεται χρόνο F/d_i για να το κατεβάσει.
- Πρέπει να κατέβουν (συγκεντρωτικά) NF bits.
- Ταχύτερος εφικτός ρυθμός upload (υποθέτοντας ότι όλοι οι κόμβοι στέλνουν αρχεία στον ίδιο ομότιμο): $u_s + \sum u_i$



$$D_{P2P} \geq \max \{ F/u_s, F/d_{min}, NF/(u_s + \sum u_i) \}$$

Διανομή μεγάλου αρχείου από Server



Σύγκριση μοντέλων Client-server και P2P

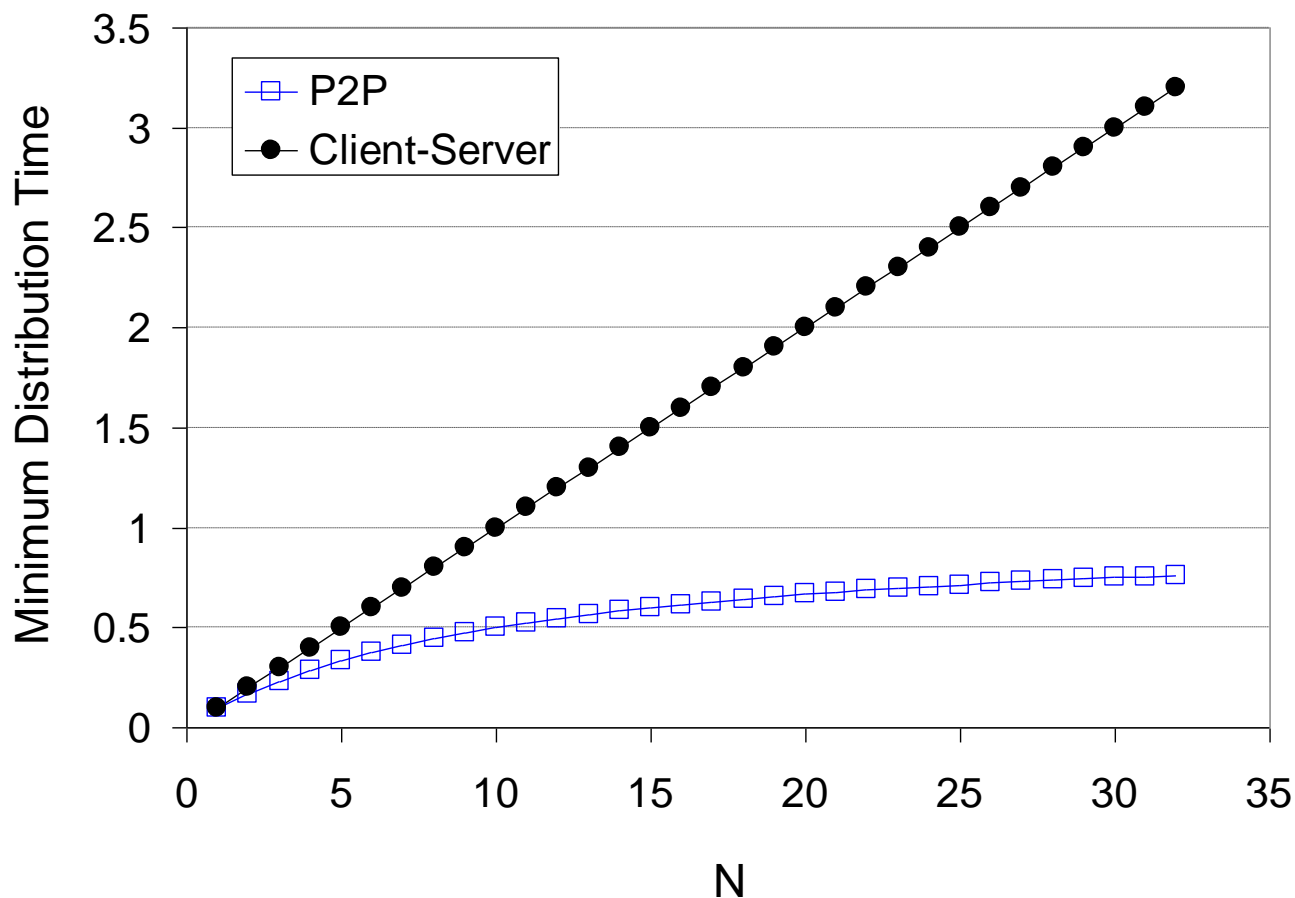
- Χρόνος Download:
 - Client-server: $\max\{NF / u_s, F / d_{\min}\}$
 - Peer-to-peer: $\max\{F / u_s, F / d_{\min}, NF / (u_s + \sum_i(u_i))\}$
- Η λύση peer-to-peer είναι αυτοκλιμακούμενη
 - Πολύ χαμηλότερη απαίτηση σε εύρος ζώνης server.
 - Ο χρόνος διανομής αυξάνει αργά με το N .
- Αλλά...
 - Οι peers μπορεί να εμφανίζονται και να φεύγουν.
 - Οι peers πρέπει να βρουν ο ένας τον άλλον.
 - Οι peers πρέπει να θέλουν να βοηθήσουν ο ένας τον άλλον.

Διανομή μεγάλου αρχείου από Server



Σύγκριση μοντέλων Client-server και P2P

client upload rate = u , $F/u=1h$, $u_s=10u$, $d_{\min} \geq u_s$



Διανομή αρχείου P2P



Δυσκολίες για τη λύση P2P

- Οι peers έρχονται και φεύγουν.
 - Οι peers συνδέονται περιστασιακά.
 - Μπορεί να εμφανιστούν ή να φύγουν οποτεδήποτε.
 - Ή να επανέλθουν με διαφορετική διεύθυνση IP.
- Πώς εντοπίζονται οι σχετικοί peers;
 - Οι peers που είναι online τώρα.
 - Οι peers που έχουν το περιεχόμενο που θέλουμε.
- Πώς να δοθούν κίνητρα στους peers να παραμείνουν στο σύστημα;
 - Γιατί να μη φύγουν αμέσως μετά τη λήξη του download;
 - Γιατί να επιφορτίζονται να δίνουν περιεχόμενο σε οποιονδήποτε άλλον;

P2P: BitTorrent



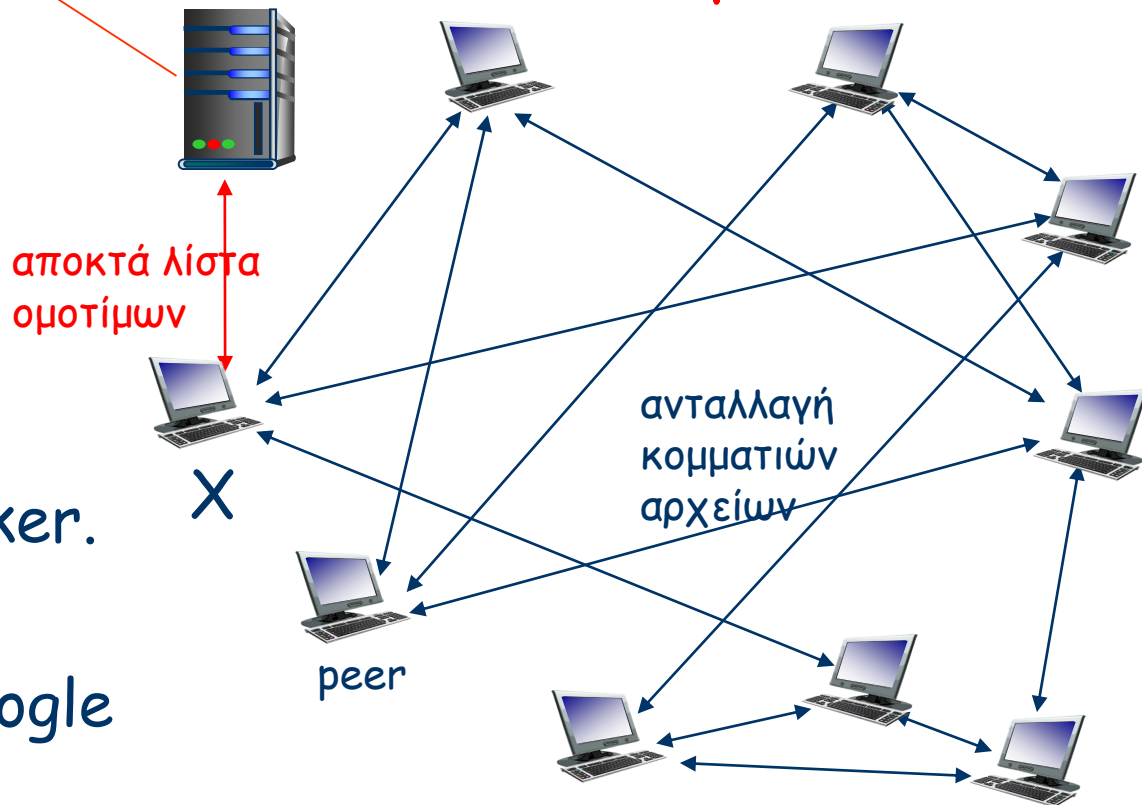
- Εστιάζει στην αποτελεσματική προσκόμιση (*fetching*), όχι στην αναζήτηση (*searching*).
- Διανομή του ίδιου αρχείου σε πολλούς peers.
- Ένας εκδότης, πολλοί host που κατεβάζουν.
- Παρεμπόδιση ελεύθερου κατεβάσματος.

P2P: BitTorrent



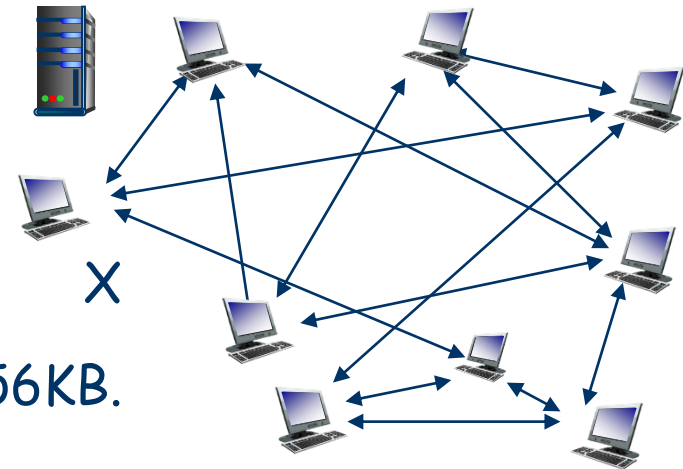
tracker: ανιχνεύει ομότιμους που συμμετέχουν στο torrent.

torrent: ομάδα ομότιμων που ανταλλάσσουν μεγάλα κομμάτια ενός αρχείου (τυπικό μέγεθος 256KB).



- Κεντριοποιημένο δίκτυο, λόγω tracker.
- .torrent αρχεία
- Αναζήτηση στο Google για .torrents

P2P: BitTorrent



- Αρχείο διαιρούμενο σε κομμάτια των 256KB.
- Ομότιμος εντασσόμενος στο torrent:
 - Δεν έχει κομμάτια, αλλά τα συγκεντρώνει με τον χρόνο.
 - Εγγράφεται στον tracker για να λάβει λίστα ομοτίμων, συνδέεται σε υποσύνολο ομοτίμων ("γείτονες").
- Ενώ κάνει download, ο ομότιμος κάνει upload κομμάτια σε άλλους ομότιμους.
- Ομότιμοι μπορεί να μπαίνουν και να βγαίνουν.
- Μόλις ένας ομότιμος αποκτήσει όλο το αρχείο, μπορεί (ατομιστικά) να φύγει ή (αλτρουϊστικά) να μείνει.

P2P: BitTorrent



Σύνοψη λειτουργίας

Μαζική παρουσία

- **Join:** Επικοινωνία με τον "tracker" server, λήψη λίστας από peers.
- **Publish:** Μέσω του tracker server.
- **Search:** Out-of-band. π.χ., χρήση του Google για να βρεθεί tracker για το αρχείο που θέλουμε.
- **Fetch:** Κατέβασμα κομματιών του αρχείου από τους σχετικούς peers. Ανέβασμα κομματιών που έχεις για αυτούς.

P2P: BitTorrent



Tracker

- Κόμβος υποδομής
 - Ανιχνεύει peers που συμμετέχουν στο torrent.
- Οι peers γράφονται στον tracker.
 - Ο peer γράφεται μόλις εμφανιστεί.
 - Ο peer πληροφορεί περιοδικά τον tracker ότι είναι ακόμη παρών.
- Ο tracker επιλέγει peers για κατέβασμα
 - Επιστρέφει μια τυχαία ομάδα από peers μαζί με τις IP διευθύνσεις τους.
 - Οπότε, ο νέος peer γνωρίζει σε ποιον θα απευθυνθεί για δεδομένα.

P2P: BitTorrent



Chunks

- Ένα μεγάλο αρχείο διαιρείται σε μικρότερα κομμάτια (chunks) σταθερού μεγέθους.
 - Τυπικό μέγεθος κομματιού 256 Kbytes.
- Επιτρέπονται ταυτόχρονες μεταφορές.
 - Κατέβασμα κομματιών από διάφορους γείτονες.
 - Αποστολή κομματιών σε άλλους γείτονες.
- Πληροφόρηση για τα κομμάτια που έχουν οι γείτονες.
 - Περιοδικές ερωτήσεις προς αυτούς για τη λίστα.
- Το αρχείο αποκτάται όταν κατεβαστούν όλα τα κομμάτια.

P2P: BitTorrent



Σειρά ζήτησης κομματιών

- Ποια κομμάτια να ζητηθούν;
 - Μπορούν να κατεβούν με τη σειρά, όπως π.χ. κάνει ένας HTTP client;
- Πρόβλημα: πολλοί peers έχουν τα αρχικά κομμάτια.
 - Οι peers έχουν λίγα να μοιραστούν μεταξύ τους.
 - Περιορισμός της κλιμάκωσης του συστήματος.
- Πρόβλημα: κατά σύμπτωση κανείς δεν έχει σπάνια κομμάτια.
 - π.χ., τα κομμάτια του τέλους του αρχείου.
 - Περιορισμός της ικανότητας περάτωσης του κατεβάσματος.
- Λύσεις: τυχαία επιλογή και τα σπάνια πρώτα.



Πρώτα το σπάνιο κομμάτι

- Ποια κομμάτια να ζητηθούν πρώτα;
 - Τα κομμάτια με τα λιγότερα διαθέσιμα αντίγραφα.
 - π.χ., το σπανιότερο κομμάτι πρώτο.
- Ωφέλη για τον peer:
 - Αποφυγή σταματήματος κατεβάσματος όταν μερικοί peers αποχωρήσουν.
- Ωφέλη για το σύστημα:
 - Αποφυγή σταματήματος κατεβάσματος για όλους τους peers που περιμένουν ένα αρχείο.
 - Ισοστάθμιση φορτίου εξισώνοντας τους αριθμούς των αντιγράφων των τεμαχίων.



Παρεμπόδιση free-riding

- Μεγάλο πλήθος χρηστών είναι free-riders.
 - Οι περισσότεροι δεν μοιράζονται αρχεία και δεν απαντούν σε ερωτήσεις.
 - Άλλοι περιορίζουν τον αριθμό των συνδέσεων ή την ταχύτητα ανεβάσματος.
- Λίγοι peers λειτουργούν ουσιαστικά ως servers.
 - Συνεισφέρουν στο κοινό καλό.
 - Καθίστανται hubs που βασικά δρουν ως servers.
- Το BitTorrent εμποδίζει το free riding.
 - Επιτρέπει τους πιο γρήγορους peers να κατεβάζουν από άλλον peer.
 - Περιστασιακά επιτρέπει σε μερικούς free loaders να κατεβάζουν.



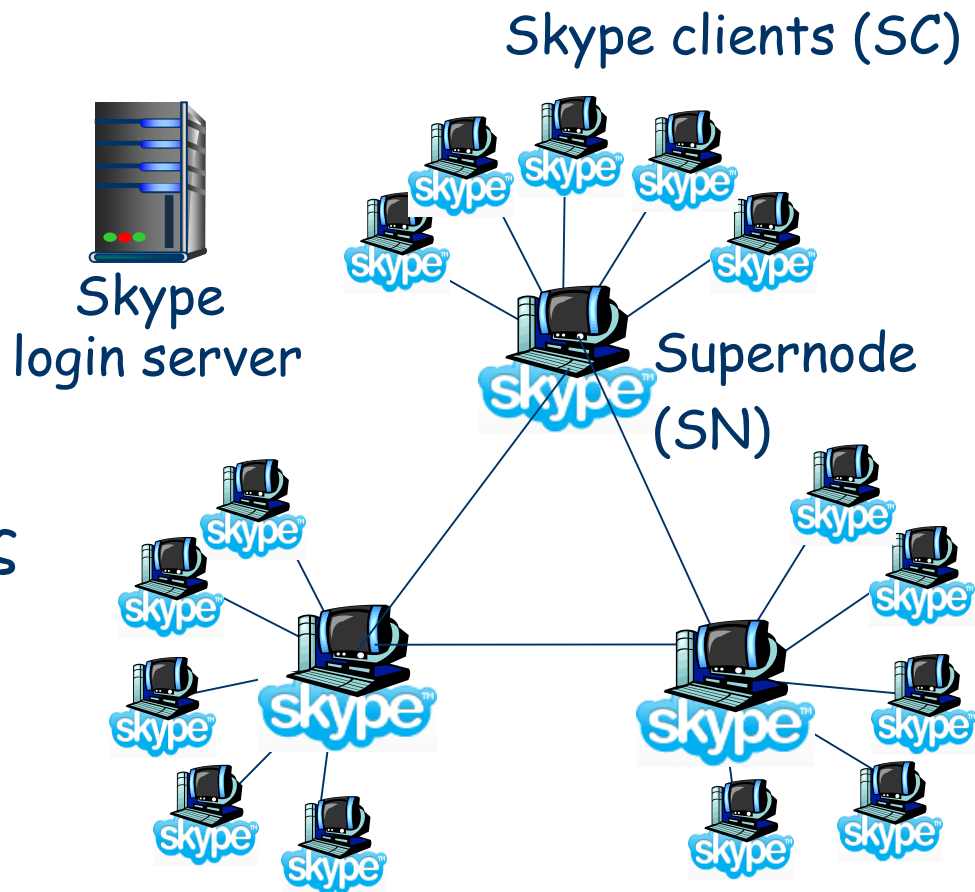
Παρεμπόδιση free-riding

- Ο peer έχει περιορισμένο εύρος ζώνης ανεβάσματος και πρέπει να το μοιράζει σε πολλούς peers.
- Προτεραιότητα ανάλογα με εύρος ζώνης ανεβάσματος.
 - Υποστήριξη των γειτόνων που ανεβάζουν με τον υψηλότερο ρυθμό.
- Επιβράβευση των 4 πρώτων γειτόνων:
 - Μετράει την ταχύτητα που στέλνει κάθε άλλος peer.
 - Ανταποκρίνεται στέλνοντας στους 4 πρώτους peers.
 - Επαναυπολογίζει τους top4 κάθε 10 sec.
- Οπτιμιστική επανεπιλογή:
 - Δοκιμάζει τυχαία έναν νέο γείτονα κάθε 30 sec.
 - Έτσι, ο νέος γείτονας έχει μια ευκαιρία να συμμετάσχει.

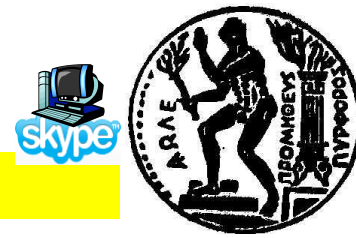
P2P: Skype



- P2P (pc-to-pc, pc-to-phone, phone-to-pc) Voice-Over-IP (VoIP) εφαρμογή.
 - Επίσης IM
- Proprietary πρωτόκολλο στρώματος εφαρμογής.
- Ιεραρχική διάρθρωση.

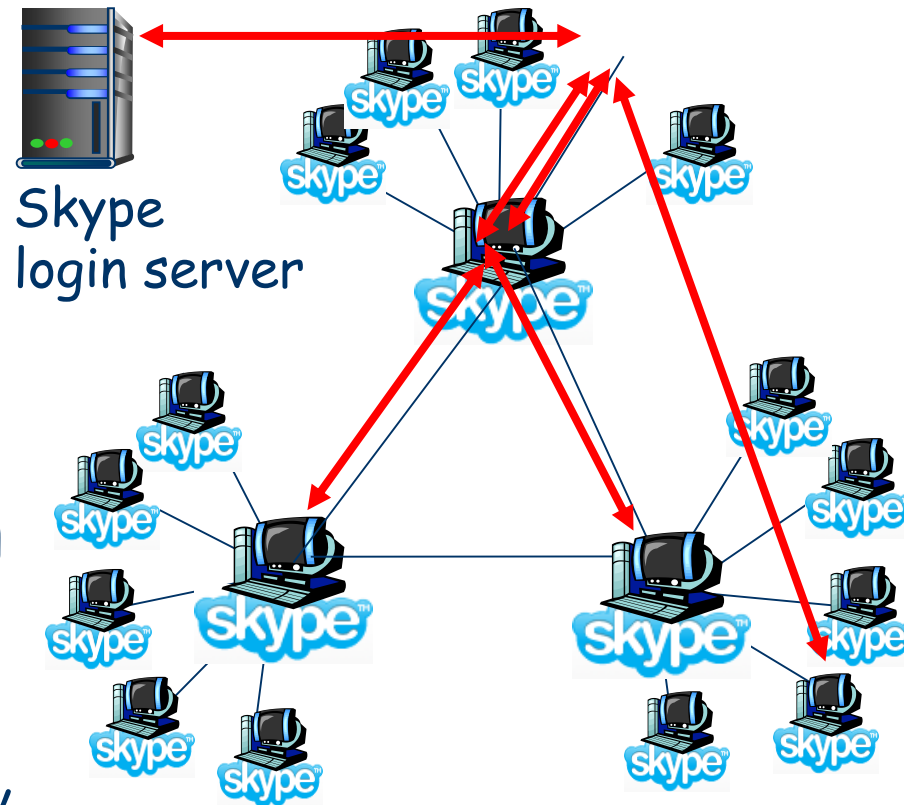


P2P: Skype



Πραγματοποίηση κλήσης

- Ο χρήστης ξεκινά το Skype.
- Ο SC εγγράφεται στον SN
 - λίστα από bootstrap SNs.
- Ο SC κάνει log in (πιστοποίηση αυθεντικότητας).
- Κλήση: SC καλεί τον SN με την ID του καλούμενου.
 - SN επικοινωνεί με άλλους SN (άγνωστο πρωτόκολλο, μπορεί πλημμύρα) για να βρει τη διεύθυνση του καλούμενου και επιστρέφει την διεύθυνση στον SC.
- Ο SC επικοινωνεί άμεσα με τον καλούμενο πάνω από TCP.

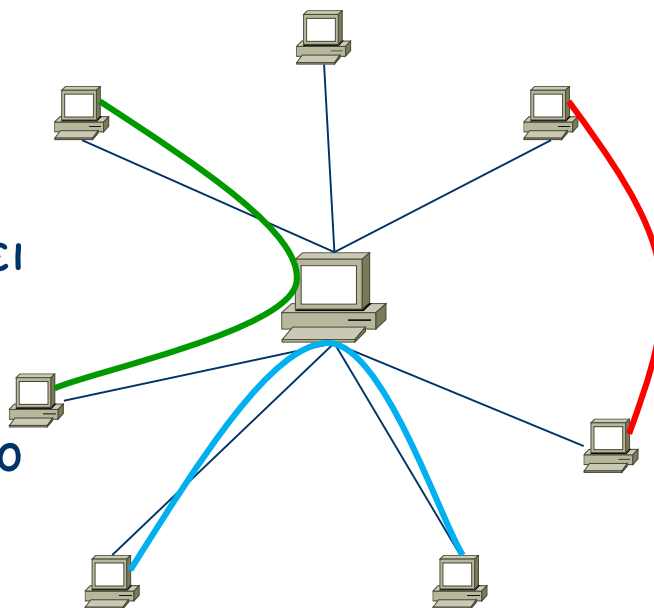




Διέλευση μέσω NAT και Firewall

➤ Αν ο καλών είναι πίσω από NAT, η σηματοδοσία κλήσης (TCP) προωθείται από κάποιον κόμβο που έχει public IP address.

➤ Αν είτε ο καλών είτε ο καλούμενος ή και οι δύο είναι πίσω από NAT, επιλέγεται από τους SN κάποιος relay κόμβος, οι peer αρχίζουν σύνοδο με τον κοινό κόμβο και κίνηση φωνής (UDP) προωθείται μέσω του κοινού κόμβου.



➤ Αν και ο καλών και ο καλούμενος είναι πίσω από NAT και από UDP-restricted firewall, τότε η κίνηση σηματοδοσίας και φωνής προωθούνται μέσω ενός άλλου κόμβου με TCP.

➤ Αν και ο καλών και ο καλούμενος έχουν public IP address, η σηματοδοσία της κλήσης (TCP) και η κίνηση φωνής (UDP) πραγματοποιείται απευθείας μεταξύ τους.