



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών

Δίκτυα Επικοινωνιών Ι

Ενότητα 2: Επίπεδο Εφαρμογής

Διδάσκων: Καθ. Ιωάννης Σταυρακάκης

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

1

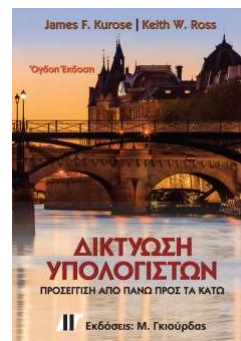
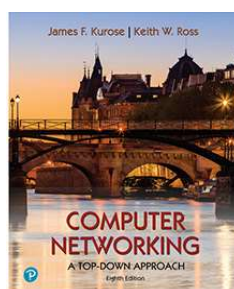
Δίκτυα Επικοινωνιών Ι

Θεματικές Ενότητες (ΘΕ) μαθήματος:

- ΘΕ1: Εισαγωγή**
(Κεφ. 1 του βιβλίου)
- ΘΕ2: Επίπεδο Εφαρμογής**
(Κεφ. 2 του βιβλίου)
- ΘΕ3: Επίπεδο Μεταφοράς**
(Κεφ. 3 του βιβλίου)
- ΘΕ4: Επίπεδο Δικτύου**
Επίπεδο Δεδομένων (Κεφ. 4 του βιβλίου)
Επίπεδο Ελέγχου (Κεφ. 5 του βιβλίου)

Συνιστώμενο Βιβλίο:
Computer Networking: A Top-Down Approach, by Kurose & Ross,
Addison-Wesley, 8^η Έκδοση

Ελληνική Μετάφραση:
Εκδόσεις : Μ. Γκιούρδας



Οι περισσότερες από τις διαφάνειες αυτής της ενότητας αποτελούν προσαρμογή και απόδοση στα ελληνικά των διαφανειών που συνοδεύουν το βιβλίο Computer Networking: A Top-Down Approach, J.F. Kurose and K.W. Ross, 8/E, Addison-Wesley.

All material copyright 1996-2021
J.F. Kurose and K.W. Ross. All Rights Reserved
Προσαρμογή και επιμέλεια της απόδοσης των πρωτότυπων διαφανειών στα ελληνικά :
Ιωάννης Σταυρακάκης

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

2

Κεφάλαιο 2: Επίπεδο εφαρμογής

Οι στόχοι μας:

- ❑ Εννοιολογικά, σχεδιαστικά θέματα πρωτοκόλλων δικτυακών εφαρμογών
 - Μοντέλα υπηρεσιών επιπέδου μεταφοράς
 - Μοντέλο πελάτη εξυπηρετητή (client-server)
 - Μοντέλο ομότιμων (peer-to-peer)
- ❑ Μαθαίνουμε για τα πρωτόκολλα εξετάζοντας δημοφιλή πρωτόκολλα επιπέδου εφαρμογής
 - HTTP
 - SMTP / POP3 / IMAP
 - DNS
 - Βίντεο Συνεχούς Ροής , Δίκτυα Διανομής Περιεχομένου
- ❑ Προγραμματίζοντας δικτυακές εφαρμογές
 - Socket API

Μερικές δικτυακές εφαρμογές

- ❑ Απομακρυσμένη σύνδεση (remote login)
- ❑ Ηλεκτρονικό ταχυδρομείο (e-mail)
- ❑ Αναζήτηση στον Ιστό
- ❑ Κοινή χρήση αρχείων μεταξύ ομότιμων (P2P file sharing)
- ❑ Κοινωνική δικτύωση
- ❑ Ιστός (web)
- ❑ Μηνύματα κειμένου (instant messaging)
- ❑ IP τηλεφωνία (voice over IP, π.χ. Skype)
- ❑ Τηλεσυνδιάσκεψη πραγματικού χρόνου (π.χ., zoom)
- ❑ Δικτυακά παιχνίδια πολλών χρηστών (multi-user network games)
- ❑ Ροή αποθηκευμένου βίντεο (YouTube, Hulu, Netflix)
- ❑ ...

* Αναπτύχθηκαν META την εισαγωγή της πολυεπίπεδης (7) αρχιτεκτονικής δικτύωσης, εκτός από τις 2 πρώτες

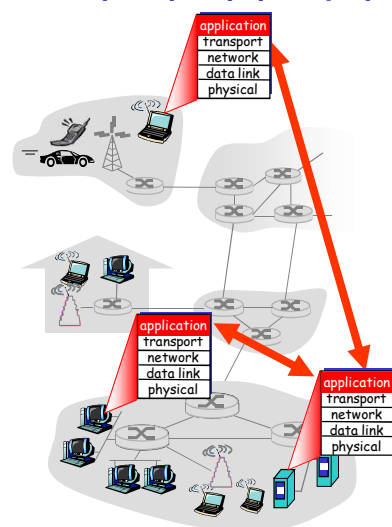
Δημιουργώντας μια δικτυακή εφαρμογή

Ανάπτυξη προγραμμάτων που

- ❑ τρέχουν σε (διαφορετικά) τερματικά συστήματα
- ❑ επικοινωνούν πάνω από το δίκτυο
- ❑ π.χ., το λογισμικό του εξυπηρετή web επικοινωνεί με το λογισμικό του browser

Δεν υπάρχει ανάγκη να γραφτεί λογισμικό για συσκευές του πυρήνα του δικτύου

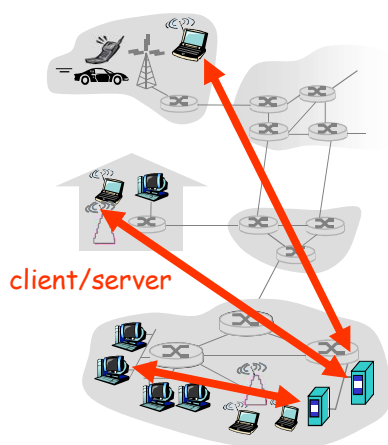
- ❑ Οι συσκευές του πυρήνα του δικτύου δεν τρέχουν εφαρμογές χρήστη
- ❑ Το ότι οι εφαρμογές είναι στα τερματικά συστήματα επιτρέπει την ταχεία ανάπτυξη, διάδοσή τους



Αρχιτεκτονικές εφαρμογών

- ❑ Πελάτη-εξυπηρετή (Client-server)
- ❑ Μεταξύ ομότιμων (Peer-to-peer (P2P))

Αρχιτεκτονική πελάτη-εξυπηρέτη



Εξυπηρέτης (server):

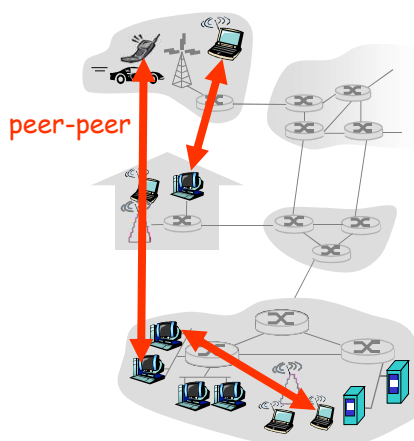
- ❑ Διαρκώς ενεργός υπολογιστής
- ❑ Μόνιμη διεύθυνση IP
- ❑ Συχνά σε Κέντρα δεδομένων (data centers) για κλιμάκωση

Πελάτες (clients):

- ❑ Επικοινωνούν με τον εξυπηρέτη
- ❑ Ενδεχομένως να έχουν διακοπτόμενη σύνδεση
- ❑ Ενδεχομένως έχουν δυναμική διεύθυνση IP
- ❑ Δεν επικοινωνούν απευθείας μεταξύ τους
- ❑ Examples: HTTP, IMAP, FTP

Αρχιτεκτονική μεταξύ ομότιμων

- ❑ Δεν υπάρχει διαρκώς ενεργός εξυπηρέτης
- ❑ Τυχαία τερματικά συστήματα επικοινωνούν απευθείας μεταξύ τους
- ❑ Ομότιμοι ζητούν υπηρεσίες από άλλους ομότιμους, παρέχουν υπηρεσίες με τη σειρά τους σε άλλους ομότιμους
 - Αυτοεπεκτασιμότητα – νέοι ομότιμοι αυξάνουν την ικανότητα παροχής, καθώς και τη ζήτηση, υπηρεσιών
- ❑ Ομότιμοι έχουν διακοπτόμενη σύνδεση και αλλάζουν IP διευθύνσεις
 - Πολύπλοκη διαχείριση
- ❑ Παράδειγμα: P2P διαμοιρασμός αρχείων (file sharing)



Διεργασίες που επικοινωνούν

Διεργασία (process): πρόγραμμα που τρέχει σε ένα υπολογιστή (process: an instantiation of a program)

- Στον **ίδιο υπολογιστή**, δύο διεργασίες επικοινωνούν χρησιμοποιώντας **δια-διεργασιακή επικοινωνία** (ορίζεται από το λειτουργικό).
- Οι διεργασίες σε **διαφορετικούς υπολογιστές** επικοινωνούν ανταλλάσσοντας **μηνύματα**

clients, servers

Διεργασία πελάτη: διεργασία που εκκινεί την επικοινωνία

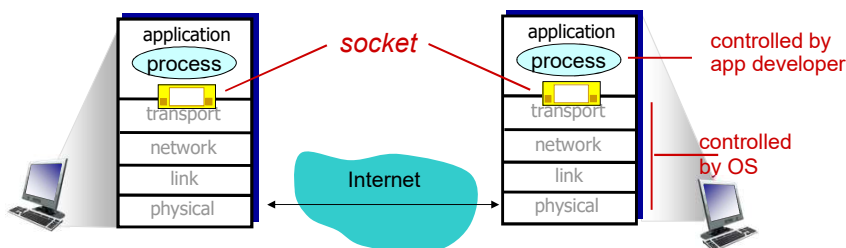
Διεργασία εξυπηρέτης: διεργασία που αναμένει να επικοινωνήσουν μαζί της

- Σημείωση: εφαρμογές με αρχιτεκτονικές P2P έχουν διεργασίες πελάτες και διεργασίες εξυπηρέτες

Sockets

Socket / API (Application Programmer's Interface) : διεπαφή ανάμεσα στην εφαρμογή και το δίκτυο (επίπεδο μεταφοράς)

- Η διεργασία στέλνει/λαμβάνει μηνύματα στο/από το socket της
- socket ανάλογο "πόρτας"
 - Η διεργασία αποστολέας στέλνει μήνυμα "έξω από την πόρτα"
 - Η διεργασία αποστολέας βασίζεται στην υποκείμενη δικτυακή υποδομή (επίπεδα Μεταφοράς, Δικτύου, ...) στην άλλη πλευρά της πόρτας για να παραδώσει το μήνυμα στο socket της διεργασίας παραλήπτη
 - Εμπλέκονται δύο sockets: ένα σε κάθε πλευρά



Διευθυνσιοδότηση διεργασιών

- Για να λαμβάνει μία διεργασία μηνύματα πρέπει να έχει ένα αναγνωριστικό (*identifier*)
- Κάθε υπολογιστής έχει μία μοναδική 32-bit διεύθυνση IP
- **E:** Αρκεί η IP διεύθυνση του υπολογιστή που τρέχει η διεργασία για την αναγνώριση της διεργασίας;
- **A:** Όχι, πολλές διεργασίες μπορεί να τρέχουν στον ίδιο υπολογιστή
- Το **αναγνωριστικό** περιλαμβάνει τόσο τη **διεύθυνση IP** όσο και τον **αριθμό θύρας** που σχετίζεται με τη διεργασία στον υπολογιστή
- Παραδείγματα αριθμών θύρας:
 - Εξυπηρέτης HTTP: 80
 - Εξυπηρέτης Mail: 25
- Για να σταλεί μήνυμα HTTP στον εξυπηρέτη web `gaia.cs.umass.edu`:
 - **Διεύθυνση IP:** 128.119.245.12
 - **Αριθμός θύρας:** 80
- Περισσότερα σε λίγο...

Ένα πρωτόκολλο επιπέδου εφαρμογής ορίζει

- Τα είδη των μηνυμάτων που ανταλλάσσονται,
 - π.χ. αίτηση, απάντηση
 - Σύνταξη μηνύματος:
 - τι πεδία στα μηνύματα & πώς αυτά διαχωρίζονται
 - Σημασιολογία των πεδίων
 - σημασία της πληροφορίας των πεδίων
 - Κανόνες για το **πότε** και **πώς** οι διεργασίες στέλνουν και απαντούν σε μηνύματα
- Ανοιχτά πρωτόκολλα:**
 - ορίζονται στα RFCs, όλοι έχουν πρόσβαση σε όλες τις λεπτομέρειες του πρωτοκόλλου
 - επιτρέπουν τη διαλειτουργικότητα
 - π.χ. HTTP, SMTP
 - Ιδιοταγή πρωτόκολλα:**
 - π.χ. Skype, Zoom

Τι υπηρεσίες μεταφοράς απαιτούν οι εφαρμογές;

Ακεραιότητα δεδομένων

- ❑ Κάποιες εφαρμογές (π.χ. μεταφορά αρχείου, διαδικτυακές συναλλαγές) απαιτούν 100% αξιόπιστη μεταφορά δεδομένων
- ❑ Άλλες εφαρμογές (π.χ. ήχου), ανέχονται κάποιες απώλειες

Χρονισμός

- ❑ κάποιες εφαρμογές (π.χ. τηλεφωνία Διαδικτύου, διαδραστικά παιχνίδια) απαιτούν χαμηλή καθυστέρηση για να είναι «αποτελεσματικές»

Ρυθμαπόδοση (Throughput)

- ❑ Κάποιες εφαρμογές (π.χ., multimedia) απαιτούν κάποια ελάχιστη ρυθμαπόδοση για να είναι «αποτελεσματικές»
- ❑ Άλλες εφαρμογές («ελαστικές») «βολεύονται» με όση ρυθμαπόδοση πάρουν

Ασφάλεια

- ❑ Κρυπτογράφηση, ακεραιότητα δεδομένων...

Απαιτήσεις υπηρεσιών μεταφοράς: κοινές εφαρμογές

Εφαρμογή	Απώλεια δεδομένων	Ρυθμαπόδοση	Ευαισθησία ως προς το χρόνο
Μεταφορά αρχείου	όχι απώλειες	ελαστική	όχι
e-mail	όχι απώλειες	ελαστική	όχι
Έγγραφα Web	όχι απώλειες	ελαστική	όχι
Ήχος/ βίντεο πραγματικού χρόνου	ανοχή στις απώλειες	ήχος: 5kbps-1Mbps, βίντεο:10kbps-5Mbps	ναι, 10δες msec
Αποθηκευμένος ήχος/βίντεο	ανοχή στην απώλειες	ίδια με παραπάνω	ναι, λίγα secs
Διαδραστικά παιχνίδια	ανοχή στην απώλειες	ως λίγα kbps	ναι, 10δες msec
Μηνύματα κειμένου	όχι απώλειες	ελαστική	ναι και όχι

Υπηρεσίες των πρωτοκόλλων μεταφοράς

Υπηρεσία του TCP:

- ❑ **αξιόπιστη μεταφορά δεδομένων** μεταξύ διεργασίας αποστολής και λήψης
- ❑ **έλεγχος ροής:** ο αποστολέας δεν υπερφορτώνει τον παραλήπτη
- ❑ **έλεγχος συμφόρησης:** επιβράδυνση αποστολέα όταν το δίκτυο είναι υπερφορτωμένο
- ❑ **συνδεδεισμενός (connection-oriented):** απαιτείται αρχικοποίηση μεταξύ διεργασιών πελάτη και εξυπηρετή
- ❑ **δεν παρέχει:** συγχρονισμό, εγγυήσεις ως προς ελάχιστη ρυθμαπόδοση, ασφάλεια

Υπηρεσία του UDP:

- ❑ **αναξιόπιστη μεταφορά δεδομένων** μεταξύ διεργασίας αποστολής και λήψης
- ❑ **δεν παρέχει:** αξιοπιστία, έλεγχο ροής, έλεγχο συμφόρησης, εγγύηση ως προς τον χρόνο ή τη ρυθμαπόδοση, ασφάλεια, η αρχικοποίηση σύνδεσης.

E: Γιατί υπάρχει το UDP τότε;

Εφαρμογές Διαδικτύου, και πρωτόκολλα μεταφοράς

<u>εφαρμογή</u>	<u>Πρωτόκολλο επιπέδου εφαρμογής</u>	<u>πρωτόκολλο μεταφοράς</u>
file transfer/download	FTP [RFC 959]	TCP
e-mail	SMTP [RFC 5321]	TCP
Web documents	HTTP 1.1 [RFC 7320]	TCP
Internet telephony	SIP [RFC 3261], RTP [RFC 3550], or proprietary	TCP or UDP
streaming audio/video	HTTP [RFC 7320], DASH	TCP
interactive games	WOW, FPS (proprietary)	UDP or TCP

«Ασφαλίζοντας» το TCP

TCP & UDP sockets:

- Δεν υπάρχει κρυπτογράφηση
- Κωδικοί που στέλνονται σαν κείμενο (ακρυπτογράφητο) στο socket διατρέχουν το διαδίκτυο σαν κείμενο (!)

Transport Layer Security (TLS)

- Παρέχει κρυπτογραφημένες συνδέσεις TCP
- Διασφάλιση δεδομένων
- Αυθεντικοποίηση από άκρη σε άκρη

TLS υλοποιείται στο επίπεδο εφαρμογής

- Εφαρμογές χρησιμοποιούν TSL libraries, οι οποίες στη συνέχεια χρησιμοποιούν TCP
- Ακρυπτογράφητο κείμενο που στέλνεται στο "socket" διατρέχει το διαδίκτυο *κρυπτογραφημένο*
- Περισσότερα στο Κεφ 8

Αρχές δικτυακών εφαρμογών (περίληψη)

- Εφαρμογές: καταναμημένες διεργασίες που αλληλοεπιδρούν, ανταλλάσσοντας μηνύματα.
- Μοντέλο Πελάτη-Εξυπηρετή και μεταξύ ομοτίμων (P2P)
- Sockets, διευθυνσιοδότηση διεργασιών
- Υπηρεσίας επιπέδου μεταφοράς – παραδείγματα υπηρεσιών
- Ασφάλεια και sockets

Επίπεδο Εφαρμογής: επισκόπηση

- Αρχές δικτυακών εφαρμογών
- Web και HTTP
- E-mail, SMTP, IMAP
- Domain Name System DNS
- Εφαρμογές P2P
- video streaming και δίκτυα διανομής περιεχομένου
- socket programming with UDP and TCP



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

19

Web και HTTP (επιγραμματικά)

- Web, HTTP επισκόπηση
- HTTP συνδέσεις
 - ❖ TCP, ακαταστατική»
 - ❖ Παραμένον, μη-παραμένον
- HTTP μηνύματα
 - ❖ Αιτήσεις, αποκρίσεις
- HTTP cookies
- Web caches
- Conditional HTTP GET
- HTTP/2, HTTP/3

20

Web και HTTP

Μια σύντομη επισκόπηση ...

- ❑ Μία **ιστοσελίδα** (web page) αποτελείται από **αντικείμενα (objects)**, που μπορεί να είναι αποθηκευμένα σε διαφορετικούς εξυπηρέτες web.
- ❑ Το αντικείμενο μπορεί να είναι αρχείο HTML, εικόνα JPEG, Java applet, αρχείο ήχου, ...
- ❑ Η ιστοσελίδα διευθυνσιοδοτείται με ένα **URL** και αποτελείται από ένα **βασικό αρχείο HTML** που περιλαμβάνει **διάφορα αντικείμενα στα οποία γίνεται αναφορά, καθένα** από τα οποία διευθυνσιοδοτείται επίσης με ένα **URL**, π.χ.

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

όνομα υπολογιστή
όνομα διαδρομής

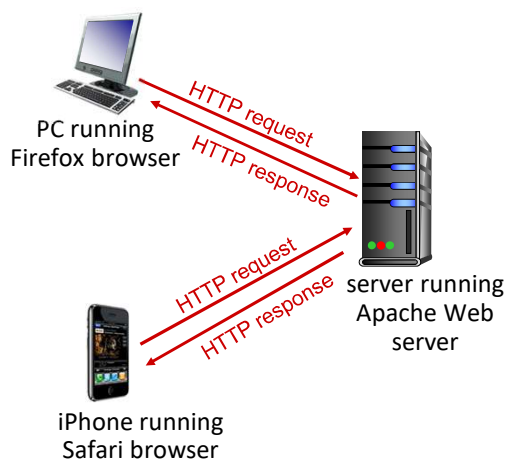
Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

21

Επισκόπηση του HTTP

HTTP: HyperText Transfer Protocol
(Πρωτόκολλο Μεταφοράς Υπερκειμένου)

- ❑ Πρωτόκολλο επιπέδου εφαρμογής του Web
- ❑ Μοντέλο πελάτη/εξυπηρέτη
 - **πελάτης:** browser που ζητά, λαμβάνει, (χρησιμοποιώντας το HTTP πρωτόκολλο) και “παρουσιάζει” αντικείμενα του Web
 - **εξυπηρέτης:** Web server στέλνει (χρησιμοποιώντας το HTTP πρωτόκολλο) αντικείμενα ως απόκριση σε αιτήσεις



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

22

Επισκόπηση του HTTP (συνέχεια)

HTTP Χρησιμοποιεί TCP:

- ❑ ο πελάτης ξεκινά σύνδεση TCP (δημιουργεί socket) με τον εξυπηρέτη, θύρα 80
- ❑ ο εξυπηρέτης αποδέχεται τη σύνδεση TCP από τον πελάτη
- ❑ ανταλλάσσονται μηνύματα HTTP (μηνύματα πρωτοκόλλου επιπέδου εφαρμογής) μεταξύ browser (πελάτης HTTP) και Web server (εξυπηρέτης HTTP)
- ❑ κλείνει η σύνδεση TCP

Το HTTP είναι ακαταστατικό (stateless)

- ❑ ο εξυπηρέτης δε διατηρεί πληροφορία σχετικά με προηγούμενες αιτήσεις του πελάτη

παρενθετικά

Τα πρωτόκολλα που διατηρούν την κατάσταση (state) είναι πολύπλοκα

- ❑ πρέπει να διατηρείται προηγούμενη ιστορία (κατάσταση)
- ❑ αν ο εξυπηρέτης /πελάτης καταρρεύσει, οι απόψεις τους για την κατάσταση μπορεί να μην είναι συνεπείς.

Συνδέσεις HTTP: 2 ειδών

Μη παραμένον HTTP (Non-persistent HTTP)

1. Ανοίγει σύνδεση TCP
2. Στέλνεται το πολύ ένα αντικείμενο πάνω από την σύνδεση TCP
3. Κλείνει η σύνδεση TCP

η λήψη πολλαπλών αντικειμένων
απαιτεί πολλαπλές συνδέσεις

Παραμένον HTTP (Persistent HTTP)

- ❑ Ανοίγει σύνδεση TCP
- ❑ Πολλαπλά αντικείμενα μπορεί να σταλούν πάνω από την ίδια σύνδεση TCP μεταξύ πελάτη και εξυπηρέτη
- ❑ Κλείνει η σύνδεση TCP

Μη παραμένον HTTP: παράδειγμα

Υποθέστε ότι ο χρήστης εισάγει το URL

www.someSchool.edu/someDepartment/home.index

(περιέχει κείμενο, αναφορές σε 10 εικόνες jpeg)



1a. Ο πελάτης HTTP ξεκινά σύνδεση TCP προς τη διεργασία του εξυπηρέτη HTTP στο `www.someSchool.edu` στη θύρα 80

2. Ο HTTP client στέλνει HTTP μήνυμα αίτησης (request message) (που περιέχει το URL) στη socket της σύνδεσης TCP. Το μήνυμα υποδεικνύει ότι ο client θέλει το αντικείμενο `someDepartment/home.index`

χρόνος



1b. Ο εξυπηρέτης HTTP στον υπολογιστή `www.someSchool.edu` περιμένει σύνδεση TCP στη θύρα 80, "αποδέχεται" τη σύνδεση, ειδοποιεί τον πελάτη

3. Ο εξυπηρέτης HTTP λαμβάνει το μήνυμα αίτησης, σχηματίζει μήνυμα απόκρισης (response message) που περιέχει το αντικείμενο που ζητήθηκε και στέλνει το μήνυμα στη socket του

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

25

Μη παραμένον HTTP: παράδειγμα

Υποθέστε ότι ο χρήστης εισάγει το URL

www.someSchool.edu/someDepartment/home.index



5. Ο πελάτης HTTP λαμβάνει το μήνυμα απόκρισης που περιέχει το html αρχείο, απεικονίζει την html. Επεξεργαζόμενος το html αρχείο, βρίσκει 10 αναφερόμενα αντικείμενα jpeg

χρόνος



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



4. Ο εξυπηρέτης HTTP κλείνει τη σύνδεση TCP

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

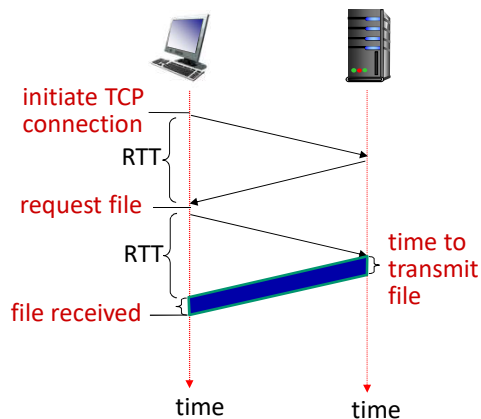
26

Μη παραμένον HTTP: χρόνος απόκρισης

RTT (round trip time): ο χρόνος που απαιτείται για ένα μικρό πακέτο να ταξιδέψει από τον πελάτη στον εξυπηρέτη και να επιστρέψει πίσω

Χρόνος απόκρισης:

- ένα RTT για την έναρξη της σύνδεσης TCP
- ένα RTT για την αίτηση HTTP και επιστροφή των πρώτων bytes της απόκρισης HTTP
- χρόνος μετάδοσης αρχείου



χρόνος απόκρισης του *non-persistent HTTP* = $2RTT$ + χρόνος μετάδοσης αρχείου

Παραμένον HTTP (HTTP1.1)

Θέματα με το μη παραμένον HTTP:

- απαιτεί 2 RTT ανά αντικείμενο
- το λειτουργικό επιβαρύνεται για κάθε σύνδεση TCP
- οι browsers ανοίγουν συχνά **παράλληλες συνδέσεις TCP** για να φέρουν αντικείμενα που αναφέρονται

Παραμένον HTTP (HTTP1.1)

- ο εξυπηρέτης αφήνει τη σύνδεση ανοιχτή αφού στείλει την απόκριση
- τα επόμενα HTTP μηνύματα μεταξύ του ίδιου πελάτη / εξυπηρέτη στέλνονται μέσω της ανοιχτής σύνδεσης
- ο πελάτης στέλνει αιτήσεις μόλις συναντήσει αναφερόμενο αντικείμενο
- ένα RTT για όλα τα αναφερόμενα αντικείμενα μειώνοντας τον χρόνο στον μισό

Μήνυμα αίτησης HTTP

- δύο είδη HTTP μηνυμάτων : *αίτηση (request)*, *απόκριση (response)*
- μήνυμα HTTP αίτησης :
 - ASCII (μορφή αναγνώσιμη από τον άνθρωπο)

γραμμή αίτησης (request line)
(εντολές GET, POST, HEAD) →

γραμμές κεφαλίδας
(header lines)

carriage return, line feed
στην αρχή γραμμής
υποδεικνύει το τέλος
της κεφαλίδας

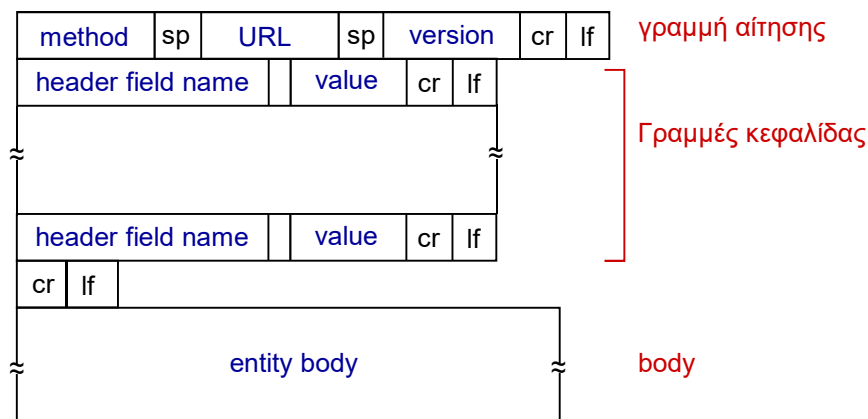
```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10.15; rv:80.0) Gecko/20100101 Firefox/80.0 \r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Connection: keep-alive\r\n
\r\n
```

carriage return character
line-feed character

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

29

Μήνυμα αίτησης HTTP: format



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

30

Άλλα μηνύματα αίτησης του HTTP

Μέθοδος POST :

- Ιστοσελίδες συχνά περιλαμβάνουν φόρμες
- Τα στοιχεία της συμπληρωμένης φόρμας στέλνονται από τον πελάτη στον εξυπηρέτη στο «entity body» του μηνύματος αίτησης του HTTP POST

Μέθοδος GET (και για αποστολή δεδομένων του χρήστη στον εξυπηρέτη):

- Τα δεδομένα περιλαμβάνονται στο URL πεδίο του HTTP GET μηνύματος αίτησης (μετά από ένα '?'):

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

Μέθοδος HEAD:

- Ζητά μόνο κεφαλίδες που θα είχαν σταλεί εάν το συγκεκριμένο URL είχε ζητηθεί με την μέθοδο HTTP GET.

Μέθοδος PUT:

- Ανεβάζει ένα αρχείο / αντικείμενο στον εξυπηρέτη
- Αντικαθιστά αρχείο που τυχόν ήδη υπάρχει στο URL με το περιεχόμενο στο «entity body» του HTTP POST μηνύματος αίτησης

Μήνυμα απόκρισης HTTP

γραμμή κατάσταση
(protocol status code status phrase)

γραμμές κεφαλίδας
(header lines)

data, e.g., αρχείο
HTML που ζητήθηκε

```
HTTP/1.1 200 OK
Date: Tue, 08 Sep 2020 00:53:20 GMT
Server: Apache/2.4.6 (CentOS)
       OpenSSL/1.0.2k-fips PHP/7.4.9
       mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 01 Mar 2016 18:57:50 GMT
ETag: "a5b-52d015789ee9e"
Accept-Ranges: bytes
Content-Length: 2651
Content-Type: text/html; charset=UTF-8
\r\n
data data data data data ...
```


Κωδικοί κατάστασης απόκρισης HTTP

- Κωδικοί κατάστασης εμφανίζονται στην 1η γραμμή στο μήνυμα απόκρισης εξυπηρέτη προς πελάτη
- Μερικοί κωδικοί κατάστασης:
 - 200 OK**
 - Επιτυχές αίτημα, το αντικείμενο εμφανίζεται παρακάτω σε αυτό το μήνυμα
 - 301 Moved Permanently**
 - Το αιτούμενο αντικείμενο έχει μετακινηθεί, η νέα θέση αναφέρεται παρακάτω σε αυτό το μήνυμα (στο πεδίο: θέση)
 - 400 Κακό Αίτημα**
 - Το αίτημα δεν είναι κατανοητό από τον εξυπηρέτη
 - 404 Not Found**
 - Το αιτούμενο αρχείο δεν βρίσκεται σε αυτόν τον εξυπηρέτη
 - 505 HTTP έκδοση δεν υποστηρίζεται**

Διατηρώντας κατάσταση χρήστη/εξυπηρέτη: cookies

Σημείωση: HTTP GET/response αλληλεπίδραση είναι **ακαταστατική (stateless)**

- Δεν γίνονται ανταλλαγές HTTP μηνυμάτων πολλών-βημάτων για να ολοκληρωθεί «συναλλαγή» στο Web
 - Δεν χρειάζεται να κρατείται η «κατάσταση» πολυβηματικής ανταλλαγής από τον πελάτη/εξυπηρέτη
 - Τα HTTP αιτήματα είναι ανεξάρτητα από άλλα τέτοια αιτήματα
 - Δεν χρειάζεται να «ανανήψει» ο πελάτης / εξυπηρέτης από μια συναλλαγή που εκτελέστηκε μερικώς αλλά ποτέ δεν ολοκληρώθηκε.

Διατηρώντας κατάσταση χρήστη/εξυπηρετή: cookies

Cookies χρησιμοποιούνται από ιστοσελίδες και browsers χρηστών για να διατηρούν κάποια κατάσταση συναλλαγών.

4 σχετικά σημεία / συστατικά:

- Γραμμή cookie στην κεφαλίδα μηνύματος απόκρισης εξυπηρετή
- Γραμμή cookie στην κεφαλίδα της επόμενης HTTP αίτησης χρήστη
- Ο browser χρήστη κρατά αρχείο με cookies στον υπολογιστή χρήστη.
- Η ιστοσελίδα (εξυπηρετής) κρατά ιστορικό (βάση δεδομένων) συναλλαγών με cookies.

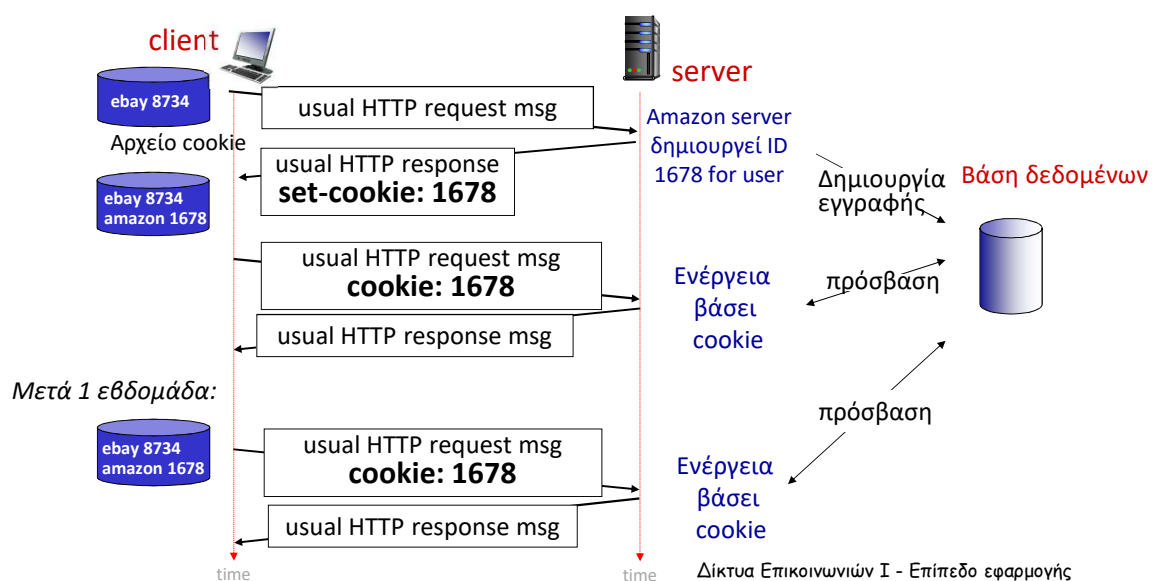
Παράδειγμα:

- Η Δάφνη επισκέπτεται για πρώτη φορά ιστότοπο για διαδικτυακή αγορά, με χρήση του browser του laptop της
- Με την άφιξη του πρώτου HTTP αιτήματος ο ιστότοπος δημιουργεί:
 - μοναδικό ID (δηλ "cookie")
 - εγγραφή για αυτό το ID στη βάση δεδομένων του
- Επακόλουθα HTTP αιτήματα από τη Δάφνη προς αυτό τον ιστότοπο θα αναφέρει την τιμή του ID του cookie, βοηθώντας τον να «αναγνωρίσει» την Δάφνη

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

35

Διατηρώντας κατάσταση χρήστη/εξυπηρετή: cookies



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

36

HTTP cookies: μερικά σχόλια

Χρήσεις cookies:

- authorization
- shopping carts
- recommendations
- user session state (Web e-mail)

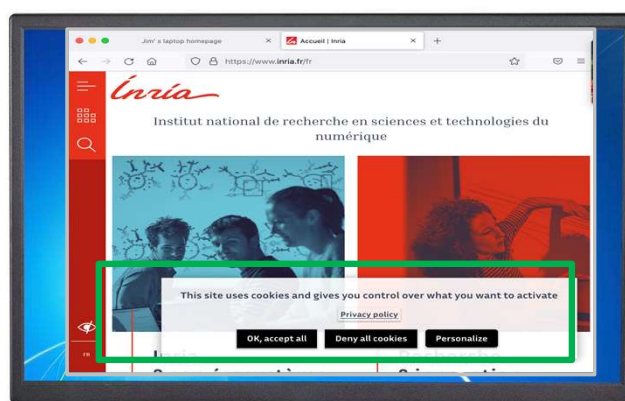
Πως «κρατείται» η κατάσταση?

- Στα άκρα που υλοποιούν το πρωτόκολλο: καταγράφεται η κατάσταση στον αποστολέα/ παραλήπτη κατά την εκτέλεση πολλαπλών συναλλαγών
- Στα μηνύματα: cookies στα HTTP μηνύματα εμπεριέχουν πληροφορία για την κατάσταση

HTTP cookies: μερικά σχόλια

_____ παρενθετικά cookies και Ιδιωτικότητα:

- Τα cookies δίνουν την δυνατότητα σε ιστοτόπους να μάθουν πολλά για εσάς.
- Cookies τρίτων, επιτρέπουν να παρακολουθείται ένας χρήστης σε πολλούς ιστότοπους.
- Συναίνεση απαιτείται για cookies μιας και θεωρούνται προσωπικά δεδομένα (GDPR – General Data Protection Regulation).



Ο χρήστης έχει απόλυτο έλεγχο στο αν επιτρέψει ή όχι cookies

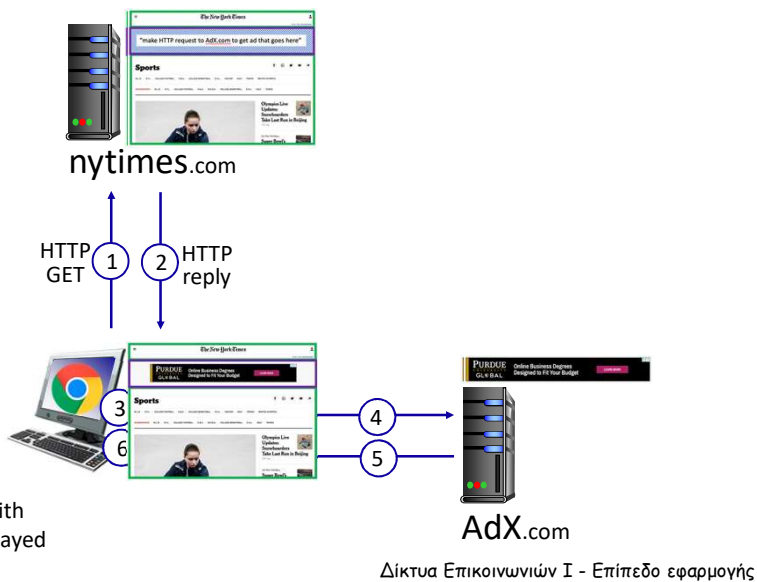
Παράδειγμα: προβολή της ιστοσελίδας των NY Times

1 GET base html file
2 from nytimes.com

4 fetch ad from
5 AdX.com

7 display composed
page

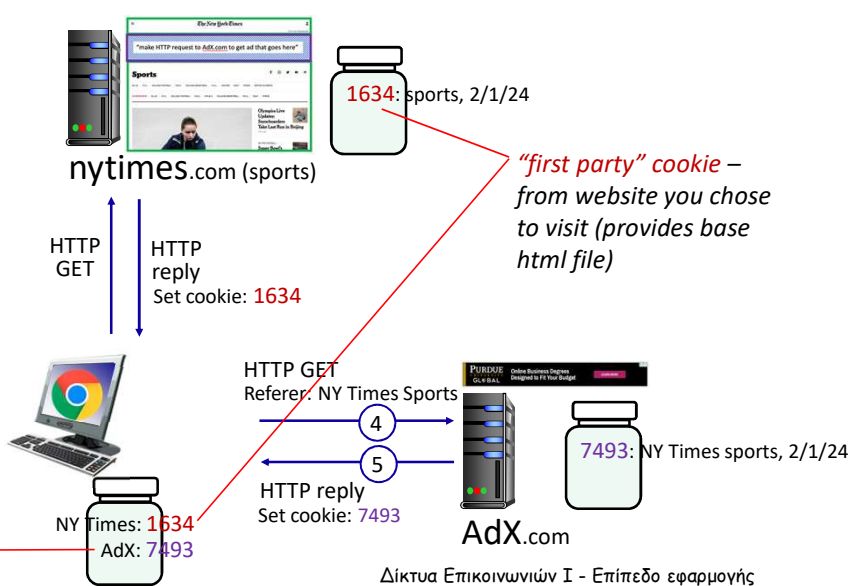
NY times page with
embedded ad displayed



39

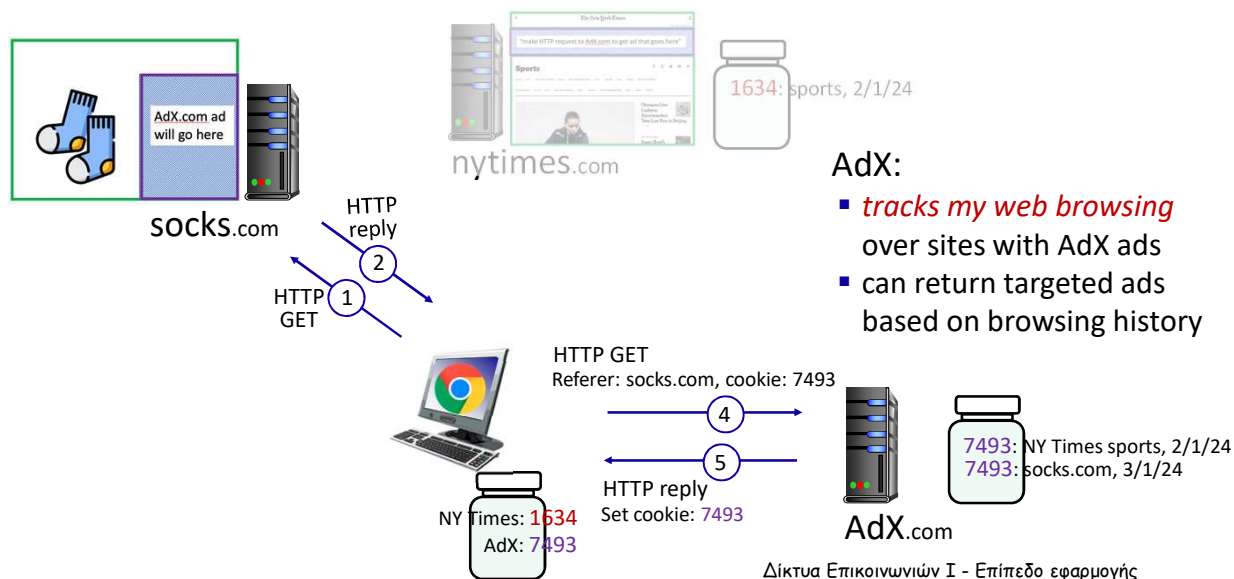
Cookies: παρακολουθώντας την πλοήγηση κάποιου στο web

*"third party" cookie –
from website you did not
choose to visit*



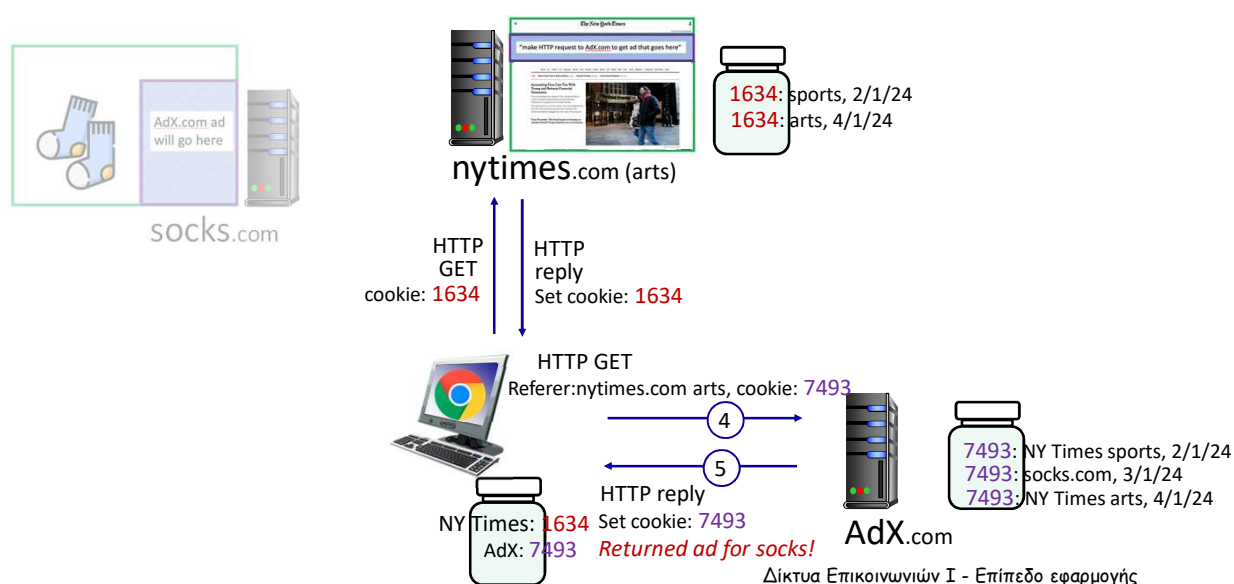
40

Cookies: παρακολουθώντας την πλοήγηση κάποιου στο web



41

Cookies: παρακολουθώντας την πλοήγηση κάποιου στο web (1 ημέρα μετά)



42

Cookies: παρακολούθηση συμπεριφοράς περιήγησης χρηστών

Τα Cookies μπορούν να χρησιμοποιηθούν για:

- να παρακολουθούν τη συμπεριφορά χρηστών σε συγκεκριμένο ιστότοπο (**first party cookies**)
- να παρακολουθούν τη συμπεριφορά χρηστών σε πολλαπλούς ιστότοπους (**third party cookies**), χωρίς καν να τους έχει επισκεφθεί ο χρήστης (!)
- η παρακολούθηση δεν είναι αντιληπτή στο χρήστη:
 - ❖ Πέρα από την παραπομπή σε tracker από κάποια διαφήμιση μέσω αιτήματος HTTP GET, μπορεί να είναι κάποιος άορατος σύνδεσμος

Τα cookies παρακολούθησης τρίτων:

- απενεργοποιούνται (by default) σε Firefox, Safari browsers
- Προβλέπεται να απενεργοποιηθούν σε Chrome browser (2023)

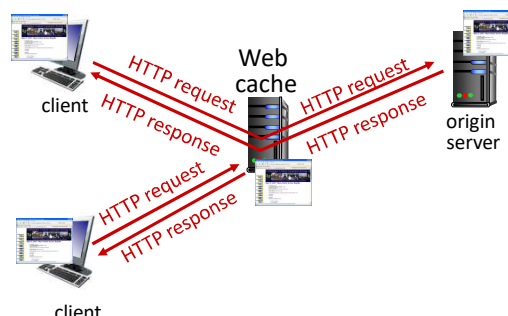
Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

43

Web caches (proxy server)

Goal: να εξυπηρετούνται κάποιες αιτήσεις πελατών χωρίς εμπλοκή του εξυπηρετή προέλευσης

- Ο χρήστης παραμετροποιεί τον browser για πρόσβαση στο Web μέσω της (τοπικής) cache
- Ο browser στέλνει όλες τις αιτήσεις HTTP στην cache
 - Το αντικείμενο στην cache: η cache επιστρέφει το αντικείμενο
 - Διαφορετικά η cache ζητά το αντικείμενο από τον εξυπηρετή προέλευσης, μετά επιστρέφει το αντικείμενο στον πελάτη



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

44

Περισσότερα για το Web caching

- Η cache δρα τόσο ως πελάτης όσο και ως εξυπηρετής
 - εξυπηρετής για τον αρχικό αιτούντα πελάτη
 - πελάτης για τον εξυπηρετή προέλευσης
- Ο εξυπηρετής προέλευσης ενημερώνει την cache για το αν επιτρέπεται η αποθήκευση στην cache κάποιου αντικειμένου, στην κεφαλίδα μηνύματος απόκρισης

```
Cache-Control: max-age=<seconds>
```

```
Cache-Control: no-cache
```

Γιατί Web caching ;

- Μείωση του χρόνου απόκρισης των αιτήσεων πελατών
 - Η cache είναι πιο κοντά στον πελάτη
- Μείωση της κίνησης στη ζεύξη πρόσβασης ενός οργανισμού
- Υπάρχουν πολλές caches στο Διαδίκτυο
 - επιτρέπει σε «φτωχούς» πάροχους περιεχομένου να παραδίδουν αποτελεσματικά το περιεχόμενό τους

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

45

Παράδειγμα Caching

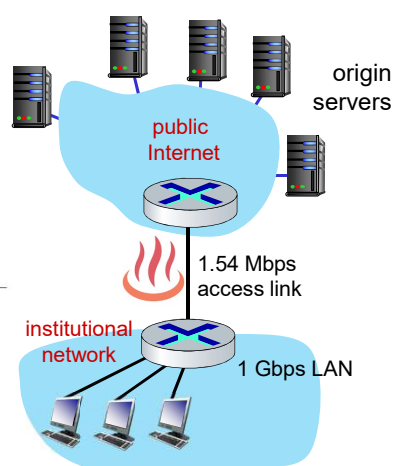
Υποθέσεις:

- Ρυθμός ζεύξης πρόσβασης: 1.54 Mbps
- RTT από το δρομολογητή του ιδρύματος προς κάθε εξυπηρετή προέλευσης και πίσω στο δρομολογητή = 2 sec
- Μέσο μέγεθος αντικειμένου : 100K bits
- Μέσος ρυθμός αιτήσεων από τους browsers προς τους εξυπηρετές προέλευσης = 15/sec
 - Μέσος ρυθμός δεδομένων προς τους browsers: 1.50 Mbps

Απόδοση:

- Αξιοποίηση της ζεύξης πρόσβασης = .97
- Αξιοποίηση του LAN : .0015
- Συνολική καθυστέρηση = καθυστέρηση Διαδικτύου + καθυστέρηση πρόσβασης + καθυστέρηση LAN = 2 sec + minutes + msecs

πρόβλημα!
Μεγάλες ουρές
σε υψηλή
αξιοποίηση



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

46

Επιλογή 1: αγορά ταχύτερης ζεύξης πρόσβασης

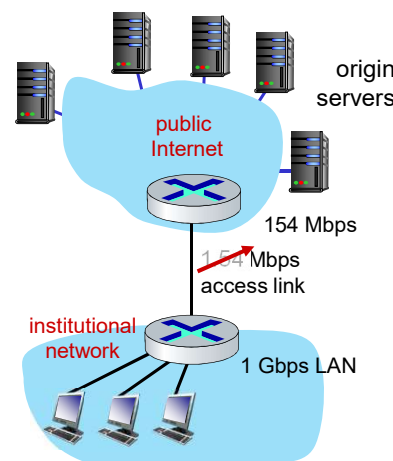
Υποθέσεις:

- Ρυθμός ζεύξης πρόσβασης: ~~1.54 Mbps~~ → 154 Mbps
- RTT από το δρομολογητή του ιδρύματος προς κάθε εξυπηρετή προέλευσης και πίσω στο δρομολογητή = 2 sec
- Μέσο μέγεθος αντικειμένου = 100 Kbits
- Μέσος ρυθμός αιτήσεων από τους browsers προς τους εξυπηρετές προέλευσης = 15/sec
 - Μέσος ρυθμός δεδομένων προς τους browsers: 1.50 Mbps

Performance:

- Αξιοποίηση της ζεύξης πρόσβασης = ~~.97~~ → .0097
- Αξιοποίηση του LAN : .0015
- Συνολική καθυστέρηση = καθυστέρηση Διαδικτύου + καθυστέρηση πρόσβασης + καθυστέρηση LAN
- = 2 sec + minutes + microseconds = 2 sec + ~~minutes~~ + usecs

Κόστος: ταχύτερη ζεύξη (ακριβότερη!)



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής 47

Επιλογή 2: εγκατάσταση τοπικής cache

Υποθέσεις:

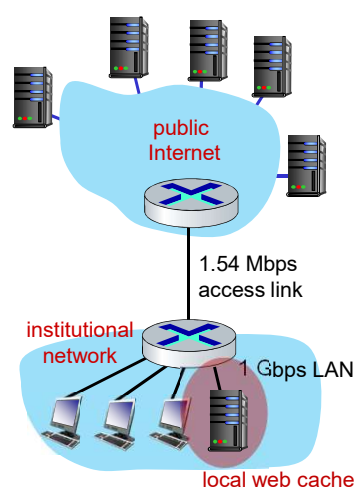
- Ρυθμός ζεύξης πρόσβασης: 1.54 Mbps
- RTT από το δρομολογητή του ιδρύματος προς κάθε εξυπηρετή προέλευσης και πίσω στο δρομολογητή = 2 sec
- Μέσο μέγεθος αντικειμένου = 100 Kbits
- Μέσος ρυθμός αιτήσεων από τους browsers προς τους εξυπηρετές προέλευσης = 15/sec
 - Μέσος ρυθμός δεδομένων προς τους browsers: 1.50 Mbps

Κόστος: web cache (φθηνή!)

Απόδοση:

- Αξιοποίηση του LAN?
- Αξιοποίηση της ζεύξης πρόσβασης?
- Συνολική καθυστέρηση ?

πώς υπολογίζεται η αξιοποίηση της ζεύξης και η καθυστέρηση;

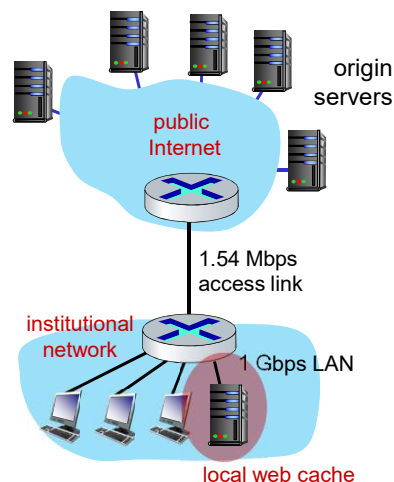


Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής 48

Υπολογισμός αξιοποίησης της ζεύξης πρόσβασης και καθυστέρησης με cache

Έστω ότι το ποσοστό επιτυχίας της cache είναι 0.4:

- 40% των αιτήσεων εξυπηρετούνται στην cache με ελάχιστη καθυστέρηση (msec)
- 60% των αιτήσεων εξυπηρετούνται από εξυπηρετή προέλευσης
 - ρυθμός δεδομένων προς browsers μέσω της ζεύξης πρόσβασης = $0,6 * 1,5 \text{ Mbps} = 0,9 \text{ Mbps}$
 - **Αξιοποίηση ζεύξης πρόσβασης = $0,9/1,54 = 0,58$ (μικρή msec) καθυστέρηση αναμονής ζεύξης πρόσβασης)**
- Μέση συνολική καθυστέρηση (end-to-end):
 - = $0.6 * (\text{καθυστέρηση από εξυπηρετές προέλευσης})$
 - + $0.4 * (\text{καθυστέρηση από εξυπηρετήρηση από cache})$
 - = $0.6 (2.01) + 0.4 (\sim \text{msecs}) = \sim 1.2 \text{ secs}$



Μικρότερη end-to-end καθυστέρηση σε σχέση με τη ζεύξη των 154 Mbps (και φθηνότερο!)

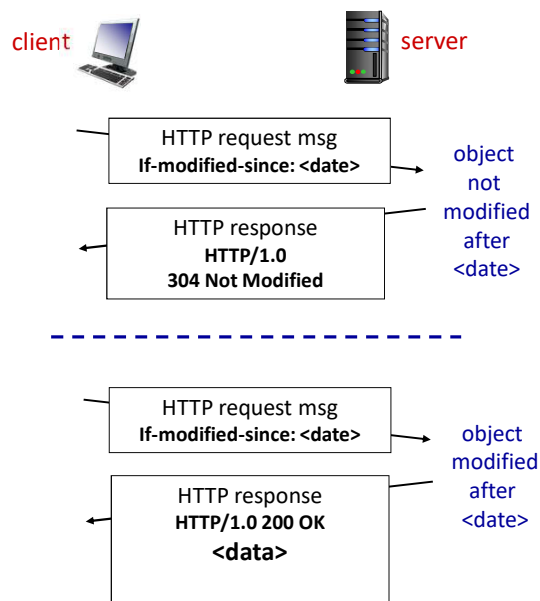
Δίκτυα Επικοινωνιών I - Επίπεδο εφαρμογής

49

Get υπό συνθήκη (Conditional GET)

Σκοπός: να μην στέλνεται το αντικείμενο αν η cache έχει ενήμερη (up-to-date) έκδοση

- χωρίς την καθυστέρηση μετάδοσης του αντικειμένου (χωρίς την άσκοπη χρήση δικτυακών πόρων)
- **πελάτης:** καθορίζει την ημερομηνία του αποθηκευμένου αντιγράφου στην αίτηση HTTP
If-modified-since: <date>
- **εξυπηρετής:** η απόκριση δεν περιέχει αντικείμενο αν το αντίγραφο στην cache δεν έχει τροποποιηθεί:
HTTP/1.0 304 Not Modified



Δίκτυα Επικοινωνιών I - Επίπεδο εφαρμογής

50

HTTP/2

Στόχος: Βελτίωση της καθυστέρησης σε αιτήσεις πολλαπλών αντικειμένων

HTTP1.1: Δημιουργεί **πολλαπλά, εν σειρά (pipelined) GETs** πάνω από μία TCP σύνδεση

- Ο εξυπηρέτης αποκρίνεται με τη σειρά των αιτημάτων GET (FCFS: first-come-first-served scheduling)
 - Με την πολιτική FCFS, μπορεί μικρά αντικείμενα να πρέπει να περιμένουν την μετάδοση τους πίσω από μεγάλα αντικείμενα (**head-of-line (HOL) blocking**)
- Η ανάνηψη από απώλειες (επαναμετάδοση χαμένων TCP τμημάτων) «παγώνει» την μετάδοση του αντικειμένου

HTTP/2

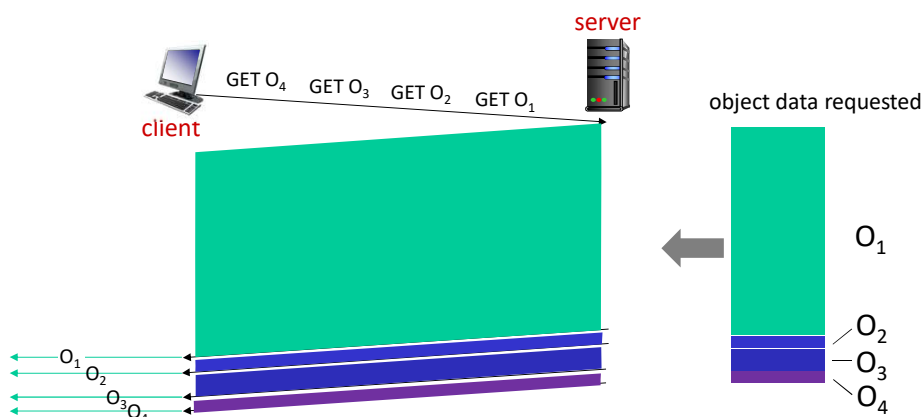
Στόχος: Βελτίωση της καθυστέρησης σε αιτήσεις πολλαπλών αντικειμένων

HTTP/2: [RFC 7540, 2015] καλύτερη ευελιξία στον εξυπηρέτη στην αποστολή αντικειμένων στον πελάτη:

- Τα βασικά παραμένουν ίδια όπως στο HTTP1.1 (methods, status codes, βασικές κεφαλίδες, ...)
- Η σειρά αποστολής των αντικειμένων καθορίζεται από τον πελάτη (όχι υποχρεωτικά First Come First Serve (FCFS))
- *push* για κάποια αντικείμενα που δεν έχουν ζητηθεί
- Τα αντικείμενα διαιρούνται σε πλαίσια τα οποία χρονο-προγραμματίζονται ώστε να αποφεύγεται Head-of-Line (HOL) blocking

HTTP/2: αποτρέποντας HOL blocking

HTTP 1.1: πελάτης ζητά 1 μεγάλο αντικείμενο (π.χ., αρχείο video) και 3 μικρότερα αντικείμενα



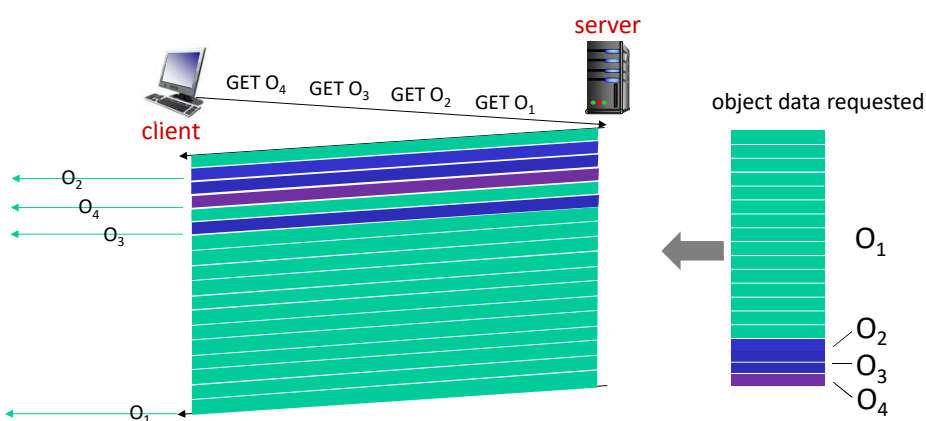
Τα αντικείμενα παραδίδονται με τη σειρά που ζητήθηκαν :
O₂, O₃, O₄ περιμένουν πίσω από O₁

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

53

HTTP/2: αποτρέποντας HOL blocking

HTTP/2: αντικείμενα τεμαχίζονται σε πλαίσια, η μετάδοση πλαισίων γίνεται όχι σαν ενιαίο μπλοκ (interleaved)



O₂, O₃, O₄ παραδίδονται γρήγορα, O₁ παραδίδονται
γρήγορα καθυστερεί λίγο παραπάνω

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

54

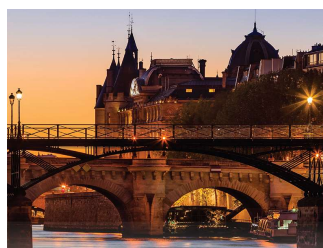
Από το HTTP/2 στο HTTP/3

HTTP/2 πάνω από απλή (μία) TCP σύνδεση σημαίνει:

- Η ανάληψη από απώλειες πακέτων εξακολουθεί να «παγώνει» (φρενάρει) όλες τις μεταδόσεις αντικειμένων
 - όπως στο HTTP 1.1, οι browsers έχουν κίνητρο να ανοίξουν πολλαπλές συνδέσεις TCP για να μειώσουν το «φρενάρισμα» και να αυξήσουν γενικότερα τη ρυθμαπόδοση
- Δεν υπάρχει ασφάλεια πάνω από την TCP σύνδεση
- **HTTP/3**: προσθέτει ασφάλεια, έλεγχο σφαλμάτων και συμφόρησης ανά αντικείμενο (περισσότερο pipelining) πάνω στο UDP
 - Περισσότερα για το HTTP/3 στο επίπεδο μεταφοράς

Επίπεδο Εφαρμογής: επισκόπηση

- Αρχές δικτυακών εφαρμογών
- Web και HTTP
- E-mail, SMTP, IMAP
- Domain Name System DNS
- Εφαρμογές P2P
- video streaming και δίκτυα διανομής περιεχομένου
- socket programming with UDP and TCP



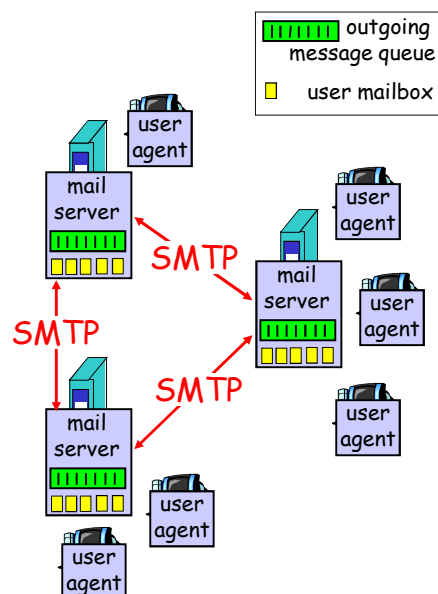
Ηλεκτρονικό Ταχυδρομείο

Τρία κύρια συστατικά μέρη:

- ❑ Πράκτορες χρήστη (user agents)
- ❑ Εξυπηρέτες ταχυδρομείου (mail servers)
- ❑ simple mail transfer protocol: SMTP

Πράκτορας Χρήστη (User Agent)

- ❑ Δηλ. "mail reader" (αναγνώστης ταχυδρομείου)
- ❑ Σύνθεση, επεξεργασία, ανάγνωση μηνυμάτων αλληλογραφίας
- ❑ π.χ., Outlook, Mozilla Thunderbird, iPhone mail client
- ❑ Εξερχόμενα, εισερχόμενα μηνύματα αποθηκευμένα στον εξυπηρέτη



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

57

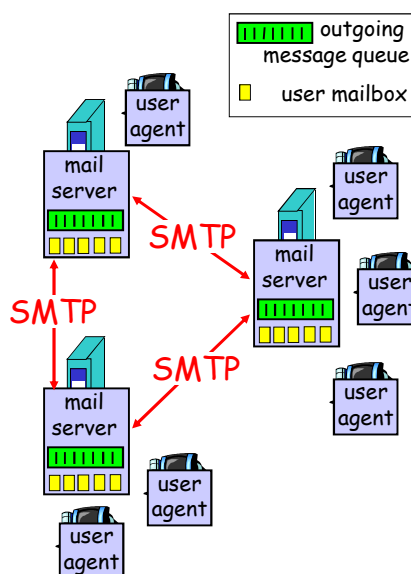
Ηλεκτρονικό ταχυδρομείο: εξυπηρέτες ταχυδρομείου (mail servers)

Εξυπηρέτες Ταχυδρομείου
(Mail Servers)

- Η ταχυδρομική θυρίδα (mailbox) περιέχει τα εισερχόμενα μηνύματα του χρήστη
- Ουρά εξερχομένων μηνυμάτων αλληλογραφίας που πρέπει να σταλούν

Πρωτόκολλο SMTP μεταξύ εξυπηρετών ταχυδρομείου για την αποστολή email

- πελάτης : εξυπηρέτης ταχυδρομείου αποστολέα
- «εξυπηρέτης»: εξυπηρέτης ταχυδρομείου παραλήπτης

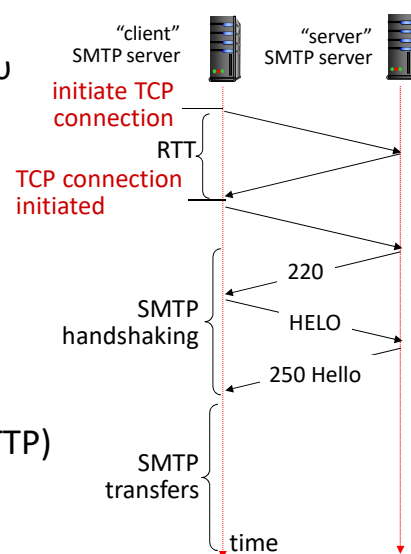


Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

58

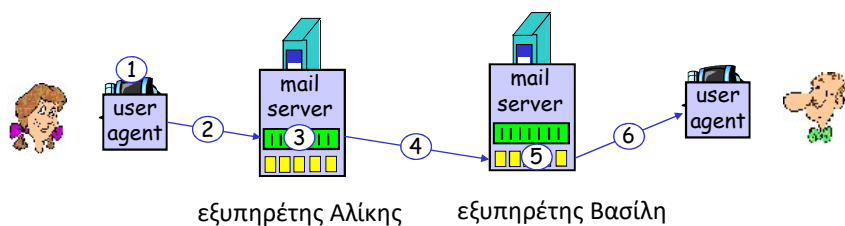
Ηλεκτρονικό Ταχυδρομείο: SMTP [RFC 5321]

- ❑ Χρησιμοποιεί TCP για την αξιόπιστη μεταφορά μηνυμάτων από τον πελάτη (εξυπηρετή email που ξεκινά τη σύνδεση) στον εξυπηρετή, θύρα 25
 - Απευθείας μεταφορά: από τον εξυπηρετή αποστολέα (πελάτη) προς τον εξυπηρετή παραλήπτη
- ❑ Τρεις φάσεις της μεταφοράς
 - χειραψία (χαιρετισμός)
 - μεταφορά μηνυμάτων
 - τερματισμός
- ❑ Αλληλεπίδραση εντολής / απόκρισης (όπως το HTTP)
 - **εντολές:** κείμενο ASCII
 - **απόκριση:** κωδικός κατάστασης και φράση

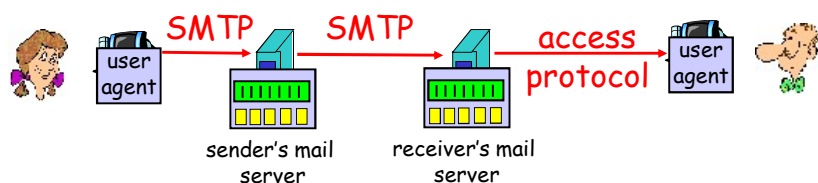


Σενάριο: Η Αλίκη στέλνει μήνυμα στον Βασίλη

- 1) Η Αλίκη χρησιμοποιεί πράκτορα χρήστη (user agent-UA) για τη σύνθεση του μηνύματος "προς" vassilis@di.uoa.gr
- 2) Ο πράκτορας της Αλίκης στέλνει το μήνυμα στον εξυπηρετή ταχυδρομείου της, το μήνυμα τοποθετείται στην ουρά μηνυμάτων
- 3) Η πλευρά του πελάτη του SMTP ανοίγει TCP σύνδεση με τον εξυπηρετή ταχυδρομείου του Βασίλη
- 4) Ο πελάτης SMTP στέλνει το μήνυμα της Αλίκης πάνω από τη σύνδεση TCP
- 5) Ο εξυπηρετής ταχυδρομείου του Βασίλη τοποθετεί το μήνυμα στην ταχυδρομική θυρίδα του Βασίλη
- 6) Ο Βασίλης χρησιμοποιεί το δικό του πράκτορα για να το διαβάσει



Λήψη email: Πρωτόκολλα προσπέλασης ταχυδρομείου



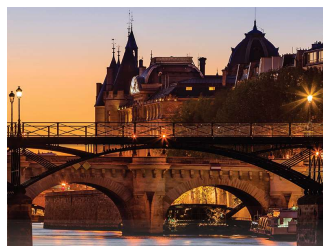
- SMTP: παράδοση/αποθήκευση email στον εξυπηρέτη του παραλήπτη
- Πρωτόκολλο προσπέλασης ταχυδρομείου (mail access protocol): ανάκτηση από τον εξυπηρέτη
 - **IMAP**: Internet Mail Access Protocol [RFC 3501]: μηνύματα αποθηκεύονται στον εξυπηρέτη, IMAP παρέχει δυνατότητα ανάκτησης, διαγραφής αποθηκευμένων μηνυμάτων στον εξυπηρέτη
 - **POP**: Post Office Protocol [RFC 1939]: εξουσιοδότηση και «κατέβασμα» (“download”)
- **HTTP**: Gmail, Hotmail, Yahoo! Mail, κτλ. παρέχουν μια web διεπαφή πάνω από το SMTP (για αποστολή), IMAP(POP) (για ανάκτηση) μηνυμάτων

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

61

Επίπεδο Εφαρμογής: επισκόπηση

- Αρχές δικτυακών εφαρμογών
- Web και HTTP
- E-mail, SMTP, IMAP
- Domain Name System DNS
- Εφαρμογές P2P
- video streaming και δίκτυα διανομής περιεχομένου
- socket programming with UDP and TCP



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

62

DNS: Domain Name System (Σύστημα Ονομασίας Τομέων)

Άνθρωποι: Πολλά αναγνωριστικά (ταυτότητα, διαβατήριο, ΑΜΚΑ, ΑΦΜ..)

Δικτυωμένοι Υπολογιστές / Δρομολογητές:

- IP διευθύνσεις (32bits) – για δρομολόγηση δεδομένων
- «ονόματα», π.χ. di.uoa.gr – που χρησιμοποιούν άνθρωποι

Ερώτηση: Πως θα απεικονίσουμε ονόματα σε IP διευθύνσεις και το αντίστροφο?

Domain Name System (DNS):

- ❑ **Κατανεμημένη βάση**, υλοποιούμενη από ιεραρχία πολλών *εξυπηρετών ονόματος*
- ❑ **Πρωτόκολλο επιπέδου εφαρμογής:** Υπολογιστές, DNS εξυπηρετές επικοινωνούν για επίλυση ονομάτων (μετάφραση διευθύνσεων/ονομάτων)
 - ❖ Σημείωση: είναι βασική λειτουργία του κορμού δικτύου, που **υλοποιείται σαν πρωτόκολλο επιπέδου εφαρμογής**.
 - ❖ Πολυπλοκότητα στο «άκρο» του δικτύου.

DNS: Υπηρεσίες, δομή

Υπηρεσίες DNS

- ❑ Μετάφραση ονόματος (hostname) σε διεύθυνση IP
- ❑ Ψευδώνυμα υπολογιστών (host aliasing) -ευκολομημόνευτα
 - Ψευδώνυμα → Κανονικά ονόματα
- ❑ Ψευδώνυμα εξυπηρετών ταχυδρομείου (mail server aliasing)
- ❑ Κατανομή φορτίου
 - Εξυπηρετές-αντίγραφα Web: πολλές διευθύνσεις IP αντιστοιχούν σε ένα όνομα

Γιατί όχι κεντρικοποιημένο;

- ❑ μοναδικό σημείο αποτυχίας
- ❑ όγκος κίνησης
- ❑ απομακρυσμένη κεντρικοποιημένη βάση δεδομένων
- ❑ συντήρηση

A: **δεν κλιμακώνει!**

- Comcast DNS εξυπηρετές μόνο: 600B DNS queries/day
- Akamai DNS εξυπηρετές μόνο: 2.2T DNS queries/day

Παρατηρήσεις για το DNS

Τεράστια κατανεμημένη βάση:

- ~ δισεκατομμύρια εγγραφές

Διαχειρίζεται τρισεκατομμύρια queries/ημέρα:

- Πολύ περισσότερα reads από writes
- Η απόδοση έχει τεράστια σημασία: σχεδόν κάθε διαδικτυακή συναλλαγή αλληλεπιδρά με το DNS – σημασία στα msecs!

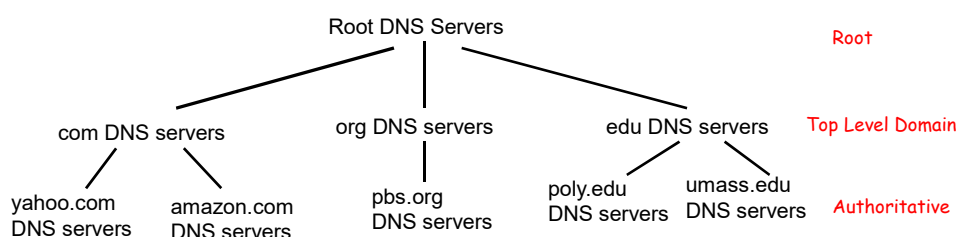
Οργανωτικά και φυσικά κατανεμημένη:

- Εκατομμύρια διαφορετικών οργανισμών φροντίζουν για τις εγγραφές τους

“αδιάτρητη”: αξιόπιστη, ασφαλής



Κατανεμημένη, Ιεραρχική Βάση Δεδομένων



Ο πελάτης θέλει τη διεύθυνση IP για το www.amazon.com (μια πρώτη προσέγγιση):

- Ο πελάτης ρωτά έναν εξυπηρετή ρίζας (root server) για να βρει τον εξυπηρετή DNS com (com DNS server)
- Ο πελάτης ρωτά τον εξυπηρετή DNS com για να πάρει τον εξυπηρετή DNS amazon.com (amazon.com DNS server)
- Ο πελάτης ρωτά τον εξυπηρετή DNS amazon.com (amazon.com DNS server) για να πάρει τη διεύθυνση IP του www.amazon.com

DNS: Εξυπηρέτες ονομάτων ρίζας (Root name servers)

- έρχεται σε επαφή με αυτόν ένας εξυπηρέτης ονομάτων που δεν μπορεί να μεταφράσει το όνομα με άλλο τρόπο.
 - παίρνει την αντιστοιχία & την επιστρέφει στον τοπικό εξυπηρέτη ονομάτων
 - Ή απλά παραπέμπει σε άλλον εξυπηρέτη
- **Πολύ σημαντική** λειτουργία του διαδικτύου:
 - Το Διαδίκτυο δεν θα λειτουργούσε χωρίς αυτήν
 - DNSSEC – παρέχει ασφάλεια (αυθεντικοποίηση ακεραιότητα μηνύματος)
- ICANN (Internet Corporation for Assigned Names and Numbers) διαχειρίζεται το root DNS domain

13 εξυπηρέτες ονομάτων ρίζας παγκοσμίως, με πολλαπλά αντίγραφα καθενός σε διάφορα μέρη (~1000 συνολικά).



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

67

Top Level Domain (TLD) και αυθεντικοί εξυπηρέτες

- Εξυπηρέτες τομέων ανώτατου επιπέδου (**Top-level domain (TLD) servers**):
 - ❖ Υπεύθυνοι για com, org, net, edu, aero, jobs, museums και όλους τους ανώτατου επιπέδου τομείς χωρών, π.χ. uk, fr, ca, jp
 - ❖ Η Network Solutions διατηρεί εξυπηρέτες για τους .com, .net TLD
 - ❖ Η Educause για τους .edu TLD
- Αυθεντικοί εξυπηρέτες DNS (**Authoritative DNS servers**):
 - ❖ Οι εξυπηρέτες DNS ενός οργανισμού που παρέχουν αυθεντικές αντιστοιχίσεις ονομάτων υπολογιστών σε διευθύνσεις IP για τους εξυπηρέτες του οργανισμού (π.χ., Web, mail)
 - ❖ Μπορεί να διατηρείται από τον οργανισμό ή πάροχο υπηρεσιών

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

68

Τοπικός Εξυπηρέτης Ονομάτων (Local DNS Name Server)

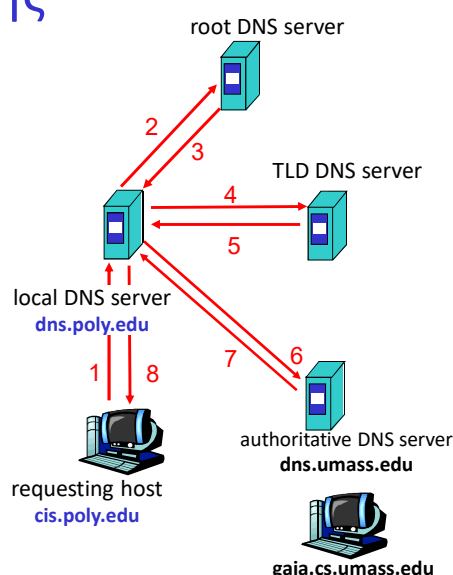
- ❑ όταν ένας υπολογιστής πραγματοποιεί ένα ερώτημα DNS, το ερώτημα στέλνεται στον τοπικό εξυπηρέτη DNS
 - Ο τοπικός DNS απαντά:
 - Παρέχοντας την αιτούμενη μετάφραση όνομα-σε-διεύθυνση, από την τοπική cache των πρόσφατων ζευγαριών μετάφρασης (μπορεί να μην είναι ενημερωμένη!)
 - Λειτουργώντας ως προoxy, προωθώντας το ερώτημα στην ιεραρχία
- ❑ κάθε ISP (περιφερειακός ISP, εταιρία, πανεπιστήμιο) έχει έναν. Για να τον βρείτε:
 - MacOS: % scutil --dns
 - Windows: >ipconfig /all
- ❑ Ο τοπικός DNS δεν ανήκει αυστηρά στην ιεραρχία

Παράδειγμα μετάφρασης ονόματος DNS

Παράδειγμα: Ο υπολογιστής στο cis.poly.edu θέλει τη διεύθυνση IP του gaia.cs.umass.edu

Επαναληπτικό ερώτημα (Iterated query):

- Ο εξυπηρέτης που ρωτήθηκε απαντά με το όνομα του εξυπηρέτη που πρέπει να ερωτηθεί
- “Δεν ξέρω το όνομα αλλά ρώτα αυτόν τον εξυπηρέτη”

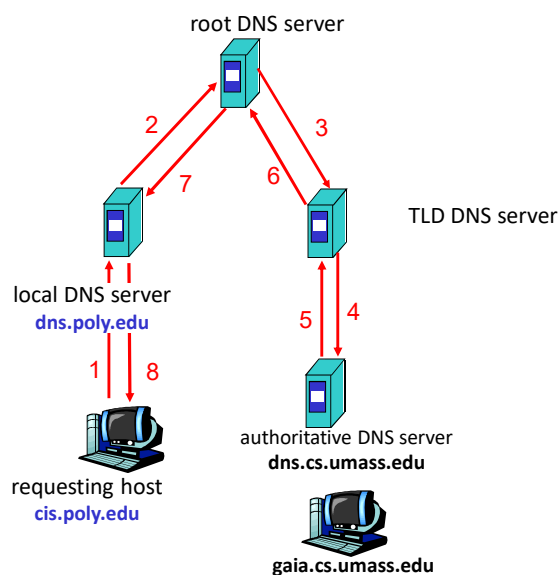


Παράδειγμα μετάφρασης ονόματος DNS

Παράδειγμα: Ο υπολογιστής στο cis.poly.edu θέλει τη διεύθυνση IP του gaia.cs.umass.edu

Αναδρομικό ερώτημα (Recursive query):

- Εναποθέτει το βάρος της μετάφρασης ονόματος στον εξυπηρέτη που ρωτήθηκε
- Βαρύ φορτίο στα ανώτερα επίπεδα της ιεραρχίας (χρησιμοποιείται πολύ λιγότερο από την προσέγγιση με επαναληπτικό ερώτημα)



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

71

Προσωρινή αποθήκευση (caching) DNS πληροφορίας και ενημέρωση εγγραφών

- Όταν ένας εξυπηρέτης ονομάτων μάθει μια αντιστοιχία, την **αποθηκεύει προσωρινά** και **αμέσως** επιστρέφει μια αποθηκευμένη αντιστοιχία όταν του ζητηθεί.
 - Caching βελτιώνει το χρόνο απόκρισης
 - Τα περιεχόμενα της προσωρινής μνήμης (cache) λήγουν (εξαλείφονται) μετά από κάποιο χρόνο (TTL)
 - Οι εξυπηρέτες TLD τυπικά αποθηκεύονται προσωρινά σε τοπικούς εξυπηρέτες ονομάτων
- Οι προσωρινά αποθηκευμένες εγγραφές μπορεί **να μην είναι ενημερωμένες**
 - ❖ αν ο name host αλλάξει IP διεύθυνση, μπορεί να μην γίνει γνωστό στο Διαδίκτυο μέχρι να λήξουν όλα τα TTLs
 - ❖ **βέλτιστης προσπάθειας μετάφραση ονόματος-σε-διεύθυνση!**

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

72

Εγγραφές DNS

DNS: κατακευμεμένη βάση που αποθηκεύει εγγραφές πόρων [resource records (RR)]

RR format: (name, value, type, ttl)

❑ Type=A (Address record)

- **name** είναι το όνομα του υπολογιστή
- **value** είναι η διεύθυνση IP

❑ Type=NS (Name Server record)

- **name** είναι τομέας (domain) (π.χ., foo.com)
- **value** είναι το όνομα υπολογιστή (hostname) του αυθεντικού εξυπηρετή ονομάτων για αυτόν τον τομέα

❑ Type=CNAME (aliasing record)

- **name** είναι ψευδώνυμο για κάποιο κανονικό (πραγματικό) όνομα
 - www.ibm.com είναι στην πραγματικότητα servereast.backup2.ibm.com
- **value** είναι το κανονικό όνομα

❑ Type=MX (for mail server record)

- **value** είναι το όνομα του εξυπηρετή mail που σχετίζεται με το **name**

Εισαγωγή εγγραφών στο DNS

- ❑ Παράδειγμα: νέα startup εταιρία "Network Utopia"
- ❑ Εγγράφει το όνομα networkutopia.com στον **DNS registrar** (π.χ., Network Solutions)
 - Παρέχει ονόματα, διευθύνσεις IP του αυθεντικού εξυπηρετή ονομάτων (κύριου και δευτερεύοντα)
 - Ο registrar εισάγει δύο RRs (NS και A) στον εξυπηρετή TLD com:

Domain	Όνομα υπολογιστή για αυτό το Domain
(networkutopia.com, dns1.networkutopia.com, NS)	
(dns1.networkutopia.com, 212.212.212.1, A)	
Όνομα υπολογιστή	IP υπολογιστή

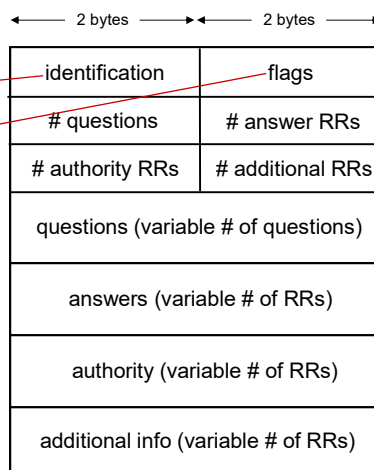
- ❑ Δημιουργεί τοπικά αυθεντικό εξυπηρετή με IP 212.212.212.1 με εγγραφές:
 - ❖ Type A για το www.networkutopia.com (web server),
 - ❖ Type MX για το networkutopia.com (mail server)

Μηνύματα DNS πρωτοκόλλου

Τα DNS *query* (αίτημα) και *reply* (απόκριση) μηνύματα έχουν το ίδιο *format*:

Κεφαλίδα μηνύματος:

- **identification:** 16 bit # για αίτημα, η απόκριση στο αίτημα χρησιμοποιεί το ίδιο #
- **flags:**
 - Αίτημα ή απόκριση
 - recursion είναι επιθυμητή
 - recursion είναι διαθέσιμη
 - Η απόκριση είναι authoritative



Μηνύματα DNS πρωτοκόλλου

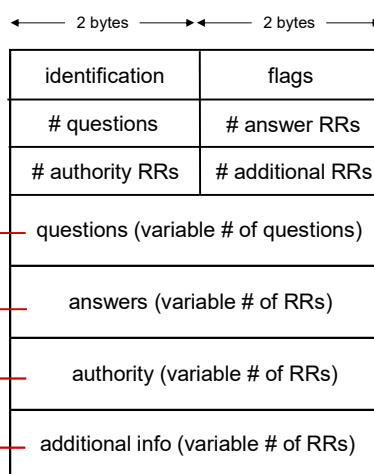
Τα DNS *query* (αίτημα) και *reply* (απόκριση) μηνύματα έχουν το ίδιο *format*:

name, type fields για ένα αίτημα

RRs στην απόκριση στο αίτημα

records για authoritative servers

Για τυχόν πρόσθετη «βοηθητική» πληροφορία



Επίπεδο Εφαρμογής: επισκόπηση

- Αρχές δικτυακών εφαρμογών
- Web και HTTP
- E-mail, SMTP, IMAP
- Domain Name System DNS
- Εφαρμογές P2P
- video streaming και δίκτυα διανομής περιεχομένου
- socket programming with UDP and TCP



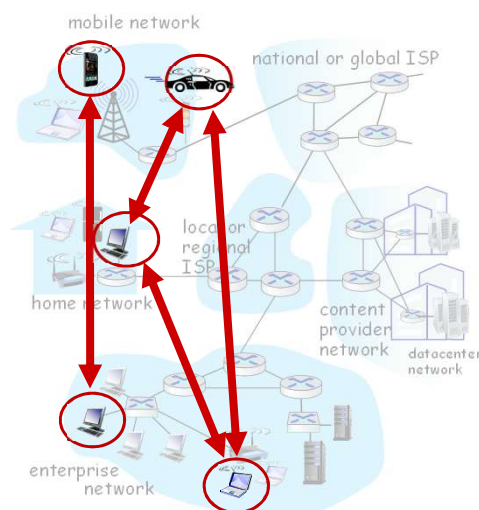
Δίκτυα Επικοινωνιών I - Επίπεδο εφαρμογής

77

Peer-to-peer (P2P) αρχιτεκτονική

- Δεν υπάρχει διαρκώς ενεργός εξυπηρέτης
- Τυχαία τερματικά συστήματα επικοινωνούν απευθείας
- Οι ομότιμοι ζητούν από - και παρέχουν σε άλλους - ομότιμους υπηρεσίες
αυτοκλιμάκωση – νέοι ομότιμοι αυξάνουν το δυναμικό παροχής υπηρεσιών και τη ζήτηση για υπηρεσίες
- Οι ομότιμοι είναι συνδεδεμένοι διακοπτόμενα και αλλάζουν διευθύνσεις IP

Παραδείγματα: P2P Διανομή αρχείου (BitTorrent), Μετάδοση (streaming) (KanKan), VoIP(Skype)



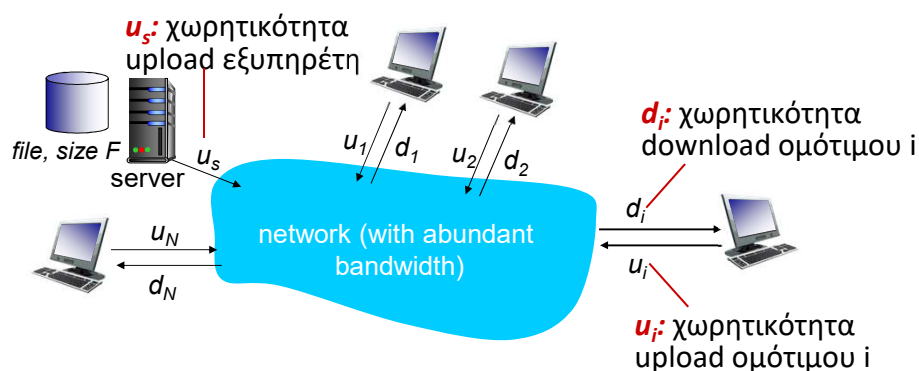
Δίκτυα Επικοινωνιών I - Επίπεδο εφαρμογής

78

Διανομή αρχείου: Server-Client vs P2P

Ερώτηση : Πόσος χρόνος απαιτείται για τη διανομή αρχείου (μεγέθους F) από έναν εξυπηρέτη σε N ομότιμους;

- η upload/download χωρητικότητα του ομότιμου είναι περιορισμένος πόρος

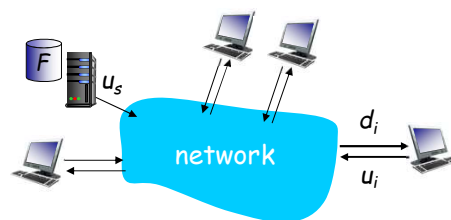


Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

79

Χρόνος διανομής αρχείου: Πελάτης - εξυπηρέτης (client-server)

- Ο **εξυπηρέτης** στέλνει (upload) σειριακά N αντίγραφα:
 - Χρόνος αποστολής ενός αντίγραφου: F/u_s
 - Χρόνος αποστολής N αντίγραφων: NF/u_s
- Κάθε **πελάτης** πρέπει να κατεβάσει (download) το αντίγραφο
 - d_{\min} = ο ρυθμός download του πελάτη με τον ελάχιστο ρυθμό.
 - χρόνος download πελάτη με τον ελάχιστο ρυθμό: F/d_{\min}



Χρόνος για να διανεμηθεί το F σε N πελάτες χρησιμοποιώντας την προσέγγιση client/server

$$D_{c-s} \geq \max \{ NF/u_s, F/d_{\min} \}$$

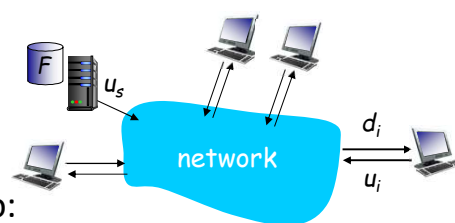
Αυξάνει γραμμικά με το N

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

80

Χρόνος διανομής αρχείου: P2P

- ❑ Ο **εξυπηρετής** πρέπει να ανεβάσει τουλάχιστον ένα αντίγραφο:
 - ❖ χρόνος F/u_s
- ❑ Κάθε **πελάτης** πρέπει να κατεβάσει το αντίγραφο:
 - ❖ ελάχιστος χρόνος κατεβάσματος για τον πελάτη F/d_i
- ❑ Οι πελάτες πρέπει να κατεβάσουν αθροιστικά NF bits
 - ❖ Μέγιστος συνολικός ρυθμός ανεβάσματος (που περιορίζει το μέγιστο ρυθμό κατεβάσματος) είναι: $u_t = u_s + \sum u_i$



χρόνος διανομής του F
σε N πελάτες με την
P2P προσέγγιση

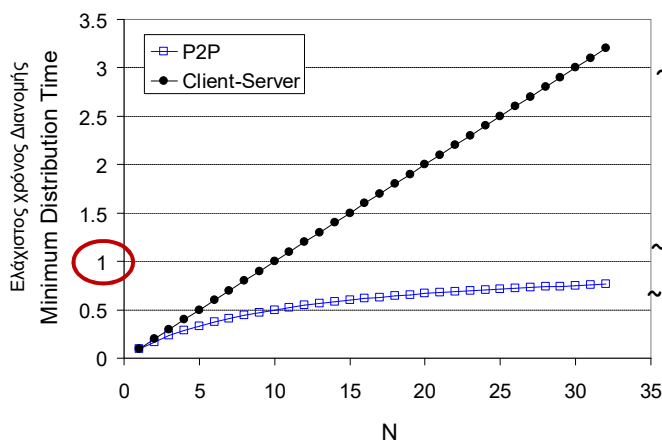
$$D_{P2P} \geq \max\{F/u_s, F/d_{min}, NF/(u_s + \sum u_i)\}$$

αυξάνει γραμμικά με το N

ομοίως και αυτό, καθώς κάθε ομότιμος φέρει ικανότητα παροχής υπηρεσιών

Server-client vs. P2P: Παράδειγμα

Ρυθμός upload πελάτη = u , $F/u = 1$ ώρα, $u_s = 10u$, $d_{min} \geq u$



$\sim N/10$

$$(D_{c-s} \geq \max\{NF/u_s, F/d_{min}\})$$

$$\sim N/(10+N) = (10+N-10)/(10+N) = 1-N/(10+N) \xrightarrow{N \rightarrow \infty} 1$$

$$\sim N/(10+N) \leftarrow \leftarrow \leftarrow NF/(10u+Nu)$$

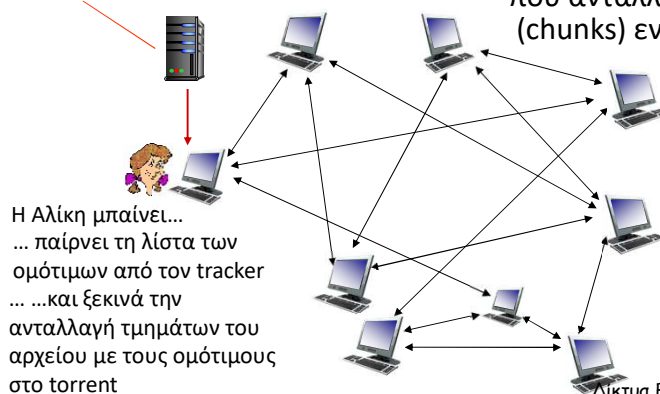
$$(D_{P2P} \geq \max\{F/u_s, F/d_{min}, NF/(u_s + \sum u_i)\})$$

P2P file distribution: BitTorrent

- το αρχείο διαιρείται σε τμήματα (chunks) των 256Kb
- οι ομότιμοι στο torrent στέλνουν/λαμβάνουν τμήματα ενός αρχείου

tracker: παρακολουθεί τους ομότιμους που μετέχουν στο torrent

torrent: ομάδα ομότιμων που ανταλλάσσουν τμήματα (chunks) ενός αρχείου

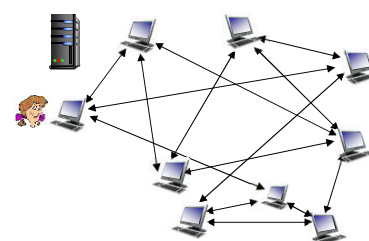


Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

83

BitTorrent

- Ομότιμος που συνδέεται στο torrent:
 - Δεν έχει chunks, αλλά θα τα συσσωρεύσει με την πάροδο του χρόνου από άλλους ομότιμους
 - Εγγράφεται στον tracker για να πάρει τη λίστα των ομότιμων, συνδέεται σε υποσύνολο των ομότιμων («γείτονες»("neighbors"))



- Ενώ κατεβάζει, ο ομότιμος ανεβάζει chunks σε άλλους ομότιμους
- Ο ομότιμος μπορεί να αλλάξει τους ομότιμους με τους οποίους ανταλλάσσει chunks
- **churn** : Ομότιμοι ενδέχεται να έρχονται και να φεύγουν
- Όταν ο ομότιμος έχει ολόκληρο το αρχείο, ενδέχεται (εγωϊστικά) να φύγει ή (αλτροϊστικά) να παραμείνει στο torrent

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

84

BitTorrent: αίτηση, αποστολή chunks αρχείων

Ζητώντας Chunks

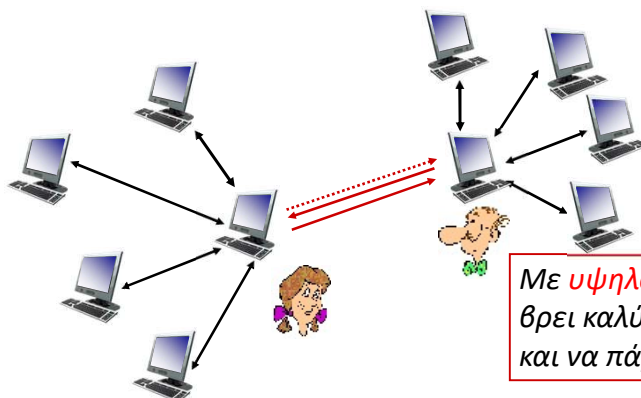
- ❑ Κάθε χρονική στιγμή, διαφορετικοί ομότιμοι έχουν διαφορετικό υποσύνολο chunks του αρχείου
- ❑ Περιοδικά, ένας ομότιμος (Αλίκη) ρωτά κάθε ομότιμο σχετικά με τη λίστα των chunks που έχει
- ❑ Η Αλίκη στέλνει αίτηση για τα chunks που της λείπουν, πρώτα το σπανιότερο (για να γίνουν λιγότερο σπάνια)

Στέλνοντας Chunks: tit-for-tat

- ❑ Η Αλίκη στέλνει chunks στους τέσσερις ομότιμους που στέλνουν στον υψηλότερο ρυθμό
 - Άλλοι ομότιμοι αγνοούνται από την Αλίκη (δε λαμβάνουν chunks από αυτή)
 - Επαναξιολογεί τους top 4 κάθε 10 secs
- ❑ κάθε 30 secs: διαλέγει τυχαία άλλον ομότιμο, ξεκινά να στέλνει chunks
 - Σταματάει να αγνοεί αυτόν τον ομότιμο (optimistically unchoke)
 - Ο νέος ομότιμος που επιλέχτηκε ενδέχεται να συμμετάσχει στους top 4

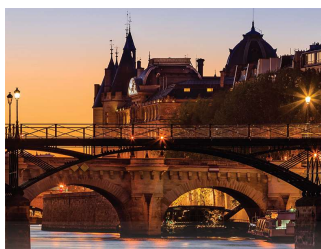
BitTorrent: tit-for-tat

- (1) Η Αλίκη "optimistically unchokes" τον Βασίλη
- (2) Η Αλίκη γίνεται ένας από τους top-four προμηθευτές του Βασίλη, ο Βασίλης ανταποδίδει
- (3) Ο Βασίλης γίνεται ένας από τους top-four προμηθευτές της Αλίκης



Επίπεδο Εφαρμογής: επισκόπηση

- Αρχές δικτυακών εφαρμογών
- Web και HTTP
- E-mail, SMTP, IMAP
- Domain Name System DNS
- Εφαρμογές P2P
- video streaming και δίκτυα διανομής περιεχομένου
- socket programming with UDP and TCP



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

87

Video Streaming and CDNs: context

- Χαρακτηριστικά Βίντεο
- Μετάδοση βιντεο-ροών
 - Buffering
 - Αναπαραγωγή
- DASH: δυναμική, οδηγούμενη από τον πελάτη μετάδοση βιντεο-ροής.
- Δίκτυα Διανομής Περιεχομένου (CDNs) - παραδείγματα

88

Video Streaming and CDNs: context

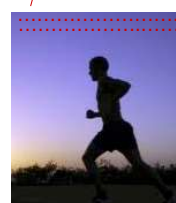
- Κίνηση Βίντεο: βασικός καταναλωτής Internet Πόρων
 - Netflix, YouTube, Amazon Prime: 80% of residential ISP traffic (2020)
- **Πρόκληση** : κλιμάκωση - Πως να εξυπηρετηθούν ~1B users?
- **Πρόκληση** : ετερογένεια
 - διαφορετικοί χρήστες διαφορετικές δυνατότητες (π.χ., ενσύρματες σε σχέση με ασύρματες-κινητές συσκευές – πολύ διαφορετικές χωρητικότητες πρόσβασης)
- **Λύση**: κατανεμημένη υποδομή επιπέδου εφαρμογής



Multimedia: video

- βίντεο: αλληλουχία εικόνων που εμφανίζονται με σταθερό ρυθμό
 - e.g., 24 images/sec
- ψηφιακή εικόνα: πίνακας από pixels
 - each pixel represented by bits
- Κωδικοποίηση: χρήση πλεονασμού **εντός** και **ανάμεσα** σε εικόνες για να μειωθεί # bits που απαιτείται για την κωδικοποίηση
 - Χωρική (spatial)– εντός της εικόνας
 - Χρονική (temporal) – από την μία στην επόμενη εικόνα)

spatial κωδικοποίηση - παράδειγμα:
αντί για N τιμές του ίδιου χρώματος (μωβ), στείλε μόνο δύο τιμές: τιμή χρώματος (μωβ) και αριθμός επαναλήψεων της ίδια τιμής (N)



frame i



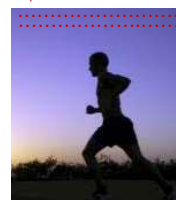
frame $i+1$

temporal κωδικοποίηση - παράδειγμα:
αντί να σταλεί ολόκληρη η εικόνα στέλνονται μόνο οι αλλαγές

Multimedia: video

- **CBR: (constant bit rate):**
κωδικοποίηση βίντεο σταθερού ρυθμού
- **VBR: (variable bit rate):**
κωδικοποίηση βίντεο μεταβλητού ρυθμού, ανάλογα με το βαθμό κωδικοποίησης (spatial, temporal)
- **examples:**
 - MPEG 1 (CD-ROM) 1.5 Mbps
 - MPEG2 (DVD) 3-6 Mbps
 - MPEG4 (often used in Internet, 64Kbps – 12 Mbps)

spatial coding example: instead of sending N values of same color (all purple), send only two values: color value (purple) and number of repeated values (N)



frame i

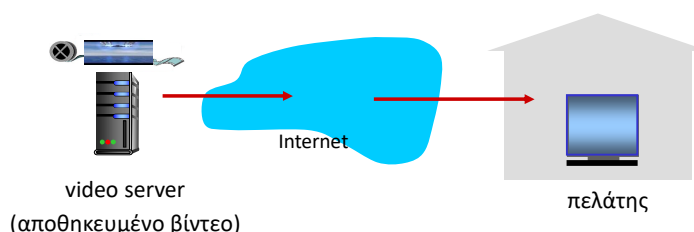


frame $i+1$

temporal coding example: instead of sending complete frame at $i+1$, send only differences from frame i

Σενάριο μετάδοσης αποθηκευμένου βίντεο

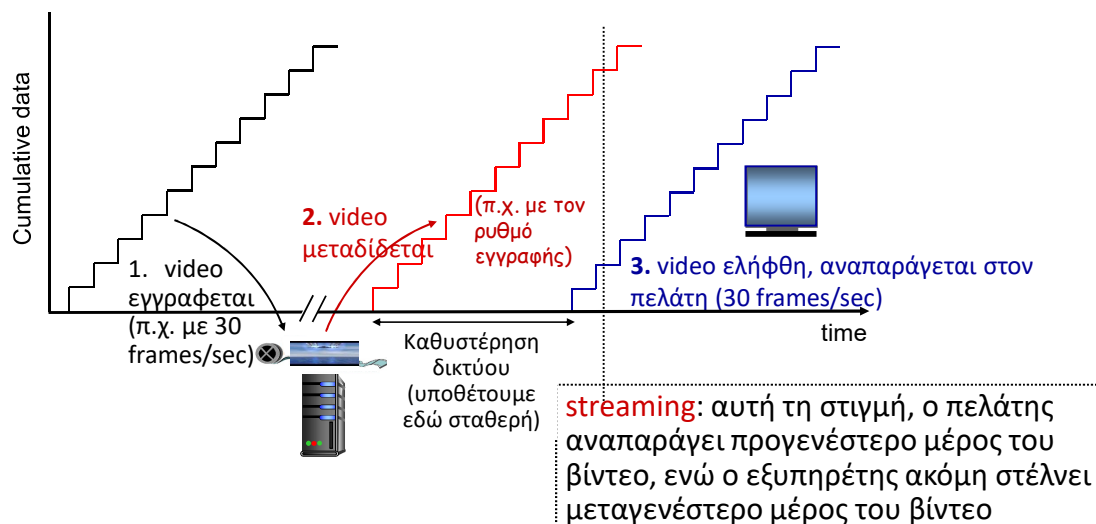
Απλό σενάριο:



Main challenges:

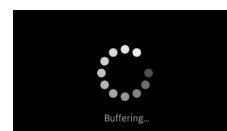
- BW μεταξύ εξυπηρέτη-πελάτη θα μεταβάλλεται με το χρόνο, λόγω μεταβαλλόμενου επιπέδου συμφόρησης στο δίκτυο (οικία, δίκτυο πρόσβασης, κορμό δικτύου, εξυπηρέτη βίντεο)
- Απώλειες / καθυστέρηση πακέτων λόγω συμφόρησης θα καθυστερούν την αναπαραγωγή, ή θα οδηγήσουν σε φτωχή ποιότητα βίντεο

Μετάδοση αποθηκευμένου βίντεο

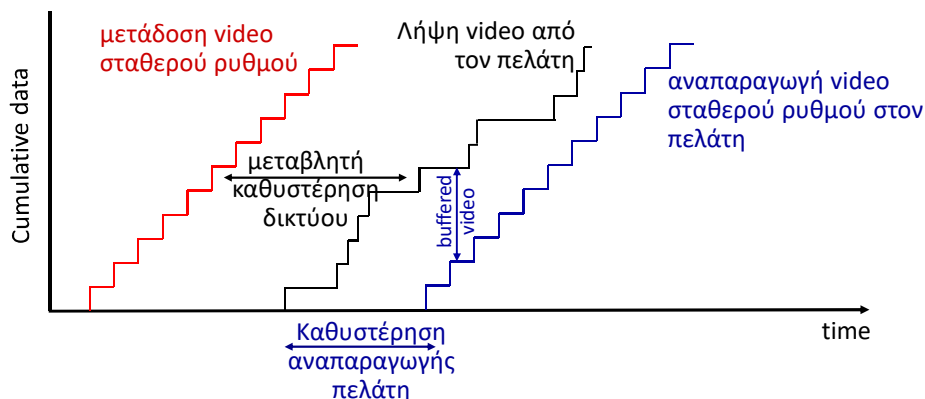


Μετάδοση αποθηκευμένου βίντεο: προκλήσεις

- **Ο περιορισμός συνεχούς αναπαραγωγής:** κατά την αναπαραγωγή στον πελάτη, ο χρονισμός αναπαραγωγής πρέπει να συμπίπτει με τον αρχικό συγχρονισμό
 - ... αλλά, **οι καθυστερήσεις δικτύου είναι μεταβλητές (jitter)**, οπότε απαιτείται **buffer στον πελάτη** ώστε να ικανοποιείται ο περιορισμός συνεχούς αναπαραγωγής
- Άλλες προκλήσεις:
 - Αλληλεπίδραση πελάτη: pause, fast-forward, rewind, jump through video
 - Πακέτα βίντεο μπορεί να χαθούν, να επαναμεταδοθούν



Μετάδοση αποθηκευμένου βίντεο: buffer αναπαραγωγής



- **Αποθήκευση (buffering) στον πελάτη και η καθυστέρηση αναπαραγωγής:** αντισταθμίζουν την καθυστέρηση και την μεταβλητότητα της (*delay-jitter*) που εισάγει το δίκτυο.

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

95

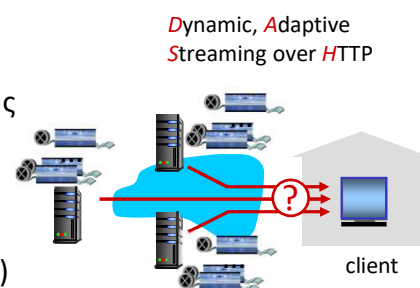
Πολυμέσα συνεχούς ροής: DASH

□ εξυπηρέτης (*server*):

- ❖ διαιρεί το αρχείο βίντεο σε πολλαπλά κομμάτια
- ❖ κάθε κομμάτι κωδικοποιείται σε πολλούς διαφορετικούς ρυθμούς
- ❖ Διαφορετικές κωδικοποιήσεις αποθηκεύονται σε διαφορετικά αρχεία
- ❖ Τα αρχεία αποθηκεύονται σε διάφορους κόμβους (CDN)
- ❖ *αρχείο manifest*: παρέχει URLs για διαφορετικά κομμάτια

□ πελάτης (*client*):

- ❖ περιοδικά υπολογίζει το εύρος ζώνης ανάμεσα σε πελάτη και εξυπηρέτη
- ❖ βάζει του αρχείου *manifest*, αιτείται ένα κομμάτι κάθε φορά
 - επιλέγει μέγιστο ρυθμό κωδικοποίησης ανάλογα με το εύρος ζώνης της σύνδεσής του κάθε στιγμή.
 - δυναμική εναλλαγή ανάμεσα στις διαθέσιμες επιλογές κωδικοποίησης (ανάλογα με το διαθέσιμο εύρος ζώνης κάθε στιγμή) και από διαφορετικούς εξυπηρέτες



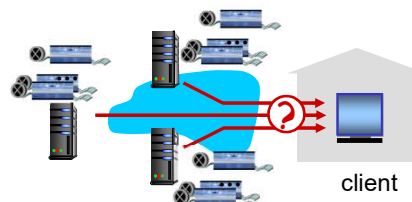
Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

96

Πολυμέσα συνεχούς ροής: DASH

□ “εξυπνάδα” στον πελάτη: ο πελάτης επιλέγει

- ❖ **πότε** να αιτηθεί κομμάτι (αποφυγή buffer over /underflow)
- ❖ τον **ρυθμό κωδικοποίησης** που θα αιτηθεί (υψηλότερη ποιότητα σε συνθήκες περισσότερου διαθέσιμου εύρους ζώνης)
- ❖ **πού** να αιτηθεί το κομμάτι (είτε έναν URL εξυπηρετή πλησιέστερό του ή με υψηλότερο διαθέσιμο εύρος ζώνης)



Μετάδοση βιντεοροής = κωδικοποίηση + DASH + αποθήκευση αναπαραγωγής

Δίκτυα Διανομής Περιεχομένου Content Delivery Networks (CDNs)

πρόκληση: πώς να διανείμω (stream) περιεχόμενο (ανάμεσα σε εκατομμύρια βίντεο) σε εκατοντάδες χιλιάδες ταυτόχρονους χρήστες;

- **επιλογή 1:** μοναδικός, μεγάλος “mega-server”
 - μοναδικό σημείο αποτυχίας
 - σημείο δικτυακής συμφόρησης
 - μακρύ μονοπάτι προς απομακρυσμένους πελάτες

.... απλά: αυτή η λύση **δεν μπορεί να κλιμακώσει**

Content distribution networks (CDNs)

πρόκληση: πώς να διανείμω (stream) περιεχόμενο (ανάμεσα σε εκατομμύρια βίντεο) σε εκατοντάδες χιλιάδες ταυτόχρονους χρήστες;

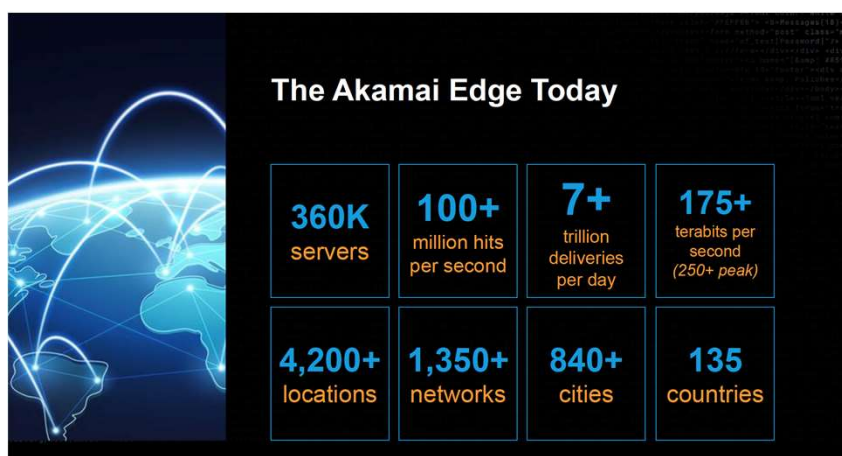
- **επιλογή 2:** αποθήκευση/προσφορά πολλαπλών αντιγράφων των βίντεο σε πολλαπλά γεωγραφικά κατανομημένα σημεία εξυπηρέτησης (**CDN**)
 - **βαθεία εισχώρηση (enter deep):** εγκατάσταση CDN εξυπηρετών σε πολλά δίκτυα προσπέλασης
 - κοντά σε χρήστες
 - π.χ., Akamai, 240,000 εξυπηρετές σε > 120 χώρες (2015)
 - **κοντά στη βάση (bring home):** μικρότερος αριθμός (δεκάδες) μεγαλύτερων συμπλεγμάτων εξυπηρετών σε IXP (Points of Presence) κοντά (αλλά όχι μέσα) στα δίκτυα προσπέλασης
 - π.χ., Limelight



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

99

Akamai σήμερα:



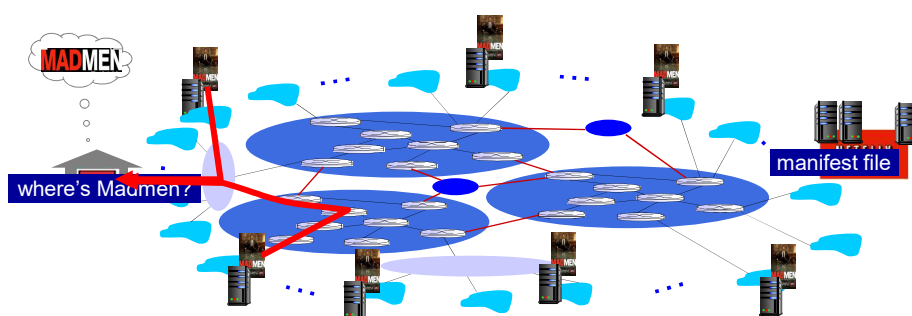
Source: <https://networkingchannel.eu/living-on-the-edge-for-a-quarter-century-an-akamai-retrospective-downloads/>

Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

100

Content distribution networks (CDNs) – παράδειγμα streaming video

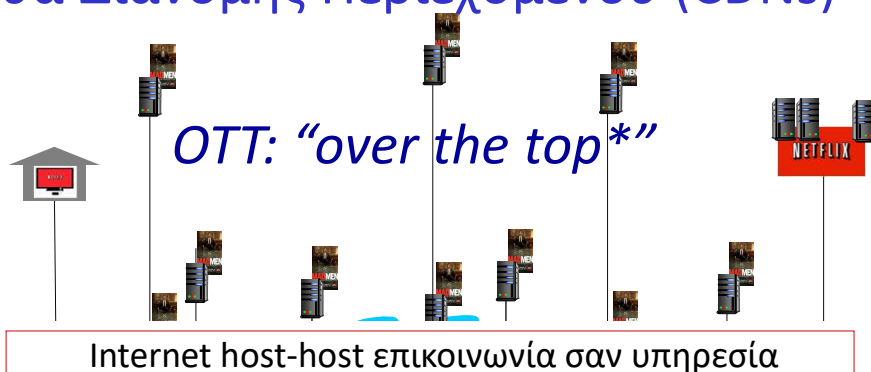
- CDN: αποθηκεύει αντίγραφα του περιεχομένου (π.χ. MADMEN) στους κόμβους CDN
- ο χρήστης αιτείται περιεχόμενο, ο πάροχος επιστρέφει manifest
 - Βάσει του manifest, ο πελάτης κατεβάζει περιεχόμενο με το μέγιστο εφικτό ρυθμό
 - μπορεί να επιλέξει διαφορετικό ρυθμό ή αντίγραφο αν το μονοπάτι υποστεί συμφόρηση



Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

101

Δίκτυα Διανομής Περιεχομένου (CDNs)



Internet host-host επικοινωνία σαν υπηρεσία

OTT προκλήσεις: πως αντιμετωπίζεται η συμφόρηση στο Διαδίκτυο από τα «άκρα»?

- Τι περιεχόμενο να τοποθετηθεί σε ποιόν CDN κόμβο?
- Από ποιόν CDN κόμβο να ληφθεί κάποιο το περιεχόμενο? Με τι ρυθμό?

* of an IP infrastructure

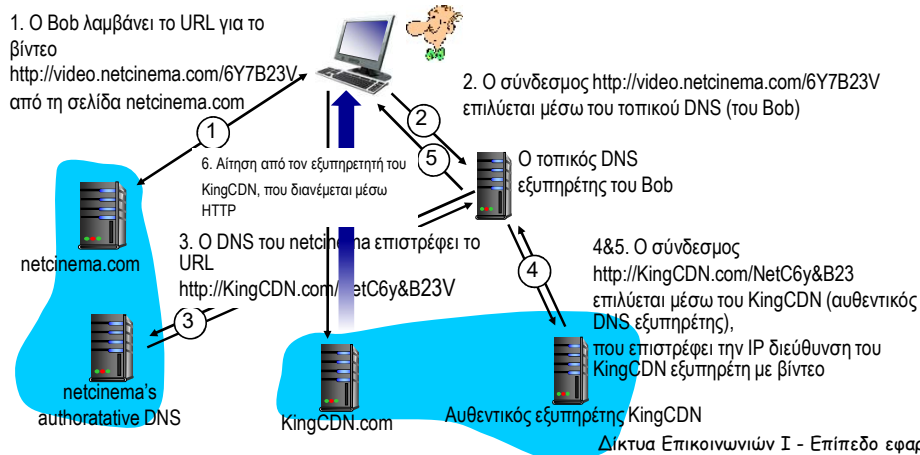
Δίκτυα Επικοινωνιών Ι - Επίπεδο εφαρμογής

102

Δίκτυα Διανομής Περιεχομένου: Παράδειγμα NetCinema

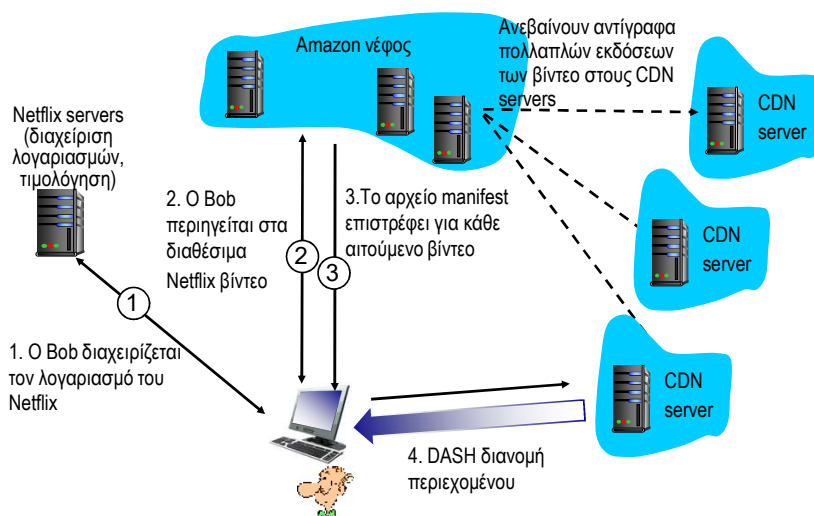
Ο Bob (πελάτης) αιτείται το βίντεο <http://video.netcinema.com/6Y7B23V>

- Το βίντεο είναι αποθηκευμένο στο CDN στη θέση <http://KingCDN.com/NetC6y&B23V>



103

Παράδειγμα Netflix



104

Κεφάλαιο 2: Σύνοψη

Η επισκόπησή μας των δικτυακών εφαρμογών τώρα ολοκληρώθηκε!

- ❑ Αρχιτεκτονικές εφαρμογής
 - Πελάτης-εξυπηρετής
 - P2P
- ❑ Απαιτήσεις υπηρεσίας μιας εφαρμογής:
 - αξιοπιστία, εύρος ζώνης, καθυστέρηση
- ❑ Μοντέλο υπηρεσίας μεταφοράς Διαδικτύου
 - συνδεδειστροφής (connection-oriented), αξιόπιστη: TCP
 - μη αξιόπιστη, δεδομενογράμματα (datagrams): UDP
- ❑ συγκεκριμένα πρωτόκολλα:
 - HTTP
 - SMTP, POP, IMAP
 - DNS
 - P2P: BitTorrent
- ❑ Βίντεο Συνεχούς Ροής , Δίκτυα Διανομής Περιεχομένου

Κεφάλαιο 2: Σύνοψη

Το πιο σημαντικό: μάθαμε για τα πρωτόκολλα!

- ❑ Τυπική ανταλλαγή μηνυμάτων αιτήματος/απόκρισης:
 - Ο πελάτης ζητά πληροφορία ή υπηρεσία
 - Ο εξυπηρετής αποκρίνεται με δεδομένα, κωδικό κατάστασης
- ❑ Μορφές μηνυμάτων:
 - Κεφαλίδες: πεδία που δίνουν πληροφορία σχετικά με τα δεδομένα
 - Δεδομένα : πληροφορία που επικοινωνείται
- ❑ Σημαντικά θέματα:
 - ❑ Κεντροποιημένο - κατακεντρωμένο (centralized vs. decentralized)
 - ❑ ακαταστατικό - με κατάσταση (stateless vs. stateful)
 - ❑ κλιμάκωση (scalability)
 - ❑ αξιόπιστη - μη αξιόπιστη μεταφορά μηνύματος
 - ❑ “πολυπλοκότητα στα άκρα του δικτύου”