Δίκτυα Υπολογιστών 6/11 6ο

**Το Domain Name System (DNS)**

DNS: Domain Name System

**Domain Name System (DNS):**  
▪ *κατανεμημένη βάση δεδομένων*υλοποιείται μέσω ιεραρχίας πολλών *εξυπηρετών ονομάτων*▪ *πρωτόκολλο επιπέδου εφαρμογών:*Η/Υ, εξυπηρέτες ονομάτων επικοινωνούν για να αντιστοιχίσουν *resolve* ονόματα (μετάφραση όνομα/δνση)

• *Σημ*.: βασική λειτουργία Διαδικτύου, υλοποιούμενη ως πρωτόκολλο επιπέδου εφαρμογών (και ο πελάτης και ο εξυπεέτης είναι εφαρμογές για να τρέχουν)  
• πολυπλοκότητα στα “άκρα” του δικτύου

*άνθρωποι:* πολλά αναγνωριστικά:  
• ΑΜΚΑ, όνομα, αριθμός διαβατηρίου  
*Η/Υ στο Διαδίκτυο, δρομολογητές:*• δνση IP (32 bit) – χρησιμοποιείται  
για διευθύνσεις στα δεδομενογράμματα  
• “όνομα”, π.χ., cs.umass.edu - –  
χρήση από ανθρώπους  
*Ε:* πώς αντιστοιχούμε δνση IP σε  
όνομα και αντίστροφα;

DNS: υπηρεσίες, δομή

υπηρεσίες DNS:  
▪ Μετάφραση όνομα Η/Υ-σε- δνση IP  
▪ ψευδώνυμα Η/Υ  
• canonical, alias names

▪ mail server aliasing  
▪ κατανομή φόρτου(hotmail,gmail)

• αντίγραφα εξυπηρετών Web:πολλές δνσεις IP αντιστοιχούν σε ένα όνομα

(για πολύ λίγο χρόνο εμφανίζεται ένα όνομα συμβολικό, γιατί το 1ο ερώτημα που θα πάρει το DNS να πάρει στο επόμενο ερώτημα)

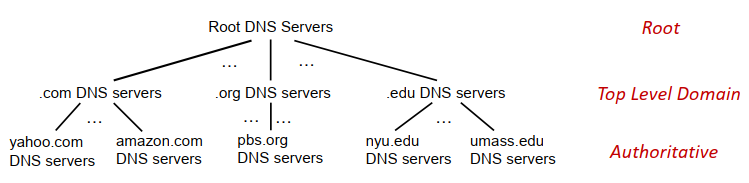
*Ε: Γιατί να μηνκεντρικοποιήσουμε το DNS;*▪ μοναδικό σημείο αποτυχίας  
▪ όγκος κυκλοφορίας (για να πηγαίνει προς μία διευθηνση)  
▪ απομακρυσμένη κεντρικοποιημένηβάση δεδομένων  
▪ συντήρηση (τεράστιας βάσης δεδομένων)

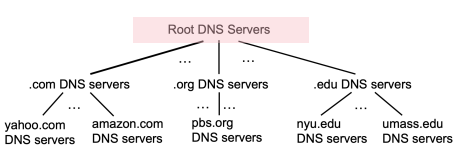
*Α: δεν κλιμακώνει!*▪ Μόνον DNS servers Comcast: 600 διςερωτήματα DNS/ημέρα  
▪ Μόνον DNS servers Akamai: 2,2 τρις DNS ερωτήματα/ημέρα

Σκέψεις γύρω από το DNS (πως είναι σχεδιασμένη αυτή η βάση δεδομενών )

τεράστια κατανεμημένη βάση δεδομένων:  
▪ ~ δις εγγραφές, κάθε μία απλή  
χειρίζεται πολλά *τρις* ερωτημάτων/ημέρα:  
▪ *πολύ* περισσότερες αναγνώσεις από εγγραφές  
▪ *απόδοση έχει σημασία:* σχεδόν κάθε συναλλαγή στο Διαδίκτυο διαδρά με το DNS – τα msecs μετράν!  
οργανωτικά, φυσικά αποκεντρωμένο:  
▪ εκατομμύρια διαφορετικών οργανισμών υπεύθυνοι για τις εγγραφές τους  
“αλεξίσφαιρο”: αξιοπιστία, ασφάλεια

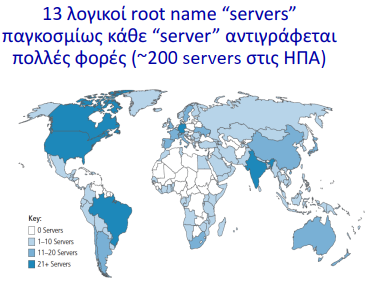
DNS: μία κατανεμημένη, ιεραρχική βάση δεδομένων



πελάτης θέλει IP για www.amazon.com; 1η προσέγγιση:  
▪ πελάτης ρωτά root server για να βρει τον εξυπηρέτη DNS για το .com  
▪ πελάτης ρωτά εξυπηρέτη DNS του .com για να βρει εξυπηρέτη DNS του amazon.com  
▪ πελάτης ρωτά εξυπηρέτη DNS του amazon.com για να βρει την δνση IP του  
www.amazon.com

DNS: root name servers

▪ επίσημο σημείο επαφής εσχάτης λύσης από   
name servers που δεν μπορούν να βρουν  
μετάφραση για το όνομα

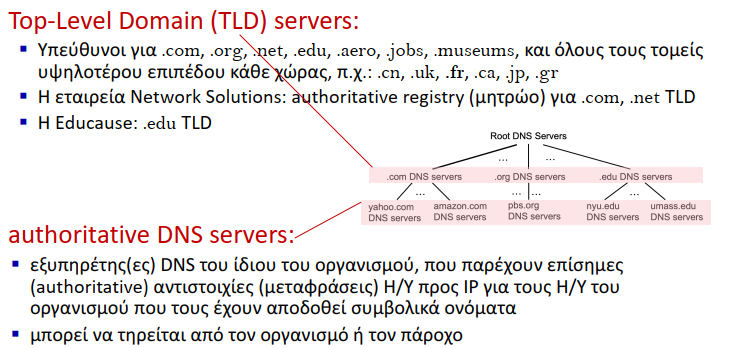
▪ επίσημο σημείο επαφής εσχάτης λύσης από name servers που δεν μπορούν να βρουν μετάφραση για  
το όνομα  
▪ *απίστευτα σημαντική* λειτουργία  
του Διαδικτύου

• Διαδίκτυο δεν θα λειτουργούσε αλλιώς!

• DNSSEC – παρέχει ασφάλεια  
(αυθεντικοποίηση, ακεραιότητα μηνυμ.)

▪ ICANN (Internet Corporation for Assigned Names and Numbers) διαχειρίζεται το root DNS domain

Top-Level Domain και authoritative servers



Τοπικοί εξυπηρέτες DNS (Seconderr Local Topical Servers)

▪ όταν Η/Υ κάνει ερώτημα DNS-> στην cache, το ερώτημα αποστέλλεται στον *τοπικό* του εξυπηρέτη DNS

• Τοπικός εξυπηρέτης DNS επιστρέφει απόκριση, απαντώντας:

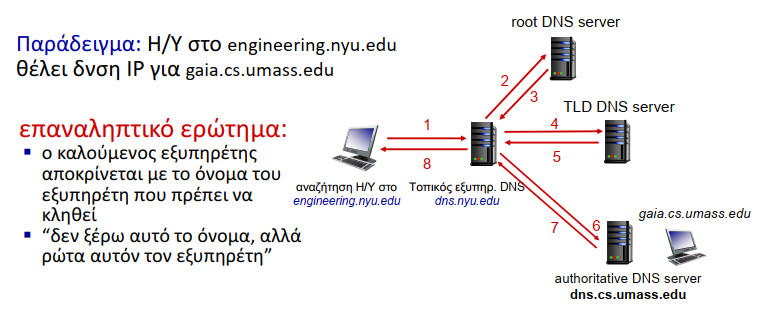
• από την τοπική του cache προσφάτων μεταφράσεων ζευγών όνομα<->δνση (πιθανώς παρωχημένες!)

• προωθώντας αίτημα στην ιεραρχία DNS για αντιστοίχιση  
• κάθε Πάροχος έχει τον δικό του τοπικό εξυπηρέτη ονομάτων DNS. Για να βρείτε τον δικό σας:

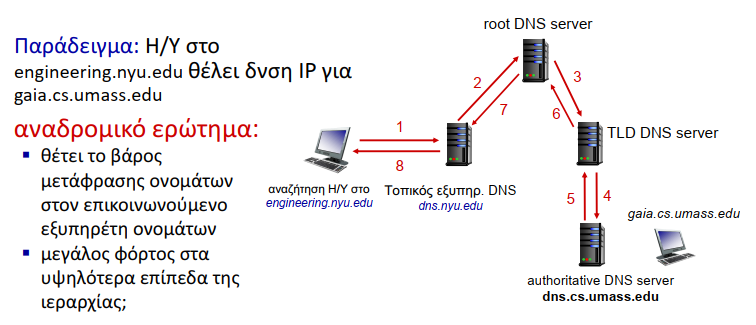
• MacOS: % scutil –dns

• Windows: >ipconfig /all

▪ Τοπικός εξυπηρέτης DNS δεν ανήκει αυστηρά στην ιεραρχία

Μετάφραση ονόματος στο DNS: επαναληπτικό ερώτημα

Μετάφραση ονόματος στο DNS: αναδρομικό ερώτημα



Caching της πληροφορίας DNS

*▪ μόλις (οιοσδήποτε) εξυπηρέτης ονομάτων μάθει μία αντιστοιχία, την τοποθετεί στην cache, και αμέσως επιστρέφει μία cached αντιστοιχία ως απόκριση σε ερώτημα*

*• caching βελτιώνει χρόνο απόκρισης*

*• καταχωρίσεις cache timeout (εξαφανίζονται) μετά κάποιον χρόνο (TTL)*

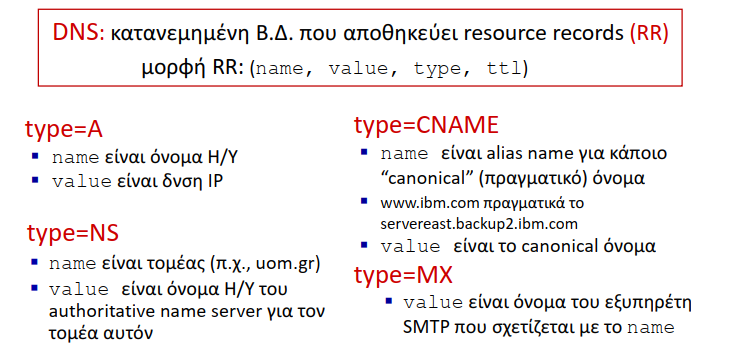
*• Δνσεις TLD servers τυπικά αποθηκεύονται (cached) σε τοπικούς εξυπηρέτες ονομάτων*

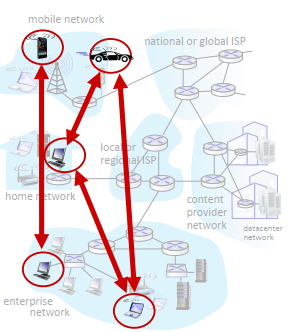
*▪ cached καταχωρήσεις μπορεί να έχουν λήξει*

*• εάν Η/Υ με όνομα αλλάξει δνση IP, ίσως δεν είναι γνωστό σε όλο το  
Διαδίκτυο έως ότου λήξουν όλα τα TTL!*

*• μετάφραση όνομα-σε-δνση βέλτιστης προσπάθειας!*

Εγγραφές DNS



*Εφαρμογές P2P*

*Αρχιτεκτονική peer-to-peer (P2P)*

*▪ εξυπηρέτης δεν είναι διαρκώς ενεργός  
▪ οπιαδήποτε τερματικά συστήματα επικοινωνούν απ’ ευθείας μεταξύ τους  
▪ Peers (ομότιμοι) ζητούν υπηρεσία από άλλους peers, παρέχουν με την σειρά τους υπηρεσία σε άλλους peers*

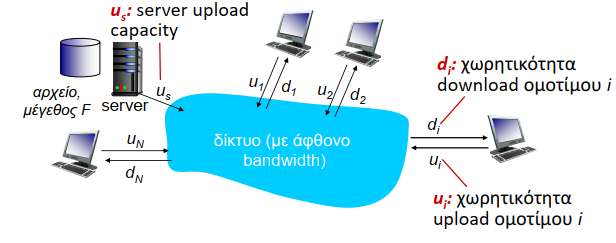
*• αυτο-κλιμάκωση – νέοι peers φέρνουν νέα χωρητικότητα, και απαιτήσεις υπηρεσίας*

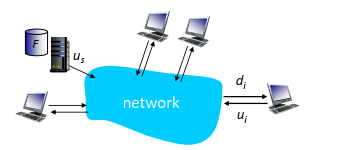
*▪ peers συνδέονται κατά διαστήματα και  
αλλάζουν δνσεις IP*

*• πολύπλοκη διαχείριση  
▪ παραδείγματα: διαμοιρασμός αρχείων P2P  
(BitTorrent), streaming (KanKan), VoIP (Skype)*

*Διανομή αρχείων: πελάτης-εξυπηρέτης έναντι P2P*

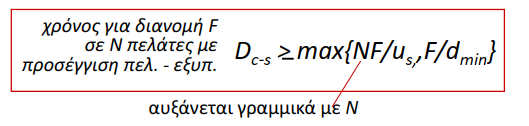
***Ε****: πόσος χρόνος για διανομή αρχείου (μέγεθος F) από έναν εξυπηρέτη σε N ομοτίμους;*

*• Η χωρητικότητα upload/download ομοτίμου είναι περιορισμένος πόρος*

Χρόνος διανομής αρχείου: πελάτης-εξυπηρέτης

▪ *μετάδοση εξυπηρέτη:* πρέπει να στείλει σειριακά (ανεβάσει) *N*αντίγραφα του αρχείου:

• χρόνος αποστολής 1 αντιγράφου: *F/us*

• χρόνος αποστολής *Ν* αντιγράφων: *NF/ us*▪ *πελάτης:* κάθε ένας πρέπει να κατεβάσει 1 αντίγραφο αρχείου

• *dmi*n = min ρυθμός πελατών για κατέβασμα

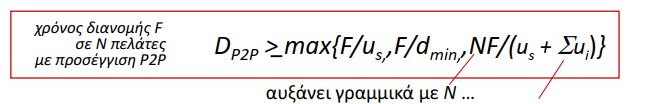
• min χρόνος κατεβάσματος (από όλους): *F/ dmi*n Χρόνος διανομής αρχείου: P2P

▪ μετάδοση εξυπηρέτη:πρέπει να ανεβάσει τουλάχιστον ένα αντίγραφο

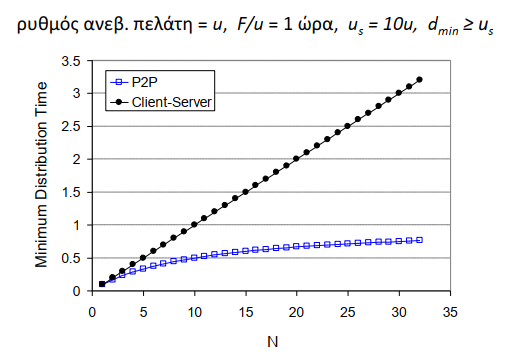
• χρόνος αποστολής 1 αντιγράφου: *F/us*▪ πελάτης:κάθε πελάτης πρέπει να κατεβάσει 1 αντίγραφο

• min χρόνος κατεβ. από πελάτη: *F/dmin*

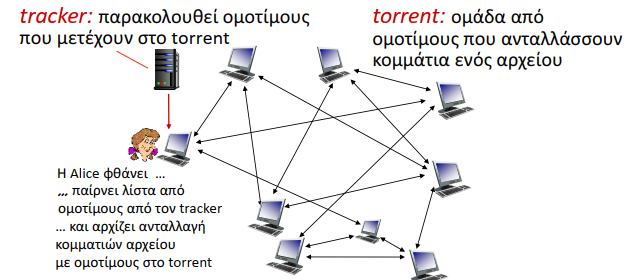
▪ πελάτες:ως σύνολο πρέπει να κατεβάσουν *NF* bits

• max ρυθμός ανεβ. (περιορίζοντας max ρυθμό κατεβ.) είναι *us +* S*ui*

… αλλά το ίδιο και αυτός, αφού κάθε peer φέρνει πρόσθετη χωρητικότητα

Πελάτης-εξυπηρέτης έναντι P2P: παράδειγμα

Διανομή αρχείων P2P: BitTorrent

▪ αρχείο διαιρείται σε κομμάτια 256Kb  
▪ ομότιμοι στο torrent αποστέλλουν/λαμβάνουν κομμάτια του αρχείου

▪ ομότιμος που συνδέεται με torrent:

• δεν έχει κομμάτια, αλλά θα συσσωρεύσει με τον χρόνο από άλλους ομοτίμους

• εγγράφεται στον tracker για να λάβει λίστα από ομοτίμους, συνδέεται σε υποσύνολο ομοτίμων (“γείτονες”)

▪ ενώ κατεβάζει, ανεβάζει κομμάτια σε άλλους ομοτίμους

▪ μπορεί να αλλάζει ομοτίμους με τους οποίους ανταλλάσσει κομμάτια

▪ churn (φύρα): ομότιμοι μπορούν να έρχονται και να φεύγουν

▪ μόλις ένας ομότιμος έχει ολόκληρο το αρχείο, μπορεί (εγωιστικά) να φύγει ή (αλτρουιστικά) να παραμείνει στο torrent

BitTorrent: ζήτηση, αποστολή κομματιών αρχείων

Αποστέλλοντας κομμάτια: μία σου και  
μία μου  
▪ η Alice στέλνει κομμάτια στους πρώτους 4 ομοτίμους που επί του παρόντος της στέλνουν κομμάτια *με τον μέγιστο ρυθμό*

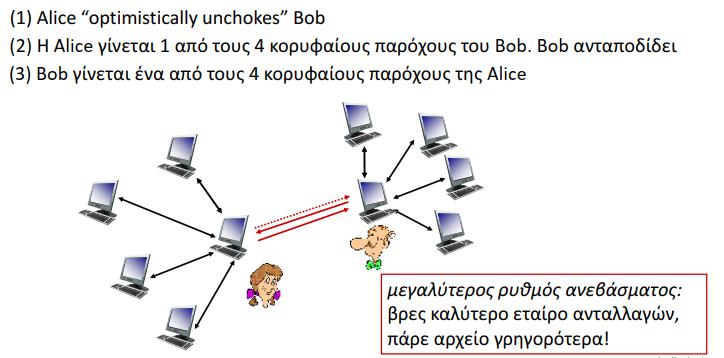
• οι άλλοι πνίγονται από την Alice (δεν λαμβάνουν κομμάτια από αυτήν)  
• επαναξιολογεί τους κορυφαίους 4 κάθε 10 secs

▪ κάθε 30 secs: επιλέγει τυχαία έναν άλλον ομότιμο, αρχίζει να του στέλνει κομμάτια

• αυτόν “optimistically unchoke”

• αυτός μπορεί να γίνει ένας από τους κορυφαίους 4 της

Ζητώντας κομμάτια:  
▪ σε οποιαδήποτε χρονική  
στιγμή, διάφοροι ομότιμοι  
έχουν διάφορα υποσύνολα  
κομματιών του αρχείου  
▪ περιοδικά, η Alice ζητά από  
κάθε ομότιμο λίστα των  
κομματιών που έχουν  
▪ η Alice ζητά τα κομμάτια που  
της λείπουν από ομοτίμους,  
με πρώτο το πιο σπάνιο

BitTorrent: μία σου και μία μου

video streaming και δίκτυα διανομής περιεχομένου

Video Streaming και CDNs: πλαίσιο

*▪ κυκλοφορία stream video: βασικός καταναλωτής bandwidth Διαδικτύου*

*• Netflix, YouTube, Amazon Prime: 80% της κυκλοφορίας κατεβάσματος σε οικιακούς Παρόχους (2020)  
▪ πρόκληση: κλιμάκωση – πώς να πάμε σε ~1 δις  
χρήστες  
▪ πρόκληση: ετερογένεια*

*▪ διαφορετικοί χρήστες έχουν διαφορετικές ικανότητες (π.χ., ενσύρματοι έναντι κινητών, πλούσιοι σε bandwidth έναντι πτωχών σε αυτό)*

*▪ λύση: κατανεμημένη υποδομή, στο επίπεδο  
εφαρμογών*

Πολυμέσα: βίντεο

▪ *βίντεο: ακολουθία από εικόνες εμφανιζόμενες με σταθερό ρυθμό*

*• π.χ., 24 εικόνες/sec  
▪ ψηφιακή εικόνα: πίνακας από pixel (εικονοστοιχεία)*

*• κάθε pixel αναπαρίσταται από bit  
▪ κωδικοποίηση: χρησιμοποιείστε πλεονασμό μέσα και μεταξύ εικόνων για μείωση # bit που χρησιμοποιούνται για κωδικοποίηση εικόνων*

*• χωρική (μέσα στην εικόνα)*

*• χρονική (από μία εικόνα στην επόμενη)*

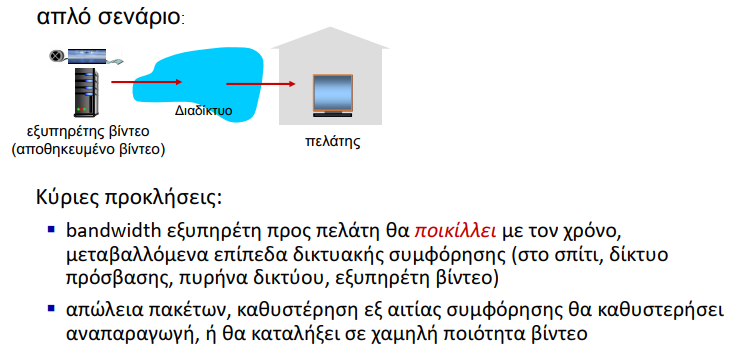
▪ CBR (Constant Bit Rate): σταθερός ρυθμός κωδικοποίησης βίντεο  
▪ VBR (Variable Bit Rate): ο ρυθμός κωδικοποίησης βίντεο μεταβάλλεται ως ποσό χωρικών, χρονικών μεταβολών

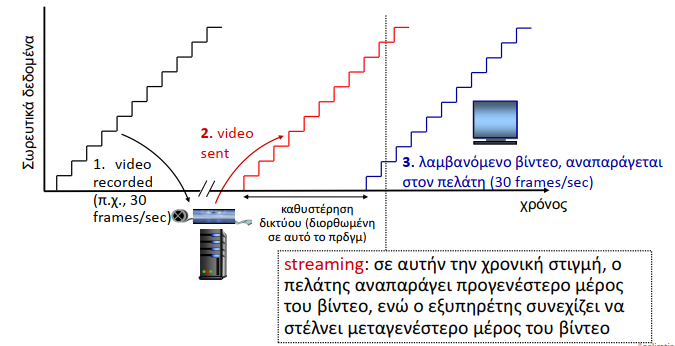
▪ παραδείγματα:

• MPEG 1 (CD-ROM) 1,5 Mbps

• MPEG2 (DVD) 3-6 Mbps

• MPEG4 (συχνή χρήση στο Διαδίκτυο, 64Kbps – 12 Mbps)

Streaming αποθηκευμένου βίντεο

Streaming αποθηκευμένου βίντεο

Προκλήσεις

▪ περιορισμός συνεχούς αναπαραγωγής: κατά την αναπαραγωγή βίντεο στον πελάτη, χρονισμός αναπαραγωγής πρέπει να ταιριάζει με τον αρχικό

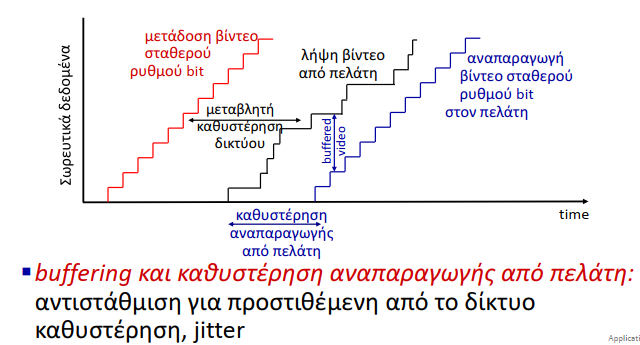
• … αλλά δικτυακές καθυστερήσεις είναι μεταβλητές (jitter), έτσι χρειαζόμαστε buffer

πλευράς πελάτη to για εκπλήρωση του περιορισμού αυτού  
▪ άλλες προκλήσεις:

• διαδραστικότητα πελάτη: pause, fast-forward, rewind, άλματα μέσα στο βίντεο

• πακέτα βίντεο μπορεί να χαθούν, επαναμεταδοθούν

Streaming αποθηκευμένου βίντεο: buffering για αναπαραγωγή



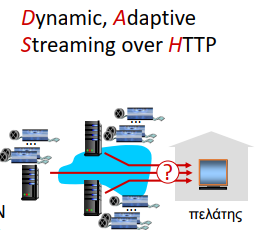
Streaming πολυμέσων: DASH

εξυπηρέτης:

▪ διαιρεί το αρχείο βίντεο σε πολλαπλά κομμάτια

▪ κάθε κομμάτι αποθηκεύεται, κωδικοποιείται σε πολλαπλούς διαφορετικούς ρυθμούς

▪ κωδικοποιήσεις διαφορετικού ρυθμού αποθηκεύονται σε διαφορετικά αρχεία

▪ αρχεία αντιγράφονται σε διάφορους κόμβους CDN

▪ manifest file: παρέχει URLs για διαφορετικά κομμάτια

πελάτης:

▪ περιοδικά εκτιμά το bandwidth από εξυπηρέτη προς πελάτη

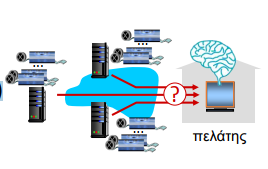
▪ συμβουλεύεται το manifest, ζητά ένα κομμάτι την φορά

• επιλέγει μέγιστο διατηρήσιμο ρυθμό κωδικοποίησης σύμφωνα με το τρέχον bandwidth

• μπορεί να επιλέγει διαφορετικούς ρυθμούς κωδικοποίησης σε διαφορετικά

χρονικά σημεία (εξαρτάται από το διαθέσιμο bandwidth κάθε φορά), και από

διαφορετικούς εξυπηρέτε

▪*“νοημοσύνη”* στον πελάτη: ο πελάτης προσδιορίζει:

• **πότε**να ζητήσει κομμάτι (ώστε να μην λάβει χώρα buffer starvation ή υπερχείλιση)

• **τι ρυθμό κωδικοποίησης**να ζητήσει (υψηλότερη ποιότητα όταν υπάρχει διαθέσιμο bandwidth)  
• **πού**να ζητήσει κομμάτι (μπορεί να ζητήσει από τον εξυπηρέτη URL που είναι “κοντά” στον πελάτη ή που έχει μεγάλο bandwidth διαθέσιμο)

**Streaming video = κωδικοποίηση + DASH + buffering αναπαραγωγής**

Content distribution networks (CDNs)

**“Δίκτυα Διανομής Περιεχομένου”**  
*πρόκληση:* πώς να επιτύχουμε ροή περιεχομένου (επιλεγμένου από εκατομμύρια βίντεο) σε εκατοντάδες χιλιάδες ταυτόχρονους χρήστες;  
▪ *επιλογή 1:* ένας, μεγάλος “mega-server”

• μοναδικό σημείο αποτυχίας  
• σημείο δικτυακής συμφόρησης  
• μεγάλη (και πιθανώς με συμφόρηση) διαδρομή προς απομακρυσμένους πελάτες

…. πολύ απλά: αυτή η λύση *δεν κλιμακώνει*

▪ *επιλογή 2:* αποθήκευση/εξυπηρέτηση πολλαπλών αντιγράφων από διάφορα βίντεο σε πολλαπλές τοποθεσίες γεωγραφικά κατανεμημένες *(CDN)*

• *enter deep*: σπρώξτε εξυπηρέτες CDN βαθιά σε πολλά δίκτυα πρόσβασης

• κοντά στους χρήστες  
• Akamai: 240.000 εξυπηρέτες εγκατεστημένοι σε > 120 χώρες (2015)

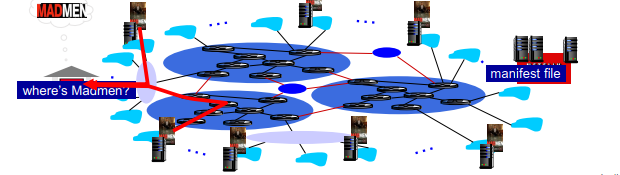
• *bring home*: μικρότερος αριθμός (10δες) από μεγαλύτερες συστοιχίες σε POP (σημεία παρουσίας) κοντά (αλλά όχι μέσα) σε δίκτυα πρόσβασης

• χρησιμοποιείται από την Limelight

▪ CDN: αποθηκεύει αντίγραφα περιεχομένου (π.χ., MADMEN) σε κόμβους CDN

▪ συνδρομητής ζητά περιεχόμενο, πάροχος υπηρεσίας επιστρέφει manifest

• χρησιμοποιώντας το manifest, ο πελάτης ανακτά περιεχόμενο στον μέγιστο υποστηριζόμενο ρυθμό  
• μπορεί να επιλέξει διαφορετικό ρυθμό ή αντίγραφο εάν συμφόρηση σε δικτυακή διαδρομή





*Προκλήσεις OTT:* βγάζοντάς τα πέρα με συμφόρηση στο Διαδίκτυο από τα “άκρα”

▪ τι περιεχόμενο να τοποθετήσουμε και σε ποιόν κόμβο CDN;  
▪ από ποιόν κόμβο CDN να ανακτήσουμε περιεχόμενο; Με ποιόν ρυθμό;