

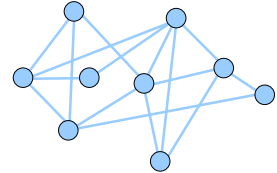
Διαχείριση Δεδομένων σε Συστήματα Ομότιμων Κόμβων

Εισαγωγή

Μεγάλος αριθμός από κόμβους (συνήθως υπολογιστές στην «άκρη» του διαδικτύου)

Συμμετέχουν στον υπολογισμό

- Προσφέροντας πόρους (π.χ., μουσικά αρχεία) - servers (εξυπηρετητές)
- Χρησιμοποιώντας πόρους που προσφέρουν άλλοι κόμβοι - clients (πελάτες)



Εισαγωγή

Κάποιες Ιδιότητες

- Αποκέντρωση (και στη διαχείριση)
- Αυτονομία (οι κόμβοι αποφασίζουν τι δεδομένα να αποθηκεύσουν, ποια μηνύματα να προωθήσουν κλπ)
- Δυναμική Συμμετοχή (churn)
- Εγωιστική Συμπεριφορά ("free-riding")
- Προβλήματα Εμπιστοσύνης (trust) - επίσης ασφάλεια, privacy

Βασικός Στόχος

- Αποδοτικό μοίρασμα πόρων και κόστους

Εισαγωγή

Θα δούμε το γενικό πρόβλημα της ΑΝΑΖΗΤΗΣΗΣ

Ένας κόμβος θέτει ένα ερώτημα για κάποιο πόρο (ή δεδομένο): πως θα το βρει, δηλαδή πως θα εντοπίσει τον κόμβο ή τους κόμβους που το έχουν

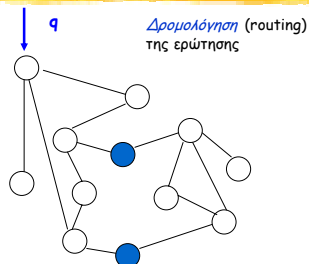
Το Πρόβλημα της Αναζήτησης

Παρατηρήσεις

Πόροι προς Διαμοιρασμό

- Συνήθως αρχεία (θnutella, Napster, Freenet, KaZaa)
- Αποθηκευτικός χώρος (lockss)
 - Υπολογιστική δύναμη (seti@home, Entropia)
 - Bandwidth (PeerCast)

Η ερώτηση συνήθως το «όνομα» κάποιου αρχείου ή κάποιες λέξεις κλειδιά



Κεντρικό Ευρετήριο (Napster)

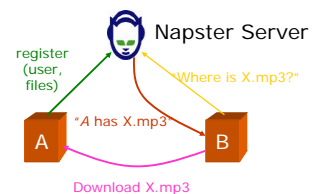
Υπάρχει ένα κεντρικό ευρετήριο με πληροφορία για όλα τα αρχεία που υπάρχουν στο σύστημα

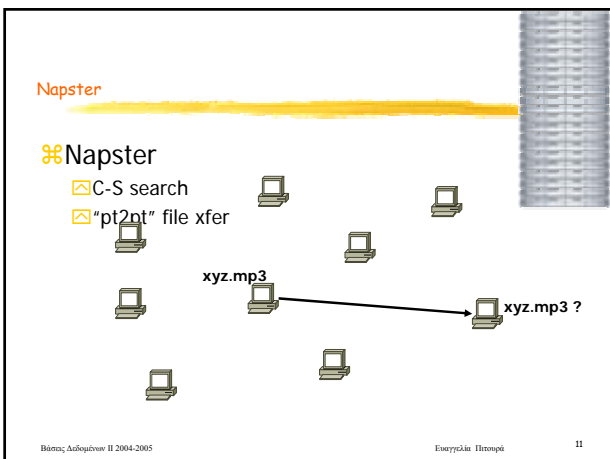
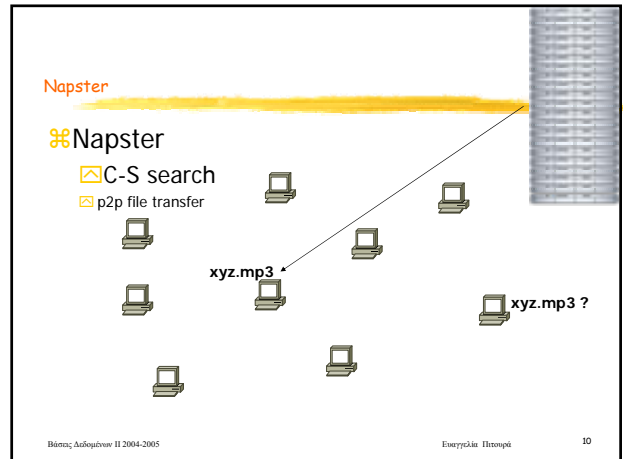
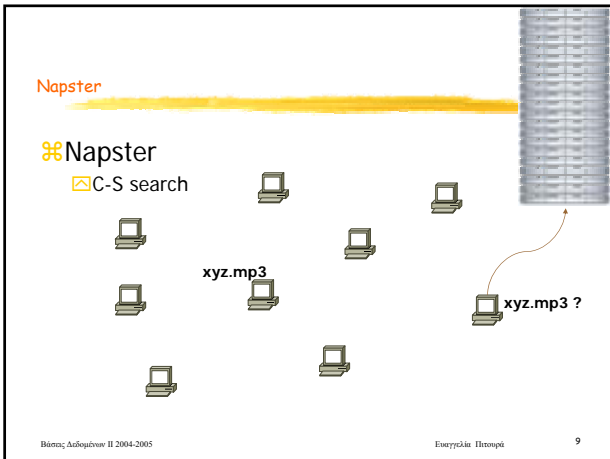
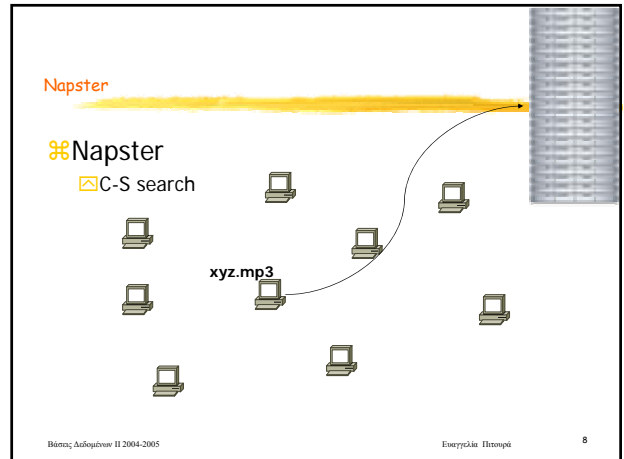
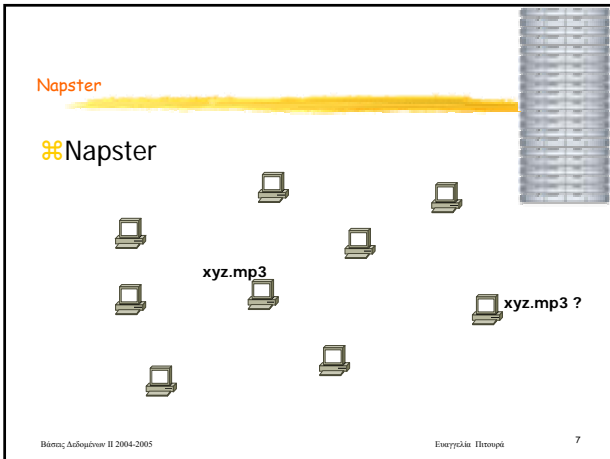
- Register/Lookup (p2p μεταφορά αρχείου)
- αντί για Register (crawler!)

Πολλή γρήγορη αναζήτηση, αλλά

Προβλήματα

- Αποτυχίες
- Φορτίο (σκεφτείτε τις ενημερώσεις) αλλά και
- Νομικές αγωνές





Η Ιστορία του Napster

- ☒ **May 1999:** Napster Inc. file share service founded by Shawn Fanning and Sean Parker
- ☒ **Dec 7 1999:** Recording Industry Association of America (RIAA) sues Napster for copyright infringement
- ☒ **April 13, 2000:** Heavy metal rock group Metallica sues Napster for copyright infringement
- ☒ **April 27, 2000:** Rapper Dr. Dre sues Napster
- ☒ **May 3, 2000:** Metallica's attorney claims 335,000 Internet users illegally share Metallica's songs via Napster
- ☒ **July 26, 2000:** Court orders Napster to shut down
- ☒ **Oct 31, 2000:** Bertelsmann becomes a partner and drops lawsuit
- ☒ **Feb 12, 2001:** Court orders Napster to cease trading copyrighted songs and to prevent subscribers to gain access to content on its search index that could potentially infringe copyrights
- ☒ **Feb 20, 2001:** Napster offers \$1 billion to record companies (rejected)
- ☒ **March 2, 2001:** Napster installs software to satisfy the order

Βίτσας, Δαδογιάννης II 2004-2005

Ερωτήματα Πίττογιά

12

Κατανεμημένη Προσέγγιση

Κατανεμημένο Μοντέλο

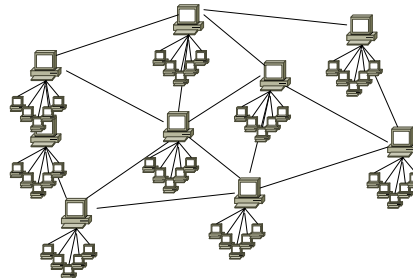
Δεν υπάρχει κεντρικό ευρετήριο
Παραδείγματα: Freenet, Gnutella

Ιεραρχικό Μοντέλο

Μερικοί κόμβοι «πιο» ίσοι από τους άλλους "super-peers" - υπερ-ομότιμοι
Υβριδικό μεταξύ (πλήρως) κατανεμημένου και κεντρικοποιημένου
Παράδειγμα: KaZaa, DNS

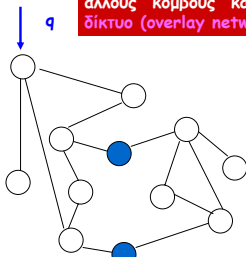
Υπερ-Ομότιμοι

Είτε ορίζονται από την αρχή είτε προάγονται



Η έννοια του Ιδεατού Δικτύου

Κάθε κόμβος συνδέεται με (γνωρίζει) έναν αριθμό από άλλους κόμβους κατασκευάζοντας έτσι ένα ιδεατό δίκτυο (overlay network)



Μια ερώτηση (αίτηση) για κάποιο πόρο υποβάλλεται σε έναν κόμβο του συστήματος

Δραματοποίηση της ερώτησης στο ιδεατό δίκτυο προς τους κόμβους με σχετικά δεδομένα

Πως κατασκευάζουμε το "overlay" και πως δρομολογούμε τις ερωτήσεις σε αυτό

Η έννοια του Ιδεατού Δικτύου

Παρατήρηση

Το ιδεατό δίκτυο κτίζεται από-πάνω (on top) του φυσικού - IP-δικτύου

Δύο γείτονες στο ιδεατό δίκτυο, πολλά hops μακριά στο φυσικό δίκτυο

Δύο Είδη Συστημάτων Ομοτίμων

Δυο φιλοσοφίες:

Αδόμητα (Unstructured):

Καμία υπόθεση για το που τοποθετούνται οι κόμβοι
Κάθε κόμβος συνδέεται «τυχαία» με κάποιους κόμβους του δικτύου

Δομημένα (Structured):

το ιδεατό δίκτυο ακολουθεί «αυστηρή» και συγκεκριμένη δομή, π.χ. Δακτύλιος (CHORD), Grid (CAN)

τα δεδομένα (ή πληροφορία/ευρετήριο) για τα δεδομένα τοποθετούνται σε συγκεκριμένους κόμβους

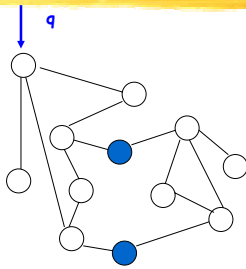
(+) γρήγορη αναζήτηση - συνήθως λογαριθμική στον αριθμό των κόμβων του δικτύου

(-) κόστος συντήρησης του δικτύου και εξισορρόπησης του φορτίου

Δρομολόγηση σε Αδόμητα Συστήματα Ομοτίμων Κόμβων

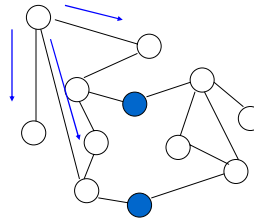
- Δεν υπάρχει κεντρικός εξυπηρετητής
- **Σύνδεση Κόμβου:** Για να συνδεθεί κάποιος κόμβος στο p2p δίκτυο πρέπει να ξέρει τουλάχιστον έναν κόμβο του
- **Αναζήτηση Κόμβου:** Κάποιο είδος μετάδοσης σε όλους (broadcast) Πλημμύρα (flooding)

Δρομολόγηση σε Αδόμετα Συστήματα Ομότιμων Κόμβων

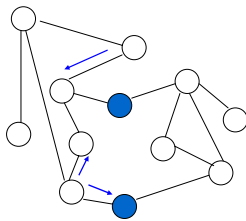


Κάποια μορφή πλημμύρας (**Flooding**):
Κάθε κόμβος προωθεί το μήνυμα στους γείτονες του οι οποίοι με τη σειρά το προωθούν στους γείτονες τους κλπ μέχρι να βρεθεί ο κόμβος με τη πληροφορία

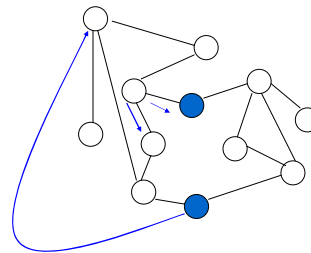
Δρομολόγηση σε Αδόμετα Συστήματα Ομότιμων



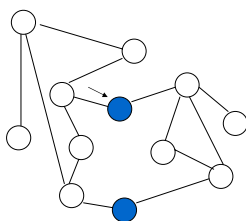
Δρομολόγηση σε Αδόμετα Συστήματα Ομότιμων



Δρομολόγηση σε Αδόμετα Συστήματα Ομότιμων



Δρομολόγηση σε Αδόμετα Συστήματα Ομότιμων



Αδόμετα Συστήματα Ομότιμων Κόμβων

TTL (Time-to-Leave)

Η αναζήτηση περιορίζεται σε ένα TTL αριθμό από βήματα

Είτε **BFS** (Breadth-First-Search) είτε **DFS** (Depth-First-Search)

BFS καλύτερος χρόνος απόκρισης, αλλά πολύ μεγάλος αριθμός από μηνύματα

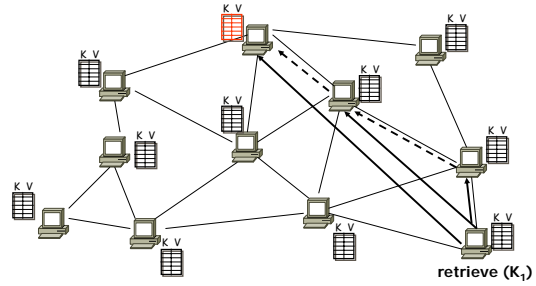
Παρατήρηση: Η αναζήτηση με **BFS** συνεχίζεται σε ένα μονοπάτι ακόμα και αν το αντικείμενο εντοπιστεί σε κάποιο άλλο

Αδόμενα Συστήματα Ομότιμων Κόμβων

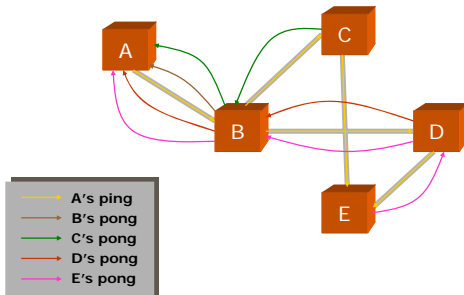
Αναδρομική (Recursive) και Επαναληπτική (Iterative) Αναζήτηση

Αν ο κόμβος που έθεσε την ερώτηση απευθύνει την ερώτηση απευθείας (απευθείας με IP) ή αναδρομικά (κόμβο-κόμβο)
 Το αποτέλεσμα ακολουθεί ή όχι το μονοπάτι της ερώτησης

Αδόμενα Συστήματα Ομότιμων Κόμβων



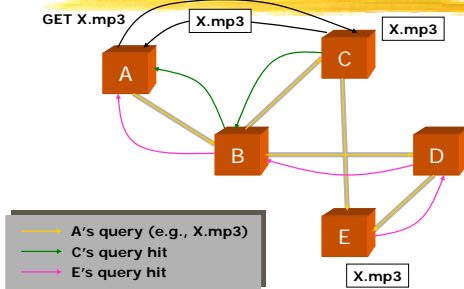
Γnutella: Είσοδος Κόμβου (Ping/Pong)



Γnutella: Είδη Μηνυμάτων

Type	Description	Contained Information
Ping	Announce availability and probe for other servents	None
Pong	Response to a ping	IP address and port# of responding servent; number and total kb of files shared
Query	Search request	Minimum network bandwidth of responding servent; search criteria
QueryHit	Returned by servents that have the requested file	IP address, port# and network bandwidth of responding servent; number of results and result set
Push	File download requests for servents behind a firewall	Servent identifier; index of requested file; IP address and port to send file to

Γnutella: Αναζήτηση (Query/QueryHit/GET)



Γnutella: Είδη Μηνυμάτων

Type	Description	Contained Information
Ping	Announce availability and probe for other servents	None
Pong	Response to a ping	IP address and port# of responding servent; number and total kb of files shared
Query	Search request	Minimum network bandwidth of responding servent; search criteria
QueryHit	Returned by servents that have the requested file	IP address, port# and network bandwidth of responding servent; number of results and result set
Push	File download requests for servents behind a firewall	Servent identifier; index of requested file; IP address and port to send file to

Gnutella

- Δεν υπάρχει κεντρικός εξυπηρετητής
- Για να συνδεθεί κάποιος κόμβος στο p2p δίκτυο πρέπει να ξέρει τουλάχιστον έναν κόμβο του
- Περιορισμένη Μετάδοση σε Όλους
 - Κάθε κόμβος στέλνει τα πακέτα που δεχεται σε όλους τους γείτονες του (συνήθως 4)
 - **time-to-live (TTL)** συνήθως 7
 - Κύκλοι (loops): συνήθως τα πακέτα έχουν μοναδικά ids

«Free-riding» στην Gnutella

- ⌘ 24 ώρες παρακολούθηση:
 - ⊗ 70% των χρηστών της Gnutella δε μοιράζονται **καθόλου αρχεία**
 - ⊗ 50% όλων των απαντήσεων έρχονται από το **top 1%** των κόμβων
 - ⊗ Σημαντικό θέμα «κίνητρα» (incentives) για συμμετοχή

Αδόμητα Συστήματα Ομότιμων Κόμβων

- Οι κόμβοι συνδέονται μεταξύ τους με τυχαίο κόμβο
- Η θέση των δεδομένων (ευρετηρίου ή μετα-δεδομένων) δεν ελέγχεται από το σύστημα
- Δεν υπάρχει εγγύηση για την επιτυχία μιας αναζήτησης (ακόμα και αν το αντικείμενο υπάρχει)
- Δεν υπάρχει κάποιο όριο (bound) στο χρόνο αναζήτησης
- Ο εντοπισμός των **μη δημοφιλών** δεδομένων είναι ιδιαίτερα δύσκολος
- 60% της κίνησης του δικτύου p2p αναζητήσεις

Αδόμητα Συστήματα Ομότιμων Κόμβων

Τροποποίηση

Επικοινωνία μόνο με ένα υποσύνολο των γειτόνων κάθε κόμβου

Ποιο «υποσύνολο»;

- Τυφλές αναζητήσεις
- Αναζητήσεις με πληροφορία (Ευρετήρια!)

Αδόμητα Συστήματα Ομότιμων Κόμβων: Τυχαίοι Περίπατοι

Τυχαίοι Περίπατοι (Random Walks):

Ο κόμβος που ρωτά στέλνει την ερώτηση σε k (με τυχαία επιλογή) γείτονες του

Σε κάθε επόμενο βήμα, η ερώτηση προωθείται από τους γείτονες σε *έναν* μόνο νέο γείτονα (με τυχαία επιλογή)

Δηλαδή, κάθε μία από τις k ερωτήσεις ακολουθεί το δικό της μονοπάτι

Μειώνει τον αριθμό των μηνυμάτων σε $k \times TTL$

Κάποια μορφή εξισορρόπησης φορτίου

Μειώνει τους κύκλους (την επανα-επίσκεψη του ίδιου κόμβου)

Αδόμητα Συστήματα Ομότιμων Κόμβων: Ευρετήρια Δρομολόγησης

Ευρετήρια Δρομολόγησης

Κάθε κόμβος κρατά ένα

Τοπικό ευρετήριο: για τα δεδομένα που είναι αποθηκευμένα σε αυτόν

Ευρετήρια Δρομολόγησης:

ένα για κάθε ακμή - για τα δεδομένα που είναι προσπελάσιμα σε μονοπάτια που ξεκινούν από αυτή την ακμή

Αδόμητα Συστήματα Ομότιμων Κόμβων: Ευρετήρια Δρομολόγησης

Ευρετήρια Δρομολόγησης

Τι κρατάμε στο ευρετήριο;

- Είδος πληροφορίας (πχ ανεστραμμένα αρχεία)
- Πόσοι κόμβοι (ορίζοντας)
 - εκθετική μείωση με την απόσταση

Δομημένα Συστήματα Ομότιμων Κόμβων

Τα δεδομένα (ή το ευρετήριο) δεν τοποθετείται σε «τυχαίους» κόμβους

Αντιστοιχία των δεδομένων (του id τους, π.χ. του ονόματος ενός αρχείου) σε ένα συγκεκριμένο κόμβο του συστήματος
id δεδομένου → id κόμβου

Πως; Με μια συνάρτηση κατακερματισμού (hash function)
Distributed Hash Tables (DHT)

Αναζήτηση σε DHT

1. Lookup(id-δεδομένου) μας δίνει το id του κόμβου που έχει πληροφορία για το δεδομένο
2. ΔΡΟΜΟΛΟΓΗΣΗ προς αυτόν τον κόμβο (βασική αρχή της δρομολόγησης: σε κάθε βήμα μειώνουμε την απόσταση προς τον κόμβο)

Δομημένα Συστήματα Ομότιμων Κόμβων

Θα δούμε δύο διάσημα DHT

- CAN
- CHORD

CAN

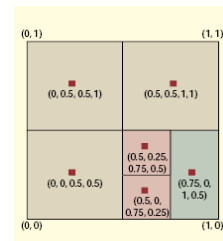
▪ Εικονικό σύστημα συντεταγμένων d διαστάσεων οργανωμένο σε d-torus

▪ Ο χώρος συντεταγμένων διαμοιράζεται δυναμικά σε υπερ-ορθογώνια που καλούνται **ζώνες (zones)**

▪ Σε κάθε κόμβο του συστήματος ανατίθεται και μία ζώνη και κάθε κόμβος χαρακτηρίζεται από τις συντεταγμένες της ζώνης που καλύπτει

$((x_1, y_1), (x_2, y_2))$ κάτω αριστερά και πάνω δεξιά σημεία

Παράδειγμα 2-διάστατου $[0, 1] \times [0, 1]$ CAN με 5 κόμβους



CAN

▪ Κάθε κλειδί K ενός δεδομένου (π.χ. το όνομα K του αρχείου V) αντιστοιχίζεται σε ένα σημείο P του χώρου χρησιμοποιώντας μια ομοιόμορφη συνάρτηση κατακερματισμού και αποθηκεύεται στον κόμβο του οποίου η ζώνη περιλαμβάνει το P

▪ **Αναζήτηση** του (K, V): εφαρμογή της ίδιας συνάρτησης κατακερματισμού στο (K, V) οπότε προκύπτει το P δηλαδή ο κόμβος που διατηρεί το (K, V).

(Βήμα 2: δρομολόγηση) Πως θα φτάσουμε στον P από τον κόμβο που κάναμε την ερώτηση;

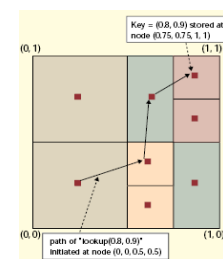
CAN

Δρομολόγηση Ερώτησης

▪ Κάθε κόμβος διατηρεί έναν πίνακα δρομολόγησης με τις IP διευθύνσεις και τις ζώνες που καλύπτει κάθε άμεσος γείτονας του στο χώρο συντεταγμένων

▪ Η ερώτηση δρομολογείται κάθε φορά προς το γείτονα που είναι πιο κοντά στον κόμβο που αποθηκεύει το κλειδί

σε κάθε βήμα, ο κόμβος ελέγχει τις συντεταγμένες των γειτόνων του και προωθεί την ερώτηση στο γείτονα του οποίου η (π.χ. ευκλείδεια) απόσταση από το που είναι πιο κοντά στον κόμβο P που έχει το δεδομένο

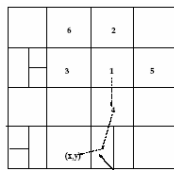


▪ Για χώρο διάστασης d χωρισμένο σε n ίσες ζώνες κάθε κόμβος έχει $2d$ γείτονες και το μέσο μήκος ενός μονοπατιού είναι $(d/4) \times n^{1/d}$

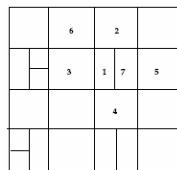
▪ Κάθε κόμβος αποθηκεύει μικρό αριθμό από πληροφορίες (μόνο για τους γείτονες του) ανεξάρτητο από το συνολικό αριθμό κόμβων του δικτύου

Είσοδος Κόμβου στο Δίκτυο

- Ο καινούργιος κόμβος επιλέγει *τυχαία* ένα σημείο P
- Υποθέτοντας ότι υπάρχει τουλάχιστον ένας κόμβος, στέλνει ένα μήνυμα συνένωσης στον κόμβο στον οποίο ανήκει το P .
- Το μήνυμα συνένωσης *δρομολογείται* στο κόμβο που κατέχει το P .
- Όταν το μήνυμα φτάσει στον ιδιοκτήτη του P , ο κόμβος διαιρεί τον εαυτό του στα δύο κατά μήκος μιας από τις διαστάσεις
- Ενημέρωση δομών (γείτονες) και δεδομένα που κατέχει

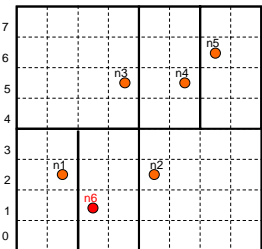
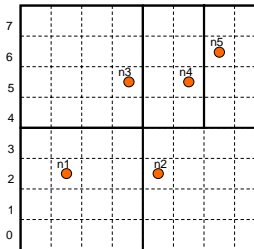


ο αριθμός των γειτόνων του κόμβου είναι 1 σε ποσοστό $(k-2)$
 If's coordinate neighbor set = {2,3,4,5}
 If's coordinate neighbor set = {}



If's coordinate neighbor set = {2,3,4,7}
 If's coordinate neighbor set = {1,2,4,5}

Εισαγωγή κόμβου n_6



Διαγραφή, Ανάκαμψη και Συντήρηση

- **Εθελοντική αποχώρηση:** ο κόμβος παραδίδει τη ζώνη του και τα δεδομένα (K,V) σε ένα γείτονα
- **Αποτυχία δικτύου:** Μη-προσβάσιμοι κόμβοι θέτουν σε λειτουργία ένα αλγόριθμο άμεσης ανακατάληψης που δεσμεύει τη περιοχή του κόμβου στο γείτονα
 - Ανίχνευση αποτυχίας μέσω περιοδικών μηνυμάτων
 - Ανακατάληψη της ζώνης από το γείτονα με τη μικρότερη ζώνη

Θέματα εξισορρόπησης φορτίου

Οι *κόμβοι* τοποθετούνται σε έναν ιδεατό δακτύλιο 2^n θέσεων μέσω μιας συνάρτησης κατακερματισμού που εφαρμόζεται στο *id* του κόμβου

Τα *δεδομένα* τοποθετούνται στους κόμβους εφαρμόζοντας μια συνάρτηση κατακερματισμού στο κλειδί τους

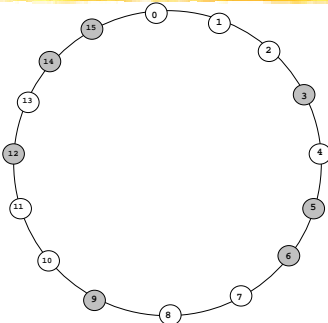
Αναζήτηση:

1. Εφαρμόζεται η συνάρτηση κατακερματισμού στον κλειδί του δεδομένου που ψάχνουμε
2. Η ερώτηση δρομολογείται προς τον κόμβο που έχει το κλειδί

- Οι κόμβοι είναι τοποθετημένοι σε έναν νοητό δακτύλιο.
- Συνάρτηση κατακερματισμού εφαρμόζεται σε κάθε κόμβο (στην IP διεύθυνση του)
- Δίνει έναν αριθμό από 0 έως $2^m - 1$

Παράδειγμα

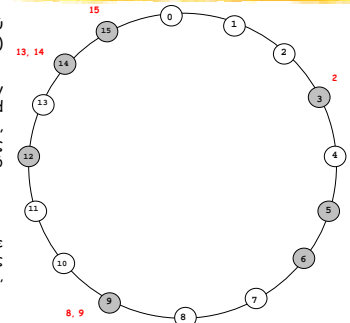
Δακτύλιος με $m = 4$ και 16 θέσεις
 Κόμβοι υπάρχουν στις θέσεις 3, 5, 6, 9, 12, 14 και 15
 δηλαδή, δεν υπάρχουν κόμβοι σε όλες τις ιδεατές θέσεις του δακτύλιου



- Συνάρτηση κατακερματισμού εφαρμόζεται στο κλειδί K (id) κάθε δεδομένου
- Το K αποθηκεύεται στον πρώτο κόμβο του οποίου το id είναι ίσο ή μεγαλύτερο του K , $successor(K)$ (πρώτος κόμβος με τη φορά του ρολογιού από το K)

Παράδειγμα

Έχουν εισαχθεί τα δεδομένα με κλειδί (μετά την εφαρμογή της συνάρτησης κατακερματισμού) 2, 8, 9, 13, και 14.



Αναζήτηση

Για τη δρομολόγηση, αρκεί κάθε κόμβος να ξέρει μόνο τους γείτονες του, αλλά τότε $O(N)$ μηνύματα

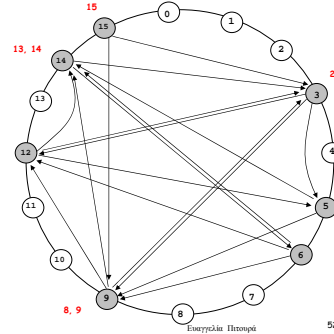
Βελτίωση

Κάθε κόμβος εκτός από τον άμεσο γείτονα του στο δακτύλιο συνδέεται και με τους κόμβους που βρίσκονται σε απόσταση $2^0, 2^1, 2^2, \dots, 2^{m-1}$ (Όλες οι πράξεις γίνονται mod 2^m).

Συγκεκριμένα, συνδέεται με τον πρώτο κόμβο που συναντά σε τουλάχιστον τόση απόσταση.

Ο πρώτος κόμβος που ακολουθεί τον κόμβο n σε απόσταση τουλάχιστον 2^i , συμβολίζεται ως $successor(n + 2^i)$.

Σημείωση: Στο CAN, κάθε κόμβος συνδέεται με d άλλους κόμβους, όπου d η διάσταση του CAN.



Παράδειγμα, ο κόμβος 12 συνδέεται με τους:

$successor(12 + 2^0) \bmod 16 = 14$ (πρώτος που υπάρχει μετά τον 13)

$successor(12 + 2^1) \bmod 16 = 14$

$successor(12 + 2^2) \bmod 16 = 3$ (πρώτος που υπάρχει μετά τον 0)

$successor(12 + 2^3) \bmod 16 = 5$ (πρώτος που υπάρχει μετά τον 4)

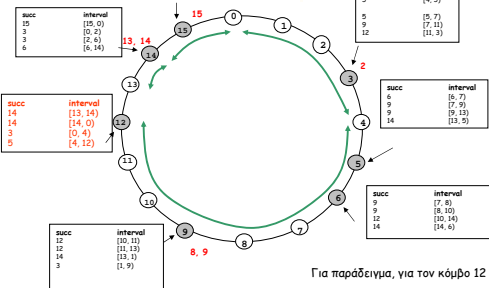
Διαισθητικά, επιτρέπει άλματα σε μακρινούς κόμβους

finger table: Η πληροφορία με τους γείτονες (successors) ενός κόμβου σώζεται σε έναν πίνακα m θέσεων ανά κόμβο

Επίσης, αποθηκεύουμε στον πίνακα (για ευκολία) και ποια περιοχή (διάστημα στο δακτύλιο) καλύπτει κάθε successor.

Η i -οστή εγγραφή του πίνακα του κόμβου n ($n.finger(i).interval$) αφορά αυτό το διάστημα που καλύπτεται από τον i -στο successor του κόμβου n , συγκεκριμένα το διάστημα:

$[(n + 2^{i-1}) \bmod 2^m, (n + 2^i) \bmod 2^m)$.

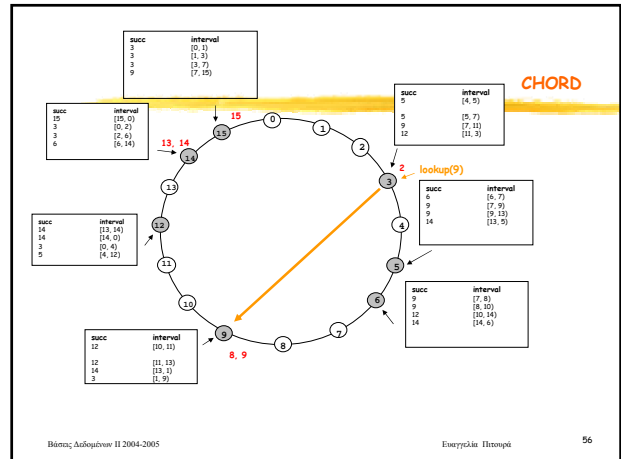


Για παράδειγμα, για τον κόμβο 12

CHORD

Για την αναζήτηση ενός κλειδιού K , ο κόμβος n ψάχνει στο finger table για τον κόμβο P που είναι ο αμέσως προηγούμενος του K - Ο successor του P θα είναι ο κόμβος που είναι αποθηκευμένο το δεδομένο με κλειδί K .

Για να το βρει τον P , κάθε κόμβος, κοιτά στο finger table του για να βρει το interval στο οποίο ανήκει το K και στη συνέχεια επικοινωνεί (στέλνει μήνυμα) στον κόμβο που καλύπτει το interval (δηλαδή) στο αντίστοιχο succ.



CHORD

Εισαγωγή Κόμβου n στο Δίκτυο

Ο n επικοινωνεί με έναν τυχαίο κόμβο n' και η ερώτηση δρομολογείται στη θέση που πρέπει να εισαχθεί ο n (δηλαδή, στον αμέσως προηγούμενο κόμβο)

Βήμα 1: Κατασκευή του finger table του n

Lookup($n + 2^0$), Lookup($n + 2^1$), ..., Lookup($n + 2^{m-1}$)

Κόστος $O(m \log(N))$

Βελτιστοποίηση:

Έλεγχος αν ίδιοι successors

Πρακτική βελτιστοποίηση:

Ζητά από έναν άμεσο γείτονα του το δικό του finger table

CHORD

Εισαγωγή Κόμβου n στο Δίκτυο (συνέχεια)

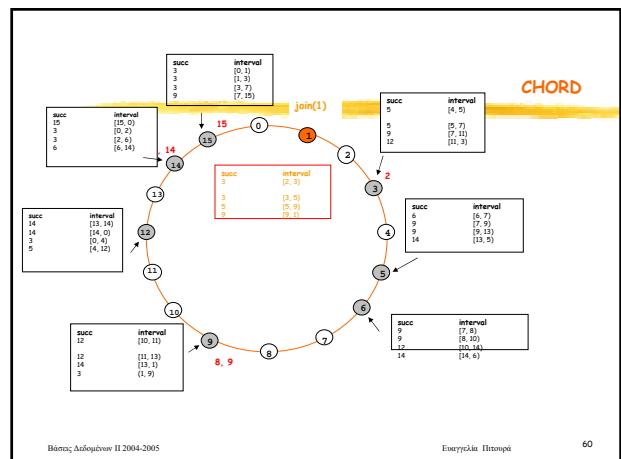
Βήμα 2: Τροποποίηση των finger tables κόμβων που ήδη υπάρχουν

Θα αλλάξουμε τον i -στό finger ενός κόμβου p αν:

- (1) ο p προηγείται του n τουλάχιστον κατά 2^{i-1}
- (2) το i -στό finger του p έπεται του n (π.χ.: lookup($n - 2^i$) κλπ)

Βήμα 3: Μετακίνηση περιεχομένου

Μπορεί να πάρει μόνο δεδομένα που το ακολουθούν



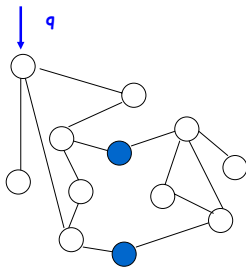
CHORD

- Αναζήτηση $O(\log N)$
- Κάθε κόμβος αποθηκεύει πληροφορία για $O(\log N)$ άλλους κόμβους
- Εισαγωγή/διαγραφή κόμβου $O(\log^2 N)$

Συγκριτικός Πίνακας

	Paradigm	Search Type	Search Cost (messages)	Autonomy
Gnutella	Breadth-first search on graph	String comparison	$2 * \sum_{i=0}^{m-1} C * (C-1)^i$	very high
FreeNet	Depth-first search on graph	String comparison	$O(\log n) ?$	very high
Chord	Implicit binary search trees	Equality	$O(\log n)$	restricted
CAN	d-dimensional space	Equality	$O(d * n^{(1/d)})$	high
P-Grid	Binary prefix trees	Prefix	$O(\log n)$	high

Το Πρόβλημα της Αναζήτησης



Παρατήρηση
 Η ερώτηση συνήθως το «όνομα» κάποιου αρχείου ή κάποιες λέξεις κλειδιά, αλλά

SQL ερωτήματα (Join, Range queries (ερωτήσεις διαστήματος))
 XML αρχεία

Συγκριτικός Πίνακας

Περισσότερα για το πρόβλημα της Αναζήτησης

- ☑ Query Expressiveness - Εκφραστικότητα (είδη ερωτήσεων)
- ☑ Comprehensiveness - Πληρότητα (όλα τα διαθέσιμα αποτελέσματα ή απλώς τα πρώτα κ ή τα καλύτερα κ αποτελέσματα)
- ☑ Τοπολογία
- ☑ Τοποθέτηση Δεδομένων
- ☑ Δρομολόγηση Μηνυμάτων

Comparison

	Gnutella	CAN	Others?
Expressivness	****		
Comprehensivness	**		
Autonomy	****		
Efficiency	*		
Robustness	***		
Topology	pwr law		
Data Placement	arbitrary		
Message Routing	flooding		

Comparison

	Gnutella	CAN	Others?
Expressivness	****	*	
Comprehensivness	**	****	
Autonomy	****	**	
Efficiency	*	***	
Robustness	***	**	
Topology	pwr law	grid	
Data Placement	arbitrary	hashing	
Message Routing	flooding	directed	

Άλλα Θέματα

- **Σημασιολογικά Ιδεατά Δίκτυα** Οι κόμβοι επιλέγουν τους γείτονές τους με βάση το περιεχόμενο τους ή τα ενδιαφέροντά τους
- **Αντίγραφα/Caching**

Καταμεμημένες Βάσεις Δεδομένων vs ΣΟΚ (p2p)

- ⌘ Κατανομή των δεδομένων
 - ☒ Καταμεμημένες ΒΔ: οριζόντιος ή κάθετος καταμερισμός σχέσεων με βάση τη συχνότητα των ερωτήσεων
 - ☒ Παράλληλες ΒΔ: RR, διάστημα, κατακερματισμός
 - ☒ Δε γνωρίζουμε το σύνολο των κόμβων (ούτε το σχήμα των δεδομένων)
- ⌘ Διαφάνεια Κατανομής (Distribution Transparency)
- ⌘ Τεμαχισμός και Αντίγραφα (Data Allocation and Fragmentation)
- ⌘ Επεξεργασία SQL Ερωτήσεων
- ⌘ Δοσοληψίες