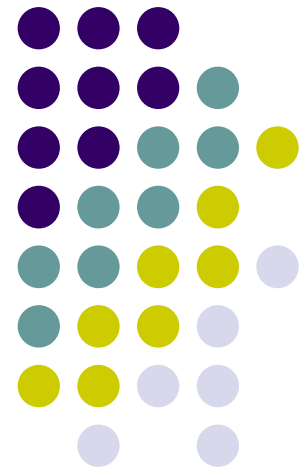


Κατανεμημένα Συστήματα

Μάθημα #2





Περιεχόμενα

- Υπολογιστικά Συστήματα πολλών CPUs
- Λειτουργικά Συστήματα Υπολογιστικών Συστημάτων
- Ενδιάμεσο Λογισμικό (Middleware)
- Οργάνωση Κατανεμημένων Συστημάτων
 - Το Μοντέλο Πελάτη-Διακομιστή
 - Ομότιμα Συστήματα (P2P Systems)

Υπολογιστικά Συστήματα πολλών CPUs



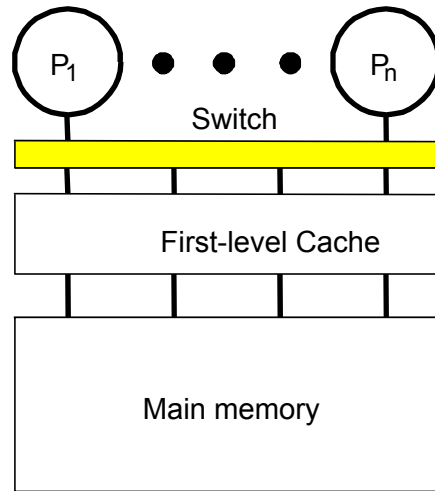
Συστήματα τα οποία διαθέτουν κοινό χώρο διευθύνσεων (shared data space):

- Πολυεπεξεργαστές – Συστήματα Κοινής Μνήμης (Shared-memory systems - **multiprocessors**)

Συστήματα τα οποία υποστηρίζουν πέρασμα μηνυμάτων:

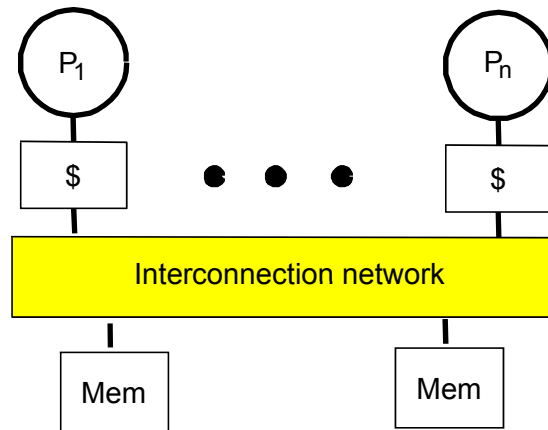
- Ομοιογενή πολυ-υπολογιστικά συστήματα
 - * Πολυυπολογιστές - Συστήματα Κατανεμημένης Μνήμης (Distributed-memory systems - multicomputers)
 - * Clusters (Networks of Workstations (NoWs) and Beowulf Clusters)
- Ετερογενή πολυ-υπολογιστικά συστήματα

Πολυεπεξεργαστές



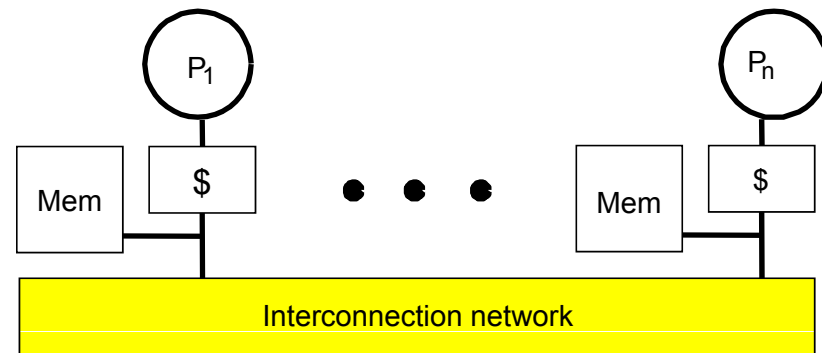
Shared Cache

Scale →

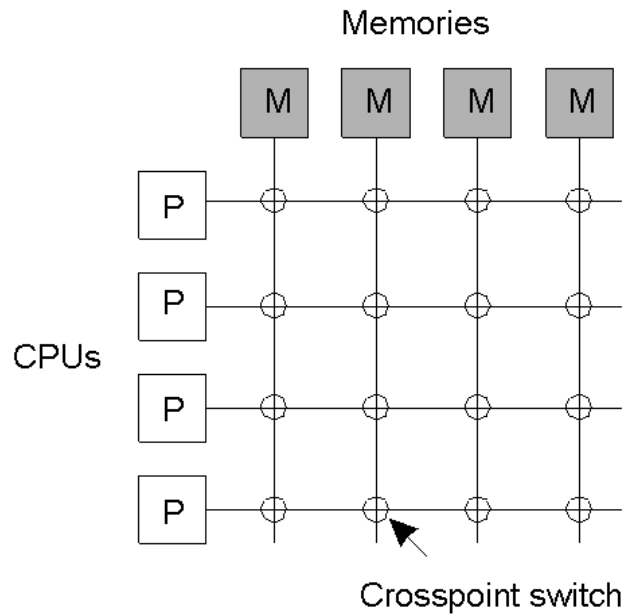


**Centralized Memory
(Uniform Memory Access -UMA)**

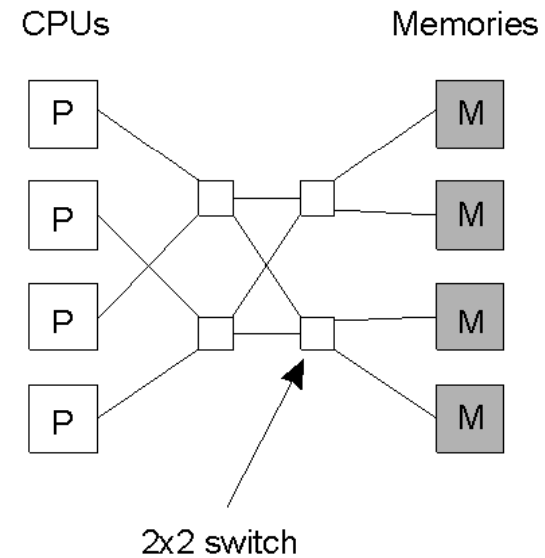
**Distributed Memory
(Non Uniform Memory Access - UMA)**



Πολυεξεργαστές



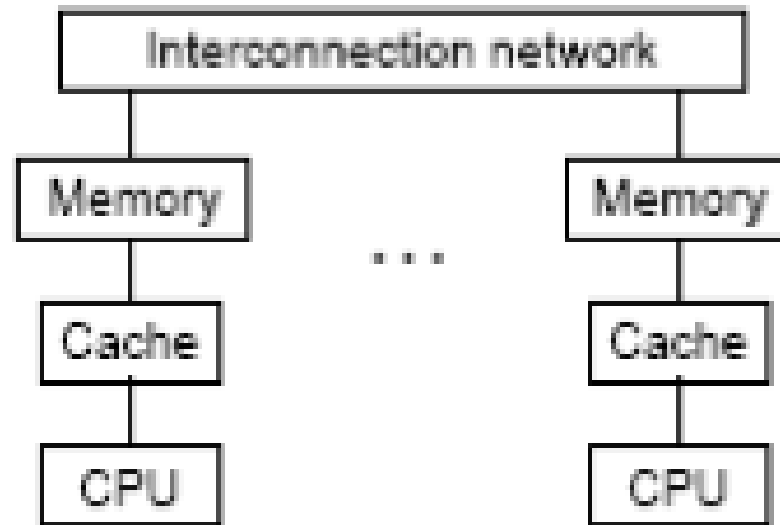
(a)



(b)

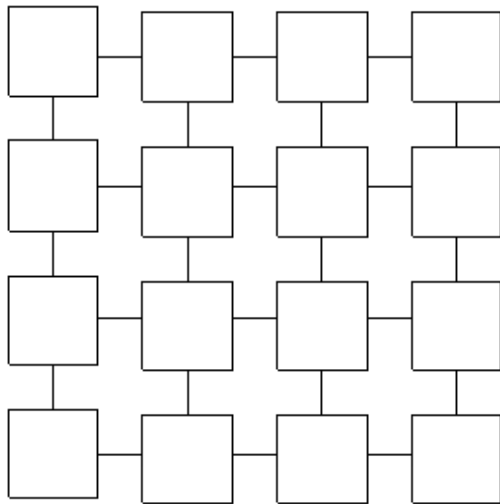
- a) A crossbar switch
- b) An omega switching network

Ομοιογενή Πολυ-υπολογιστικά Συστήματα

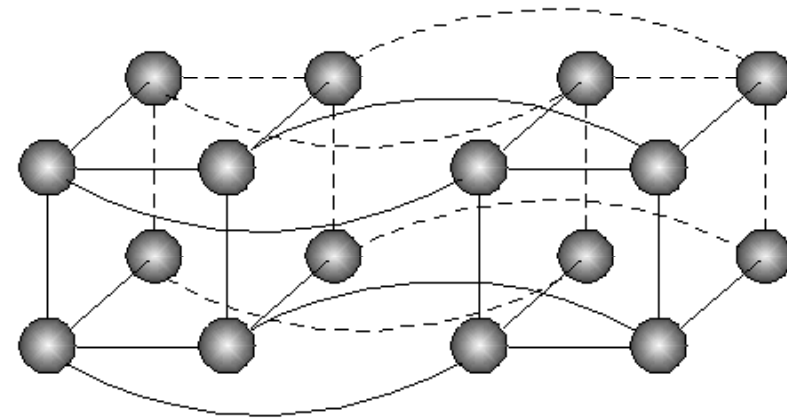


Structure of distributed-memory machines.

Πολυπολογιστές - Συστήματα Κατανεμημένης Μνήμης



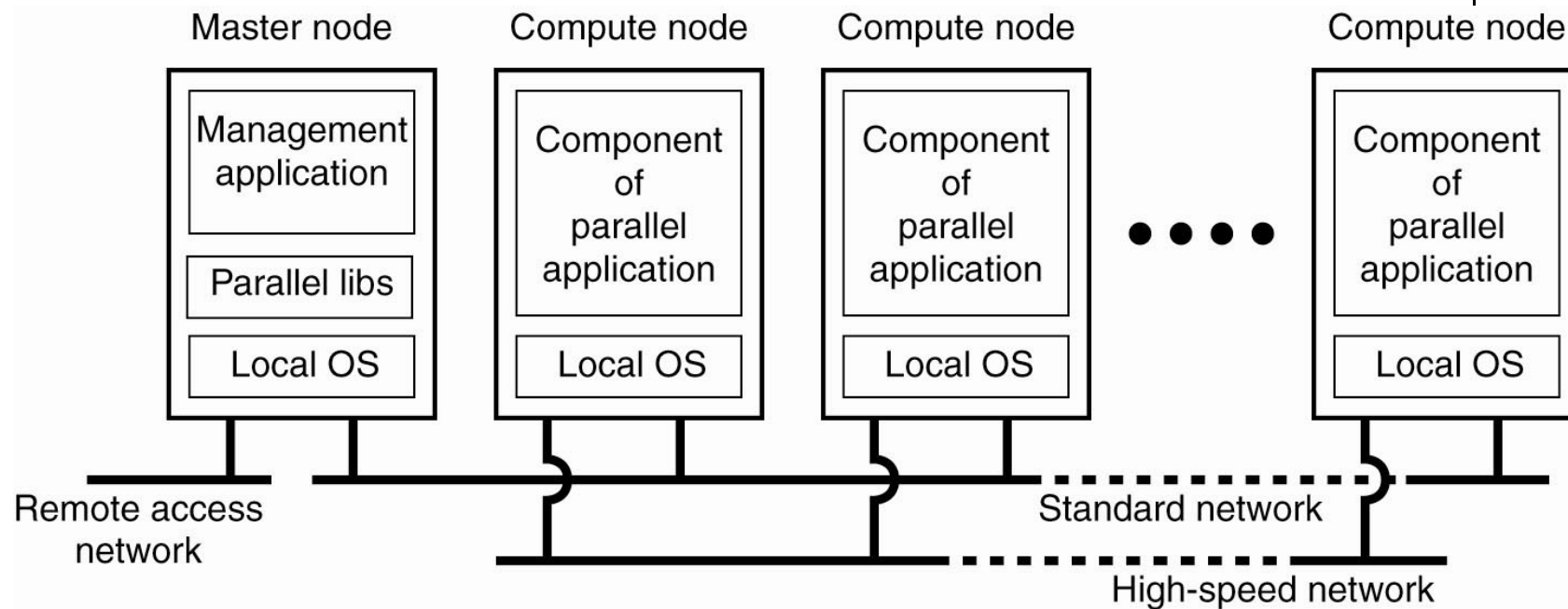
(a)



(b)

- a) Grid
- b) Hypercube

Clusters



An example of a cluster computing system.

Ετερογενή Πολυ-υπολογιστικά Συστήματα



- Οι υπολογιστές που αποτελούν μέρη του κατακευματισμένου συστήματος μπορεί να διαφέρουν (μονοεπεξεργαστικά συστήματα, πολυεπεξεργαστές, ομοιογενείς πολυ-υπολογιστές).
- Το δίκτυο διασύνδεσης πολύ ετερογενές (π.χ., τοπικά δίκτυα τα οποία συνδέονται με ένα δικτυακό κορμό υψηλής ταχύτητας).
- Λόγω της κλίμακας μεγέθους των ετερογενών πολυ-υπολογιστών, της εγγενούς ετερογένειάς τους και της έλλειψης μιας καθολικής άποψης του συστήματος, η δημιουργία εφαρμογών για αυτά τα συστήματα απαιτεί εξελιγμένο λογισμικό.

Λειτουργικά Συστήματα Υπολογιστικών Συστημάτων



- **Μονοεπεξεργαστικά λειτουργικά συστήματα**
Το λειτουργικό σύστημα υλοποιεί μια εικονική μηχανή που παρέχει στις εφαρμογές μηχανισμούς πολυδιεργασίας (multitasking).
- **Πολυεπεξεργαστικά λειτουργικά συστήματα:** χρησιμοποιούνται στους πολυεπεξεργαστές
Κύρια διαφορά από τα μονοεπεξεργαστικά: τα δεδομένα πρέπει να προστατεύονται από την ταυτόχρονη προσπέλαση
- **Κατανεμημένα λειτουργικά συστήματα** (Distributed operating systems – DOS) ή **Πολυ-υπολογιστικά** λειτουργικά συστήματα: χρησιμοποιούνται στα ομοιογενή πολυ-υπολογιστικά συστήματα
- **Δικτυακά λειτουργικά συστήματα** (Network operating systems – NOS)

Κατανεμημένα λειτουργικά συστήματα ή Πολυ-υπολογιστικά λειτουργικά συστήματα



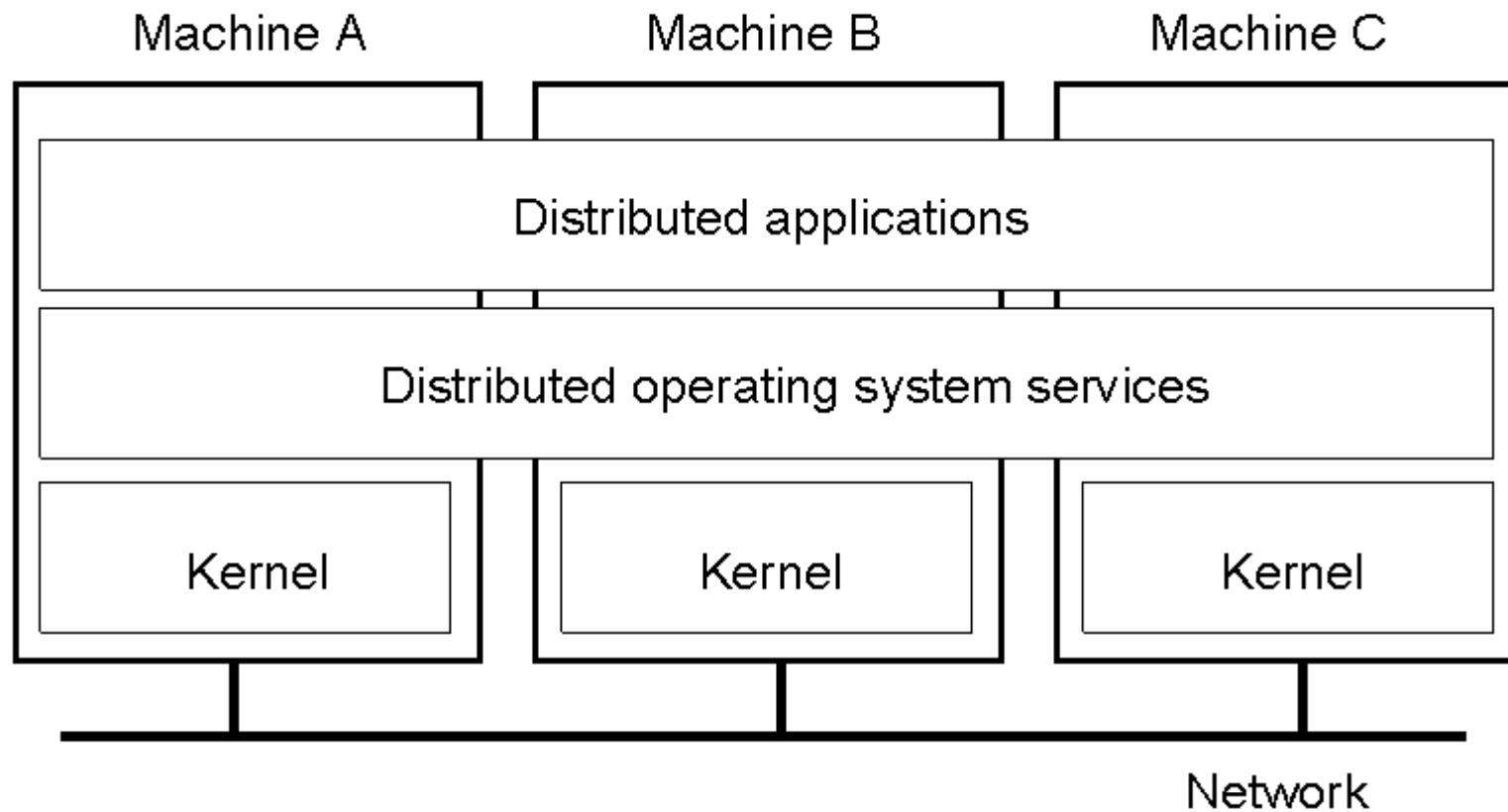
- Διαχειρίζονται το υποκείμενο υλικό ως ενιαίο σύστημα.
- Παρέχουν στις εφαρμογές μηχανισμούς μεταβίβασης μηνυμάτων
- Κάθε κόμβος διαθέτει πυρήνα με υπομονάδες για τη διαχείριση τοπικών πόρων (μνήμη, δίσκους, cpu) και τη διαχείριση της επικοινωνίας με άλλους κόμβους
- Πάνω από κάθε τοπικό πυρήνα, κοινό επίπεδο λογισμικού υλοποιεί το λειτουργικό σύστημα ως εικονική μηχανή που υποστηρίζει ταυτόχρονη εκτέλεση εργασιών

Στο επίπεδο αυτό υλοποιούνται μηχανισμοί όπως: ανάθεση εργασίας σε επεξεργαστή, επικοινωνία μεταξύ διεργασιών, συγκάλυψη αστοχιών λογισμικού, διαφανής αποθήκευση.

Το επίπεδο αυτό μπορεί να παρέχει ένα αφηρημένο μοντέλο

πολυεπεξεργαστή.

Γενική δομή πολυ-υπολογιστικού λειτουργικού συστήματος



Δικτυακά λειτουργικά συστήματα



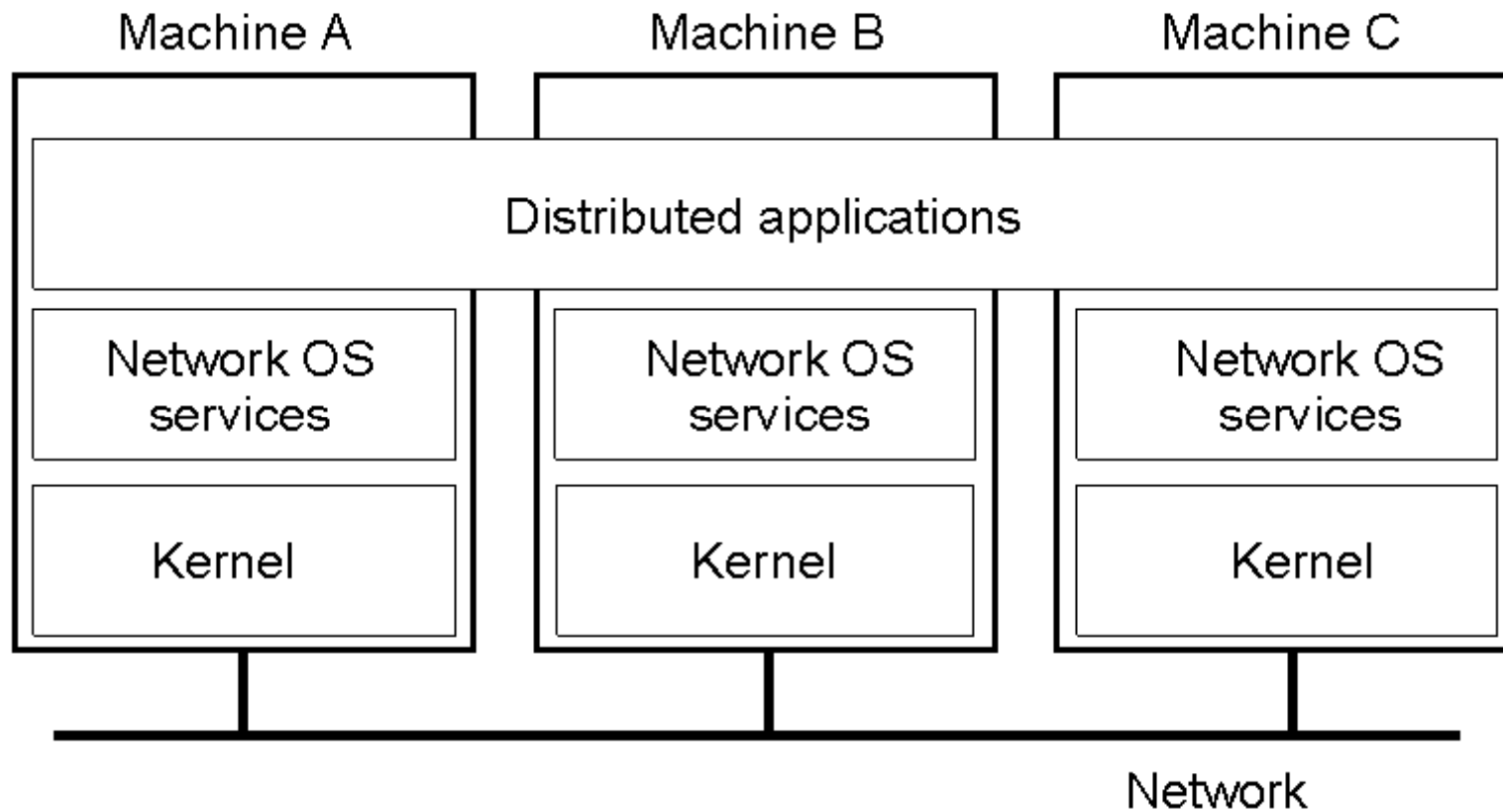
- Οικοδομούνται από σύνολο συστημάτων, καθένα με το δικό του λειτουργικό σύστημα.
- Οι μηχανές και τα λειτουργικά τους συστήματα μπορεί να διαφέρουν.
- Παρέχουν μηχανισμούς που επιτρέπουν στους χρήστες μιας μηχανής να χρησιμοποιούν τις υπηρεσίες που είναι διαθέσιμες σε άλλη μηχανή.
- Έλλειψη διαφάνειας – Άδεια πρόσβασης για κάθε μηχανή
- Επεκτασιμότητα: μία νέα μηχανή συνδέεται στο δίκτυο και γνωστοποιείται η παρουσία της στις άλλες μηχανές

Παραδείγματα υπηρεσιών:

rlogin μηχανή - *η επιλογή της μηχανής γίνεται ρητά από τον χρήστη*

rcp μηχανή1:αρχείο1 μηχανή2:αρχείο2 - *ο χρήστης πρέπει να είναι ενημερωμένος για τη θέση των αρχείων*

Γενική δομή δικτυακού λειτουργικού συστήματος



Ενδιάμεσο Λογισμικό (Middleware)

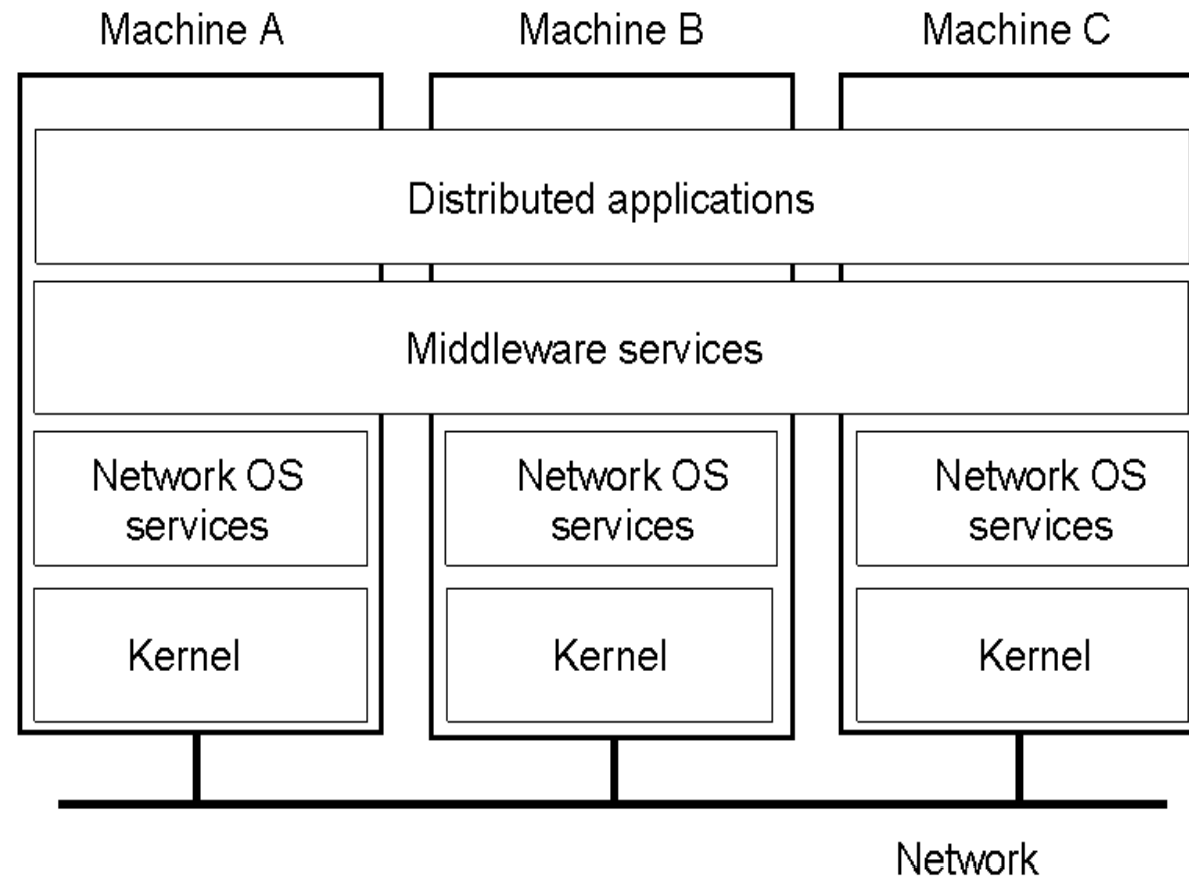


Είναι δυνατόν να δημιουργηθεί κατανεμημένο σύστημα το οποίο θα συνδυάζει τη διαφάνεια και σχετική ευχρηστία των κατανεμημένων λειτουργικών συστημάτων (DOS) με την επεκτασιμότητα και την ανοιχτή λειτουργία των δικτυακών (NOS)?

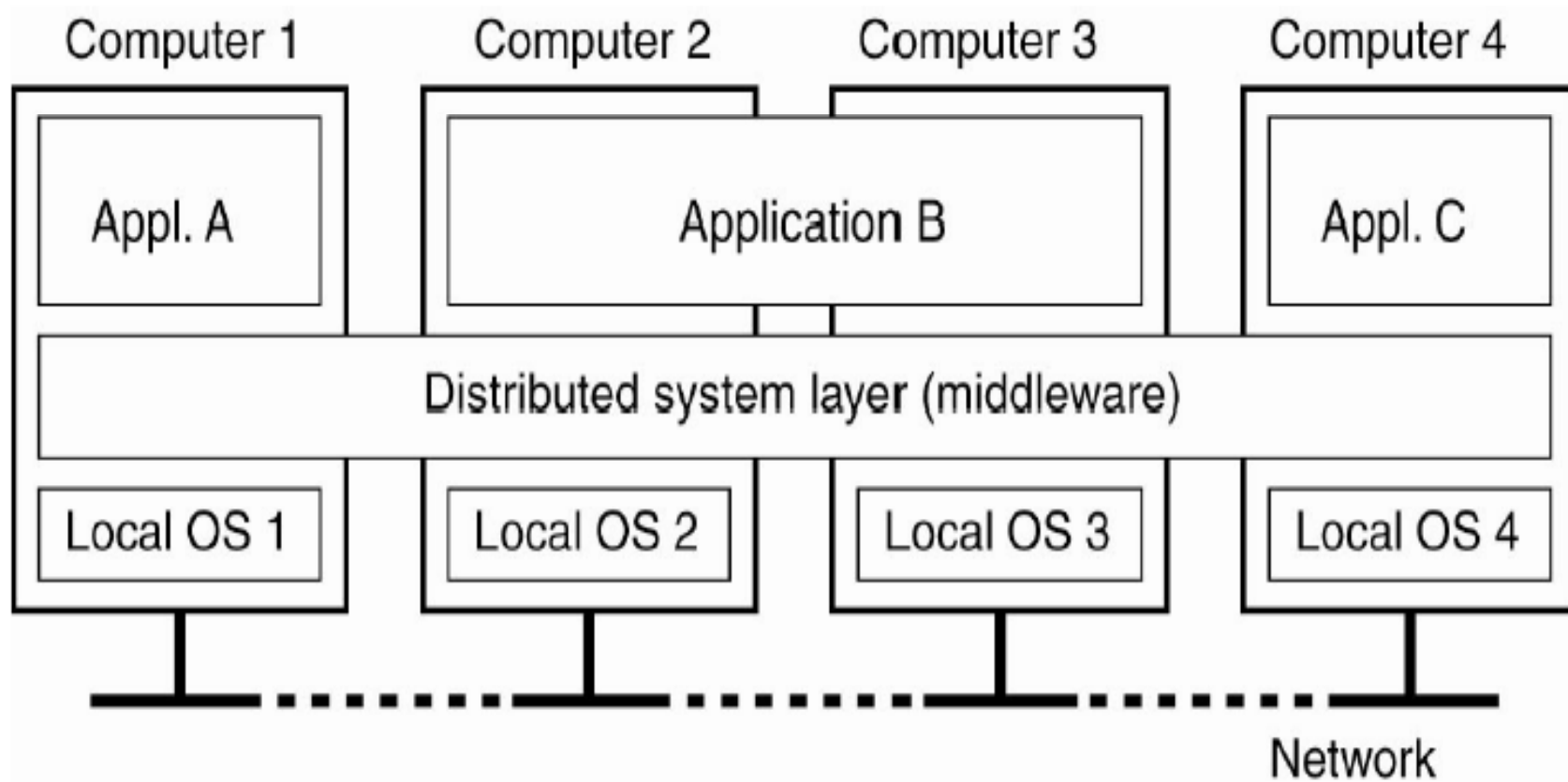
Ενδιάμεσο Λογισμικό (middleware): NOS + πρόσθετο επίπεδο λογισμικού

Έτσι, κρύβεται η ετερογένεια των μηχανών αφού απομονώνονται οι εφαρμογές από το υλικό και εξασφαλίζεται η διαφάνεια της κατανεμημένης λειτουργίας

Γενική δομή κατανεμημένου συστήματος με τη μορφή Ενδιάμεσου Λογισμικού



A distributed system organized as middleware.
The middleware layer extends over multiple machines,
and offers each application the same interface.



An overview of DOS (Distributed Operating Systems) NOS (Network Operating Systems) Middleware



System	Description	Main Goal
DOS	Tightly-coupled operating system for multi-processors and homogeneous multicomputers	Hide and manage hardware resources
NOS	Loosely-coupled operating system for heterogeneous multicomputers (LAN and WAN)	Offer local services to remote clients
Middleware	Additional layer atop of NOS implementing general-purpose services	Provide distribution transparency

Μοντέλα ενδιάμεσου λογισμικού



- **Ενδιάμεσο λογισμικό που βασίζεται στα κατανεμημένα συστήματα αρχείων:**

Η επικοινωνία ανάγεται στην προσπέλαση των διαμοιραζόμενων αρχείων.

Η διαφάνεια της κατανεμημένης λειτουργίας περιορίζεται στα αρχεία (π.χ. απαιτείται η ρητή εκκίνηση διεργασιών σε συγκεκριμένες μηχανές).

- **Ενδιάμεσο λογισμικό που βασίζεται στις Κλήσεις Απομακρυσμένων Διαδικασιών (Remote Procedure Calls, RPC):**

Μια διεργασία μπορεί να καλεί διαδικασία που υλοποιείται σε απομακρυσμένη μηχανή.

Η καλούσα διεργασία δεν αντιλαμβάνεται ότι πραγματοποιήθηκε επικοινωνία μέσω δικτύου (πέρα από κάποια απώλεια απόδοσης)

- **Ενδιάμεσο λογισμικό που διαθέτει την έννοια του κατανεμημένου αντικειμένου:**

Κάθε αντικείμενο τοποθετείται σε μία μηχανή και η διασύνδεσή του (μέθοδοι που υλοποιεί) γίνεται διαθέσιμη σε πολλές άλλες μηχανές. Όταν μια διεργασία καλεί μια μέθοδο, η υλοποίηση της διασύνδεσης στη μηχανή όπου εκτελείται η διεργασία μετασχηματίζει την κλήση της μεθόδου σε μήνυμα στο αντικείμενο. Το αντικείμενο εκτελεί τη μέθοδο και στέλνει πίσω το αποτέλεσμα.



Μοντέλα ενδιάμεσου λογισμικού

Παγκόσμιος Ιστός: Χρήση του **μοντέλου κατανεμημένων εγγράφων** ήτοι οι πληροφορίες είναι οργανωμένες ως έγγραφα τα οποία είναι αποθηκευμένα σε ένα μηχανήμα κάπου στον κόσμο. Τα έγγραφα περιέχουν συνδέσμους τους οποίους αν ακολουθήσουμε, τα έγγραφα στα οποία αυτοί παραπέμπουν, προσκομίζονται στην οθόνη μας

Υπηρεσίες ενδιάμεσου λογισμικού



- Υπηρεσίες διαφανούς προσπέλασης μέσω της παροχής υψηλού επιπέδου μηχανισμών επικοινωνίας (π.χ. RPCs).
- Υπηρεσίες ονομασίας που επιτρέπουν την κοινή χρήση και αναζήτηση οντοτήτων και λαμβάνουν υπόψη θέματα επεκτασιμότητας και σταθερότητας των θέσεων των οντοτήτων.
- Μηχανισμοί αποθήκευσης ή διατήρησης (persistence mechanisms) (π.χ., κατανεμημένα συστήματα αρχείων, ενσωματωμένες Β.Δ., σύνδεση εφαρμογών με Β.Δ.).
- Υπηρεσίες κατανεμημένων συναλλαγών που εξασφαλίζουν την **ατομικότητα** (η συναλλαγή είτε επιτυγχάνει στο σύνολό της ή τα σχετικά δεδομένα μένουν ανεπηρέαστα).
- Υπηρεσίες ασφάλειας σε συνδυασμό με την ανάγκη για επεκτασιμότητα. Το ενδιάμεσο λογισμικό δε μπορεί να στηρίζεται στους μηχανισμούς ασφάλειας των υποκείμενων Λ.Σ.

Υπηρεσία Ονομάτων



- Το κατανεμημένο σύστημα πρέπει να υποστηρίζει *υπηρεσία ονομάτων (naming service)*, ώστε οι διεργασίες του να καθορίζουν τους πόρους με τους οποίους θέλουν να επικοινωνούν
- Η υπηρεσία ονομάτων
 - υλοποιεί έναν κοινό χώρο ονομάτων για ολόκληρο το σύστημα και μεταφράζει τα ονόματα των πόρων σε διευθύνσεις,
 - είναι ανάλογη με την υπηρεσία ευρετηρίων των συστημάτων αρχείων που επιτρέπει τη μετάφραση του ονόματος ενός αρχείου σε μια δομή δεδομένων που περιγράφει πως και που αποθηκεύεται το αρχείο σε μια συσκευή αποθήκευσης.
- Η υπηρεσία ονομάτων είναι επαρκής όταν όλες οι διεργασίες γνωρίζουν τα ονόματα των πόρων που θέλουν να χρησιμοποιήσουν.



Υπηρεσία αρχείου

- Η *υπηρεσία αρχείου* παρέχει τη δυνατότητα σε μηχανές του κατανεμημένου συστήματος να προσπελάνουν (π.χ. για ανάγνωση και εγγραφή) αρχεία που βρίσκονται σε απομακρυσμένες μηχανές με τον ίδιο τρόπο που προσπελάνουν τα τοπικά τους αρχεία.
- Η υπηρεσία υποστηρίζεται από ένα κατανεμημένο σύστημα εξυπηρετητών αρχείων και συνοδεύεται από *υπηρεσία ευρετηρίου (directory service)* που επιτρέπει διεργασίες όπως τη δημιουργία, διαγραφή και αναζήτηση ενός αρχείου.

Υπηρεσία Συναλλαγών



Οι κατανεμημένες συναλλαγές χρησιμοποιούνται σε δεδομένα που μπορεί να είναι διαμοιρασμένα σε πολλά μηχανήματα.

Συχνά μια κατανεμημένη εφαρμογή εκτελεί μια σειρά συναλλαγών (*transactions*), π.χ. ακολουθία ενεργειών ανάγνωσης ή εγγραφής, σε πολλούς πόρους με **ατομικό (*atomic*) τρόπο, δηλαδή είτε όλες τις λειτουργίες είτε καμιά.**

- Η ιδιότητα αυτή είναι ιδιαίτερα σημαντική για περιπτώσεις που απαιτείται συγκάλυψη αστοχιών.
- Για παράδειγμα, η μεταφορά χρημάτων από μία τράπεζα σε μία άλλη απαιτεί αφαίρεση χρημάτων από το λογαριασμό αποστολής και πρόσθεση χρημάτων στο λογαριασμό λήψης, με τον περιορισμό όμως ότι και οι δύο κινήσεις θα πρέπει να εκτελεστούν μαζί για να έχει νόημα η συναλλαγή.



Υπηρεσία Συναλλαγών

- Για να επιτευχθεί ατομικότητα, η *υπηρεσία συναλλαγών* (*distributed transaction service*) αναλαμβάνει να **συγχρονίσει όλους** τους πελάτες που επιθυμούν να εκτελέσουν ταυτόχρονα συναλλαγές με τους ίδιους πόρους, έτσι ώστε οι συναλλαγές τους να φαίνεται ότι εκτελέστηκαν σαν ατομικές πράξεις με κάποια σειρά.
- Στα ΚΣ η υλοποίηση των συναλλαγών είναι πολύ δυσκολότερη, λόγω των καθυστερήσεων και της αναξιοπιστίας του δικτύου επικοινωνίας. Είναι επίσης δύσκολο να επεκταθούν σε μηχανήματα με ευρεία γεωγραφική διασπορά.



Υπηρεσία Αναπαραγωγής

- Συχνά δεδομένα ενός ΚΣ διατίθενται σε πολλαπλά αντίγραφα, για λόγους αύξησης επίδοσης ή αξιοπιστίας. Η υπηρεσία αναπαραγωγής (*replication service*) διαχειρίζεται την αναπαραγωγή και το συγχρονισμό των αντιγράφων, αποκρύπτοντας το πλήθος και τη θέση των αντιγράφων.
- Η αναπαραγωγή μπορεί να είναι στατική (σταθερό πλήθος αντιγράφων) ή δυναμική (δημιουργία και καταστροφή), ενώ ο συγχρονισμός μπορεί να γίνεται άμεσα, περιοδικά ή με άλλο τρόπο.
- Η επιλογή των ρυθμίσεων εξαρτάται από τις απαιτήσεις κάθε εφαρμογής, π.χ. ταυτόχρονη πρόσβαση από πολλούς πελάτες.
- Παράδειγμα είναι οι *mirrored web servers* που προσπελούν το πλησιέστερο ή το λιγότερο φορτωμένο αντίγραφο, αυξάνοντας την επίδοση και αξιοπιστία.

Υπηρεσία Ασφάλειας



- **Στα συγκεντρωτικά συστήματα η υπηρεσία ασφάλειας αφορά**
 - Την πιστοποίηση της ταυτότητας των χρηστών
 - Τη διαφύλαξη του απορρήτου τους
 - Τον έλεγχο πρόσβασης στους πόρους του συστήματος
- **Στα ΚΣ, αυτά περιπλέκονται επειδή**
 - έχουμε επικοινωνία μέσω ενός δικτύου το οποίο γενικά δεν είναι υπό τον έλεγχό μας
 - υπάρχει ανάγκη συνεργασίας αυτόνομων μηχανών που υποστηρίζουν διαφορετικούς μηχανισμούς και πολιτικές ασφάλειας

Υπηρεσία Ασφάλειας



- Σε ένα συγκεντρωτικό σύστημα η ασφάλεια υλοποιείται κυρίως μέσω **μηχανισμών προστασίας του πυρήνα του λειτουργικού συστήματος**.
- Σε ένα ΚΣ δεν υπάρχει προστατευμένη λειτουργία, και απαιτούνται **κρυπτογραφικοί μηχανισμοί** για την επίτευξη του επιθυμητού βαθμού ασφάλειας.
- Μία κατανεμημένη υπηρεσία ασφάλειας
 - ✓ παρέχει μηχανισμούς ελέγχου πρόσβασης σε πόρους με βάση ψηφιακά πιστοποιητικά χωρίς να παραβιάζει την αυτονομία των επί μέρους μηχανών.
 - ✓ προσφέρει πιστοποίηση ταυτότητας των χρηστών και δεν επιτρέπει υποκλοπή, τροποποίηση ή εισαγωγή μηνυμάτων από τρίτους

Ενδιάμεσο λογισμικό και ανοιχτή λειτουργία



Σκοπός

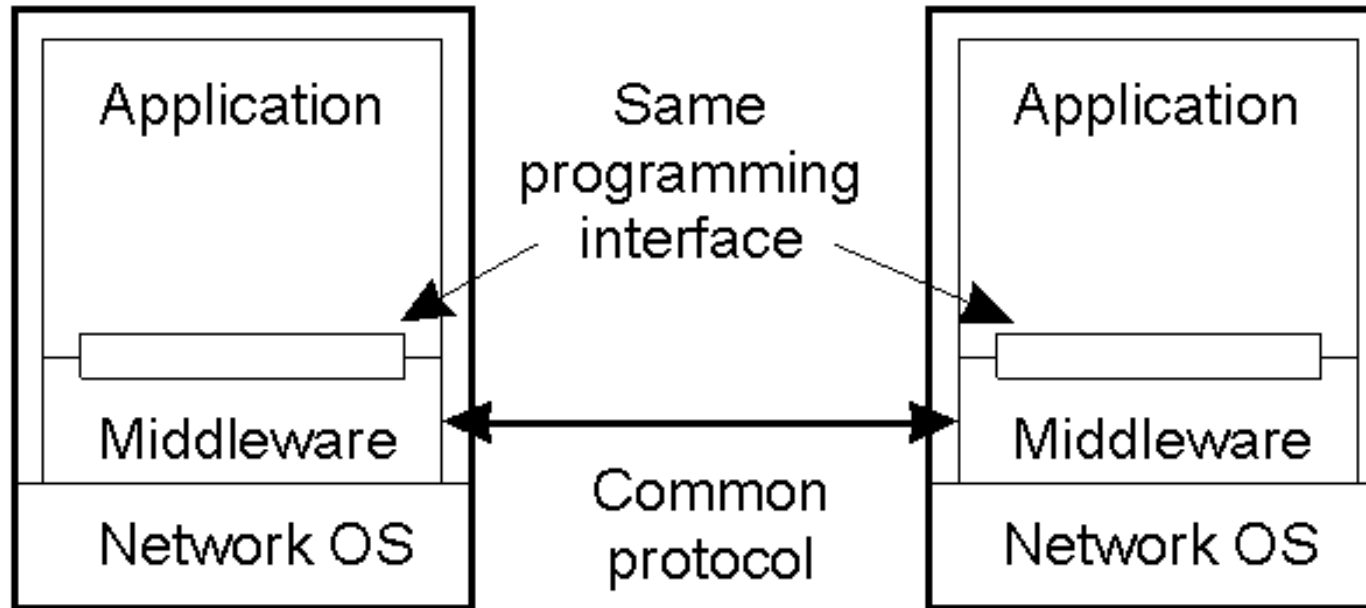
- Να παρέχει υπηρεσίες σύμφωνα με κάποιους κανόνες, οι οποίοι περιγράφουν τη σύνταξη και τη σημαντική (semantics) των υπηρεσιών.
- Οι υπηρεσίες καθορίζονται μέσω διασυνδέσεων (interfaces) που περιγράφονται σε μια Γλώσσα Ορισμού Διασυνδέσεων (Interface Definition Language, IDL)

Ένας Ορισμός Διασύνδεσης επιτρέπει σε μια οποιαδήποτε διεργασία που χρειάζεται μια διασύνδεση να «συνομιλήσει» με μια άλλη διεργασία που παρέχει τη διασύνδεση

Ανεξάρτητοι παραγωγοί μπορούν να κατασκευάσουν διαφορετικές υλοποιήσεις των ίδιων interfaces με αποτέλεσμα διαφορετικά ΚΣ να λειτουργούν με ακριβώς τον ίδιο τρόπο

Ένας κατάλληλος ορισμός interface πρέπει να είναι πλήρης και ουδέτερος

Ενδιάμεσο λογισμικό και ανοιχτή λειτουργία



Τα πρωτόκολλα του ενδιάμεσου λογισμικού και οι διασυνδέσεις που παρέχουν για τις εφαρμογές, θα πρέπει να είναι ίδια.

Πληρότητα στις διασυνδέσεις: π.χ. θα πρέπει να προδιαγράφουν ποια μορφή θα έχουν οι αναφορές στις οντότητες (π.χ., μέσω URL διευθύνσεων ή μέσω διευθύνσεων δικτύου)

Ενδιάμεσο λογισμικό και ανοιχτή λειτουργία



Η πληρότητα και η ουδετερότητα αποτελούν σημαντικούς παράγοντες:

- **Δια-λειτουργικότητας (interoperability):** Χαρακτηρίζει το κατά πόσον δύο υλοποιήσεις συστημάτων ή στοιχείων που προέρχονται από διαφορετικούς κατασκευαστές μπορούν να συνυπάρχουν και να συνεργάζονται βασισόμενες απλώς η μία στις υπηρεσίες της άλλης, όπως αυτές καθορίζονται από ένα κοινό πρότυπο.
- **Φορητότητας (portability):** Χαρακτηρίζει το κατά πόσον μια εφαρμογή που έχει αναπτυχθεί για ένα κατανεμημένο σύστημα Α μπορεί να εκτελεστεί χωρίς τροποποιήσεις σε ένα διαφορετικό κατανεμημένο σύστημα Β, το οποίο υλοποιεί τις ίδιες διασυνδέσεις με το Α.
- **Ευελιξίας (flexibility):**
 - δυνατότητα ένα σύστημα να διαμορφωθεί από διαφορετικά στοιχεία υλοποιημένα από διαφορετικούς κατασκευαστές
 - Το σύστημα είναι μία συλλογή επιμέρους στοιχείων τα οποία μπορούν εύκολα να προσαρμοστούν ή να αντικατασταθούν



Item	Distributed OS		Network OS	Middleware-based OS
	Multiproc.	Multicomp.		
Degree of transparency	Very High	High	Low	High
Same OS on all nodes	Yes	Yes	No	No
Number of copies of OS	1	N	N	N
Basis for communication	Shared memory	Messages	Files	Model specific
Resource management	Global, central	Global, distributed	Per node	Per node
Scalability	No	Moderately	Yes	Yes
Openness	Closed	Closed	Open	Open

Οργάνωση Κατανεμημένων Συστημάτων



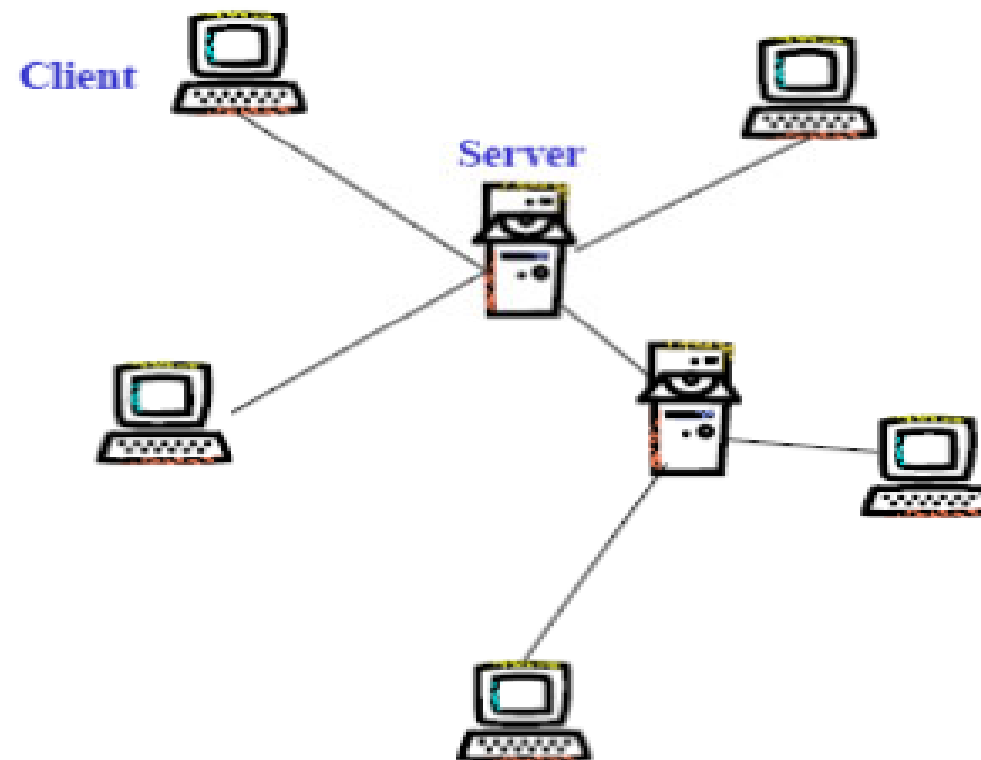
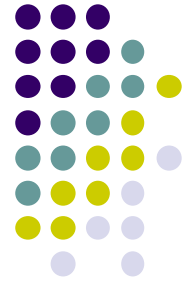
- Το Μοντέλο Πελάτη-Διακομιστή
- Ομότιμα Συστήματα



Το Μοντέλο Πελάτη-Διακομιστή

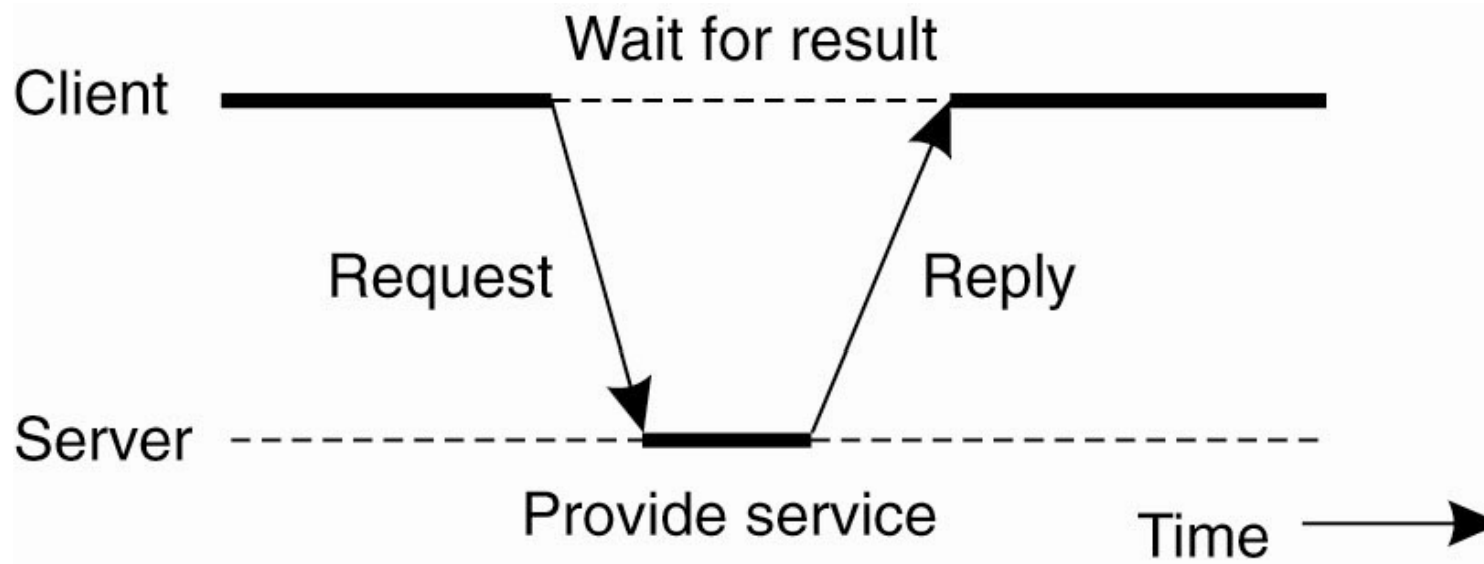
- Ένα σύστημα client-server είναι ένα σύστημα στο οποίο το δίκτυο ενώνει διάφορους υπολογιστικούς πόρους, ώστε οι clients (front end) να μπορούν να ζητούν υπηρεσίες από έναν server (back end) που προσφέρει πληροφορίες ή πρόσθετη υπολογιστική ισχύ
- Το client-server μοντέλο βασίζεται, συνήθως, σε ένα απλό πρωτόκολλο αίτησης/απάντησης (request/reply).
 - Ο πελάτης στέλνει ένα μήνυμα (αίτηση) ζητώντας από τον εξυπηρετητή κάποια υπηρεσία.
 - Ο εξυπηρετητής ενεργοποιείται άμεσα ή προσθέτει την αίτηση σε μια ουρά. Τελικά αποκρίνεται εκτελώντας μια σειρά από ενέργειες και επιστρέφει μια απάντηση, συνήθως με τα δεδομένα που ζητήθηκαν ή ένα μήνυμα λάθους.

Το Μοντέλο Πελάτη-Διακομιστή



Οργάνωση Κατανεμημένων Συστημάτων

– Το Μοντέλο Πελάτη-Διακομιστή



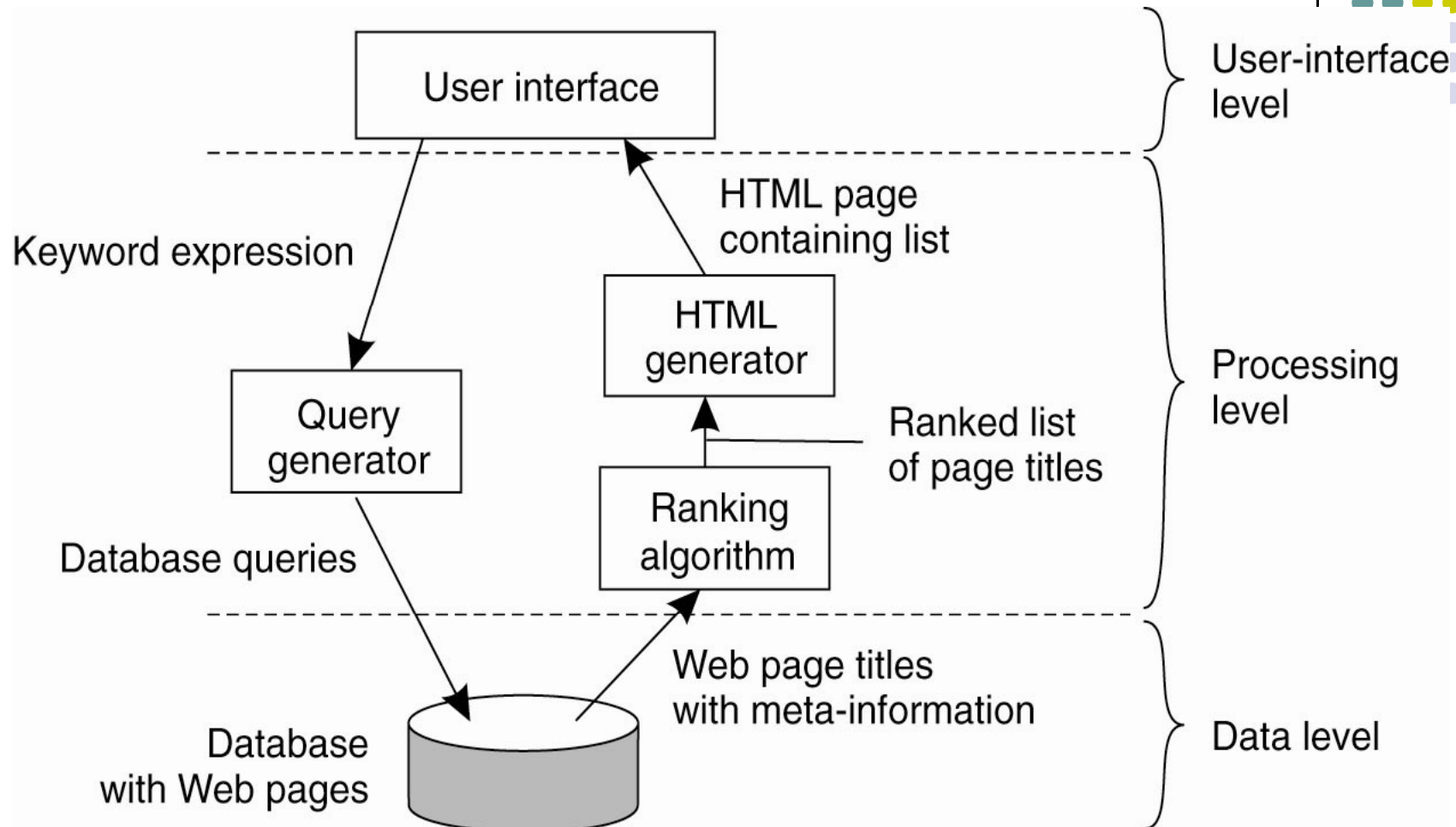
Γενική αλληλεπίδραση μεταξύ διεργασίας-πελάτη και διεργασίας διακομιστή

Επίπεδα Εφαρμογής



- *Επίπεδο διεπαφής χρήστη: χειρίζεται την αλληλεπίδραση με το χρήστη*
- *Επίπεδο επεξεργασίας: παρέχει τις λειτουργικές δυνατότητες της εφαρμογής*
- *Επίπεδο δεδομένων: βάση δεδομένων ή σύστημα αρχείων*

Επίπεδα Εφαρμογής



Οργάνωση μιας μηχανής αναζήτησης του Internet σε τρία διαφορετικά επίπεδα

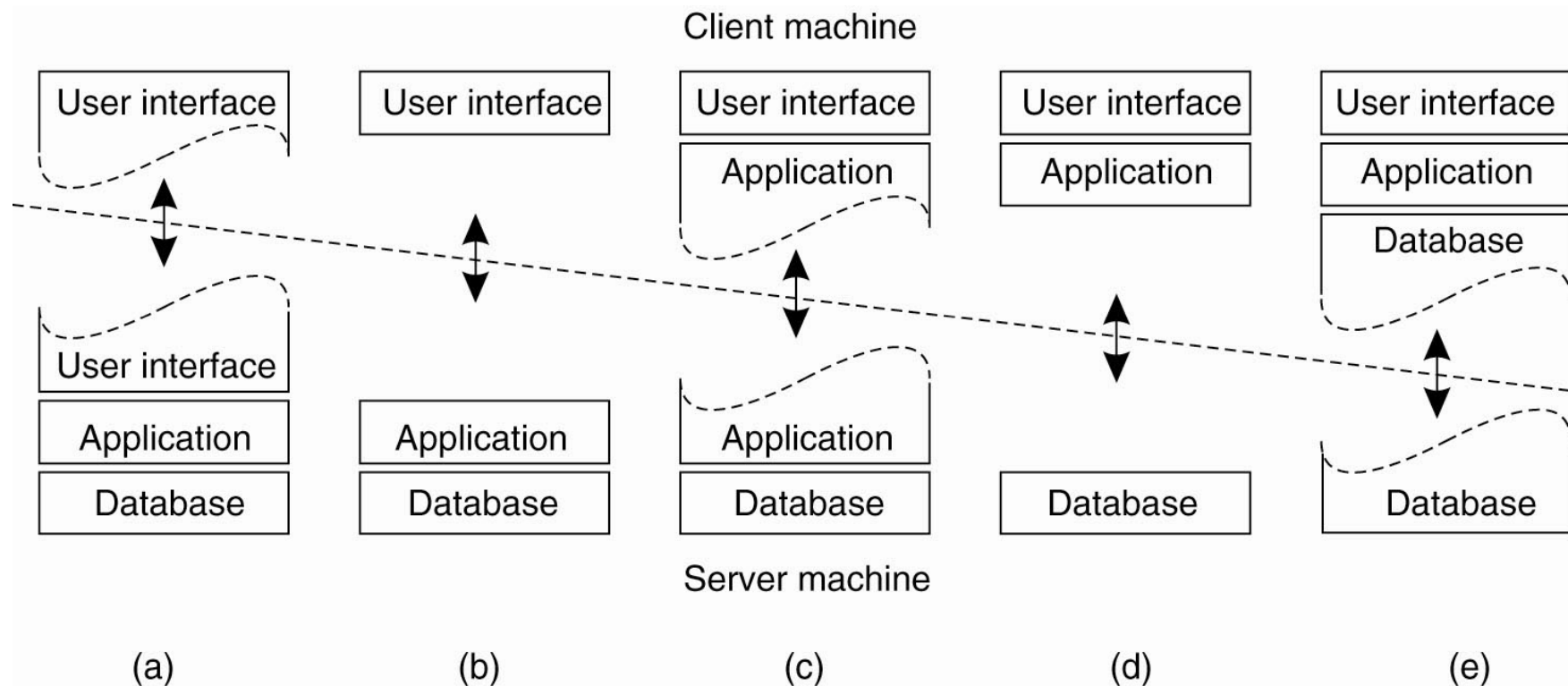
Πολυστρωματικές Αρχιτεκτονικές (Multitiered architectures)



Η απλούστερη οργάνωση περιλαμβάνει δύο τύπους μηχανών:

- 1. Μία μηχανή πελάτη η οποία υλοποιεί το επίπεδο διεπαφής του χρήστη**
- 2. Μία μηχανή διακομιστή η οποία υλοποιεί το επίπεδο επεξεργασίας και το επίπεδο δεδομένων**

Πολυστρωματικές αρχιτεκτονικές (multitiered architectures)



Τρόποι οργάνωσης πελάτη-διακομιστή σε μια διστρωματική
(two-tiered) αρχιτεκτονική

Πολυστρωματικές αρχιτεκτονικές



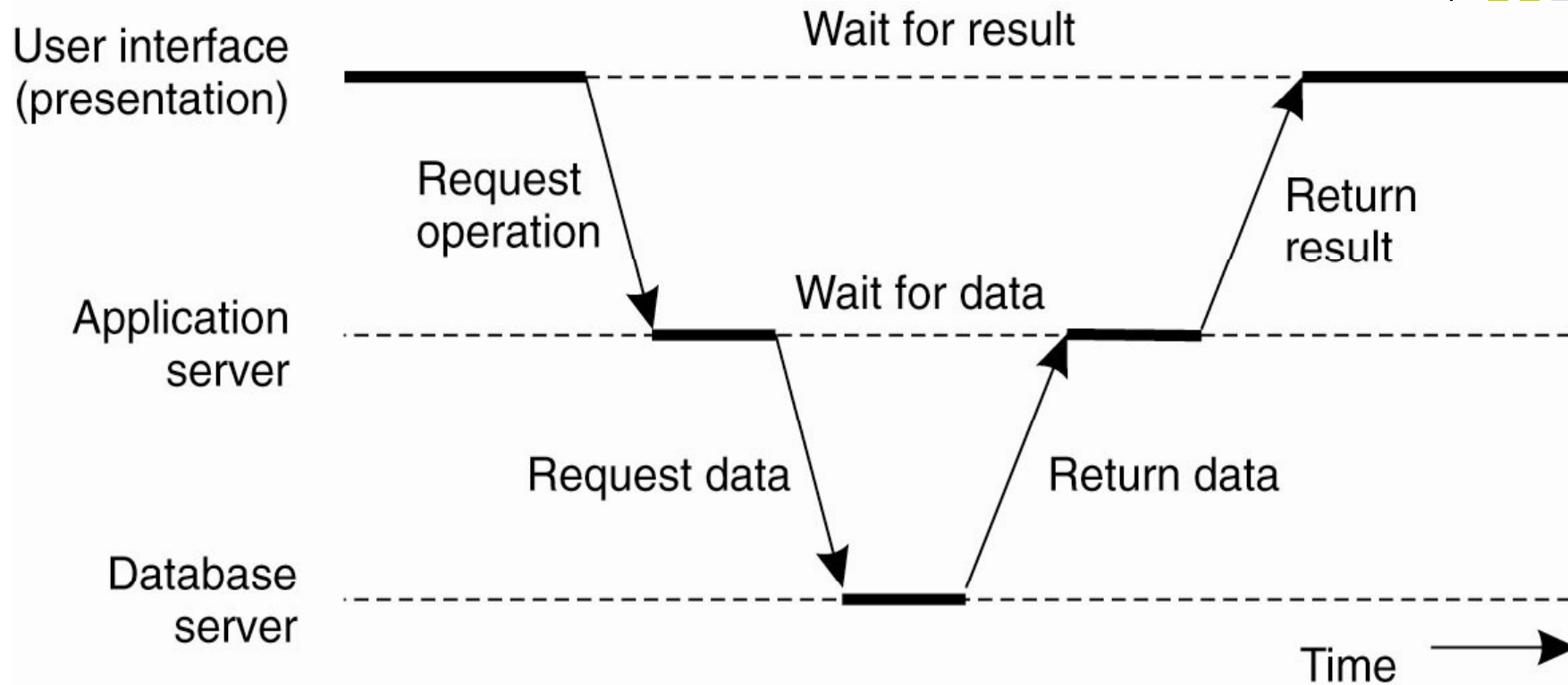
Οργάνωση (α): στον πελάτη βρίσκεται μόνο το μέρος της διασύνδεσης χρήστη που εξαρτάται από το τερματικό και στην εφαρμογή η δυνατότητα απομακρυσμένου ελέγχου στην παρουσίαση των δεδομένων της.

Οργάνωση (γ): μετακινείται στον πελάτη ένα μέρος της εφαρμογής (π.χ., ο χρήστης συμπληρώνει και ελέγχει φόρμες που απαιτεί η εφαρμογή, αλληλεπιδρώντας με την εφαρμογή).

Οργάνωση (δ): η εφαρμογή εκτελείται στον πελάτη αλλά οι λειτουργίες στη βάση δεδομένων εκτελούνται στον διακομιστή.

Οργάνωση (ε): όπως η (δ) αλλά ο τοπικός δίσκος περιέχει αντίγραφο μέρους των δεδομένων.

Πολυστρωματικές αρχιτεκτονικές



Τριστρωματική (Three-tiered) Αρχιτεκτονική :

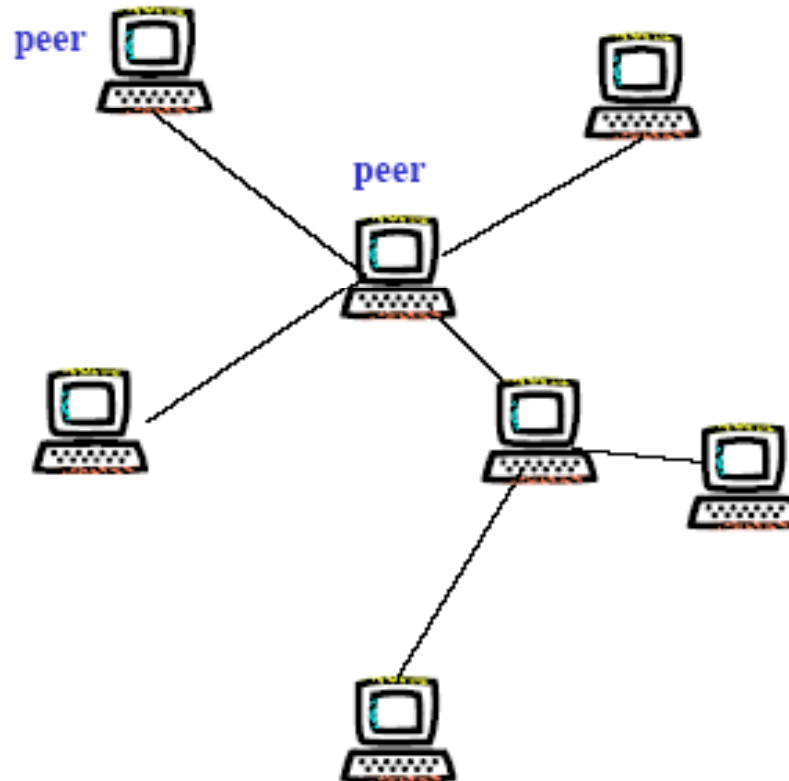
τα προγράμματα που αποτελούν μέρος του επιπέδου
επεξεργασίας μπορούν να βρίσκονται σε ξεχωριστό διακομιστή

Ομότιμα Συστήματα (Peer-to-Peer Systems)



Κάθε κόμβος διαθέτει το ίδιο λογισμικό και επομένως συμπεριφέρεται ως πελάτης και ως διακομιστής.

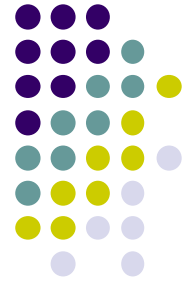
- **Ομότιμη** δράση των κόμβων.
- Οι κόμβοι επικοινωνούν απ'ευθείας και όχι μέσω ενός server



Peer-to-Peer Systems (κίνητρο)



- Το Internet έχει τρεις πολύτιμους πόρους
 - Πληροφορίες
 - Υπολογιστική Ισχύ
 - Διαδικτύωση των παραπάνω (bandwidth)
- Όλα υπο-χρησιμοποιούνται εξαιτίας (κυρίως) του παραδοσιακού μοντέλου client-server!!



Ομότιμα Συστήματα (P2P Systems)

- **Αξιοποίηση των ελεύθερων πόρων συστημάτων**
- **Δημιουργία περισσότερο κλιμακούμενων συστημάτων**
- **Δημιουργία συστημάτων με μεγαλύτερη διαθεσιμότητα**
- **Κατάργηση μονοπωλίων στην διάθεση της πληροφορίας**
- **Αυτο-οργάνωση αντί κεντρικής διαχείρισης**

Εφαρμογές Ομότιμων Συστημάτων



- *Κατανεμημένος Υπολογισμός*
- *Διαμοιρασμός Αρχείων*
- *Συνεργατικές Εφαρμογές*
- Εφαρμογές Κατανεμημένου Υπολογισμού απαιτούν την αποσύνθεση μεγάλων προβλημάτων σε μικρότερα, παράλληλα προβλήματα
- Διαμοιρασμός αρχείων απαιτεί αποδοτική αναζήτηση και μεταφορά αρχείων μέσα σε wide area networks
- Συνεργατικές εφαρμογές απαιτούν μηχανισμούς ενημέρωσης (update mechanisms) για περιβάλλοντα με πολλούς χρήστες (multi-user environments)



Παραδείγματα

- Κατανεμημένου Υπολογισμού (π.χ., SETI@home, Search for Extraterrestrial Intelligence)
- Διαμοιρασμός αρχείων (π.χ., Gnutella, BitTorrent, eMule)
- Συνεργατικές (π.χ., Magi, Groove, Jabber, Chat – icq, yahoo!, etc)
- Πλατφόρμες (π.χ., JXTA)

SETI@home

([MilkyWay@home](#), [Einstein@home](#))



- Αναζήτηση εξωγήινης ζωής με πάνω από 3 εκατομμύρια PC να αναλύουν δεδομένα που συλλέγονται από ραδιο-τηλεσκόπια
- Αρχιτεκτονικά, το Seti@Home βασίζεται στο client-server μοντέλο
- Οι κεντρικοί servers κρατούν τεράστιο όγκο δεδομένων που συλλέγονται από το Arecibo radio telescope που «ακούει» τα σήματα από τον ουρανό
- Αυτά τα δεδομένα πρέπει να αναλυθούν για περιέργα ραδιοκύματα που ίσως σημαίνουν ότι υπάρχει και άλλη ζωή

SETI@home



- Anybody with an at least intermittently Internet-connected computer can participate in SETI@home by running a free program that downloads and analyzes radio telescope data.
- The SETI@home distributed computing software runs either as a screensaver or continuously while a user works, making use of processor time that would otherwise be unused.
- With over 5.2 million participants worldwide, the project is the distributed computing project with the most participants to date. SETI@home has the ability to compute over 769 teraFLOPS.



Gnutella

- Το Gnutella είναι ένα πρωτόκολλο για διαμοιρασμό αρχείων
- Επιτρέπει σε χρήστες να ψάξουν και να κατεβάσουν αρχεία από άλλους χρήστες συνδεδεμένους στο Internet
- Η δρομολόγηση στο δίκτυο επιτυγχάνεται μέσω της πλήρους αναμετάδοσης (**broadcasting**) των μηνυμάτων



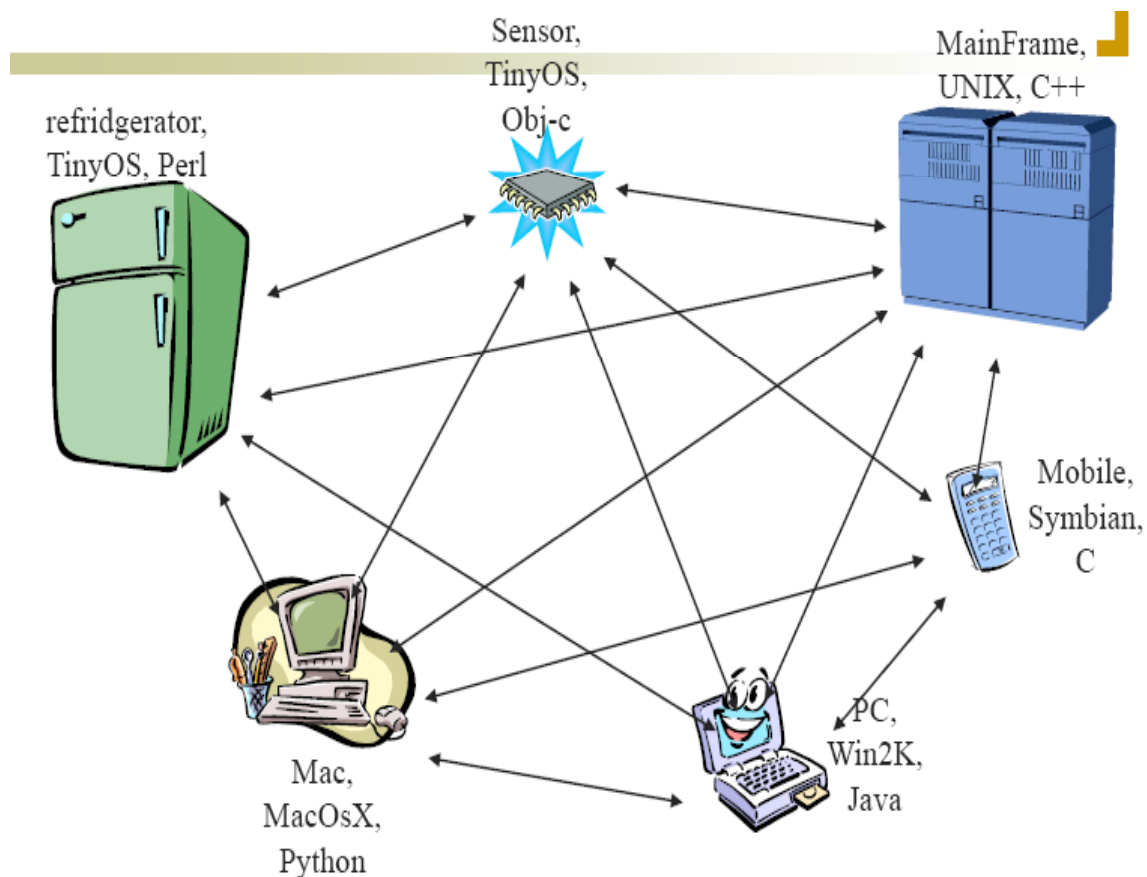
JXTA (Juxtapose):

μία υπολογιστική πλατφόρμα P2P,
που ορίζει ένα σύνολο πρωτοκόλλων,
που μπορούν να χρησιμοποιηθούν για
την δημιουργία εφαρμογών.

■ JXTA (<http://www.jxta.org>)

■ Τρεις στόχοι

- Λειτουργία με πολλά διαφορετικά P2P συστήματα
- Ανεξαρτησία από λειτουργικό σύστημα, δίκτυο, κλπ
- Παρουσία παντού (Ubiquity)
 - Σε κάθε ηλεκτρονική συσκευή





JXTA-Κατηγορίες peers

- **edge peers**, μέλη που συνήθως έχουν χαμηλής ταχύτητας σύνδεση.
- **Rendezvous peer** είναι ειδικού σκοπού μέλος που έχει ως σκοπό να συντονίζει τα άλλα μέλη του ομότιμου δικτύου. Επίσης προωθεί τα μηνύματα. Αν τα απλά μέλη βρίσκονται σε διαφορετικά υποδίκτυα πρέπει τουλάχιστον να έχουν ένα Rendezvous peer.
- **Relay peer** επιτρέπει σε μέλη που βρίσκονται πίσω από firewalls να συνδέονται σε ένα JXTA δίκτυο. Αυτό μπορεί να γίνει χρησιμοποιώντας ένα πρωτόκολλο όπως το [HTTP](#).



Τύποι Ομότιμων Συστημάτων

- Υβριδικά Ομότιμα Συστήματα
- Αδόμητα Ομότιμα Συστήματα
- Ιεραρχικά Ομότιμα Συστήματα
- Δομημένα Ομότιμα Συστήματα

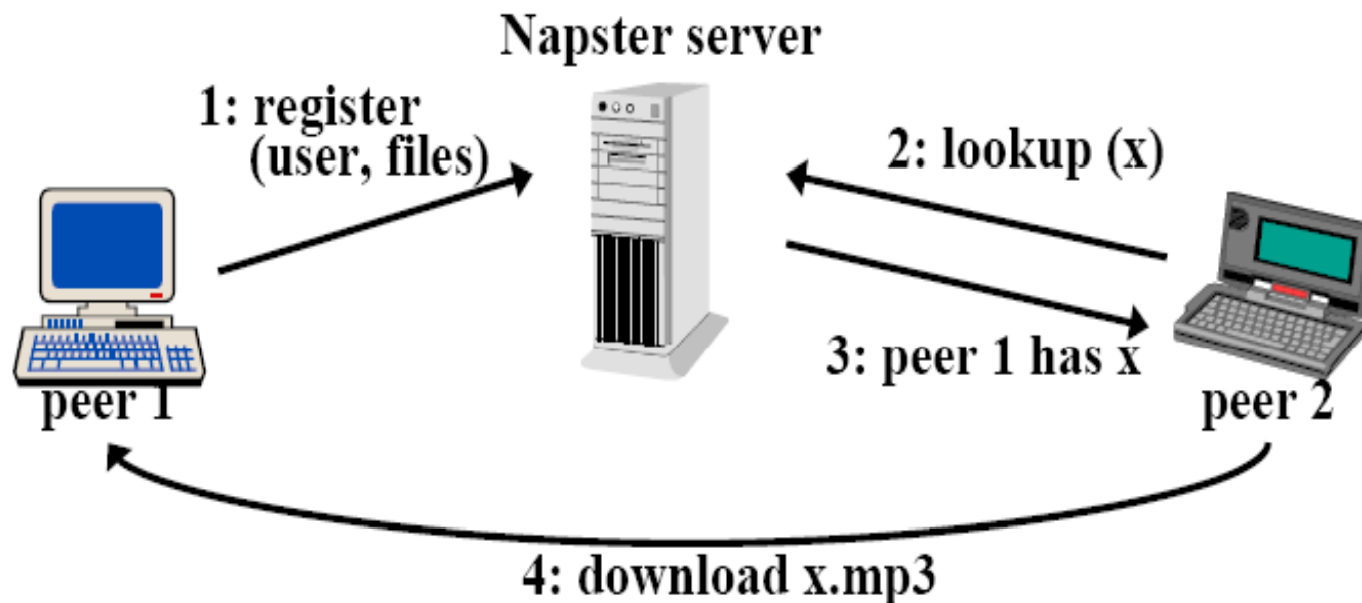
Υβριδικά Ομότιμα Συστήματα (Hybrid P2P systems)



Υπάρχει ένας κεντρικός διακομιστής

- **Διαμοιρασμός αρχείων:** *Αν ένας κόμβος διαθέτει ένα αρχείο, οι άλλοι μπορούν να το αναζητήσουν στον κεντρικό διακομιστή και να το κατεβάσουν από τη μηχανή όπου βρίσκεται*

Napster (1998-2001): διαμοιρασμός MP3



Google (Client-Server) vs. Napster (P2P)



Ίδιας κλίμακας συστήματα

- εκατομμύρια αναζητήσεις ημερησίως
- Terabytes+ δεδομένων

Google

- Στηρίζεται σε πάνω από 100.000 μηχανές
- το στήσιμο μιας τέτοιας εφαρμογής έχει μεγάλο κόστος

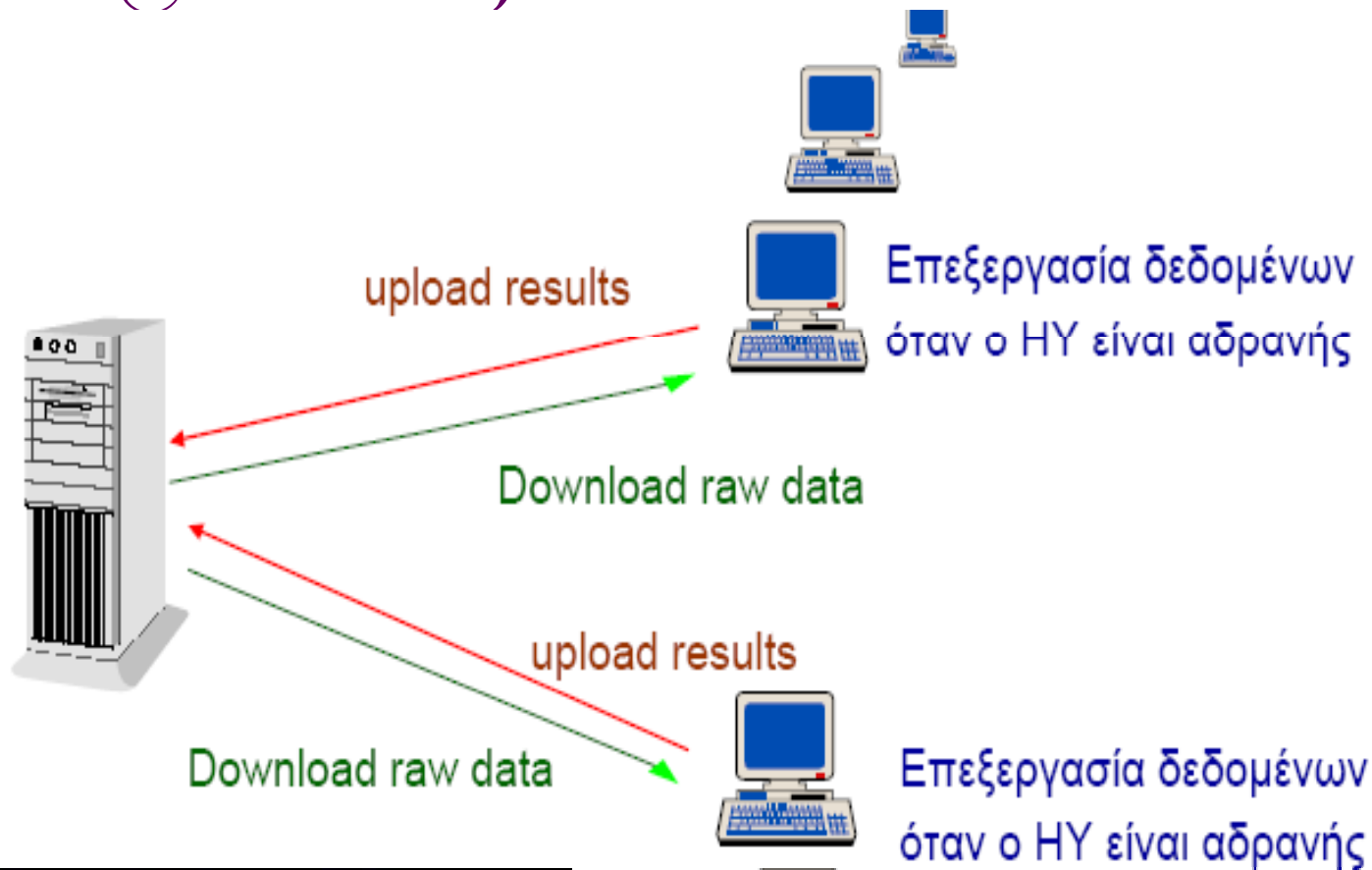
Napster

- ο server χρησιμοποιεί μόνο 100 μηχανές
- μικρό κόστος αφού το κόστος αποθήκευσης και μεταφοράς των μουσικών αρχείων χρεώνεται στις μηχανές των χρηστών του συστήματος

Υβριδικά Ομότιμα Συστήματα



Διαμοιρασμός υπολογιστικών πόρων (π.χ.
SETI@home κ.α.)



Υβριδικά Ομότιμα Συστήματα

Πλεονεκτήματα



Διαμερισμός πόρων

- αποθηκευτικών, αφού οι χρήστες του συστήματος αποθηκεύουν τα αρχεία, όχι ο διακομιστής
- υπολογιστικών, αφού οι χρήστες του συστήματος εκτελούν τον υπολογισμό όχι ο διακομιστής
- επικοινωνίας, αφού το κατέβασμα αρχείων γίνεται μεταξύ των χρηστών χωρίς να παρεμβάλλεται ο διακομιστής
- εισαγωγής στοιχείων αφού
 - οι χρήστες του συστήματος εισάγουν τα δεδομένα στο σύστημα
 - οι χρήστες του συστήματος τα κατηγοριοποιούν

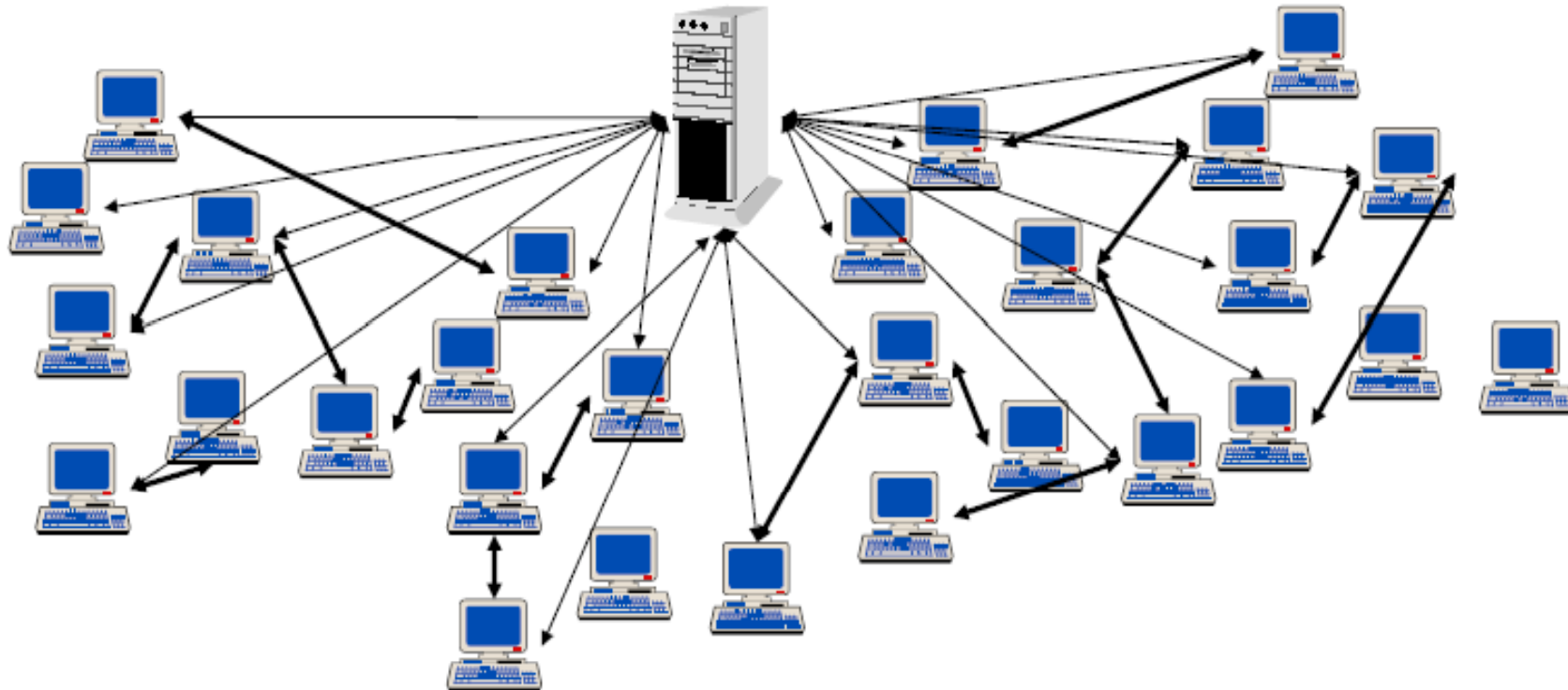
Η αποκέντρωση επιτρέπει τη δημιουργία εφαρμογών παγκόσμιας κλίμακας με μικρό κόστος!

Υβριδικά Ομότιμα Συστήματα

Μειονεκτήματα:



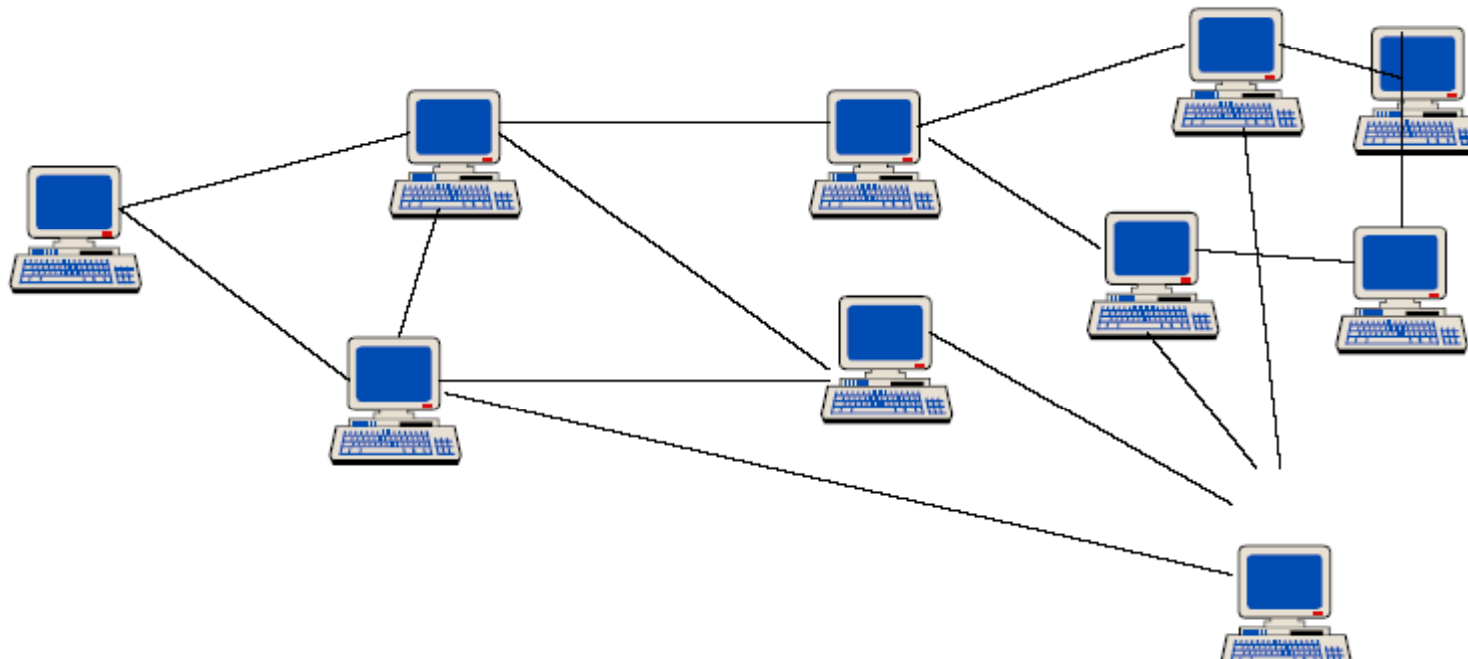
Central Point of Failure



Αδόμητα Ομότιμα Συστήματα (Unstructured ή Pure P2P systems)



- Δεν υπάρχει κεντρικός εξυπηρετητής
Gnutella (1999)



Gnutella

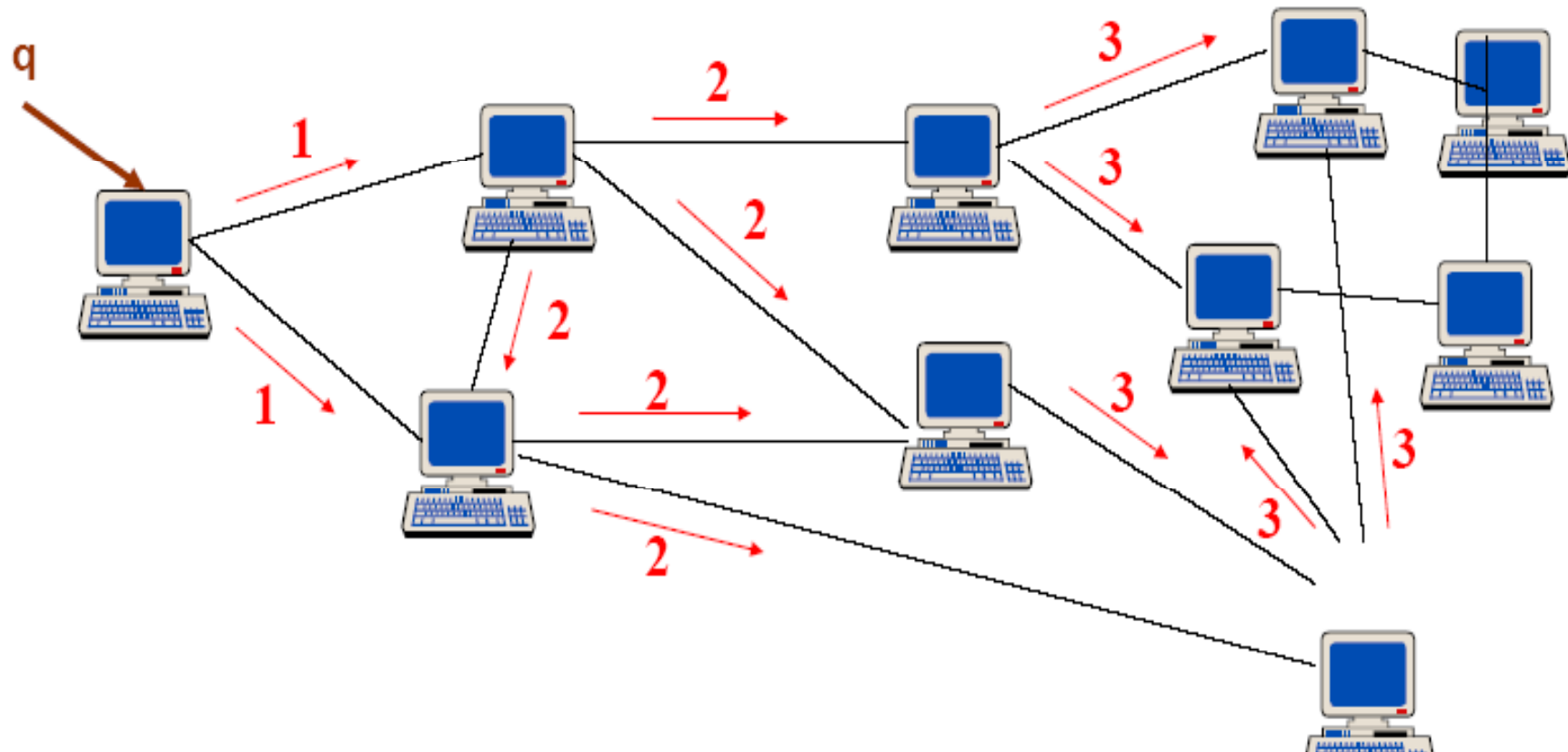


- Το δίκτυο αποτελείται από κόμβους που ονομάζονται **servents** από το **SERV(er) + cliENT**.
- Η εισαγωγή ενός κόμβου στο δίκτυο γίνεται με την σύνδεση σε κάποιον κόμβο που είναι συνδεδεμένος στο δίκτυο
- Κάθε κόμβος στέλνει περιοδικά μηνύματα ring για να βρίσκει νέο-εισαχθέντες κόμβους. Έτσι διατηρείται ο δυναμικός χαρακτήρας του δικτύου
- Τα μηνύματα είτε γίνονται broadcast είτε back-propagated
- Κάθε μήνυμα έχει ένα τυχαίο μοναδικό id
- Κάθε κόμβος διαθέτει λίγη μνήμη για τα μηνύματα που πρόσφατα έχει δρομολογήσει την οποία χρησιμοποιεί για να μην ξανακάνει broadcast αλλά και για να υλοποιήσει το back-propagation
- Ορίζεται μια τιμή για το χρόνο ζωής TTL
- Οι κόμβοι αποφασίζουν που να συνδεθούν στο δίκτυο βασιζόμενοι αποκλειστικά σε τοπική πληροφορία. Το Gnutella είναι ένα δίκτυο το οποίο οργανώνεται μόνο του.

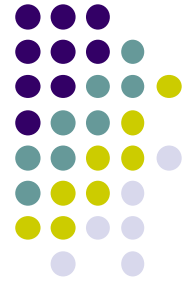
Αδόμητα Ομότιμα Συστήματα



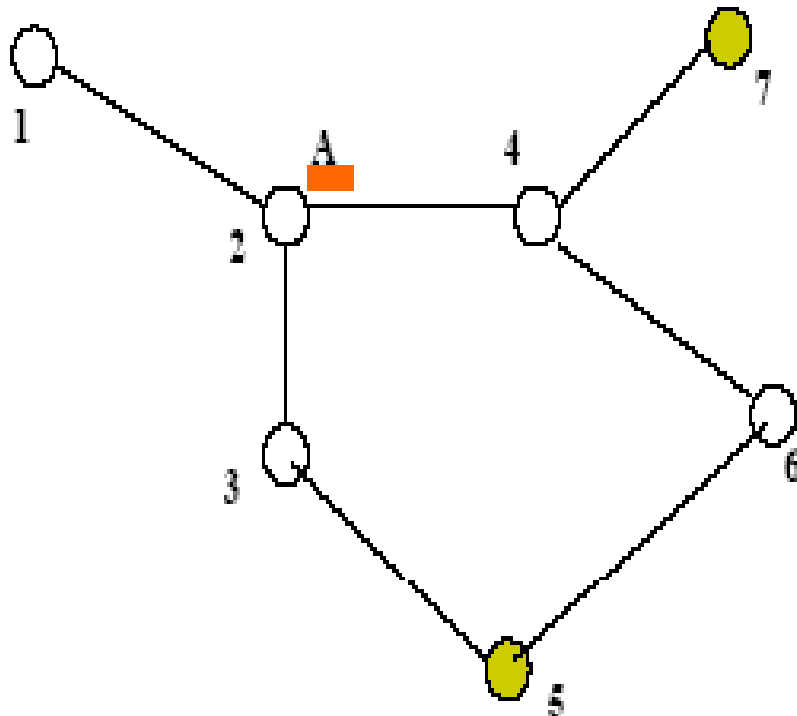
Χρήση κατακλυσμού μηνυμάτων (message flooding) για τον
εντοπισμό πόρων (resource discovery)



Gnutella



Εντοπισμός πόρων

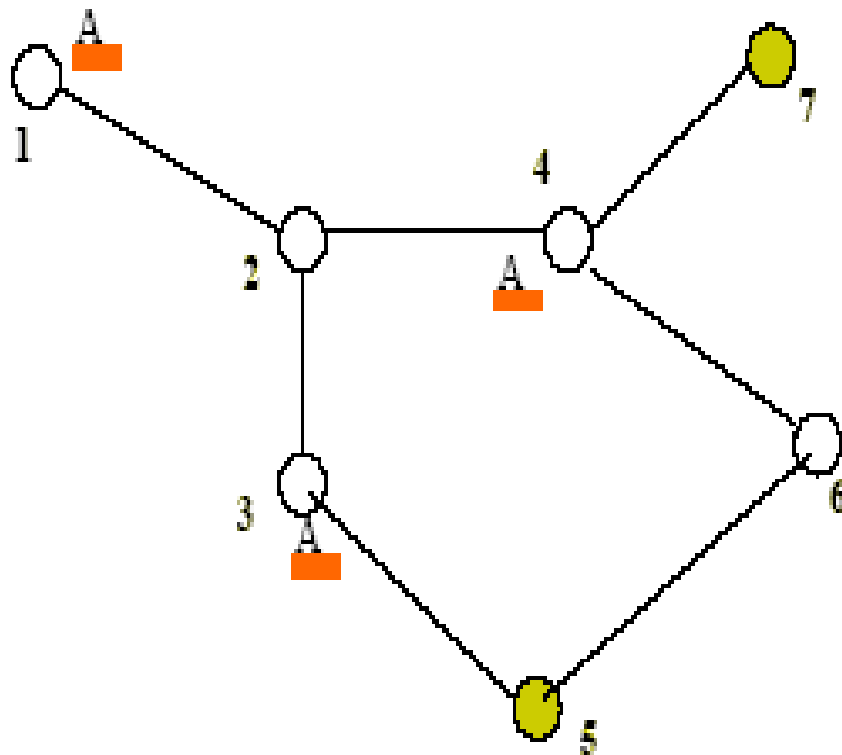


Βήματα:

- Ο κόμβος 2 αρχίζει μια αναζήτηση για το αρχείο A.



Εντοπισμός πόρων

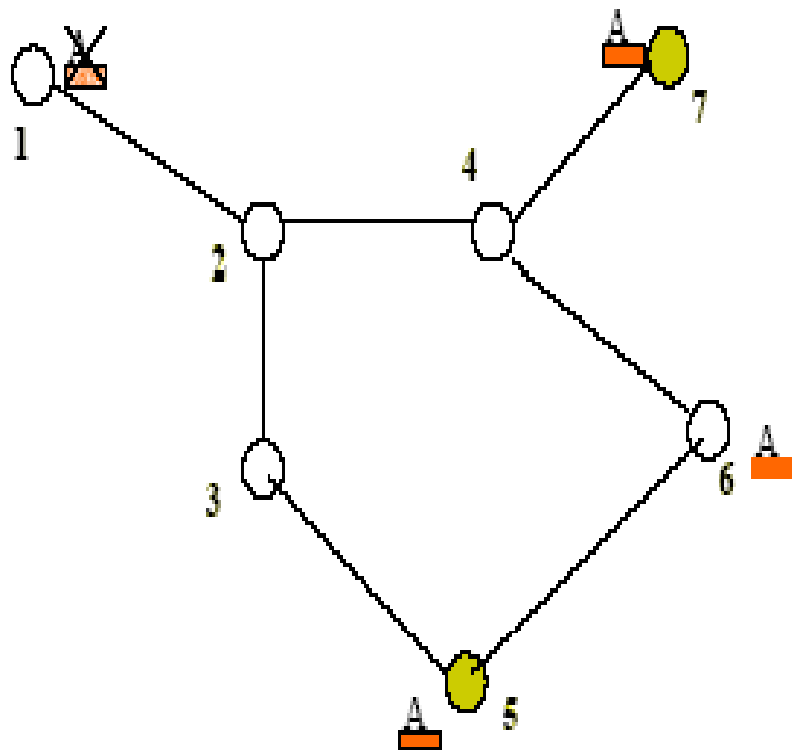


Βήματα:

- Ο κόμβος 2 αρχίζει μια αναζήτηση για το αρχείο A.
- Στέλνει μήνυμα σε όλους τους γείτονες.



Εντοπισμός πόρων

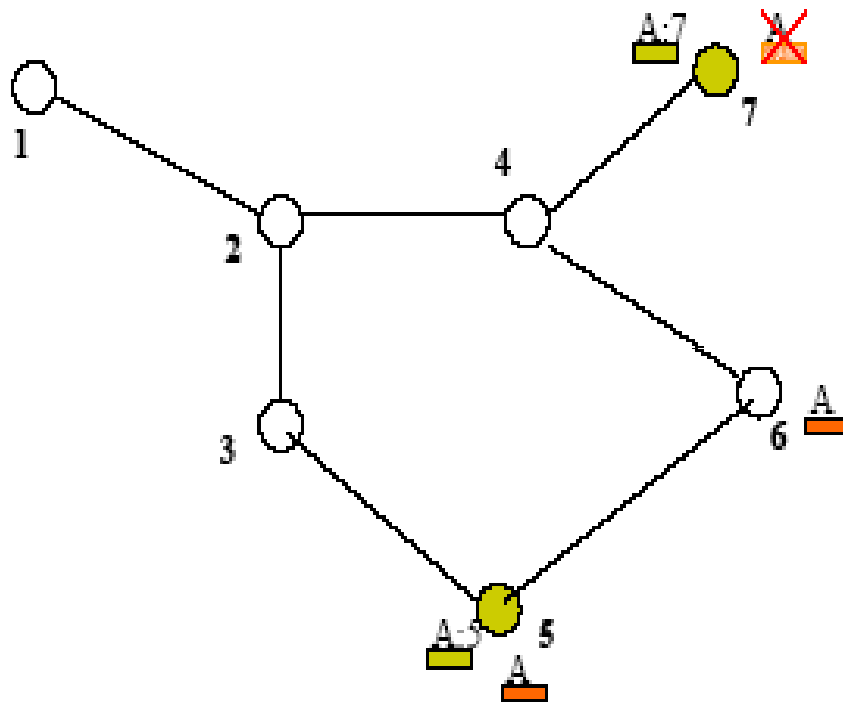


Βήματα:

- Ο κόμβος 2 αρχίζει μια αναζήτηση για το αρχείο A.
- Στέλνει μήνυμα σε όλους τους γείτονες.
- Οι γείτονες προωθούν το μήνυμα.



Εντοπισμός πόρων

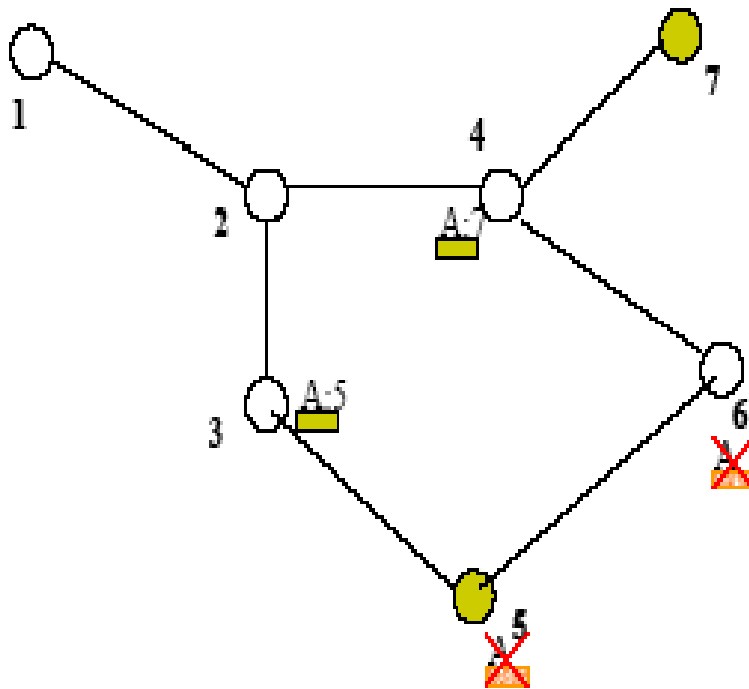


Βήματα:

- Ο κόμβος 2 αρχίζει μια αναζήτηση για το αρχείο A.
- Στέλνει μήνυμα σε όλους τους γείτονες.
- Οι γείτονες προωθούν το μήνυμα.
- Οι κόμβοι που έχουν το αρχείο A στέλνουν απάντηση.



Εντοπισμός πόρων



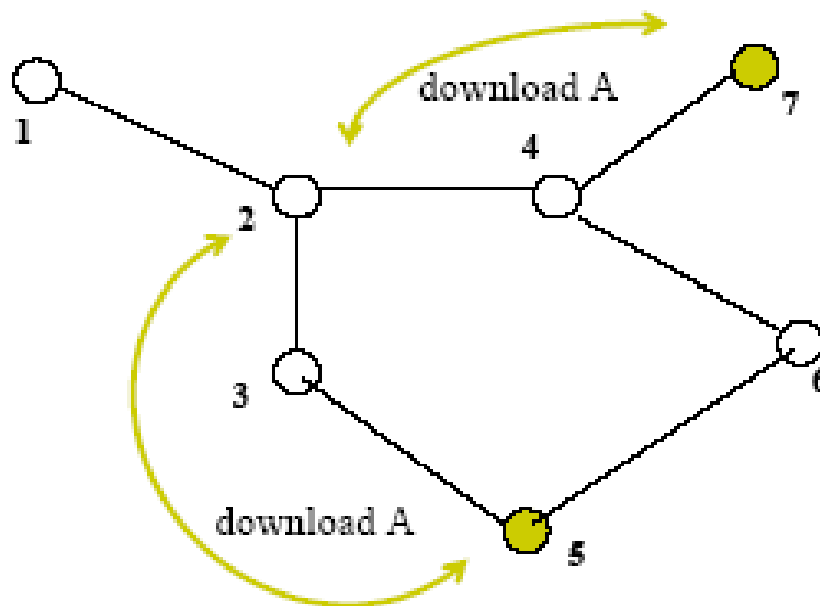
Βήματα:

- Ο κόμβος 2 αρχίζει μια αναζήτηση για το αρχείο A.
- Στέλνει μήνυμα σε όλους τους γείτονες.
- Οι γείτονες προωθούν το μήνυμα.
- Οι κόμβοι που έχουν το αρχείο A στέλνουν απάντηση.
- Η απάντηση διαδίδεται προς τα πίσω (back-propagated).

Gnutella



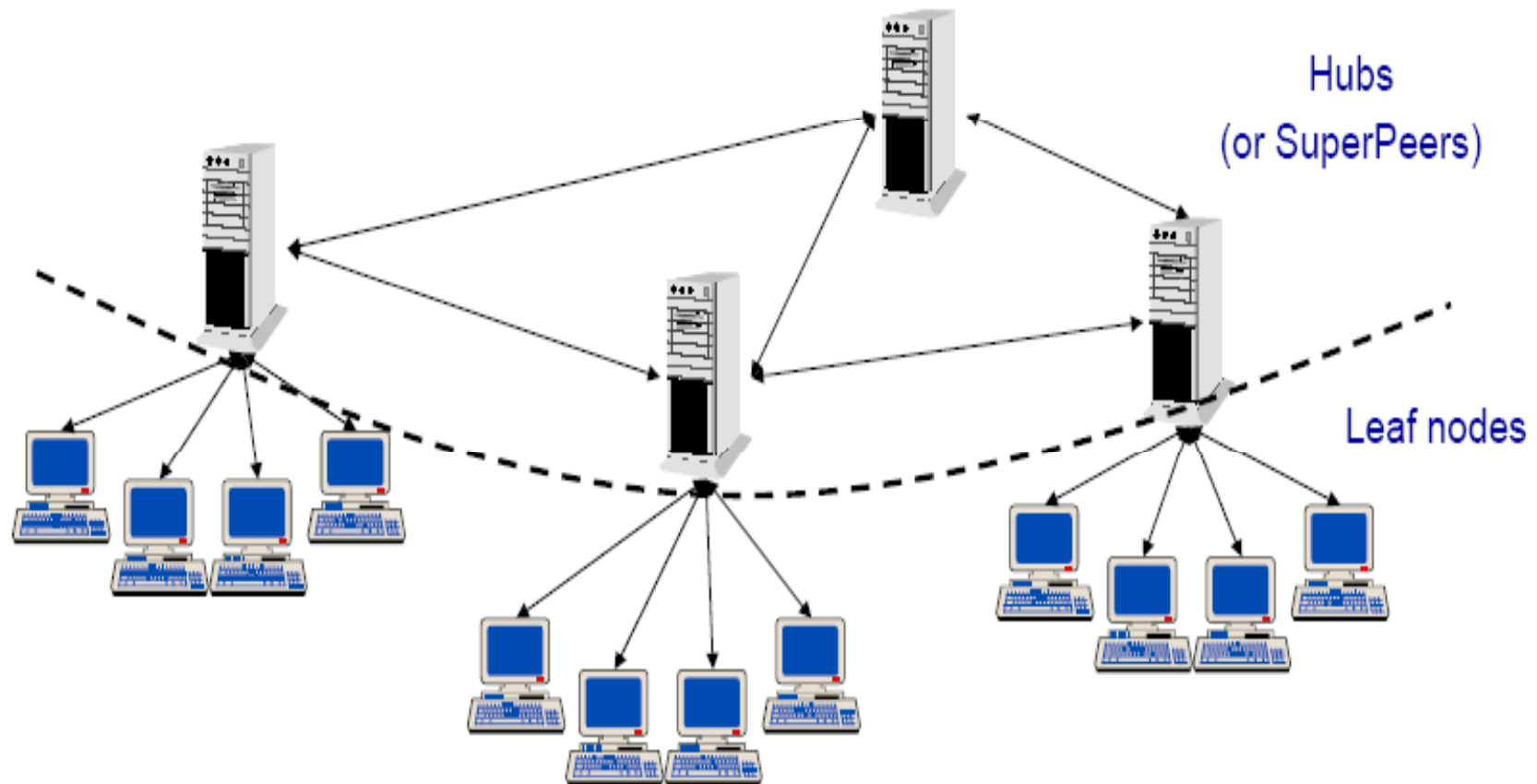
Εντοπισμός πόρων



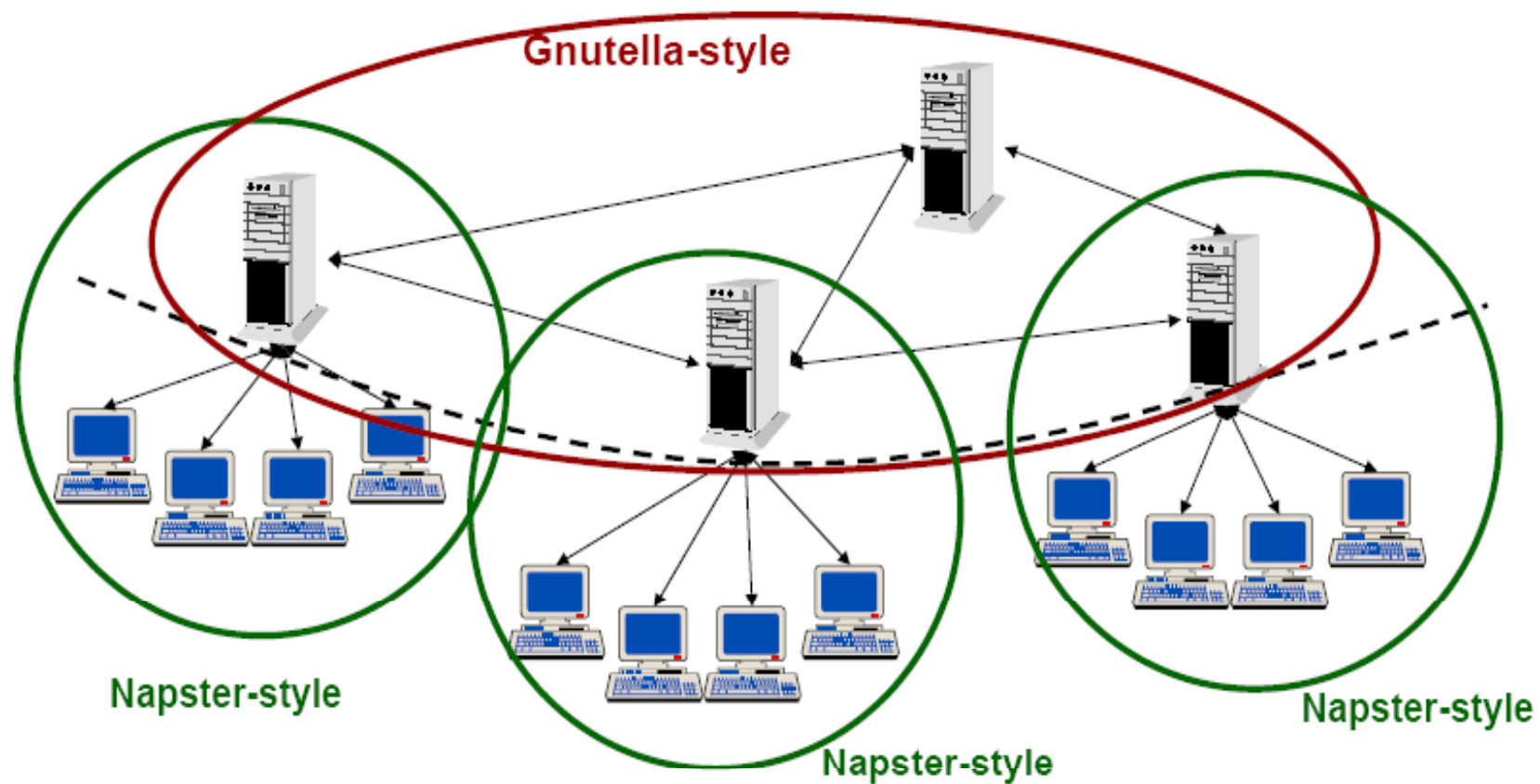
Βήματα:

- Ο κόμβος 2 αρχίζει μια αναζήτηση για το αρχείο A.
- Στέλνει μήνυμα σε όλους τους γείτονες.
- Οι γείτονες προωθούν το μήνυμα.
- Οι κόμβοι που έχουν το αρχείο A στέλνουν απάντηση.
- Η απάντηση διαδίδεται προς τα πίσω (back-propagated).
- Download "A"

Ιεραρχικά Ομότιμα Συστήματα (Hierarchical P2P Systems – π.χ. Kazaa)

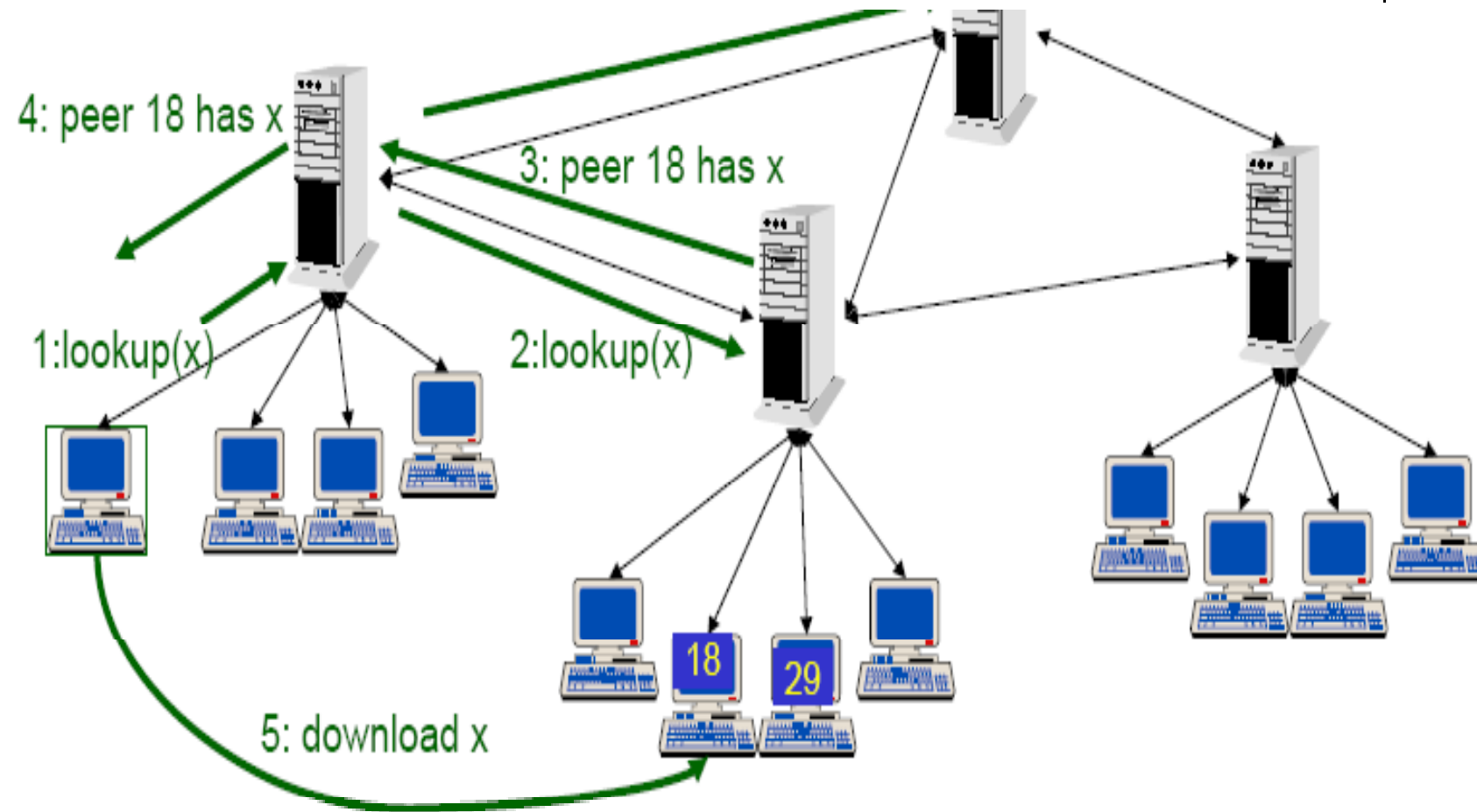


Ιεραρχικά Ομότιμα Συστήματα: *Συνδυασμός υβριδικών και αδόμητων συστημάτων (π.χ. Kazaa)*



Ιεραρχικά Ομότιμα Συστήματα

Εντοπισμός Πόρων



Ιεραρχικά Ομότιμα Συστήματα

Εντοπισμός Πόρων



- Η έννοια του superpeer / supernode
- Κάθε supernode αποθηκεύει καταλόγους όπως οι servers του Napster
- Οι supernodes αλλάζουν
- Οποιοσδήποτε κόμβος μπορεί να γίνει (και να παραμείνει) supernode αρκεί να έχει καλή φήμη (reputation – π.χ. με βάση τη διάρκεια που ο χρήστης είναι συνδεδεμένος, τον αριθμό uploads κ.α. πιο πολύπλοκα)
- Ένας κόμβος αναζητά πάντα μέσω του πιο κοντινού του supernode

BitTorrent

Ξεχωριστή περίπτωση



- Στοχεύει στην αποδοτική μεταφόρτωση (download) των αρχείων παρά στην αναζήτηση - ένας εκδότης, πολλοί downloaders
- Κατανομή του αρχείου σε πολλούς κόμβους ταυτόχρονα
 - Διαίρεση αρχείου σε πολλά κομμάτια (chunks)
 - Replication διαφορετικών κομματιών σε διαφορετικούς κόμβους
 - Οι κόμβοι μπορούν να ανακατασκευάσουν όλο το αρχείο από τα κομμάτια του
- Επιτρέπει ταυτόχρονα downloads
 - Διαφορετικά μέρη του αρχείου από διαφορετικούς κόμβους
 - Ιδιαίτερα αποδοτικό για μεγάλα αρχεία
- Tracker: κόμβος με ειδικό ρόλο (centralized ή distributed)
 - Καταγράφει τους κόμβους που συμμετέχουν στο δίκτυο
 - Ο tracker επιλέγει τους κόμβους για downloading

Δομημένα Ομότιμα Συστήματα (Structured P2P Systems)



Chord, Pastry, Tapestry κ.α.

Σκοπός:

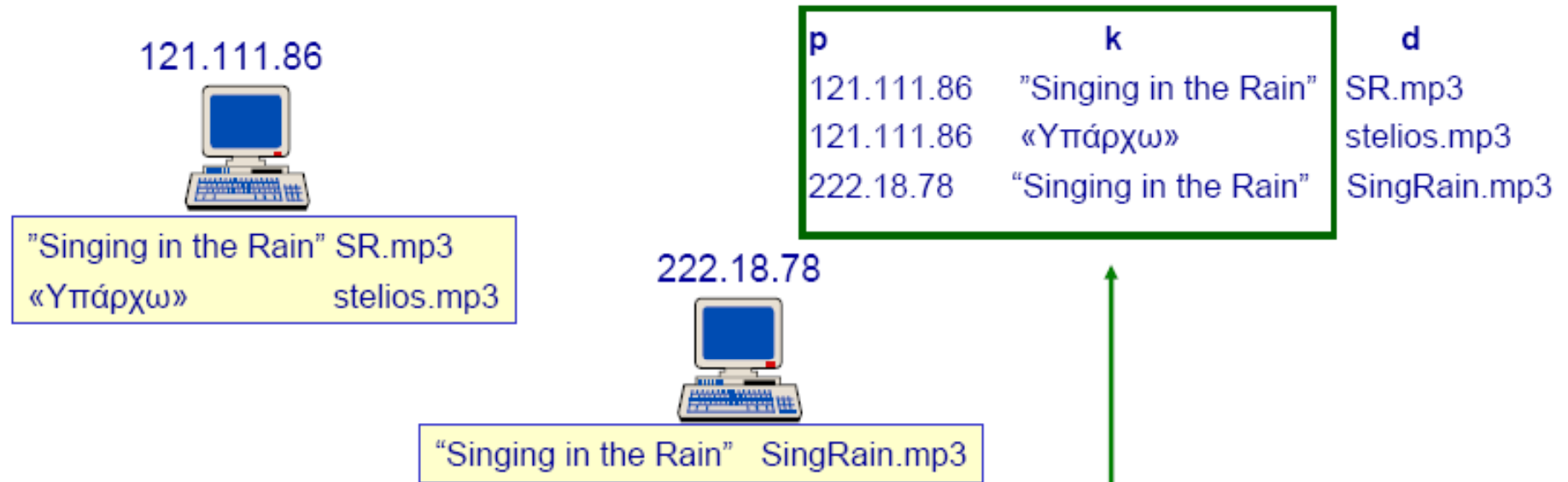
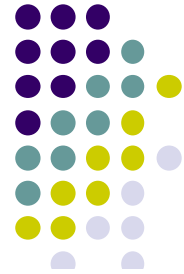
Γρήγορος εντοπισμός πόρων χωρίς τη χρήση
κεντρικού διακομιστή και ανταλλάσσοντας λίγα μηνύματα

Κοινά χαρακτηριστικά των δομημένων ομότιμων συστημάτων

- κάθε κόμβος διατηρεί ένα μικρό τμήμα του καθολικού ευρετηρίου (πίνακας δρομολόγησης)
- οι αναζητήσεις γίνονται με προώθηση μηνυμάτων προς τη «σωστή» κατεύθυνση

Δομημένα Ομότιμα Συστήματα

Εντοπισμός πόρων



Έστω peer με δνση p που αποθηκεύει στοιχείο d που χαρακτηρίζεται από το κλειδί k
Ζητούμενο: Δοθέντος k (ή συνθήκης πάνω στο k) εντόπισε τον peer που έχει το d ,
δηλαδή βρες το ζεύγος ευρετηρίου (k,p) .

(άρα το ευρετήριο μας αποτελείται από ζεύγη της μορφής (k,p))

Κρίσιμο ερώτημα: Πως μπορούμε να (α) **φτιάξουμε**, (β) **συντηρήσουμε** και (γ) να **χρησιμοποιήσουμε** ένα τέτοιο ευρετήριο χωρίς κεντρικό έλεγχο;



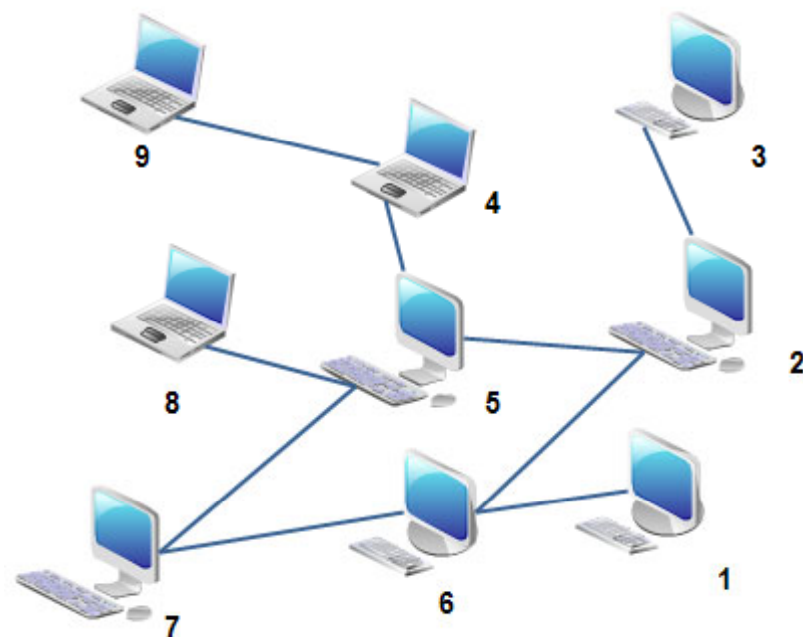
Επικοινωνία σε δίκτυο P2P

- **Ταυτοποίηση μελών:** Πρόκειται για διαδικασία που επιτρέπει το διαχωρισμό μελών μεταξύ τους. Αυτό γίνεται μέσω του ορισμού ενός μοναδικού προσδιοριστή GUID (Globally Unique ID).
- **Εντοπισμός μελών:** πρόκειται για τον εντοπισμό ενός μέλους με συγκεκριμένο GUID μέσα στο δίκτυο. Μέλη του δικτύου μπορεί να συνδέονται και να αποσυνδέονται διαρκώς. Αν όλα τα μέλη γνώριζαν όλα τα άλλα τότε η διατήρηση της πληροφορίας αυτής θα ήταν αδύνατη.



Αντιστοίχιση GUID

- GUID είναι ένα string με $m=64$ ή 128 bits
- Αντιστοίχιση GUID
 - Τα μέλη παίρνουν ένα GUID αφού συνδεθούν

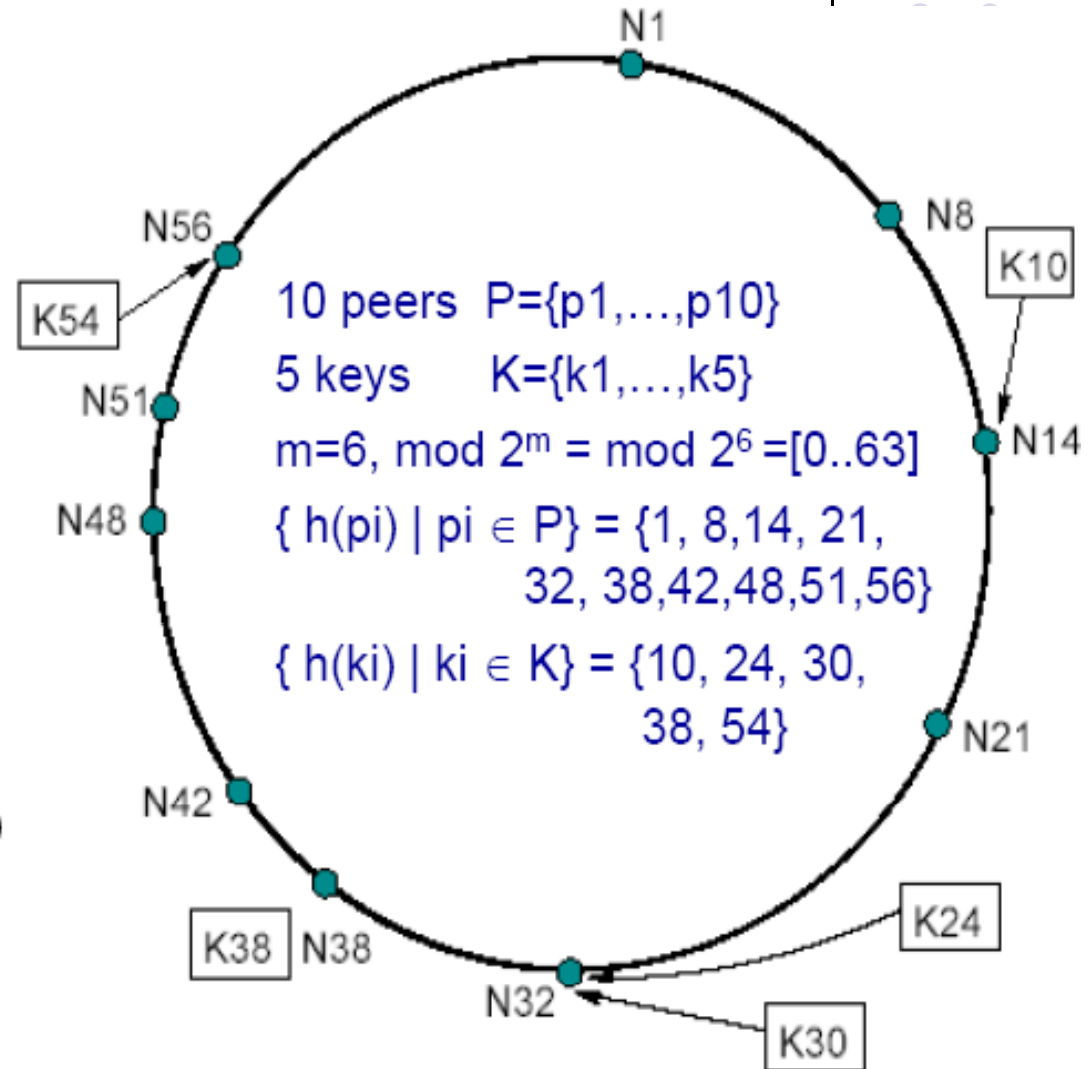


Chord

Χρήση Distributed Hash Tables (DHT)



- Κατακερματισμός (Hashing)
κλειδιών (k) και διευθύνσεων (p)
σε δυαδικά κλειδιά με m -bits
 - π.χ. $m=6$, $h(\text{«υπάρχω»})=11$,
 $h(196.178.0.1)=3$
- Τα δυαδ. κλειδιά τοποθετούνται
σε έναν κύκλο modulo 2^m
 - Για $m=8$, κυκλική διάταξη των
αριθμών $0 \dots 255$
- Ένα κλειδί k εκχωρείται στον
πρώτο κόμβο p τ.ω. $h(p) \geq h(k)$
- Αυτός ο κόμβος λέγεται
successor(k)



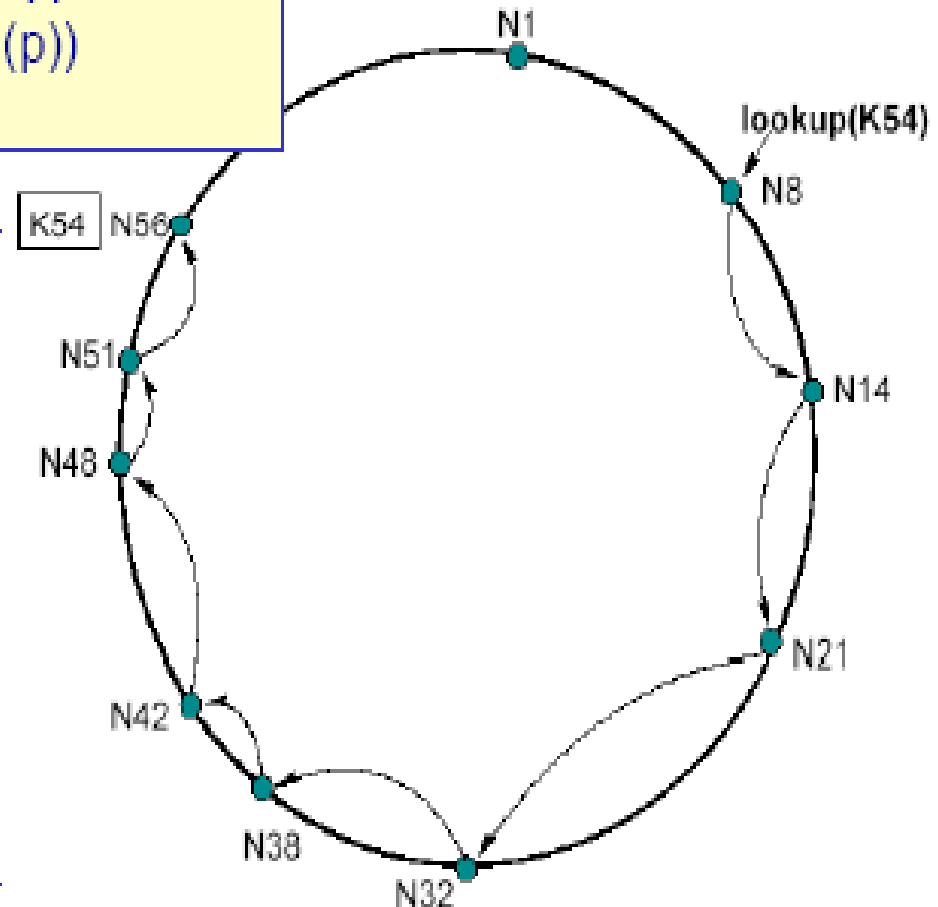
Chord

Απλός τρόπος εντοπισμού κόμβων



Έστω ότι κάθε κόμβος p ξέρει την δνση μόνο του επόμενου του (του p' με $h(p') > h(p)$)

```
// ask node n to find the successor of id
n.find_successor(id)
  if (id in (n; successor])
    return successor;
  else
    // forward the query around the circle
    return successor.find_successor(id);
```

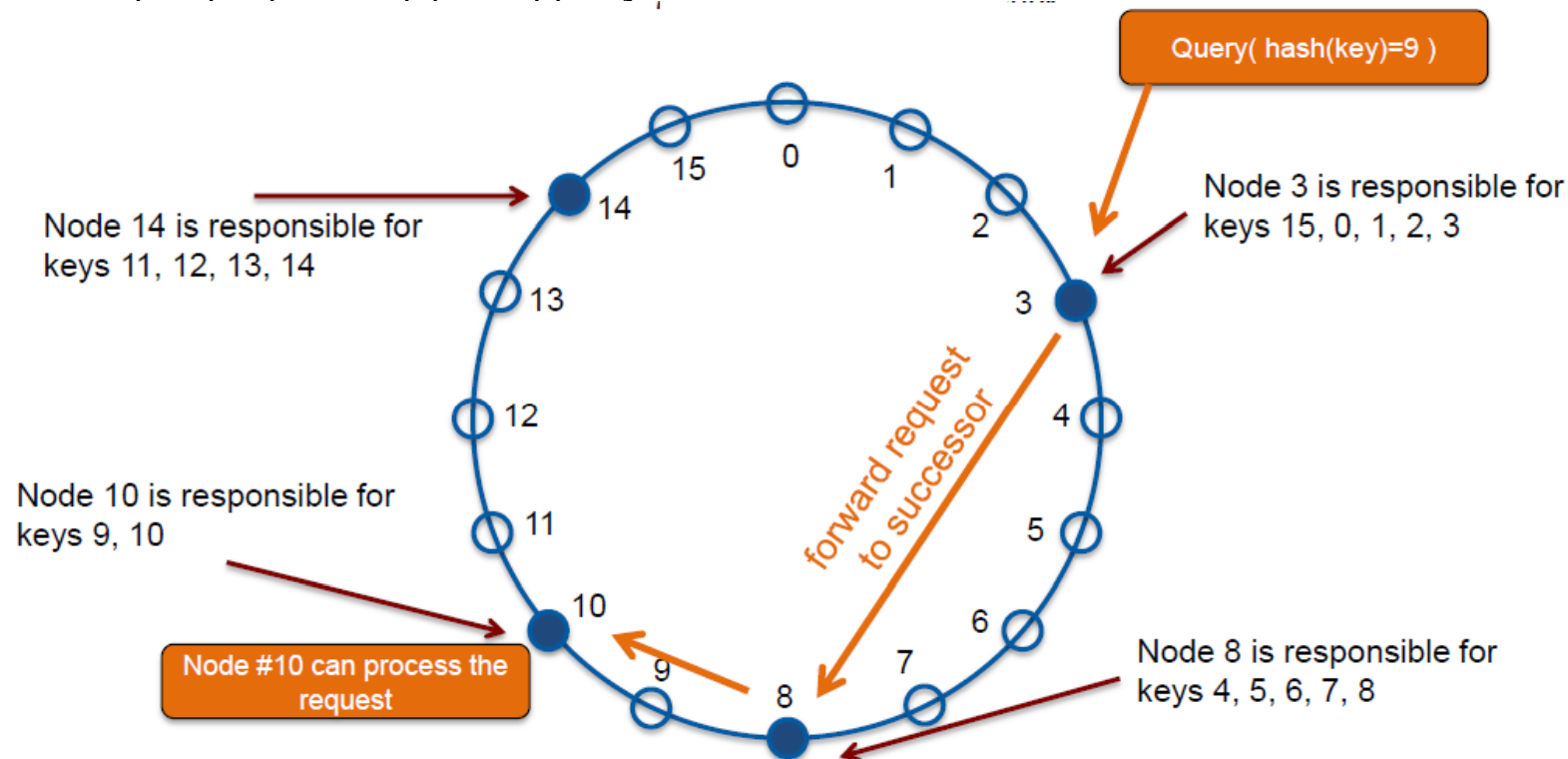


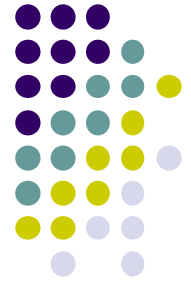
=> Number of messages linear in the number of nodes !

Απλός τρόπος εντοπισμού κόμβων (συν.)



- Τα δεδομένα αποθηκεύονται στον διάδοχο κόμβο (**successor**). Κάθε κόμβος είναι υπεύθυνος για ένα σύνολο (εύρος) κλειδιών.
- Οποιοσδήποτε κόμβος μπορεί να λάβει ερώτημα για ένα key (insert ή query). Αν το $\text{hash}(\text{key})$ δεν ανήκει στο εύρος κλειδιών του, το προωθεί στον διάδοχο
- Η διαδικασία συνεχίζεται μέχρι να βρεθεί ο υπεύθυνος κόμβος
 - Η χειρότερη περίπτωση: με p κόμβους, διασχίζει $p-1 \rightarrow O(N)$
 - Η μέση περίπτωση: $p/2$ κόμβους





Βήματα σύνδεσης σε δίκτυο

- Ένας κόμβος δημιουργεί το δίκτυο (boot)
- Ο κόμβος που συνδέεται στέλνει ένα μήνυμα "join" στον αρχικό κόμβο (boot peer), και παίρνει ένα GUID.
- Ο κόμβος που συνδέθηκε ζητά αντιγραφή πίνακα δρομολόγησης από τον κόμβο boot.
- Ο κόμβος που συνδέθηκε διορθώνει τον πίνακα δρομολόγησης με τους σωστούς κόμβους για αυτόν.

Peer Routing Table

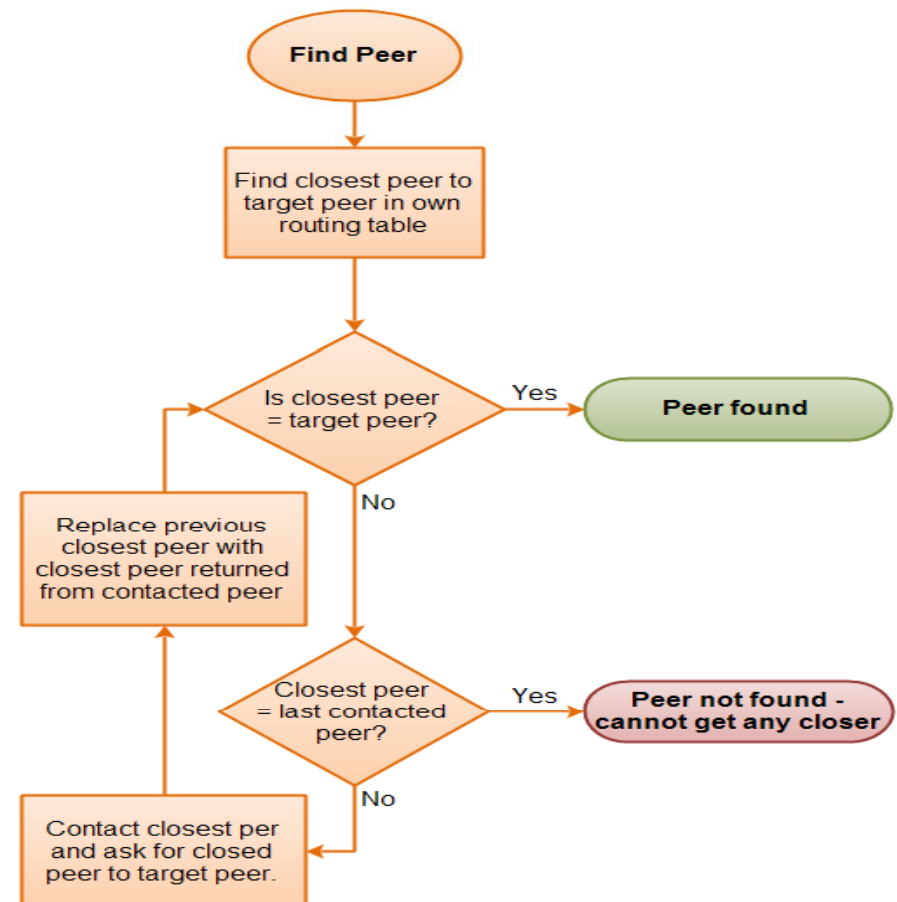


- Οι εγγραφές είναι της μορφής (GUID, IP Address, TCP Port)
- Ένα μέλος διαλέγει ποια άλλα μέλη να αντιστοιχεί με βάση την απόσταση από το δικό του GUID, και το GUID του κόμβου που συνδέεται.
- Ο πίνακας δρομολόγησης έχει τόσα κελιά όσα τα bits του GUID. Άρα αν το GUID είναι 64 bit, έχουμε 64 κελιά στον πίνακα δρομολόγησης. Επομένως με 64 bit "GUID χώρο" έχουμε δυνατότητα για 2^{64} GUID's (peers) και κάθε κόμβος χρειάζεται ένα πίνακα 64 θέσεων.

Εύρεση κόμβων



- `targetPeer = 14; // or some other number.`
`closestPeer`
`=routingTable.findClosest(targetGUID);`
`prevClosestPeer = null;`
`while(closestPeer != targetPeer &&`
`closestPeer != prevClosestPeer) {`
`prevClosestPeer = closestPeer;`
`closestPeer = askForClosestPeer(closestPeer,`
`targetPeer); }`
`if(closestPeer == targetPeer) { // found } else {`
`// not found,`
`closestPeer = prevClosestPeer }`



Chord

Εντοπισμός πόρου με χρήση πίνακα δρομολόγησης



- Επιπλέον πληροφορία δρομολόγησης για επιτάχυνση
- Κάθε κόμβος n έχει έναν **πίνακα δρομολόγησης** με m εγγραφές
 - οι m αυτοί κόμβοι έχουν εκθετικά αυξανόμενη απόσταση από τον n
- Η i εγγραφή του πίνακα έχει την δ νση του πρώτου κόμβου με κλειδί μεγαλύτερο ή ίσο με $n + 2^{i-1}$

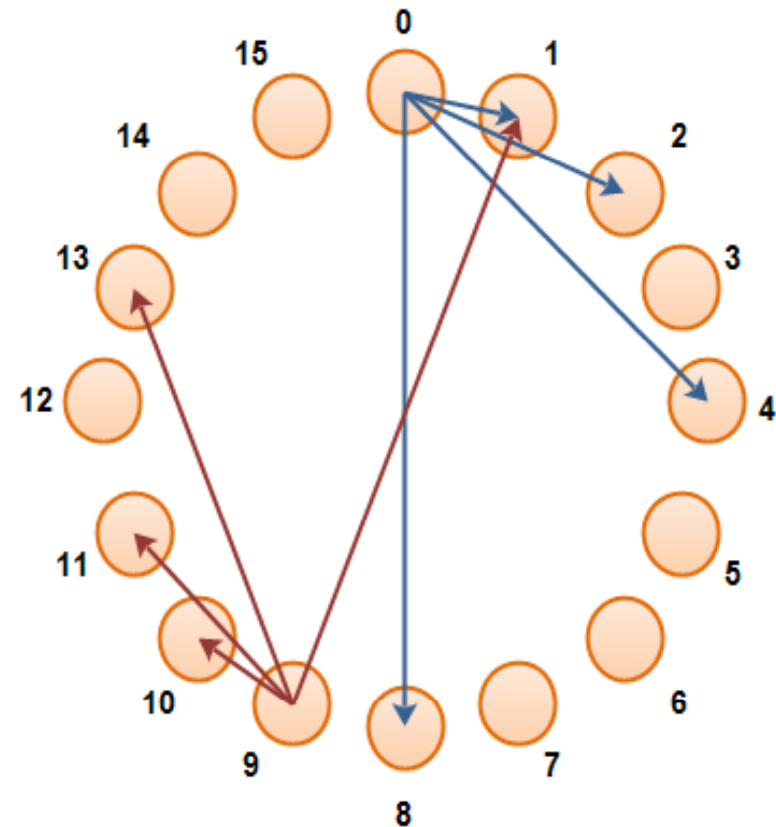
$$finger[i] = successor(n + 2^{i-1})$$

Παράδειγμα του finger routing table για $n=0$



A Peer Routing Table for Peer with GUID 0 (4 bit GUID size)

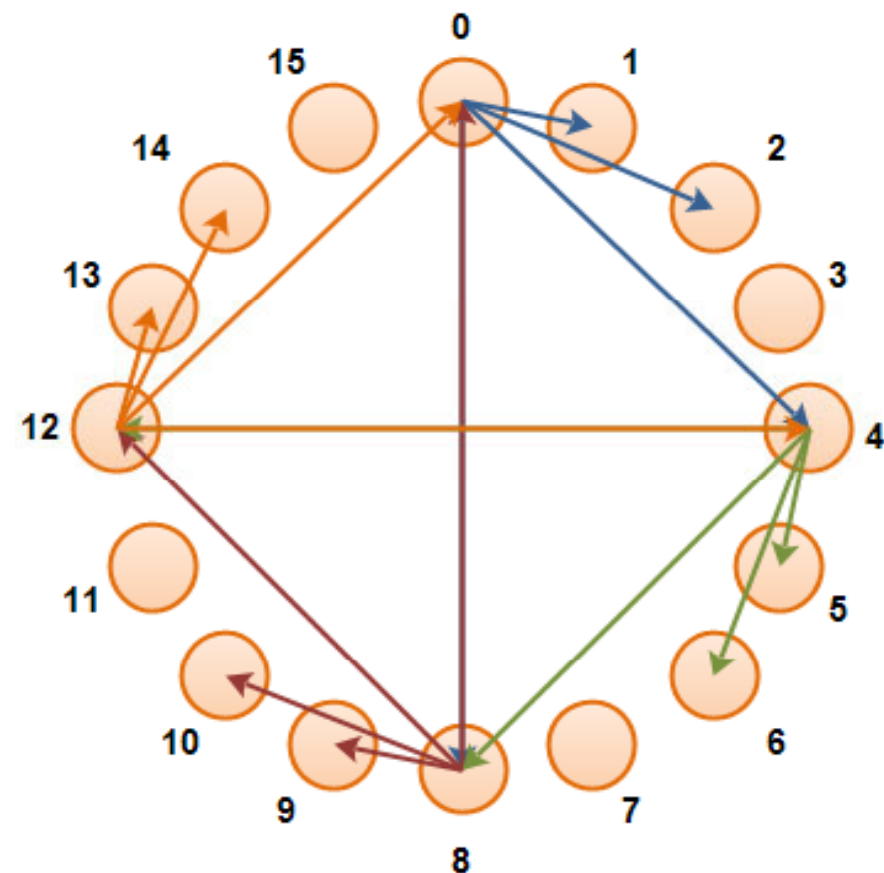
Cell Index	Referenced GUID	GUID Distance
0	1	1 (2^0)
1	2	2 (2^1)
2	4	4 (2^2)
3	8	8 (2^3)



Παράδειγμα finger routing table m=4



Chord				
Peer GUID	$2^0 = 1$	$2^1 = 2$	$2^2 = 4$	$2^3 = 8$
0	1	2	4	8
1	2	3	5	9
2	3	4	6	10
3	4	5	7	11
4	5	6	8	12
5	6	7	9	13
6	7	8	10	14
7	8	9	11	15
8	9	10	12	0
9	10	11	13	1
10	11	12	14	2
11	12	13	15	3
12	13	14	0	4
13	14	15	1	5
14	15	0	2	6
15	0	1	3	7



Chord

Εντοπισμός πόρου με χρήση πίνακα δρομολόγησης



Finger table: $m=6$

$$\text{finger}[i] = \text{successor}(n + 2^{i-1})$$

$n=8$

$$\text{finger}[1] = \text{succ}(n+1) = \text{succ}(9) = 14$$

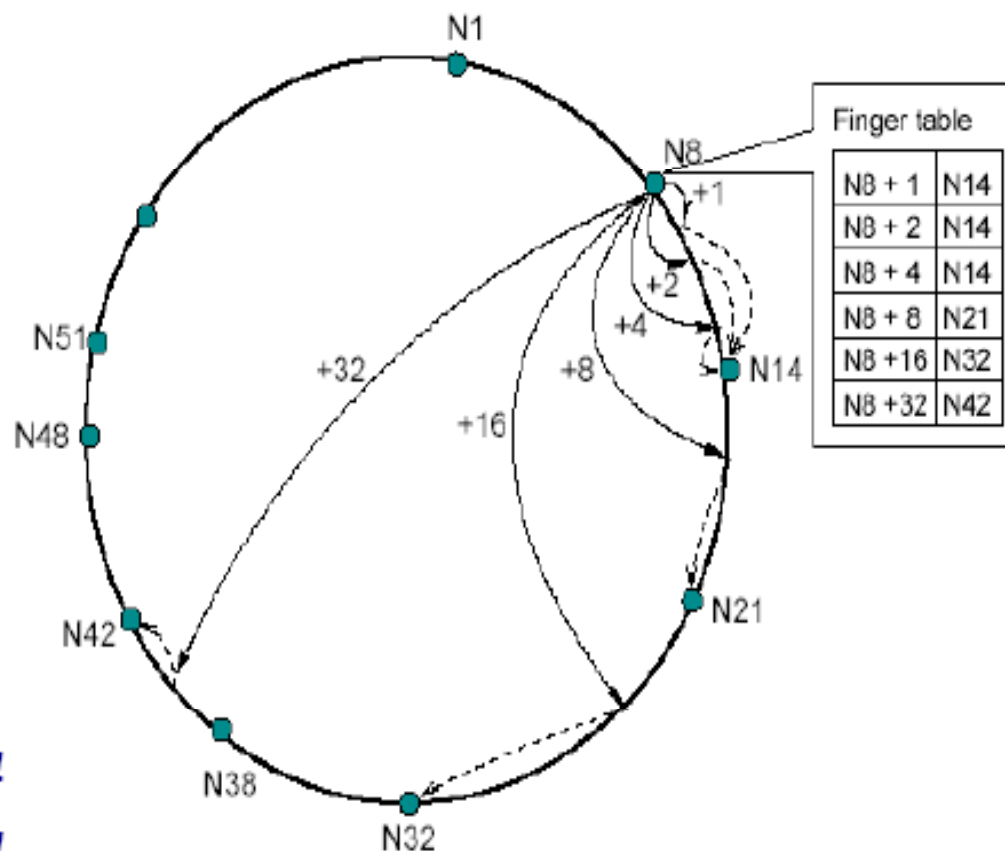
$$\text{finger}[2] = \text{succ}(n+2) = \text{succ}(10) = 14$$

$$\text{finger}[3] = \text{succ}(n+4) = \text{succ}(12) = 14$$

$$\text{finger}[4] = \text{succ}(n+8) = \text{succ}(16) = 21$$

$$\text{finger}[5] = \text{succ}(n+16) = \text{succ}(24) = 32$$

$$\text{finger}[6] = \text{succ}(n+32) = \text{succ}(40) = 42$$



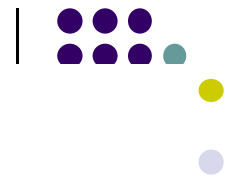
Chord

Εντοπισμός πόρου με χρήση πίνακα δρομολόγησης



- ❑ Αν δεν έχεις εσύ την ίδια ακριβώς ταυτότητα με το προς εντοπισμό κλειδί (οπότε έχεις και το κλειδί), ψάξε στον πίνακα δρομολόγησης να βρεις τον πλησιέστερο ‘προηγούμενο’ κόμβο σε σχέση με το ‘κλειδί’.
- ❑ Συνέχισε (επανέλαβε) τη διαδικασία από αυτόν τον κόμβο.
- ❑ Αν είσαι εσύ ο πλησιέστερος ‘προηγούμενος’ επέστρεψε τον αμέσως ‘επόμενό’ σου κόμβο.

Αφού οι εγγραφές των πινάκων δρομολόγησης είναι εκθετικά αύξουσες, η αναζήτηση λαμβάνει λογαριθμικό χρόνο.



Search Process

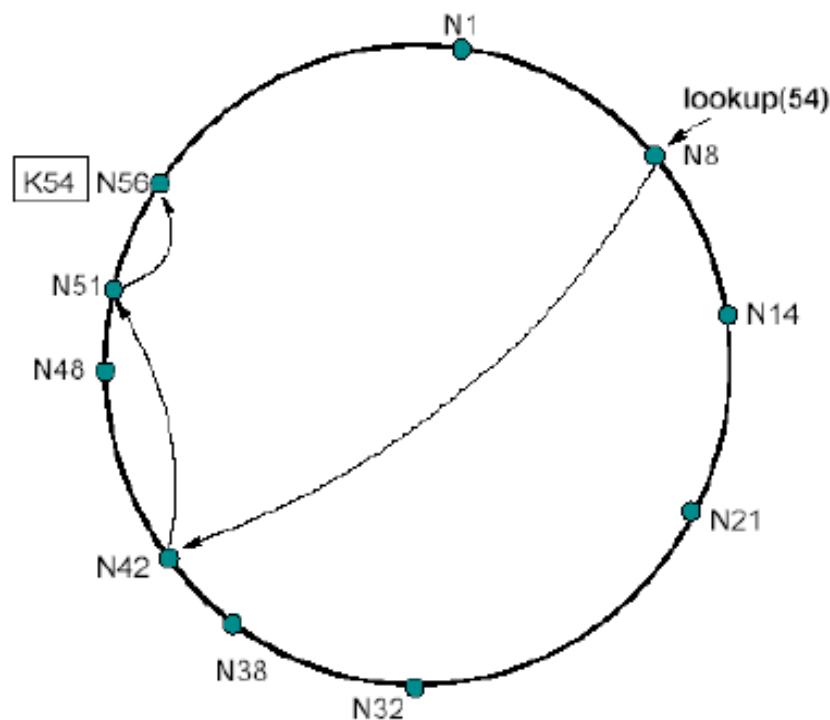
```
n.find_successor(id)           //ask node n to find id's successor
    n' = find_predecessor(id);
    return n'.successor;

n.find_predecessor(id)         // ask node n to find id's predecessor
    n' = n;
    while(id  $\notin$  (n', n'.successor])
        n' = n'.closest_preceding_finger(id);
    return n';

n.closest_preceding_finger(id) //return closest finger preceding id
for i=m downto 1
    if(finger[i].node  $\in$  (n, id))
        return finger[i].node;
return n;
```


Chord

Εντοπισμός πόρου με χρήση πίνακα δρομολόγησης



N8

Finger table	
$N8 + 1$	N14
$N8 + 2$	N14
$N8 + 4$	N14
$N8 + 8$	N21
$N8 + 16$	N32
$N8 + 32$	N42

N42

Finger table	
$N42 + 1$	N48
$N42 + 2$	N48
$N42 + 4$	N48
$N42 + 8$	N51
$N42 + 16$	N1
$N42 + 32$	N14

Εύρεση με ανταλλαγή τριών μηνυμάτων

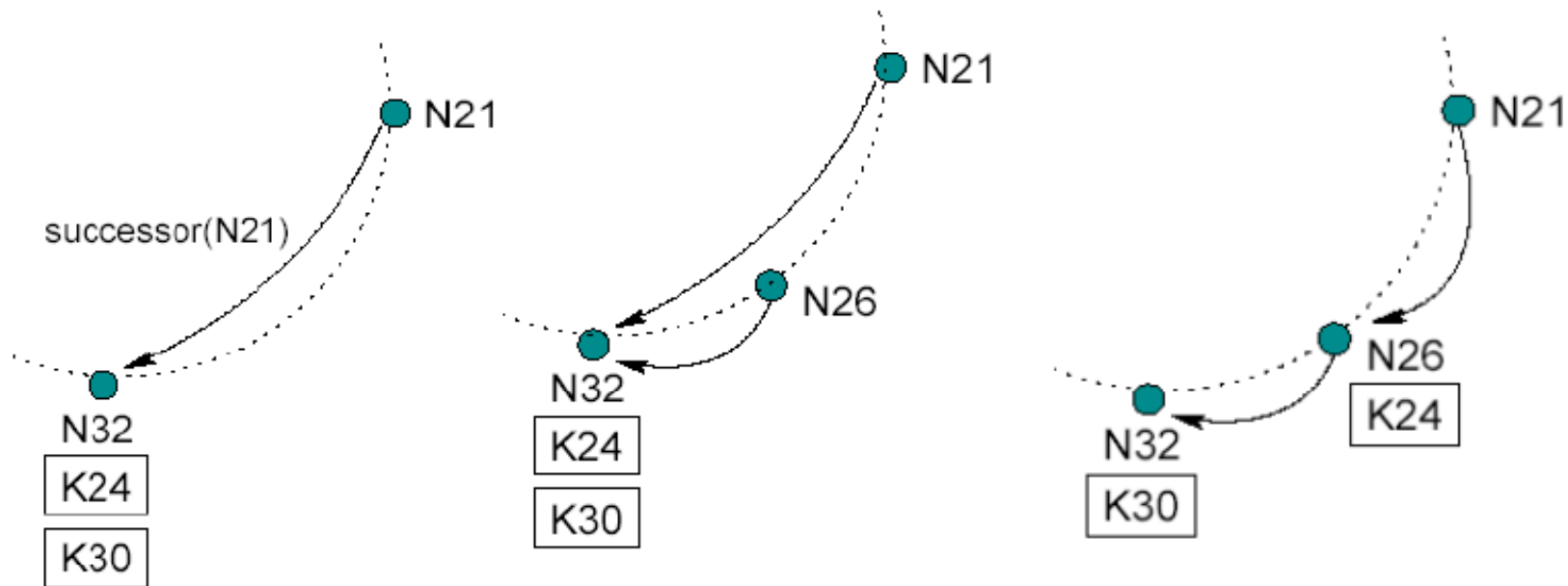
- Search in finger table for the nodes which most immediately precedes id
- Invoke `find_successor` from that node
- \Rightarrow Number of messages $O(\log N)$

Chord

Εντοπισμός πόρου με χρήση πίνακα δρομολόγησης – Εισαγωγή νέου κόμβου



- Ο νέος κόμβος πρέπει να φτιάξει τον πίνακα δρομολόγησης του
- Το κόστος κατασκευής του είναι αυτό της αναζήτησης
- Οι άλλοι κόμβοι πρέπει να ενημερώσουν τους δικούς τους πίνακες



Chord



Βασικά σημεία:

- Κάθε κόμβος αποθηκεύει πληροφορία για μικρό αριθμό κόμβων (m)
- Κάθε κόμβος ξέρει περισσότερα για τους κοντινούς του όρους (απ'ότι για τους μακρινούς)

Επιδόσεις

- Χρόνος αναζήτησης: $O(\log n)$ (με μεγάλη πιθανότητα)
- Πλήθος Μηνυμάτων: $O(\log n)$ (επιλεκτική δρομολόγηση)
- Κόστος αποθήκευσης: $O(\log n)$ (πίνακας δρομολόγησης)
- Κόστος εισόδου/εξόδου κόμβου: $O(\log^2 n)$
- Κόστος ενημέρωσης: μικρό (περίπου σαν το κόστος αναζήτησης)

Chord software

- 3000 lines of C++ code, Library to be linked with the application, provides a `lookup(key)` – function: yields the IP address of the node responsible for the key. Notifies the node of changes in the set of keys the node is responsible for

Wireless Sensor Networks – Ασύρματα Δίκτυα Αισθητήρων



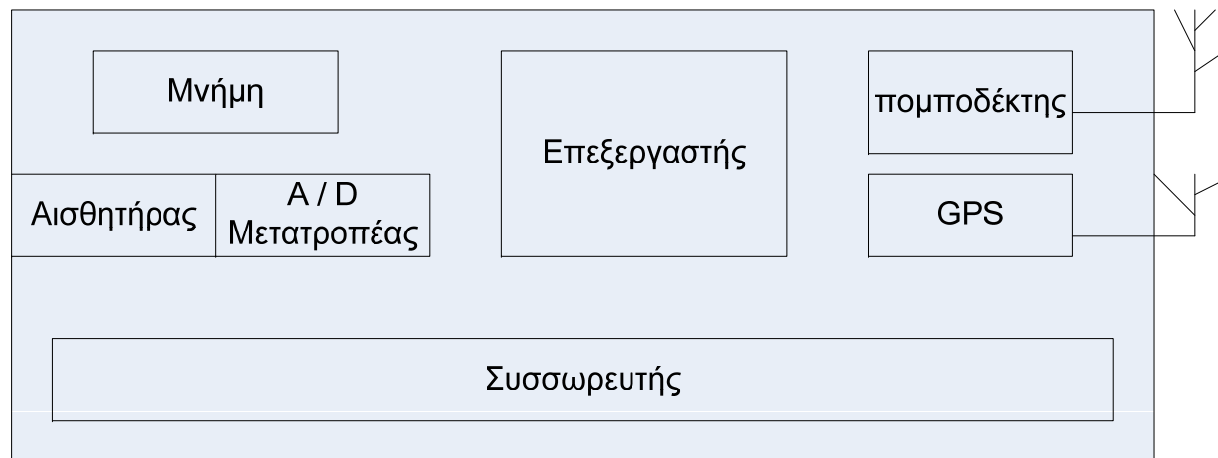
- Ένα δίκτυο αισθητήρων αποτελείται από ένα μεγάλο πλήθος κόμβων αισθητήρων,
 - Συνήθως τοποθετούνται σε πυκνή διάταξη είτε μέσα στο φαινόμενο προς παρατήρηση είτε πολύ κοντά σε αυτό.
- Οι κόμβοι περιλαμβάνουν:
 - τμήματα που μετρούν διάφορα φυσικά μεγέθη (θερμοκρασία, πίεση κτλ.)
 - τμήματα που επεξεργάζονται τις μετρήσεις
 - τμήματα υπεύθυνα για τη μετάδοση αυτών των μετρήσεων ή για τη μεταγωγή δεδομένων από άλλους κόμβους
- Οι κόμβοι είναι χαμηλού κόστους και έχουν περιορισμένα ενεργειακά αποθέματα.

Τα επιμέρους στοιχεία ενός αισθητήρα

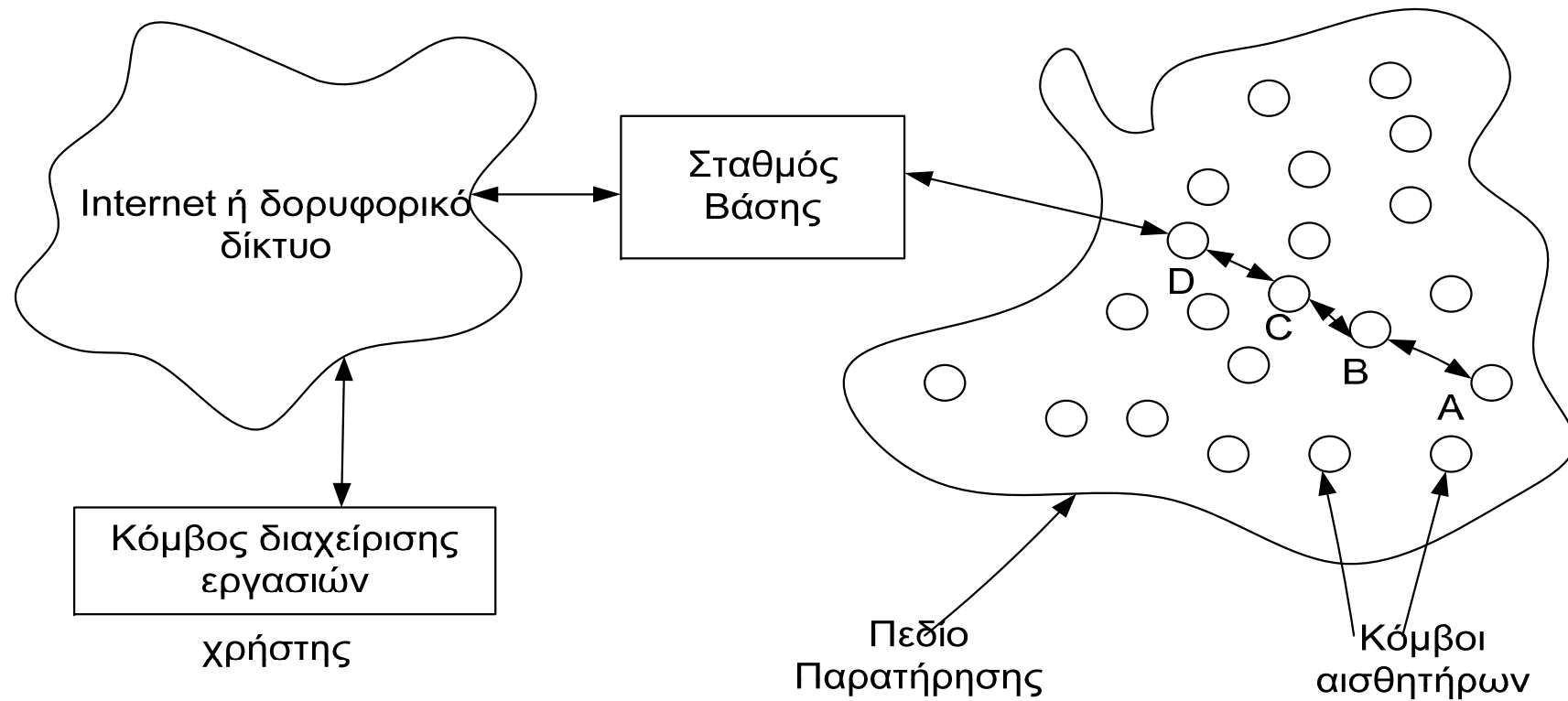


Στη βασική αρχιτεκτονική ενός κόμβου περιλαμβάνονται τα παρακάτω βασικά υποσυστήματα:

- Υποσύστημα αισθητήρων (αισθητήρας-μετατροπέας A/D)
- Υποσύστημα επεξεργασίας (μνήμη)
- Υποσύστημα επικοινωνιών
- Υποσύστημα τροφοδοσίας



Αρχιτεκτονική WSN





Wireless Sensor Networks

Sensors for a wide variety of conditions:

- temperature
- motion
- noise levels
- light conditions
- humidity
- object presence
- mechanical stress levels on attached objects

Applications



Environmental applications

- **forest fire detection**
- **flood detection**
- **precision agriculture**
- **weather forecast**

Health applications

- **telemonitoring of human physiological data**
- **tracking and monitoring doctors/patients**

Home applications

- **home automation (local/remote management of home devices)**
- **smart environments (adapt to the people's needs)**

More applications

- **security / military**
- **vehicle tracking and detection**

Sensor Networks

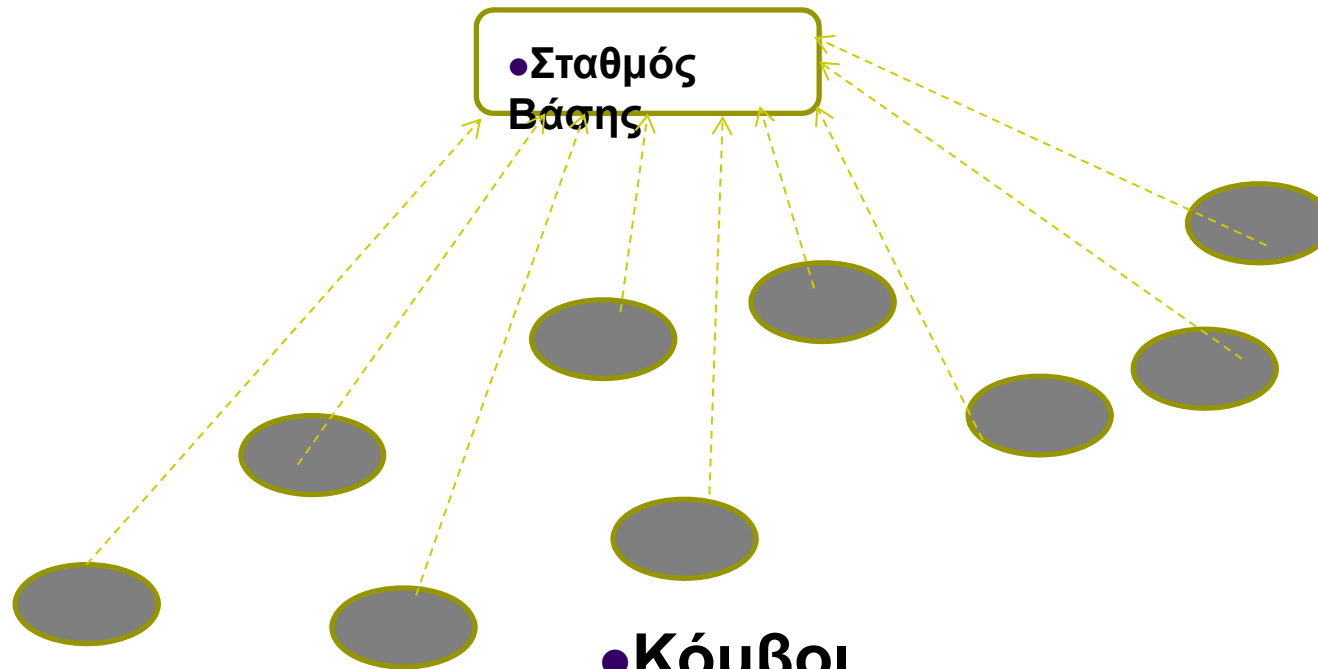


- **Questions concerning sensor networks:**
- **How do we (dynamically) set up an efficient tree in a sensor network?**
- **How does aggregation of results take place? Can it be controlled?**
- **What happens when network links fail?**

Μετάδοση δεδομένων από το δίκτυο στο σταθμό βάσης

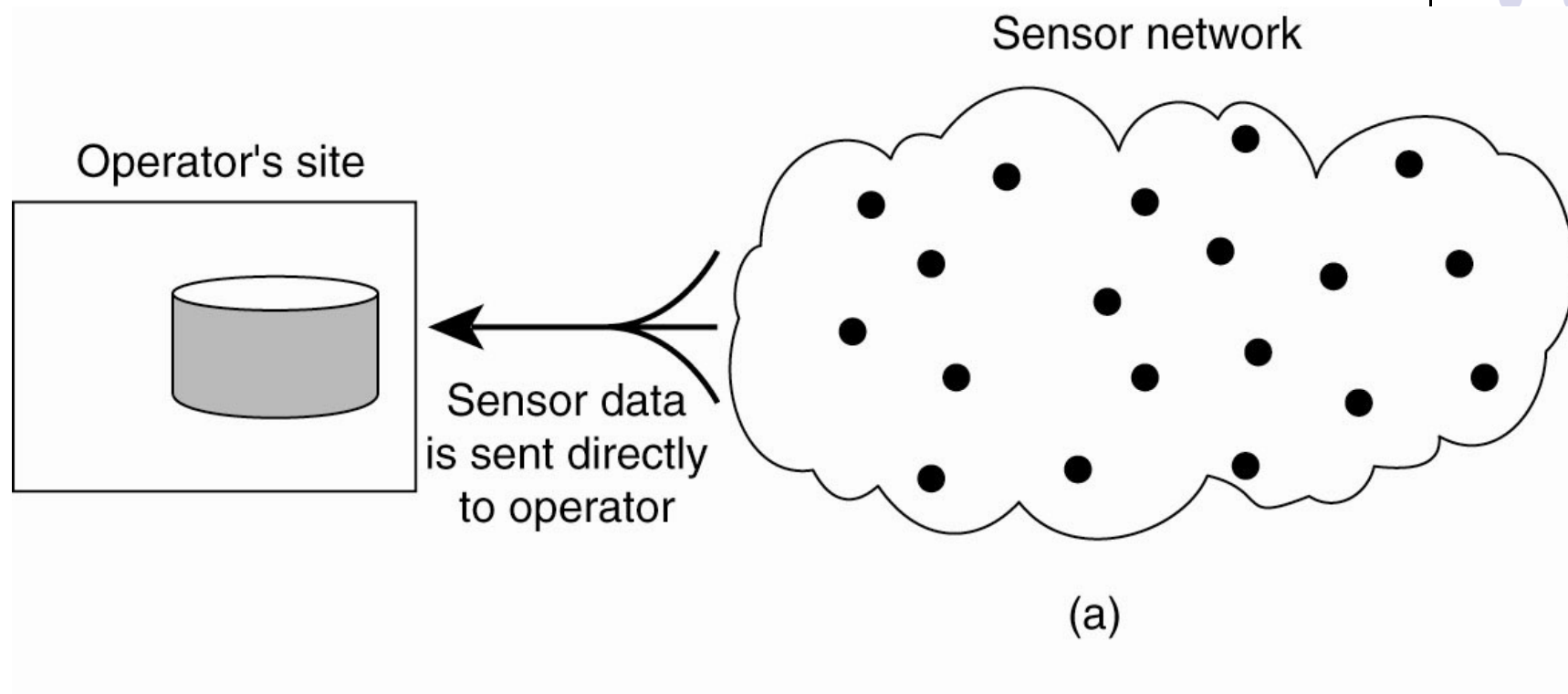


- Απευθείας επικοινωνία:



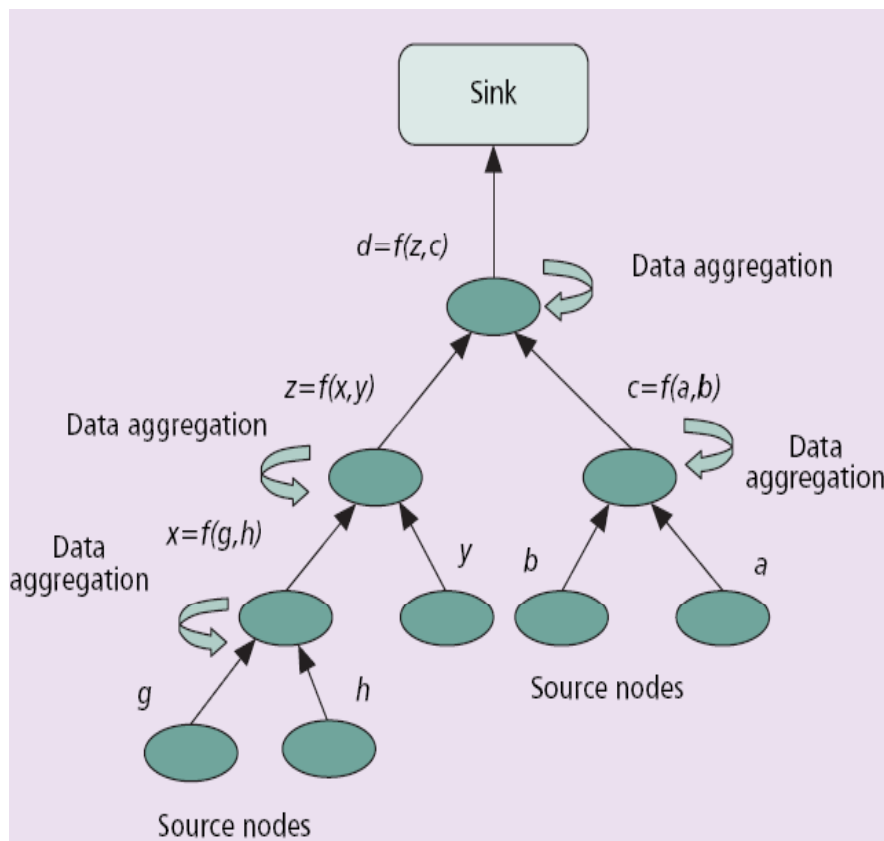
- **Κόμβοι αισθητήρων**
- Μεγάλο το ενεργειακό κόστος
- Πολλές φορές ανέφικτη: μεγάλη απόσταση από το σταθμό βάσης
- Πολλές οι παρεμβολές μεταξύ παράλληλων μεταδόσεων

Sensor Networks



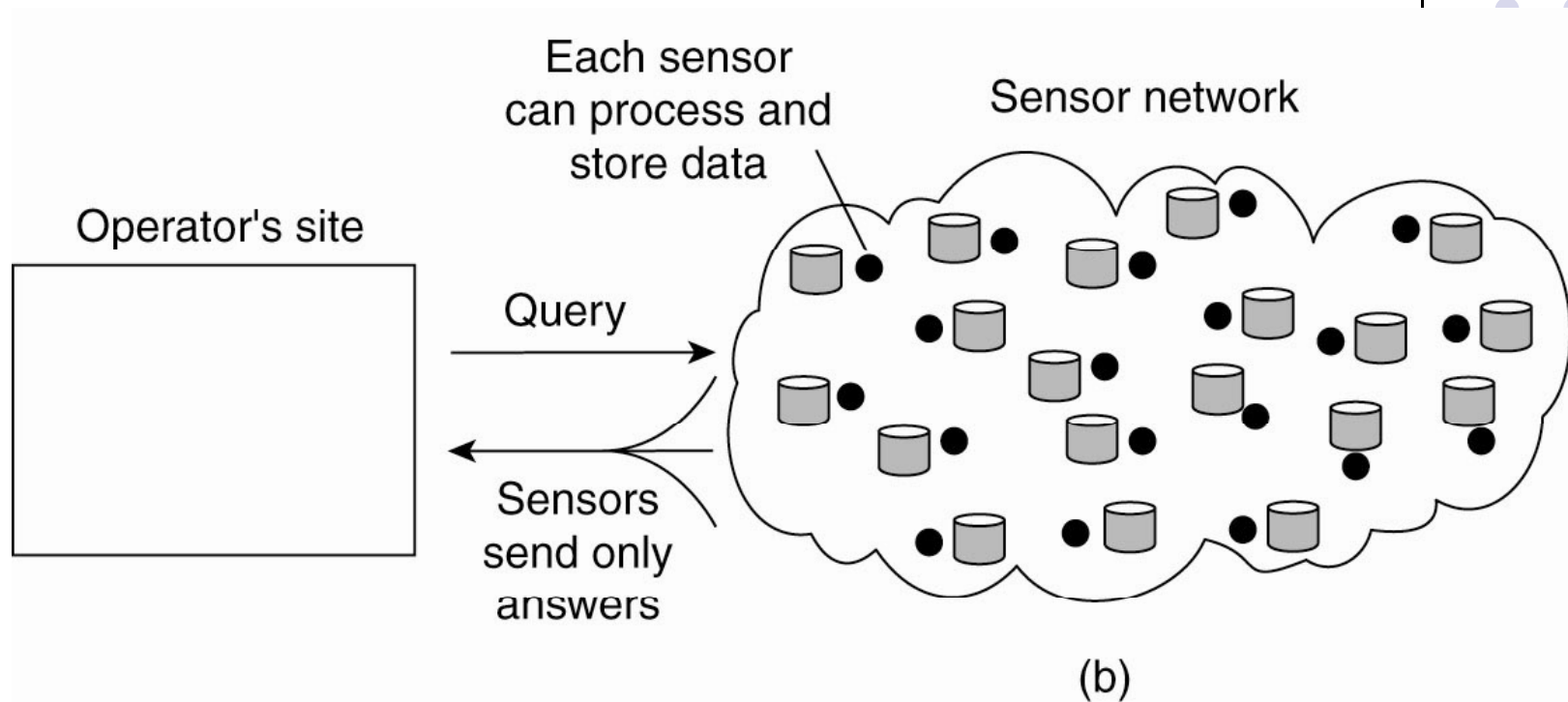
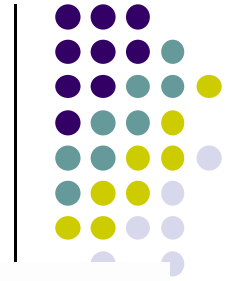
Organizing a sensor network database, while storing and processing data (a) only at the operator's site

Μετάδοση δεδομένων μέσω άλλων κόμβων



- Εφικτή λύση
- Κάθε κόμβος στέλνει:
 - τα δικά του δεδομένα
 - τα δεδομένα άλλων κόμβων που λαμβάνει
- Συνήθως επεξεργασία σε κάθε κόμβο για μείωση της διακινούμενου όγκου δεδομένων
- Data aggregation (Συνάθροιση δεδομένων):
 - απλές αριθμητικές λειτουργίες όπως άθροισμα των δεδομένων εισόδου, μέγιστο, ελάχιστο κτλ.
- Data fusion (Σύμφυση δεδομένων):
 - χρήση τεχνικών επεξεργασίας σήματος ή τεχνικών συμπίεσης για τη μείωση του όγκου δεδομένων

Sensor Networks



Organizing a sensor network database, while storing and processing data only at the sensors.