

# Κινητός και Διάχυτος Υπολογισμός (Mobile & Pervasive Computing)

Δημήτριος Κατσάρος, Ph.D.

Χειμώνας 2005

Διάλεξη 2η

# Ιστοσελίδα του μαθήματος

- [http://skyblue.csd.auth.gr/~dimitris/courses/mpc\\_fall05.htm](http://skyblue.csd.auth.gr/~dimitris/courses/mpc_fall05.htm)
- Θα τοποθετούνται οι διαφάνειες του επόμενου μαθήματος
- Σταδιακά θα τοποθετηθούν και τα research papers που αντιστοιχούν σε κάθε διάλεξη

# Περιεχόμενα

- **Διάχυτος Υπολογισμός**
- Περίληψη των Ασύρματων Επικοινωνιών
- Κινητός Υπολογισμός
- Αρχιτεκτονικές Λογισμικού για Κινητό Υπολογισμό

# Το όραμα του Mark Weiser ...

- “The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.”

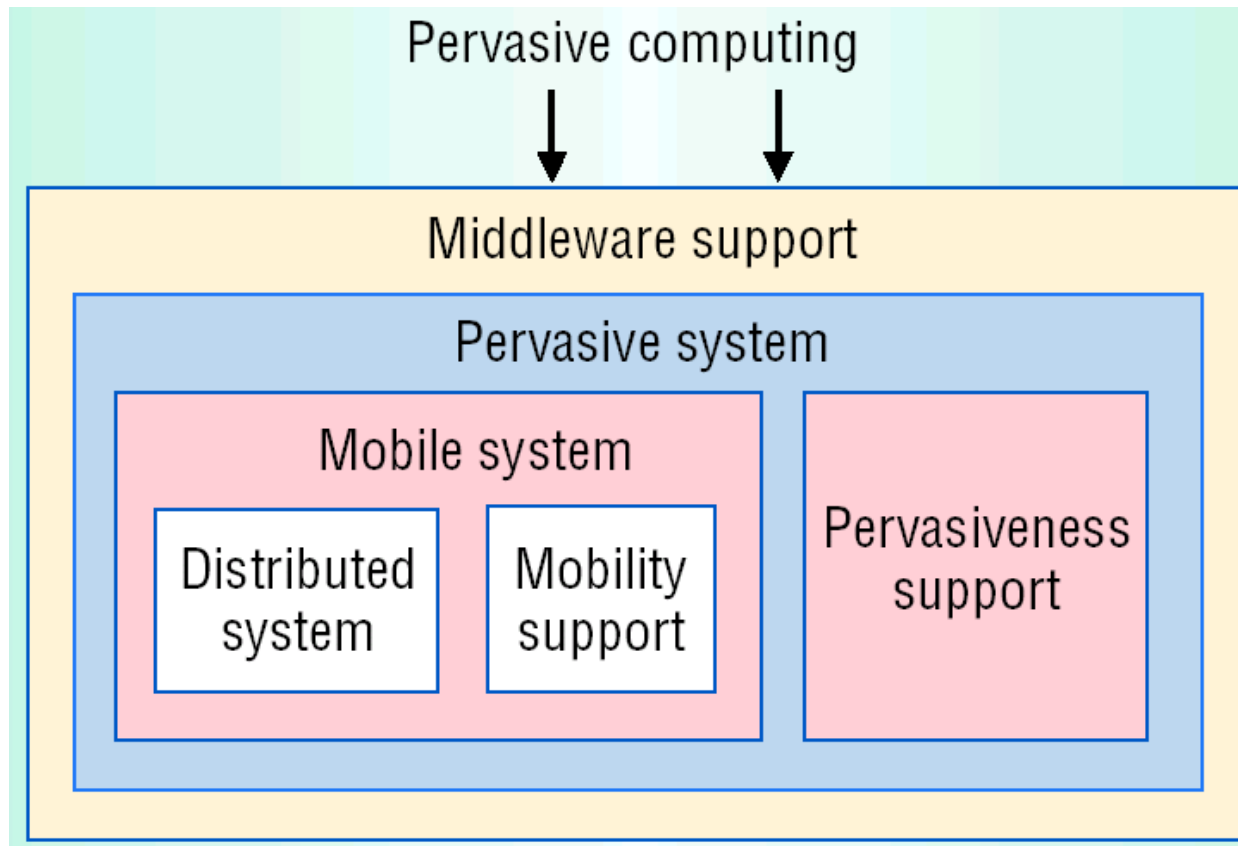
**Scientific American**, Sept.1991, 94-104

IEEE Pervasive Computing, Jan.-Mar. 2002, 19-25

# Εξέλιξη του Διάχυτου Υπολογισμού

- **Desktop Υπολογιστές (PC)**
- **Κατανεμημένοι Υπολογιστές (Distributed)**
  - Τα δίκτυα έδωσαν ώθηση στα PC ώστε να επιτύχουν κατανεμημένο υπολογισμό
- **Κινητοί Υπολογιστές (Mobile)**
  - Ενοποίηση κυψελοειδούς (cellular) τεχνολογίας και Web
  - Διαχωρίζουν τη συσκευή από την κάρτα Subscriber Identity Module (SIM)

# Ο Διάχυτος Υπολογισμός απαιτεί



- Συσκευές
- Δικτύωση
- Middleware
- Εφαρμογές

# Συσκευές για Διάχυτο Υπολογισμό

- Συσκευές
  - Εισόδου
    - Ποντίκι, πληκτρολόγιο, LEDs, κ.τ.λ.
  - Ασύρματες κινητές συσκευές
    - Pagers, PDAs, cell phones, palmtops, κ.τ.λ.
  - Έξυπνες συσκευές
    - Ευφυή συστήματα, δάπεδα με αισθητήρες, κ.τ.λ.

# Φορητές συσκευές

## Pager

- Λήψη μόνο
- Κείμενο
- Μικρή οθόνη



## PDA

- Graphics
- WWW
- Αναγνώριση χαρακτήρων



## Palmtop

- Μικρό πληκτρολόγιο
- “Ελαφρές” εκδόσεις εφαρμογών



## Κινητό τηλέφωνο

- Φωνή
- Δεδομένα
- Επίδειξη απλού κειμένου



## Laptop

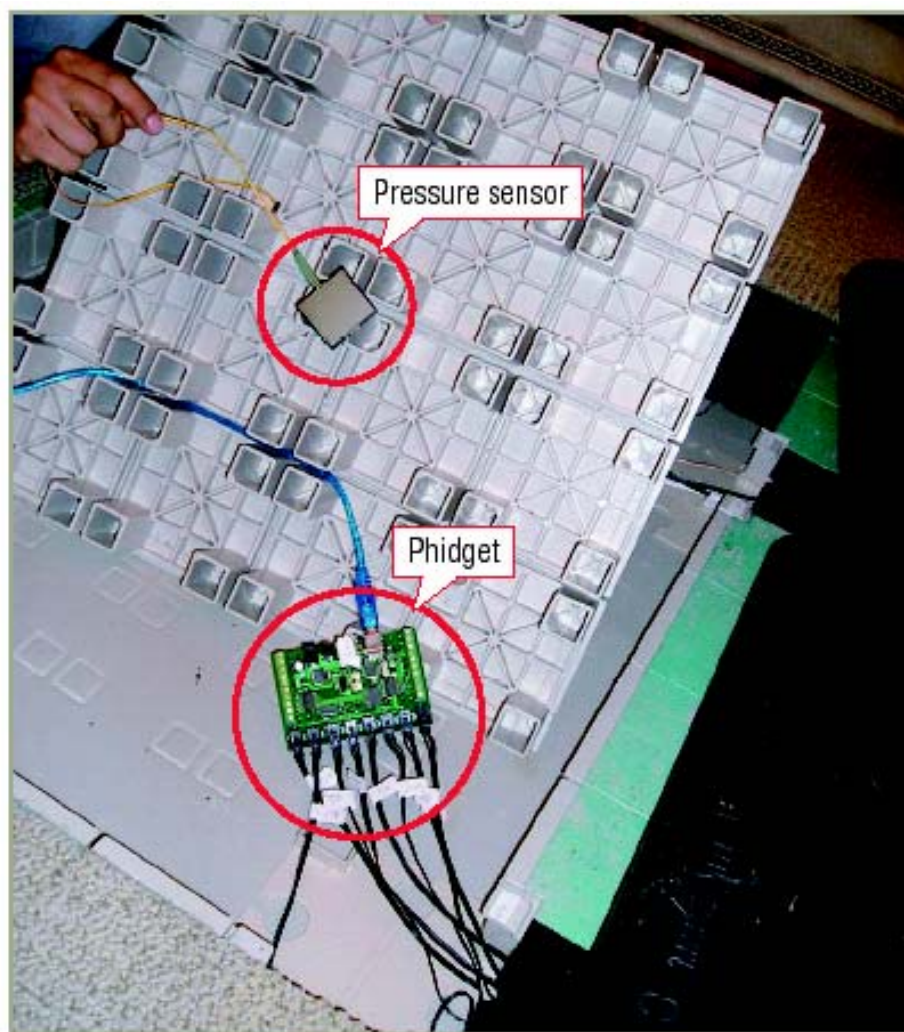
- Πλήρως λειτουργικό
- Πλήρεις εκδόσεις των τυπικών εφαρμογών



Επίδοση και μέγεθος

# “Συσκευές” για Διάχυτο Υπολογισμό

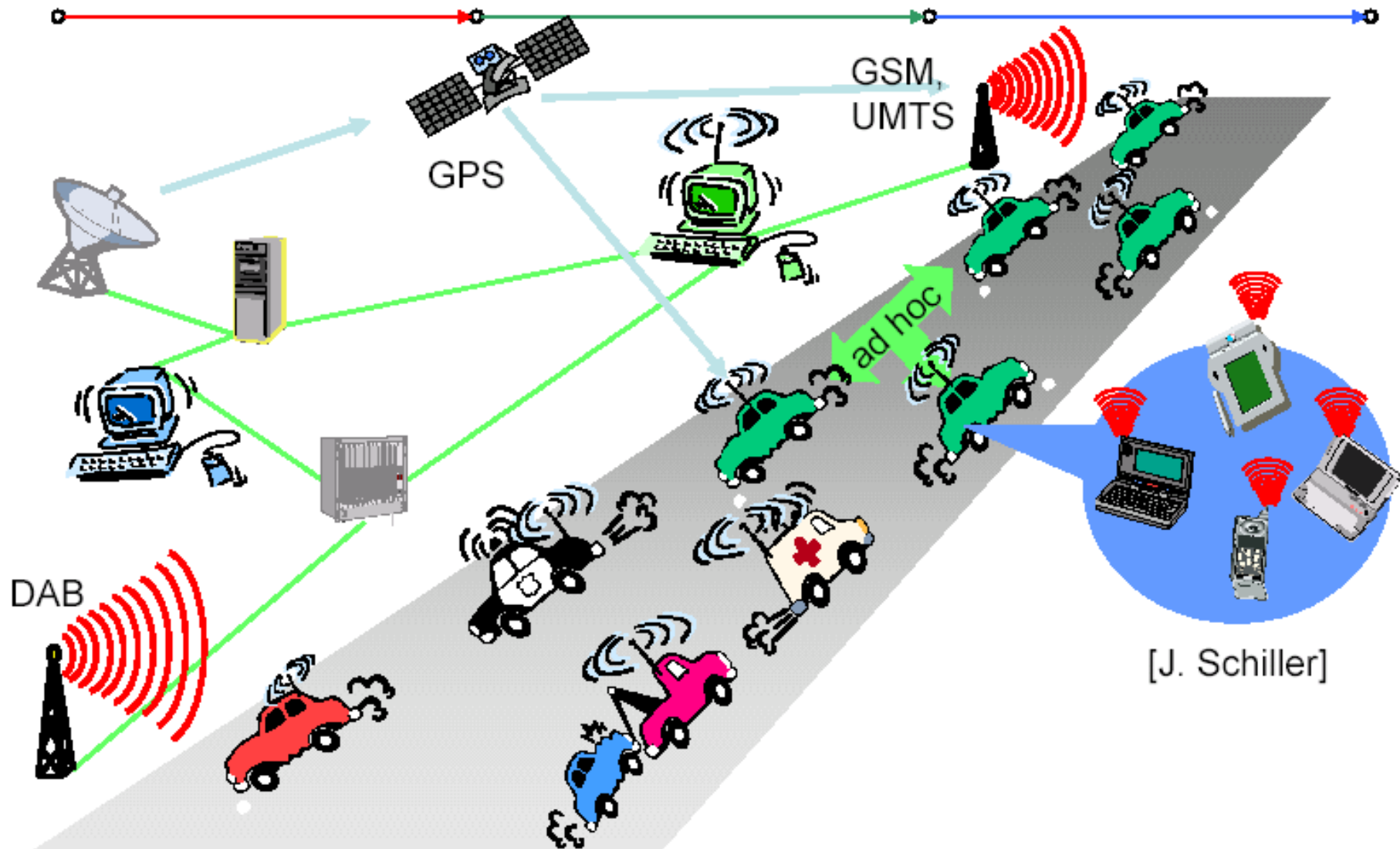
9



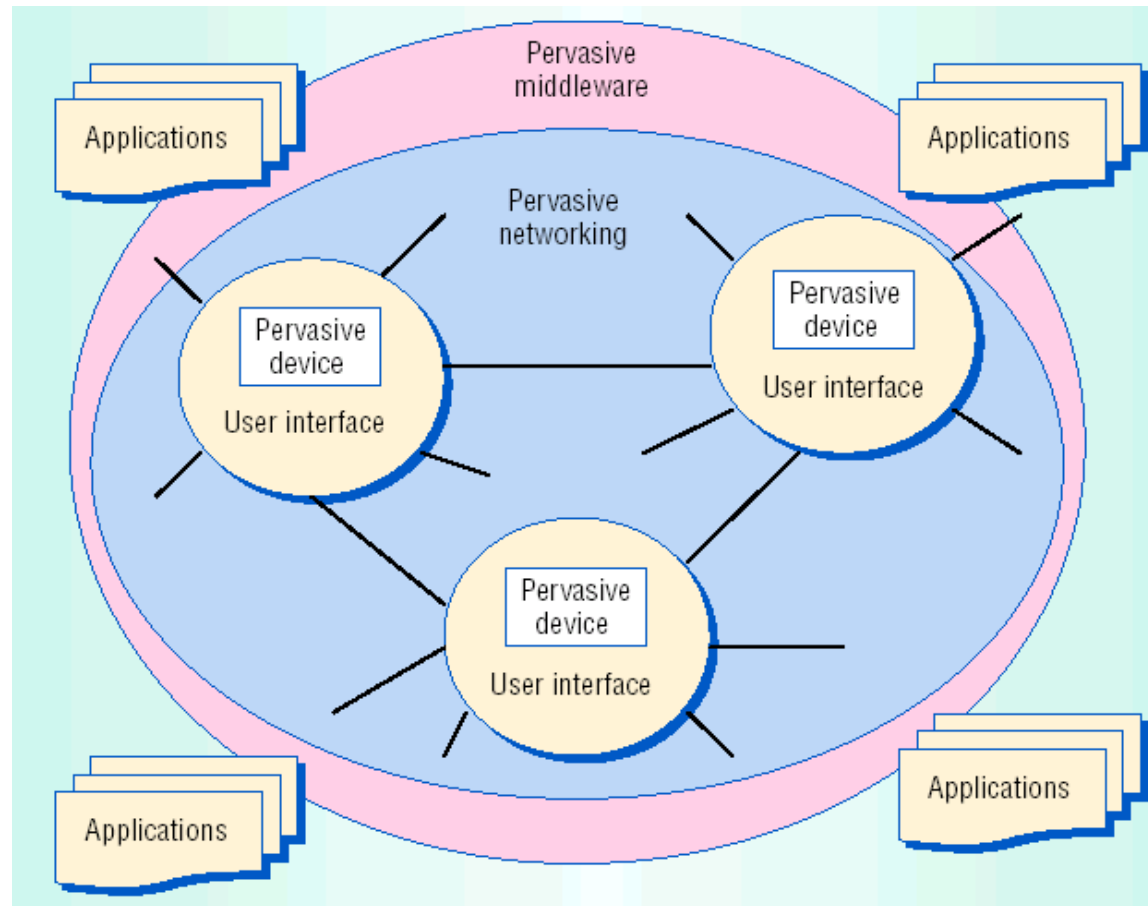
# Δικτύωση για Διάχυτο Υπολογισμό

- *Cellular* - GSM (Europe+), TDMA & CDMA (US)
  - FM: 1.2-9.6 Kbps; Digital: 9.6-14.4 Kbps (ISDN-like υπηρεσίες)
- *Public Packet Radio* - Ιδιωτικό
  - 19.2 Kbps (raw), 9.6 Kbps (effective)
- *Private και Shared Mobile Radio*
- *Wireless LAN* - wireless LAN bridge (IEEE 802.11)
  - Radio ή Infrared frequencies: 1.2 Kbps-15 Mbps
- *Paging Networks* – τυπικά one-way communication
  - low receiving power consumption
- *Satellites* – wide-area coverage (GEOS, MEOS, LEOS)
  - LEOS: 2.4 Kbps (uplink), 4.8Kbps (downlink)

# Δικτύωση για Διάχυτο Υπολογισμό

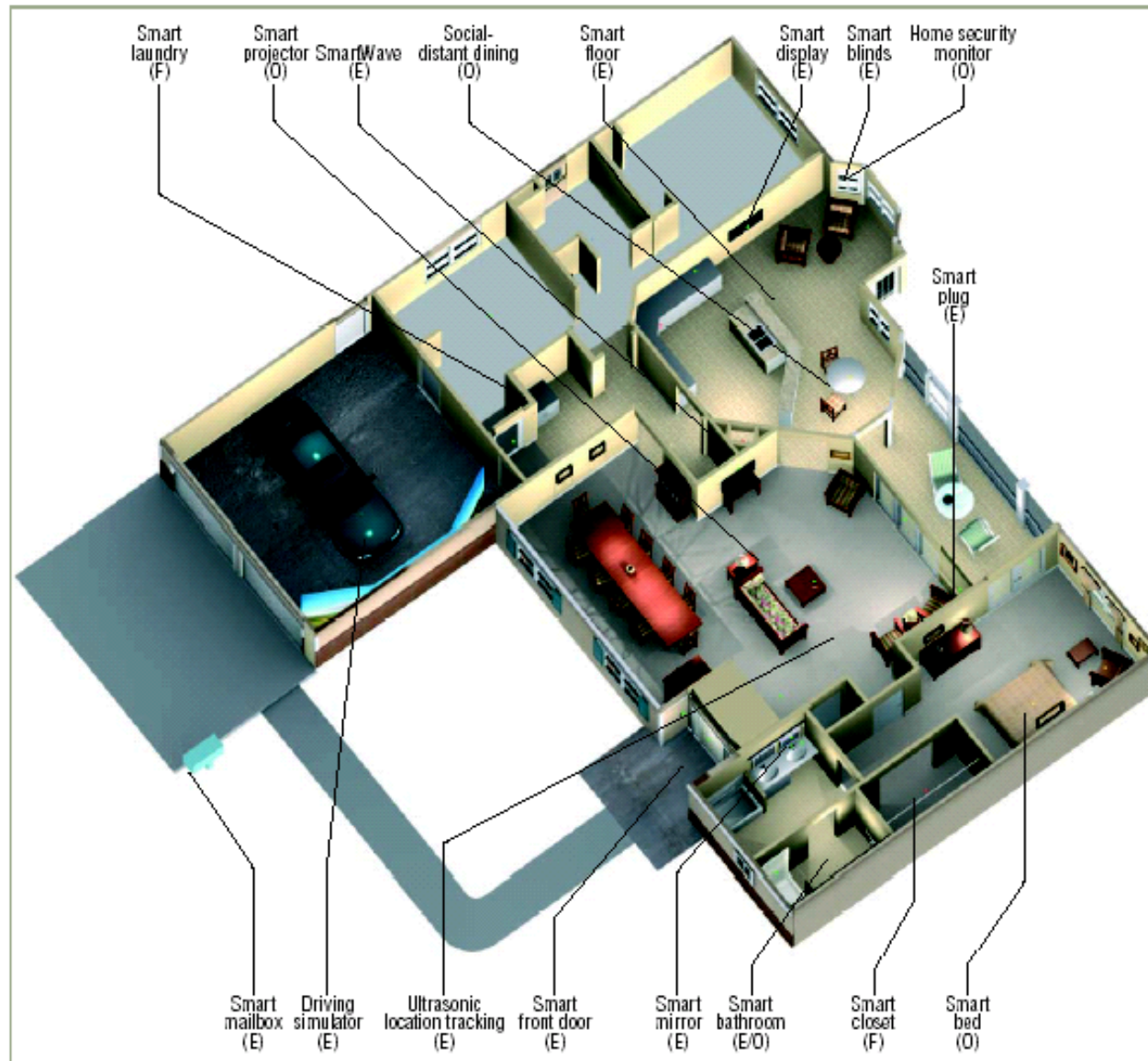


# Middleware για Διάχυτ. Υπολογισμό

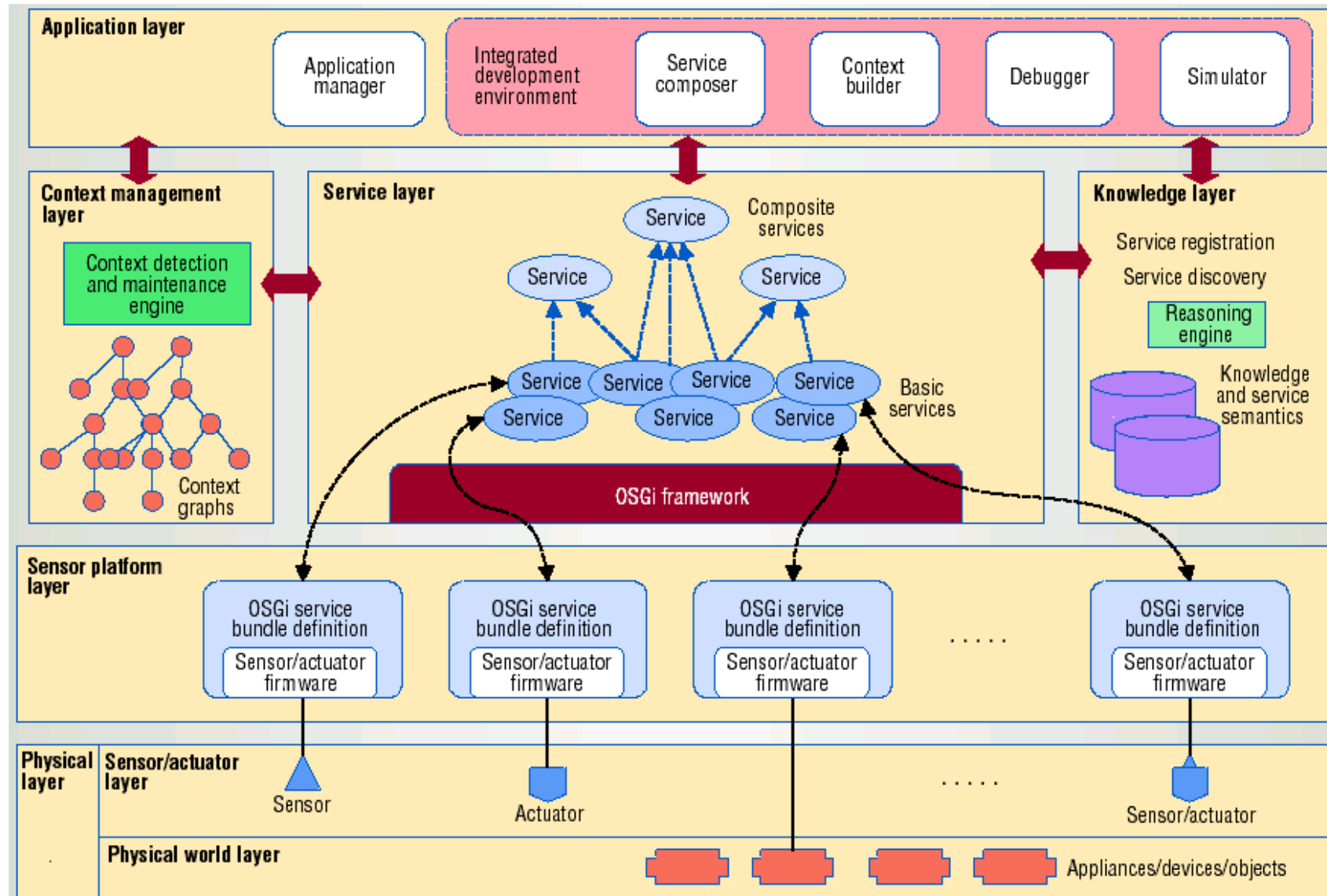


Ο Διάχυτος Υπολογισμός είναι περισσότερο **περιβαλλοντικο-κεντρικός** από ότι ο κινητός υπολογισμός ή ο υπολογισμός στο Web. **Οι εφαρμογές κατευθύνουν το middleware και το είδος της δικτύωσης.**

# Εφαρμογές, εφαρμογές ...



# Αρχιτεκτονική middleware



# Αρχιτεκτ. Middleware στο Έξυπνο Σπίτι

- Φυσικό επίπεδο
- Επίπεδο πλατφόρμας αισθητήρων
- Επίπεδο υπηρεσιών
- Επίπεδο γνώσης
- Επίπεδο διαχείρισης context
- Επίπεδο εφαρμογής

# Εφαρμογές, εφαρμογές ...

- Συσκευές για επικοινωνίες
- Προειδοποίησης
- Μετάδοσης δεδομένων



# Προκλήσεις για Διάχυτο Υπολογισμό

- Κλιμάκωση (Scalability)
- Ετερογένεια (Heterogeneity)
- Ολοκλήρωση (Integration)
- Αδιαφάνεια (Invisibility)
- Γνώση του περιβάλλοντος (Context awareness)
- Διαχείριση του περιβάλλοντος (Context management)

# Περιεχόμενα

- Διάχυτος Υπολογισμός
- **Περίληψη των Ασύρματων Επικοινωνιών**
- Κινητός Υπολογισμός
- Αρχιτεκτονικές Λογισμικού για Κινητό Υπολογισμό

# Η ιστορία των Radio Επικοινωνιών

- **1880: Hertz – Initial demonstration of practical radio communication**
- **1897: Marconi – Radio transmission to a tugboat over an 18 mi path**
- **1921: Detroit Police Department: -- Police car radio dispatch (2 MHz frequency band)**
- **1933: FCC (Federal Communications Commission) – Authorized four channels in the 30 to 40 MHz range**
- **1938: FCC – Ruled for regular service**
- **1946: Bell Telephone Laboratories – 152 MHz (Simplex)**
- **1956: FCC – 450 MHz (Simplex)**
- **1959: Bell Telephone Laboratories – Suggested 32 MHz band for high capacity mobile radio communication**
- **1964: FCC – 152 MHz (Full Duplex)**
- **1964: Bell Telephone Laboratories – Active research at 800 MHz**
- **1969: FCC – 450 MHz (Full Duplex)**
- **1974: FCC – 40 MHz bandwidth allocation in the 800 to 900 MHz range**
- **1981: FCC – Release of cellular land mobile phone service in the 40 MHz bandwidth in the 800 to 900 MHz range for commercial operation**

# Η ιστορία των Radio Επικοινωνιών

- **1981: AT&T and RCC (Radio Common Carrier) reach an agreement to split 40 MHz spectrum into two 20 MHz bands. Band A belongs to nonwireline operators (RCC), and Band B belongs to wireline operators (telephone companies). Each market has two operators.**
- **1982: AT&T is divested, and seven RBOCs (Regional Bell Operating Companies) are formed to manage the cellular operations**
- **1982: MFJ (Modified Final Judgment) is issued by the government DOJ. All the operators were prohibited to (1) operate long-distance business, (2) provide information services, and (3) do manufacturing business**
- **1983: Ameritech system in operation in Chicago**
- **1984: Most RBOC markets in operation**
- **1986: FCC allocates 5 MHz in extended band**
- **1987: FCC makes lottery on the small MSA and all RSA licenses**
- **1988: TDMA (Time Division Multiple Access) voted as a digital cellular standard in North America**
- **1992: GSM (Groupe Speciale Mobile) operable in Germany D2 system**

# Η ιστορία των Radio Επικοινωνιών

- **1993: CDMA (Code Division Multiple Access) voted as another digital cellular standard in North America**
- **1994: American TDMA operable in Seattle, Washington**
- **1994: PDC (Personal Digital Cellular) operable in Tokyo, Japan**
- **1994: Two of six broadband PCS (Personal Communication Service) license bands in auction**
- **1995: CDMA operable in Hong Kong**
- **1996: US Congress passes Telecommunication Reform Act Bill**
- **1996: The auction money for six broadband PCS licensed bands (120 MHz) almost reaches 20 billion US dollars**
- **1997: Broadband CDMA considered as one of the third generation mobile communication technologies for UMTS (Universal Mobile Telecommunication Systems) during the UMTS workshop conference held in Korea**
- **1999: ITU (International Telecommunication Union) decides the next generation mobile communication systems (e.g., W-CDMA, cdma2000, etc)**

# 1G Cellular Συστήματα & Υπηρεσίες

|       |                                                                                                             |
|-------|-------------------------------------------------------------------------------------------------------------|
| 1970s | Developments of radio and computer technologies for 800/900 MHz mobile communications                       |
| 1976  | WARC (World Administrative Radio Conference) allocates spectrum for cellular radio                          |
| 1979  | NTT (Nippon Telephone & Telegraph) introduces the first cellular system in Japan                            |
| 1981  | NMT (Nordic Mobile Telephone) 900 system introduced by Ericsson Radio System AB and deployed in Scandinavia |
| 1984  | AMPS (Advanced Mobile Phone Service) introduced by AT&T in North America                                    |

# 2G Cellular Συστήματα & Υπηρεσίες

|             |                                                                                                                                           |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1982</b> | <b>CEPT (Conference Europeenne des Post et Telecommunications) established GSM to define future Pan-European Cellular Radio Standards</b> |
| <b>1990</b> | <b>Interim Standard IS-54 (USDC) adopted by TIA (Telecommunications Industry Association)</b>                                             |
| <b>1990</b> | <b>Interim Standard IS-19B (NAMPS) adopted by TIA</b>                                                                                     |
| <b>1991</b> | <b>Japanese PDC (Personal Digital Cellular) system standardized by the MPT (Ministry of Posts and Telecommunications)</b>                 |
| <b>1992</b> | <b>Phase I GSM system is operational</b>                                                                                                  |
| <b>1993</b> | <b>Interim Standard IS-95 (CDMA) adopted by TIA</b>                                                                                       |
| <b>1994</b> | <b>Interim Standard IS-136 adopted by TIA</b>                                                                                             |
| <b>1995</b> | <b>PCS Licenses issued in North America</b>                                                                                               |
| <b>1996</b> | <b>Phase II GSM operational</b>                                                                                                           |
| <b>1997</b> | <b>North American PCS deploys GSM, IS-54, IS-95</b>                                                                                       |
| <b>1999</b> | <b>IS-54: North America<br/>IS-95: North America, Hong Kong, Israel, Japan, China, etc<br/>GSM: 110 countries</b>                         |

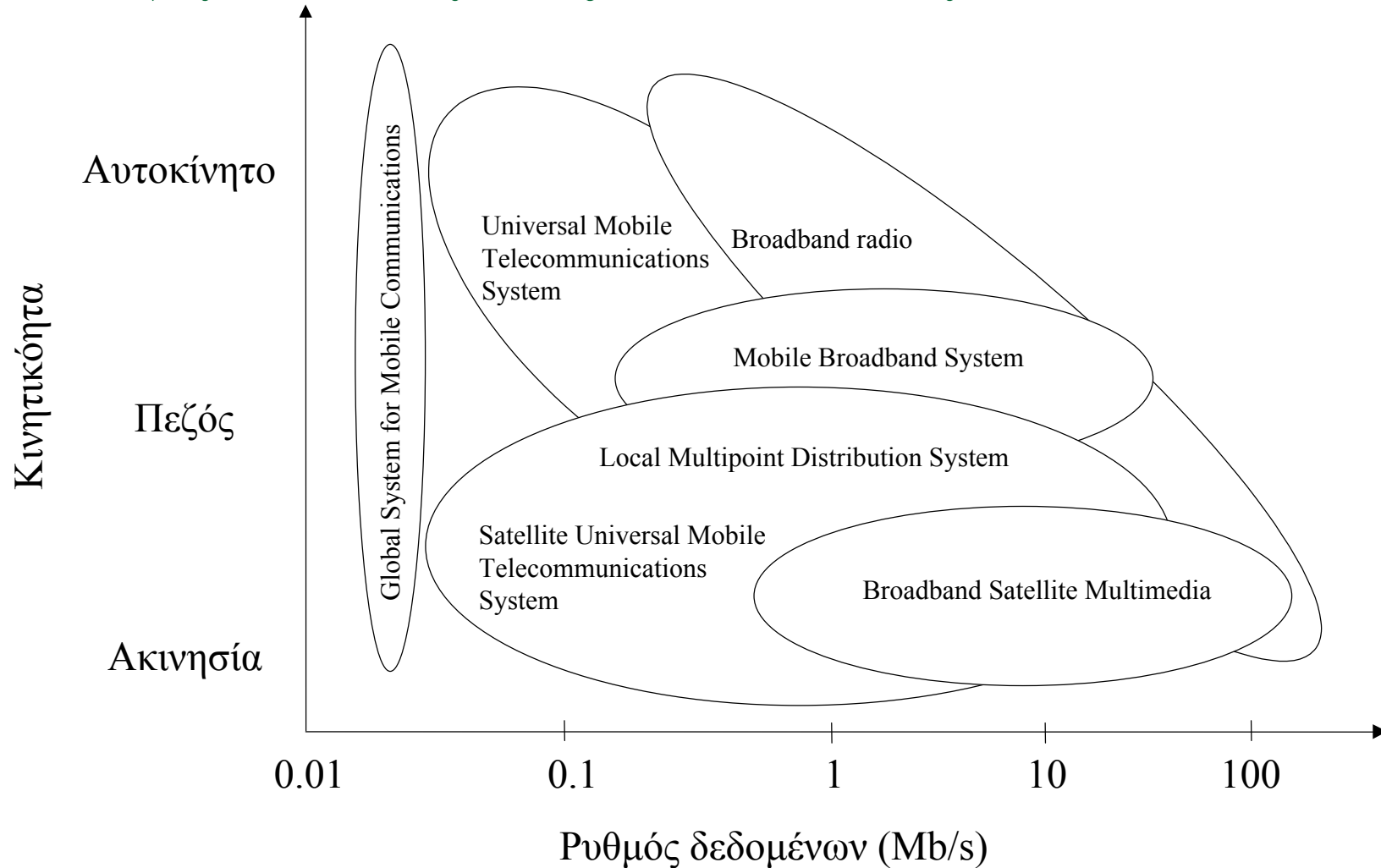
# 3G Cellular Συστήματα & Υπηρεσίες

- **IMT-2000 (International Mobile Telecommunications-2000):**  
Εκπλήρωση της ιδέας του anywhere, anytime communication.
- **Χαρακτηριστικά-κλειδιά του IMT-2000 περιλαμβάνουν:**
  - Υψηλό βαθμό κοινής σχεδίασης ανά τον κόσμο;
  - Συμβατότητα των υπηρεσιών μέσα στο IMT-2000 και με εκείνες των σταθερών δικτύων;
  - Υψηλή ποιότητα;
  - Μικρά τερματικά για χρήση ανά τον κόσμο;
  - Ικανότητα roaming σε όλο τον κόσμο;
  - Ικανότητα για πολυμεσικές εφαρμογές, και μεγάλο εύρος υπηρεσιών και τερματικών.

# 3G Cellular Συστήματα & Υπηρεσίες

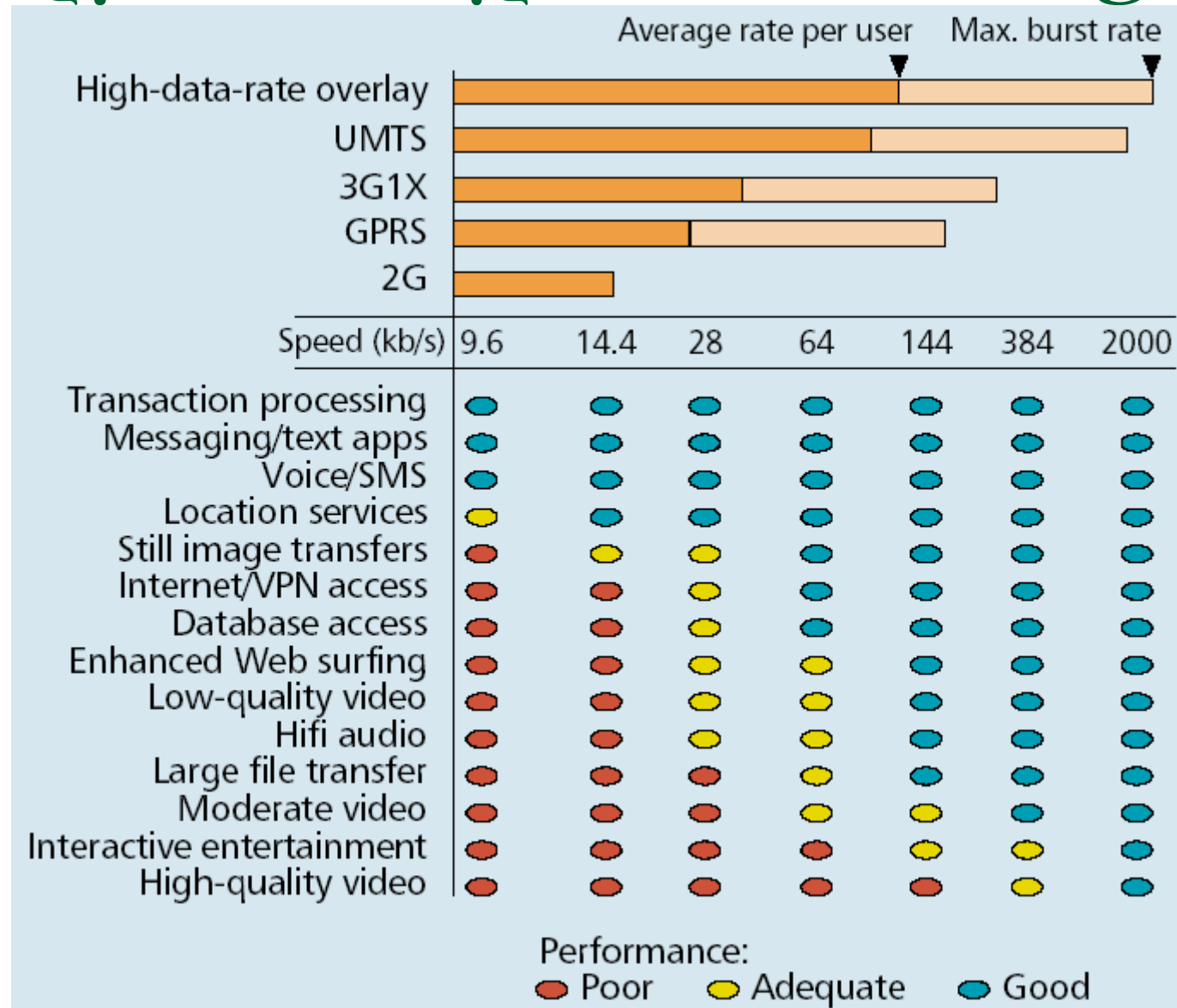
- **Σημαντική συνιστώσα του IMT-2000 είναι η ικανότητα να παρέχει υψηλό ρυθμό δεδομένων:**
  - 2 Mbps για σταθερά περιβάλλοντα;
  - 384 Kbps για indoor/outdoor και στους πεζούς;
  - 144 kbps για αυτοκίνητα.
- **Εργασία Standardization:**
  - Προδιαγραφές συντάχθηκαν το 1999
  - Σε ανάπτυξη
- **Αναμενόμενη υπηρεσία:**
  - Ξεκίνησε τον Οκτώβριο 2001 στην Ιαπωνία (W-CDMA)

# Χωρητικότητα μετάδοσης

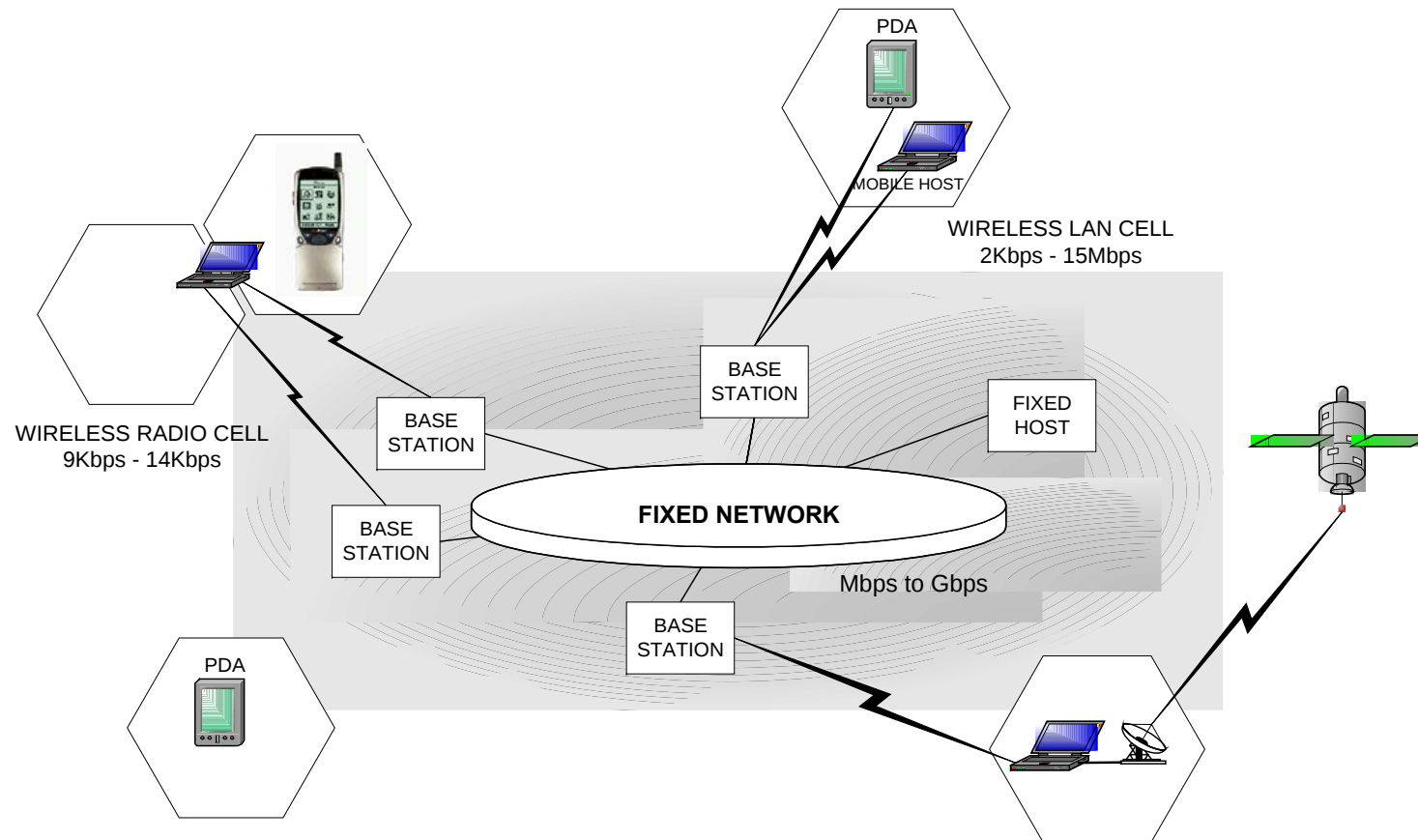


Χωρητικότητα μετάδοσης ως συνάρτηση της κινητικότητας σε μερικά radio συστήματα

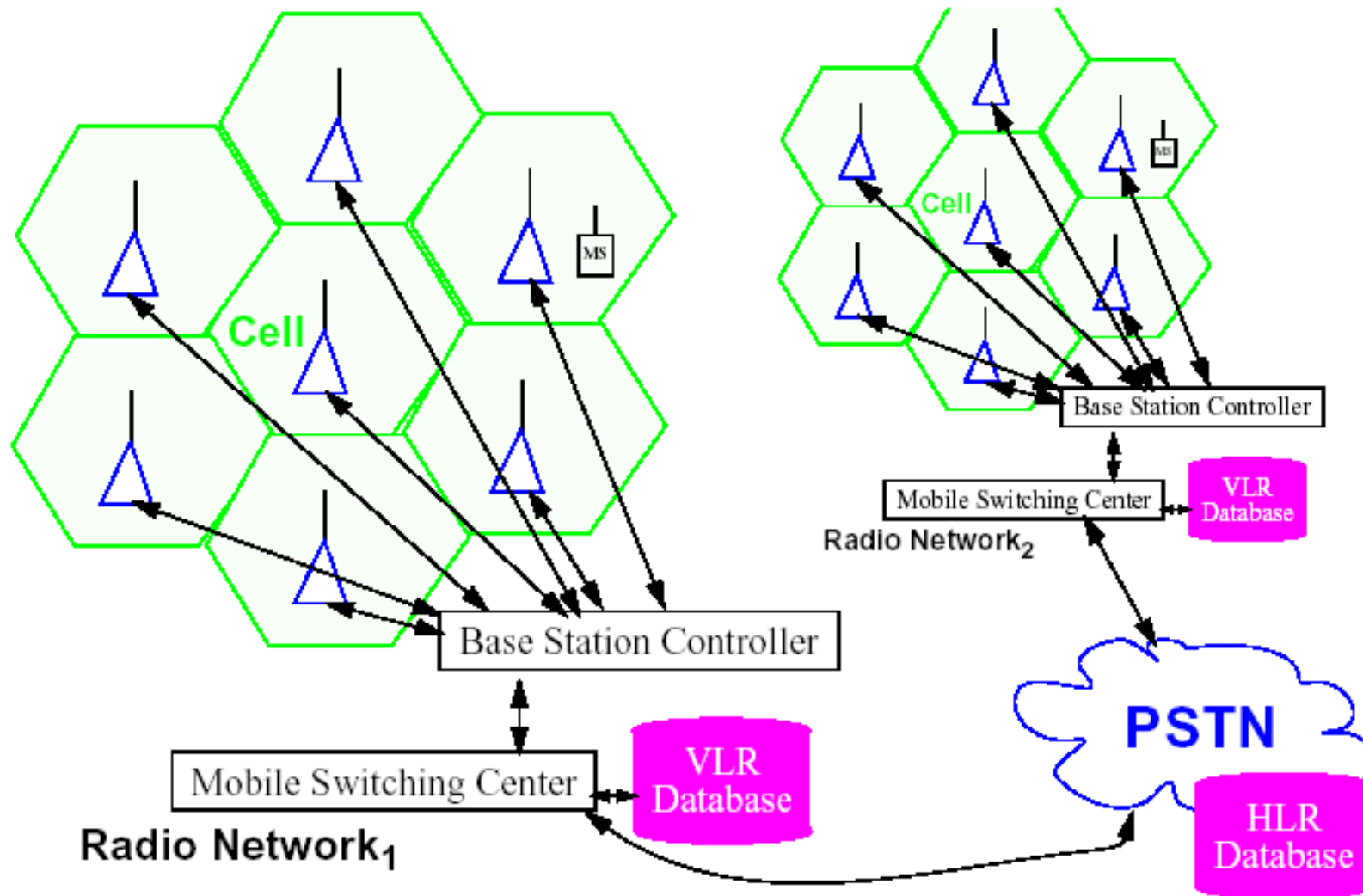
# Ασύρματες υπηρεσίες & throughput



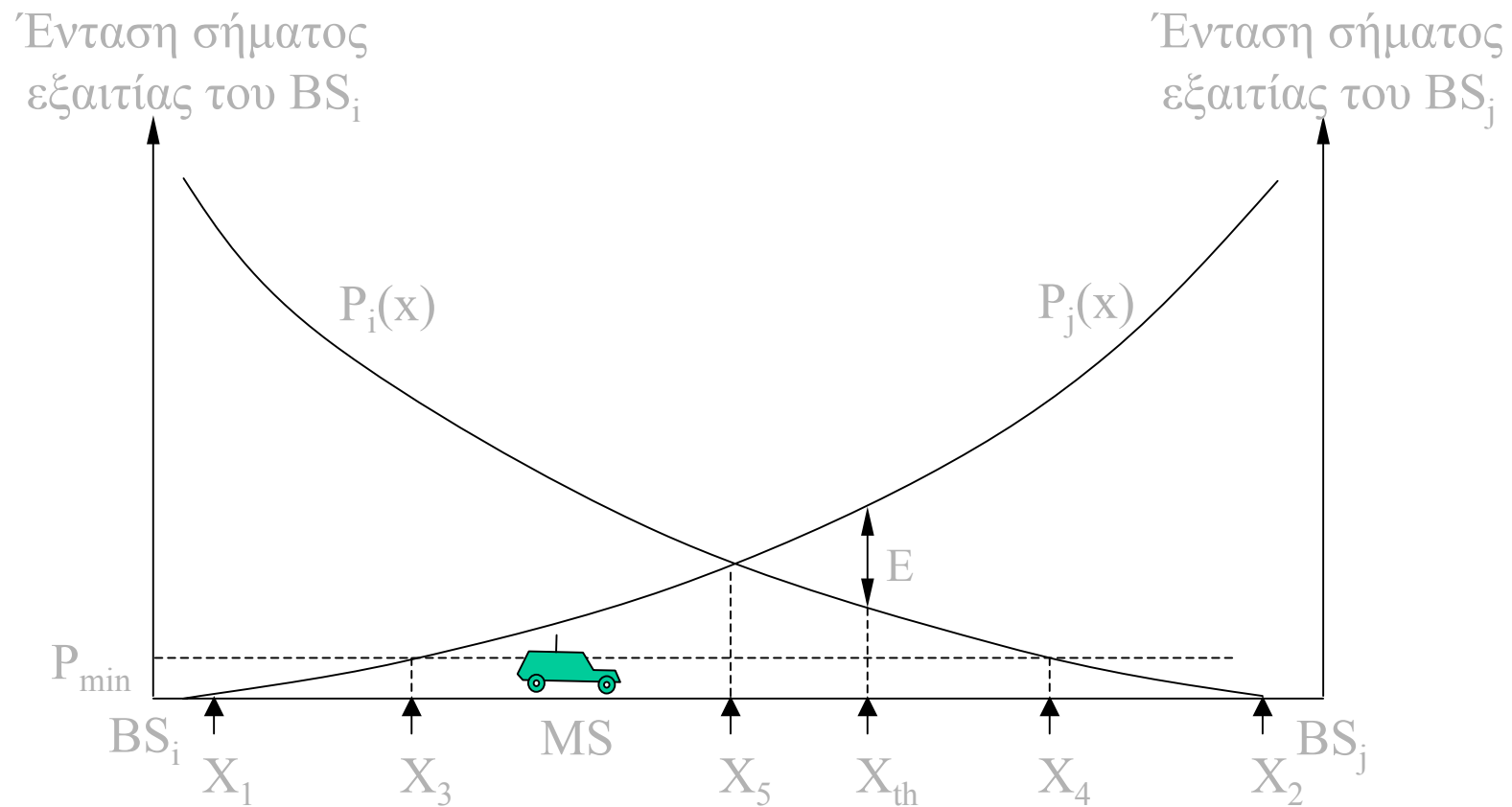
# Αρχιτεκτονική κινητού δικτύου



# Αρχιτεκτονική PCS



# Διαδικασία handoff (handover)



# Περιεχόμενα

- Διάχυτος Υπολογισμός
- Περίληψη των Ασύρματων Επικοινωνιών
- **Κινητός Υπολογισμός**
- Αρχιτεκτονικές Λογισμικού για Κινητό Υπολογισμό

# Κινητός Υπολογισμός: Περιεχόμενα

- **Παράδειγμα-κίνητρο**
- Ζητήματα: Κινητικότητα, Ασύρματη επικοινωνία, Φορητότητα
- Αρχιτεκτονικές λογισμικού
- Υποστήριξη σε επίπεδο συστήματος

# Κάνουμε πάρτι την Παρασκευή



- Ενημέρωση του ημερολογίου του Smart Phone's με τα ονόματα των προσκεκλημένων.
- Σημείωση για παραγγελία φαγητού από Pizza Hut.
- Ενημέρωση της λίστας αγορών με τις προτιμήσεις σε ποτά των προσκεκλημένων.

# Κάνουμε πάρτι την Παρασκευή

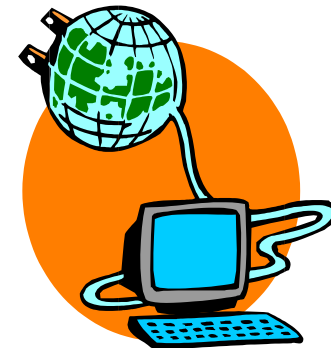
- Το AutoPC ανιχνεύει ένα κοντινό supermarket που διαφημίζει τις πωλήσεις του.



- Προσπελάζει τη λίστα αγορών μας και το ημερολόγιό μας στο Smart Phone.
- Μας ενημερώνει ότι η μπύρα και ... είναι σε έκπτωση. Μας υπενθυμίζει ότι το επόμενο rendez-vous είναι σε 1 ώρα.
- Υπάρχει χρόνος γιατί η κίνηση είναι χαμηλή, όπως μας ενημέρωσε η Traffic Report υπηρεσία.

# Κάνουμε πάρτι την Παρασκευή

- Το Smart Phone μας υπενθυμίζει ότι πρέπει να παραγγείλουμε φαγητό το βράδυ.
- Κάνει download το μενού από την Pizza Hut στο PC και επιδεικνύει τις προτιμήσεις των προσκεκλημένων.
- Αποστέλλει τη λίστα αγορών σε αντίστοιχο PC του supermarket.
- Έτσι όλα θα είναι έτοιμα όταν πρέπει.



# “Κινητές” εφαρμογές

- Θα δημιουργήσουν μια εντελώς νέα κλάση εφαρμογών
  - Νέες μαζικές αγορές σε συνδυασμό με το Web
  - Mobile Information Appliances – συνδυασμός personal computing και consumer electronics
- Εφαρμογές:
  - Vertical: vehicle dispatching, tracking, point of sale
  - Horizontal: mail enabled applications, filtered information provision, collaborative computing...

# Κινητός Υπολογισμός: Περιεχόμενα

- Παράδειγμα-κίνητρο
- **Ζητήματα: Κινητικότητα, Ασύρματη επικοινωνία, Φορητότητα**
- Αρχιτεκτονικές λογισμικού
- Υποστήριξη σε επίπεδο συστήματος

# Ασύρματος & Κινητός Υπολογισμός

- Στόχος: Πρόσβαση στην πληροφορία Anywhere, Anytime, και in Anyway.
- Συνώνυμα: Mobile, Nomadic, Wireless, Invisible, Ubiquitous Computing, Pervasive Computing (?).
- Διάκριση:
  - Fixed wired network: Traditional distributed computing.
  - Fixed wireless network: Wireless computing.
  - Wireless network: Mobile Computing.

## ➤ Ζητήματα-κλειδιά:

Ασύρματα επικοινωνία, Κινητικότητα, Φορητότητα.

# Χαρακτηριστικά ασύρματου δικτύου

- Μεταβαλλόμενη σύνδεση
  - Χαμηλό εύρος ζώνης και χαμηλή αξιοπιστία
- Συχνές αποσυνδέσεις
  - Προβλεπόμενες ή ξαφνικές
- Ασύμμετρη επικοινωνία
  - Κανάλι εκπομπής
- Χρηματικό κόστος
  - Χρεώσεις ανά σύνδεση ή ανά πακέτο/μήνυμα

➤ Η σύνδεση είναι ασθενής, διακοπτόμενη και ακριβή

# Φορητές Συσκευές

- PDAs, Personal Communicators
  - Ελαφρές, μικρές και ανθεκτικές στη χρήση, ώστε να μπορούμε να τις μεταφέρουμε εύκολα
  - Κουτά τερματικά [InfoPad, ParcTab projects], palmtops, wristwatch PC/Phone, walkstations
- Κάνουν χρήση AA+ /Ni-Cd/Li-Ion μπαταριών
- Ίσως δεν έχουν σκληρό δίσκο
- I/O συσκευές: Χωρίς ποντίκι, αλλά με “Στυλό”
- Ασύρματη σύνδεση σε Πληροφοριακά Δίκτυα
  - Είτε υπέρυθρα ή cellular τηλέφωνα
- Ειδικό HW (για συμπίεση/κρυπτογράφηση)

# Χαρακτηριστικά φορητότητας

- Περιορισμοί εξαιτίας της Ενέργειας της Μπαταρίας
  - transmit/receive, disk spinning, display, CPUs, memory consume power
- Η διάρκεια ζωής της μπαταρίας θα βελτιωθεί (σχετικά) λίγο
  - Απαιτείται energy-efficient υλικό (CPUs, memory) και λογισμικό συστήματος
  - “Επιδιωκόμενες” (planned) αποσυνδέσεις - *doze mode*

## ➤ Κατανάλωση Ενέργειας vs. Χρήση Πόρων

# Χαρακτηριστικά φορητότητας

- Περιορισμοί εξαιτίας των πόρων
    - Οι κινητοί υπολογιστές είναι φτωχοί σε πόρους
    - Μικρό μέγεθος προγράμματος – interpret script languages (Mobile Java)
    - Υπολογιστικός και επικοινωνιακός φόρτος δεν πρέπει να είναι μεγάλος
  - Μικρά μεγέθη οθονών
- Ασυμμετρία μεταξύ των στατικών και των κινητών υπολογιστών

# Χαρακτηριστικά κινητικότητας

- Αλλαγές θέσης
  - Διαχείριση θέσης – το κόστος για τον εντοπισμό προστίθεται στην επικοινωνία
- Ετερογένεια στις υπηρεσίες
  - Περιορισμοί εύρους ζώνης και μεταβλητότητα
- Δυναμική replication των δεδομένων
  - Τα δεδομένα και οι υπηρεσίες ακολουθούν τους κινητούς χρήστες
- Επερώτηση των δεδομένων - απαντήσεις εξαρτώμενες από τη θέση (location-based responses)
- Ασφάλεια και authentication
- Η διαμόρφωση (configuration) του συστήματος δεν είναι πλέον στατική

# Τι πρέπει να επανεξετάσουμε?

- Λειτουργικά Συστήματα
- Συστήματα Αρχείων
- Συστήματα Βάσεων Δεδομένων
- Αρχιτεκτονικές Επικοινωνιών
- Υλικό και Αρχιτεκτονική
- Real-Time, multimedia, QoS
- Ασφάλεια
- Απαιτήσεις εφαρμογών και σχεδίαση
- Σχεδιασμός PDA: Interfaces, Γλώσσες

# Επεξερ. Επερωτημάτων/Συναλλαγών

- Η μέριμνα μετακινείται από το CPU time και το network delay στη ενέργεια της μπαταρίας και το κόστος της επικοινωνίας (μαζί με το οικονομικό κόστος)
- Οι αλλαγές παίρνουν τη μορφή long-running transactions
  - Οι κόμβοι συνεχίζουν να λειτουργούν και σε κατάσταση αποσύνδεσης
  - Απαιτούνται νέα μοντέλα συναλλαγών
- Μετακίνηση δεδομένων vs. Μετακίνηση Επερώτησης/Συναλλαγής
- Context (location) based απαντήσεις στα επρωτήματα
- Consistency, autonomy, recovery
  - Προσεγγιστικές απαντήσεις
  - Σταθερή αποθήκευση για logs, δεδομένα -- stabilize στους servers?
- Παροχή ομοιόμορφης πρόσβασης σε ετερογενές περιβάλλον
- Σχεδιασμός νέων human-computer interfaces (pen-based computing)
- *Updated system info*: Location information, user profiles

# Επανεμφανιζόμενα ζητήματα

- Διαχείριση αποσυνδέσεων
  - Στρατηγικές caching
  - Διαχείριση ασυνεπειών (inconsistencies)
- Delayed write-back και prefetch: αξιοποίηση του ανενεργού χρόνου του δικτύου
  - Αύξηση των απαιτήσεων σε μνήμη
- Buffering/batching: επιτρέπει μαζικές (bulk) μεταφορές
- Διαμέριση και replication
  - προκαλείται από τη μετακίνηση (relocation)
- Συμπίεση: αυξάνει την αποτελεσματική BW
  - Αυξάνει τις απαιτήσεις σε ενέργεια μπαταρίας
- Η λήψη απαιτεί λιγότερη ενέργεια από ότι η αποστολή

# Ζητήματα στην παροχή πληροφορίας

- Χρέωση
- Κλιμάκωση: massive # users που προσπελάζουν read-only δεδομένα
- ασφάλεια (δεδομένων και user profiles/location)
  - Κατανάλωση πόρων σε foreign περιβάλλοντα
  - Βλάβη σε σταθερούς hosts?
- Προτεραιότητα ενεργειών κατά την επανασύνδεση

# Κινητικότητα σε Εφαρμογές ΒΔ

- Απαίτηση για *προσαρμογή* σε διαρκώς μεταβαλλόμενο περιβάλλον:
    - Σύνδεση δικτύου
    - Διαθέσιμοι πόροι και υπηρεσίες
  - Με μεταβολή και (επανα)διαπραγμάτευση:
    - Του διαχωρισμού των καθηκόντων μεταξύ των κινητών και των σταθερών στοιχείων
    - Της ποιότητας των δεδομένων στον mobile host
- Παράδειγμα: *Fidelity* (βαθμός στον οποίο ένα αντίγραφο (copy) ενός δεδομένου “ταιριάζει” με το original αντίγραφο στον server)

# Προσαρμοστικότητα

Πού θα πρέπει να τοποθετήσουμε την υποστήριξη για την κινητικότητα και την προσαρμοστικότητα?



## Laissez-Faire

(-) οι εφαρμογές πρέπει να γραφούν ξανά, το οποίο μπορεί να είναι εξαιρετικά δύσκολο

(-) το σύστημα δεν εστιάζει σε ειδικούς μηχανισμούς για να επιλύσει ασύμβατες απαιτήσεις της εφαρμογής ή για να εξοικονομήσει πόρους.

## Application Transparent

(+) οι υπάρχουσες εφαρμογές εξακολουθούν να δουλεύουν χωρίς ανάγκη αλλαγής

(-) πολύ γενική, δεν μπορεί να αξιοποιήσει τη σημασιολογία των εφαρμογών

(-) ίσως δεν είναι εφικτή (π.χ., κατά τη διάρκεια μακράς αποσύνδεσης)

# Προσαρμοζόμενες εφαρμογές

- Απαίτηση:
  - Μέτρηση της QoS και επικοινωνία με την εφαρμογή
    - Μηχανισμός για **παρακολούθηση** του επιπέδου και της ποιότητας της πληροφορίας και **ενημέρωση των εφαρμογών** για τυχόν αλλαγές.
  - Programmer Interface για Application-Aware Adaptation
    - Οι εφαρμογές πρέπει να είναι **agile**: ικανές να λαμβάνουν γεγονότα (events) με ασύγχρονο τρόπο και να αντιδρούν ανάλογα
  - **Κεντρικό σημείο** για διαχείριση πόρων και εξουσιοδότηση όποιας αίτησης από μέρους κάποιας εφαρμογής.

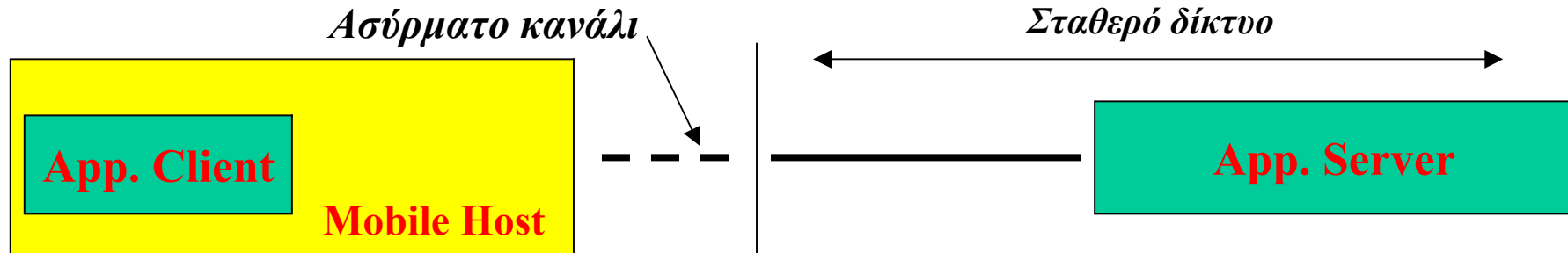
# Κινητός Υπολογισμός: Περιεχόμενα

- Παράδειγμα-κίνητρο
- Ζητήματα: Κινητικότητα, Ασύρματη επικοινωνία, Φορητότητα
- **Αρχιτεκτονικές λογισμικού**
- Υποστήριξη σε επίπεδο συστήματος

# Περιορισμοί κινητού υπολογισμού

- Οι συσκευές: φτωχές σε πόρους και μη “αξιόπιστες”
- Η σύνδεση του δικτύου: χαμηλού εύρους ζώνης και διακόπτεται συχνά
- Η κινητικότητα επιτείνει τα προβλήματα αυτά
- Απαιτείται εξέταση της λειτουργικότητας των mobile hosts
  - “Κουτά” τερματικά ?
  - Οι παραπάνω περιορισμοί υπονοούν ότι ο mobile host πρέπει να έχει (κάποια) λειτουργικότητα για να λειτουργεί αυτόνομα όταν αποσυνδέεται
- **Ζητήματα**
  - Κατάλληλο μοντέλο για συστήματα κινητού υπολογισμού
  - Η κινητικότητα θα είναι διαφανής στις εφαρμογές?

# Το μοντέλο Client-Server (& επεκτάσεις)

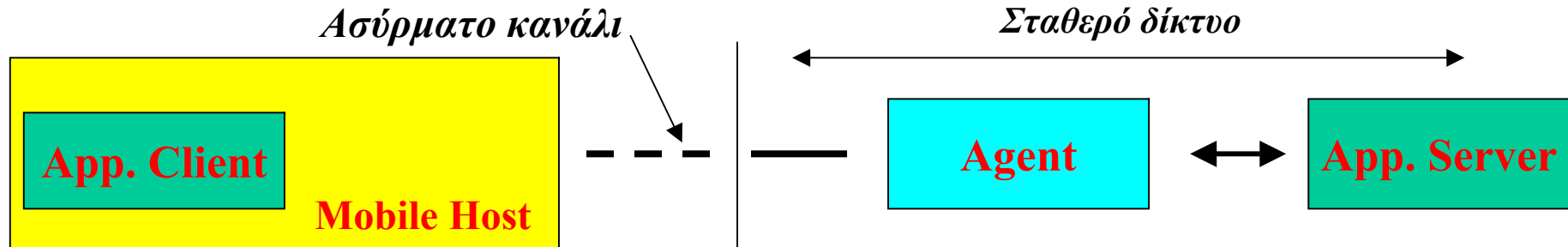


- Λειτουργικότητα και δεδομένα είναι κατανεμημένα σε διάφορους servers του σταθερού δικτύου
- Δεν υπάρχει σαφής διάκριση μεταξύ mobile client και server στο σταθερό δίκτυο (σκεφτείτε τι συμβαίνει στην αποσύνδεση)
- Τύπος του μηχανισμού επικοινωνίας
  - Άμεση ανταλλαγή μηνυμάτων μεταξύ client και server
    - Προβληματική μέθοδος: αργό και μη αξιόπιστο κανάλι
  - (Queued) Remote Procedure Calls
    - Η “σύγχρονη” προσέγγιση προκαλεί blocking εξαιτίας των αποσυνδέσεων και δεν ελαττώνει τη φόρτο επικοινωνίας
    - Η “ασύγχρονη” προσέγγιση αποθηκεύει προσωρινά τις RPC (σε log) και όταν συνδεθεί τις εκτελεί (βελτιστοποιήσεις στο log)

# Το μοντέλο Client-Agent-Server

- Γενικά το μοντέλο **C-A-S** μειώνει αισθητά την επίδραση του μικρού εύρους ζώνης και την μικρής αξιοπιστίας του ασύρματου καναλιού
- Agent = Αντιπρόσωπος (ή proxy) του client
  - Κάθε επικοινωνία προς/από τον client περνάει μέσα από τον agent
  - Δίνουμε την λειτουργικότητα του client στον agent
- Δυο (υπο)μοντέλα
  - Client-ServerAgent-Server
  - Client-ClientAgent-Server

# Το μοντέλο Client-SAgent-Server

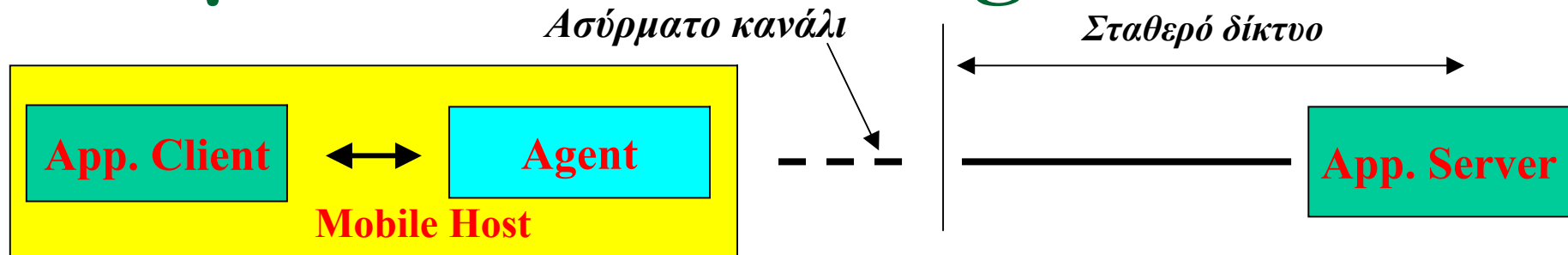


- **C-SA-S**: Το μοντέλο Client/Server-side Agent/Server
- Διαχωρίζει την αλληλεπίδραση μεταξύ mobile client και server: *client-agent* και *agent-server*
  - Διαφορετικά πρωτόκολλα για κάθε μέρος της αλληλεπίδρασης
  - Κάθε μέρος μπορεί να εκτελείται ανεξάρτητα από το άλλο
    - Χειρίζεται τα δεδομένα πριν την αποστολή τους στον client
- Το μοντέλο αυτό είναι κατάλληλο για “light-weight” clients με περιορισμένη υπολογιστική ισχύ και πόρους

# Υποχρεώσεις του Agent

- Δίνουμε την λειτουργικότητα του client στον agent
  - Οι αιτήσεις του client στέλνονται στον agent και τα αποτελέσματα στέλνονται στον client όταν γίνει η σύνδεση
- Messaging και queuing
- Πιο ενεργό ρόλο
  - Εκκίνηση/διακοπή ειδικών λειτουργιών στον mobile client
  - Εκτελεί συμπίεση (κατάλληλη για κάθε εφαρμογή)
  - Ομαδοποιεί αιτήσεις
  - Αλλάζει την σειρά της μετάδοσης (ανάλογα με τη σημαντικότητα)

# Το μοντέλο Client-CAgent-Server



- **C-CA-S**: Το μοντέλο Client/Client-side Agent/Server
  - caching
  - prefetching στο υπόβαθρο

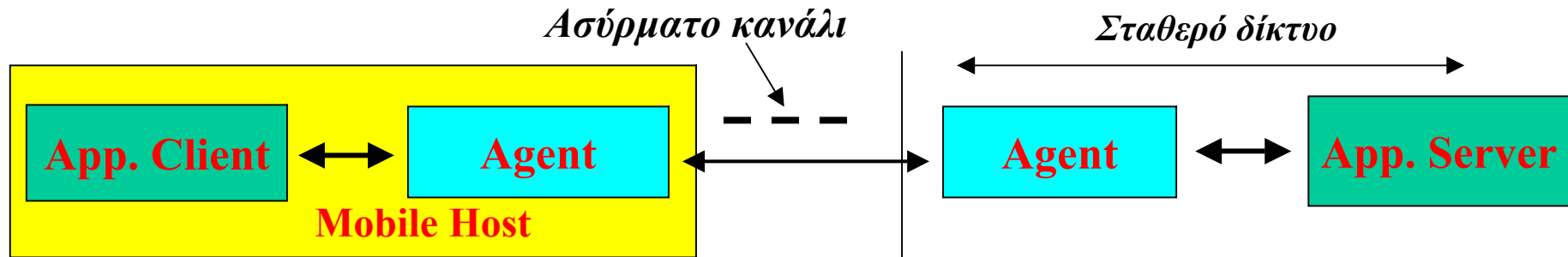
# Ο ρόλος του Agent

- Συνεχής παρουσία του mobile client στο σταθερό δίκτυο
- Ένας agent μπορεί να αντιπροσωπεύει περισσότερους του ενός mobile clients
- Εξειδικευμένος για κάθε Εφαρμογή (Υπηρεσία)
  - Παρέχει ένα mobile-aware στρώμα σε ειδικές υπηρεσίες ή εφαρμογές (π.χ., Web-browsing ή πρόσβαση σε βάσεις δεδομένων)
- Κάθε εφαρμογή μπορεί να έχει πολλούς agents
  - Ελαστικότητα, αλλά πιο περίπλοκο στρώμα διαχείρισης
- Φίλτρα
  - Παρέχουν agents που λειτουργούν πάνω σε πρωτόκολλα
  - Π.χ., ένας MPEG-agent ή ένας TCP-agent

# Μειονεκτήματα μοντέλου C/A/S

- Αποτυγχάνει να διατηρήσει τον υπολογισμό όταν γίνεται αποσύνδεση
- Δεν μπορεί να επιτύχει βελτιστοποιήσεις στην επικοινωνία από τον mobile client προς το σταθερό δίκτυο

# Το μοντέλο Client-Intercept-Server



- **C-I-S: Client/Intercept/Server Model**
  - Caching, prefetching, κ.τ.λ.
  - Διάφορες βελτιστοποιήσεις επικοινωνίας και στα δυο άκρα του καναλιού
    - Π.χ., ασύγχρονες queued RPC
    - hoarding
  - Μετακίνηση υπολογισμού μεταξύ των agents
- Το μοντέλο αυτό είναι κατάλληλο για “heavy-weight” clients με αρκετή υπολογιστική ισχύ και πόρους

# Το μοντέλο των Κινητών Agents

- Οι κινητοί agents είναι μετακινούμενες διεργασίες (processes) που σχετίζονται με κάποια “περιήγηση”
  - Έχει τα δικά του δεδομένα και τη δική του κατάσταση εκτέλεσης
  - Προχωρά ανεξάρτητα και αυτόνομα από τον mobile client μετά την υποβολή του
- Υλοποίηση των agents των προαναφερθέντων μοντέλων ως mobile agents, π.χ.,
  - Οι server-side agents μπορούν να μετακινούνται κατά τη διάρκεια του handoff
  - Οι client-side agent μετακινούνται δυναμικά πάνω και μακριά από τον client

# Προτερήματα των Κινητών Agents

- Ικανότητα να **αλληλεπιδρούν** και να συνεργάζονται με άλλους agents
- **αυτονομία**, δηλ., η εκτέλεση προχωρά με μικρή ή καθόλου παρεμβολή από τον client
- **interoperability**, δηλ. εκτελούνται σε πολλές διαφορετικές πλατφόρμες
- Ικανότητα να **αντιδρούν** σε εξωτερικά ερεθίσματα-γεγονότα
- **κινητικότητα**

# Συνολικά για τους Κινητών Agents

- Παρέχουν: αποδοτικό, ασύγχρονο τρόπο για αναζήτηση πληροφορίας
- Δεν επηρεάζονται από “μεταβαλλόμενα” δυναμικά πειβάλλοντα
- Υποστηρίζουν “λειτουργία σε αποσύνδεση”
  - Σε στιγμή σύνδεσης: υποβάλλεται ο agent
  - Σε αποσύνδεση: αναζητεί πληροφορία, εκτελεί επεξεργασία
  - Σε επανασύνδεση: επιστρέφει τα αποτελέσματα
- Παρουσιάζουν προβλήματα **ασφάλειας**
- Μπορούν να χρησιμοποιηθούν σε συνδυασμό με τα υπολογιστικά μοντέλα που προαναφέρθησαν

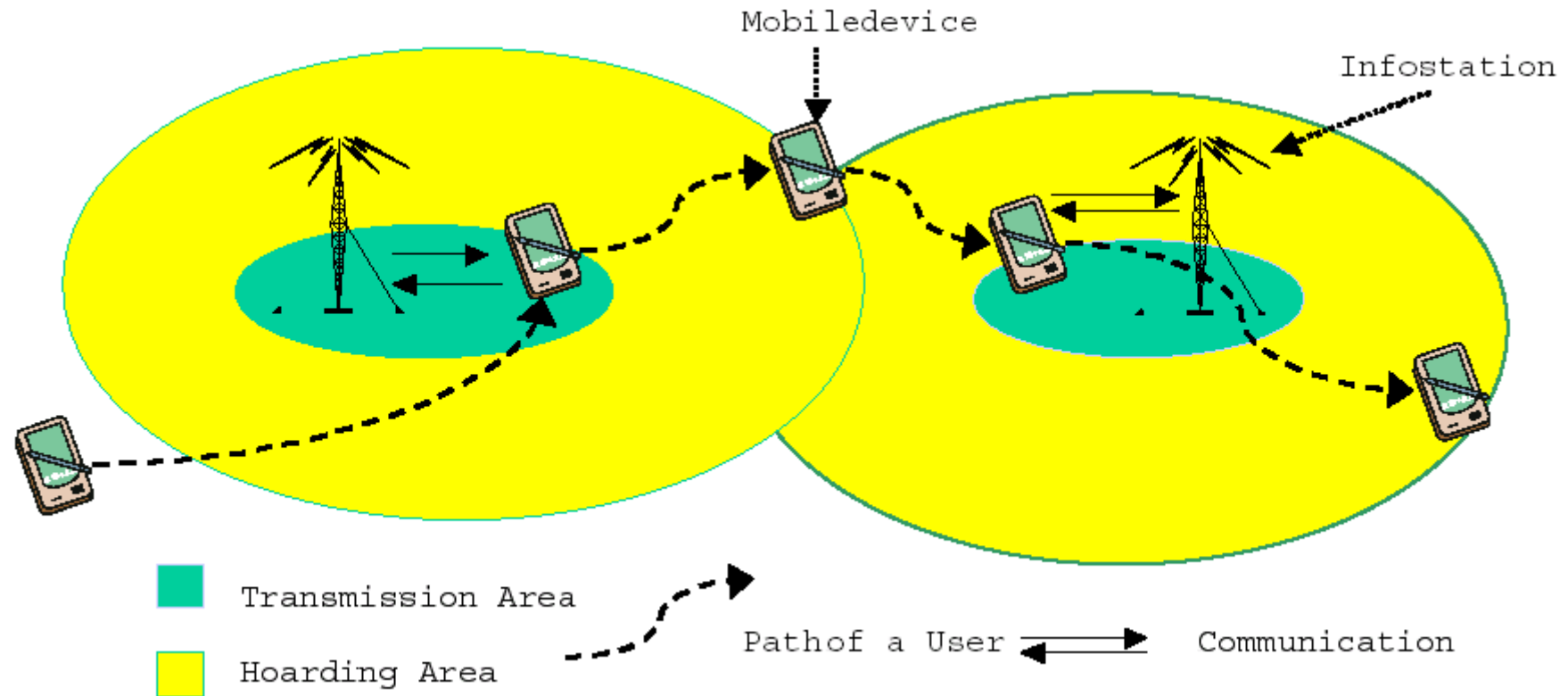
# Κινητός Υπολογισμός: Περιεχόμενα

- Παράδειγμα-κίνητρο
- Ζητήματα: Κινητικότητα, Ασύρματη επικοινωνία, Φορητότητα
- Αρχιτεκτονικές λογισμικού
- **Υποστήριξη σε επίπεδο συστήματος**

# Υποστήριξη σε επίπεδο συστήματος

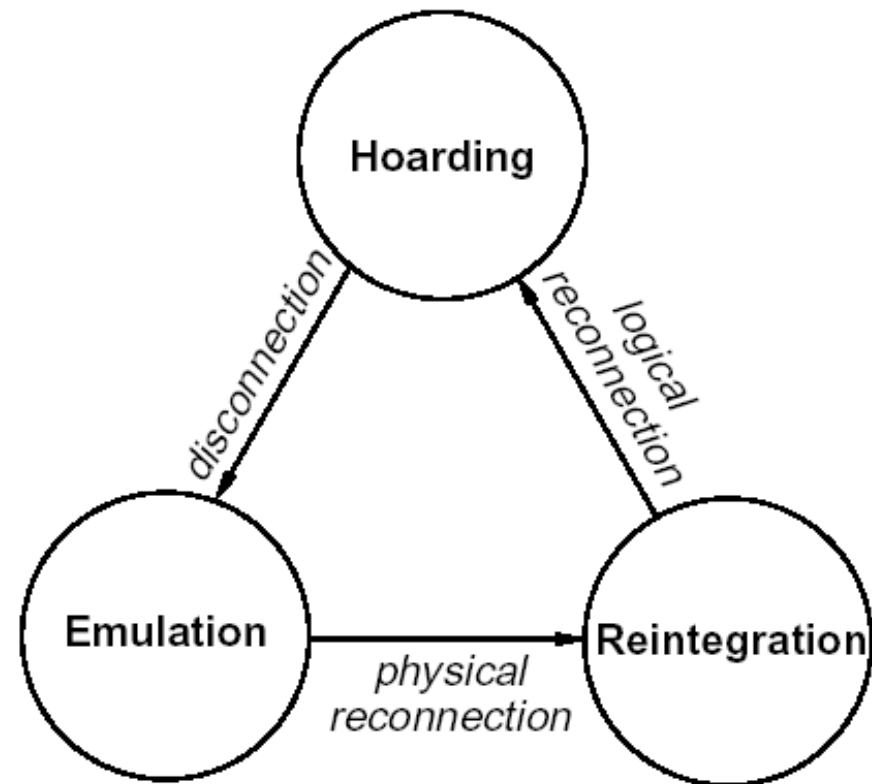
- **Αποσυνδέσεις (disconnections):** αυτόνομη λειτουργία μετά τη διακοπή της σύνδεσης
- **Ασθενής σύνδεση (weak connectivity):** η λειτουργία θα πρέπει να προσαρμόστεί σε χαμηλό εύρος ζώνης, μεγάλη καθυστέρηση πρόσβασης (latency) και υψηλό κόστος πρόσβασης
- **Κινητικότητα (mobility):** διατήρηση λειτουργιών “on-the-move”, π.χ., συναλλαγών, Web browsing
- **Αποκατάσταση αποτυχίας (failure recovery):** τεχνικές για αποκατάσταση του υπολογισμού ή/και των δεδομένων μετά από αποτυχίες του καναλιού, του mobile client

# Μοντέλο της αποσύνδεσης



# Υποστήριξη για αποσυνδέσεις

- **Βασική ιδέα**: όταν αναμένεται αποσύνδεση, τα δεδομένα και ο υπολογισμός μεταφέρονται στον mobile client
- Η “προφόρτωση” δεδομένων αποκαλείται **hoarding**
- Εναλλακτικές:
  - Μετακινούμε τα δεδομένα στον client, ή
  - Κάνουμε replicate τα δεδομένα στον client



# Ζητήματα στο hoarding

|                     | Ζήτημα                                             | Προσέγγιση                                                           |
|---------------------|----------------------------------------------------|----------------------------------------------------------------------|
| <b>Hoarding</b>     | Μονάδα hoarding                                    | Εξαρτάται από την εφαρμογή, π.χ., αρχείο ή εγγραφές ή αντικείμενο    |
|                     | Ποια δεδομένα?                                     | A) Καθορίζεται από το χρήστη. B) Πρόβλεψη                            |
|                     | Πότε να εκτελέσουμε?                               | A) Πριν την αποσύνδεση. B) Περιοδικά.                                |
|                     | Αιτήσεις για δεδομένα που δεν υπάρχουν στον πελάτη | A) “Πετάει” exception. B) Μπαίνουν σε μια ουρά.                      |
| <b>Αποσύνδεση</b>   | Τι γράφουμε στο log?                               | A) Τιμές δεδομένων. B) Timestamps. Γ) Λειτουργίες                    |
|                     | Πότε βελτιστοποιούμε το log?                       | A) Αυξητικά. B) Πριν την επανασύνδεση                                |
|                     | Πώς βελτιστοποιούμε το log?                        | Εξαρτάται από το σύστημα.                                            |
| <b>Επανασύνδεση</b> | Πώς κάνουμε την επανασύνδεση?                      | Εκτελούμε ξανά το log.                                               |
|                     | Πώς επιλύσουμε τις συγκρούσεις?                    | A) Αξιοποιούμε τη σημασιολογία της εφαρμογής.<br>B) Αυτόματη επίλυση |

# Κύρια ζητήματα στο hoarding

- Πώς αποφασίσουμε ποια δεδομένα θα κάνουμε hoarding?
  - Οι clients δηλώνουν τις προτιμήσεις τους
  - Κάνουμε προβλέψεις με βάση το “ιστορικό”
- Τι γίνεται όταν και άλλοι clients (π.χ., στο σταθερό δίκτυο) μπορούν/επιθυμούν να τα τροποποιήσουν ?

# Hoarding σε Συστήματα Αρχείων

- Το hoarding βασίζεται σε επέκταση του μηχανισμού caching για να λάβει υπόψη τις αποσυνδέσεις
- Caching σε αποσύνδεση είναι διαφορετικό από caching σε σύνδεση
  - Cache misses δεν εξυπηρετούνται
  - Ενημερώσεις (updates) από τον mobile client δεν μεταδίδονται στο server και αντίστροφα
- Το hoarding είναι παρόμοιο με το prefetching
  - Το prefetching εκτελείται συνεχώς στο υπόβαθρο
  - Το hoarding είναι ζωτικής σημασίας (εξαιτίας των cache misses)
  - Το hoarding υπερ-εκτιμά τις ανάγκες του χρήστη
  - Η μικρή cache στον mobile clients περιορίζει την εκτίμηση του hoarding

# Hoarding στο CODA σύστ. αρχείων

|                     | Ζήτημα                                             | Προσέγγιση                                                                                   |
|---------------------|----------------------------------------------------|----------------------------------------------------------------------------------------------|
| <b>Hoarding</b>     | Μονάδα hoarding                                    | Αρχείο                                                                                       |
|                     | Ποια δεδομένα?                                     | Καθορίζεται από το χρήστη + Πρόβλεψη (προτεραιότητά στα πρόσφατα αρχεία)                     |
|                     | Πότε να εκτελέσουμε?                               | Περιοδικά με τη διεργασία <i>hoard walk</i> .                                                |
|                     | Αιτήσεις για δεδομένα που δεν υπάρχουν στον πελάτη | Επιλεκτική εξυπηρέτηση, ανάλογα με την “υπομονή”                                             |
| <b>Αποσύνδεση</b>   | Τι γράφουμε στο log?                               | (replay.log) Όλα τα ορίσματα των κλήσεων συστήματος και version numbers των αρχείων.         |
|                     | Πότε βελτιστοποιούμε το log?                       | Περιοδικά                                                                                    |
|                     | Πώς βελτιστοποιούμε το log?                        | A) Λειτουργίες που αναιρούν τον εαυτό τους ή/και προηγούμενες. B) Ομαδοποιώντας λειτουργίες. |
| <b>Επανασύνδεση</b> | Πώς κάνουμε επανασύνδεση?                          | Εκτελεί το replay log ως μια transaction. Κάνει lock τα αντικείμενα που αναφέρονται στο log. |
|                     | Πώς επιλύουμε τις συγκρούσεις?                     | Με το Application-Specific Resolver (ASR)                                                    |

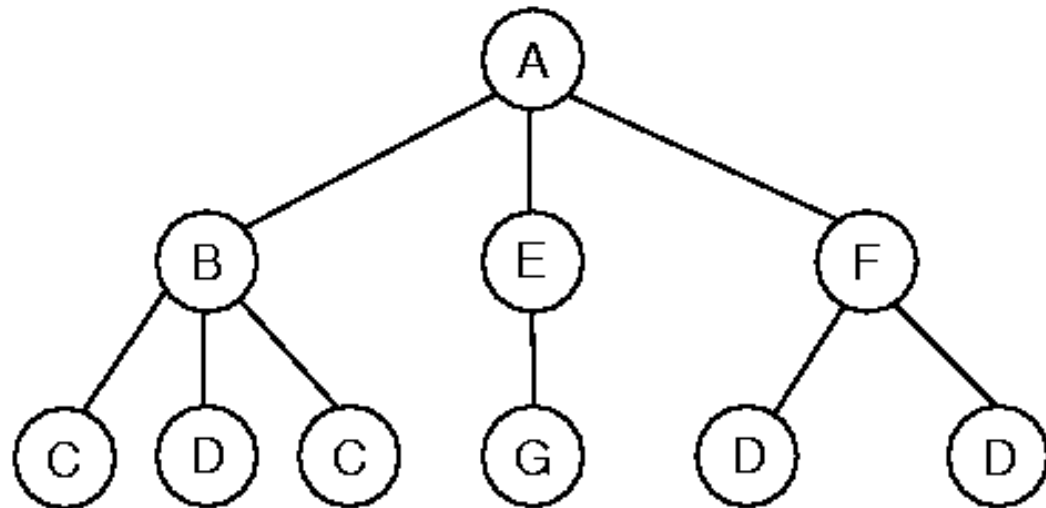
# Hoarding σε Βάσεις Δεδομένων

- Τι να κάνω hoarding
  - RelDB: Tuples / Συλλογές από tuples / Πίνακες
  - OODB: Αντικείμενα / Συλλογές αντικειμένων / Κλάσεις
  - Τα αποτελέσματα ερωτήσεων (queries): άρα materialized views στον mobile client  $\Rightarrow$  answering queries from materialized views
- Τα υπόλοιπα ζητήματα όπως και στα συστήματα αρχείων

# Hoarding με Δένδρο Εξαρτήσεων

- Αρχεία (προγράμματα και αρχεία δεδομένων) είναι οι κόμβοι του δένδρου, και υπάρχει μια ακμή από τον πατέρα A στο παιδί B, εάν
  - Το πρόγραμμα A εκτελεί το πρόγραμμα B,
  - Το πρόγραμμα A ανοίγει το αρχείο δεδομένων B.
- Η σειρά των αδερφών αντικατοπτρίζει τη χρονολογία των προσπελάσεων.

1. Το πρόγραμμα A εκτελεί το πρόγραμμα B
2. Το B ανοίγει τα αρχεία C και D, με αυτή τη σειρά
3. Το B ανοίγει το C ξανά
4. Το A εκτελεί τα προγράμματα E και F
5. Το F ανοίγει το D
6. Το E εκτελεί το πρόγραμμα G
7. Το F ανοίγει το D ξανά



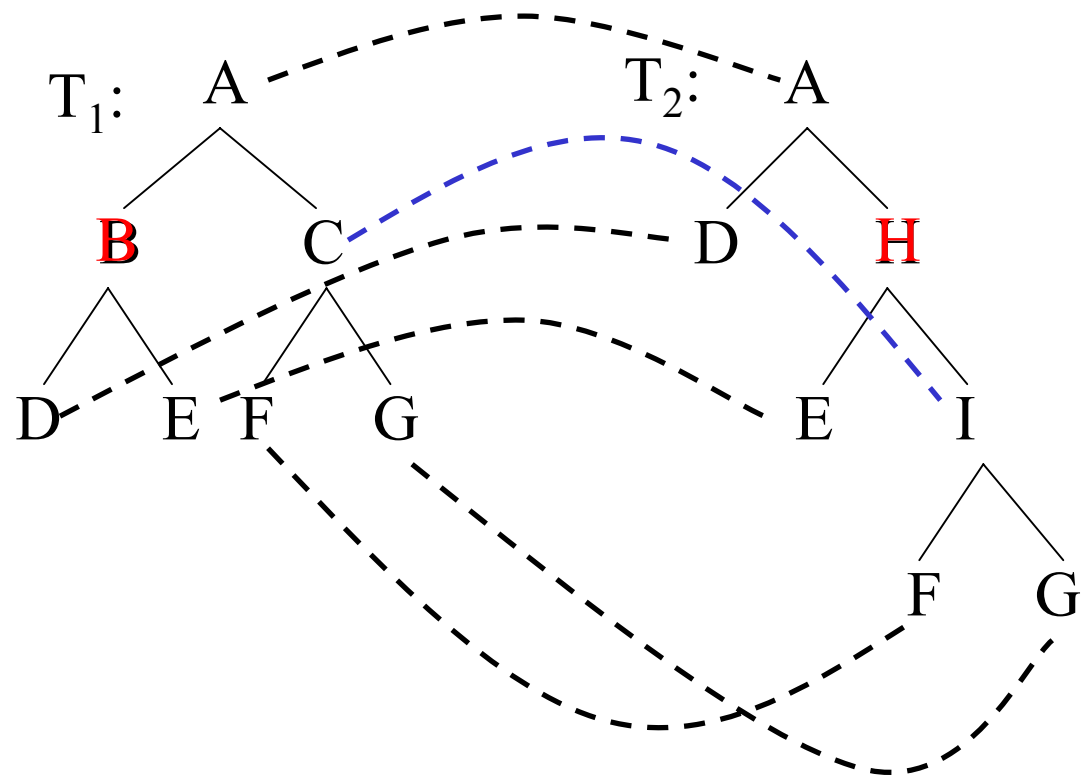
# Hoarding με Δένδρο Εξαρτήσεων

- Αλγόριθμος για hoarding με Δένδρο εξαρτήσεων
  - Χρησιμοποιούμε μια συνάρτηση **COMPATIBILITY**(*w\_tree*, *p\_tree*), η οποία επιστρέφει μια τιμή μεταξύ 0 και 1, για την ομοιότητα ενός working δένδρου *tree w\_tree* και ένα pattern δένδρο *p\_tree*. Εάν η τιμή της είναι πάνω από μια σταθερά **MATCH\_THRESHOLD**, τότε λέμε ότι δένδρα είναι αρκετά όμοια και μπορούμε να κάνουμε hoarding τα αρχεία που περιγράφει το *p\_tree*.
  - Προβλήματα:
    - Ο αριθμός των αρχείων που θα κάνουμε hoarding είναι μεγαλύτερος από ότι μπορεί να χωρέσει η cache
    - Οι εσφαλμένες προβλέψεις την καθιστούν μη αποτελεσματική
  - Θέτουμε ένα άνω όριο (**PREFETCH\_CAPACITY**) στον αριθμό των αρχείων που θα κάνουμε hoarding και τοποθετούμε στην cache μόνο τα ανώτερα επίπεδα του δένδρου.

# Hoarding με Δένδρο Εξαρτήσεων

- Tree edit distance: ο **ελάχιστος** αριθμός λειτουργιών tree edit (εισαγωγή κόμβου, διαγραφή κόμβου, μετονομασία ετικέτας) που απαιτούνται για να μετατραπεί ένα δένδρο σε κάποιο άλλο.
- Δεδομένων δυο δένδρων  $T_1$  και  $T_2$ , υπάρχει αλγόριθμος υπολογισμού της tree edit distance σε  $O(|T_1| |T_2| h(T_1) h(T_2))$ .

- Π.χ., η tree edit distance είναι **3** (delete B, insert H, relabel C σε I)



# Hoarding με Γράφημα Εξαρτήσεων

- Ένα **Γράφημα Εξαρτήσεων** (Dependency Graph, DG), διατηρεί το pattern των προσπελάσεων σε διαφορετικά αρχεία.
- Το γράφημα αυτό έχει έναν κόμβο για κάθε αρχείο που έχει προσπελαστεί κάποια στιγμή στο παρελθόν.
- Υπάρχει μια ακμή από τον κόμβο  $X$  στον κόμβο  $Y$ , εάν και μόνο εάν (κάποια στιγμή) το αρχείο  $Y$  προσπελάστηκε μέσα σε  $w$  προσπελάσεις μετά το  $X$ , όπου το  $w$  είναι το **lookahead window**.
- Το βάρος πάνω στην ακμή είναι το κλάσμα του αριθμού των προσπελάσεων από το  $X$  στο  $Y$ , προς τον συνολικό αριθμό των προσπελάσεων στο  $X$ .
- Η **ακρίβεια** και η “**επιβάρυνση**” του hoarding μπορεί να ελεγχθεί με μια παράμετρο ορισμένη από το χρήστη, το **κατώφλι αποκοπής** (cut-off threshold) για τα βάρη των ακμών.

# Hoarding με Γράφημα Εξαρτήσεων

- Έστωσαν οι παρακάτω ακολουθίες προσπελάσεων (sessions) του χρήστη:
  - ABCACBD
  - CDABDCBDA
- Lookahead window:  $w = 2$
- Για την πρώτη ακολουθία, βρίσκω τα διαδοχικά παράθυρα μήκους  $w+1$ , δηλ.,
  - ABC
  - BCA
  - CAC
  - ACB
  - CBD
  - BD

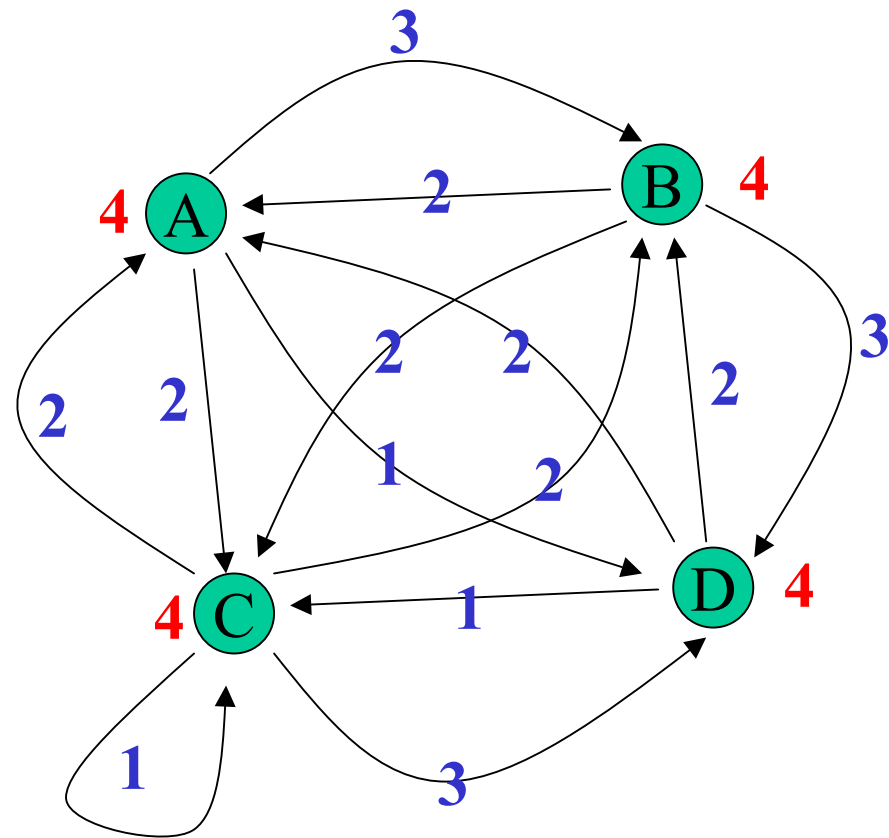
Κάθε παράθυρο, υπονοεί εξάρτηση της προσπέλασης του πρώτου με κάθε μια από τις επόμενες, δηλ.,

- ABC:  $A \Rightarrow B$  &  $A \Rightarrow C$
- BCA:  $B \Rightarrow C$  &  $B \Rightarrow A$
- CAC:  $C \Rightarrow A$  &  $C \Rightarrow C$
- ACB:  $A \Rightarrow C$  &  $A \Rightarrow B$
- CBD:  $C \Rightarrow B$  &  $C \Rightarrow D$
- BD:  $B \Rightarrow D$

# Hoarding με Γράφημα Εξαρτήσεων

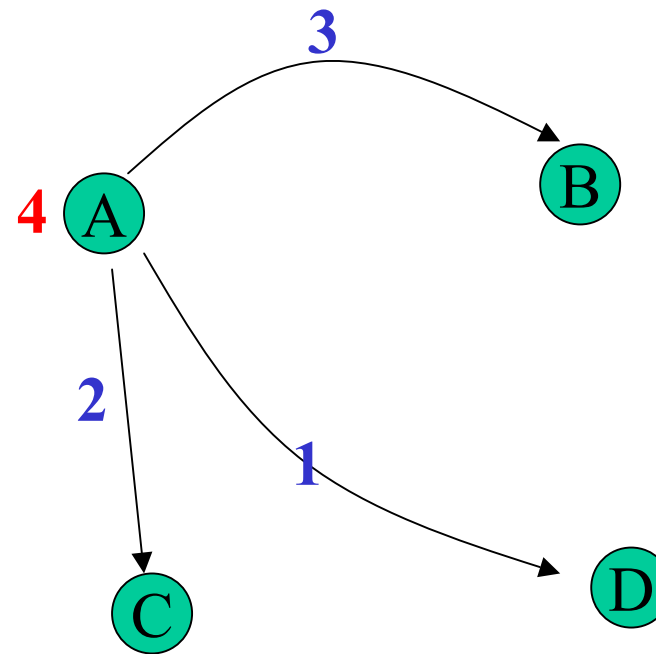
- Συνολικές εμφανίσεις

- |                        |                        |
|------------------------|------------------------|
| – A: 4                 | – $C \Rightarrow A: 2$ |
| – B: 4                 | – $C \Rightarrow B: 2$ |
| – C: 4                 | – $C \Rightarrow C: 1$ |
| – D: 4                 | – $C \Rightarrow D: 3$ |
| – $A \Rightarrow A: 0$ | – $D \Rightarrow A: 2$ |
| – $A \Rightarrow B: 3$ | – $D \Rightarrow B: 2$ |
| – $A \Rightarrow C: 2$ | – $D \Rightarrow C: 1$ |
| – $A \Rightarrow D: 1$ | – $D \Rightarrow D: 0$ |
| – $B \Rightarrow A: 2$ |                        |
| – $B \Rightarrow B: 0$ |                        |
| – $B \Rightarrow C: 2$ |                        |
| – $B \Rightarrow D: 3$ |                        |



# Hoarding με Γράφημα Εξαρτήσεων

- Σε ένα περιβάλλον με:
  - μικρής διάρκειας αποσυνδέσεις
  - μικρή cache στον mobile client
  - τελευταία προσπέλαση στο αρχείο A



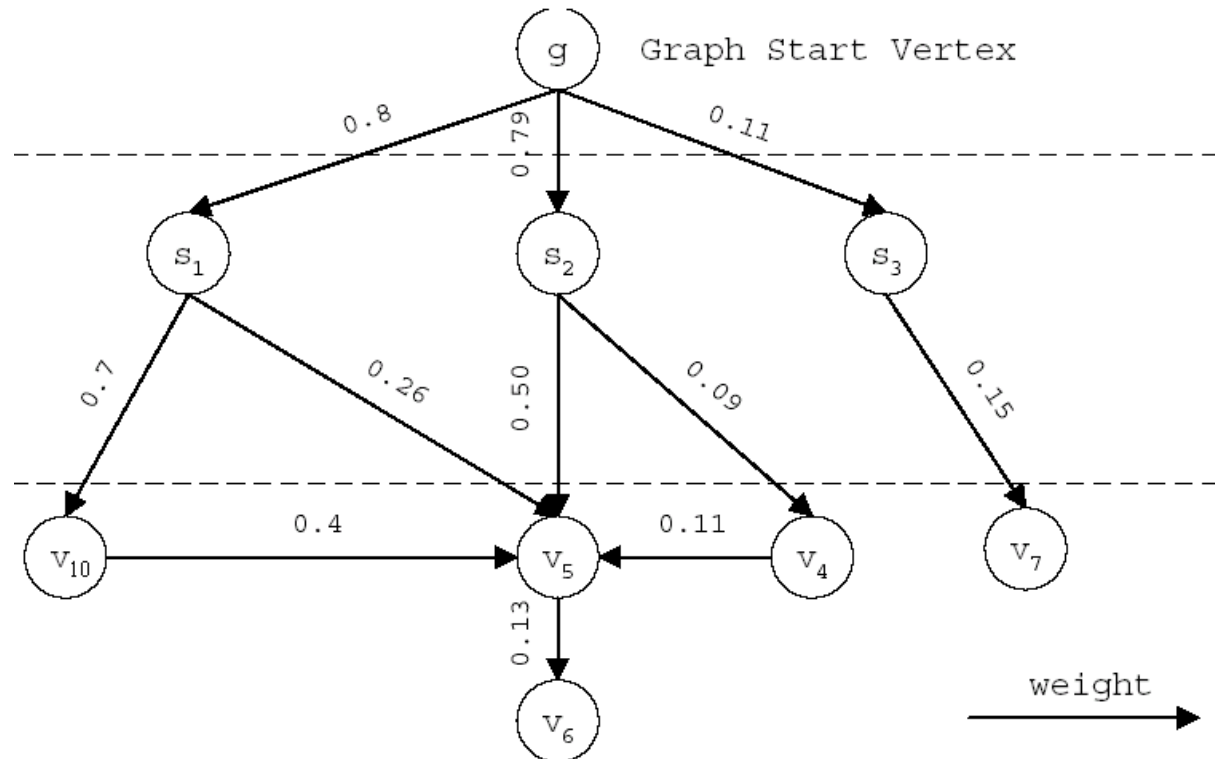
- Θα έθετα κατώφλι = 60% και θα έκανα hoarding το:
  - αρχείο B, επειδή  $\frac{3}{4}=0.75 > 0.60$

# Hoarding με Βελτιωμένο ΓΕ

- Κάποιος κόμβος θεωρείται η “αρχή” του γραφήματος. Διασχίζουμε το γράφημα με:
  - **Breadth First Search**: εξετάζονται τα αδέρφια (siblings) πρώτα
  - **Depth First Search**: εξετάζονται τα παιδιά (children) πρώτα
  - $\gamma_{\text{BFS}}$ : καθορίζει τον αριθμό των siblings που θα λάβουν μέρος στην αναζήτηση
  - $\gamma_{\text{DFS}}$ : καθορίζει το βάθος μέχρι το οποίο θα διεξαχθεί η αναζήτηση

# Hoarding με Βελτιωμένο ΓΕ

| Κόμβος   | Μέγεθος |
|----------|---------|
| $s_2$    | 512     |
| $v_4$    | 624     |
| $v_5$    | 824     |
| $v_{10}$ | 3800    |
| $s_1$    | 5800    |
| $v_8$    | 110     |
| $v_6$    | 552     |
| $s_3$    | 642     |
| $v_7$    | 476     |



- Μέγεθος cache: **12000**
- BFS:  $s_1, s_2, s_3, v_{10}, v_5$  : **11578**
- DFS:  $s_1, v_{10}, v_5, v_6, s_2$  : **11488**

## Σύγκριση του ΔΕ με το ΓΕ

- Το ΔΕ παρουσιάζει πιο λεπτομερή εικόνα για τις εξαρτήσεις, ενώ το ΓΕ “χάνει” την ακριβή δομή τους
- Το ΓΕ αποτυπώνει μόνο ζεύγη εξαρτήσεων
- Η εύρεση των αρχείων για hoarding με το ΔΕ είναι πιο δύσκολη
  - ο υπολογισμός της συνάρτησης **COMPATIBILITY(w\_tree, p\_tree)** είναι χρονοβόρος
  - δυσκολία στην εύρεση κατάλληλης συνάρτησης
- Το ΓΕ είναι πιο μικρό σε απαιτήσεις χώρου

# Υποστήριξη για ασθενή σύνδεση

- Τι είναι η ασθενής σύνδεση
  - Αργό (slow) δίκτυο
  - Ακριβή πρόσβαση
  - Χάνεται περιοδικά για μικρά διαστήματα
- Αντιμετώπιση της ασθενούς συνδέσεως
  - Έξυπνη χρήση του εύρους ζώνης
  - Ελάττωση της επικοινωνίας

# Ασθενής σύνδεση σε συστημ. αρχείων

- Επανεξέταση της κύριας λειτουργίας που σχετίζεται με το δίκτυο: caching
  - Χειρισμός των misses
  - Συχνότητα αποστολής των αλλαγών που εκτελούνται από τον client
  - Εγκυρότητα των cached αντικειμένων
  - Ενημέρωση του mobile client για την εγκυρότητα των δεδομένων της cache (σε επόμενη Διάλεξη: **Invalidation Reports**)

# Cache misses σε Ασθενή σύνδεση

- Η εξυπηρέτηση ενός cache miss προκαλεί μεγάλη καθυστέρηση ή είναι ακριβή
  - Επιλεκτική εξυπηρέτηση ανάλογα με τις ανάγκες της εφαρμογής και τη σύνδεση
- Αποστολή αλλαγών
  - Επιθετική πολιτική
    - Ελαττώνει την πιθανότητα βελτιστοποιήσεων στο log
    - Αυξάνει την επικοινωνία
    - Επιτυγχάνει συνέπεια στα δεδομένα
    - Ελαττώνει την πιθανότητα συγκρούσεων
    - Κρατά μικρό το μέγεθος του log
  - “Τεμπέλικη” (Lazy) πολιτική
    - Δεν επιτρέπει αποδοτική χρήση της cache, αφού τα τροποποιημένα δεδομένα δεν μπορούν να διωχτούν από την cache
    - Δεν εγγυάται “strong consistency”

# Σύνδεση και Συστήματα Αρχείων

|                             | Με σύνδεση                 | Χωρίς σύνδεση             | Ασθενής σύνδεση                                         | Fetch-only                |
|-----------------------------|----------------------------|---------------------------|---------------------------------------------------------|---------------------------|
| <b>Cache miss</b>           | Εξυπηρέτηση                | Exception                 | (Επιλεκτική)<br>Εξυπηρέτηση                             | Εξυπηρέτηση               |
| <b>Αποστολή των αλλαγών</b> | Άμεση                      | Όταν γίνει η επανασύνδεση | Περιοδική ή όταν υπάρχει μικρή κυκλοφορία στο κανάλι    | Όταν γίνει η επανασύνδεση |
| <b>Εγκυρότητα δεδομένων</b> | Η τελευταία (πιο πρόσφατη) | Ό,τι έχει η cache         | Ό,τι έχει η cache (έγκυρη ή μη) ή επικοινωνία με server | Ό,τι έχει η cache         |