

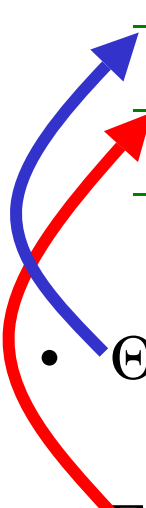
Κινητός και Διάχυτος Υπολογισμός (Mobile & Pervasive Computing)

Δημήτριος Κατσάρος, Ph.D.

Χειμώνας 2005

Διάλεξη 7η

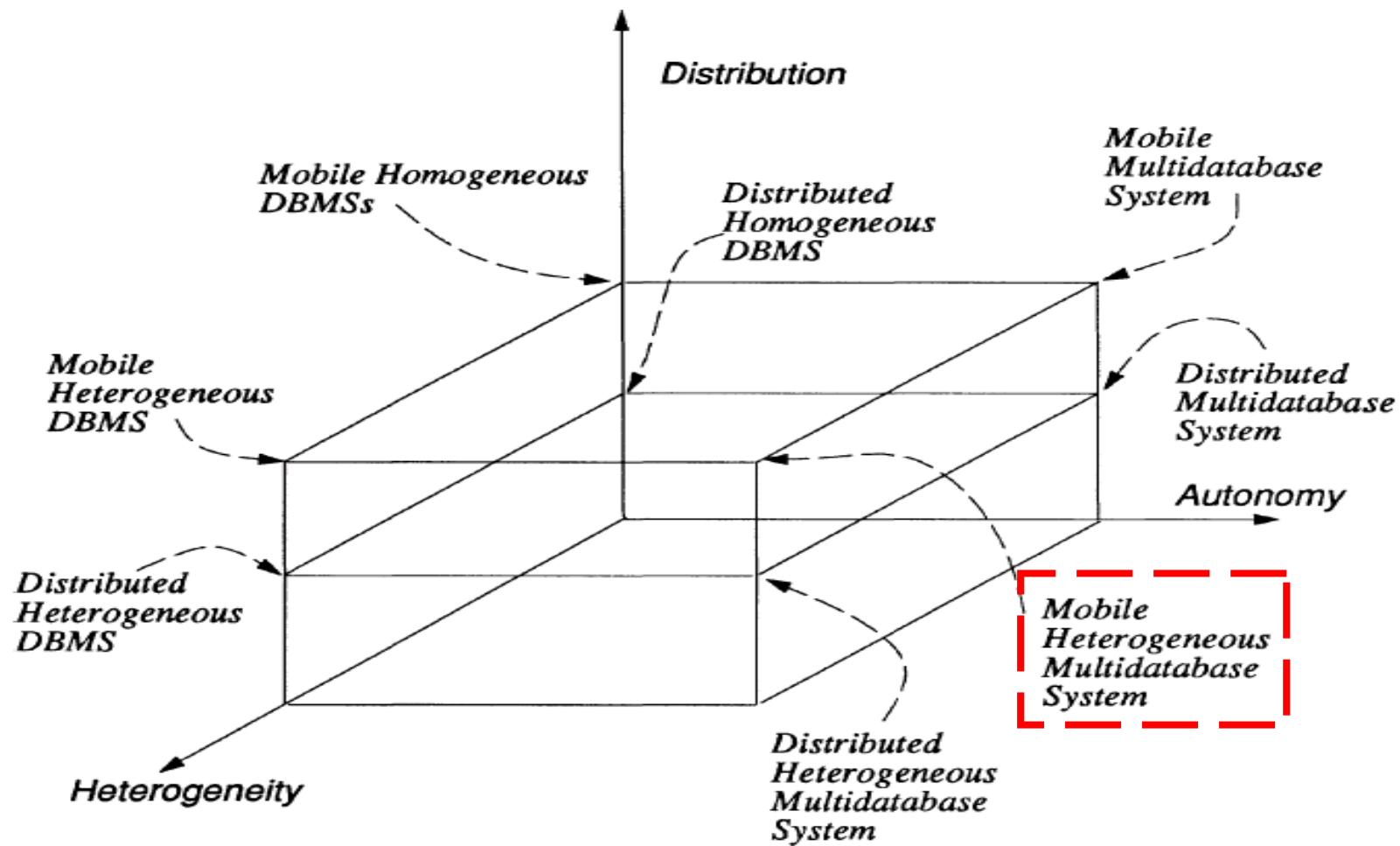
Ιστοσελίδα του μαθήματος

- http://skyblue.csd.auth.gr/~dimitris/courses/mpc_fall05.htm
 - http://skyblue.csd.auth.gr/~dimitris/courses/mpc_fall05/
 - **books/**
 - **Σύντομα: “Handbook of Wireless Networks and Mobile Computing”**
 - **lectures/**
 - **papers/**
 - **proj_papers/**
 - Θα τοποθετούνται οι διαφάνειες του επόμενου μαθήματος
 - Σταδιακά θα τοποθετηθούν και τα research papers που αντιστοιχούν σε κάθε διάλεξη
- 
- A diagram consisting of two curved arrows. A blue arrow starts from the 'proj_papers/' directory and points to the 'lectures/' directory. A red arrow starts from the 'proj_papers/' directory and points to the 'papers/' directory.

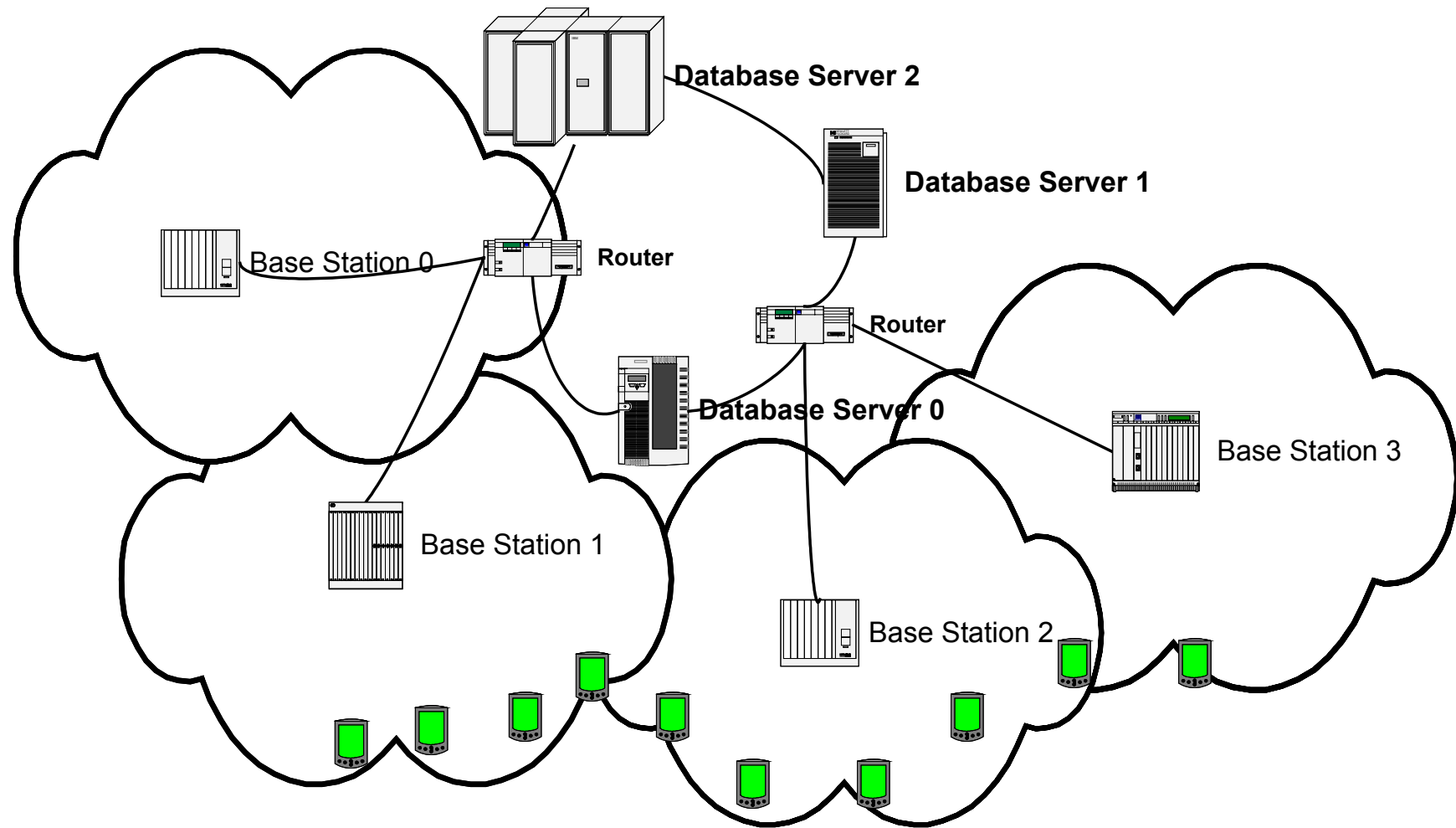
Περιεχόμενα

- **Συναλλαγές**
- Αρχιτεκτονική δικτύου
- Μοντέλα κινητικότητας
- Διαχείριση θέσης: Registration
- Διαχείριση θέσης: Paging

Κατηγοριοποίηση των DBMS



Παράδειγμα



Περιγραφή προβλήματος

- Επεξεργασία συναλλαγών σε Mobile Computing συστήματα
 - Κατανεμημένο περιβάλλον
 - Πολλαπλά DBMS
 - Ετερογενή DBMS
 - Κινητοί χρήστες

Γιατί είναι διαφορετικό?

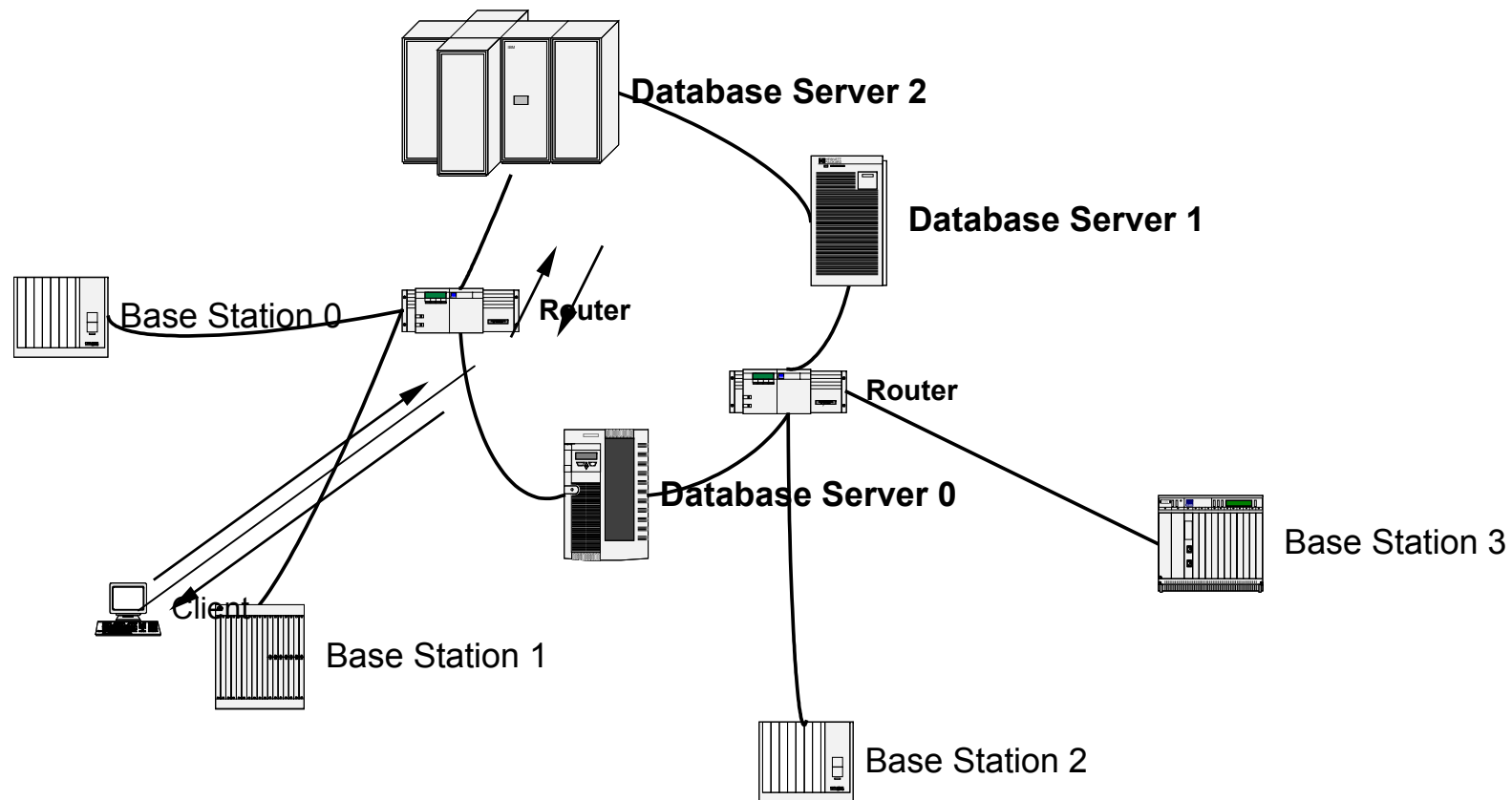
- Η κατάσταση μιας συναλλαγής “εκτείνεται” σε πολλούς σταθμούς βάσης και βάσεις δεδομένων
 - Το κινητό (ίσως) φύγει από ένα κελί πριν ολοκληρωθεί η συναλλαγή
 - Το κινητό ίσως χρειαστεί να κάνει commit ή abort τις λειτουργίες ή συναλλαγές, οι οποίες δεν είναι στον τρέχοντα σταθμό βάσης
 - Πρόβλημα recovery

Ιδιότητες συναλλαγών ACID

- **Atomicity:**
 - Είτε η συναλλαγή ολοκληρώνεται, ή δεν συμβαίνει τίποτε
- **Consistency:**
 - Μια συναλλαγή πρέπει να αρχίσει σε consistent κατάσταση και να αφήσει το σύστημα σε consistent κατάσταση
- **Isolation:**
 - Οι συναλλαγές είναι απομονωμένες από ταυτόχρονες διαδικασίες
- **Durability:**
 - Αφού μια συναλλαγή ολοκληρωθεί (committed), τα αποτελέσματά της είναι Durable

Κοινές συναλλαγές (1/2)

- Παραδοσιακή επεξεργασία συναλλαγών



Κοινές συναλλαγές (2/2)

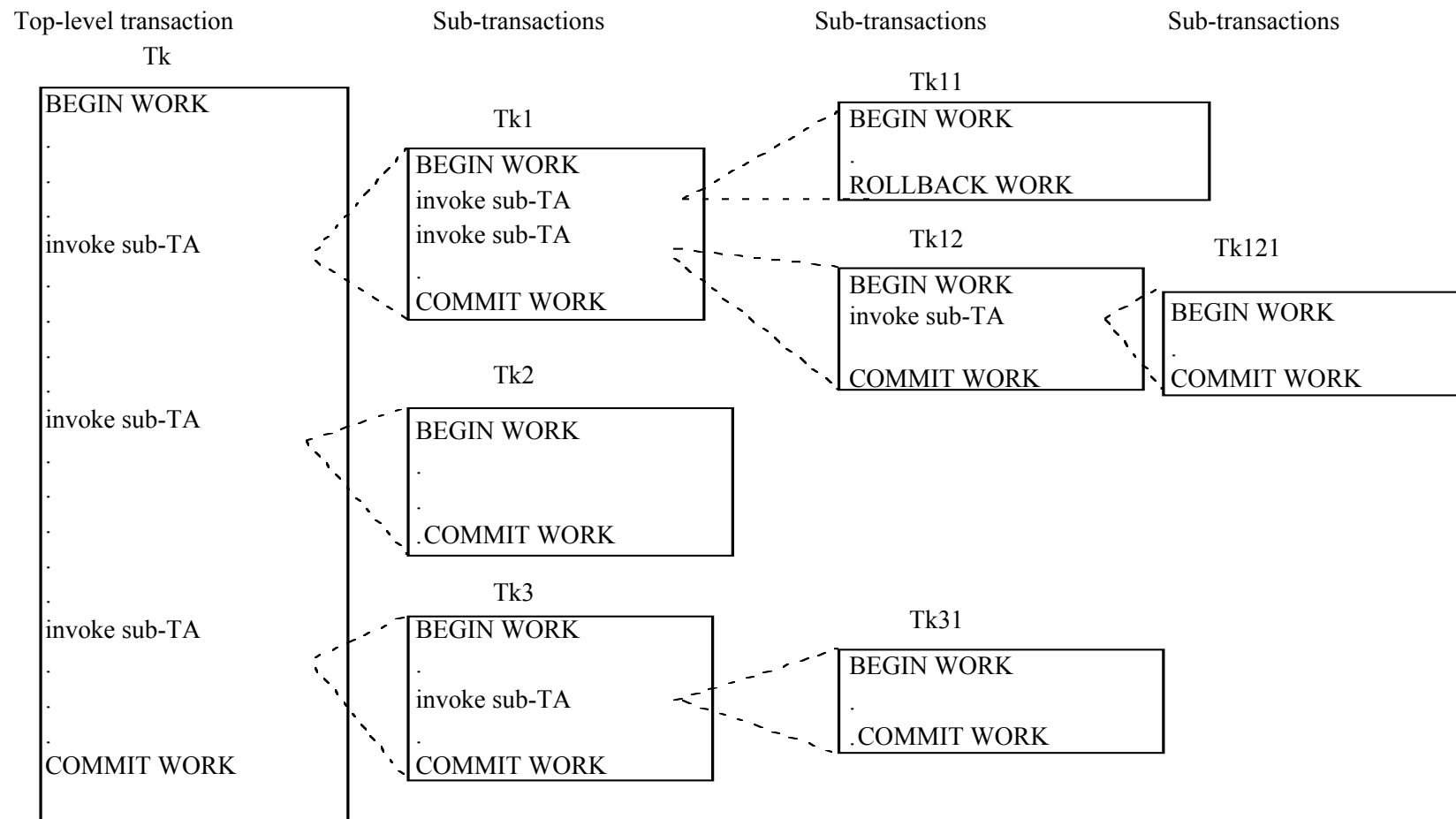
- Γνωστή database (Τυπικά μια)
- Περιορισμένη διάρκεια (σε σύγκριση με long transactions)
- Λίγες ή καθόλου αλληλεπιδράσεις με άλλα ταυτόχρονα γεγονότα
- Οι ιδιότητες ACID είναι εύκολο να επιτευχθούν

Συναλλαγές σε Multi-Database

- Λειτουργούν σε δεδομένα διαφόρων βάσεων (γνωστών ή αγνώστων)
 - Διατηρούν Multi-database Directory για την εύρεση των σχημάτων των βάσεων
- Οι βάσεις ίσως δεν γνωρίζουν η μια την άλλη
 - Οι βάσεις είναι αυτόνομες
 - Το GDBS (Global Data Base System) συντονίζεται με διαφορετικές βάσεις
 - GDBS κρατά την αρχική database άθικτη, και έτσι δεν υπάρχει global lock, ούτε two-phase lock, ούτε two phase commit
- Οι ιδιότητες ACID είναι δύσκολο να επιτευχθούν

Φωλιασμένες συναλλαγές

- Μια φωλιασμένη συναλλαγή είναι ένα δένδρο συναλλαγών, τα υποδένδρα του οποίου είναι είτε φωλιασμένες ή επίπεδες συναλλαγές



Ανοιχτές φωλιασμένες συναλλαγές

- Στόχος: Η βελτίωση του throughput
- Ιδέα: Εγκατάλειψη της ιδιότητας Isolation
 - Η τροποποίηση μιας υπο-συναλλαγής θα είναι ορατή σε όλες τις συναλλαγές
- Λύση: Compensation Function
 - Η λειτουργία abort θα επαναφέρει την τροποποίηση
- Προβλήματα: Η Compensation δεν είναι αρκετή

Long Transaction

- Ανεπίσημος ορισμός: Οι συναλλαγές που διαρκούν τον “ίδιο” χρόνο όσο είναι ο μέσος χρόνος αποτυχίας (mean time failure) του συστήματος όπου εκτελούνται
- Παράδειγμα:
 - Συναλλαγές CAD/CAD, συναλλαγές VLSI Layout
- Προβλήματα:
 - Ευπαθείς στα Crash
 - Concurrency Bottleneck

Split Transaction

- Στόχος:
 - Επίλυση των προβλημάτων των long transactions
- Ιδέα:
 - Διαίρεση των τρεχουσών συναλλαγών σε δυο serializable συναλλαγές
 - Commit των δεδομένων που δεν πρόκειται να αλλάξουν
- Πλεονέκτημα:
 - Προσαρμοζόμενη recovery, added concurrency και Serializable προσπέλαση στους πόρους

Προαπαιτούμενα για Split Transaction

- Υποθέστε ότι η συναλλαγή T σπάει σε A και B και ότι η A προηγείται της B

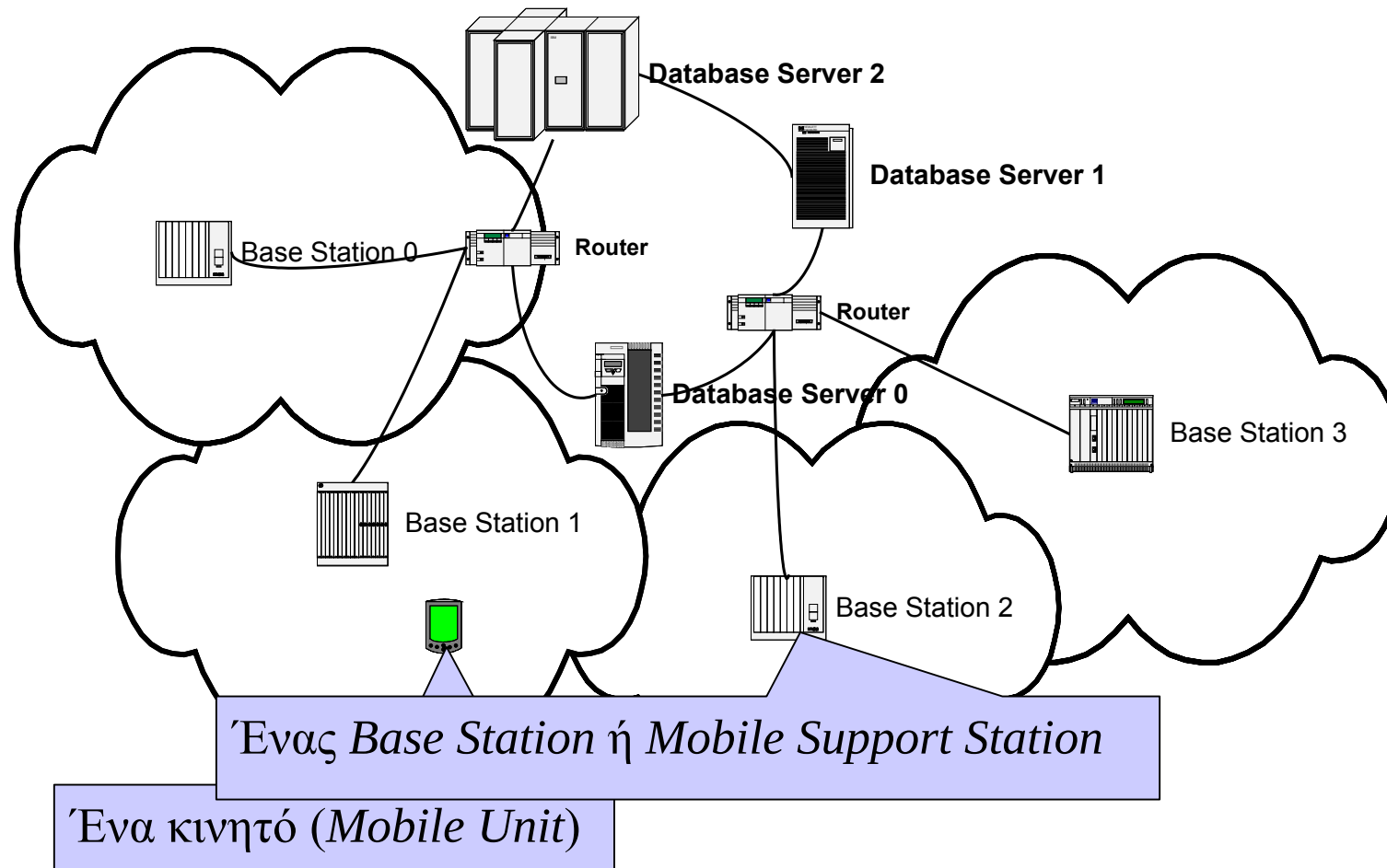
1. $AWriteSet \cap BWriteSet = BWriteLast$

2. $AReadSet \cap BWriteSet = \emptyset$

3. $BReadSet \cap AWriteSet = ShareSet$

Συναλλαγές Kangaroo

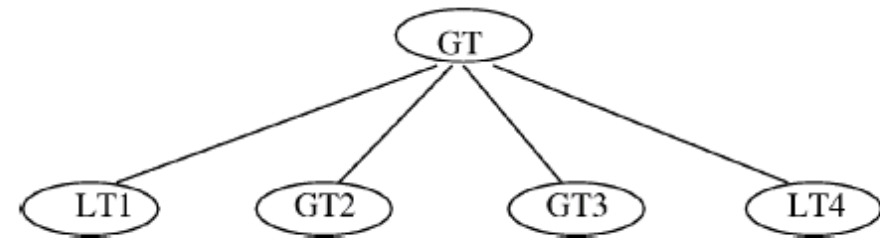
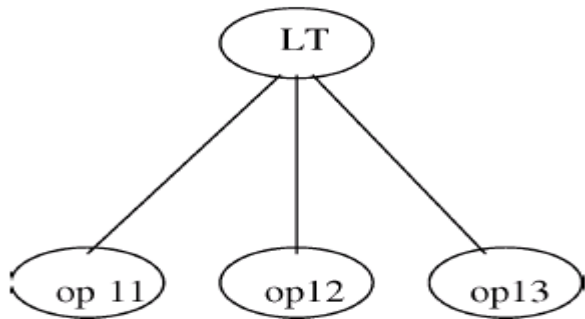
Ορολογία (1/2)



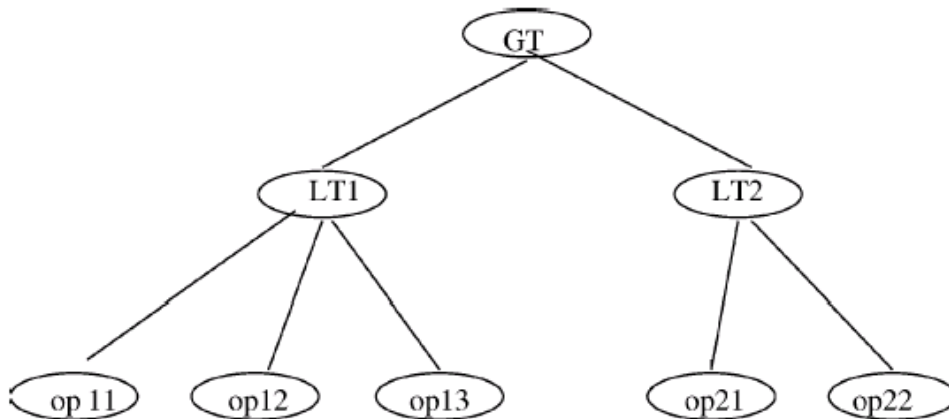
Ορολογία (2/2)

- Source System
 - Συλλογή συστημάτων που παρέχουν πληροφορίες στους κινητούς χρήστες
 - Παράδειγμα: Μια κατανεμημένη βάση δεδομένων
- Data Access Agent (DAA)
 - Ένα πρόγραμμα στους σταθμούς βάσης που παρέχει στους κινητούς χρήστες πρόσβαση στο source system
- Mobile Transaction Manager (MTM)
 - Συνιστώσα του DAA που διαχειρίζεται τις κινητές συναλλαγές

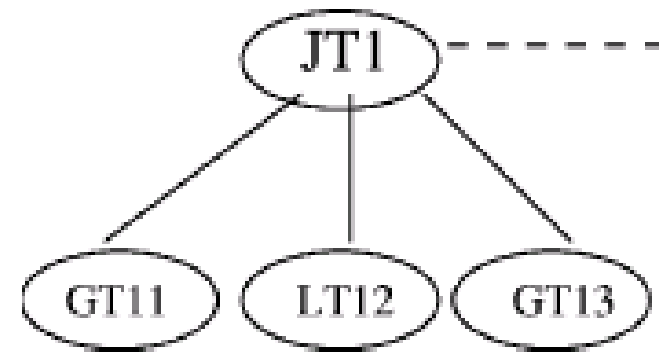
Local και Global συναλλαγές



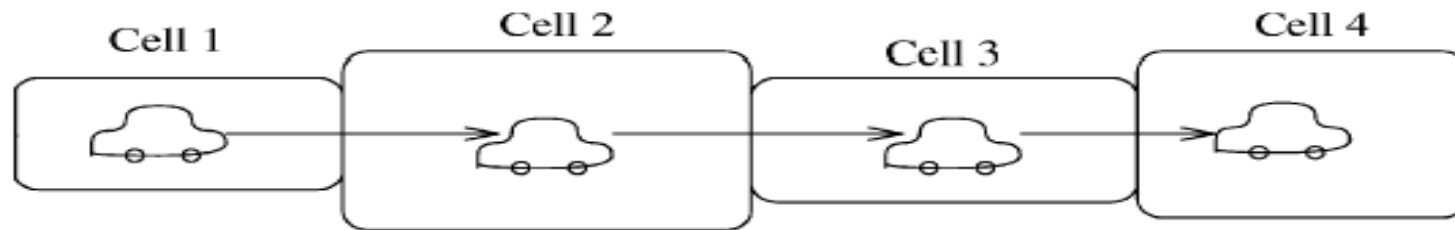
(b) Global Transaction view



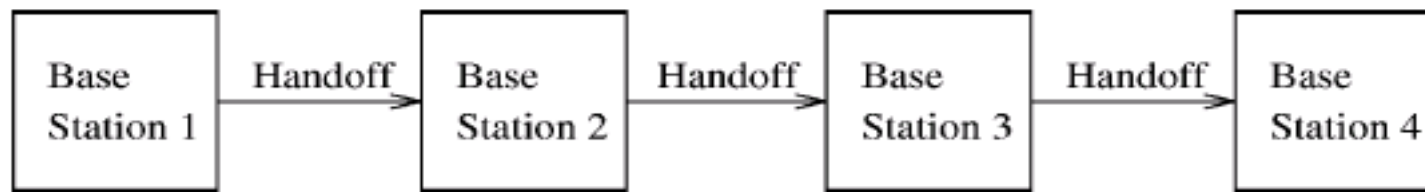
(a) Limited Global Transaction view



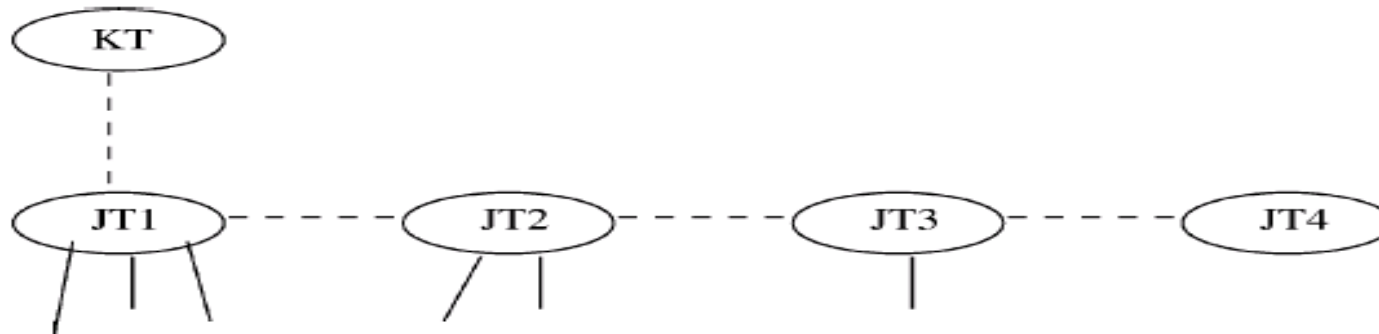
Προεπισκόπηση λύσης



(a) Movement of Mobile Unit through Cells

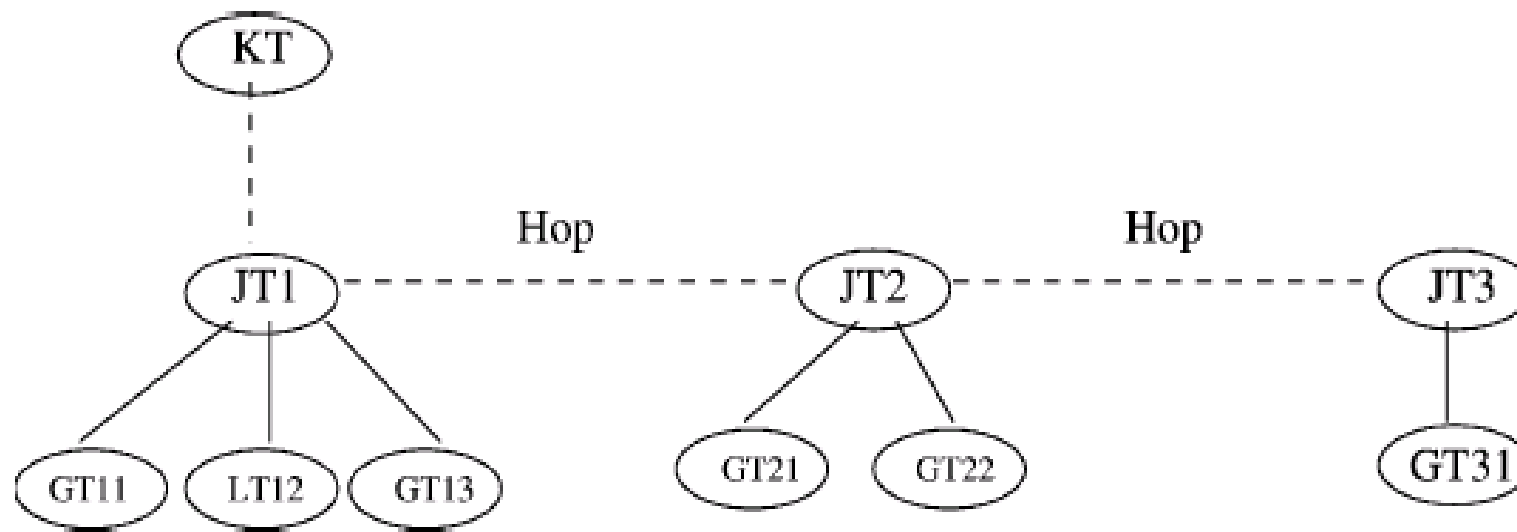


(b) Hopping from Base Station to Base Station



(c) Kangaroo Transaction

Συναλλαγή Kangaroo



Ορισμός συναλλαγής Kangaroo (1/2)

Definition 1. A *Local Transaction* LT is a sequence of read (r_i) and write (w_i) operations ending in either a commit(c_i) or abort(a_i) operation.

Definition 2. A *Global Transaction* GT is any sequence of global transactions G_j and local transactions L_j .

Definition 3. A *Joey Transaction* JT is a sequence of zero or more global transactions G_k and local transactions L_k followed by either a commit c_k , abort a_k , or split s_k operation.

Ορισμός συναλλαγής Kangaroo (2/2)

Definition 4. A *Kangaroo Transaction* KT is a sequence of one or more Joey Transactions J_l . The last Joey Transaction must end in a commit c_l or abort a_l . All Joey Transactions other than the last one must end in a split s_l .

Definition 5. The sequence of local and global transactions which belong to a Kangaroo Transaction is called its *Pouch*.

Definition 6. Two Kangaroo Transactions are said to be *Equivalent Kangaroo Transactions* if they have exactly the same pouch.

Modes επεξεργασία συναλλαγής

- Compensating Mode
 - One JT fails -> all JTs undone
 - Users provide compensating functions for committed JTs. (Difficult Problem)
 - Atomicity Provided, Isolation Violated (Author's Claim)
- Split Mode
 - One JT failure -> No more new LT or GT for that JT
 - Committed JTs are not affected
- Neither guarantees KT serializability

Επεξεργασία συναλλαγής (1/5)

- Transaction Initialization
 - MTM creates a Kangaroo Transaction and a unique KTID
 - BTKT -> MTM Transaction Log
 - MTM creates first Joey Transaction and a unique JTID
 - BTJT -> MTM Transaction Log
 - JT entry -> Transaction Status Table
 - Increase Joey Transaction Number

Επεξεργασία συναλλαγής (2/5)

- Transaction Execution
 - Translate JT operations to global and local transactions
 - ST entry -> transaction table
 - BTST -> MTM Log
 - Send native transaction to DBMS or GDBS

Επεξεργασία συναλλαγής (3/5)

- Handoff Process
 - At Original Base Station
 - Original DAA starts split operation (Problem? Possibility? How does it know?)
 - HOKT -> Log
 - Flash Log to Disk (Checkpoint)
 - At Destination Base Station
 - CTKT -> Log
 - KT entry -> Transaction Status Table

Επεξεργασία συναλλαγής (4/5)

- Transaction Commit
 - A sub-transaction commits
 - ETST -> Log
 - If compensating mode -> Update ST List
 - If split mode -> Delete ST entry
 - If no active ST for a JT, commit JT
 - ETJT -> Log
 - Update Transaction Status Table
 - User specifies the end of KT, change KT to Committing
 - If no active JT, commit KT
 - ETKT -> Log

Επεξεργασία συναλλαγής (5/5)

- Transaction Abort
 - A doubly linked list is maintained between base stations by transaction log. User provide head of the linked list.
 - Go forward the linked list to restart interrupted transaction. Go backward along the linked list to abort or compensate JTs.
 - What causes JT abort? LT and GT abort? KT abort? What's the relation between JT and LT, GT?

Άλλες λύσεις?

- Εάν αποθηκεύσουμε πληροφορία στο κινητό?
 - Τι χρειάζεται να αποθηκευτεί
 - Το Transaction ID
 - Τι άλλο χρειάζεται?
 - Μια υπηρεσία GDBS
 - Ποιο είναι το πλεονέκτημα σε σχέση με τις ΚΤ?
 - Management Complexity? Deployment Complexity? Scalability?
 - Λύνουν το handoff πρόβλημα

Κριτική

- Σχετικά περίπλοκη
- Πού βρίσκεται η υπηρεσία GDBS? Πώς λειτουργεί?
- Αφήνεται αρκετή πολυπλοκότητα στο GDBS
- Η διαδικασία Handoff (που είναι ζωτικής σημασίας) έχει πρόβλημα
 - Split Transaction prerequisites not guaranteed
 - Ο χρόνος Handoff δεν προσδιορίζεται

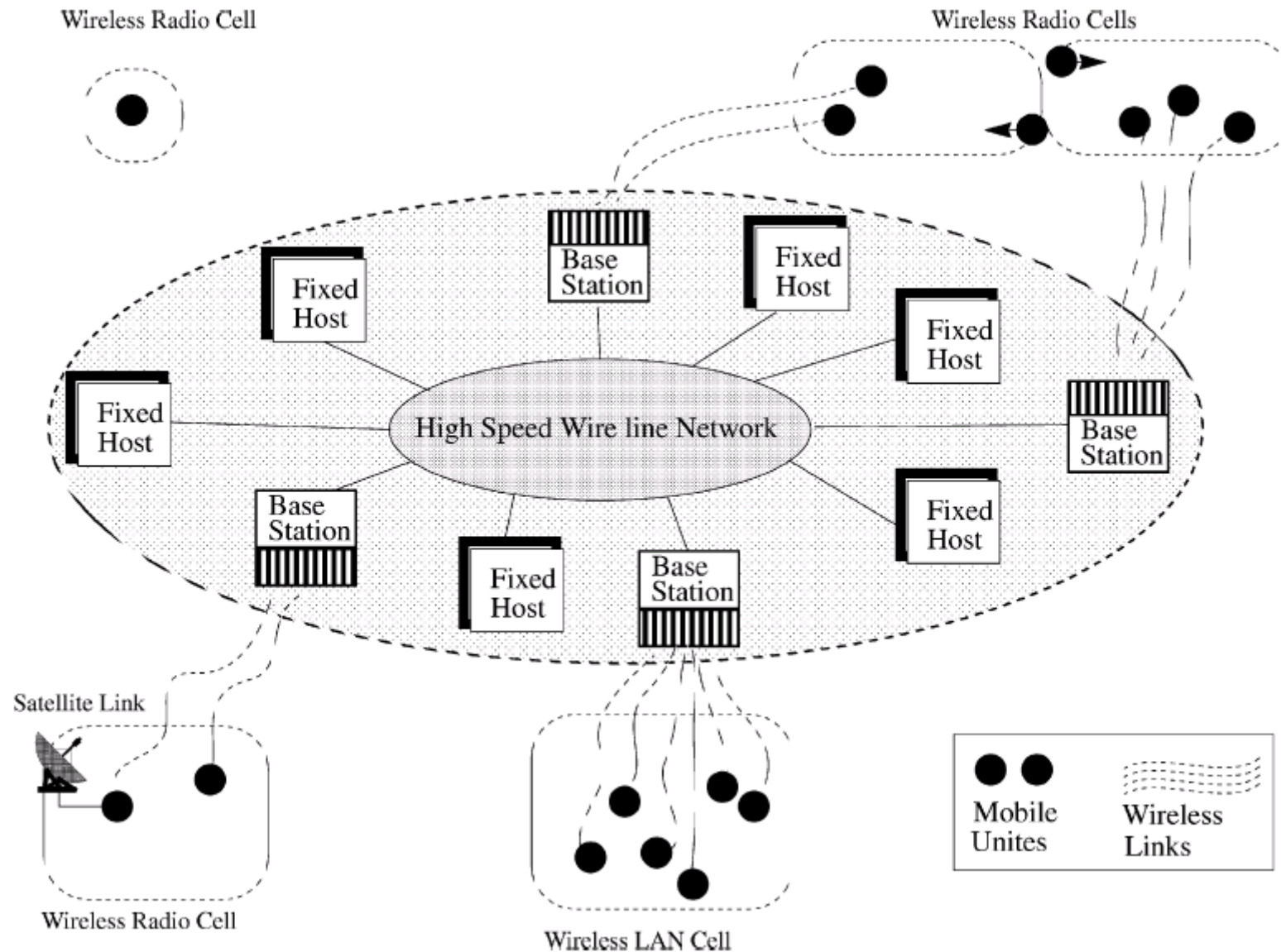
Συντομογραφίες

- GDBS: Global Database System
- BTKT: Begin Kangaroo Transaction
- HOKT: Handoff Kangaroo Transaction
- CTKT: Continuing Kangaroo Transaction
- BTJT: Begin Joey Transaction
- ETJT: Commit (End) Joey Transaction
- BTST: Begin Sub-Transaction
- ETST: End Sub-Transaction
- KTID: Kangaroo Transaction ID
- JTID: Joey Transaction ID

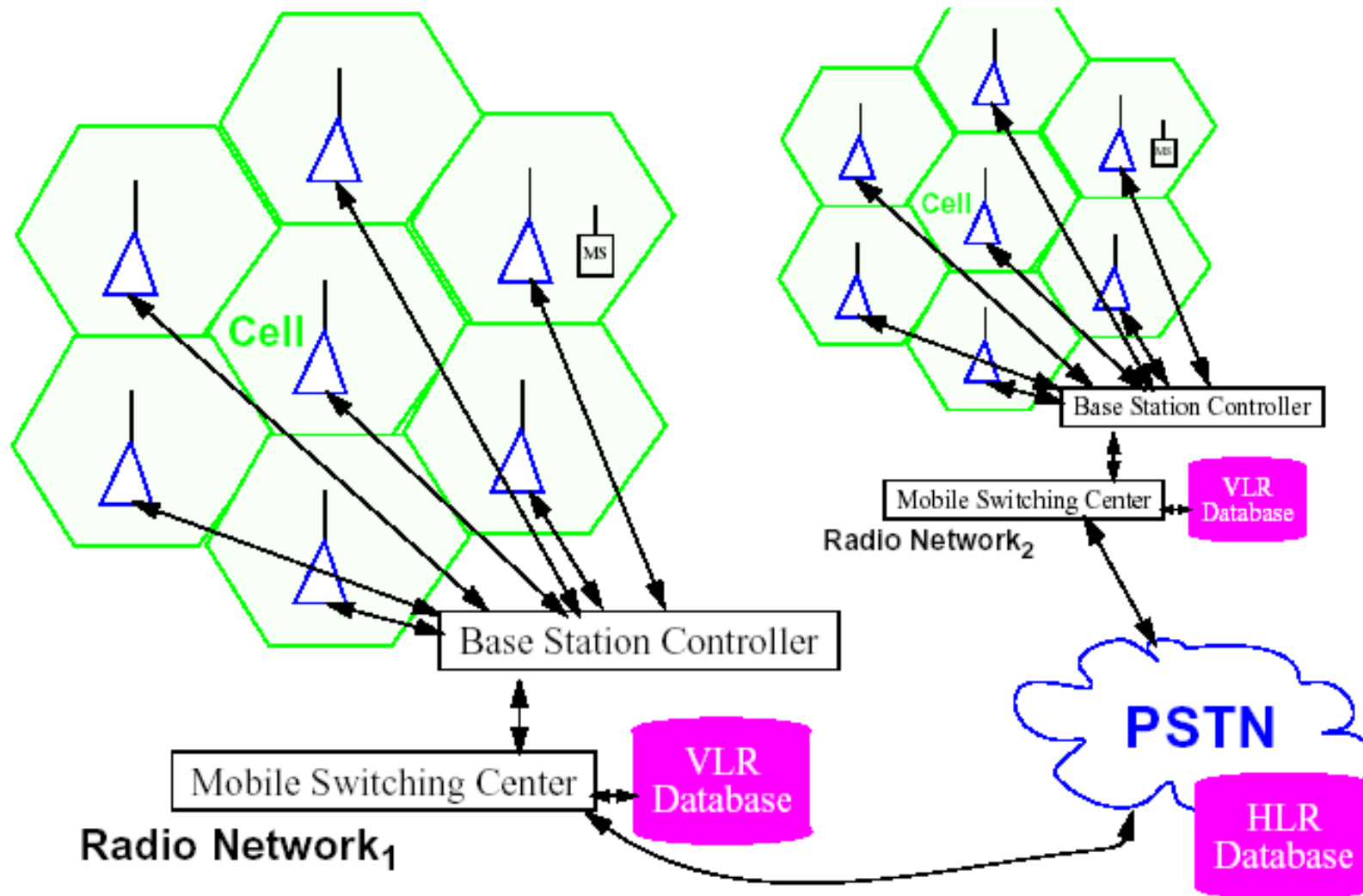
Περιεχόμενα

- Συναλλαγές
- **Αρχιτεκτονική δικτύου**
- Μοντέλα κινητικότητας
- Διαχείριση θέσης: Registration
- Διαχείριση θέσης: Paging

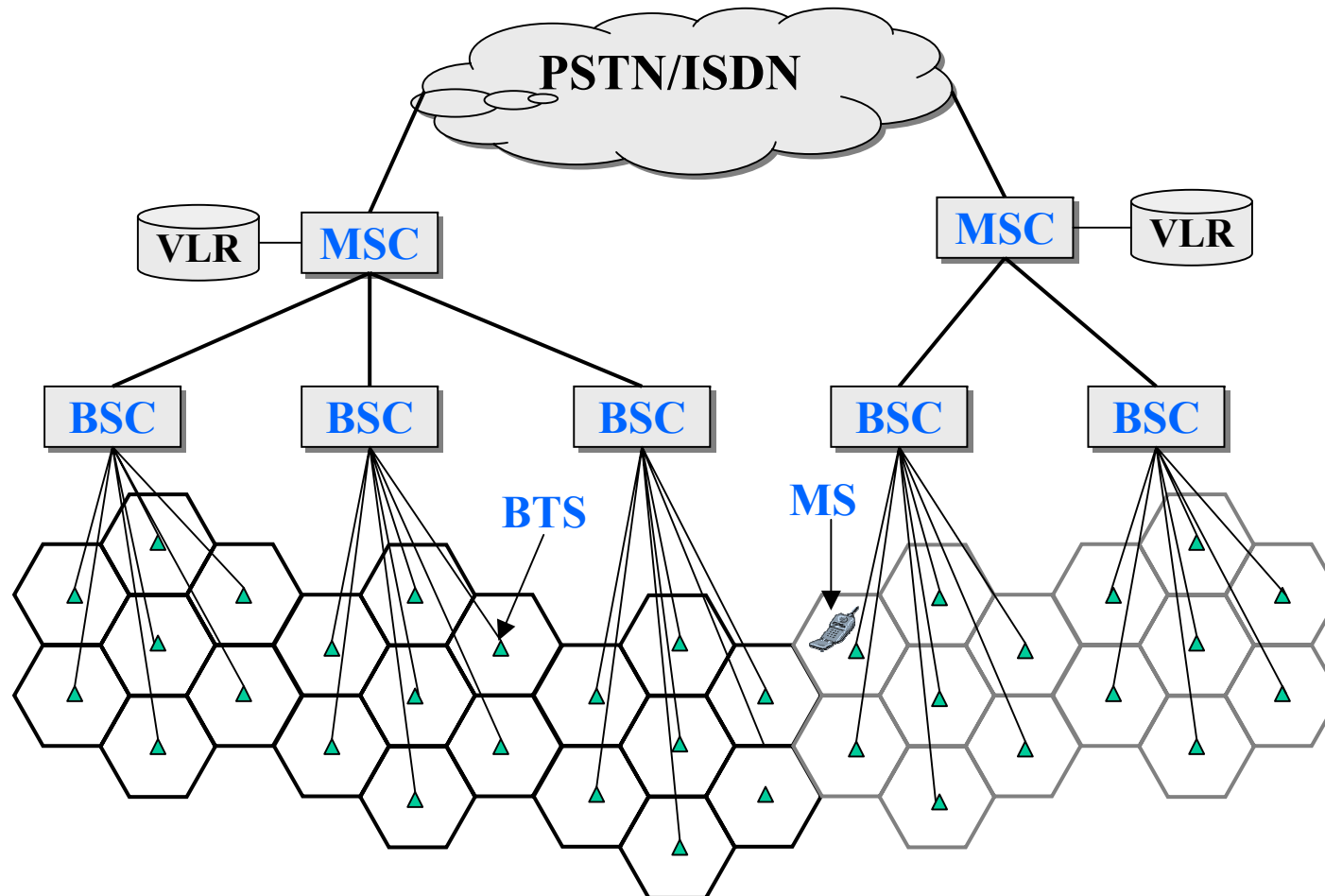
Αρχιτεκτονική κινητού δικτύου



Αρχιτ. Personal Comm. Sys. (PCS)



Αρχιτεκτονική 2G PCS



MS : Mobile Station

MSC : Mobile Services Switching Center

BTS : Base Transmitter Station VLR : Visitor Location Register

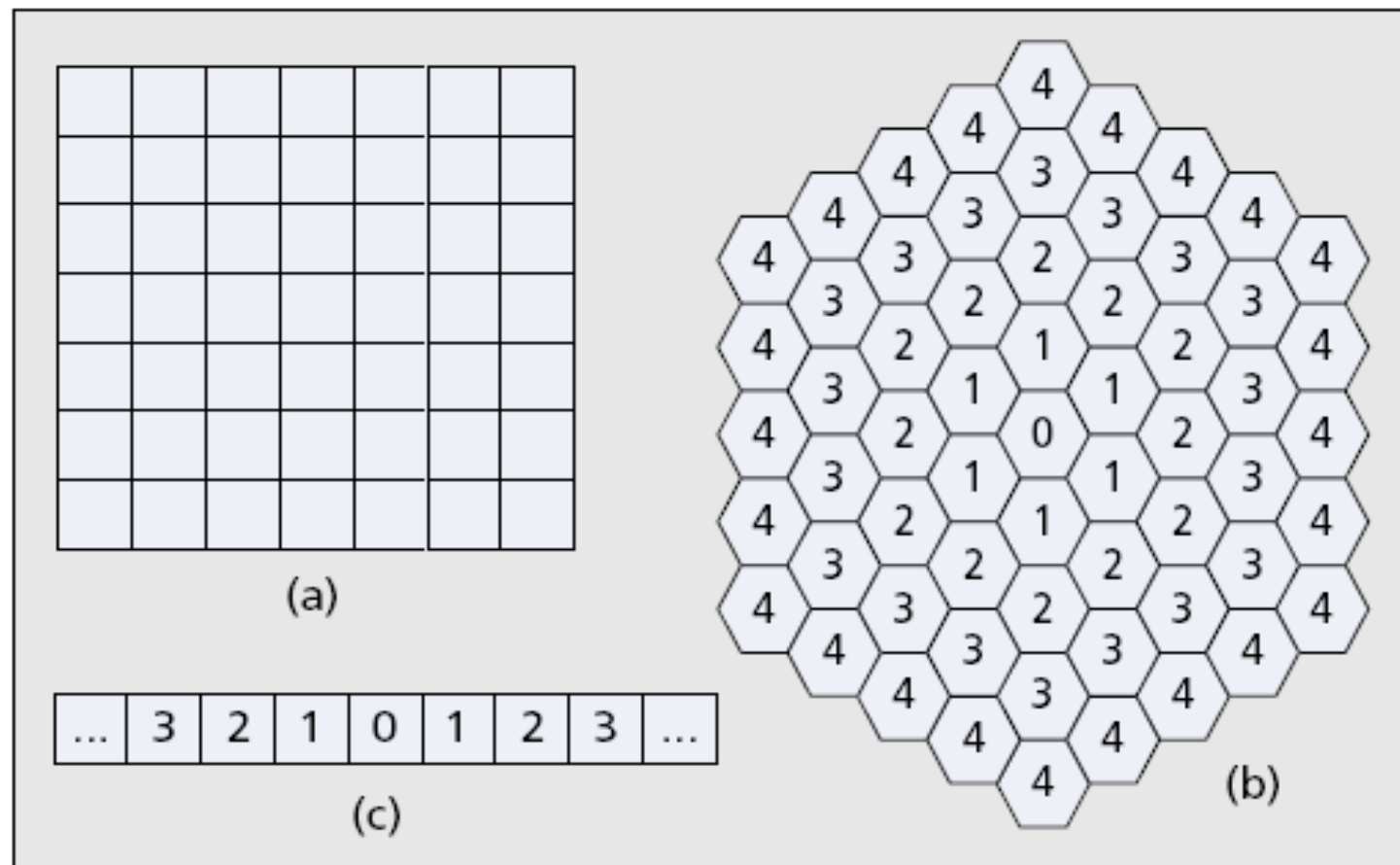
BSC : Base Station Controller

Περιεχόμενα

- Συναλλαγές
- Αρχιτεκτονική δικτύου
- **Μοντέλα κινητικότητας**
- Διαχείριση θέσης: Registration
- Διαχείριση θέσης: Paging

Τοπολογίες δικτύου (1/2)

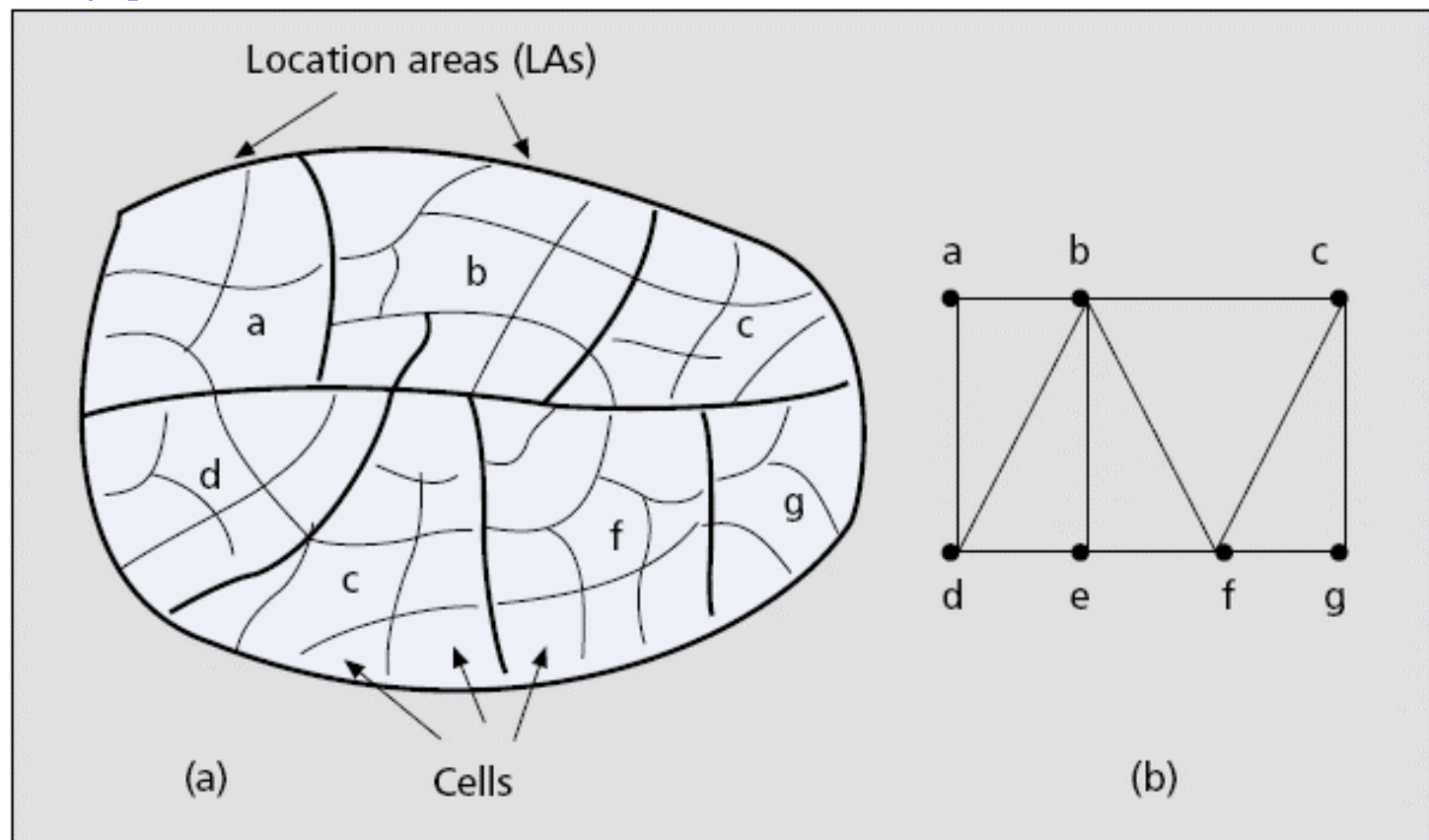
Regular / Γεωμετρικά



■ Figure 3. Regular cell topology: (a) mesh; (b) hexagonal; (c) linear.

Τοπολογίες δικτύου (2/2)

Irregular / Συμβολικά



■ Figure 1. A representation of an actual cellular network topology by a graph: a) the cell and LA topology in a cellular network; b) a graph model showing the interconnections of the LAs.

Μοντέλο Fluid Flow

- Η πιθανότητα στη θέση x στο χρόνο t ενός κινητού που ξεκινά από τη θέση x_0 στο χρόνο t_0 είναι:

$$P_x(x | x_0, t) = \frac{1}{\sqrt{2\pi D(t - t_0)}} \exp\left\{-\frac{[x - x_0 - v(t - t_0)]^2}{2D(t - t_0)}\right\}, \quad t \geq t_0,$$

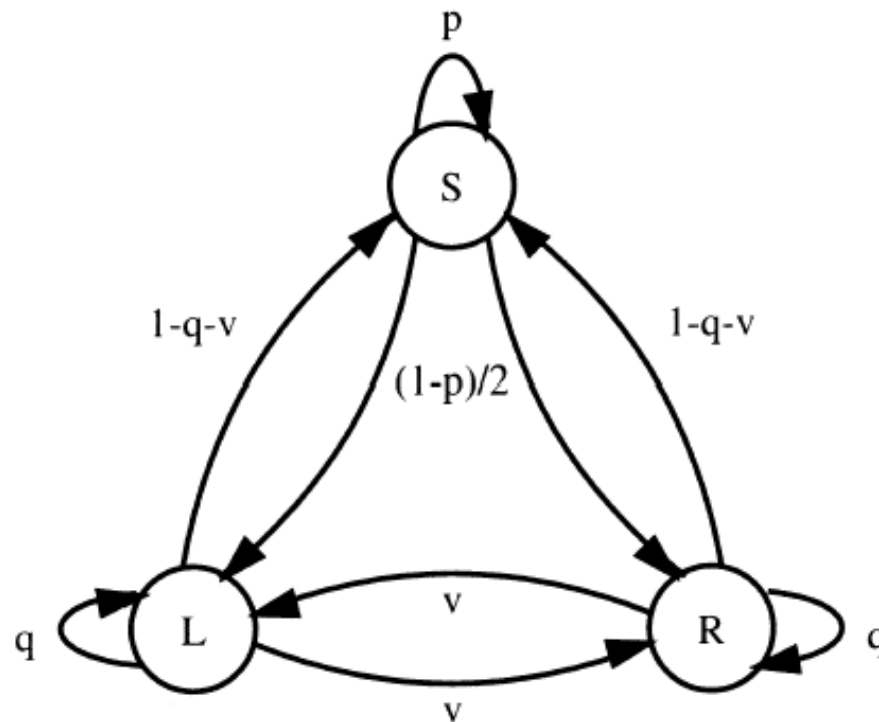
- όπου D η σταθερά διάχυσης ($\text{length}^2/\text{time}$), παράμετρος που αναπαριστά την επιτάχυνση της κίνησης, v είναι η drift ταχύτητα ($\text{length}/\text{time}$) που αναπαριστά τη μέση ταχύτητα του κινητού
- Υψηλή D και v σημαίνουν πολύ ενεργή κίνηση, ενώ χαμηλή D και v σημαίνουν μικρή αλλαγή στη θέση σε σχέση με αλλαγές στο χρόνο
- Στο μοντέλο αυτό, η κατεύθυνση δεν μοντελοποιείται
- Κατάλληλο για κίνηση οχημάτων

Μοντέλο Random Walk

- Μοντέλο διακριτού χρόνου
- Ο χρόνος διαιρείται σε slots
- Σε κάθε slot το κινητό επιλέγει τυχαία κατεύθυνση και την ακολουθεί
- Μια μόνο κίνηση σε κάθε slot
- Δεν έχει μνήμη, η επόμενη θέση εξαρτάται μόνο από την τωρινή
- Για μονοδιάστατη τοπολογία δικτύου
 - Με πιθανότητα p παραμένει στο ίδιο κελί
 - Με πιθανότητα $(1-p)/2$ μετακινείται σε γειτονικό κελί $i+1$ ή $i-1$.
- Για διδιάστατη (εξαγωνική) τοπολογία δικτύου
 - Με πιθανότητα p παραμένει στο ίδιο κελί
 - Με πιθανότητα $(1-p)/6$ μετακινείται σε κάποιο γειτονικό κελί.
- Κατάλληλο για picocells και microcells, δηλ., για πεζούς με συχνή αλλαγή κατεύθυνσης

Μοντέλο Markov Walk

- Έχει μνήμη, η τρέχουσα κίνηση εξαρτάται από προηγούμενη
- Προτάθηκε για μονοδιάστατη τοπολογία δικτύου



Μοντέλο Gauss-Markov

- Ικανό να περιγράψει τη συσχέτιση της ταχύτητας στο χρόνο
- Για κυψελοειδή και ad hoc δίκτυα
- Η συσχέτιση μεταξύ δυο διαδοχικών ταχυτήτων είναι:

$$v_k = \alpha v_{k-1} + (1 - \alpha)\mu + \sqrt{1 - \alpha^2} u_{k-1}$$

- Οι αντίστοιχες συντεταγμένες είναι:

$$x_k = x_{k-1} + v_{k-1} \cos \theta_{k-1}$$

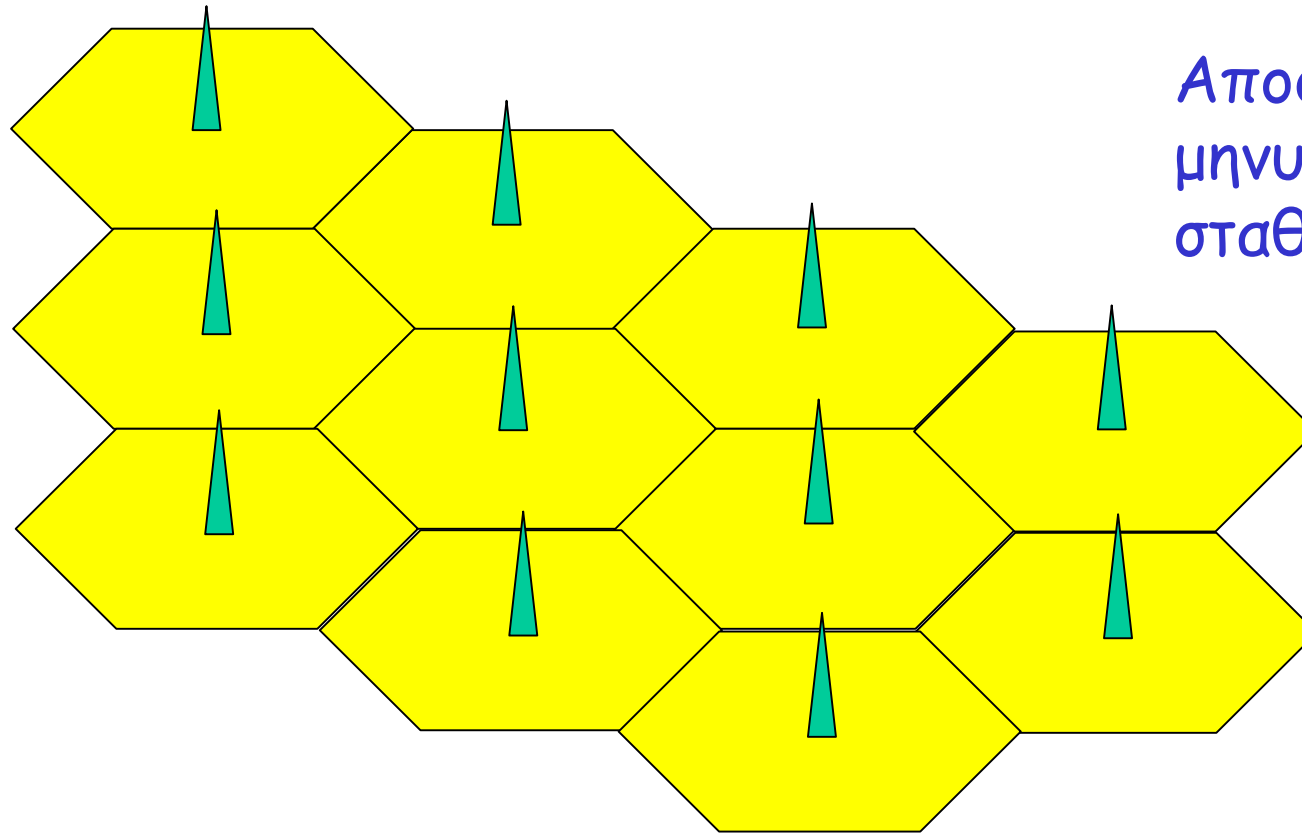
$$y_k = y_{k-1} + v_{k-1} \sin \theta_{k-1},$$

Περιεχόμενα

- Συναλλαγές
- Αρχιτεκτονική δικτύου
- Μοντέλα κινητικότητας
- **Διαχείριση θέσης: Registration**
- Διαχείριση θέσης: Paging

Το πρόβλημα της διαχείρισης θέσης

Πού είναι ο 97532468?



Αποστολή broadcast
μηνυμάτων σε κάθε
σταθμό βάσης?

Κύρια ζητήματα

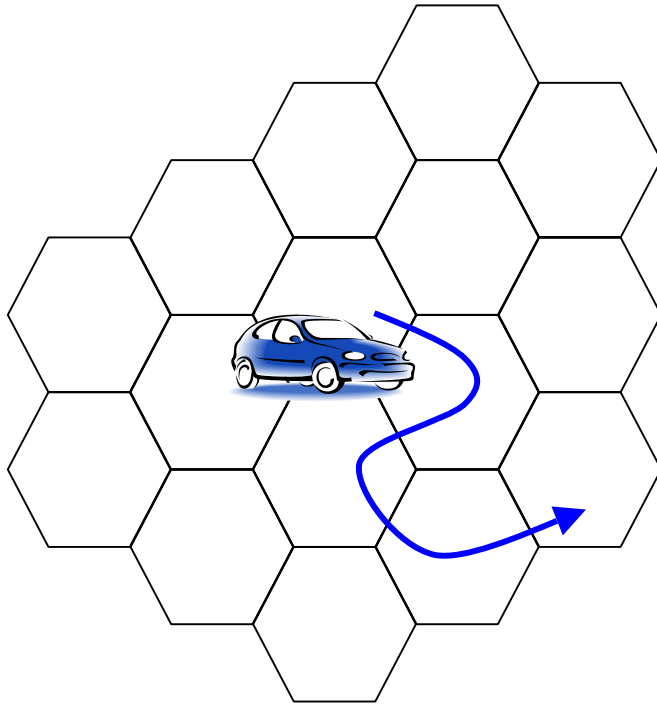
- Ενημέρωση θέσης (Location update)
 - Εκτελείται από τον κινητό πελάτη
 - Ενημερώνει το δίκτυο για την τρέχουσα θέση του
- Αναζήτηση (Paging)
 - Εκτελείται από το δίκτυο
 - Αναζήτηση σε όλα τα πιθανά κελιά μέχρι να βρεθεί
 - Ο αριθμός των κελιών, όπου θα αναζητηθεί, εξαρτάται από τη διαδικασία ενημέρωσης θέσης
- **Tradeoff-1:** Συχνή ενημέρωση θέσης vs. Λιγότερο συχνή ενημέρωση θέσης
 - Μικρή αβεβαιότητα θέσης (μικρό paging κόστος) & Μεγάλο κόστος (traffic) στο uplink κανάλι
 - Μεγάλη αβεβαιότητα θέσης (μεγάλο paging κόστος) & Μικρό κόστος (traffic) στο uplink κανάλι
- **Tradeoff-2:** Paging κόστος vs. Καθυστέρηση
 - Ταυτόχρονο paging όλων των κελιών => μικρή καθυστέρηση

Tradeoff-1

Never Update Σχήμα

Ποτέ ενημέρωση θέσης
(καθόλου κόστος).

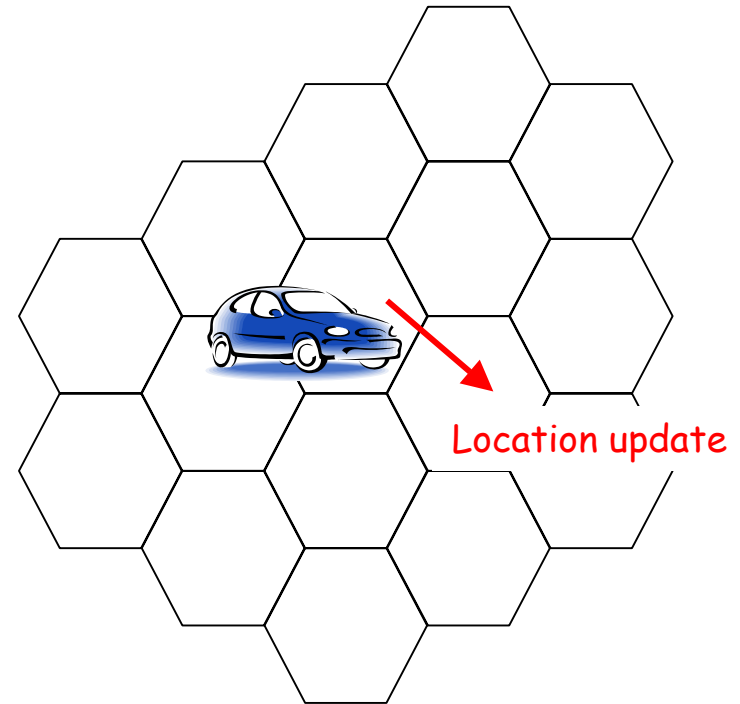
Ανάγκη αναζήτησης σε
κάθε κελί (μέγιστο κόστος).



Always Update Σχήμα

Ενημέρωση θέσης σε κάθε κελί
(μέγιστο κόστος).

Αναζήτηση σε ένα μόνο κελί
(ελάχιστο κόστος).

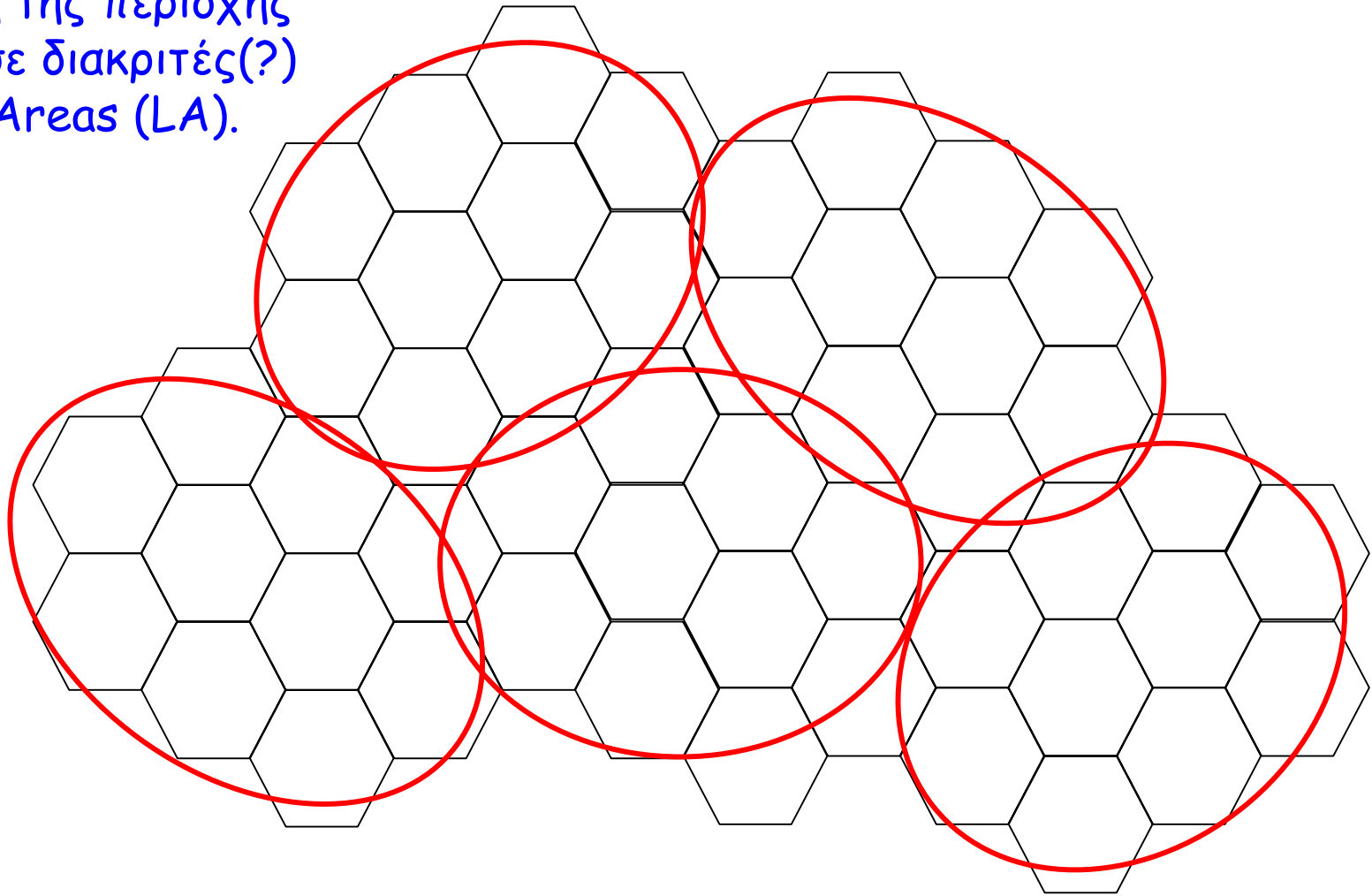


Κατηγοριοποίηση των σχημάτων ενημέρωσης θέσης

- **Καθολικά (Global):** όλοι οι κινητοί πελάτες ενημερώνουν για τη θέση τους στο ίδιο σύνολο κελιών
 - Π.χ., Location Areas, Reporting Centers
- **Τοπικά(Local) ή Individualized ή Per-User:** Κάθε κινητός πελάτης επιτρέπεται να αποφασίζει πότε και πού θα ενημερώσει για τη θέση του
 - Time-based, Movement-based, Distance-based
- **Στατικά (Static):** Το σύνολο των κελιών όπου πραγματοποιείται ενημέρωση θέση είναι προκαθορισμένο
 - Π.χ., Location Areas
- **Δυναμικά (Dynamic):** Η ενημέρωση θέσης πραγματοποιείται σε οποιοδήποτε κελί ανάλογα με την κινητικότητα του κινητού
 - Π.χ., Time-Varying Location Areas, LeZi-Update

Η τεχνική των Location Areas (1/2)

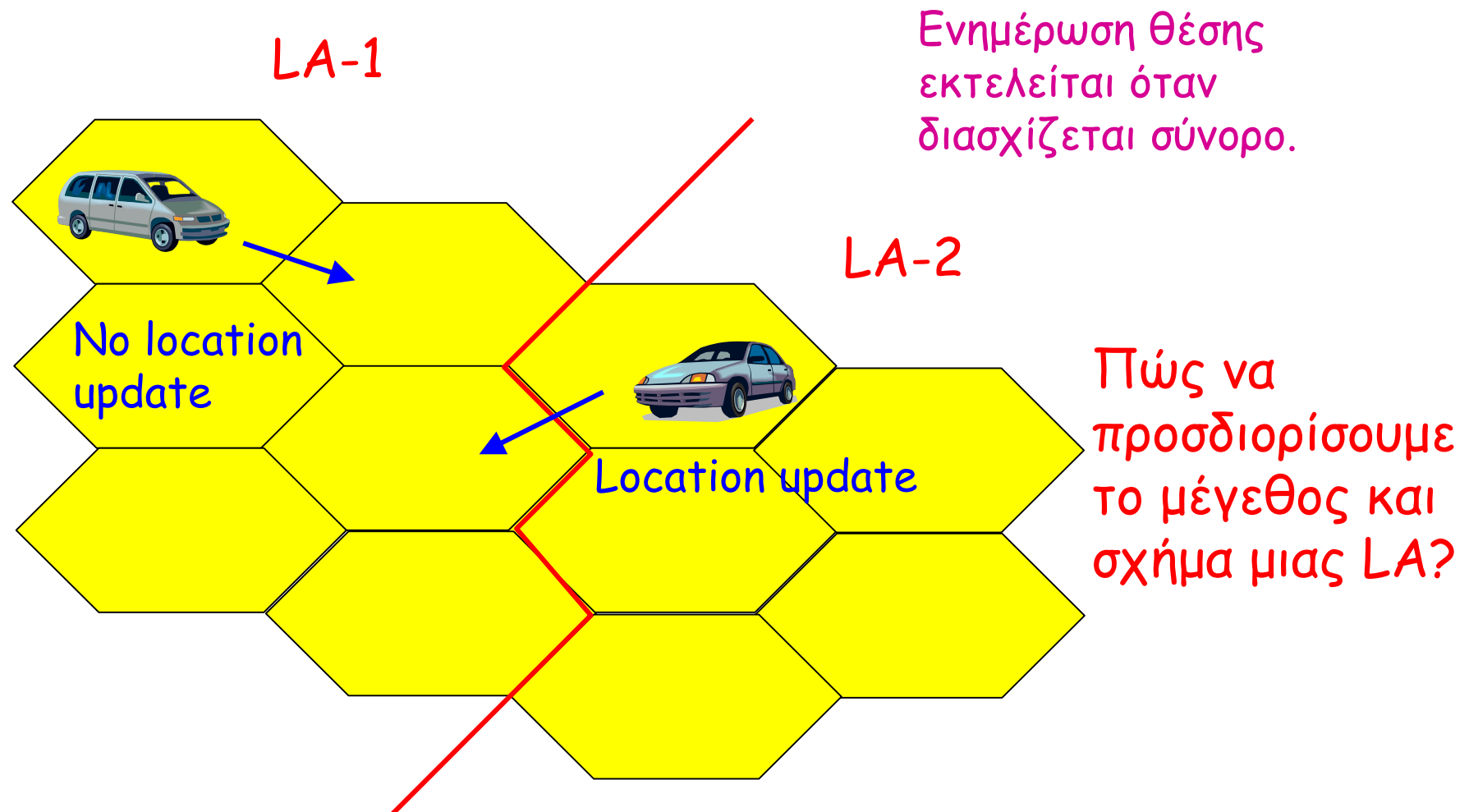
Διαμέριση της περιοχής
κάλυψης σε διακριτές(?)
Location Areas (LA).



Η τεχνική των Location Areas (2/2)

- Κάθε σταθμός βάσης εκπέμπει το ID της LA όπου ανήκει
- Το κινητό ενημερώνει για τη νέα θέση του (δηλ., το ID της νέας LA) οποτεδήποτε εισέρχεται σε νέα LA
- Για να βρεθεί το κινητό, το δίκτυο στέλνει paging μηνύματα σε όλα τα κελιά της LA, όπου αναφέρθηκε τελευταία η θέση του
- LAs: Καθολικό και στατικό σχήμα διαχείρισης θέσης
- Παραλλαγή => **Two Location Areas (TLA)**: το κινητό αποθηκεύει τις δυο τελευταίες LA που έχει επισκεφτεί. Αντικατάσταση της παλιότερης ή με μικρότερο χρόνο διαμονής, κ.τ.λ. Κατάλληλο για:
 - Μικρό Call-to-Mobility Ratio
 - Μεγάλο location update κόστος

Ενημέρωση θέσης



Location Databases

Αυτές οι δυο DB
επικοινωνούν μεταξύ
τους για να κάνουν
authentication και
ενημέρωση για την
θέση του κινητού.

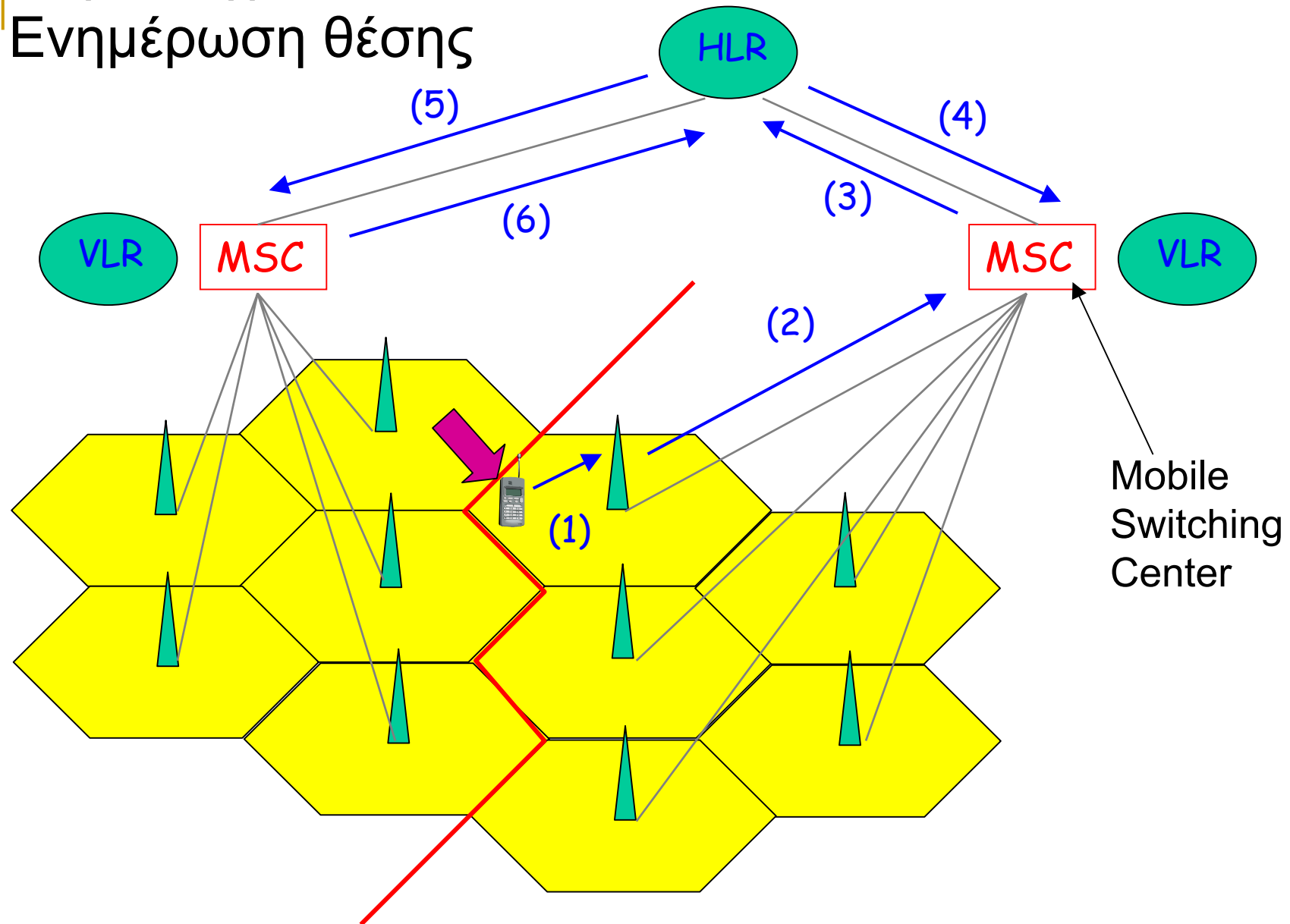
GSM Ορολογία:

Home Location
Register (HLR)

Visitor Location
Register (VLR)

- **Home Database (HLR)**
 - Κάθε κινητό σχετίζεται μόνιμα με μια home database
 - Κρατά το profile του κινητού
 - Mobile ID, authentication keys, θέση, χρέωση, κ.τ.λ.
- **Visiting Database (VLR)**
 - Κρατά τη θέση του κινητού στη δική της **service area**.

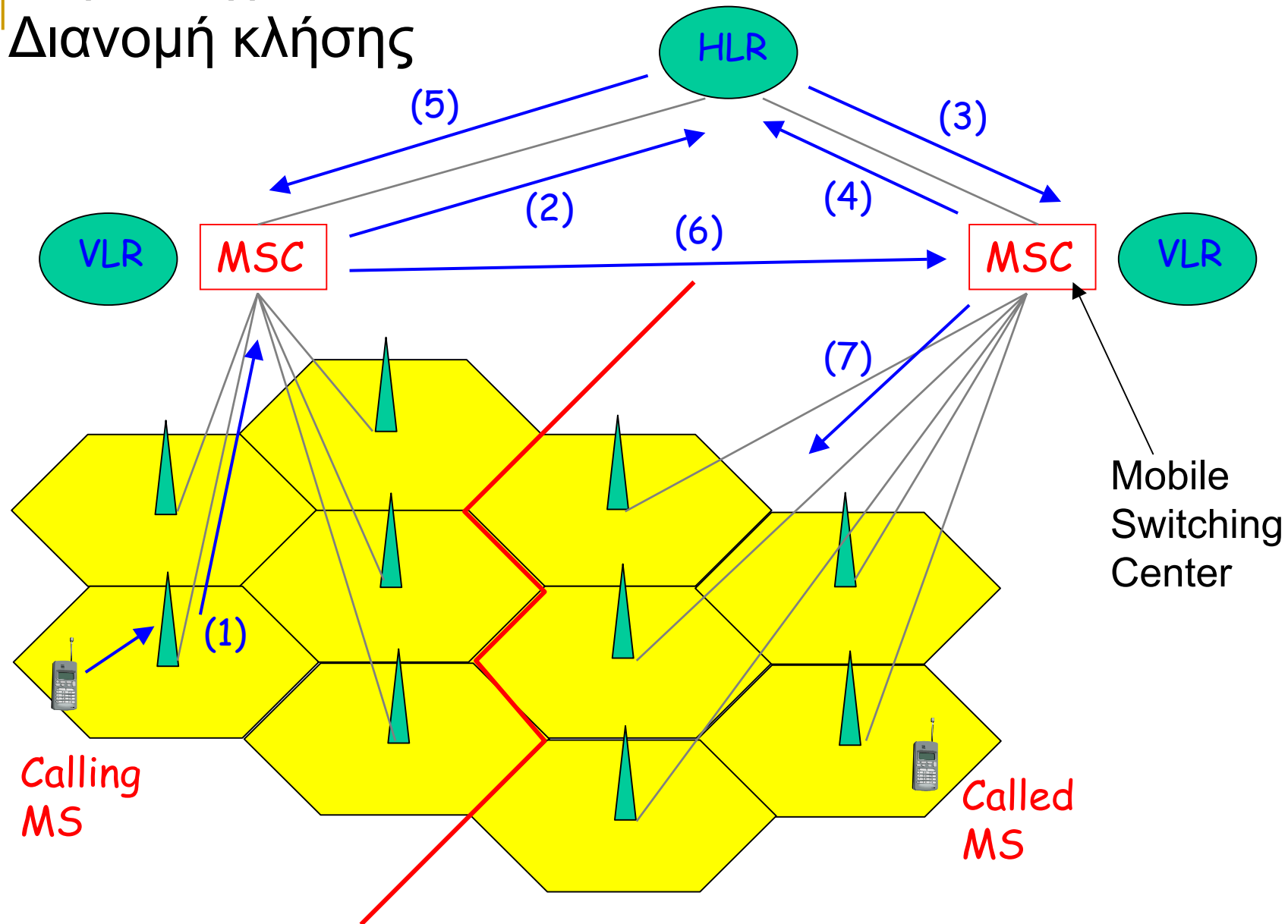
Παράδειγμα GSM: Ενημέρωση θέσης



Διαδικασία ενημέρωσης θέσης

1. Το κινητό μεταδίδει ένα μήνυμα Location Update (LU) στο νέο Σταθμό Βάσης (ΣΒ)
2. Ο ΣΒ προωθεί το σήμα LU στο MSC
3. Το MSC στέλνει LU στην HLR και ενημερώνει τη VLR
4. Η HLR κάνει τα εξής:
 - authenticates το κινητό
 - Καταγράφει το ID της νέας VLR
 - Στέλνει ACK στη νέα VLR
5. Η HLR στέλνει μήνυμα *registration cancellation* στην παλιά VLR
6. Η παλιά VLR διαγράφει την εγγραφή για το κινητό και επιστρέφει εάν ACK στη HLR

Παράδειγμα GSM: Διανομή κλήσης



Διαδικασία διανομής κλήσης

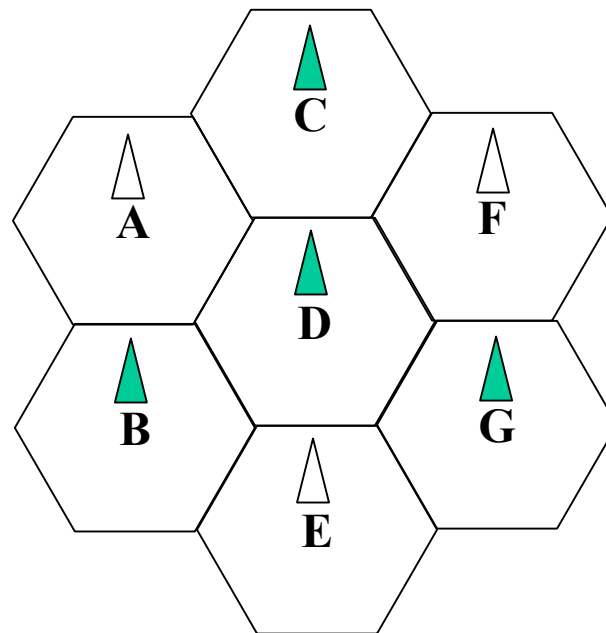
1. Το καλούν (calling) κινητό στέλνει μήνυμα *call initiation* στο MSC διαμέσου του BS
2. Το MSC στέλνει αίτηση location στην HLR του καλούμενου (called) κινητού
3. Η HLR προσδιορίζει την αντίστοιχη VLR του καλούμενου κινητού και στέλνει μήνυμα *route request* στο κινητό
4. Το MSC δεσμεύει προσωρινό ID στο ΣΒ και στέλνει αυτό το ID στη HLR
5. HLR forwards the ID to MSC of the calling MS
6. Το καλούν MSC αιτήται *call set up* στο καλούμενο MSC
7. Paging μηνύματα στέλνονται σε όλα τα κελιά της συγκεκριμένης LA

Η τεχνική των Reporting Cells (1/2)

- Ένα υποσύνολο των κελιών έχουν επιλεγεί ως reporting cells ή reporting centers (RC)
- Ο σταθμός βάσης κάθε τέτοιου κελιού εκπέμπει σήμα για να σηματοδοτήσει ότι το κελί είναι RC
- Η **γειτονιά** ενός RC i είναι τα κελιά, που δεν είναι RC, και είναι προσβάσιμα από το i χωρίς να περάσουμε από ένα άλλο RC
- Το κινητό ενημερώνει για τη θέση του οποτεδήποτε περάσει από εάν RC
- Η τεχνική RC είναι καθολική και στατική
- Έχει δυο ακραίες περιπτώσεις
 - όλα τα κελιά είναι RCs, και
 - κανένα κελί δεν είναι RC.

Η τεχνική των Reporting Cells (2/2)

- Η γειτονιά του C αποτελείται από τα κελιά: A, C και F.
- Ένα κινητό που ακολουθεί τη διαδρομή $B \rightarrow A \rightarrow C$, θα ενημερώσει για τη θέση του όταν φτάσει στο A.
- Το σύστημα στέλνει paging μηνύματα στη γειτονιά του RC όπου εμφανίστηκε τελευταία το κινητό.



Σχήμα Time-based (2/2)

- Είναι δυναμικό σχήμα
- Εάν το κατώφλι προσδιοριστεί ανά κινητό, τότε είναι και τοπικό σχήμα, επίσης
- Δεν λαμβάνει υπόψη του την κινητικότητα
- Πλεονέκτημα
 - Απλό
- Μειονέκτημα
 - Πολύ άσχημη επίδοση στη χειρότερη περίπτωση

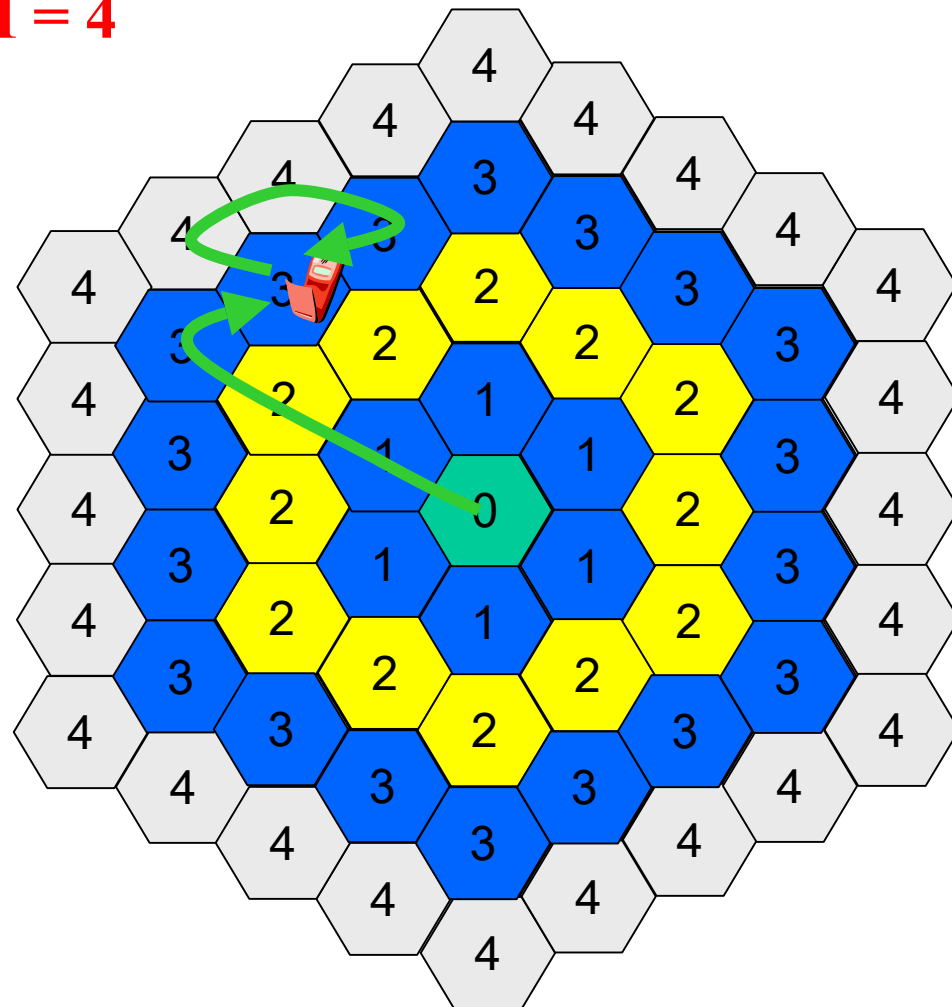
Σχήμα Movement-based

- Το κινητό διατηρεί έναν μετρητή που αρχικοποιείται στο 0
- Κάθε φορά που περνάει το σύνορο δυο κελιών, αυξάνει το μετρητή κατά ένα
- Εάν ο μετρητής γίνει ίσος με M (προκαθορισμένη σταθερά), το κινητό ενημερώνει για τη θέση του
- Συνεπώς, υπάρχει εγγύηση ότι το κινητό βρίσκεται σε δακτύλιο ακτίνας $M+1$, με κέντρο το τελευταίο κελί όπου ενημέρωσε για τη θέση του
- Το κατώφλι μπορεί να τεθεί διαφορετικό για κάθε κινητό
- Το σχήμα αυτό είναι δυναμικό και τοπικό

Παράδειγμα Movement-based

Κατώφλι: $M = 4$

Enhancement: Όταν ξαναφτάσει στο κελί, όπου έκανε την τελευταία ενημέρωση θέσης, να μην εκτελέσει ξανά ενημέρωση, δηλ., να θέσει τον μετρητή στην τιμή 0.

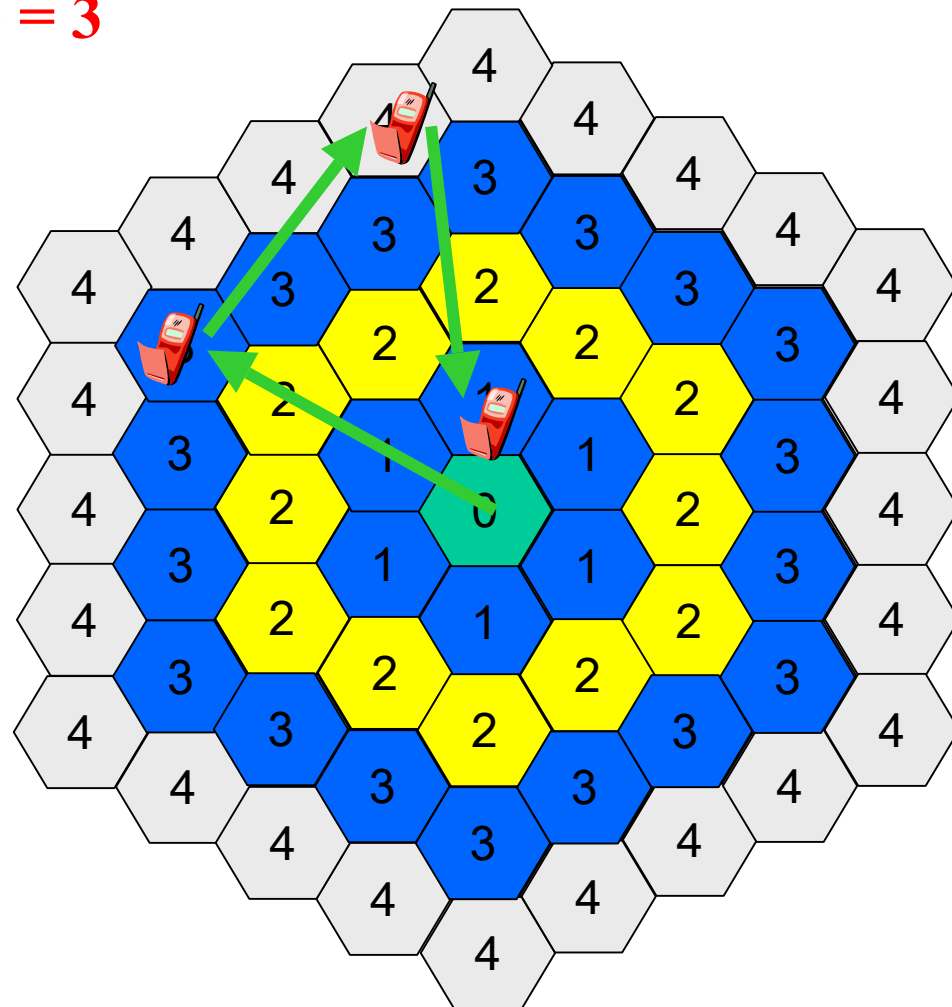


Σχήμα Distance-based (1/2)

- Το κινητό διατηρεί μια μεταβλητή που κρατά την απόσταση που έχει διανύσει
- Η απόσταση μετريέται σε αριθμό κελιών
- Εάν η διανυσθείσα απόσταση ξεπεράσει ένα κατώφλι D , το κινητό ενημερώνει για τη θέση του
- Συνεπώς, υπάρχει εγγύηση ότι το κινητό βρίσκεται σε απόσταση D από το κελί, όπου ενημέρωσε για τελευταία φορά για τη θέση του
- Το κατώφλι μπορεί να τεθεί διαφορετικό για κάθε κινητό
- Το σχήμα αυτό είναι δυναμικό και τοπικό
- Μειονεκτήματα
 - Είναι δύσκολο να υπολογίζουμε την απόσταση μεταξύ δυο κελιών
 - Δεν είναι εφικτό να αποθηκεύσει το κινητό όλα τα ζεύγη αποστάσεων

Σχήμα Distance-based (2/2)

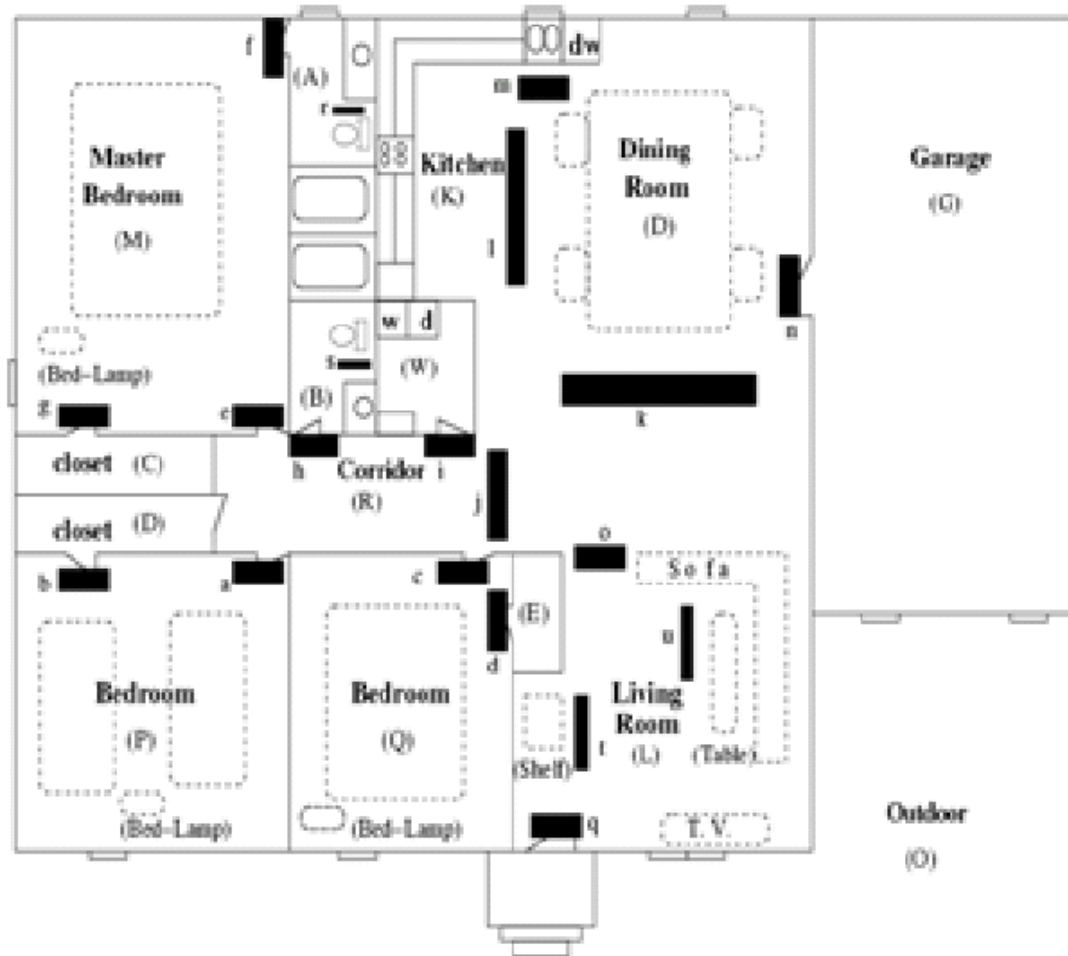
Κατώφλι: $D = 3$



Η τεχνική LeZi-Update (1/11)

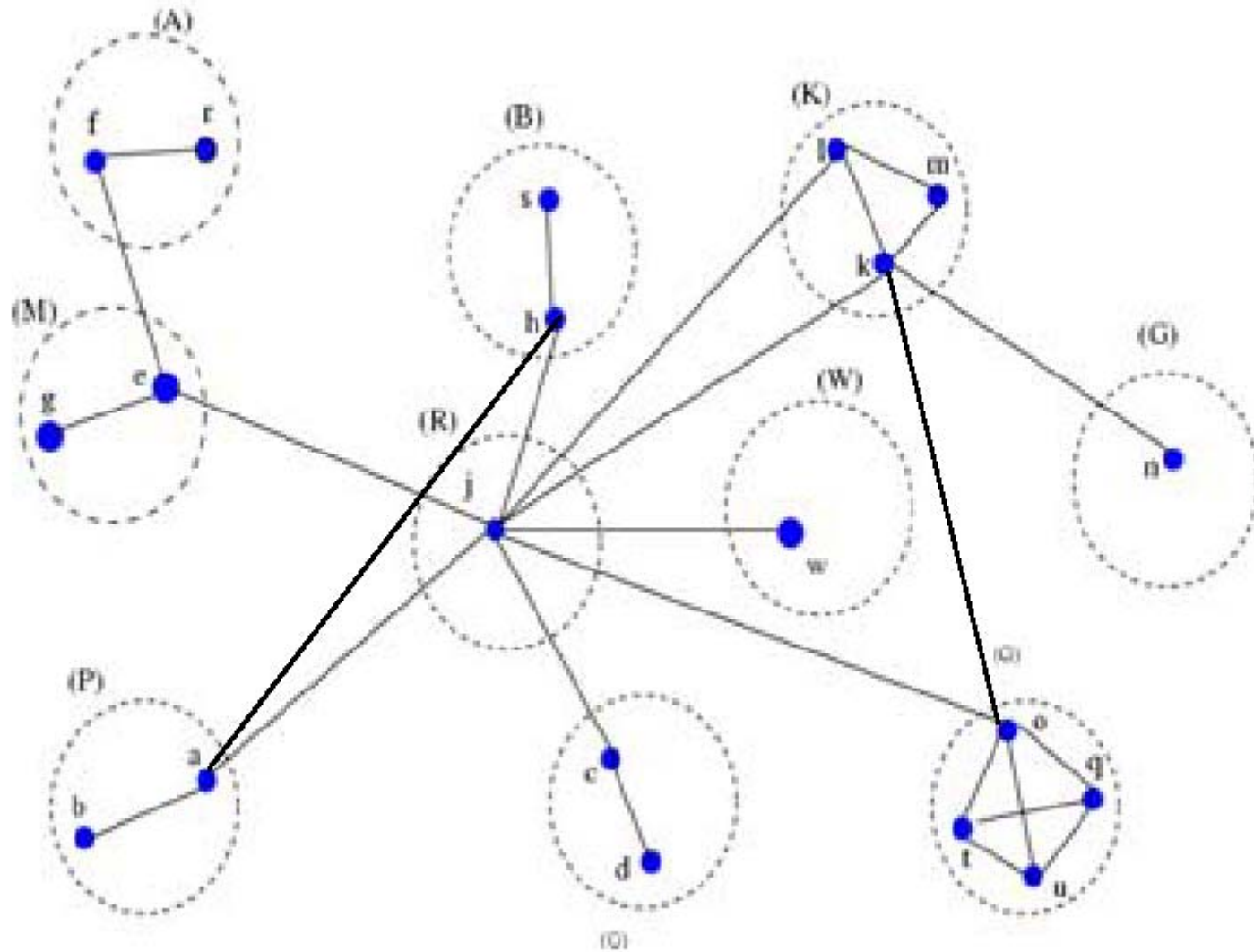
- Μοντέλα τοπολογίας δικτύου
 - γεωμετρικά μοντέλα
 - συμβολικά μοντέλα
- Τα γεωμετρικά μοντέλα
 - Εύκολα
 - Γενικά
 - Επιτρέπουν πρόβλεψη

Η τεχνική LeZi-Update (2/11)



Κίνηση: *a j k k o o j h h a a j k k o o j a a j k k o o j a a j k k ...*

Η τεχνική LeZi-Update (3/11)



Η τεχνική LeZi-Update (4/11)

- Φράση:

a j k k o o j h h a a j k k o o j a a j k k o o j a a j k k ...

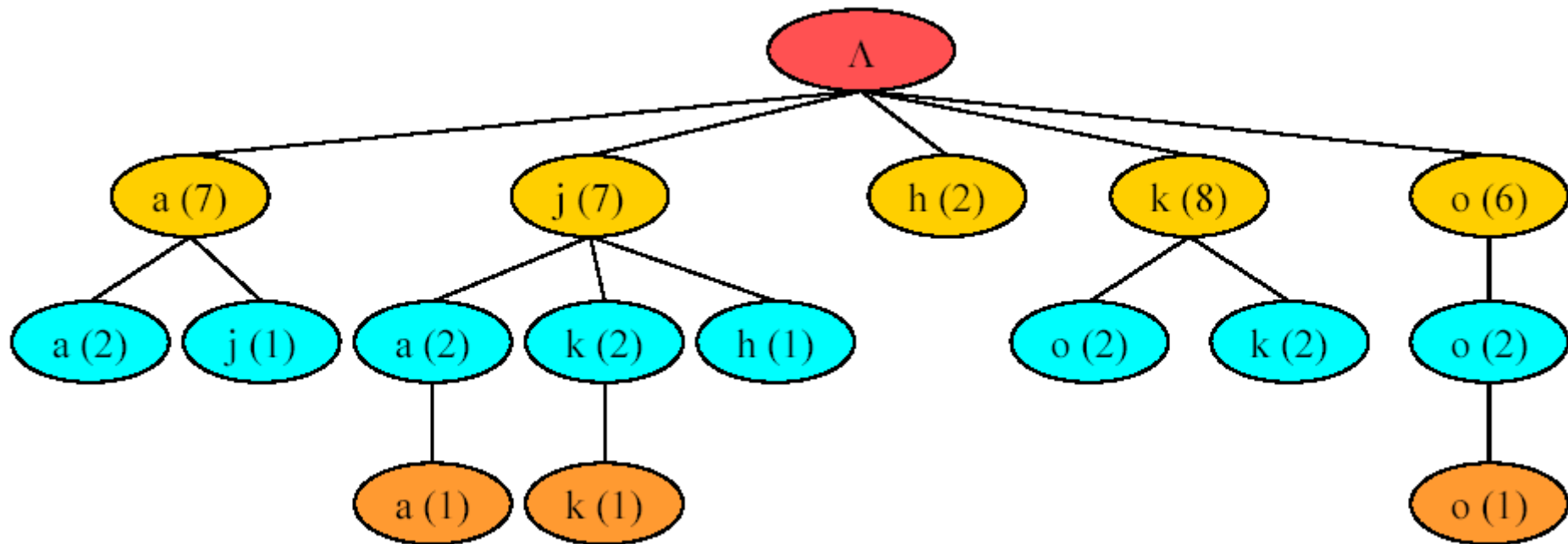
- Σταδιακό parsing της κίνησης

a, j, k, ko, o, jh, h, aa, jk, koo, ja, aj, kk, oo, jaa, jkk, ...

- Πιθανά “συμφραζόμενα” (context)

jk (**order-2**), *j* (**order-1**), *Λ* (**order-0**)

Η τεχνική LeZi-Update (5/11)



jk (<i>order-2</i>)	j (<i>order-1</i>)	Λ (<i>order-0</i>)
$k jk$ (1)	$a j$ (1)	$a(4)$ $aa(2)$ $aj(1)$
Λjk (1)	$aa j$ (1)	$j(2)$ $ja(1)$ $jaa(1)$
	$k j$ (1)	$jk(1)$ $jh(1)$ $k(4)$
	$kk j$ (1)	$ko(1)$ $koo(1)$ $kk(2)$
	$h j$ (1)	$o(4)$ $oo(2)$ $h(2)$
	Λj (2)	$\Lambda(1)$

➤ Probability of *jaa*:

- Absence in *order-2* and *order-1*;
escape probability in each order: $\frac{1}{2}$
- Probability of *jaa* in *order-0*: $\frac{1}{30}$

➤ Combined probability of phrase *jaa* :

$$(\frac{1}{2}) (\frac{1}{2}) (\frac{1}{30}) = 0.0048$$

Η τεχνική LeZi-Update (6/11)

jk (order-2)	j (order-1)	Λ (order-0)
$k jk$ (1)	$a j$ (1)	$a(4)$ $aa(2)$ $aj(1)$
Λjk (1)	$aa j$ (1)	$j(2)$ $ja(1)$ $jaa(1)$
	$k j$ (1)	$jk(1)$ $jh(1)$ $k(4)$
	$kk j$ (1)	$ko(1)$ $koo(1)$ $kk(2)$
	$h j$ (1)	$o(4)$ $oo(2)$ $h(2)$
	Λj (2)	$\Lambda(1)$

➤ Probability of k

- $\frac{1}{2}$ at the context of order-2
- Escaping into next lower order (order-1) with probability: $\frac{1}{2}$
- Probability of k at the order-1 (context of “ kk ”): $1/(1+1) = \frac{1}{2}$
- Probability of escape from order-1 to lowest order (order-0): $\frac{1}{2}$
- Probability of k at order-0 (context of Λ): $4/30$
- Combined probability of phrase $k = \frac{1}{2} + \frac{1}{2} \{ \frac{1}{2} + \frac{1}{2} (4/30) \} = 0.509$

Η τεχνική LeZi-Update (7/11)

Phrase	Probability		Phrase	Probability
<i>k</i>	0.5905		<i>ja</i>	0.0048
<i>kk</i>	0.0809		<i>jaa</i>	0.0048
<i>ko</i>	0.0048		<i>jk</i>	0.0048
<i>koo</i>	0.0048		<i>jh</i>	0.0048
<i>o</i>	0.0195		<i>a</i>	0.0905
<i>oo</i>	0.0095		<i>aa</i>	0.0809
<i>h</i>	0.0809		<i>aj</i>	0.0048
<i>j</i>	0.0095			

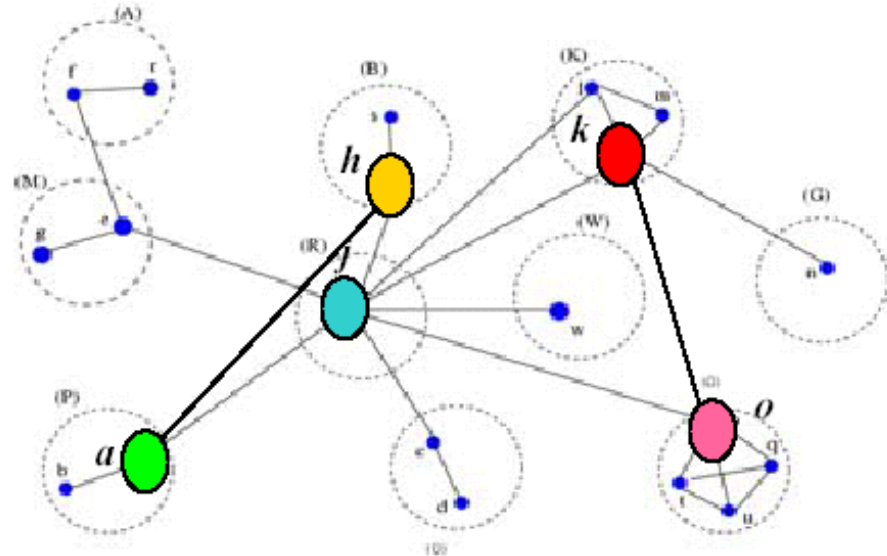
Probabilities of individual locations can be estimated by *dividing* the phrase probabilities into their constituent symbols according to *symbol-frequency* and *adding up* all such frequencies for a particular symbol (location)

Total probability for location ***k*** is:

$$0.5905 + 0.0809 + 0.0048/2 + 0.0048/3 = 0.6754$$

Η τεχνική LeZi-Update (8/11)

Location	Probability
<i>k</i>	0.6754
<i>a</i>	0.1794
<i>h</i>	0.0833
<i>o</i>	0.0346
<i>j</i>	0.0207



- Bob's movement profile: *ajkkoojhhaajkkoojaajkkoojaajkk*
- Phrases: *a, j, k, ko, o, jh, h, aa, jk, koo, ja, aj, kk, oo, jaa, jkk, ...*
- Probabilistic prediction of locations (symbols) based on their *ranking*
 - Prime *Advantages* of *Lempel-Ziv* type compression – most likely location is predicted
 - Prediction starts from *k* and proceeds along *a, h, o* and *j*

Η τεχνική LeZi-Update (9/11)

- **Movement history:** A string “ $v_1 v_2 v_3 \dots$ ” of symbols from alphabet $\mathcal{G}$
- **Inhabitant mobility model:** $\mathcal{V} = \{V_i\}$, a (piece-wise) stationary, ergodic stochastic process where V_i assumes values $v_i \in \mathcal{G}$
- **Stationarity:** $\{V_i\}$ is stationary if any of its subsequence is invariant with respect to shifts in time-axis

$$\Pr[V_1 = v_1, V_2 = v_2, \dots, V_n = v_n] = \Pr[V_{1+l} = v_1, V_{2+l} = v_2, \dots, V_{n+l} = v_n]$$

- Essentially the movement history “ $v_1, v_2, \dots, v_n$ ” reaches the system as $C(w_1), C(w_2), \dots, C(w_n)$ where w_i s are non-overlapping segments of history v_i and $C(w_i)$'s are their encoded forms
- *Minimizes $H(X)$ and asymptotically outperforms any finite-order Markov model*

$$H(X) = -p(x) \log p(x)$$

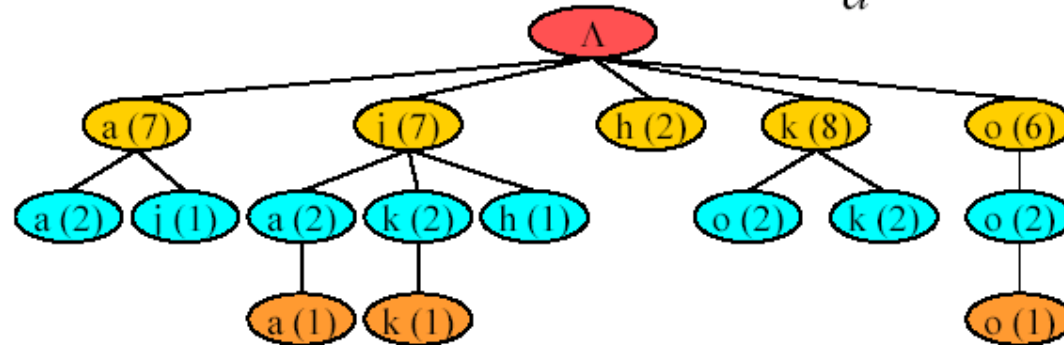
- The number of phrases is bounded by the relation: $c(n) = O\left(\frac{n}{\log n - \log \log n}\right)$

Η τεχνική LeZi-Update (10/11)

➤ Bob's movement profile: *a j k k o o j h h a a j k k o o j a a j k k o o j a a j k k ...*

➤ For a particular depth d of an LZ trie, let $H(V_i)$ represent entropy at i^{th} level.

Running-average of overall entropy is: $H(V) = \frac{1}{d} \sum_{i=1}^d H(V_i | V_1, V_2, \dots, V_{i-1})$



$$H(V_1) = \frac{7}{30} \lg\left(\frac{30}{7}\right) + \frac{7}{30} \lg\left(\frac{30}{7}\right) + \frac{2}{30} \lg\left(\frac{30}{2}\right) + \frac{8}{30} \lg\left(\frac{30}{8}\right) + \frac{6}{30} \lg\left(\frac{30}{6}\right) \approx 2.5795$$

$$H(V_2 | V_1) = \frac{7}{30} \left[\frac{2}{3} \log \frac{3}{2} + \frac{1}{3} \log 3 \right] + \frac{7}{30} \left[\frac{2}{5} \log \frac{5}{2} + \frac{2}{5} \log \frac{5}{2} + \frac{1}{5} \log 5 \right] + \frac{8}{30} \left[\frac{2}{4} \log \frac{4}{2} + \frac{2}{4} \log \frac{4}{2} \right] \\ \approx 0.9361$$

$$H(V) = \frac{H(V_1) + H(V_2 | V_1)}{2} \approx 1.789$$

Η τεχνική LeZi-Update (11/11)

➤ **Encoder:** Collects symbols and stores in the dictionary in a compressed form

Decoder: Decodes the encoded symbols and update phrase frequencies

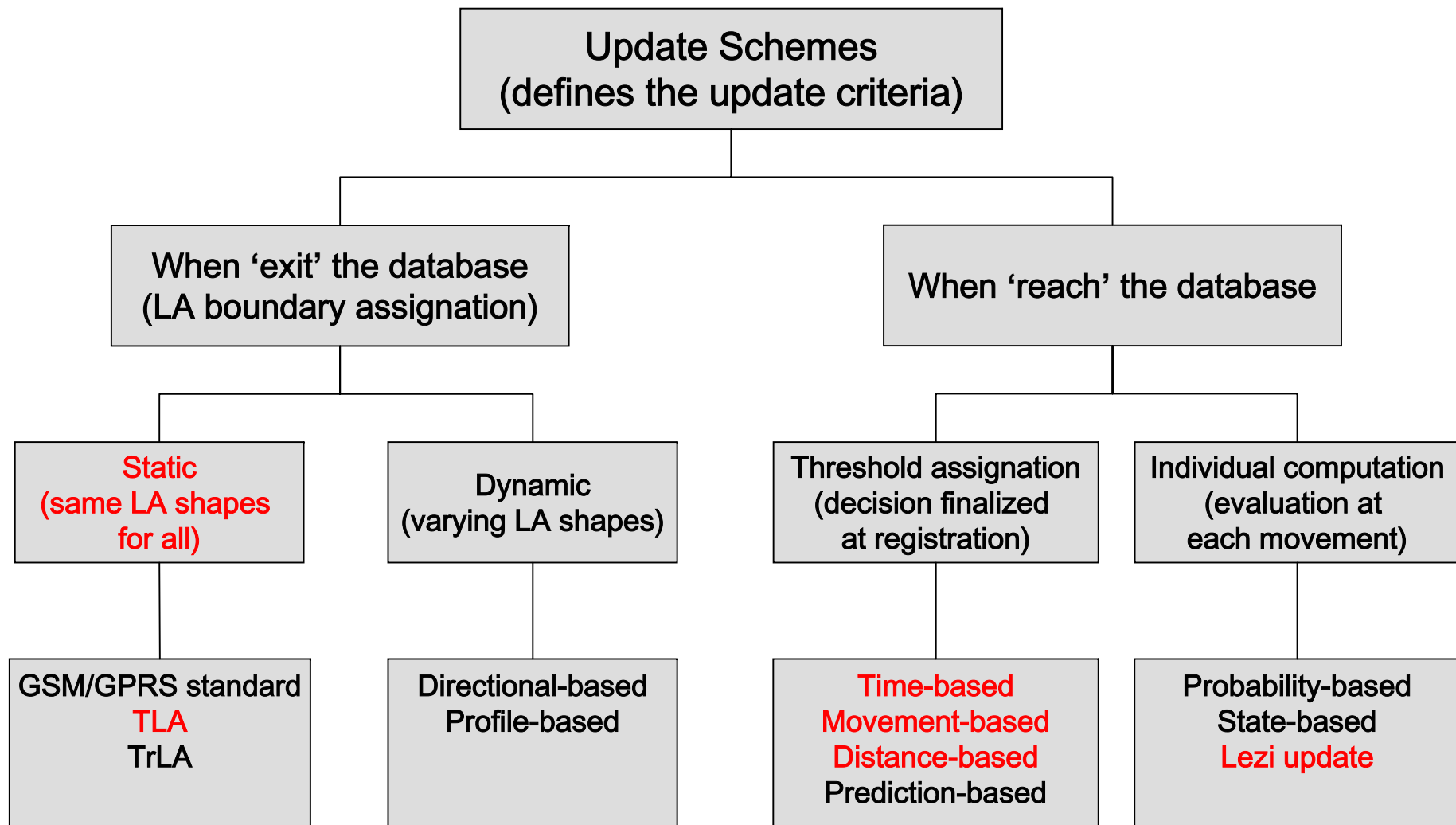
Encoder

```
Init dictionary, phrase w
loop
  wait for next symbol v
  if (w.v in dictionary)
    w := w.v
  else
    encode <index(w), v>
    add w.v to dictionary
    w := null
forever
```

Decoder

```
Initialize dictionary := empty
loop
  wait for next codeword<i, s>
  decode phrase := dictionary[i].s
  add phrase to dictionary
  increment frequency of every prefix
    of every suffix of phrase
forever
```

Κατηγοριοποίηση τεχνικών



Περιεχόμενα

- Συναλλαγές
- Αρχιτεκτονική δικτύου
- Μοντέλα κινητικότητας
- Διαχείριση θέσης: Registration
- **Διαχείριση θέσης: Paging**

Σημαντικά ζητήματα στο paging

- Δεδομένης μιας κατανομής πιθανότητας για την παρουσία του κινητού σε διάφορα “κελιά”
- Ποια είναι η ελάχιστη μέση προσπάθεια (αριθμός αναζητούμενων τοποθεσιών) μέχρι να βρεθεί το κινητό;

Πρόβλημα paging (1/3)

- Απαριθμούμε τις paging locations με $1, 2, \dots$, έτσι ώστε το κινητό είναι στη θέση i με πιθανότητα p_i
- Συσχετίζουμε τη θέση του κινητού με τυχαία μεταβλητή X , τέτοια ώστε $P\{X = i\} = p_i$
- Χωρίς βλάβη της γενικότητας, η location area n αποτελείται από υποπεριοχές (που τις συμβολίζουμε με A_n), που όλες θα γίνουν paged ταυτόχρονα.
- Η cardinality του A_n συμβολίζεται με k_n
- Η στρατηγική paging A είναι μια διατεταγμένη ακολουθία $(A_1, A_2, \dots)$ από location areas που θα γίνουν paged

Πρόβλημα paging (2/3)

- Η πιθανότητα ότι το κινητό είναι στη location area A_n είναι:

$$q_n = \sum_{i \in A_n} p_i .$$

- Εάν το κινητό είναι στη location area A_n , τότε ο αριθμός των θέσεων όπου αναζητήθηκε είναι:

$$s_n = \sum_{j=1}^n k_j .$$

- Ορίζουμε το κόστος L του paging ως τον αριθμό των location που αναζητήθηκαν μέχρι να βρεθεί το κινητό. Παρατηρούμε ότι $P\{L = s_n\} = q_n$ και αυτό:

$$E[L] = \sum_{n=1}^{\infty} s_n q_n .$$

Πρόβλημα paging (3/3)

- Αφού όλες οι θέσεις μέσα σε μια location area γίνονται paged ταυτόχρονα, η paging delay D ισούται με τον αριθμό των location areas που αναζητούνται πριν βρεθεί το κινητό. Σημειώνουμε ότι $P\{D = n\} = q_n$ και αυτό:

$$E[D] = \sum_{n=1}^{\infty} n q_n .$$

- Το πρόβλημα είναι η ελαχιστοποίηση του $E[L]$ υποκείμενο στον περιορισμό του $E[D]$ για όλες τις paging στρατηγικές.

Το βασικό θεώρημα του paging

Theorem 1. To minimize $E[L]$ or $E[D]$, more probable locations must not be searched after less probable locations. Formally, if i and j are locations with $p_i > p_j$, then the location area sequence $(A_1, A_2, \dots)$ that minimizes either $E[D]$ or $E[L]$ must satisfy $i \in A_l$ and $j \in A_m$ for some $l \leq m$.

Απόδειξη βασικού θεωρήματος (1/2)

Suppose the set $(A_1, A_2, \dots)$ is optimal but there exists $i \in A_l$ and $j \in A_m$ with $p_i < p_j$ but $l > m$. Let $(A'_1, A'_2, \dots)$ denote a new paging sequence derived from $(A_1, A_2, \dots)$ in which i and j are swapped so that $i \in A'_m$ and $j \in A'_l$. For the modified paging sequence, we define the paging cost and paging delay by L' and D' . We note that

$$\begin{aligned} E[D] - E[D'] &= lp_i + mp_j - (lp_j + mp_i) \\ &= (l - m)(p_i - p_j) \\ &> 0. \end{aligned}$$

Απόδειξη βασικού θεωρήματος (2/2)

This is a contradiction of the assumed optimality of $\{A_n\}$. Likewise for $E[L]$ we have

$$\begin{aligned} E[L] - E[L'] &= s_l p_i + s_m p_j - (s_l p_j + s_m p_i) \\ &= (s_l - s_m)(p_i - p_j) \\ &> 0, \end{aligned}$$

which also contradicts the assumed optimality. □

Location databases

- Αρχιτεκτονικές δυο επιπέδων
 - Per user location caching
 - Replication
 - Forwarding pointers
- Αρχιτεκτονικές πολλαπλών επιπέδων (Δεν θα μας απασχολήσουν στο μάθημα αυτό)
 - Caching
 - Replication
 - Forwarding pointers
 - Partitions
- **Call-to-Mobility ratio (CMR) [Local CMR]**

$$\text{CMR} \equiv \frac{\text{Call Arrival Rate}}{\text{LA Crossing Rate}} = \frac{\lambda}{\mu}$$

Per User Location Caching

- Κάθε φορά που καλείται το κινητό x , η θέση του x γίνεται **cached** στη VLR της LA του καλούντος
- Οποιαδήποτε μετέπεια κλήση για τον x που προκύπτει από τη συγκεκριμένη LA, μπορεί να χρησιμοποιήσει αυτή την πληροφορία
 - Δεν χρειάζεται επικοινωνία με την HLR του x .

Cache Invalidation

- **Eager Caching**

- Οποτεδήποτε το κινητό μπαίνει σε νέα LA, όλες οι cache entries για τη θέση αυτού του κινητού ενημερώνονται
- Το κόστος της ενημέρωσης θέσης αυξάνει εάν το κινητό μετακινείται συχνά

- **Lazy Caching**

- Δεν εκτελείται ενημέρωση της cache
- Δυο περιπτώσεις μπορεί να προκύψουν: **hit** ή **miss**
- Στη περίπτωση του miss:
 - Επικοινωνία με την HLR
 - Συνεπάγεται επιπλέον κόστος, αφού η cached VLR πρέπει να ερωτηθεί πρώτα

User Profile Replication (1/2)

- Παρατήρηση:
 - Κάθε χρήστης (συνήθως) επικοινωνεί συχνά με μικρό αριθμό πηγών
- Πώς μπορούμε να αξιοποιήσουμε αυτή την παρατήρηση?
- Τα profiles των κινητών are **replicated** σε **επιλεγμένες location βάσεις** για να **ελαττώσουν το κόστος ερώτησης της HLR**

User Profile Replication (2/2)

- Όταν γίνεται μια κλήση από κάποια LA, το αντίστοιχο MSC προσδιορίζει εάν μια replica (αντίγραφο) του profile του καλούμενου κινητού είναι διαθέσιμη τοπικά
 - Εάν ναι, δεν ερωτάται η HLR του κινητού
- Όταν το κινητό μετακινείται σε άλλη περιοχή, το δίκτυο ενημερώνει όλες τις replicas

Pointer Forwarding

- Κάθε φορά που το κινητό μετακινείται σε νέα LA, ένας forwarding pointer δημιουργείται στην παλιά VLR και δείχνει στη νέα
- Κλήσεις στο κινητό πρώτα ρωτούν τη HLR για να προσδιορίσουν την πρώτη VLR και κατόπιν ακολουθούν την “αλυσίδα” μέχρι να φτάσουν στην τρέχουσα VLR
- Το μήκος της αλυσίδας των pointers περιορίζεται σε μέγιστη τιμή N
- Αυτή η μέθοδος μπορεί να ελαττώσει το κόστος ενημέρωσης της HLR

Σύγκριση των σχημάτων

Method	Variations	Applicable when:
Caching When x is called by y, cache x's location at y's zone	<i>Eager caching:</i> Cache update overhead occurs at moves	Large LCMR Call Stability
	<i>Lazy caching:</i> Cache update overhead occurs at calls	
Replication Selectively replicate x's address at the zones from which it receives the most calls	<i>Per-user Profile Replication:</i> Additional constraints are set on the number of replicas per site and on the number of replicas per user	Large LCMR Call Stability
	<i>Working Set:</i> Adaptive and distributed: the replication sites are computed dynamically by each mobile host locally	
Forwarding Pointers When x moves, add a forwarding pointer from its old to its new address	Restrict the length of the chain of forwarding pointers	Small LCMR

Σύνοψη εύρεσης κινητών

Τι (granularity), πού (availability) πότε (currency) να αποθηκευτεί

