

# Κινητός και Διάχυτος Υπολογισμός (Mobile & Pervasive Computing)

Δημήτριος Κατσαρός

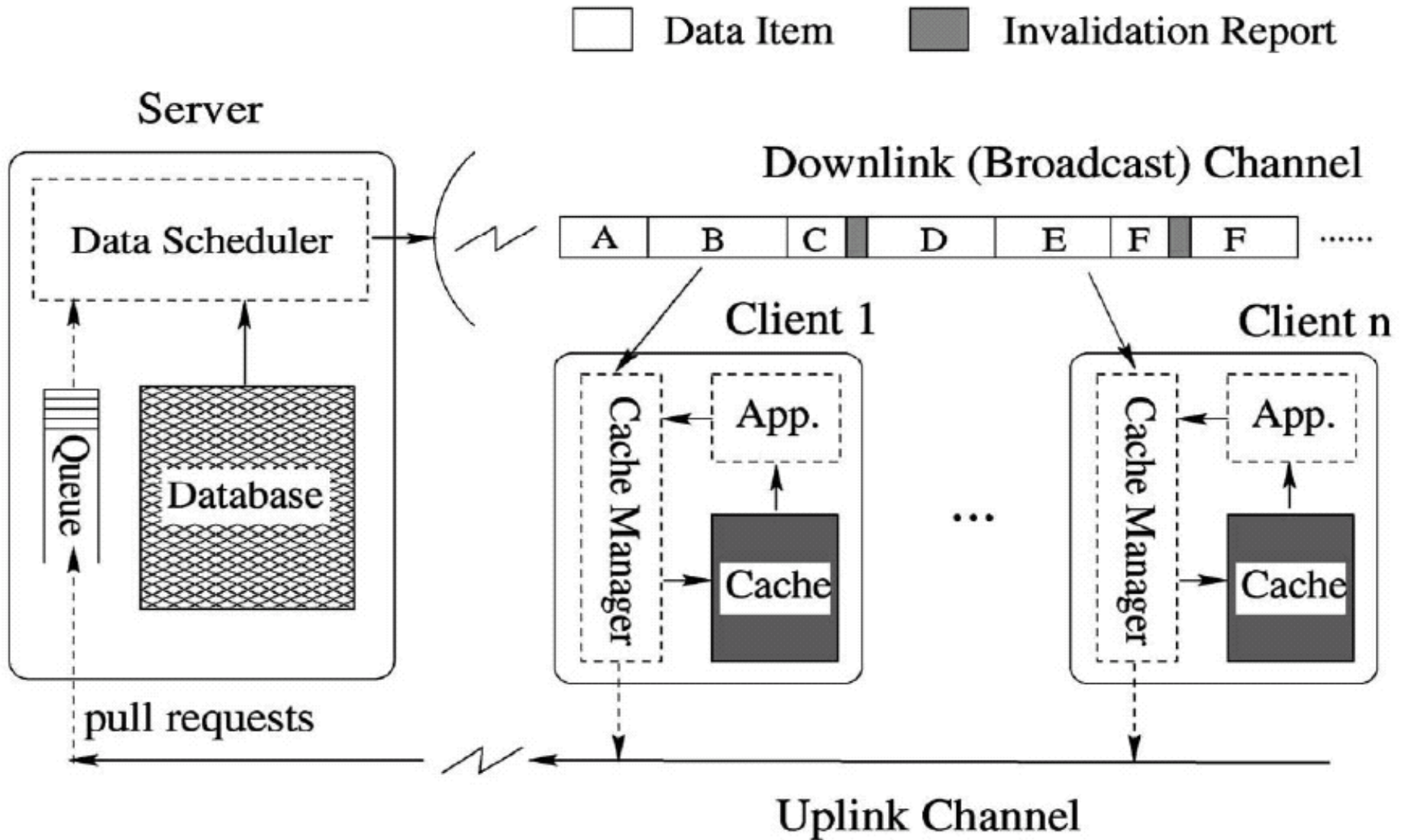
Χειμώνας 2015

Διάλεξη 10η

# Περιεχόμενα

- Πολιτική αντικατάστασης με συνέπεια  
cache

# Αρχιτεκτονική συστήματος



# Ορολογία (1/2)

- $D$ : the number of data items in the database.
- $C$ : the size of the client cache.
- $\bar{a}_i$ : mean access arrival rate of data item  $i$ ,

$$i = 1, 2, \dots, D.$$

- $\bar{u}_i$ : mean update arrival rate of data item  $i$ ,

$$i = 1, 2, \dots, D.$$

- $x_i$ : the ratio of update rate to access rate for data item  $i$ , i.e.,  $x_i = \bar{u}_i / \bar{a}_i$ ,  $i = 1, 2, \dots, D$ .
- $p_i$ : access probability of data item  $i$ ,  $p_i = \bar{a}_i / \sum_{k=1}^D \bar{a}_k$  for  $i = 1, 2, \dots, D$ .

# Ορολογία (2/2)

- $l_i$ : access latency of data item  $i$ ,  $i = 1, 2, \dots, D$ .
- $b_i$ : retrieval delay from the server (i.e., cache miss penalty) for data item  $i$ ,  $i = 1, 2, \dots, D$ .
- $s_i$ : size of data item  $i$ ,  $i = 1, 2, \dots, D$ .
- $v$ : cache validation delay, i.e., access latency of an effective invalidation report.
- $d_k$ : the data item requested in the  $k$ th access,<sup>1</sup>  $d_k \in \{1, 2, \dots, D\}$ .
- $C_k$ : the set of cached data items after the  $k$ th access,  $C_k \subseteq \{1, 2, \dots, D\}$ .
- $U_k$ : the set of cached data items that are updated between the  $k$ th access and the  $(k+1)$ th access,  $U_k \subseteq C_k$ .
- $V_k$ : the set of victims chosen to be replaced in the  $k$ th access,  $V_k \subseteq (C_{k-1} - U_{k-1})$ .

# Η πολιτική Min-SAUD

- Ορίζουμε ένα μέτρο αξίας κάθε αντικειμένου ως εξής:

$$gain(i) = \frac{p_i}{s_i} \left( \frac{b_i}{1 + x_i} - v \right)$$

- Μετά από  $k$  προσπελάσεις, επιθυμούμε να μεγιστοποιήσουμε τη συνολική αξία των αντικειμένων στην cache, δηλ, να αναγνωρίσουμε το βέλτιστο victim set  $V_k^*$ :  $V_k^* \subseteq C_{k-1} - U_{k-1}$ ,

$$V_k^* = arg \min_{V_k \subseteq (C_{k-1} - U_{k-1})} \sum_{i \in V_k} gain(i)$$

$$s.t. \quad \sum_{i \in V_k} s_i \geq \sum_{j \in (C_{k-1} - U_{k-1})} s_j + s_{d_k} - C$$

# Optimality της Min-SAUD (1/8)

- Θα δείξουμε ότι η Min-SAUD είναι βέλτιστη για το stretch μέτρο
- Υποθέτουμε Independence Reference Model
- Arrival και Updates είναι Poisson κατανομημένες
- Interarrival times για αιτήσεις και τροποποιήσεις είναι Εκθετικά κατανομημένες
- Δηλ., οι αντίστοιχες density functions είναι:

$$f(t_i^a) = \bar{a}_i e^{-\bar{a}_i t_i^a}$$

$$g(t_i^u) = \bar{u}_i e^{-\bar{u}_i t_i^u}$$

# Optimality της Min-SAUD (2/8)

- Το access cost είναι το γινόμενο πιθανότητας προσπέλασης επί stretch (access latency προς service time)
- Αγνοώντας το εύρος ζώνης, χρησιμοποιούμε το relative access cost, ορισμένο ως πιθανότητα προσπέλασης επί access latency προς μέγεθος, δηλαδή μετά από  $k$  προσπελάσεις, έχουμε:

$$S_k = \sum_{1 \leq i \leq D} p_i \cdot \frac{l_i}{s_i}$$



# Optimality της Min-SAUD (3/8)

- Θυμηθείτε ότι ακόμα και εάν έχουμε cache hit, δεν μπορούμε να το δώσουμε στην application, παρά μόνο μετά που θα δούμε την IR που περιέχει πληροφορία για το συγκεκριμένο αντικείμενο.
- Έστω ότι  $\Pr(U_i)$  είναι η πιθανότητα ότι το αντικείμενο  $i$  έχει τροποποιηθεί στο διάστημα από την τρέχουσα στιγμή μέχρι την άφιξη μιας τέτοιας IR μετά την query για το  $i$ . Η προηγούμενη εξίσωση μπορεί να γραφεί:

$$\begin{aligned}
 S_k &= \sum_{i \in C_k} \frac{p_i \cdot l_i}{s_i} + \sum_{i \notin C_k} \frac{p_i \cdot l_i}{s_i} \\
 &= \sum_{i \in C_k} \frac{p_i \cdot l_i}{s_i} \Pr(U_i) + \sum_{i \in C_k} \frac{p_i \cdot l_i}{s_i} (1 - \Pr(U_i)) + \sum_{i \notin C_k} \frac{p_i \cdot l_i}{s_i}
 \end{aligned}$$

# Optimality της Min-SAUD (4/8)

- Η εξίσωση της προηγούμενης διαφάνειας αντιστοιχεί σε τρεις περιπτώσεις, τις ακόλουθες:
  1. a cache hit but an obsolete copy,
  2. a cache hit and an up-to-date copy,
  3. a cache miss.
- Η access latency  $l_i$  είναι αντίστοιχα:

$$l_i = \begin{cases} v + b_i & \text{if } i \in C_k \text{ and an obsolete copy;} \\ v & \text{if } i \in C_k \text{ and an up-to-date copy;} \\ b_i & \text{if } i \notin C_k. \end{cases}$$

# Optimality της Min-SAUD (5/8)

- Θα παράξουμε το  $\Pr(U_i)$
- Έστω ότι  $\Pr(U'_i)$  είναι η πιθανότητα τροποποίησης του  $i$  από την τρέχουσα στιγμή μέχρι τη στιγμή της επόμενης query για το  $i$ .
- Προσεγγίζοντας την  $\Pr(U_i)$  με την  $\Pr(U'_i)$ , έχουμε:

$$\begin{aligned} \Pr(U_i) &\doteq \Pr(U'_i) = \Pr(t_i^u < t_i^a) = \int_{t_i^a=0}^{\infty} \int_{t_i^u=0}^{t_i^a} f(t_i^a) g(t_i^u) dt_i^u dt_i^a \\ &= \frac{\overline{u}_i}{\overline{u}_i + \overline{a}_i}. \end{aligned}$$

- Συνδυάζοντας τις εξισώσεις για το  $S_k$ ,  $l_i$  και  $\Pr(U_i)$ , έχουμε:

# Optimality της Min-SAUD (6/8)

$$S_k = \sum_{i \in C_k} \left( \frac{p_i(v + b_i)}{s_i} \cdot \frac{\bar{u}_i}{\bar{u}_i + \bar{a}_i} \right) + \sum_{i \in C_k} \left( \frac{p_i \cdot v}{s_i} \cdot \frac{\bar{a}_i}{\bar{u}_i + \bar{a}_i} \right) +$$

$$\sum_{i \notin C_k} \frac{p_i \cdot b_i}{s_i} = \sum_{i \in C_k} \frac{p_i(v + \frac{\bar{u}_i b_i}{\bar{u}_i + \bar{a}_i})}{s_i} + \sum_{i \notin C_k} \frac{p_i \cdot b_i}{s_i}.$$

- Με βάση την παραπάνω εξίσωση μπορούμε να δείξουμε ότι:

**Theorem 1.** *The replacement policy Min-SAUD gives better access cost, in terms of stretch, than any other replacement policy.*

# Optimality της Min-SAUD (7/8)

- Θα αποδείξουμε την optimality της Min-SAUD, εάν δείξουμε ότι το κόστος  $S_k$  είναι πάντα το μικρότερο, εάν χρησιμοποιούμε την πολιτική Min-SAUD. Με επαγωγή
- Έστω ότι  $S_w$  είναι το βέλτιστο κόστος για κάποιο  $k=w$  για μια άλλη πολιτική
- Έστω ότι  $V_{w+1}$  είναι το victim set για να κάνουμε χώρο για το  $d_{w+1}$ . Επομένως, έχουμε ότι

$$C_{w+1} = C_w - U_w \cup \{d_{w+1}\} - V_{w+1}$$

- Συνεπώς:

# Optimality της Min-SAUD (8/9)

$$\begin{aligned}
 S_{w+1} &= \sum_{i \in C_{w+1}} \frac{p_i \left( v + \frac{\bar{u}_i b_i}{\bar{u}_i + \bar{a}_i} \right)}{s_i} + \sum_{i \notin C_{w+1}} \frac{p_i \cdot b_i}{s_i} \\
 &= S_w + \sum_{i \in U_w} \left( \frac{p_i}{s_i} \left( \frac{b_i}{1 + x_i} - v \right) \right) - \frac{p_{d_{w+1}}}{s_{d_{w+1}}} \left( \frac{b_{d_{w+1}}}{1 + x_{d_{w+1}}} - v \right) \\
 &\quad + \sum_{i \in V_{w+1}} \left( \frac{p_i}{s_i} \left( \frac{b_i}{1 + x_i} - v \right) \right) \\
 &= B + \sum_{i \in V_{w+1}} \frac{p_i}{s_i} \left( \frac{b_i}{1 + x_i} - v \right),
 \end{aligned}$$

# Optimality της Min-SAUD (9/9)

- Όπου το  $B$  είναι ίσο με:

$$B = S_w + \sum_{i \in U_w} \left( \frac{p_i}{s_i} \left( \frac{b_i}{1 + x_i} - v \right) \right) - \frac{p_{d_{w+1}}}{s_{d_{w+1}}} \left( \frac{b_{d_{w+1}}}{1 + x_{d_{w+1}}} - v \right)$$

- Αφού το  $B$  δεν ελέγχεται από την πολιτική αντικατάστασης, συμπεραίνουμε ότι το ελάχιστο access cost επιτυγχάνεται όταν διώχνονται τα αντικείμενα με το μικρότερο :

$$\sum_{i \in V_{w+1}} \frac{p_i}{s_i} \left( \frac{b_i}{1 + x_i} - v \right)$$

# Κινητός και Διάχυτος Υπολογισμός (Mobile & Pervasive Computing)

Δημήτριος Κατσαρός

Χειμώνας 2015

Διάλεξη 11η



# Περιεχόμενα

- **Επαναχρησιμοποίηση καναλιών**

## Βασικές έννοιες (1/2)

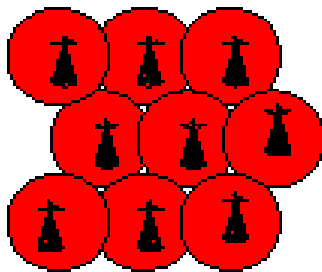
- Τα πρώτα συστήματα κινητής τηλεφωνίας δεν ήταν κυψελοειδή. Για την κάλυψη μιας μεγάλης περιοχής χρειαζόταν ένας **ισχυρός πομπός** και μια **κεραία σε υψηλό μέρος**. Δεν γινόταν επαναχρησιμοποίηση συχνοτήτων. Αυτό είχε σαν αποτέλεσμα **πολύ μικρή χωρητικότητα**
- Η ιδέα του **κυψελοειδούς συστήματος** προέκυψε από την ανάγκη επέκτασης του ραδιοτηλεφωνικού συστήματος για την κάλυψη της αυξημένης ζήτησης, η οποία δεν μπορούσε να ικανοποιηθεί απλά με την χρησιμοποίηση προσθετού φάσματος συχνοτήτων
  - **Κυψελοειδής κάλυψη**: Αντικατάσταση των μεγάλων πομπών με πολλούς μικρότερους. Γειτονικοί σταθμοί βάσης λειτουργούν με διαφορετικές ομάδες καναλιών (συχνοτήτων). Η χωρητικότητα μπορεί να αυξηθεί με επιπρόσθετο διαμέρισμα του φάσματος και της περιοχής.

# Βασικές έννοιες (2/2)

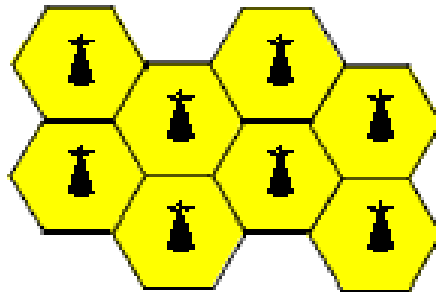
- **Κύτταρο ή κυψέλη (Cell)** – μια γεωγραφική περιοχή που καλύπτεται από ένα B.S.
- **Επαναχρησιμοποίηση συχνότητας (frequency reuse)** – το σχέδιο κατανομής των συχνοτήτων των καναλιών στους B.S

Για λόγους ευκολίας, τα κύτταρα απεικονίζονται με εξαγωγικό σχήμα. Το εξάγωνο είναι το απλούστερο σχήμα με το οποίο μπορούμε να ψηφιοποιήσουμε μια επιφάνεια (space tessellation).

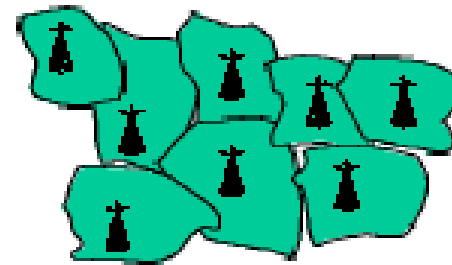
Στην πράξη, τα κύτταρα δεν είναι εξάγωνα και οι σταθμοί βάσης δεν τοποθετούνται πάντοτε στο κέντρο των κυττάρων.



Θεωρητικό  
διάγραμμα  
κάλυψης



Κυψελοειδές  
πλέγμα κυττάρων



Πραγματικό  
διάγραμμα κάλυψης

# Υπολογισμός χωρητικότητας

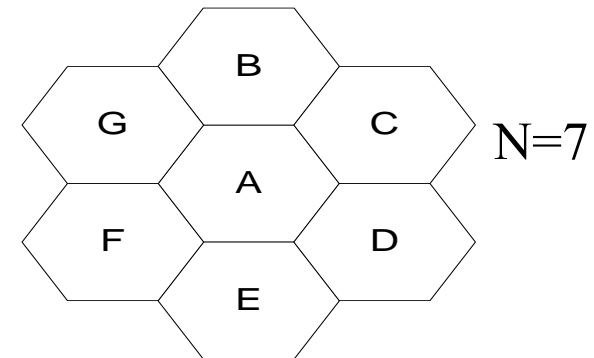
Υποθέτουμε ότι υπάρχουν  $N$  κύτταρα, και σε κάθε κύτταρο έχουμε διαθέσιμους  $k$  κανάλια (συχνότητες) από τα  $S$  διαθέσιμα. Αυτά τα  $N$  κύτταρα λέμε ότι αποτελούν μια **συστάδα (cluster)**. Ο συνολικός αριθμός καναλιών ανά συστάδα είναι

$$S = kN$$

Η συνολική χωρητικότητα που παρέχουν  $M$  συστάδες είναι:

$$C = MkN = MS$$

Μια συστάδα μπορεί να αντιγραφεί περισσότερες φορές σε μια περιοχή αν τα κύτταρα γίνουν μικρότερα (σημειώστε ότι στην περίπτωση αυτή η ισχύς εκπομπής θα πρέπει να μειωθεί κατάλληλα). Έτσι επιτυγχάνουμε **μεγαλύτερη χωρητικότητα**.

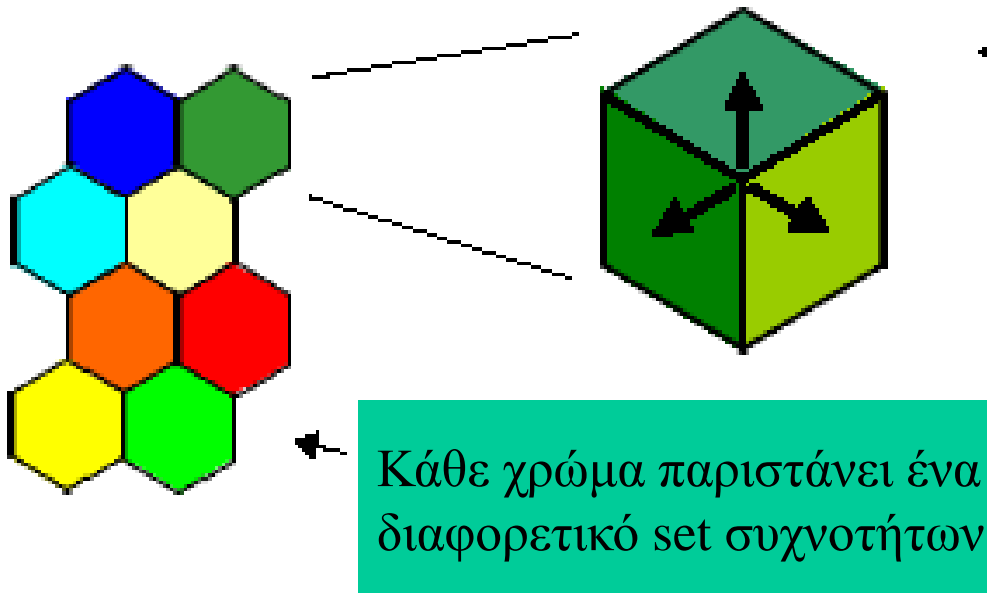


Ο **συντελεστής επαναχρησιμοποίησης συχνότητας** (frequency reuse factor) είναι ο αριθμός  $N$  των κυττάρων σε μια συστάδα.

# Επαναχρησιμοποίηση συχνοτήτων και τομεοποίηση (sectorization)

Με την επαναχρησιμοποίηση συχνοτήτων (ή κωδίκων για το CDMA) αυξάνουμε την χωρητικότητα χωρίς να χρειαζόμαστε καινούργιο φάσμα συχνοτήτων.

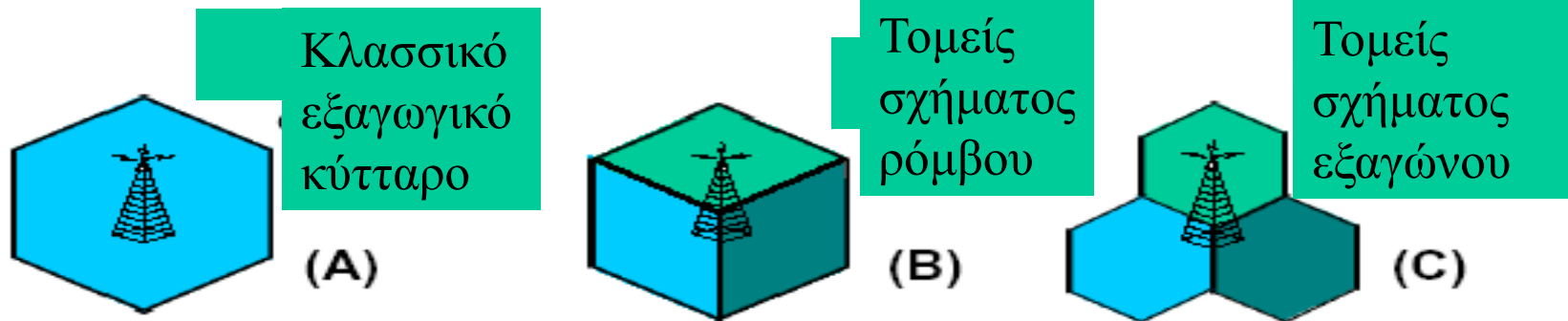
Με την διαίρεση των κυττάρων σε τομείς (sectors – sectorization) μπορούμε να αυξήσουμε ακόμα περισσότερο την χωρητικότητα



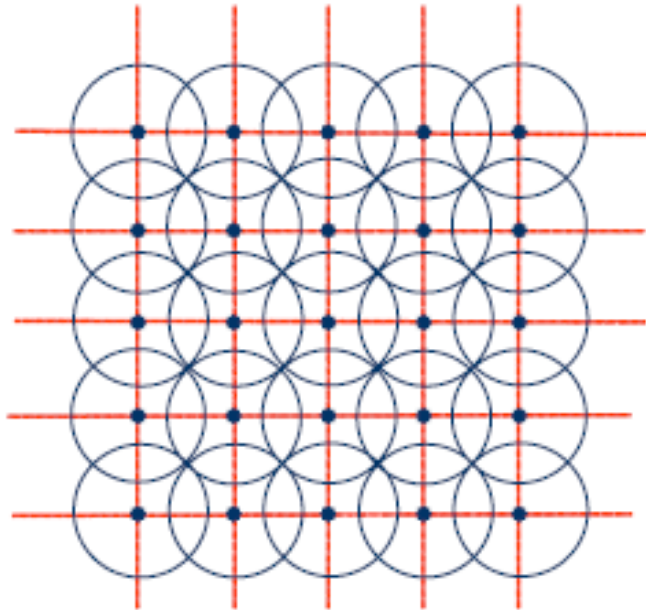
Κύτταρα με 3 τομείς είναι πολύ συνηθισμένα (κάθε τομέας είναι  $120^\circ$ ). Μερικοί ονομάζουν τους τομείς κύτταρα. Κύτταρα με 2 τομείς συνηθίζονται για την κάλυψη επαρχιακών αυτοκινητοδρόμων.

# Μορφές κυττάρων

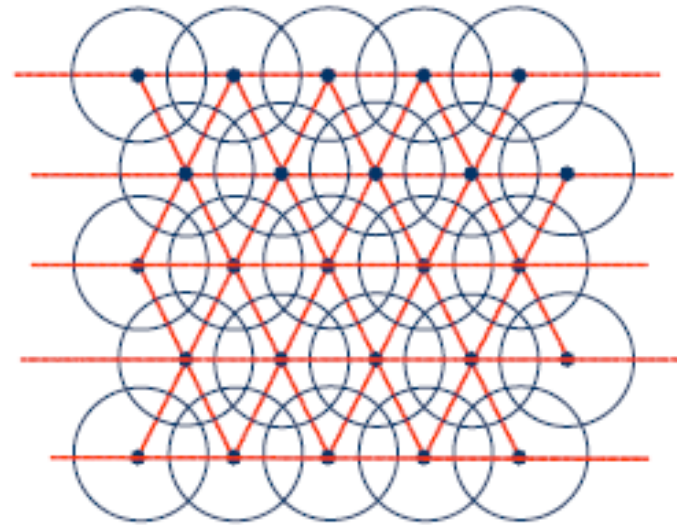
- Αρχικά, τα κύτταρα είχαν εξαγωγικό σχήμα και η κάλυψη του σταθμού βάσης ήταν **πανκατευθυντική (omni-directional)** (A)
- Με την εφαρμογή της **τμηματοποίησης**, τα κύτταρα παίρνουν σχήμα ρόμβου (B)
- Πολλοί σχεδιαστές, όμως, προτιμούν να δουλεύουν με κύτταρα εξαγωγικού σχήματος (C)



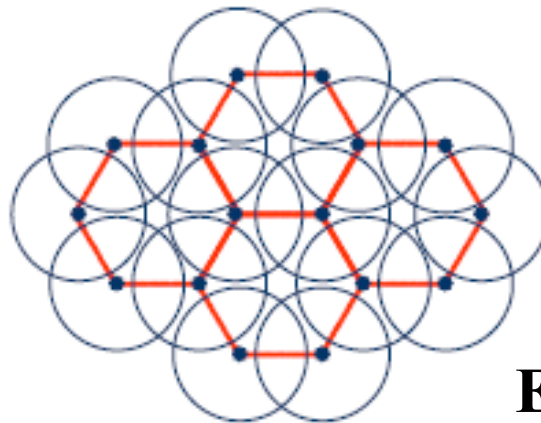
# Τοποθέτηση Σταθμών Βάσης



Σε πλέγμα



Τριγωνικό

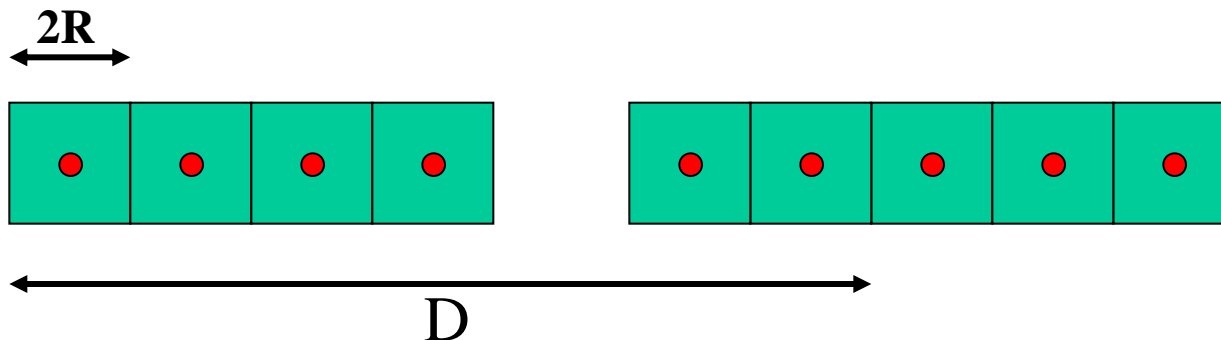


Εξαγωνικό

# Επαναχρησιμοποίηση συχνοτήτων – Μονοδιάστατα συστήματα

- Τετραγωνικά κελιά, μήκους  $2R$
- Εάν χρησιμοποιούμε τις ίδιες συχνότητες μετά από  $K$  κυψέλες, θα έχουμε  $D=2R*K$
- Αφού οι κυψέλες έχουν ίδιες διαστάσεις, τότε:

$$K = \frac{D \times 2R}{(2R)^2} = \frac{Area_{cluster}}{Area_{cell}}$$





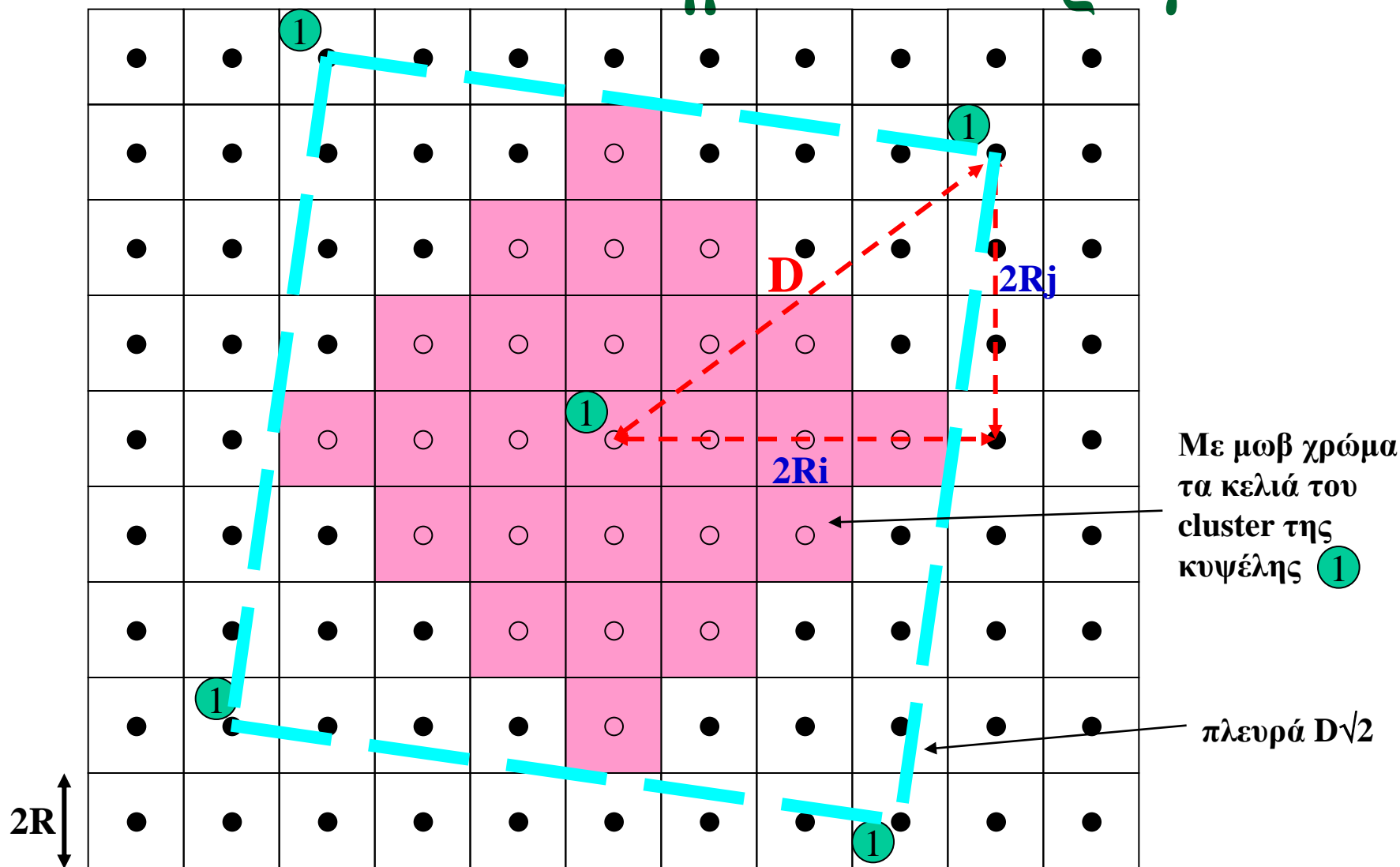
# Επαναχρησιμοποίηση συχνοτήτων – Διδιάστατα συστήματα “τετραγωνικά” με τετραγωνική τοποθέτηση των BS

- Σύστημα ορθογωνίων αξόνων
- Τα κέντρα των κυψελών απέχουν ακέραια πολλαπλάσια των ποσοτήτων  $i \cdot 2R$  και  $j \cdot 2R$
- Τότε:

$$D = 2R \times \sqrt{i^2 + j^2}$$

- Θεωρούμε τη περιοχή που ορίζεται από τα κέντρα των 4 ομοδιαυλικών κυψελών της κυψέλης 1 (επόμενη διαφάνεια)
- Λόγω συμμετρίας, αυτή η περιοχή περιέχει τις  $K$  κυψέλες της ομάδας επαναχρησιμοποίησης της κεντρικής κυψέλης και  $K/4$  των αντίστοιχων κυψελών των 4 περιφερειακών κυψελών, δηλ., συνολικά  $K + 4 \cdot (K/4) = 2K$  κυψέλες

# Επαναχρησιμοποίηση συχνοτήτων – Διδιάστατα συστήματα “τετραγωνικά”



# Επαναχρησιμοποίηση συχνότητων – Διδιάστατα συστήματα “τετραγωνικά”

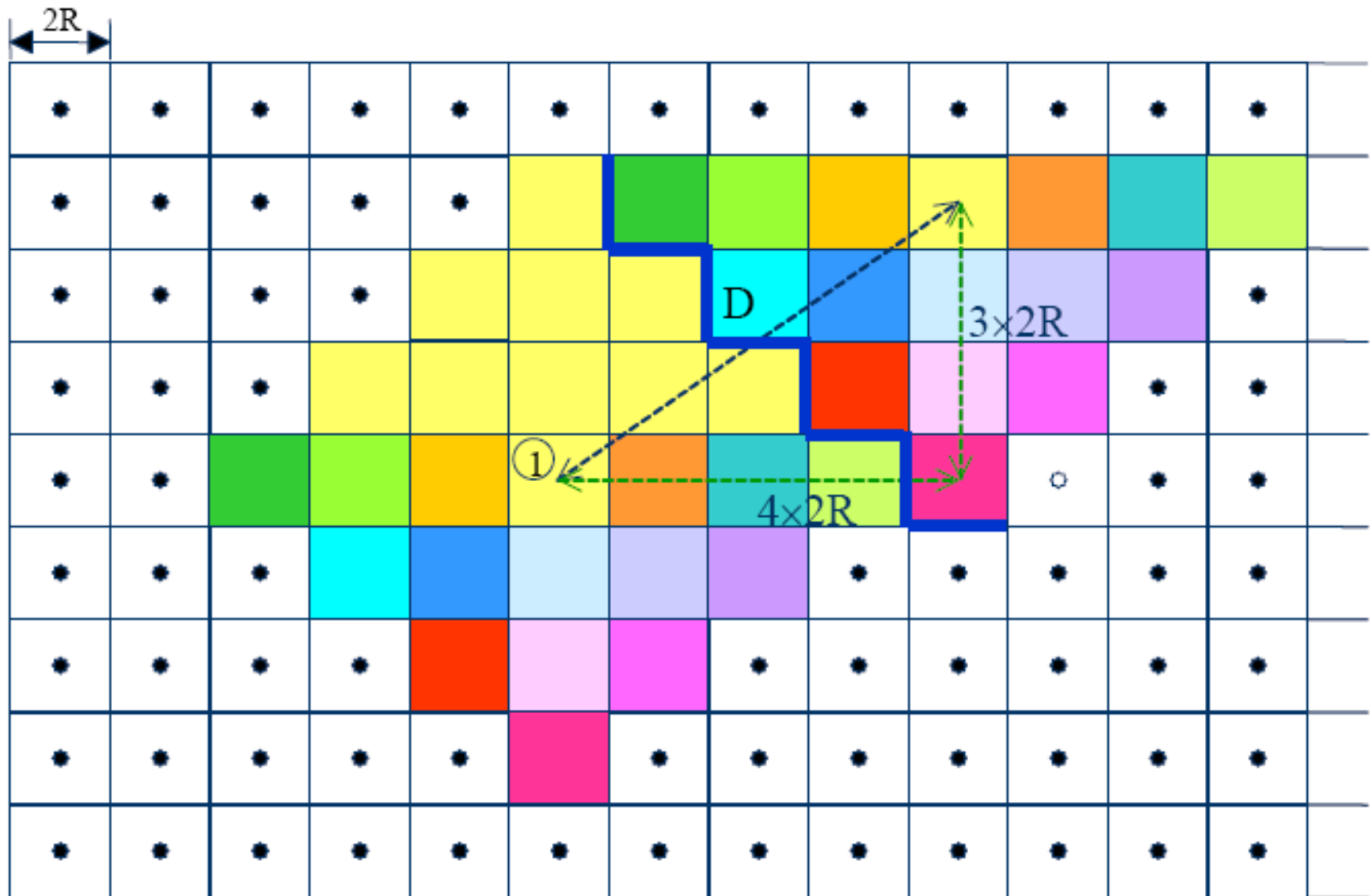
- Δεδομένου ότι το εμβαδό ενός κανονικού πολυγώνου είναι ανάλογο του τετραγώνου της πλευράς του, θα έχουμε:

$$2K = \frac{Area_{square_{D\sqrt{2}}}}{Area_{cell}} = \frac{2D^2}{(2R)^2} =$$

$$= \frac{2(i^2 + j^2) \times (2R)^2}{(2R)^2} = 2(i^2 + j^2) \Rightarrow K = (i^2 + j^2)$$

- Αφού τα  $i$  και  $j$  παίρνουν ακέραιες τιμές, το  $K$  παίρνει τιμές: 1, 2, 4, 5, 8, 9, 10, 13, 16, 17, 20, 25, ...
- Πώς βρίσκουμε τις ομοδιαυλικές κυψέλες? Ξεκινάμε από τη δοθείσα κυψέλη, πηγαίνουμε κάθετα στην κάθε πλευρά της για  $i$  βήματα, “στρίβουμε” αριστερά κατά  $90^\circ$ , και κάνουμε  $j$  βήματα

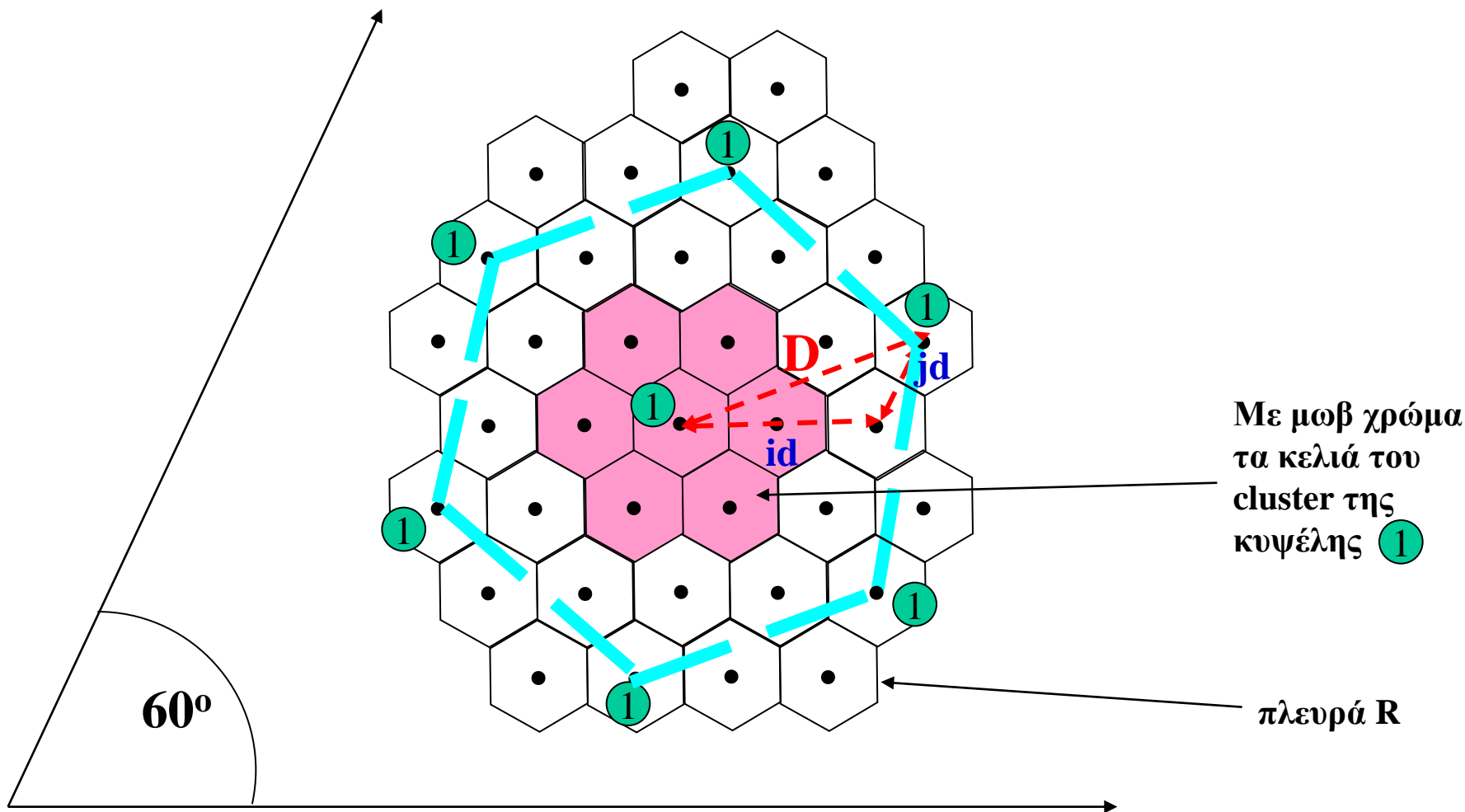
# Παράδειγμα: $K=25$ , square cell



# Επαναχρησιμοποίηση συχνοτήτων – Διδιάστατα συστήματα “εξαγωνικά” με τριγωνική τοποθέτηση BS

- Σύστημα πλαγίων αξόνων, γωνία  $60^\circ$
- Τα κέντρα των κυψελών απέχουν ακέραια πολλαπλάσια των ποσοτήτων  $i*d$  και  $j*d$
- Τότε (νόμος συνημιτόνων):  $D = (id)^2 + (jd)^2 - 2(id)(jd) \cos \theta$   
 $(\theta = 120^\circ) \Rightarrow D = d\sqrt{i^2 + ij + j^2}$
- Θεωρούμε τη περιοχή που ορίζεται από τα κέντρα των 6 ομοδιαυλικών κυψελών της κυψέλης 1 (επόμενη διαφάνεια)
- Λόγω συμμετρίας, αυτή η περιοχή περιέχει τις  $K$  κυψέλες της ομάδας επαναχρησιμοποίησης της κεντρικής κυψέλης και  $K/3$  των αντίστοιχων κυψελών των 6 περιφερειακών κυψελών, δηλ., συνολικά  $K + 6*(K/3) = 3K$  κυψέλες

# Επαναχρησιμοποίηση συχνοτήτων – Διδιάστατα συστήματα “εξαγωνικά”



# Επαναχρησιμοποίηση συχνοτήτων – Διδιάστατα συστήματα “εξαγωνικά”

- Δεδομένου ότι το εμβαδό ενός κανονικού πολυγώνου είναι ανάλογο του τετραγώνου της πλευράς του, θα έχουμε:

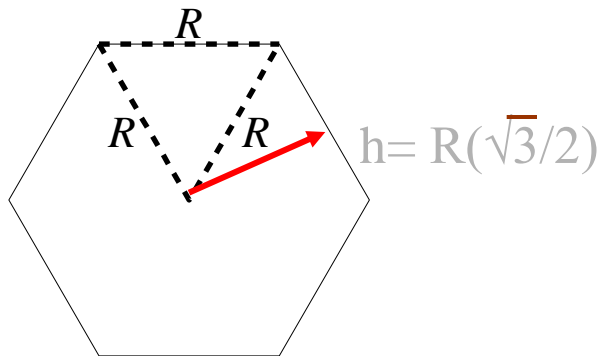
$$3K = \frac{Area_{hexagon_D}}{Area_{cell}} = \frac{D^2}{R^2}$$

$$d = R\sqrt{3} \quad K = (i^2 + ij + j^2)$$

- Αφού τα  $i$  και  $j$  παίρνουν ακέραιες τιμές, το  $K$  παίρνει τιμές: 1, 3, 4, 7, 9, 12, 13, ....
- Πώς βρίσκουμε τις ομοδιαυλικές κυψέλες? Ξεκινάμε από τη δοθείσα κυψέλη, πηγαίνουμε κάθετα στην κάθε πλευρά της για  $i$  βήματα, στρίβουμε αριστερά για  $60^\circ$  και κάνουμε  $j$  βήματα

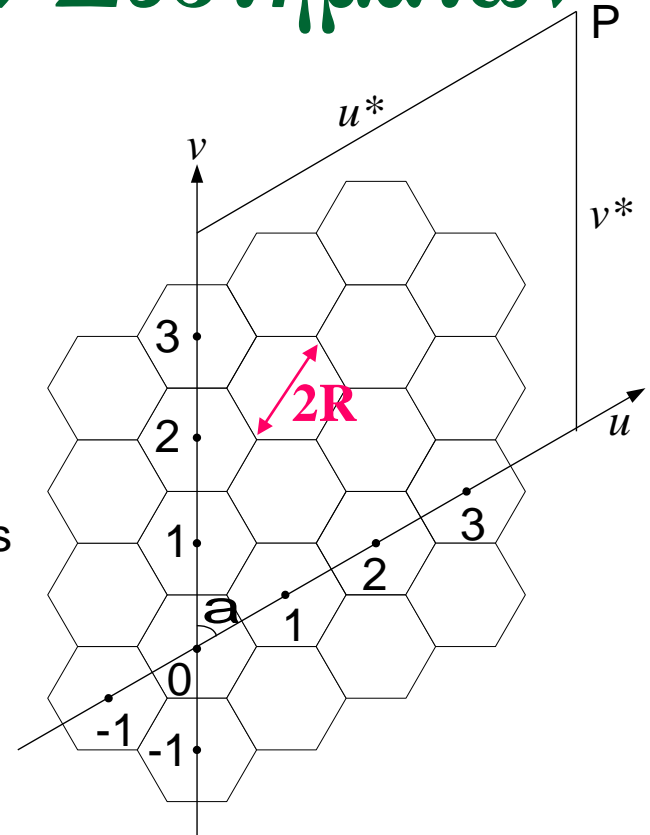
# Γεωμετρία Κυψελοειδών Συστημάτων

- Το πλάγιο σύστημα συντεταγμένων της κυψελοειδούς γεωμετρίας



$$a = 60^\circ$$

$u^*, v^*$ : coordinates of point P



- Είναι ένα βολικό σύστημα συντεταγμένων  $(u, v)$ : Μονάδα απόστασης κατά μήκος των αξόνων =  $\sqrt{3}R$  όπου  $R$  η ακτίνα του κάθε κυττάρου.
- Το κέντρο κάθε κυττάρου έχει συντεταγμένες ένα ζεύγος ακεραίων αριθμών.
- Η απόσταση των κέντρων γειτονικών κυττάρων είναι = 1



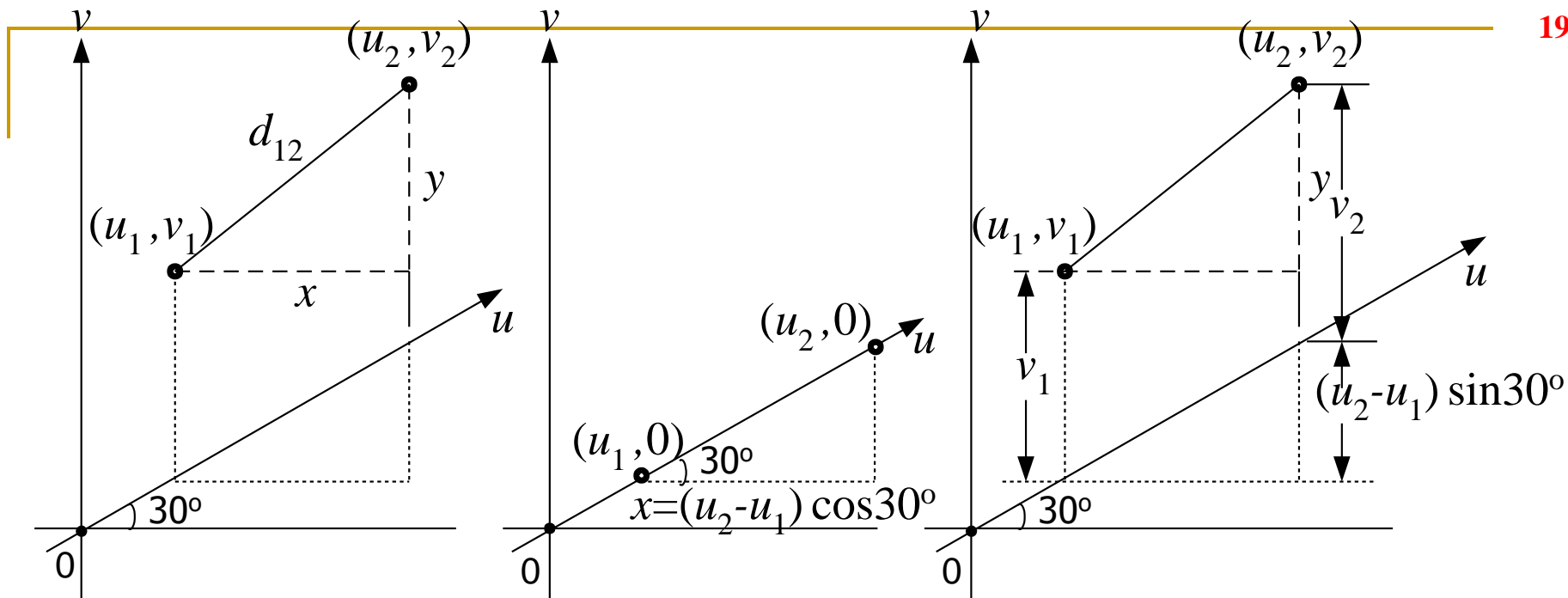
# Απόσταση μεταξύ σημείων

- **Θεώρημα:** Στο πλάγιο σύστημα συντεταγμένων  $(u,v)$  η απόσταση μεταξύ δυο σημείων  $(u_1,v_1)$  και  $(u_2,v_2)$  δίδεται από τον τύπο:

$$d_{12} = \sqrt{(u_2 - u_1)^2 + (u_2 - u_1)(v_2 - v_1) + (v_2 - v_1)^2}$$

- **Απόδειξη:** στην επόμενη διαφάνεια
- **Εφαρμογή:** αν  $(u_1,v_1) = (0,0)$  και  $(u_2,v_2) = (i,j)$  τότε η απόσταση τους είναι:

$$d_{12} = \sqrt{i^2 + ij + j^2}$$



$$d_{12}^2 = x^2 + y^2$$

$$x = (u_2 - u_1) \cos 30^\circ$$

$$y = v_2 + (u_2 - u_1) \sin 30^\circ - v_1$$

$$d_{12}^2 = (u_2 - u_1)^2 \cos^2 30^\circ + (v_2 - v_1)^2 + (u_2 - u_1)^2 \sin^2 30^\circ \\ + 2(u_2 - u_1)(v_2 - v_1) \sin 30^\circ$$

$$= (u_2 - u_1)^2 + (v_2 - v_1)^2 + (u_2 - u_1)(v_2 - v_1)$$

# Αριθμός κυττάρων σε μια συστάδα

Υποθέτουμε ότι έχουμε ένα κύτταρο στην θέση  $(0, 0)$  και ένα κύτταρο με το ίδιο σύνολο καναλιών (ομοιοκαναλικο) στην θέση  $(i, j)$ .

Τότε η απόσταση  $D$  (απόσταση επαναχρησιμοποίησης), όπως είδαμε πριν, μεταξύ των ομοιοκαναλικων κυττάρων είναι:

$$D = \sqrt{i^2 + ij + j^2}$$

□ Μια συστάδα μπορεί να παρασταθεί με ένα εξάγων ακτίνας  $R_c$ .

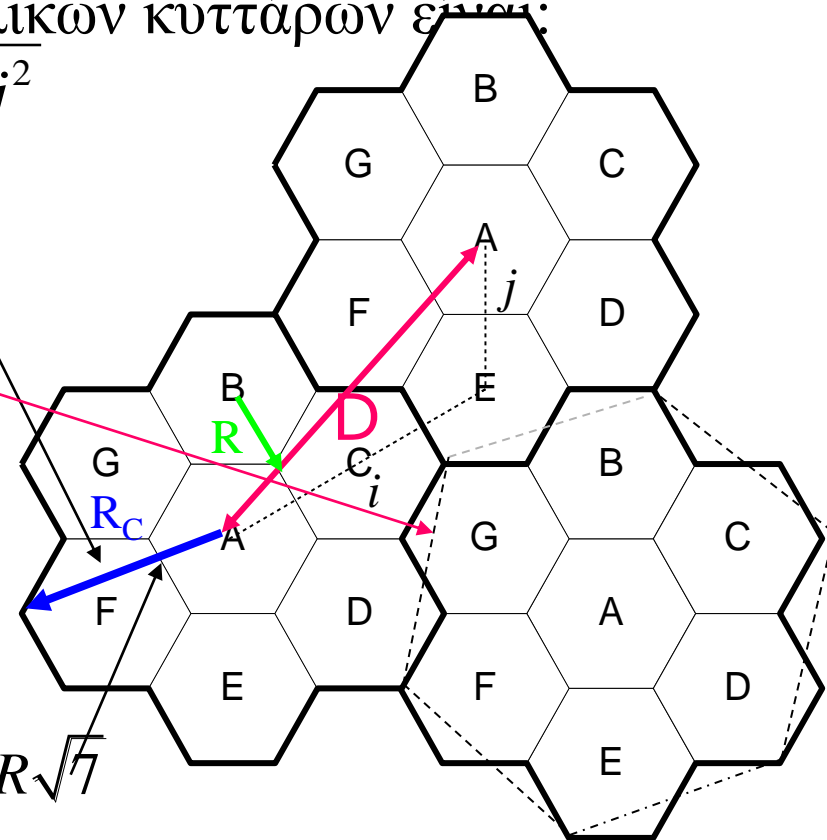
□ Η απόσταση μεταξύ των κέντρων των συστάδων είναι:

$$\sqrt{i^2 + ij + j^2}$$

□ Η ακτίνα  $R_c$  μιας συστάδας είναι

$$R_c = \frac{D}{\sqrt{3}} = R\sqrt{7}$$

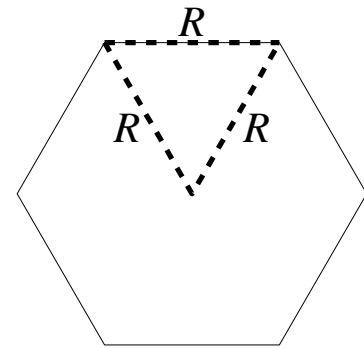
Στο σχήμα, με  $N=7$ , είναι  $i=2$  και  $j=1$   
 οπότε  $D^2 = 2^2 + 2 + 1 = 7 \rightarrow D = (R\sqrt{3})\sqrt{7}$



- Η επιφάνεια ενός εξαγώνου ακτίνας  $R$  ευρίσκεται ως εξής:

$$\text{Επιφάνεια του τριγώνου} = \frac{R}{2} R \cos 30^\circ = \frac{\sqrt{3}}{4} R^2$$

$$\text{Επιφάνεια του εξαγώνου} = 6 \frac{\sqrt{3}}{4} R^2 = \frac{3\sqrt{3}}{2} R^2$$



- Ο αριθμός των κυττάρων σε μια συστάδα

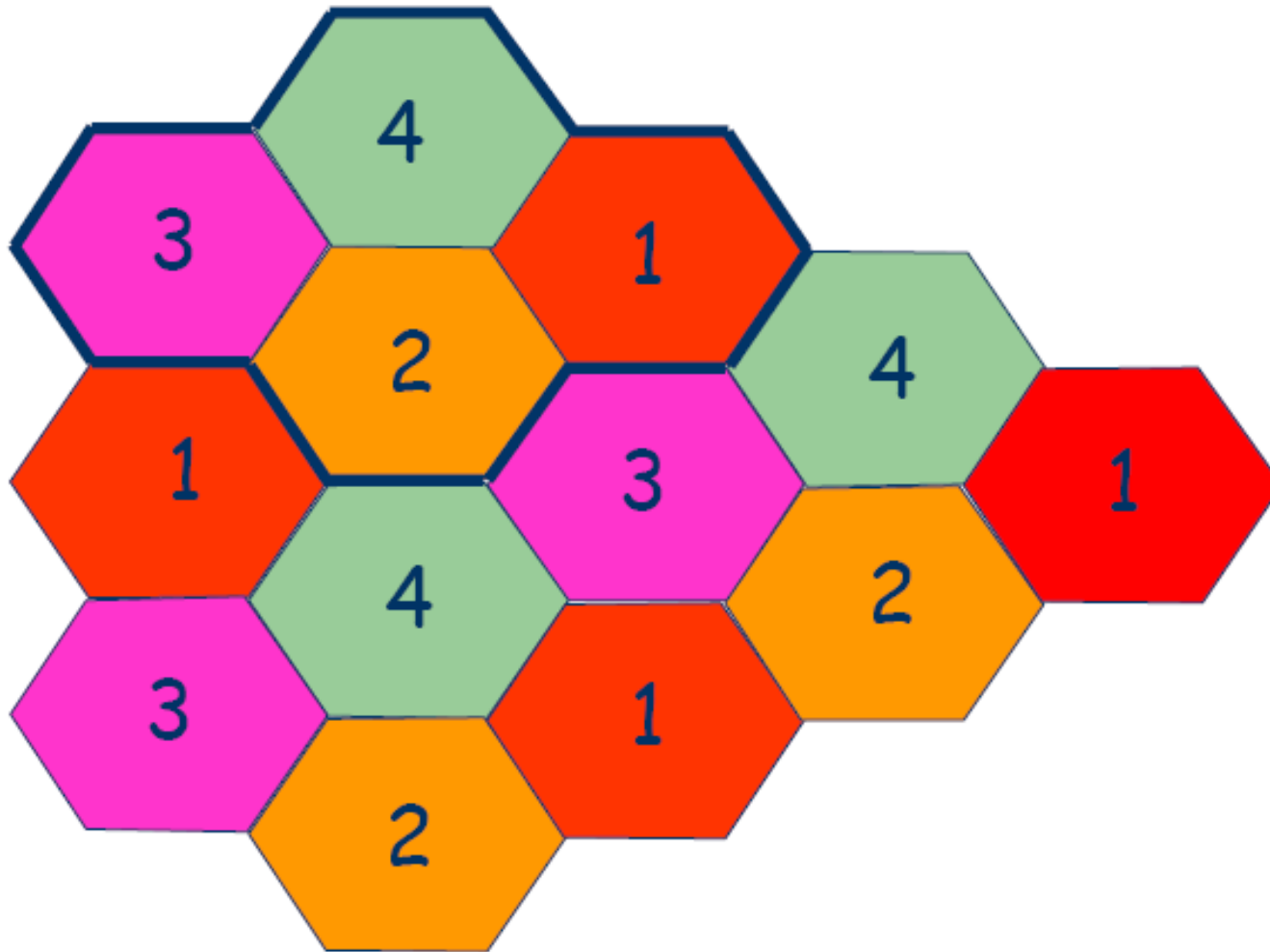
$$N = \frac{\text{επιφάνεια συσταδας}}{\text{επιφάνεια κυτταρου}} = \frac{\frac{3\sqrt{3}}{2} R_C^2}{\frac{3\sqrt{3}}{2} R^2} = \frac{D^2}{3R^2} = i^2 + ij + j^2$$

- Κατά συνεπεία οι δυνατές τιμές του  $N$  είναι 3, 4, 7, 9, 12 κ.λ.π. για απόσταση ομοιοκαναλικων κυττάρων  $(i,j) = (1,1), (0,2), (1,2), (0,3), (1,3)$

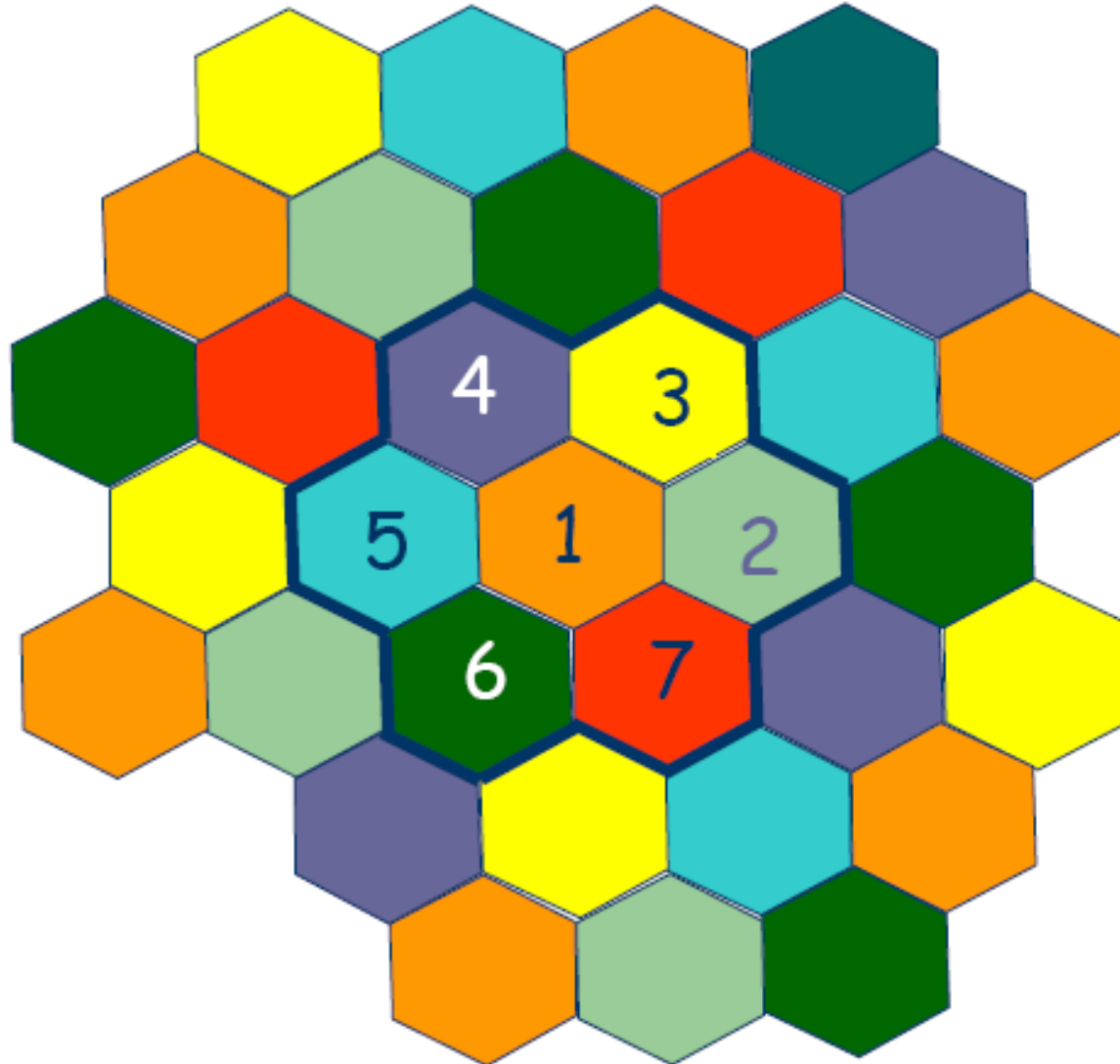
Επομένως ο λόγος  $q$  της απόστασης των ομοιοκαναλικών κυττάρων ως προς την ακτίνα του κυττάρου είναι

$$q = \frac{D}{R} = \frac{\sqrt{i^2 + ij + j^2}}{1 / \sqrt{3}} = \sqrt{3N}$$

# Παράδειγμα: $K=4$



# Παράδειγμα: $K=7$

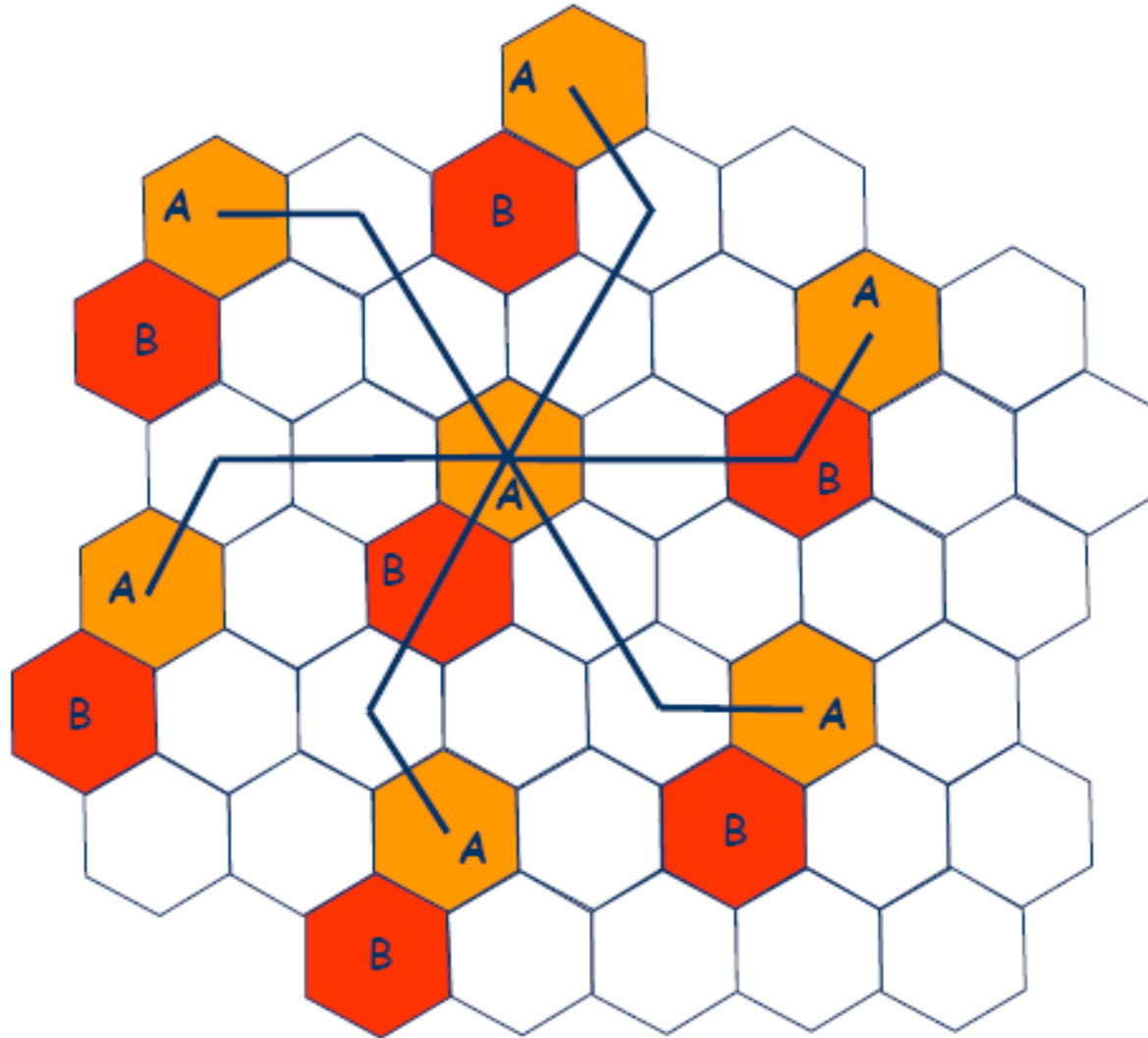


# Παράδειγμα: $K=12$





# Ομοδιαυλικές κυψέλες: $K=7$ ( $i=2, j=1$ )



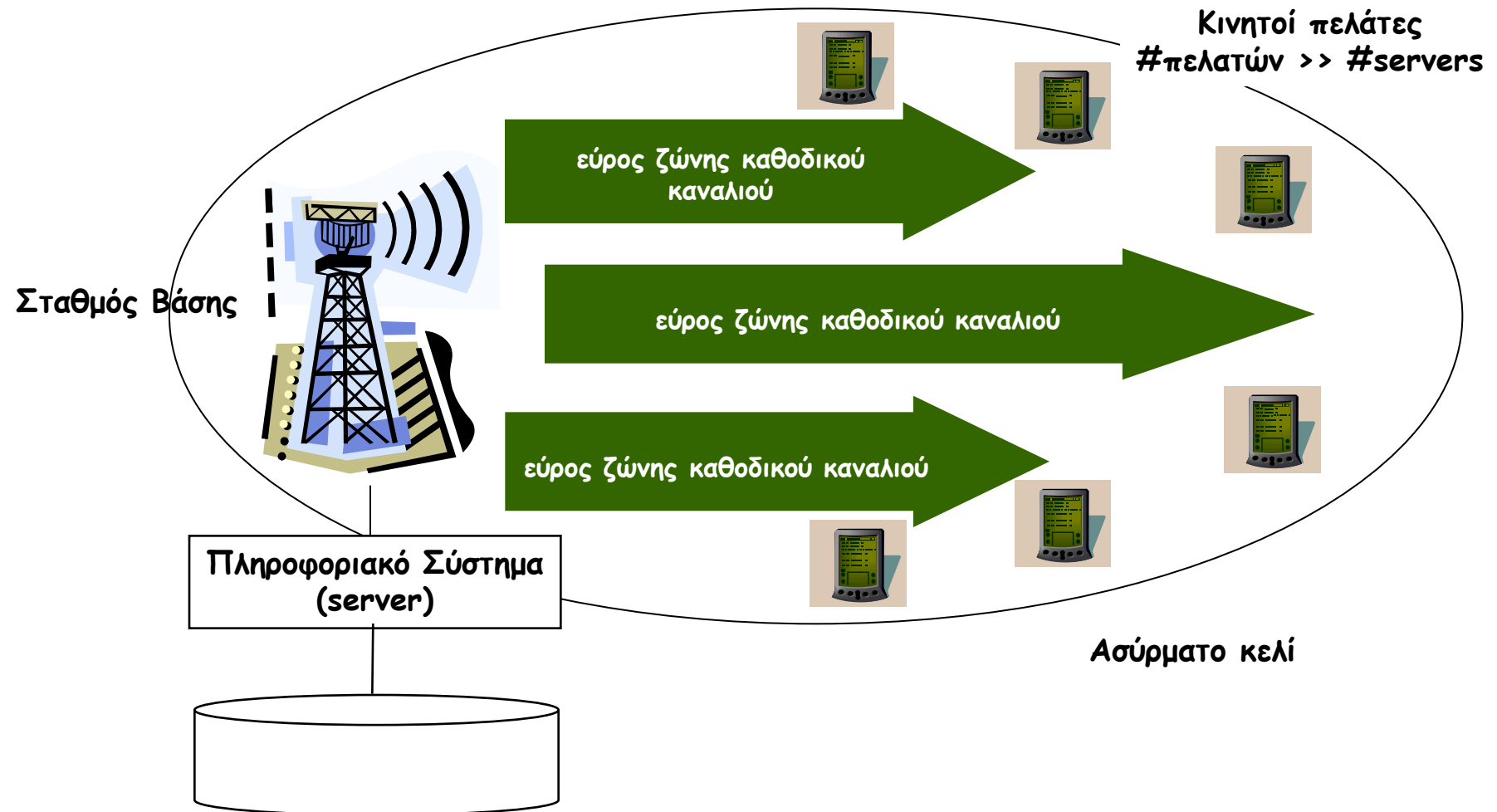
# Πολλαπλά κανάλια εκπομπής

Για λόγους όπως:

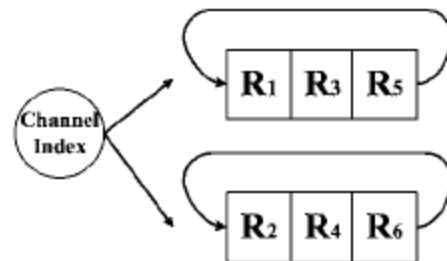
- Application scalability
  - Μια εφαρμογή αποκτά επιπλέον κανάλια για να εξυπηρετήσει μεγαλύτερο πληθυσμό
- Fault tolerance
  - Τρεις servers εκπέμπουν σε μια γεωγραφική περιοχή σε μη συνεχόμενες συχνότητες, αλλά οι δυο παθαίνουν βλάβη και τα κανάλια τους ανατίθενται στον τρίτο
- Reconfiguration of adjoining cells
  - Γειτονικά κελιά εξυπηρετούνται από διαφορετικούς servers, αλλά τα κελιά συνενώνονται και τα κανάλια ανατίθενται στον έναν από τους δυο
- Heterogeneous clients
  - Πελάτες με ετερογενείς δυνατότητες

**Είναι δυνατόν να υπάρχουν πολλαπλά κανάλια εκπομπής**

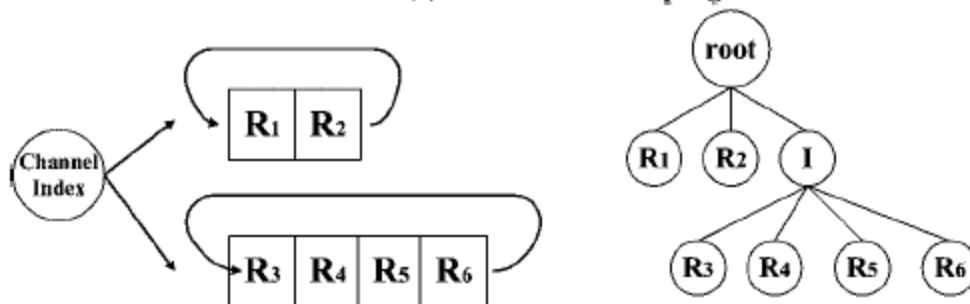
# Εμπομπή σε Πολλαπλά Κανάλια



# Ιεραρχικά προγράμματα εκπομπής



(a) A flat broadcast program

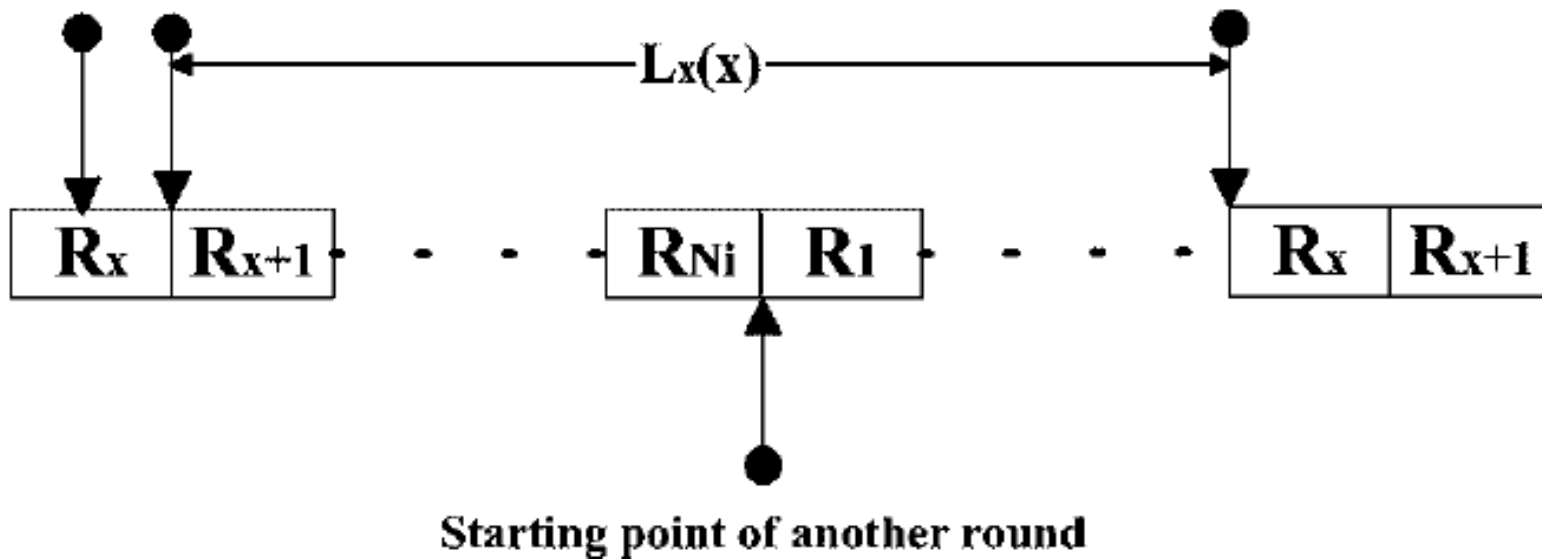


(b) A hierarchical broadcast program with its tree representation

	Access frequency						Average expected delay	
	$P_r(R_1)$	$P_r(R_2)$	$P_r(R_3)$	$P_r(R_4)$	$P_r(R_5)$	$P_r(R_6)$	in figure 1(a)	in figure 1(b)
Case 1	0.167	0.167	0.167	0.167	0.167	0.167	1	1.16
Case 2	0.25	0.25	0.125	0.125	0.125	0.125	1	1
Case 3	0.3	0.3	0.1	0.1	0.1	0.1	1	0.9
Case 4	0.4	0.4	0.05	0.05	0.05	0.05	1	0.7

# Μέση καθυστέρηση σε ένα κανάλι

Request for data item  $x$



Μέση καθυστέρηση για  
κάθε αντικείμενο στο  
κανάλι  $i$  είναι:

$$\sum_{x=1}^{N_i} (N_i - x) / N_i$$

# Δημιουργία ιεραρχικών προγραμμ.

**Property 2.** Let  $t_i = \sum_{h=1}^i N_h$  and  $t_0 = 0$ . Then,  $N_i$  data items, denoted by  $R_j$ ,  $t_{i-1} + 1 \leq j \leq t_i$ , are allocated to broadcast disk  $i$ , and  $d_i = d_{R_j}$  for  $j \in [t_{i-1} + 1, t_i]$ .

$$\sum_{j=1}^n d_{R_j} \cdot P_r(R_j)$$

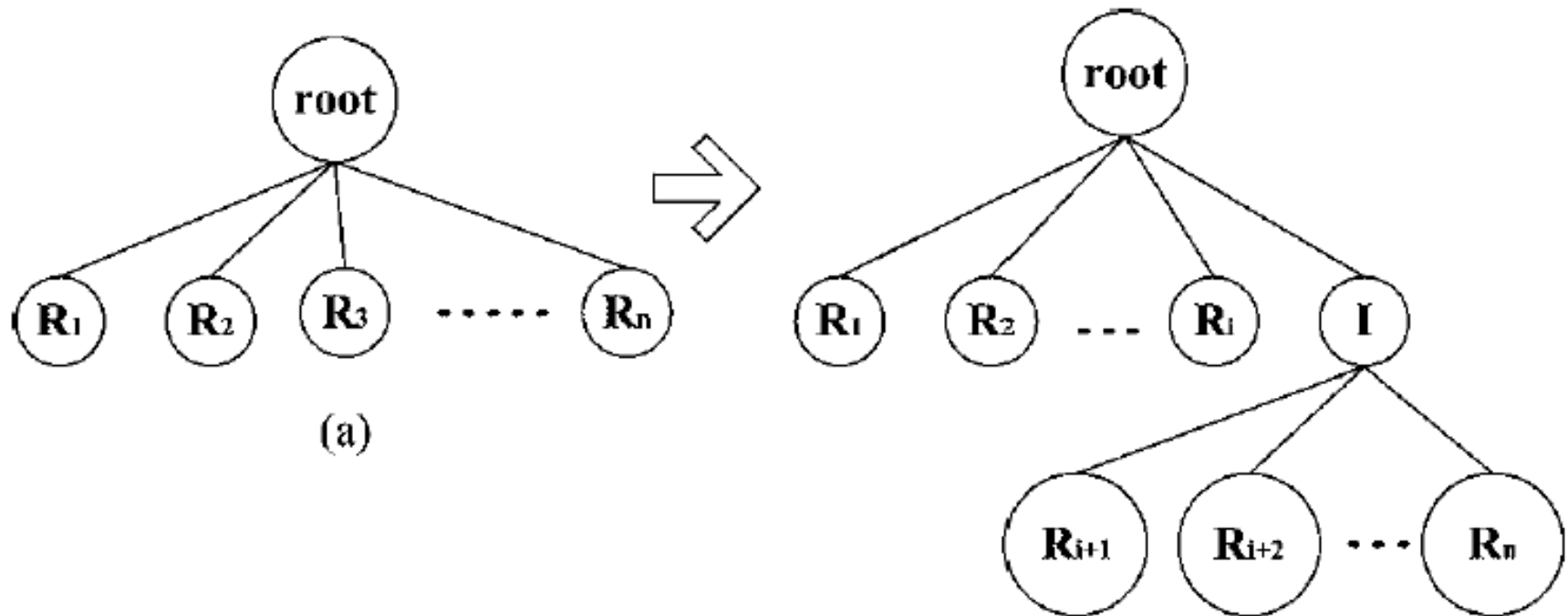
$$= \sum_{i=1}^K d_i \sum_{j=t_{i-1}+1}^{t_i} P_r(R_j)$$

$$= \sum_{i=1}^K \left( \sum_{q=1}^{N_i} \frac{N_i - q}{N_i} \right) \sum_{j=t_{i-1}+1}^{t_i} P_r(R_j),$$

where  $t_i = \sum_{h=1}^i N_h$  and  $t_0 = 0$ .

Πρόβλημα  
Δημιουργίας  
Ιεραρχικού  
Προγράμματος  
Εκπομπής

# Δενδρική αναπαράσταση



# Κόστος επιπέδου του δένδρου

**Definition 1.** Suppose that level  $v$  in the allocation tree has  $j - i + 1$  data nodes,  $R_i, R_{i+1}, \dots, R_j$ . The cost of level  $v$  is defined as

$$C_{i,j} = \sum_{k=1}^{j-i+1} \frac{(j-i+1) - k}{j-i+1} \sum_{q=i}^j P_r(R_q),$$

which is equal to  $w_v \cdot P_{L_v}$ , where  $w_v$  and  $P_{L_v}$  are, respectively, the weight of leaf nodes and the aggregate access frequency for the leaf nodes in level  $v$  of the allocation tree.

**Definition 2.** Suppose that node  $R$  has  $j - i + 1$  child data nodes,  $R_i, R_{i+1}, \dots, R_j$ , which are sorted according to the descending order of  $P_r(R_q)$ ,  $i \leq q \leq j$ , i.e.,  $P_r(R_q) \geq P_r(R_y)$  iff  $q \leq y$ . The reduction gain achieved by grouping nodes  $R_{p+1}, R_{p+2}, \dots, R_j$  and attaching them under a new child node, denoted by  $\delta(p)$ , can be formulated as  $\delta(p) = C_{i,j} - (C_{i,p} + C_{p+1,j})$ .



# Αλγόριθμος VF<sup>K</sup>

**Input.** Assume that  $R_1, \dots, R_n$  have been sorted according to the descending order of  $P_r(R_j)$ ,  $1 \leq j \leq n$ , i.e.,  $P_r(R_q) \geq P_r(R_y)$  iff  $q \leq y$ .  $K$  is the number of broadcast disks in a broadcast disk array.

**Output.** The resulting allocation tree.

**begin**

1. Create table  $AT$  with  $K$  rows;
2.  $AT(1).B = 1$ ; /\*  $AT(1).B$  records the beginning of level 1 \*/
3.  $AT(1).E = n$ ; /\*  $AT(1).E$  records the end of level 1 \*/
4.  $AT(1).LC = C_{1,n}$ ; /\*  $AT(1).LC$  records the cost of level 1 \*/
5. **for** each row  $i$  in table  $AT$  and  $i \geq 2$
6. **begin**
7.      $AT(i).B = 0$ ; /\*  $AT(i).B$  records the beginning of level  $i$  \*/
8.      $AT(i).E = 0$ ; /\*  $AT(i).E$  records the end of level  $i$  \*/
9.      $AT(i).LC = 0$ ; /\*  $AT(i).LC$  records the cost of level  $i$  \*/
10. **end**
11.  $pivot = 1$ ;

# Αλγόριθμος VFK

```

12. repeat
13.   begin
14.     Choose row  $i$  from table  $AT$  such that  $AT(i).LC$  is maximal
        among all unmarked rows;
15.     if ( $i == 1$  or  $i == pivot$ ) /* Upward and downward partition */
16.       begin
17.          $j = Partition(R_{AT(i).B}, R_{AT(i).B+1}, \dots, R_{AT(i).E});$ 
18.         {Update table  $AT$  accordingly and unmark all rows;
19.          $pivot ++;$ }
20.       end
21.     else /* Middle partition */
22.       begin
23.          $j = Partition(R_{AT(i).B}, R_{AT(i).B+1}, \dots, R_{AT(i).E});$ 
24.         if ( $AT(i-1).E - AT(i-1).B < (j - AT(i).B)$ 
25.           {Update table  $AT$  accordingly and unmark all rows;
26.            $pivot ++;$ }
27.         else {
28.           Mark row  $i$ ;
29.           Merge ( $R_{AT(i).B}, R_{AT(i).B+1}, \dots, R_j$ ) and
                ( $R_{j+1}, R_{j+2}, \dots, R_{AT(i).E}$ ) together;}
30.         end
31.       end
32.     until ( $pivot == K$ )
end

```

# Αλγόριθμος VFK

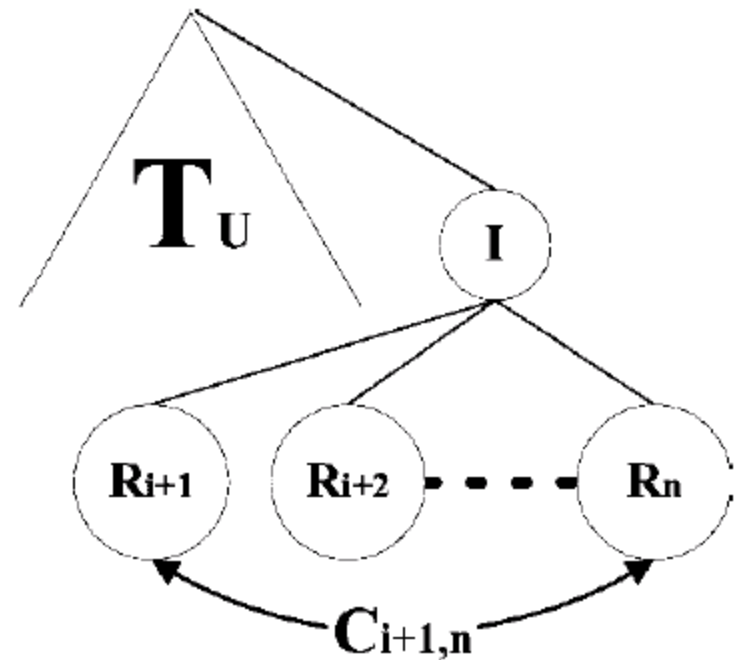
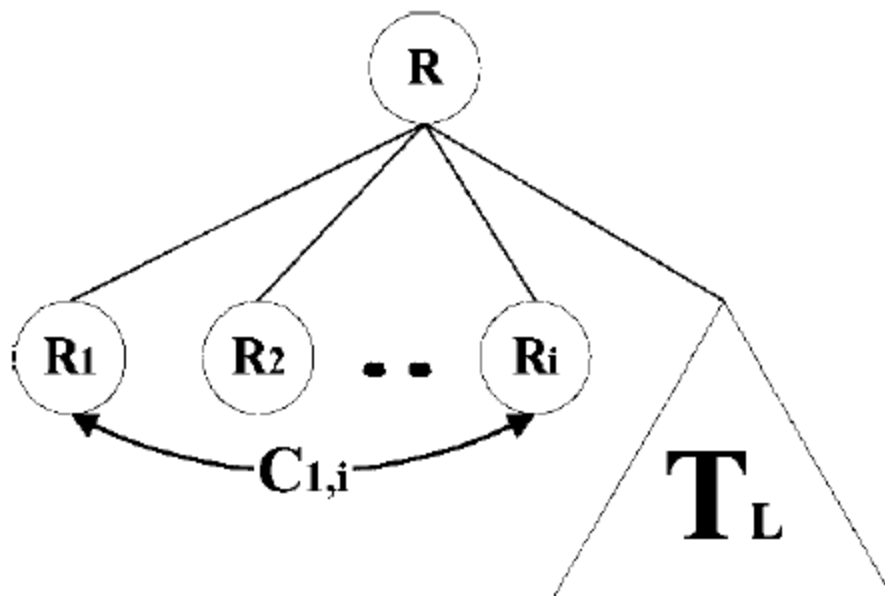
**Procedure** *Partition*( $R_i, R_{i+1}, \dots, R_j$ ).

1. Determine  $p^*$  such that

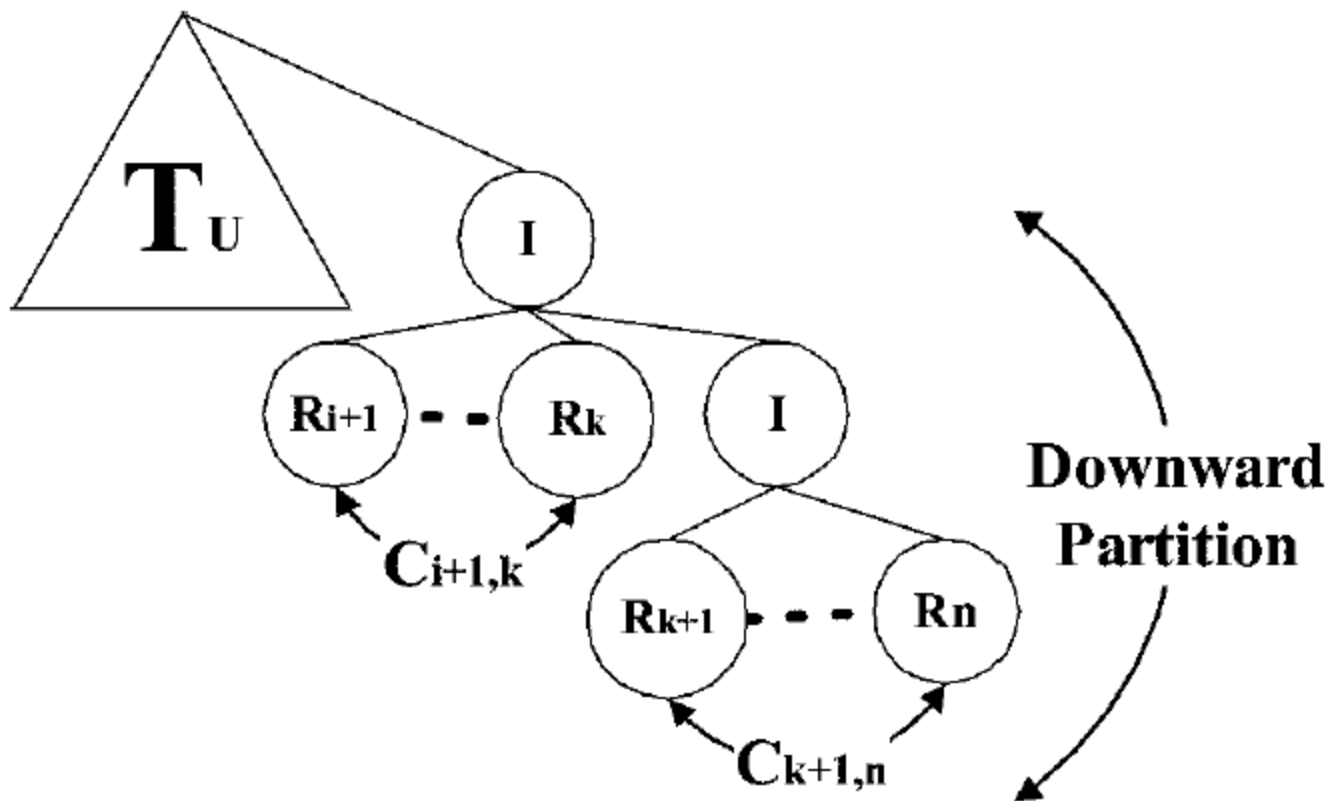
$$\delta(p^*) = \max_{\forall p \in \{i, i+(j-i+1)/2-1\}} \{\delta(p)\}.$$

2. Attach nodes  $R_{p^*+1}, R_{p^*+2}, \dots, R_j$  under a new node  $I$  in the tree.
3. Return  $p^*$ .

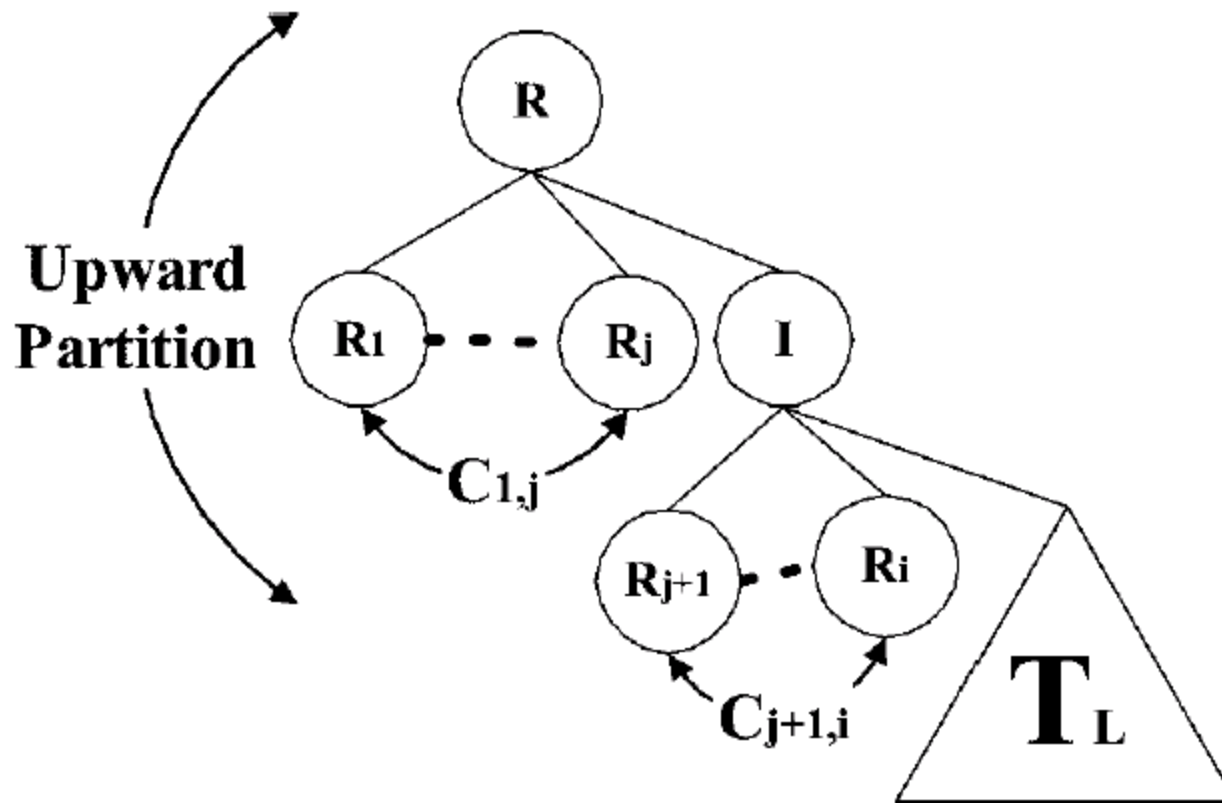
# Αλγόριθμος $VF^K$



# Αλγόριθμος $VF^K$



# Αλγόριθμος $VF^K$



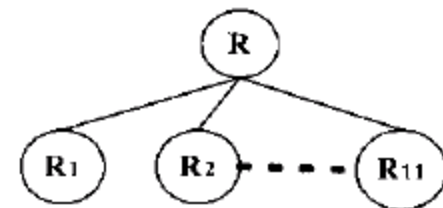
# Παράδειγμα του VF<sup>K</sup>

Data record	$R_1$	$R_2$	$R_3$	$R_4$	$R_5$	$R_6$	$R_7$	$R_8$	$R_9$	$R_{10}$	$R_{11}$
$P_r(R_i)$	0.237	0.211	0.132	0.132	0.08	0.05	0.05	0.027	0.027	0.027	0.027

(a) Partition  $(R_1, R_2, \dots, R_{11})$  is selected and decomposed to  $(R_1, \dots, R_4)$  and  $(R_5, \dots, R_{11})$ .

Level $i$	$AT(i).B$	$AT(i).E$	$AT(i).LC$
1	1	11	5*
2	0	0	0
3	0	0	0
4	0	0	0

$p$	1	2	3	4	5
$C_{1,11}$	5	5	5	5	5
$C_{1,p} + C_{p+1,11}$	3.4335	2.484	2.05	1.932	2.104
$\delta(p)$	1.5665	2.516	2.95	3.068*	2.896



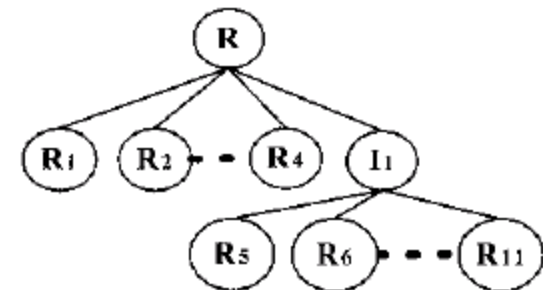
# Παράδειγμα του VF<sup>K</sup>

(b) Partition  $(R_1, R_2, \dots, R_4)$  is selected and decomposed to  $(R_1, R_2)$  and  $(R_3, R_4)$ .

Level $i$	$AT(i).B$	$AT(i).E$	$AT(i).LC$
1	1	4	1.068*
2	5	11	0.864
3	0	0	0
4	0	0	0

$p$	1	2
$C_{1,4}$	1.068	1.068
$C_{1,p} + C_{p+1,4}$	0.475	0.356
$\delta(p)$	0.593	0.712*





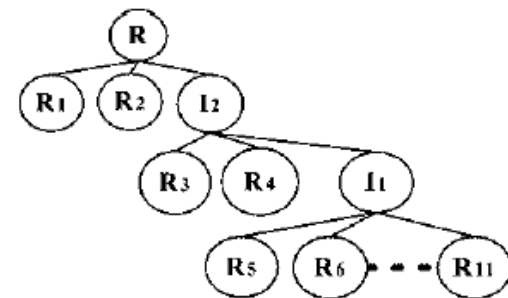
# Παράδειγμα του VF<sup>K</sup>

(c) Partition  $(R_5, R_6, \dots, R_{11})$  is selected and decomposed to  $(R_5, \dots, R_7)$  and  $(R_8, \dots, R_{11})$ .

Level	$AT(i).B$	$AT(i).E$	$AT(i).LC$
1	1	2	0.224
2	3	4	0.132
3	5	11	0.864*
4	0	0	0

$p$	5	6	7
$C_{5,11}$	0.864	0.864	0.864
$C_{5,p} + C_{p+1,11}$	0.52	0.381	0.342
$\delta(p)$	0.344	0.483	0.522*



# Παράδειγμα του VFK

(d) The final result of table  $AT$ .

Level $i$	$AT(i).B$	$AT(i).E$	$AT(i).LC$
1	1	2	0.224
2	3	4	0.132
3	5	7	0.18
4	8	11	0.162

