Κινητός και Διάχυτος Υπολογισμός (Mobile & Pervasive Computing)

Δημήτριος Κατσαρός

Χειμώνας 2015

Διάλεξη 2η

Περιεχόμενα

- Αρχιτεκτονική κινητού δικτύου
- Ασύμμετρο περιβάλλον επικοινωνίας χωρίς Ανοδικό Κανάλι
- Δίσκοι Εκπομπής (Broadcast Disks)
- **Αλγόριθμοι για Καθαρή Εκπομπή (Pure Broadcast)**

Γενίκευση των Δίσκων Εκπομπής

- Αίρουμε την υπόθεση ότι όλα τα αντικείμενα έχουν το ίδιο μέγεθος
- Εξακολουθούμε να επιδιώκουμε το γεγονός ότι οι διαδοχικές εκπομπές του ίδιου αντικειμένου είναι ίσες, διότι:

  ΛΗΜΜΑ. Το πρόγραμμα εκπομπής με την *ελάχιστη συνολική μέση καθυστέρηση* είναι εκείνο στο οποίο οι εμφανίσεις κάθε αντικειμένου απέχουν πάντα το ίδιο (**equally spaced criterion**)
- Αναζητούμε πόσο συχνά πρέπει να εκπέμπεται κάθε αντικείμενο

Ο κανόνας "Τετράγωνο της Ρίζας"

- **ΘΕΩΡΗΜΑ**. Υποθέτοντας ότι οι εμφανίσεις κάθε αντικειμένου είναι equally spaced, η *ελάχιστη συνολική μέση καθυστέρηση* επιτυγχάνεται όταν η απόσταση s_i μεταξύ διαδοχικών εμφανίσεων του αντικειμένου i είναι ανάλογη προς την τετραγωνική ρίζα του μήκους του και αντιστρόφως ανάλογη προς την τετραγωνική ρίζα της πιθανότητας προσπέλασής του. Δηλαδή

$$s_i \propto \sqrt{l_i / p_i}$$

Στην περίπτωση αυτή, η *ελάχιστη συνολική μέση καθυστέρηση* είναι ίση με:

$$\frac{1}{2}\left(\sum_{i=1}^{M}\sqrt{p_i l_i}\right)^2$$

Απόδειξη του κανόνα Square-root

- Έστω ότι s_i είναι το spacing για το αντικείμενο i.
- Η *μέση καθυστέρηση πρόσβασης* στο i είναι $t_i=s_i/2$
- Η συνολική μέση καθυστέρηση πρόσβασης για όλα τα αντικείμενα είναι

$$t = \frac{1}{2}\sum_{i=1}^{N} p_i * s_i$$

- Έστω ότι $r_i = l_i / s_i$, δηλ. είναι το κλάσμα του εύρους ζώνης που ανατίθεται στο αντικείμενο i, λόγου του κριτηρίου equal-spacing.
- Επομένως: $$\sum_{i=1}^{N} r_i = 1$$
- Άρα: $$t = \frac{1}{2}\sum_{i=1}^{N} \frac{p_i * l_i}{r_i}$$

Απόδειξη του κανόνα Square-root

- Αφού $\sum_{i=1}^{N} r_i = 1$
- Μόνο N-1 από τα r_i μπορούν να αλλαχτούν ανεξάρτητα. Η βέλτιστη τιμή για τα r_i επιτυγχάνεται εάν $\partial t/\partial r_i = 0$, για κάθε i. Επιλύουμε τις εξισώσεις αυτές, ξεκινώντας με $0 = \partial t/\partial r_1$.

$$
\begin{aligned}
0 = \frac{\partial t}{\partial r_1} &= \frac{1}{2}\frac{\partial}{\partial r_1}\left(\sum_{i=1}^{M}\frac{p_i l_i}{r_i}\right) \\
&= \frac{1}{2}\frac{\partial}{\partial r_1}\left(\frac{p_1 l_1}{r_1} + \sum_{i=2}^{M-1}\frac{p_i l_i}{r_i} + \frac{p_M l_M}{(1-\sum_{i=1}^{M-1} r_i)}\right) \\
&= \frac{1}{2}\left(-\frac{p_1 l_1}{r_1^2} + \frac{p_M l_M}{(1-\sum_{i=1}^{M-1} r_i)^2}\right) \\
&\Rightarrow \frac{p_1 l_1}{r_1^2} = \frac{p_M l_M}{(1-\sum_{i=1}^{M-1} r_i)^2}.
\end{aligned}
$$

Απόδειξη του κανόνα Square-root

- Όμοια ισχύει ότι: $$\frac{p_2 l_2}{r_2^2} = \frac{p_M l_M}{\left(1 - \sum_{i=1}^{M-1} r_i\right)^2}$$

- Από αυτές τις εξισώσεις προκύπτει ότι:

$$\frac{p_1 l_1}{r_1^2} = \frac{p_2 l_2}{r_2^2} \Rightarrow \frac{r_1}{r_2} = \sqrt{\frac{p_1 l_1}{p_2 l_2}}$$

- Και γενικά ότι:

$$\frac{r_i}{r_j} = \sqrt{\frac{p_i l_i}{p_j l_j}}, \quad \forall i, j$$

Απόδειξη του κανόνα Square-root

- Αυτό σημαίνει ότι η βέλτιστη τιμή του r_i πρέπει να είναι γραμμικά ανάλογη προς το: $\sqrt{l_i * p_i}$

- Εύκολα διαπιστώνουμε ότι υπάρχει η σταθερά αναλογίας α ίση με : $a = 1/\sum_{j=1}^{N} \sqrt{l_j * p_j}$

- ώστε $r_i = a * \sqrt{l_i * p_i}$ είναι η μόνη δυνατή λύση για τις εξισώσεις $\partial t/\partial r_i = 0$

- Με αντικατάσταση των r_i στη σχέση που δίνει το t βρίσκουμε τη βέλτιστη τιμή του.

Αλγόριθμος με βάση τον κανόνα S-r

- Το Θεώρημα πρακτικά υπονοεί ότι για να επιτύχουμε τη βέλτιστη επίδοση (ελάχιστη συνολική μέση καθυστέρηση), πρέπει να ισχύει για το spacing s_i κάθε αντικειμένου i ότι:

$$\frac{s_i^2 p_i}{l_i} = \text{constant}$$

- Ορίζοντας για κάθε αντικείμενο j τη συνάρτηση:

$$G(j) = (Q - R(j))^2 * p_j / l_j$$

- όπου Q είναι η τρέχουσα χρονική στιγμή και R(j) η στιγμή τελευταίας εκπομπής του αντικειμένου j, έχουμε τον επόμενο αλγόριθμο εκπομπής:

Αλγόρ. εκπομπής Vaidya-Hameed

Broadcast scheduling algorithm A

Step 1. Determine maximum value of $G(j)$ over all items j, $1 \leqslant j \leqslant M$. Let $G_{\max}$ denote the maximum value of $G(j)$.

Step 2. Choose item i such that $G(i) = G_{\max}$. If this equality holds for more than one item, choose any one of them arbitrarily.

Step 3. Broadcast item i at time Q.

Step 4. $R(i) = Q$.

Παράδειγμα Vaidya-Hammed

Example 1. Consider a database containing 3 items such that $p_1 = 1/2$, $p_2 = 3/8$, and $p_3 = 1/8$. Assume that items have lengths $l_1 = 1$, $l_2 = 2$ and $l_3 = 4$ time units. Figure 1 shows the items recently broadcast by the server (up to time < 100). The above algorithm is called to determine the item to be transmitted at time 100. Thus, $Q = 100$. Also, from figure 1, observe that $R(1) = 95$, $R(2) = 93$, and $R(3) = 96$. The algorithm evaluates function $G(j) = (Q - R(j))^2 p_j / l_j$ for $j = 1, 2, 3$ as 12.5, 147/16 (= 9.1875) and 0.5, respectively. As $G(j)$ is the largest for $j = 1$, item 1 is transmitted at time 100.

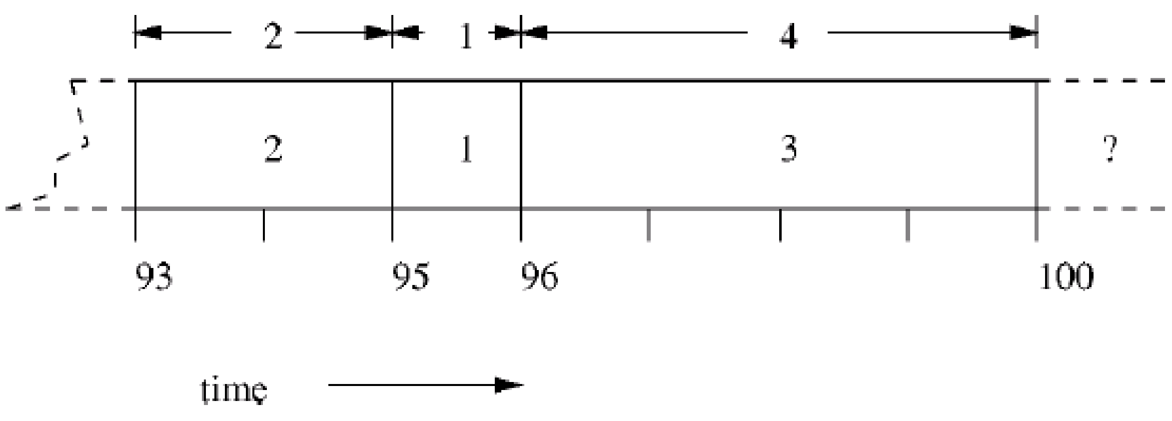


Figure 1. Example 1.

Κινητός και Διάχυτος Υπολογισμός
(Mobile & Pervasive Computing)

Δημήτριος Κατσαρός

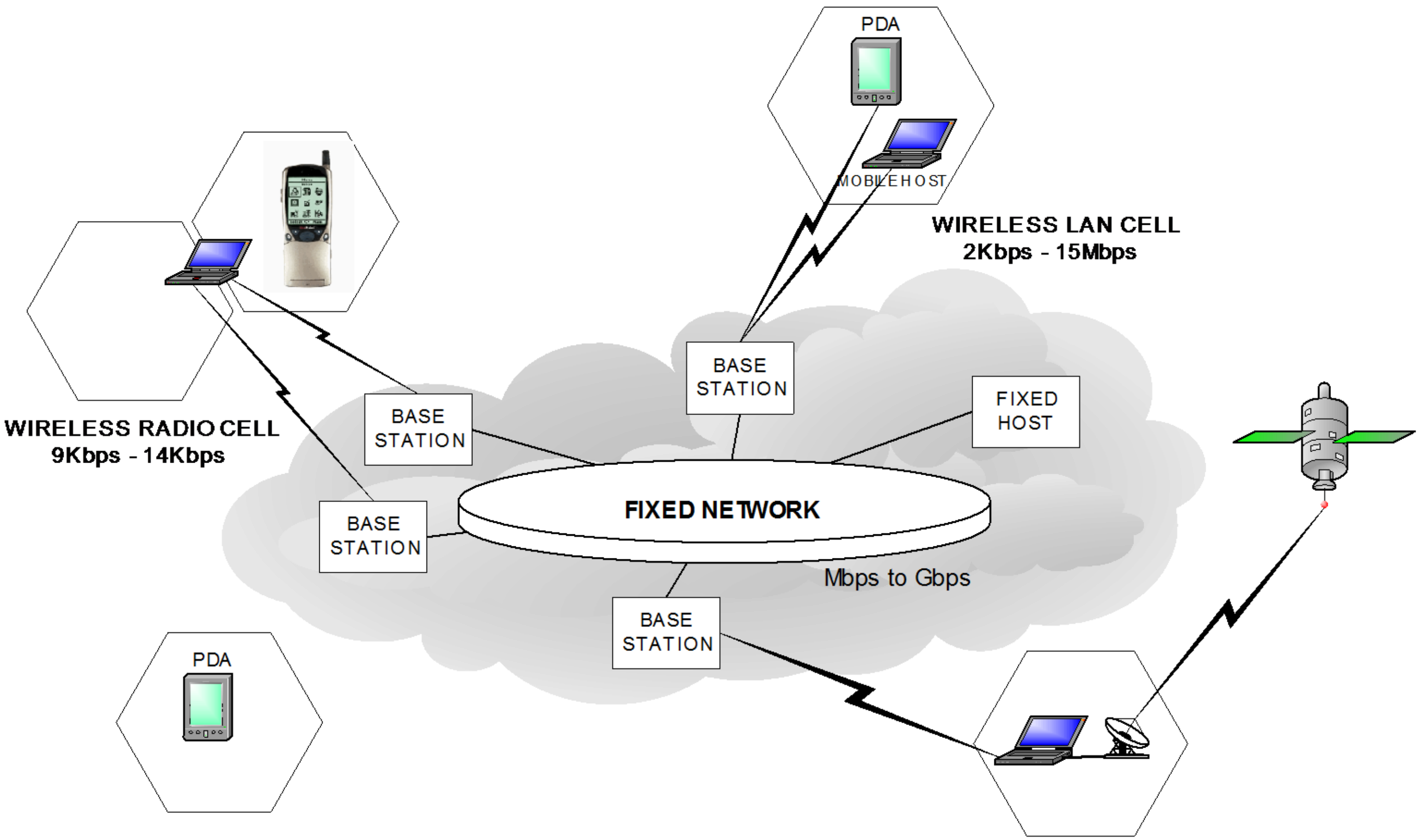
Χειμώνας 2015

Διάλεξη 3η

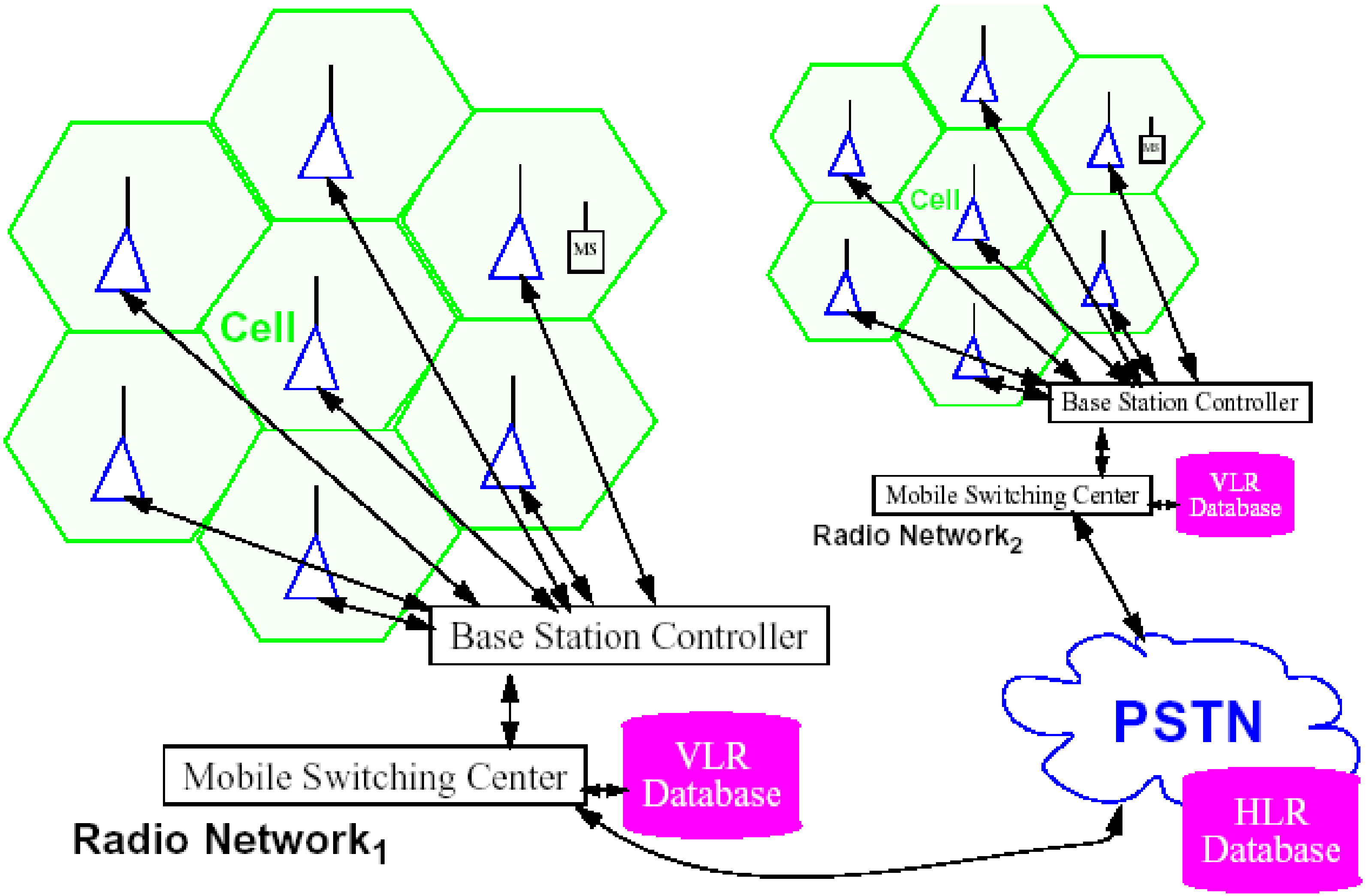
Περιεχόμενα

- **Αρχιτεκτονική κινητού δικτύου**
- Ασύμμετρο περιβάλλον επικοινωνίας με Ανοδικό Κανάλι
- Αλγόριθμοι για Κατ' Απαίτηση Εκπομπή (On-Demand Broadcast)

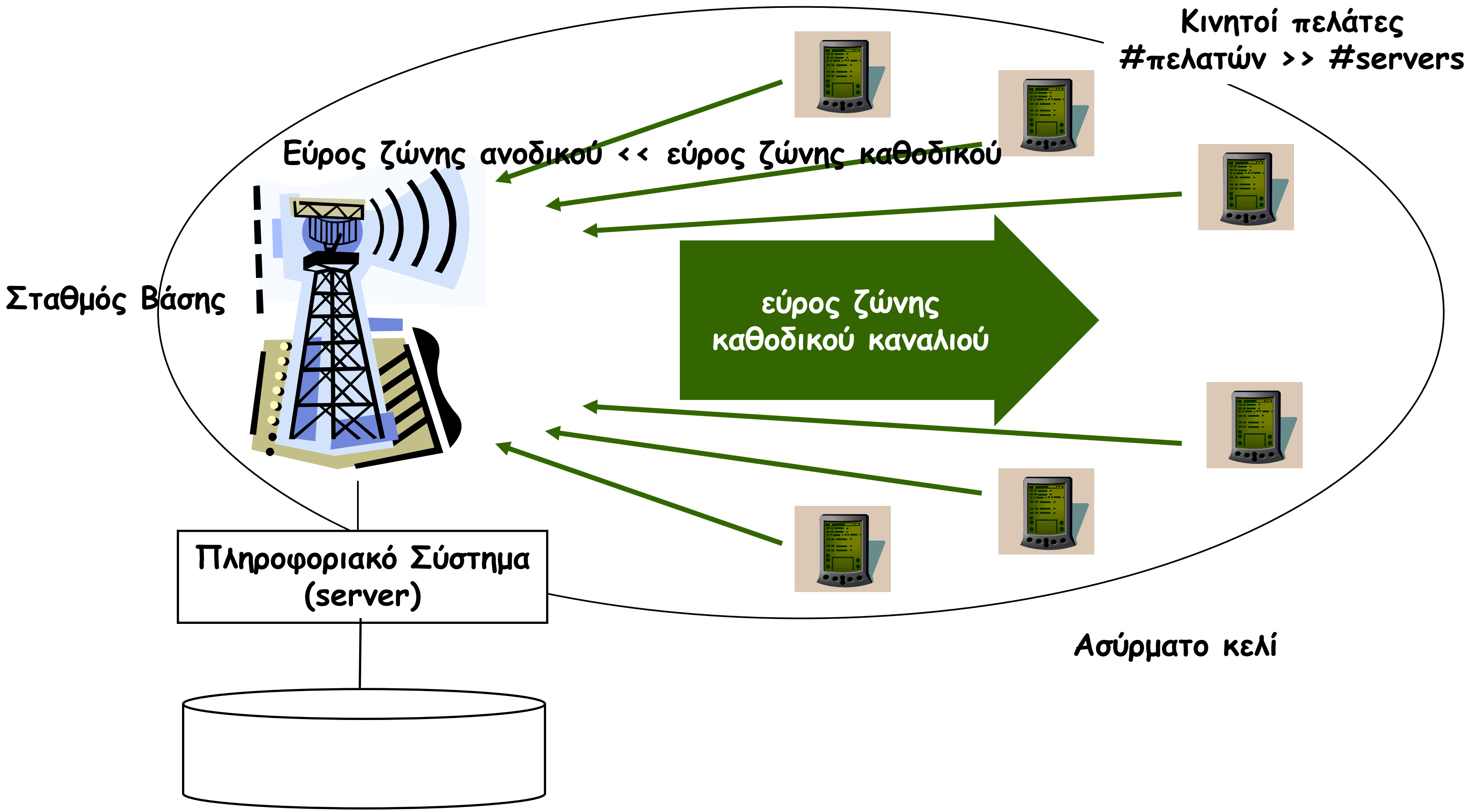
Αρχιτεκτονική κινητού δικτύου



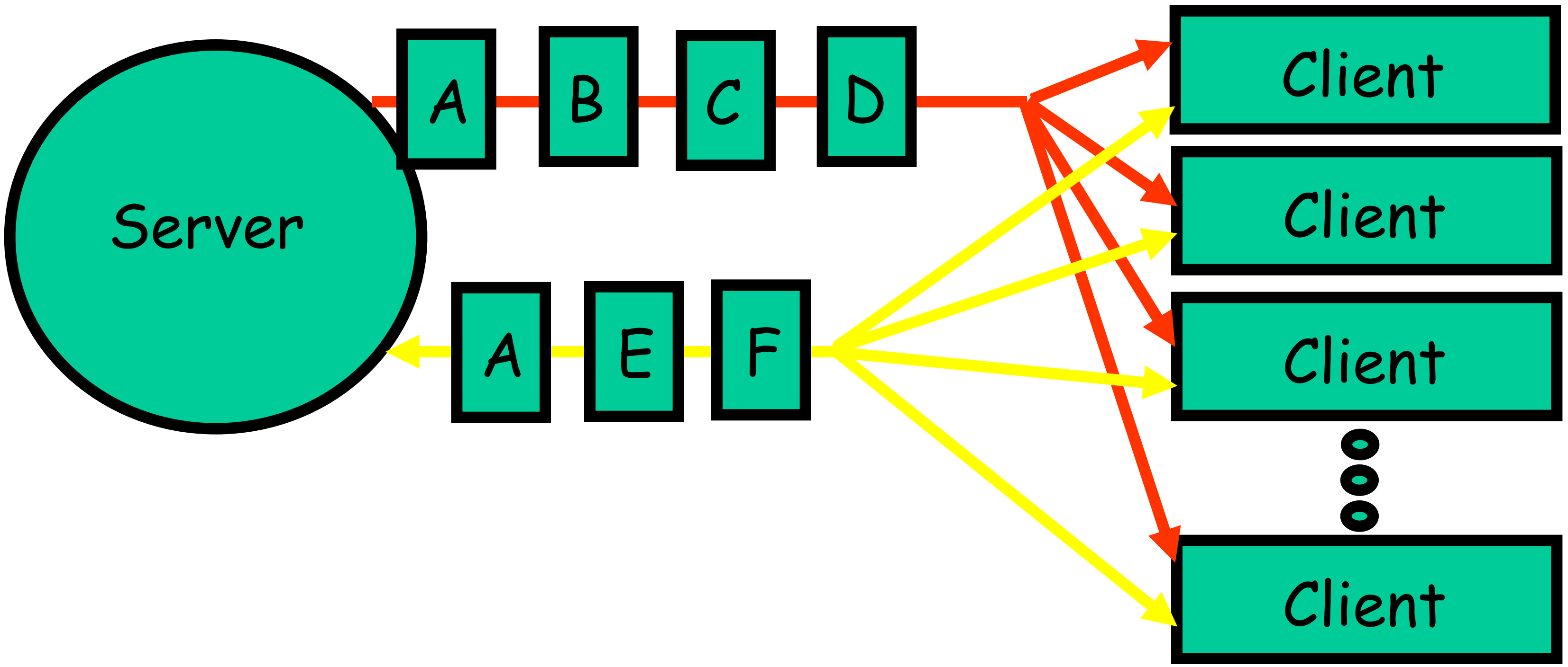
Αρχιτ. Personal Comm. Sys. (PCS)



Γενικό μοντέλο εκπομπής



Μοντέλο Κατ' Απαίτηση Εκπομπής



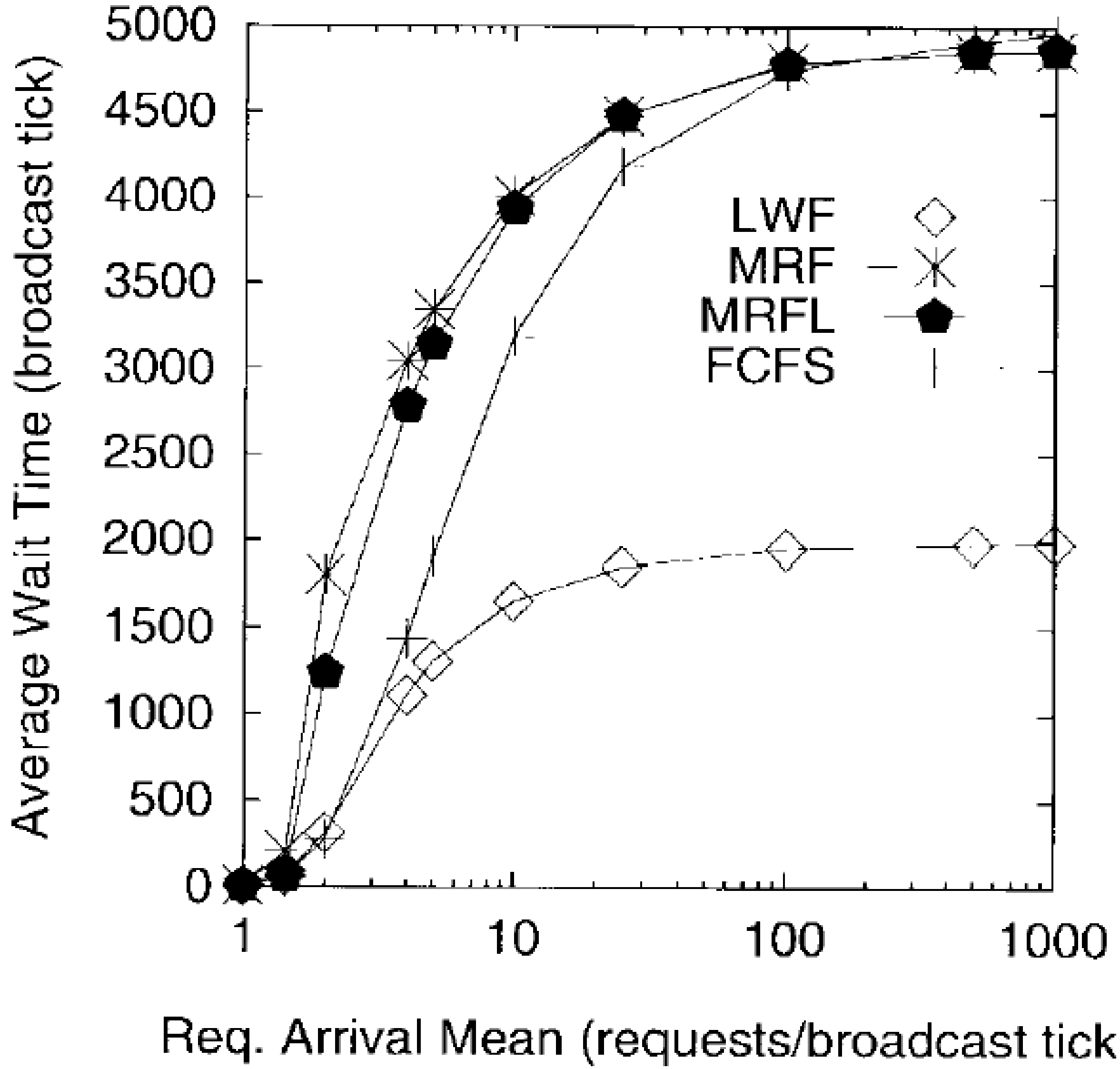
Απαιτήσεις

- **Γρήγορη απόκριση** (responsiveness)
  - Μικρή καθυστέρηση (latency) των χρηστών
    - Όχι **starvation**
    - Κόστος επιλογής του αντικειμένου προς εκπομπή
- **Ικανότητα κλιμάκωσης** (scalability)
  - Ρυθμό άφιξης των αιτήσεων
  - Μέγεθος βάσης δεδομένων
  - Ρυθμό εκπομπής
- **Ευρωστία**
  - Αλλαγές στο πρότυπο προσπέλασης

Υπάρχοντες αλγόριθμοι

- **FCFS** (First-Come-First-Served)
  - Εκπέμπει τα αντικείμενα με τη σειρά που έρχονται οι αιτήσεις γι’ αυτά. Εάν υπάρχει ήδη στην ουρά δεν προστίθεται νέα είσοδος (entry) στην ουρά
- **MRF** (Most Requests First)
  - Εκπέμπει το αντικείμενο με τις περισσότερες εκκρεμείς αιτήσεις
- **MRFL** (Most Requests First Lowest)
  - Όπως και ο MRF, αλλά σπάει τις ισοπαλίες προς χάριν του αντικειμένου που δεν έχει εκπεμφθεί για το μεγαλύτερο διάστημα
- **LWF** (Longest Wait First)
  - Εκπέμπει το αντικείμενο για το οποίο ο συνολικός χρόνος αναμονής όλων των αιτήσεων γι’ αυτό είναι ο μεγαλύτερος (δηλ. το άθροισμα του χρόνου αναμονής στην ουρά όλων των αιτήσεων γι’αυτό

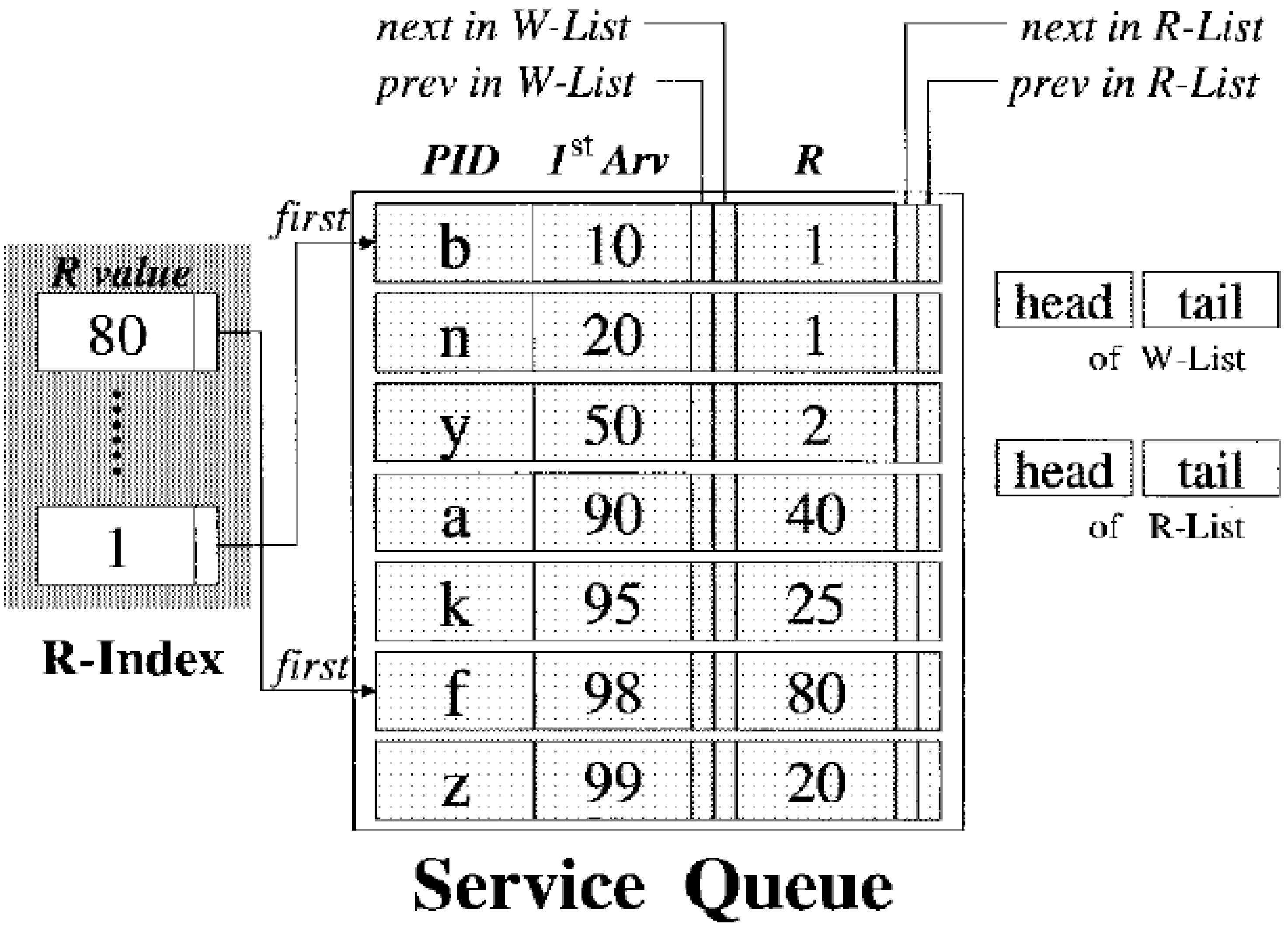
Επίδοση υπαρχόντων αλγορίθμων



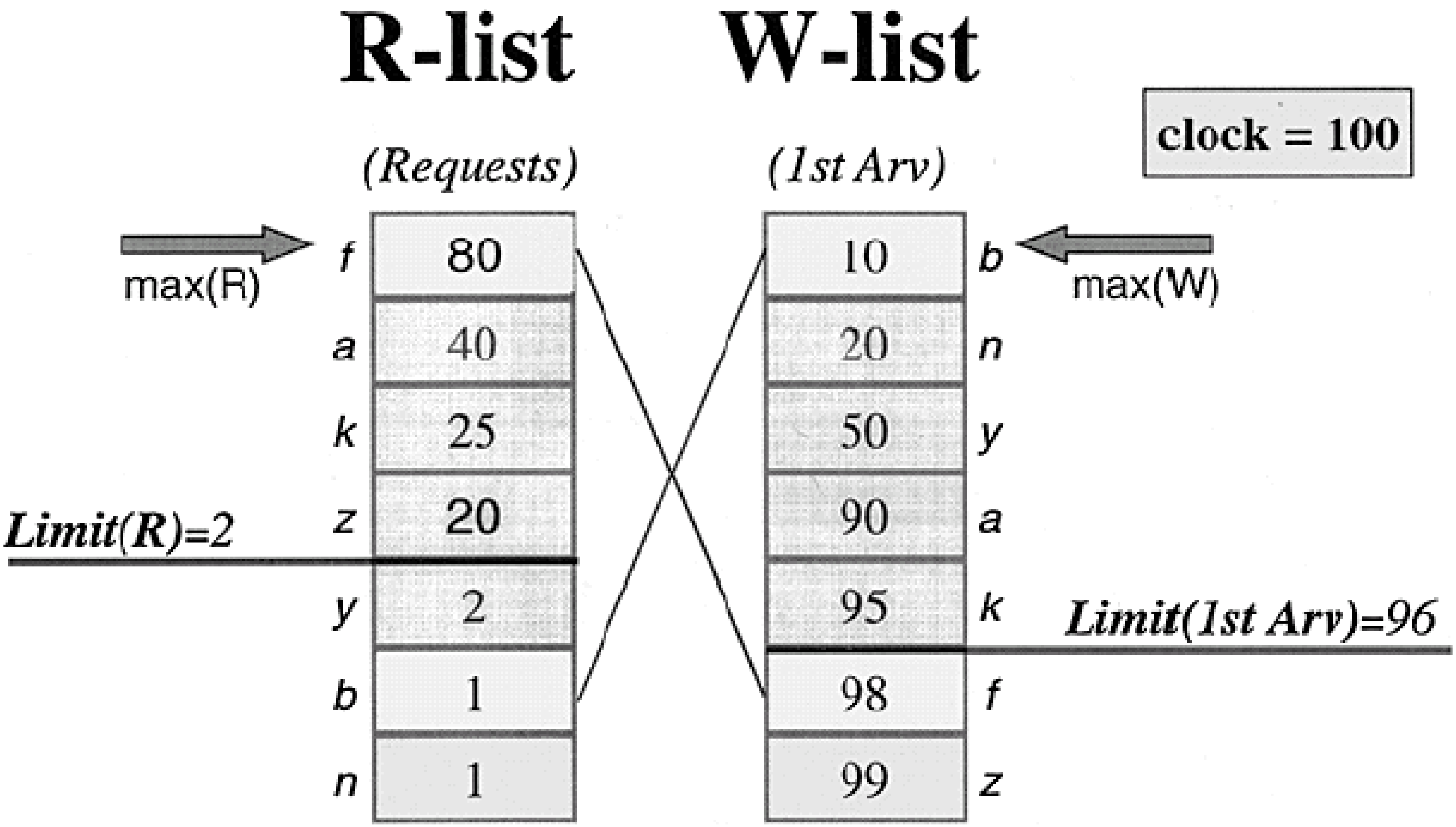
RxW: Συνδυασμός MRF & FCFS

- Εκπέμπει τη σελίδα με το μεγαλύτερο γινόμενο RxW
  - R: είναι ο αριθμός των εκκρεμών αιτήσεων για τη σελίδα
  - W: είναι ο μέγιστος χρόνος αναμονής μέσα στην ουρά, δηλ. η αίτηση πιο παλιά στο χρόνο
- Εκπέμπει μια σελίδα είτε επειδή είναι δημοφιλής είτε επειδή δεν έχει εκπεμφθεί για αρκετό χρονικό διάστημα

Η ουρά εξυπηρέτησης του RxW



Βελτίωση στην αναζήτηση του RxW



Βελτίωση στην αναζήτηση του RxW

- Εξέτασε τη σελίδα στην κορυφή της R-list
- Θέσε MAX = R x W αυτής της σελίδας
- Οι υπόλοιπες τιμές W μπορούν να περιοριστούν αξιοποιώντας την τιμή R’, δηλ., την R τιμή της επόμενης στη R-list σελίδας.
- Για να έχει μια άλλη σελίδα RxW μεγαλύτερο από το τρέχον MAX, πρέπει το αντίστοιχο W να ικανοποιεί την ανίσωση W > MAX/R’
- Έτσι θέτουμε ένα όριο στην 1stARV ως εξής

  limit(1stARV) = clock – MAX/R’
- Κατόπιν εξετάζουμε τη σελίδα στην κορυφή της W-list και κάνουμε ανάλογες ενέργειες περιορίζοντας το εύρος αναζήτησης στη R-list

Παράδειγμα βελτιωμένου RxW

The R-List and the W-List are shown as two separate lists and the current clock value is 100 ticks.

First, the entry for page *f* (the top of the R-List) is examined resulting in MAX being set to 160 and *limit(1stARV)* being set to 96.

Next, the entry for page *b* (the top of the W-List) is checked. RxW of b is less than MAX (90 versus 160) so MAX is left unchanged, but *limit(R)* is set to 2.

The algorithm then checks page *α*, which has an RxW value of 400, and so MAX is updated to 400, and *limit(1stARV)* is set to 84. The algorithm continues searching until page *y* is examined, at which point the limit on the -List is reached and the algorithm stops. In this example, page *a* has the highest value, so it is chosen to be broadcast.