Online Frequency Assignment in Wireless Communication Networks

Francis Y.L. Chin
Taikoo Chair of Engineering
Chair Professor of Computer Science
University of Hong Kong

Joint work with Dr WT Chan, Dr Deshi Ye and Dr Yong Zhang

A presentation especially for COCOON 2007, Banff

This is Hong Kong!

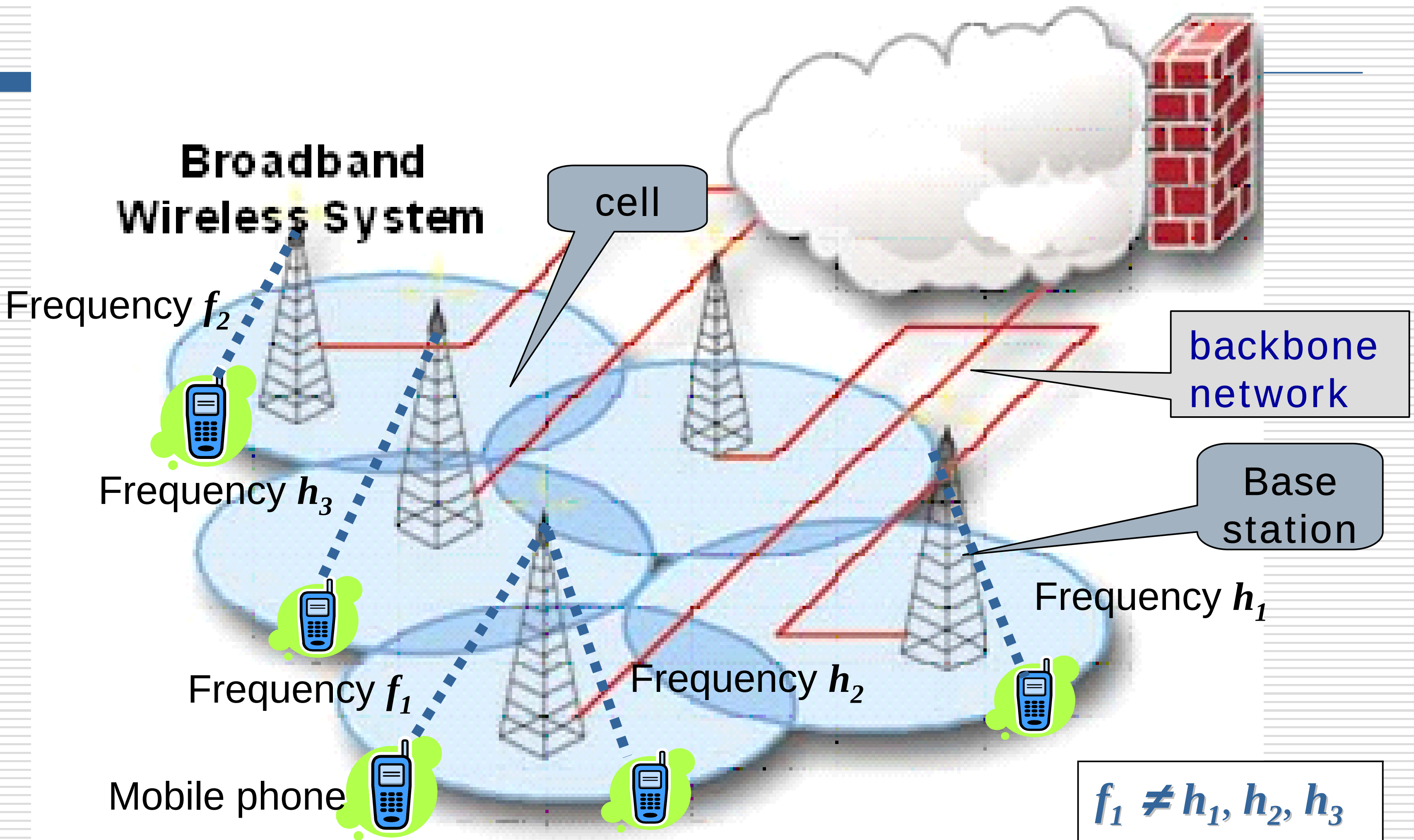
Frequency Assignment Problem (FAP) Outline

- Background
- FAP without reassignments
  - Absolute competitive ratio
  - Asymptotic competitive ratio
  - With deletion
- FAP with reassignments
- FAP with OVSF code

Frequency Assignment Problem (FAP)

- Wireless communication has a long history
- Frequency bandwidth is always a scarce resource and FAP has always been an important problem
- New problems come up because of
  - Change of technology
    - Analog to digital
    - MTS, AMPS, GSM, FDMA, CDMA
  - New algorithmic techniques
    - Approximation and online analysis

Wireless Communication Network



The same and neighboring cells should use different frequencies

$f_1 \neq h_1, h_2, h_3$

$f_2 \neq h_3$

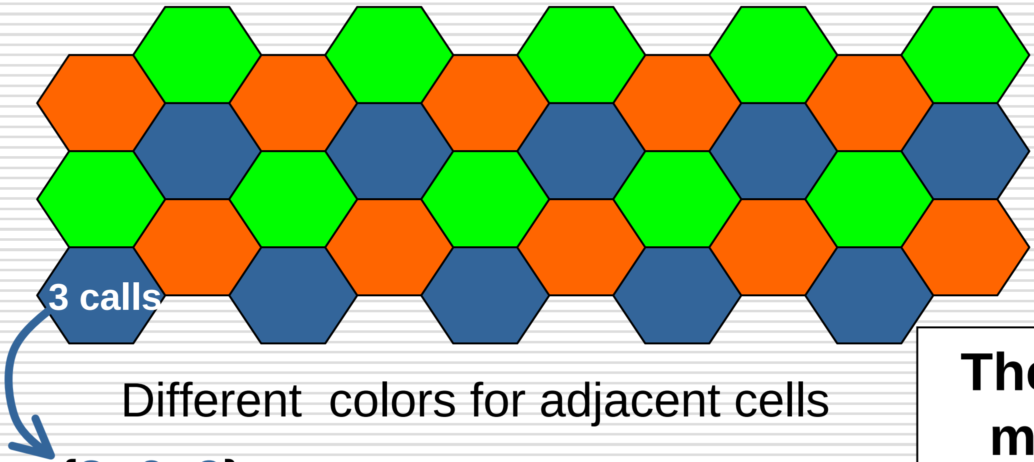
$h_2 \neq h_1, h_3$

Frequency Allocation Problem (FAP)

- Given the hexagon graph
- Frequencies: represented by {1, 2, 3, ...}
- Interference constraint:
  calls from the same or adjacent cells use different frequencies
- Online:
  - calls arrive one at a time
  - assigned frequencies never changed
  - calls never terminate
- Problem:
  minimize the span of frequencies used

Fixed Allocation

- Cells are colored by R, G, and B
- Frequencies are partitioned into 3 disjoint sets
  - $F_R = \{1, 4, 7, \ldots\}$ ← For cells with color R
  - $F_G = \{2, 5, 8, \ldots\}$ ← For cells with color G
  - $F_B = \{3, 6, 9, \ldots\}$ ← For cells with color B



Different colors for adjacent cells

{3, 6, 9}

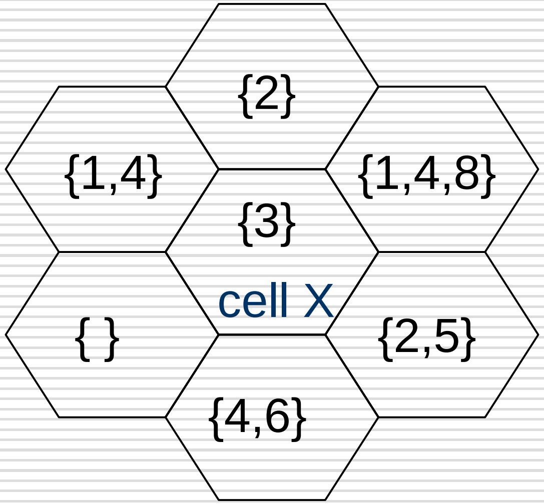
Frequency allocation rule:
Use frequencies according to the color of the cells, starting from the lowest frequency

The span of frequencies used is at most 3 times that of the optimal.
(3-competitive)

Greedy Allocation

Frequency allocation rule:

- Assign the lowest available frequency



E.g., frequency **7** is assigned to call X

The span of frequencies used is at most 17/7(≈2.43) **times that of the optimal.**
[Chan,Chin,Ye,Zhang,Zhu, IPL 2006]

The ratio is tight [Caragiannis, et al, SPAA 2000]

Frequency Assignment Problem (FAP) Outline

- Background
- FAP without reassignments
  - Absolute competitive ratio (Hybrid)
  - Asymptotic competitive ratio
  - With deletion
- FAP with reassignments
- FAP with OVSF code

Hybrid Allocation (Chan et al SPAA2007)

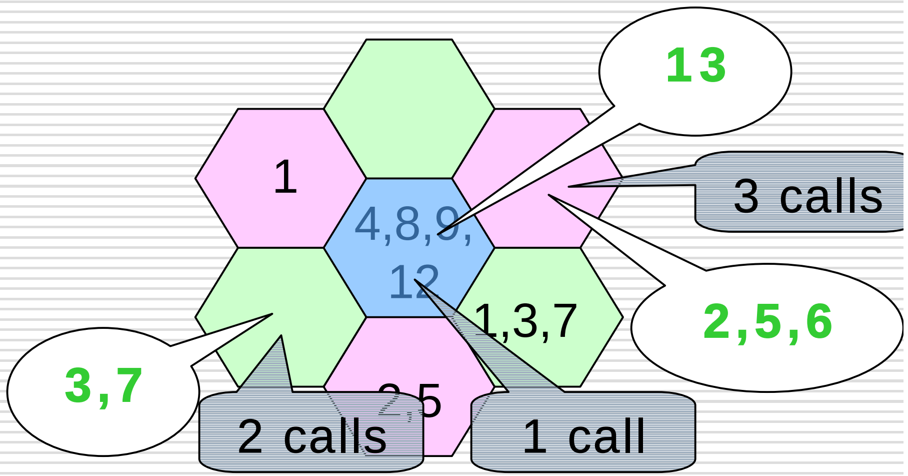
Color the cells by R, G, and B

Partition the frequencies into 4 disjoint sets ← Shared by all cells

- $F_S = \{1, 5, 9, 13\ldots\}$ ← For cells with color R
- $F_R = \{2, 6, 10, \ldots\}$ ← For cells with color G
- $F_G = \{3, 7, 11, \ldots\}$ ← For cells with color B
- $F_B = \{4, 8, 12, \ldots\}$

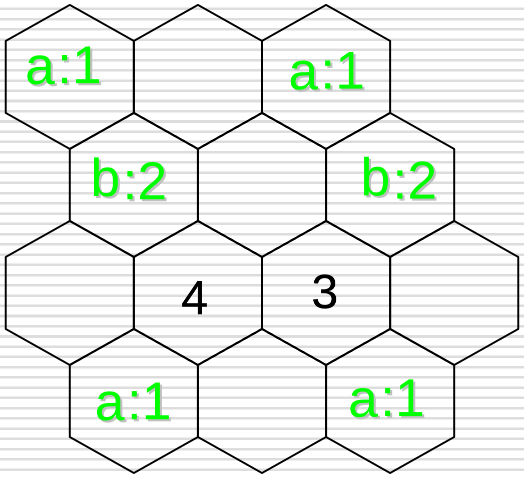
Frequency allocation rule

- Assign the lowest available frequency in $F_S \cup F_x$

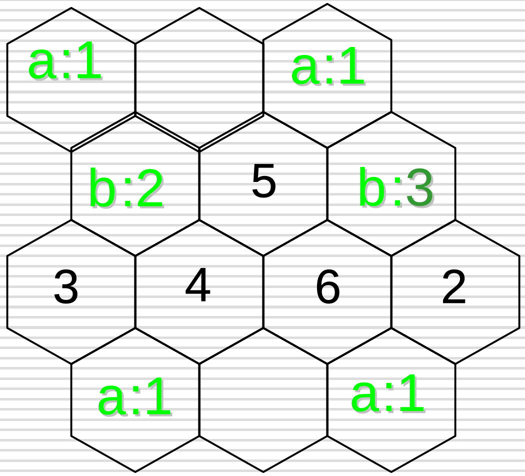


FAP – lower bound=2 (by adversary)

- A call from each cell a
- A call from each cell b



optimal needs 2.



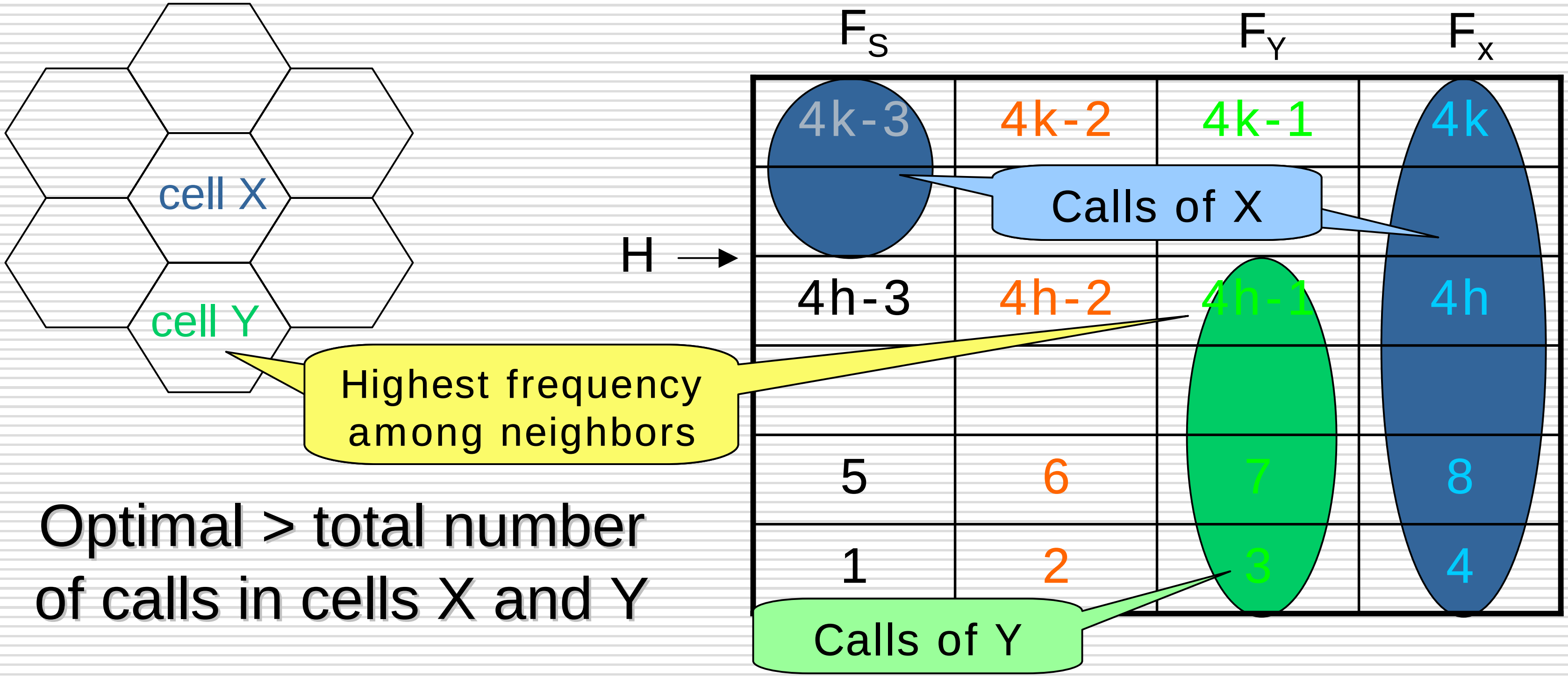
optimal needs 3.

2 is the smallest ratio that we can do for any online algorithm

Proof of the 2-competitive bound

The span of frequencies used is at most <u>twice</u> that of the optimal.

Why? **Suppose cell X has the highest frequency, 4k**



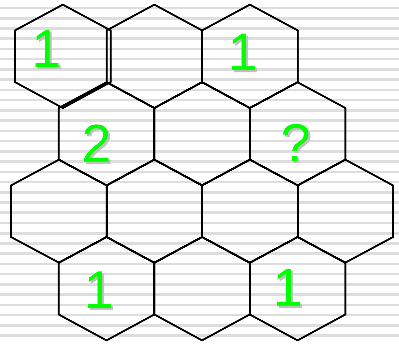
Optimal > total number of calls in cells X and Y

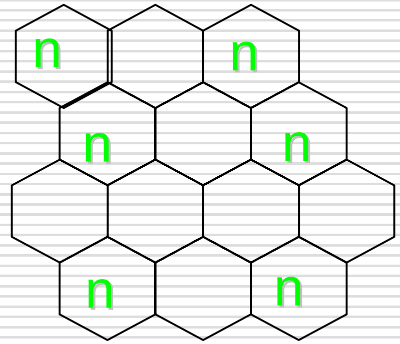
More to be done?

- Hybrid is 2-comptitive
- Lower bound is also 2
- Optimal only when the number of calls is small.
- When the number of calls tends to ∞, we can do better than 2-competitive.

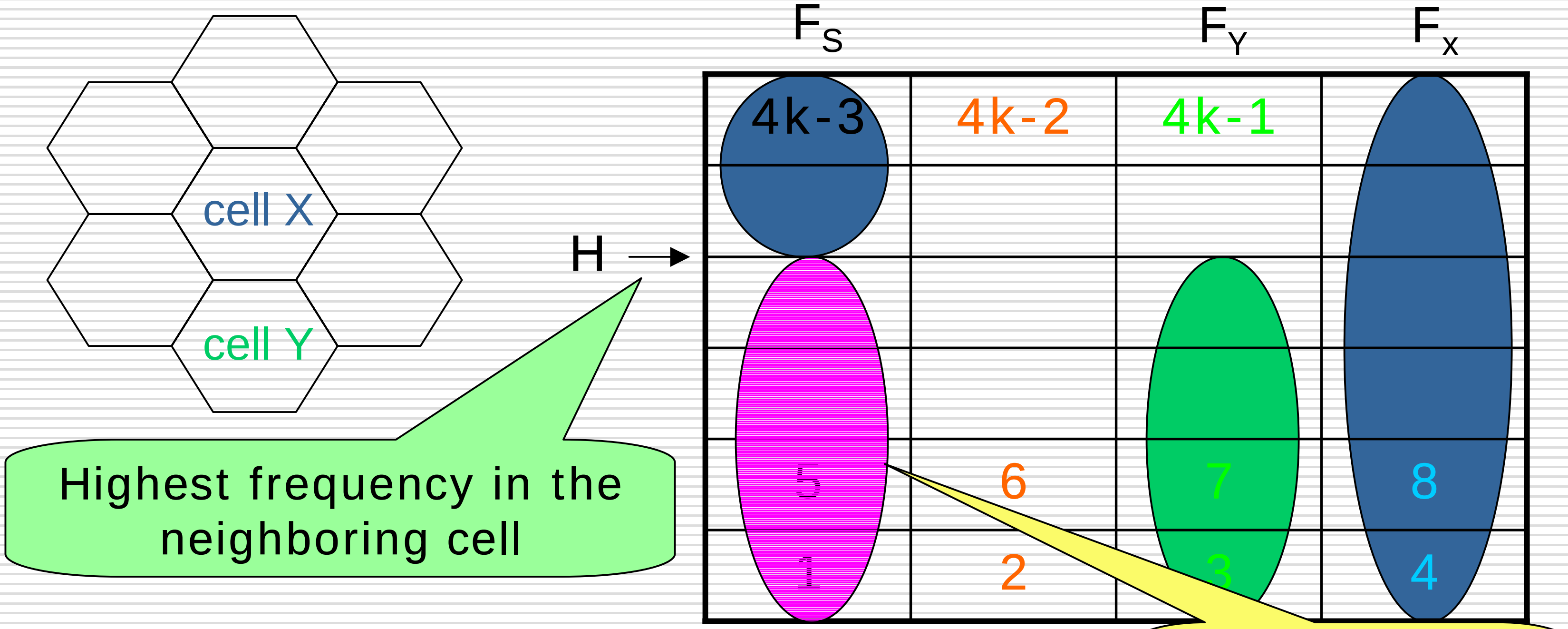
Asymptotic bounds for FAP

- In the lower bound proof
  - Optimal requires 2 or 3 frequencies
  - Adversary uses 4 or 6 frequencies with 2 discrete decisions.
  - When the number of calls are large, decisions may be not discrete.
  - Some the same and some different.





Proof of Upper Bound revisited



- When all *n* frequencies are in a single cell X, the bound of Hybrid is still 2
- The component of greedy should be increased.

Family of Hybrid Algorithms

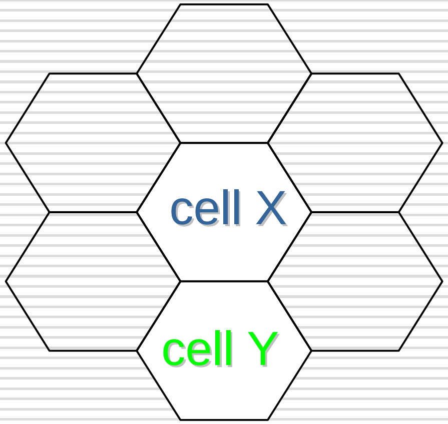
- Frequencies partitioned into groups of $\Delta = \alpha + 3\beta$

  greedy (α) — Fixed allocation (3β)

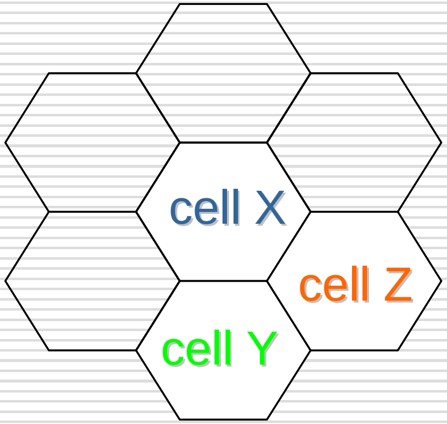
- From each group:
  - F_S is assigned α frequencies
  - Each of F_R, F_G & F_B is assigned β frequencies
- Define two parameters α and β
  - $\alpha=0$ ➔ purely fixed allocation
  - $\beta=0$ ➔ purely greedy
  - $\alpha=\beta$ ➔ ordinary hybrid (described before)

Asymptotic Upper Bound

- In the absolute case,
  we consider cell X and cell Y
  - Competitive ratio ≤
    (highest frequency in cell X) /
    (total no. of calls in X & Y)



- For the asymptotic case,
  we consider cell X and <u>two</u> of its neighbors.
  - Competitive ratio ≤
    (highest frequency in cell X) /
    (total no. of calls in X, Y & Z)



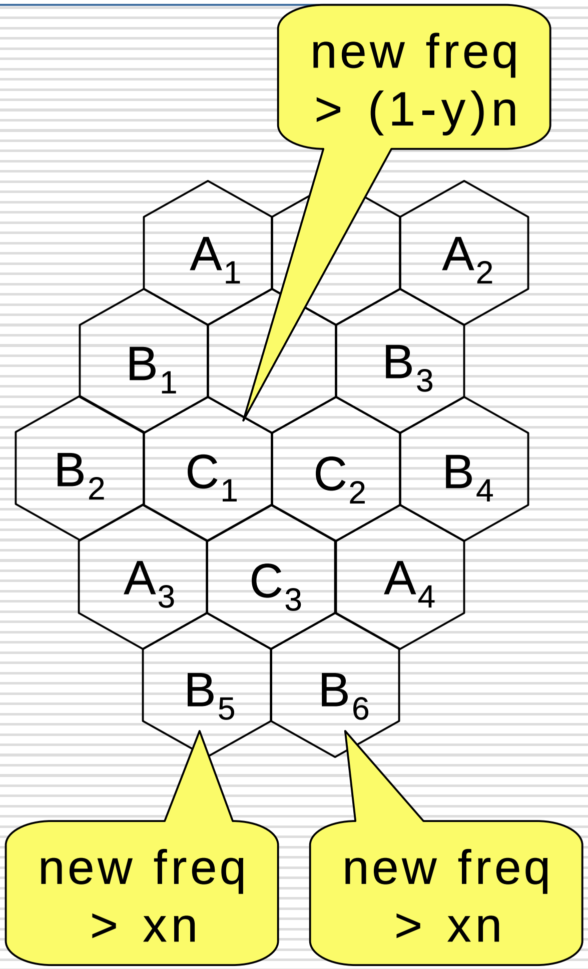
Asymptotic and Absolute Upper Bounds

- <u>Theorem</u>:
  If $\beta / \alpha \rightarrow 0.8393$ (more greedy),
  asymptotic competitive ratio $\rightarrow 1.9126$

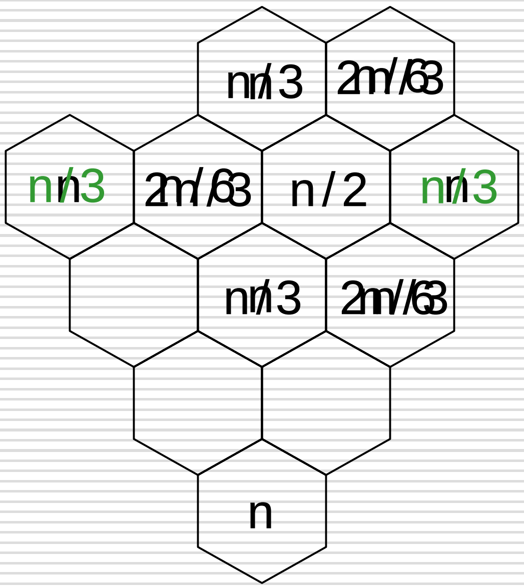
- For $\alpha=13$, $\beta=11$ ($\beta / \alpha \approx 0.8462$),
  asymptotic competitive ratio $\rightarrow 1.9167$
  and absolute competitive ratio = 2.

Asymptotic Lower Bound = 1.5 by Adversary

- n calls from A_1, A_2, A_3 and A_4
  - Common freq between any 2 cells = xn
  - No. of distinct freq > (2-x)n
  - Competitive ratio = (2-x)
  - Adversary stops if x < 0.5 as ratio > 1.5.
- n calls from B_1, B_2, B_3, B_4, B_5, B_6
  - No. of distinct freq = (2+x+y)n [$y \geq 0$]
  - Competitive ratio = (2+x+y)/2,
  - Adversary stops if x+y > 1 [so $x \geq 0.5 \geq y$]
- n calls from C_1, C_2, C_3
  - C_i gets all new freq except yn
  - Each C_i uses $\geq$ (1-y)n new frequencies,
  - The ratio is $(5+x-2y)/3 \geq 1.5$



Asymptotic Lower Bound = 2 (with-deletion)



The optimal takes
n frequencies always

- If no of frequency > 2n, stop. Otherwise
  - Choose 2 with > n/3 common frequencies.
- At least 2n/3 "new" frequencies
- At least n/3 "new" frequencies
- At least n/3 "new" frequencies
- Must be totally different from the surrounding cells.
- Optimal uses only n frequencies.

Theorem: No online algorithm has the asymptotic competitive ratio less than 2 (with deletion).

Other Variations: With deletion

- Generalized version of the online problem
- A call may terminate at any time
- Performance:
  - Fixed Allocation: 3-competitive
  - Greedy Allocation: 3-competitive [Chan, Chin, Ye, Zhang, Zhu, IPL 2006]
  - Better than 3-competitive? Still open!
- Lower bounds
  - Asymptotic competitive ratio = 2
  - Absolute competitive ratio = ?.

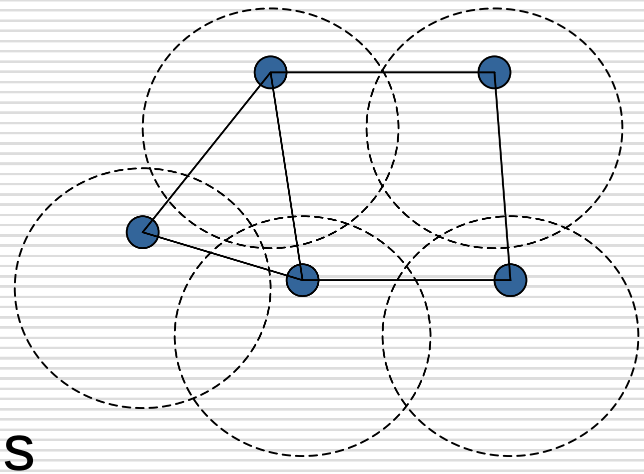
Coverage by Linear Networks

[Chan, Chin, Ye, Zhang, Zhu, ISAAC 2006]

- The geographical coverage area is divided into cells aligned in a line
- Online:
  - Upper bound=lower bound= 1.5
  - Asymptotic upper bound= 1.382
  - Asymptotic lower bound=4/3≈ 1.333
- Dynamic (with deletion):
  - Upper bound=lower bound=5/3≈ 1.667
  - Asymptotic upper bound=5/3≈ 1.667
  - Asymptotic lower bound=14/9≈ 1.556

Other Variations: Different Graphs

- k-colorable graphs
  - Online case: Hybrid algorithm is (k+1)/2-competitive
    - E.g., planar graphs are 4-colorable ➔ 5/2-competitive



- Unit disk graphs
  - Vertices ➔ circles
  - Edge ➔ intersecting circles

No result known for online or dynamic FAP

Other Variations: Reuse Distance

- Current setting ➔ reuse distance=2

- In general, reuse distance=r
  - Offline
    - r=3 ➔ 2-approximation [Kchikech, Togni 2005]
    - r≥4 ➔ 3-approximation [Kchikech, Togni 2005]

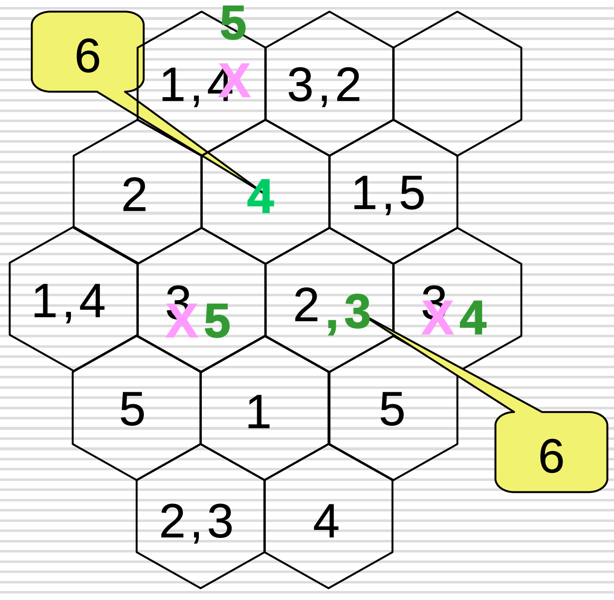
  - Online (and Dynamic)
    - By a fixed allocation scheme
    - r=3 ➔ 3-competitive [Jordan, Schwabe, ACM
    - r=4 ➔ 4-competitive J. Wireless Networks1996]

Frequency Assignment Problem (FAP) Outline

- Background
- FAP without reassignments
  - Absolute competitive ratio (Hybrid)
  - Asymptotic competitive ratio
  - With deletion
- **FAP with reassignments**
- FAP with OVSF code

FAP with Reassignments

Maximize bandwidth utilization by reassignment of frequencies



Can we get the optimal utilization?

- NP-complete even for offline
  [McDiarmid & Reed Network 2000)
- NP-complete for an algorithm with approximation ratio < 4/3)
  [3-colorability problem is NP-complete]

Approximation algorithm with ratio 4/3 exists
[Narayanan & Shenda, Algorithmica 2002;
Spare et al, J. Pure App Math 2002]

On-line Distributed Algorithms

How good is the bandwidth utilization by frequency reassignments on the the neighboring cells?

k-locality: neighboring cells with distance $\leq k$.

Fact : *k*-local α-approximation algorithm
$\rightarrow$ *k*-local α-competitive online algorithm
[Janssen, et.al., J. Algorithms 2000]

- 0-local – 3-competitive
- 1-local – 3/2- improved to 13/9-competitive
  [Chin, Zhang, Zhu, this conference]
- 2-local – 17/12- improved to 4/3-competitive
  [Sparl & Zerovnik, J. Algorithm 2005]
- 4-local – 4/3-competitive

Unsolved Problems

- Can we improve the following results?
  - 0-local – 3-competitive
  - 1-local – 13/9-competitive
  - 2-local – 4/3-competitive

- Besides better competitive ratios, do these schemes use the least number of reassignments?

Frequency Assignment Problem (FAP) Outline

- Background
- FAP without reassignments
  - Absolute competitive ratio (Hybrid)
  - Asymptotic competitive ratio
  - With deletion
- FAP with reassignments
- FAP with OVSF code

Background – Third Generation (3G)

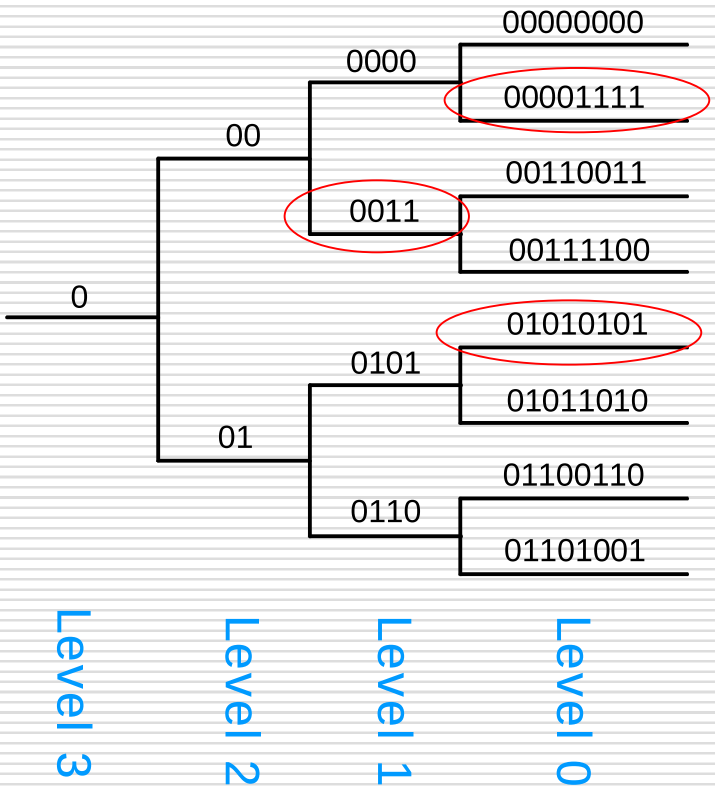
- Multiple services of 3G communication
  - Voice
  - Messaging – email, fax, etc.
  - Medium-rate multimedia – Internet access,
  - High-rate multimedia – file transfer, video
  - High-rate interactive multimedia – video teleconferencing, telemedicine, etc.

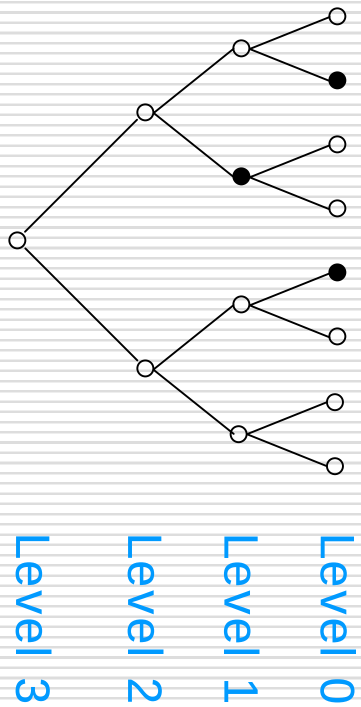
Background – OVSF Codes

Wideband Code-Division Multiple-Access (WCDMA) with OVSF code

- Structure: a complete binary tree
- Assigned nodes must be *mutual orthogonal*, i.e., each root-to-leaf path contains at most one assigned code (the 2 nodes are neither ancestor nor descendant of each other).

An Example of OVSF Codes





Data rate of level-i code is twice that of level-(i-1) code

Problem Background

Given a code operation sequence $\delta = \{C_1, C_2, \ldots, C_t, \ldots, C_n\}$

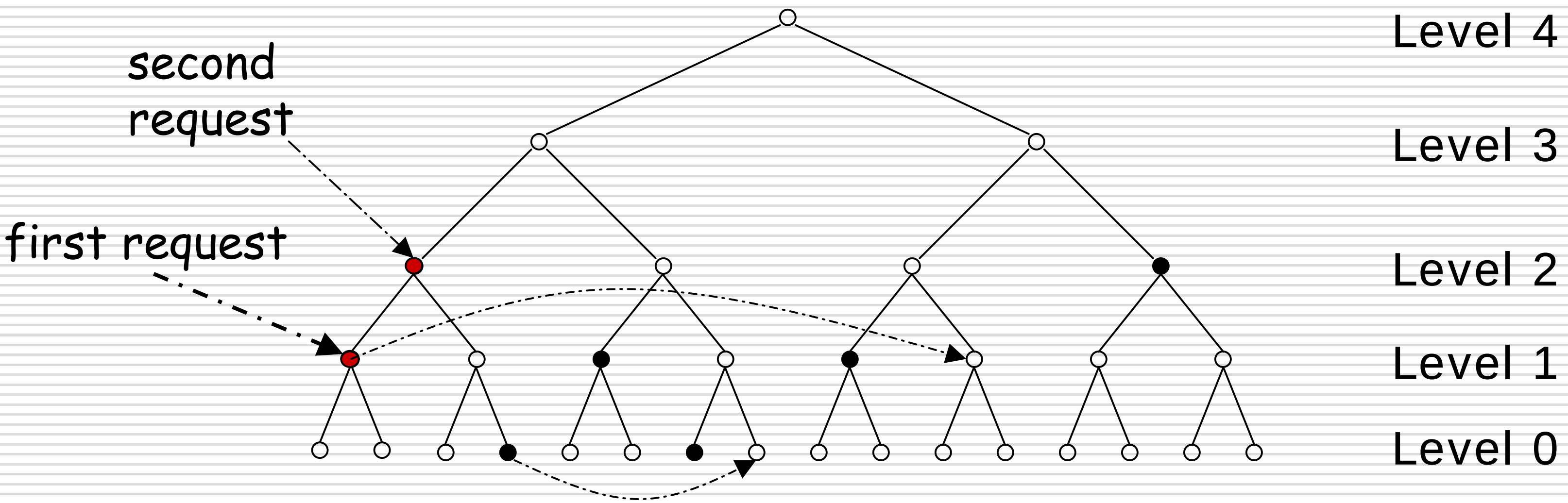
$$C_t = \begin{cases} \text{a code request for a particular level (cost = 1)} \\ \text{an assigned code release (cost = 0).} \end{cases}$$

Problems with a code request

- Code blocking or Code-tree fragmentation
- Reassignments of assigned codes might be needed.
- Each reassignment (code request + code release) costs 1 unit.

Problem Statement – an example

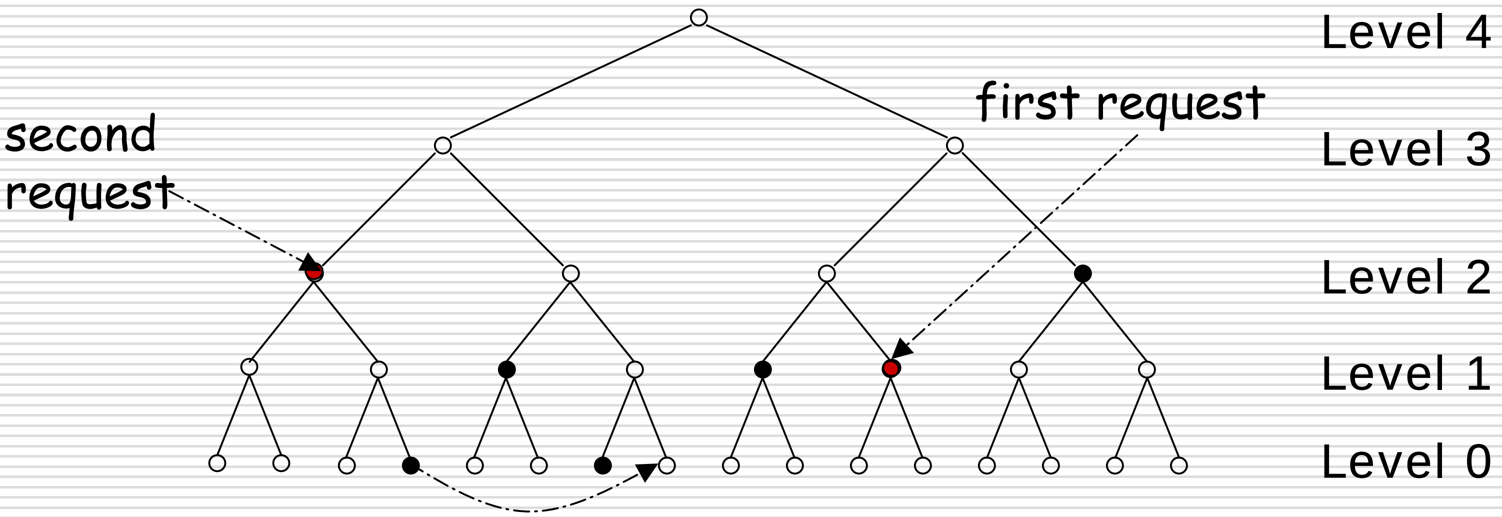
Request : a level-1 code and a level-2 code.



- Code blocking occurs.
- 2 code reassignments are needed.
- Total cost = 4 units (two assignments and two reassignments)

Another Appraoch

Request : a level-1 code and a level-2 code



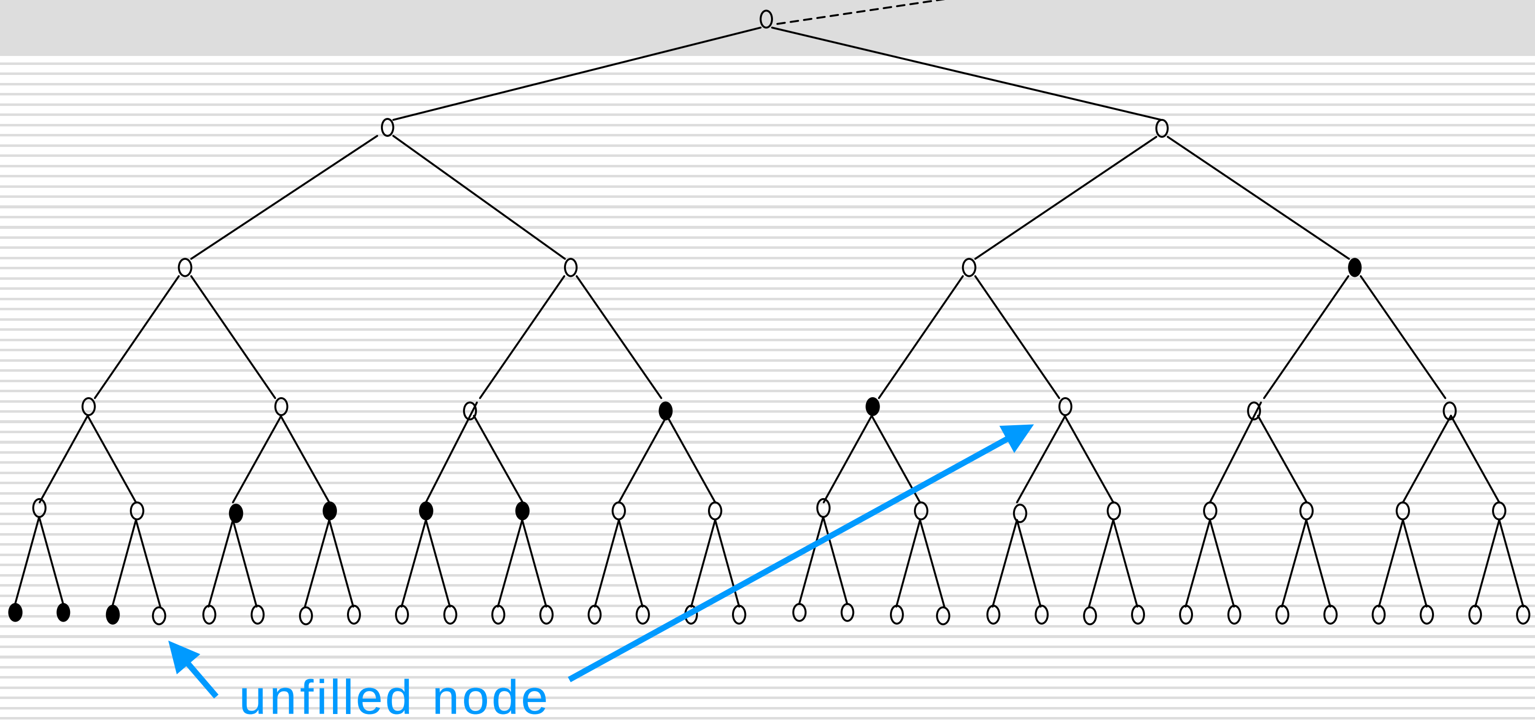
- One code reassignment is needed
- Total cost = 3 units (2 assignments and 1 reassignment)

Problem: Minimize the total cost, i.e., minimize the total number of assignments and reassignments.

An O(h)-competitive algorithm

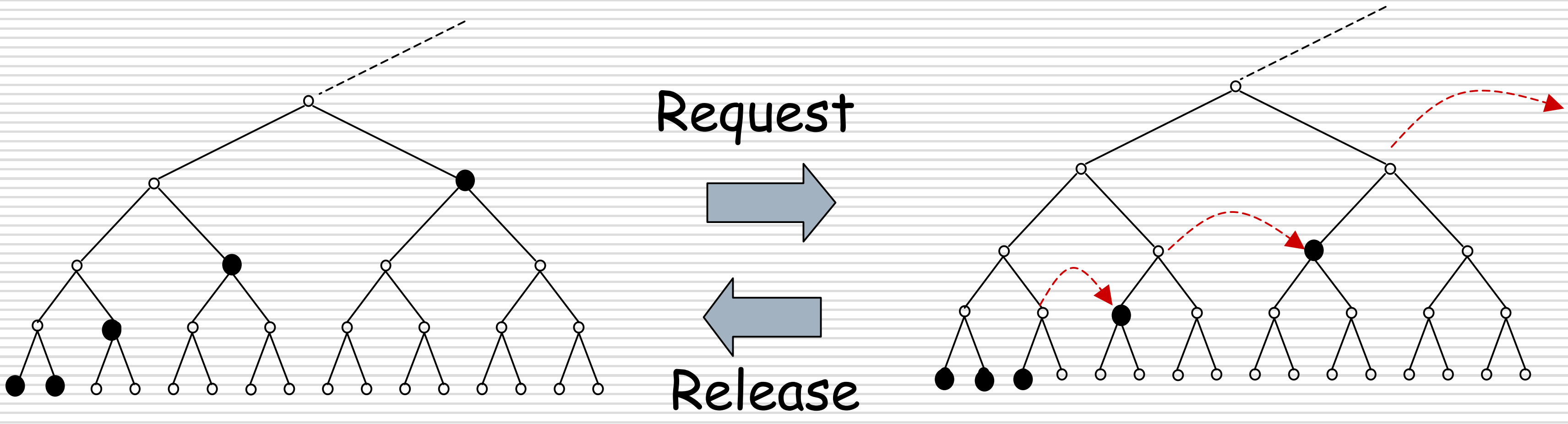
(Erlebach et al. STACS2004)

- All assigned codes are sorted and compact
  - Sorted – by level, from left to right
  - Compact – left shift as much as possible
    (at most one unfilled code at each level)



Example for O(h) reassignments

Request or release of a level-0 code



Recent Results

Code Assignment with Extra Resources

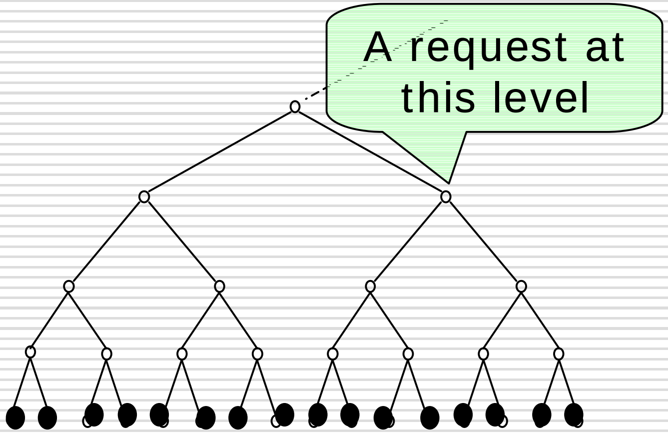
- 4-competitive with twice bandwidth
  (Erlebach et al. STACS2004)
- 5-competitive with 9/8 bandwidth
  (Chin , Zhang, Zhu, AAIM 07)

Code Assignment without Extra Resources

- Constant competitive (amortized)
  (Forisek et al.,ESA2007)
- 10-competitive by grouping levels
  (Chin, Ting, Zhu, manuscript)
- 8-competitive and 6-competitive (amortized) by a skew configuration (Chin and Zhu, manuscript)

Lower Bound

- No online algorithm can be better than 1.5-competitive [Erlebach et al, STAC 2004]
- Outline proof for the 1.5 bound
  - Start with *n* assigned leaves
  - Release alternative assigned nodes
  - Request a level-(h-2) node which requires *n*/4 reassignments

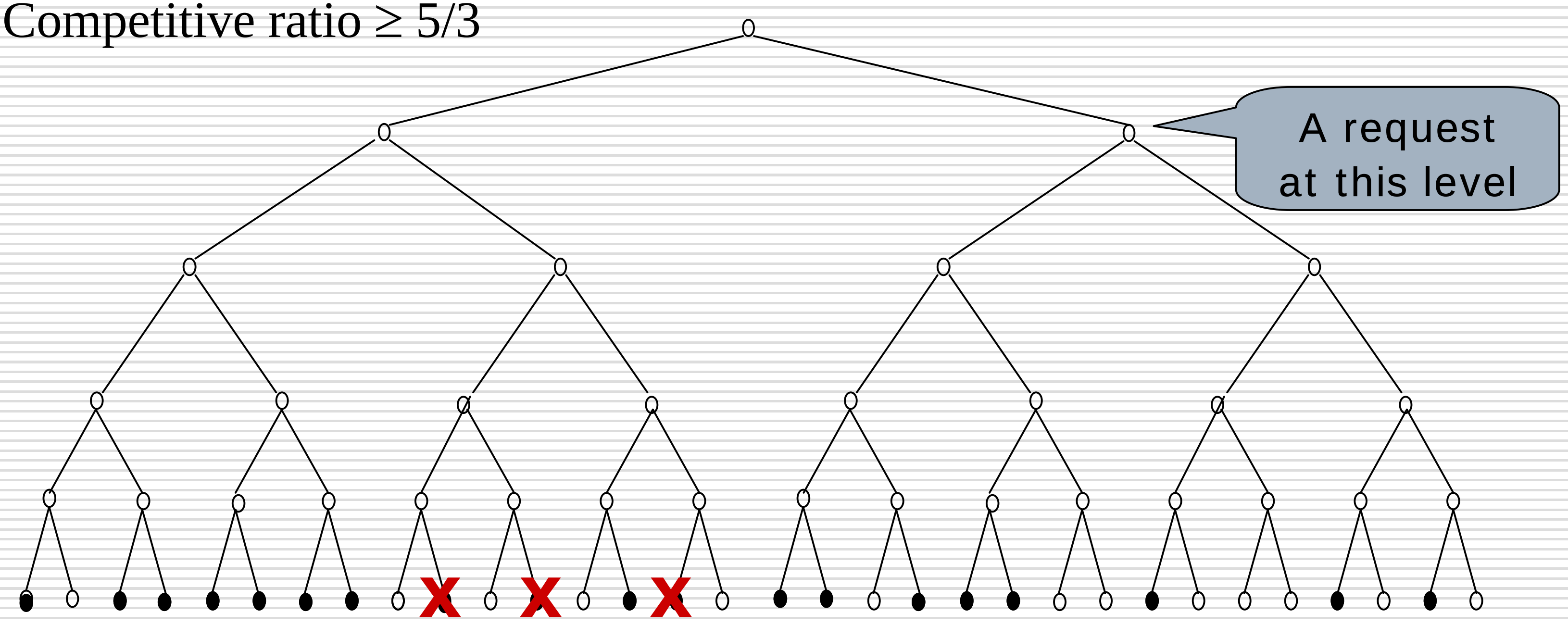


Request level	No. of leaves	No of reassign
0	16	4
1	8	2
2	4	1

- The cost by adversary $= n + n/2 - 1 + \log n - 1 = 3n/2 + \log n - 2$
- Optimal cost $= n + \log n$
- The competitive ratio is at least 3/2

An Improved Lower Bound to 5/3

- Each of the two subtrees has at least $n/4$ assigned leaves (at most 3n/4 code requests)
- The remaining steps are similar …
- The cost by adversary = $3n/4 + n/2 + \log n - 2 = 5n/4 + \log n - 2$
- Optimal cost = $3n/4 + \log n$
- Competitive ratio $\geq 5/3$



Open problem (OVSF code)

- The best algorithm is 6-competitive for single cell, lower bound is 1.75.

  Can we narrow the gap?
- What is the code assignment for cellular network, instead of single cell?
  - A 5-competitive algorithm with 19/8 resource augmentation (Chin et al, AIMM 2007)

HKU Theory Group at 2006

Thanks

COCOON 2007!

Questions?