UNIT-II: Location Management

1 Introduction

- Location Management is the process to determine the current location of a mobile terminal.
- In a PCS (Personal Communications Service) system, the location of a called portable must be determined before the connection can be established.
- In a mobile networks, the location of the terminal cannot be deduced from its endpoint address, like in a wired networks.
- Additional addressing schemes and protocols are needed to locate and track mobile terminals.
- Location management can be divided into two different services:
    - **Mobile tracking:** to keep track of the current location of the mobile terminal.
    - **Mobile locating:** to find the current location of the mobile node for the delivery of an incoming call.
- In a wireless communications system, mobile users are located in system-defined zones that correspond to bounded geographical areas.
- Location Management: managing the information required to locate wireless users who move from zone to zone.
- Components in the Location Management:
    - **Location server:** maintain the location of all the nodes in the group.
    - **Mobile Access Point**: provide a point of attachment to the network for the mobile nodes and routing capability.
    - **Mobile Node:** can be connected into network via MAP, no routing capability.
    - **System Manager:** manage the network.

Location Management Problem

- **In static networks, a terminal's network address serves two purposes:**
    - End-point identifier
    - Location identifier
  - Mobility prevents using a single address for both purposes
  - Both end-point identifier and location identifier are needed.
  - Location management keeps mapping between an end-point identifier and its location identifier
    - Basically a directory problem.
- **Two primitive operations:**
  - Lookup (search/find/paging/locating) operation: is the procedure by which the network finds the location of the mobile.
    - required when a call (message) to a user is placed (to be delivered)
  - Update (tracking/move/registration) operation: is the procedure by which the network elements update information about the location of the mobile.
    - required when a user changes its "location"
    - The information gathered during updating/tracking is used during the locating operation

Location Management: Issues

- **More precise location needs to be maintained as cell size shrink:**
  - Wide area cells are 10's – 100's km in diameter
  - Macro-cells: 1-10 km
  - Micro-cells: 100's m
  - Pico-cells: under 10 m
- **Database issues in tracking mobile users:**
  - Maintaining update intensive location information
  - Strategies to reduce location query latency (such as replication) and traffic (such as caching)
  - Consistency between replicas; Cache management polices

- **Location Management: Schemes**
  - **Several schemes have been developed which are motivated by fundamental trade-off between search operation cost and update operation cost.**
    - Schemes which try to minimize one cost tend to increase the other cost
    - Try to optimize the aggregate cost or normalized cost.
  - **Categorization:**
    - Update Scheme: Static or Dynamic
      - Static update scheme: registration areas
      - Dynamic update scheme: distance/time/movement based strategy
    - Locating Scheme: Static or Dynamic
      - Static location scheme: page all the cells in the network
      - Dynamic location scheme: expanding ring search centered at last reported location of the the user
    - Database Architecture: Flat or Hierarchical

2 Selection of LM Schemes

- Cost of location updates and lookups
- Maximum service capacity of each location database =
  - the maximum rate of updates and lookups that each database can service
- Space restrictions (size of the location database)
- Type and relative frequency of call to move operations (call-to-mobility ratio (CMR))

2.1 One-Tier Scheme

- A home database, called Home Location Register (HLR) is associated with each mobile user.
- The HLR of a user x maintains the current location of x as part of x's profile.
- To locate a user x, x's HLR is identified and queried.
- When a user x moves to a new cell, x's HLR is updated.

2.2 Two-Tier Scheme

- **Basics:**
- Visitor Location Registers (VLRs) are maintained in each zone (registration area).
- VLR in a zone stores copies of profiles of users not at home and currently located in that zone.
- When a call is placed from cell i to user x, the VLR at cell i is queried first, and only if the user is not found there, is x's HLR contacted.
- When user x moves from cell i to j, in addition to updating x's HLR, the entry of x is deleted from VLR at cell i, and a new entry for x is added to the VLR at cell j.

- **Two-Tier Scheme: Standards**
- Many current and proposed standards use this scheme:
  - Electronics Industry Association Telecommunications Industry Associations (EIA/TIA) Interim Standard 41 (IS 41) - commonly used in North America.
  - Global Systems for Mobile Communications (GSM) - used in Europe.
  - Internet Engineering Task Force (IETF) Mobile IP protocol

- **Enhancements to Basic Scheme**
- **Per User Location Caching (dynamic replication)**
  - reduces search (lookup) cost
  - increases update cost
  - exploits locality in call pattern

- **(Static) Replication**
  - Per User Profile Replication
  - Working Set Replication
- **Forwarding Pointers**
  - reduces update (move) cost

Per User Location Caching

- **Basic Idea:**
  - Every time user x is called, x's location (or a pointer to this location) is cached at the VLR in the caller's zone.
  - Any subsequent call to x originating from that zone can use this information:
  - Upon call origination the cache at the VLR of the caller's zone is checked before querying the callee's HLR.
- **Issues:**
  - cache replacement schemes (LRU can be used)
  - cache invalidation schemes
  - eager caching or lazy caching

Eager and Lazy Caching

- **Eager Caching**
  - Every time a user moves to a new location, all cache entries for this user's location are updated.
  - The cost of move operations increases for those users whose address are cached.
- **Lazy Caching**
  - the cached pointer for any given user is updated only on a cache miss
  - for lazy scheme to work better than basic scheme $p \geq C_H/C_B$ where p is the hit ratio, C_H is the cost of a lookup when there is a hit and C_B is the cost of lookup in the basic scheme.

Replication

- To reduce the lookup cost, the location of specific users is replicated at selected sites.
- Let

  α : cost savings when local lookup succeeds as opposed to a remote query,
  β : cost of updating a replica,
  $C_{i,j}$: expected number of calls made from cell j to i in a unit time.
  U_i: expected number of moves by i in unit time

  Then a replication of the location of user i at cell j is judicious if
  $\alpha * C_{i,j} \geq \beta * U_i$.

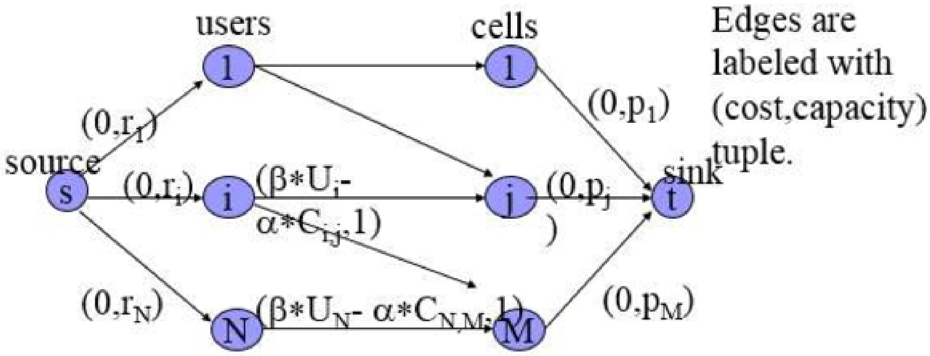
Per User Profile Replication

- Objective: to minimize the total cost of moves and calls, while maintaining
  Constraint 1: a maximum of r_i replicas for user i, and
  Constraint 2: a maximum of p_j replicas in the database of cell j.
- Replication assignment problem: The profile of user i is replicated at all cells in set R(i) such that the system cost

  $$\Sigma^{N}_{i=1}\Sigma^{M}_{j=1, j\in R(i)} (\beta * U_i - \alpha * C_{i,j})$$

  is minimized, where N is the number of users and M is the number of cells., and constraints 1 and 2 above are met.

- **Replication Assignment**
- **Flow Network based solution:**



- Solving Min-Cost Max-Flow on the Flow Network finds the required assignment.

- **Working Set (WS) Replication**
- Relies on the observation that each user communicates frequently with a small number of sources called its working set.
- Copies of location are maintained at the members of its working set.
- No constraints are placed on database storage capacity or on number of replicas per user.
- Hence, the decision to provide the information of the location of a mobile unit i to zone j can be made independently for each user.
- **Adapts to user's call and mobility patterns**

- **Working Set Adaptation**
- The inequality Q: $\alpha * C_{i,j} \geq \beta * U_i$. is evaluated locally at a mobile unit I each time:
  1. a call is set up,
  2. the mobile unit moves.
- In case 1, Q **is evaluated only if the caller's site is not a member of mobile's WS:**
  - **If the inequality holds then the caller's site becomes member of the callee's working set.**
- In case 2, the Q is evaluated for every member of WS; the members for which Q no longer holds are dropped from WS.

- **Performance of WS Replication**
- Computation overhead:
  - in case 1 all four terms of Q need to be reevaluated
  - In case 2 only the number of moves (U_i) needs to be reevaluated.
- Adaptability:
  - When call-to-mobility ratio (CMR) value is low the WS scheme performs like a scheme without replication.
  - When CMR value is high, the scheme behaves like a static scheme in which the WS for a user is fixed.
- Performance is mainly dependent upon the CMR of individual users (not on num. of users).

- **Forwarding Pointers**
- Each time a mobile unit x moves to a new location, a forwarding pointer is set up to its previous VLR to point to the new VLR.
- To establish a call, the HLR of callee is queried to find the first VLR in the forwarding pointer chain. This chain is followed to get to the current VLR of the callee.
- To bound the time taken for lookup procedure, the length of the chain is bound to a max value of K.
- Pointer compression is used to eliminate loops.
- Mobile IP protocol includes pointer forwarding in conjunction with lazy caching.

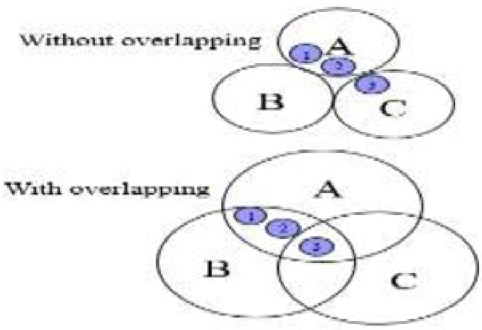
- The forwarding pointer strategy is useful for those users who receive calls infrequently relative to the rate at which they change registration areas.
- Benefits of forwarding depends also upon the cost of setting up and traversing pointers relative to the costs of updating the HLR.

Summary of Variations to 2-Tier Scheme

Method	Variations		Used When:
Caching When x is called by y, cache x's location at y' zone	Eager caching	Cache update overhead occurs at moves	Large LCMR
	Lazy Caching	Cache update overhead occurs at calls	
Replication Selectively replicate x's address at the zones from which it receives the most calls	Per-user profile Replication	Additional constraints are set on the number of replicas per site and on the number of replicas per user	Large LCMR
	Working Set	Adaptive Distributed: the replication sites are computed locally at each mobile host	
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

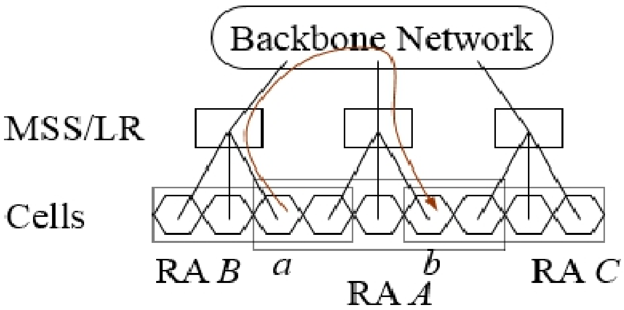
Overlapping Registration Areas

- Inter-RA hand-off: a user changes cells and RAs
- Intra-RA hand-off: a user changes cells within an RA.
- Inter-RA hand-off doesn't happen as long as the hand-off can be intra-RA.
- Inter-RA call is when caller and callee are in separate RAs
- Intra-RA call is when caller and callee are in same RA.
- A non-overlapping cell is serviced by one LR.
- A overlapping cell is serviced by multiple LRs.
    - Reduction of inter-RA hand-offs.



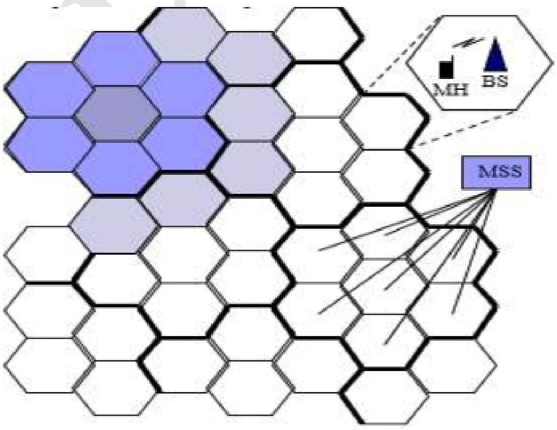
- Advantages:
    - Each RA can provide service to more mobiles within their covered area.
    - Reduces the number of inter-RA handoffs
    - Reduce the load to update mobile's HLR.
- Disadvantages:
    - the communication overhead for call-delivery and intra-RA handoff is increased.
    - the increase in overhead depends upon the underlying network topology.
    - If this overhead is ignored then the extreme configuration in which each RA has all the cells in the system becomes the "optimal" configuration.

- Dynamically Resizing RAs
  - We need to find optimal configuration (allowing overlapping RAs) i.e. configuration which minimizes load on MSSs.
  - When move and call patterns periodically change, a static scheme may not provide a good solution
  - Our Approach: Allow RAs to be dynamically adapted.
  - Periodically resize RAs to minimize MSS load:
    - Resizing criterion: load reduction due to lesser number of inter-RA handoffs > increase in load due to more expensive call delivery and intra-RA handoffs.
    - If resizing criterion is ignored then each RA will grow to maximum size.
- Negative effect of underlying conventional star topology on signaling overhead under overlapping RAs



Even though mobiles *a* and *b* belong to the same RA, any calls between them would need to go through two MSSs.

- Inclusion and Exclusion Boundary
  - In order to facilitate orderly growth and shrinking of RAs, an MSS only includes and excludes cells from its RAs current boundary.
  - Two types of boundary:
    1. Internal Boundary
    2. External Boundary

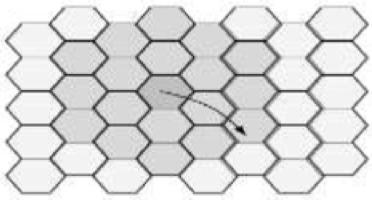


- Inclusion/Exclusion Decision
  - The decision to include or exclude a candidate cell is based on whether the resulting configuration will have a lower expected load on MSS.
  - For a given system configuration A, mobility pattern M, and call C, SystemLoad(A,M,C) is the combined signaling load (in terms of message time complexity) as a result of all the handoffs due to M and call-deliveries due to C:
    SystemLoad(A,M,C) = $\sum$Load(k,M,C).
  - In case of inter_RA handoffs and call-deliveries we spilt the signaling overhead equally between the two MSSs involved

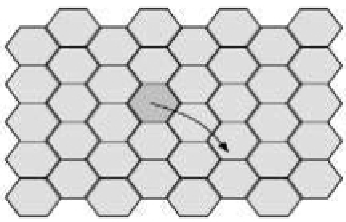
- What changes when cell *x* is included in RA *r?*
  - Handoffs to cell *x* from cells of RA *r* become intra-RA handoffs.
  - Handoffs from cell *x* to rest of RA *r* performed by users already registered in *r* become intra-RA handoffs.
  - Calls to *x* from cells of *r* are now intra-RA calls.
  - Calls from users of *r* that are in *x* to rest of *r* are now intra-RA calls.
  - Mobility of users in *r* that move out of cell *x* into a new RA is now inter-RA mobility.
  - Inter-RA calls of users in r that call from cell x is inter-RA call loading to r.
- Call the decreasing part of the load $Cost_{in}(x,r)$ and the increasing part $Cost_{ex}(x,r)$.
  - At intervals T each MSS/LR *r* computes I_Boundary(*r*) and E_Boundary(*r*) and for each cell *x* in the two sets computes $Cost_{in}(x,r)$ and $Cost_{ex}(x,r)$. By comparing the two values, it decides if it is worth keeping excluded, keeping included, including or excluding the cell *c*.

Dynamic Updating

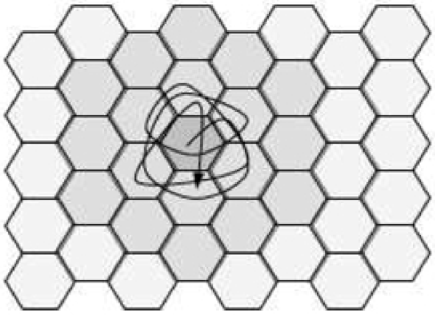
- Most schemes have fixed locations (i.e. the boundaries of registration areas) where the mobiles update.
- Users that move around boundaries cause a lot of registrations.
- **Solution: introduce dynamic update schemes that don't depend on location of mobile.**
- Time-based
  - User updates location at intervals of time T independent of actual location.
- Movement-based
  - User updates location after crossing M hops (cells) from last updated location.
- Distance-based
  - User updates location after being distance D from last updated location.
- Two metrics to evaluate schemes:
  - update rate (# of updates/sec)
  - search area (# of cells/search)
- **Time-based versus Movement-Based**
  - Update rate
    - If user crosses less than M cells per time T, then time-based makes more updates, otherwise time-based makes less updates
    - If M is average hops traversed per time T, then two schemes have same rate of updates.



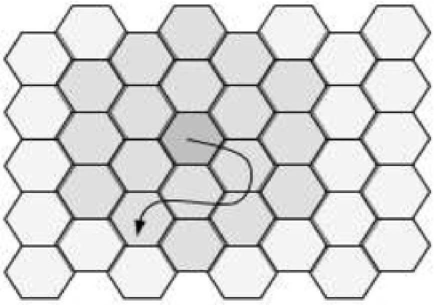
  - Search area
    - Search area in time-based is the cells that can be reached from last updated location at max user speed in time T.
    - Search area in movement-based is cells that can be reached in M or less hops from last updated location



- **Movement-based versus Distance-based**
  - Update rate
    - If D=M, distance-based will do at most as many updates as movement-based

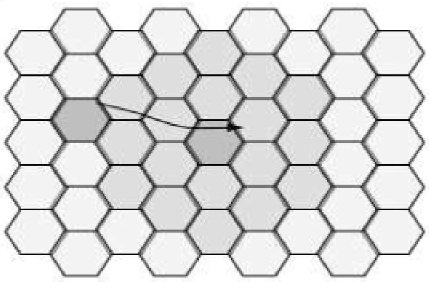


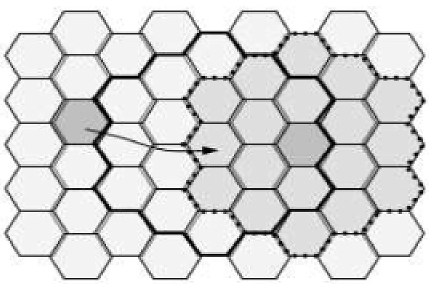
  - Search area
    - Search area is same in both schemes (cells that can are at distance D or less from last updated location)



❖ Look-ahead update

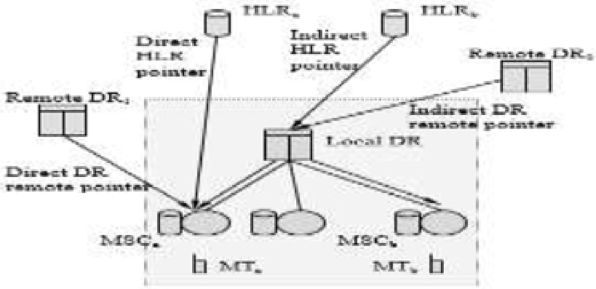
- [Tsai97] Proposed a look-ahead update scheme based on the distance-based scheme of [Bar-Noy95].
- **Mobility is modeled as a "normal walk" where the mobile tends to keep the direction of movement**
- Look-ahead update scheme: in distance-**based scheme don't update current location**, but update a location Δ hops ahead
- Under normal walk mobility model, user is more probable to cross standard circle (solid) before crossing look-ahead circle (dotted). Therefore look-ahead saves updates.





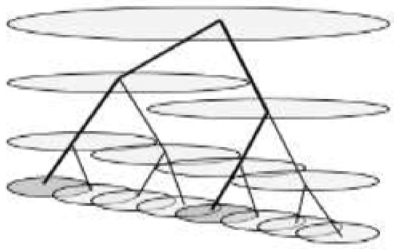
2.3 Dynamic Hierarchy

- [Ho97] Proposed a hybrid scheme with different hierarchy levels.
- Directory Registers: are a inter-mediate level of hierarchy between VLRs and HLRs.
- For each user there is the consept of Local DR, the DR that is above the VLR.
- HLRs may point to VLRs (direct pointer) or to LDRs (indirect pointer). Hence, the two hierarchies:
    - Two levels: HLR→VLR
    - Three levels: HLR →LDR →VLR
- Scheme includes Forwarding Pointers Of users various DRs to other DRs or LRs

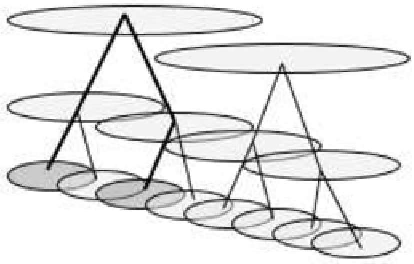


❖ **Multiple-level Hierarchical Scheme**

- If LAs are viewed as circles of radius *r*, then the outer $r/2$ part (periphery) overlaps with the inner parts (cores) of neighboring LAs, and the inner $r/2$ part (core) overlaps with the outer parts (peripheries) of neighboring LAs.
- At each level *n*, there is twice the number of the (*n*+1)-level Location Areas and half of the (*n*-1)-level Location Areas.
- Logarithmic number of levels
- For every user, there is at least one LA at each level that has a "location pointer" to a LA to the next lower level.
    - Update policy:
        - At each level, starting from lowest, if the user moves between two cells that are not in the same LA, the move is "updated" to the LA in the above level as well.
        - A movement update goes up to the LA that embraces both ending and starting cells of the user's movement.



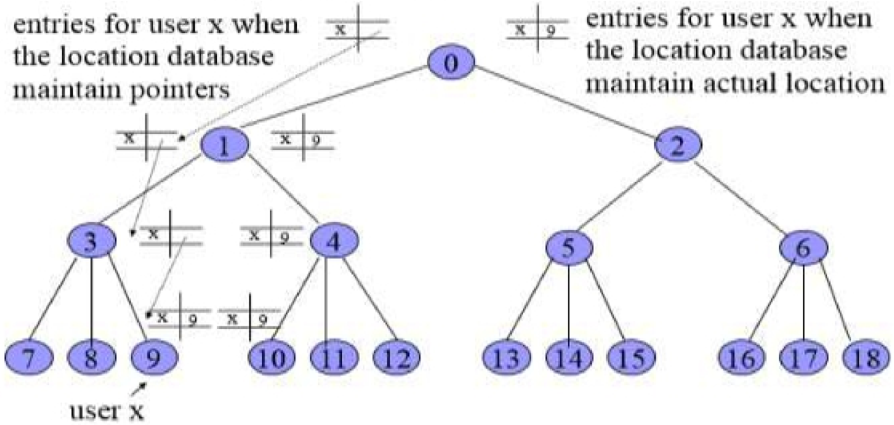
    - Search Policy:
        - If there is no downward pointer, then the search is propagated upward until a LA has a downward pointer of the user.
        - The downward pointers are followed until the user is reached.



- Scheme shows very good average and best case costs, but very bad worst case.

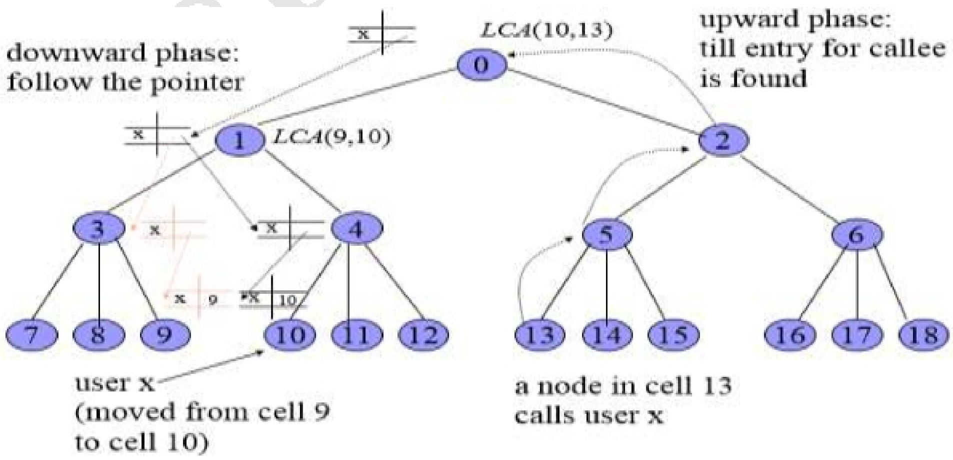
2.4 Hierarchical Schemes (HS)

- Extend two-tier schemes by maintaining a hierarchy of location databases.
- Location database at higher level contains location of users located at levels below it.
- Usually hierarchy is tree structured:
  - Location database at a leaf serves a single cell and contains entries for all users registered in that cell.
  - A database at an internal node maintains location of users registered in the set of cells in its subtree.
  - location information can be either
    - pointer to an entry at a lower level database or
    - The **user's actual current location.**



❖ **Updates/Lookups with Pointers**

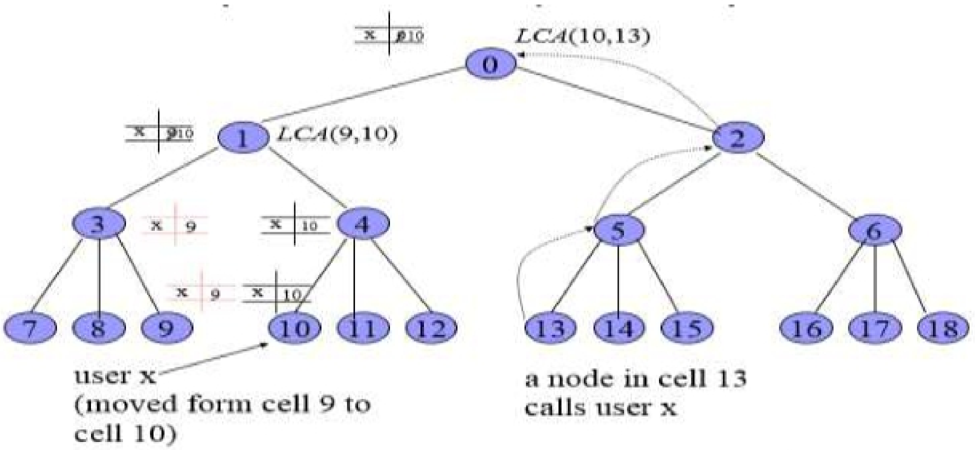
- *LCA*(*i,j*): least common ancestor of *i* and *j*.
- When user *x* moves from cell *i* to *j*, following entries for *x* in databases are updated:
  1. along the path from *j* to *LCA*(*i,j*), and
  2. along the path from *LCA*(*i,j*) to *i*.
- When a caller located at cell *i* places a call for a user *y* located at cell *j*, the lookup procedure:
  1. Queries databases starting from node *i* and proceeding upwards the tree until the first entry for *x* is encountered (at *LCA*(*i,j*)).
  2. Then the lookup procedure proceeds downwards following the pointers to node *j*.



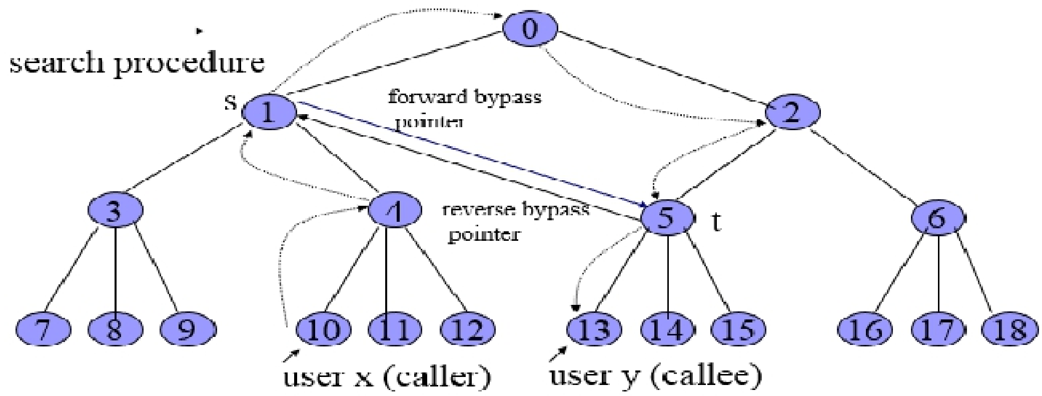
❖ **Lookup/Update with Actual Location**

- When user x moves from cell i to cell j:
  - record for x is deleted from all the databases from node i to LCA(i,j), and
  - record for x is updated to indicate the current location to be cell j in all the databases from root node to leaf node j.

- When a user x from cell i places a call to user y in cell j the lookup procedure queries database at node i proceeding upwards till node LCA(i,j).
- Compared to pointers case, in this case:
  1. Updates are more expensive operation, and
  2. Lookups are less expensive operation.

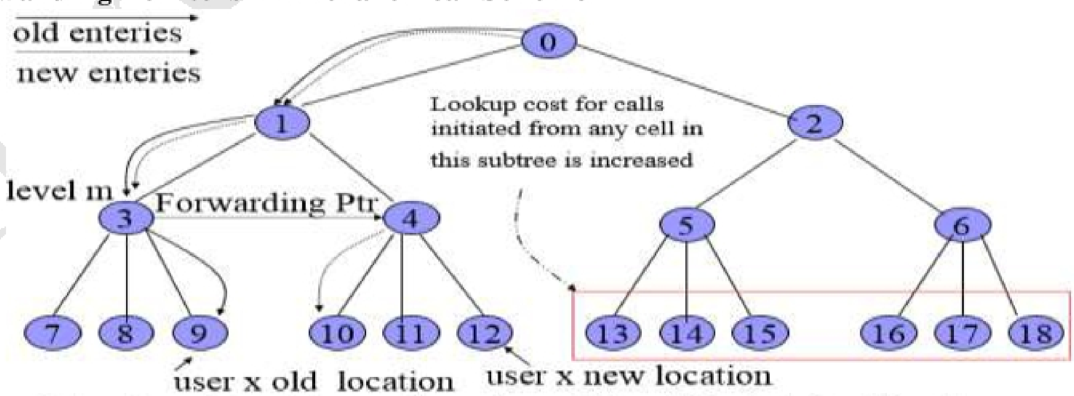


❖ Caching in Hierarchical Scheme



•Forward bypass pointer is an entry at an ancestor of caller's cell, say s, that points to an ancestor of callee's cell, say t.
•The reverse bypass pointer is from t to s.
•In simple caching both s and t are leaf nodes.
•In level caching s and t can belong to any (possibly different) levels.

❖ Forwarding Pointers in Hierarchical Scheme



Reduces the update cost in case of move from cell i to cell j, instead of updating all databases on the path from j through LCA(i,j) to i, only the databases up to a level m are updated and a forwarding pointer is set from a node s to node t, where s is the ancestor of i at level m and t is ancestor of j at level m.

- **Hierarchical Schemes: Summary**

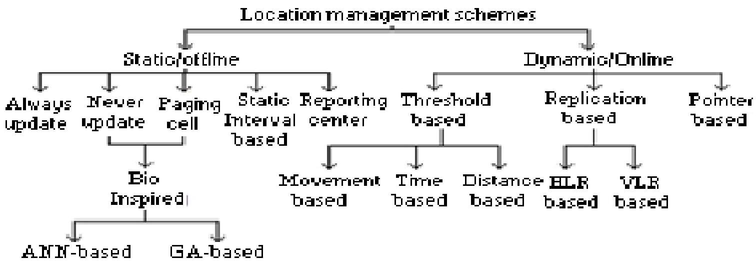
Method	Description
Caching	When x at zone i is called by user y at zone j, cache at a node on the path from j to LCA(i,j) a pointer to a node on the path from I to LCA(i,j) to be used by subsequent calls to x from zone j.
Replication	Selectively replicate x's location at internal and/or leaf database.
Forwarding Pointers	When x moves from cell i to cell j, instead of updating all databases on the path from j to LCA(i,j) and from LCA(i,j) to j, update all databases up to level m and add a forwarding pointer at the level m ancestor of I to point to the level m ancestor of j.

2.5 Hierarchical vs. Two-Tier Scheme

- No pre-assigned HLR
- Support Locality
- Increased number of operations (database operations and communication messages)
- Increased load and storage requirements at the higher-levels

3 LOCATION UPDATE SCHEMES

Location update schemes can broadly be classified into static and dynamic. Presently, most location update schemes are static due to their simplicity. As shown in Figure, Static LU schemes can be classified as Always-update, Never-update, Static Interval-based, Reporting center and Paging cell. In always update, MT updates its location upon every inter-cell movement, which incurs large number of updates for highly mobile users. Under never-update, adjacent cells in the network are grouped to function as a larger single cell called location area; update is triggered only on inter-LA movement but doing so, paging cost rises substantially for MTs with high incoming call frequency. The static interval-based scheme requires MT to update its location after every predefined, uniform time period that provides a balance between the extremes of previous two schemes. Locating highly mobile users becomes difficult under this scheme. Inversely, a stationary user regularly enforces unwanted LUs.



In reporting-center scheme, some cells are selected as reporting-centers; MT updates its location only when it enters into a new reporting-center. When a call arrives, the vicinity of the last reported reporting-center is paged to locate the target MT, selecting the reporting centers that provide quite good trade-off between paging and update operations, itself is a complex task. Paging-cell (PC) scheme groups the cells into paging areas, all cells within a paging area are paged simultaneously whereas each paging area can be paged sequentially until the target MT is located by the network upon arrival of a call, however long delay may occur in locating the MT. PC and LA based schemes are usually employed as the network topology for bio-inspired schemes that are generally based on artificial neural network or genetic algorithm. Table I illustrates the comparison between all the static LU schemes.

Table I: COMPARISON BETWEEN MAJOR STATIC LU SCHEMES

LU Scheme	Update cost	Paging cost	Location accuracy	Major drawback
Always update	high	low	1 cell	number of updates is too high
Never update	low	high	1 location area	whole LA needs to be paged
Paging cell	low	high	several cells	long time delay in large networks
Static interval based	constant	high	several cells	unnecessary updates by stationary users
Reporting center	low	high	several cells	high computational overhead
Bio inspired	sub-optimal	sub-optimal	several cells	too much computational overhead

Dynamic LU schemes work on the principle that the MTs follow different movement patterns therefore LU scheme should treat them differently, such that each MT has its own optimal LU standard based on its movement habits. Under threshold based schemes, LU occurs each time a parameter (time, movement or distance) goes beyond a threshold value, which can be modified on per user basis. In time-based scheme, MT updates its location at constant time intervals, which saves computation but incurs needless LUs when MT does not move. In movement-based scheme, LU takes place each time the MT crosses a predefined number of cells. It works better than time-based scheme, unless the MT is highly mobile. In distance-based scheme, MT performs LU when its distance from the cell where it performed the last LU, exceeds a predefined value. It requires MT to track distances, which increases computational overhead and so is the limited battery consumption on MT side. HLR based replication scheme lets the LM messages to be routed to current (nearest) HLR rather than the master HLR but it imposes extra load on current HLR, thus reducing its performance. In VLR-based replication scheme, location information of MT is replicated among VLRs so that the mobility information is more readily available to the network. Its major problem is to keep these replicas consistent and up-to-date, they must be updated whenever the user profile is updated. In pointer-forwarding schemes, some updates to the HLR are avoided by setting up a forwarding pointer from the previous VLR to the new VLR. However, the penalty is the time delay for tracking current location of MT. Table II illustrates the comparison between various dynamic location update schemes.

Table II: COMPARISON BETWEEN MAJOR DYNAMIC LU SCHEMES

LU Scheme	Update cost	Paging cost	Location accuracy	Major drawback
Time based	low	high	several cells	• unnecessary updates by stationary users
Movement based	low	high	several cells	• may suffer from ping-pong effect
Distance based	low	high	several cells	• high computational overhead on MT side
HLR level replication	low	high	1 cell to several cells	• extra burden on current HLR • increased call establishment delay
VLR level replication	low	high	several cells	• overhead of regularly updating distributed mobility information
Pointer based	low	high	several cells	• increased call establishment delay

3.1 Selective LA Update

The rationale behind the selective LA update scheme is that a daily commuter may cross a number of LAs on his/her way to and from work. However, he/she may only stay in some LAs for very short periods of time. Rather than performing location update whenever he crosses a new LA, the update process at certain LAs can be skipped. An analytical model is introduced in which the interconnections of the LAs are characterized by a graph model. A Markov movement model is used. The residence time in each LA follows a geometric distribution. A genetic algorithm is used to obtain the near-optimal solutions. For low residing probability in certain LAs and high update cost, results show that this scheme incurs a lower location management cost than the conventional LA-based scheme. For implementation, information regarding the transition probabilities and residence time is required. To estimate the transition probabilities between LAs for a particular user, his/her movements throughout the day can be observed over long periods of time. Since the LA-based update scheme is used in current PCNs, information about the frequency of his/her transition from one LA to another can be retrieved from the database.

3.2 Profile-Based

The goal of the profile-based location update scheme (also known as the alternative location strategy) is to reduce the update cost by taking advantage of the user's mobility pattern. The network maintains a profile for each user, which includes a sequential list of the LAs the user is most likely to be located at in different time periods. This list is sorted from the most to least likely LA where a user can be found. When a call arrives, the LAs on the list are paged sequentially. As long as the mobile terminal moves between LAs within the list, no location update is necessary. Location update is performed only when the mobile terminal moves to a new LA not on the list. The list may be derived from the user's movement history. As an example, in Fig. 3a, the set of LAs is {a, b, c, d, e, f, g}. Suppose during the time period (t1, t2), the sequential list for the mobile terminal is {b, a, e, f}. Thus, no update is required as long as the mobile terminal stays within those LAs. Upon a call arrival, the network will page location area b first, followed by location area a, and so on, until the mobile terminal is found or the list exhausted. For implementation, each mobile terminal must maintain a valid sequential list corresponding to a particular time interval. This list has to be updated from time to time.

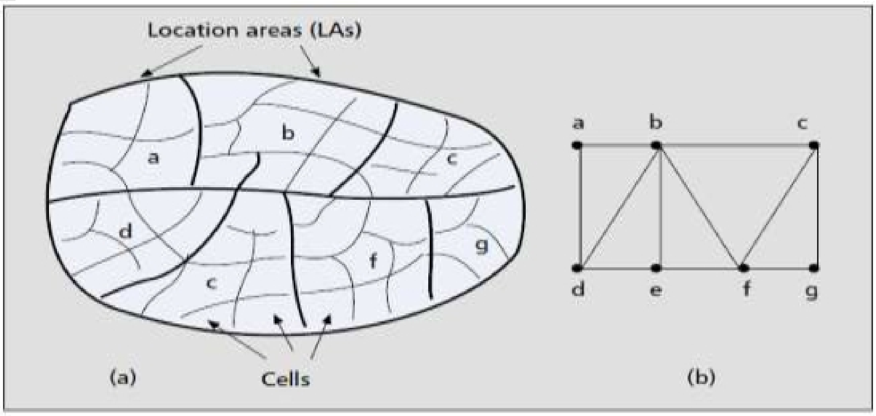


Figure 3. A representation of an actual cellular network topology by a graph: a) the cell and LA topology in a cellular network; b) a graph model showing the interconnections of the LAs.

3.3 Movement-Based

In the movement-based update scheme, each mobile terminal counts the number of boundary crossings between cells incurred by its movements. Location update is performed when this number exceeds a predefined movement threshold M (e.g., M = 6). This scheme allows the dynamic selection of the movement threshold on a per-user basis. For implementation, the mobile terminal only needs a counter to count the number of cell boundary crossings. The counter is reset whenever it reaches the movement threshold. The cell identification code (CIC) can also be used. With the CIC each cell is assigned a code, which is not necessarily unique. The code is used to **identify the cell's orientation relative** to other cells within the same LA. Each cell periodically broadcasts its identification code through the downlink control channel. The mobile terminals use this information to facilitate the update decision. For the movement-based scheme in an arbitrary cell topology, only four different codes are necessary. An analytical model is introduced to determine the optimal movement threshold. The model is applicable for mesh and hexagonal cell configurations under the assumptions of a general cell residence time distribution and symmetric random walk movement pattern. The maximum paging delay constraint is considered, and a shortest-distance-first order paging scheme is used.

3.4 Timer-Based

In the timer-based update scheme, each mobile terminal updates its location every T time units (e.g., T = 1 hour). This scheme does not require the mobile terminal to record or process location information during the time between updates. For implementation, the timer threshold can be programmed into the mobile terminal by a hardware or software timer. An analytical model is introduced to study the timer-based scheme. Assuming Gaussian distribution of user location probability and Poisson call arrivals, the update period that minimizes the cost of location update and paging is derived. Results show that the timer-based scheme performs substantially better than the LA-based scheme. A variation of the timer-based scheme called the adaptive threshold scheme is proposed. The mobile terminal transmits an update message every T¢ time units, where the parameter T¢ (referred to as the location registration threshold) is not a constant, but varies with the current signaling load on the uplink control channel of the base station. Numerical results, under the assumptions of a one-dimensional linear model and random walk mobility patterns, show that the adaptive threshold scheme has better performance than the static timer-based scheme.

3.5 Distance-Based

In the distance-based update scheme, each mobile terminal tracks the distance it has moved (in number of cells) since the last update and transmits an update signal whenever the distance exceeds a certain threshold. For implementation, the mobile terminal requires some knowledge of cell topology. In order to identify the cells within the distance threshold or along the distance threshold boundary, the mobile terminal needs to download a set of these cell IDs after each location update. Even though CIC can be used, its implementation is restricted to some particular paging strategies. The distance-based scheme has been studied extensively. The distance-based update scheme is formulated as an optimization problem. The goal is to minimize the expected total cost for update and paging within a time interval. Under a one-dimensional linear model and symmetric random walk movement patterns, the optimal distance threshold is determined by dynamic programming. An iterative approach is used to compute the optimal distance threshold in a two-dimensional hexagonal model under the assumption of symmetric random walk mobility pattern.

3.6 Predictive Distance-Based

In the predictive distance-based update scheme, the mobile terminal reports both its location and velocity during the **update process. Based on the above information, the network determines the probability density function of the mobile's location, which is used to predict the mobile terminal's location in future** time. This prediction information is made available to both the network and mobile terminal. The mobile terminal checks its position periodically and performs location update whenever its distance exceeds the threshold distance measured from the predicted location. Upon a call arrival, the network pages the mobile terminal starting from the predicted location (which may be the one that performed the last update) and outward, in a shortest-distance-first order, until the mobile terminal is found. For performance analysis, a Gauss-**Markov process is used to model the user's mobility pattern. The** Gauss-Markov model captures the **correlation of the mobile's velocity in time, and can represent different user mobility patterns, including the random** walk and constant velocity fluid-flow models. Numerical results, under the assumptions of an infinite one-dimensional linear model and Poisson call arrivals, show that the predictive distance-based scheme has better performance than the non-predictive one.

3.7 State-Based

In the state-based update scheme, the mobile terminal determines whether to perform location update based on its current state. The state information can include the time elapsed or the number of cell crossings since the last update, the cell distance between the current and last registered locations, or some other criteria. Thus, maintaining different state information corresponds to different location update schemes. The state-based scheme where the system state includes the current location and the time elapsed since the last update. A time-varying Gaussian process is used to model the user's movement. The suboptimal solution for the average cost of location update and paging under no paging delay constraint is obtained by a greedy method. Results show that the state-based update scheme achieves 10 percent improvement in average cost compared to the timer-based scheme.

3.8 LeZi Update

The idea of the LeZi update (pronounced "lazy update") algorithm is based on a compression algorithm proposed by Ziv and Lempel. The LeZi update algorithm can be considered to be a path-based update scheme in which movement history rather than current location is sent in an update message. The movement history consists of a list of zone (i.e., LA or cell) IDs the mobile terminal has crossed after the last update. The network database maintains the movement history in a compact form by a trie or digital search tree. This trie can be considered part of the user's profile. Upon a call arrival, selective paging based on the trie information is used to locate the mobile terminal.

4 PAGING SCHEMES

In attempt to locate target MT as quickly as possible, multiple methods of paging have been proposed by the researchers. The most basic method used is Simultaneous paging, where every cell in the MT's LA is paged at the same time. If LA contains large number of cells, this scheme costs terribly high. In Sequential Paging, cells within an LA are paged one after the other, in order of decreasing user dwelling possibility. If the user resides in an infrequently occupied location, long delay may occur in finding the MT. Intelligent-Paging (optimized version of Sequential Paging) calculates the specific paging areas to poll sequentially, based upon a dwelling probability matrix. However, this scheme has too much computational overhead incurred through updating and maintaining the matrix. Rule-based paging scheme is a knowledge based approach, where the current interaction of the MT with the network is represented as a set of facts, which is used to predict the location of MT when a call arrives.

Table 4.1: COMPARISON BETWEEN MAJOR PAGING SCHEMES

Paging Scheme	Paging Area	Time delay	Paging cost	Major drawback
Simultaneous	whole LA	low	high	• excessive cost for bigger LAs
Sequential	1cell to 1LA	high	depends upon number of paging-miss	• large delay occurs if user resides in an infrequently occupied location
Intelligent	1cell to 1LA	high	depends upon number of paging-miss	• too much computational overhead • long delay occurs if user resides in an infrequently occupied location
Rule based	1cell to 1LA	high	depends upon number of paging-miss	• too much computational overhead • long delay occurs if user resides in an infrequently occupied location

5 Reporting Cell Planning

5.1 Introduction to Reporting Cell Planning

Reporting Cell Planning (RCP) is one approach to Location Management. In RCP, few cells in the cellular network are assigned as reporting cells and the cells other than reporting cells (RCs) are non-reporting cells. This approach to location management was proposed in 1993 by Bar-Noy and Keller. Reporting cells can be adjacent to each other sharing a boundary or scattered in a specified coverage area. Location update is performed whenever a mobile terminal enters a reporting cell. Next location update is done only when the mobile terminal enters or crosses a new reporting cell. During call arrival to a mobile terminal, paging is restricted to the last updated RC where the MT's location update has been last performed and to its neighbouring non reporting cells. In simple terms, we can consider this set of reporting cells as a boundary, where all the call routing and call procedures are carried out. This is explained in detail in the section below.

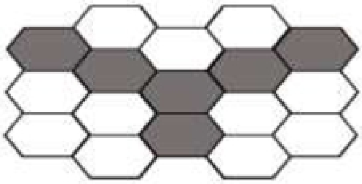


Figure 5.1: Connected RCs

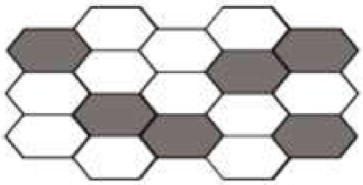


Figure 5.2: Scattered RCs

5.2 Problem Formation

In RCP location management method, mobile terminals need to update their locations whenever they cross a reporting cell. During call arrival, users are located by paging in their last location updated reporting cell as well as its neighbouring non reporting cells without crossing another reporting cell. For example, in the reporting cell configuration shown in Fig 2.8 below, cells 3, 6, 9, 10, 11, 13 and 16 are reporting cells and the others are non-reporting cells. Let us say, an MT's location has been last updated in cell 3. Therefore, when there is a call arrival to thatmobile terminal, paging is done in cells 1, 2, 3, 4, 5, 7, 8 and 12. Location Management cost consists of location update cost and paging cost. Therefore, total cost of a given cellular network is equal to the sum of total number of location updates and total number of paging transactions over a certain period of time.

$$Total\ Location\ Cost = C * N_{LU} + N_P$$

Where, C is a positive constant which represents the cost ratio of location update to paging, and is taken as 10, i.e. C=10, because cost of location update is much higher than paging cost. NLU is the total number of location updates and NP is the total number of paging transactions in the given network. In a given cellular network, each cell i is associated with two weights: movement weight (wmi) and call arrival weight (wci). wmi represents the frequency (or total number) of movement of a MTs into a cell. wci represents the frequency of call arrivals in a cell. The total number of location updates and paging transactions in a network can be calculated using these two weights as follows:

$$N_{LU} = C * \sum_{i \in S} w_{mi}$$

$$N_P = \sum_{j=0}^{N} w_{ci} * v(j)$$

Where N is the total number of cells in the network, S is the set of reporting cells and v(j) is the vicinity value of cell j. Vicinity value is defined as the maximum number of cells that can be searched if an incoming call is received in cell j. Thus, Location management cost for a reporting cell configuration is obtained as:

$$Total\ Cost = C * \sum_{i \in S} w_{mi} + \sum_{j=0}^{N} w_{cj} * v(j)$$

$$Cost\ per\ Call\ Arrival = \frac{TotalCost}{\sum_{j=0}^{N} w_{cj}}$$

Movement and call weights are predefined. The objective is to minimize the cost per call arrival with a trade-off between update cost and paging cost and find the corresponding set of optimal reporting cell set.

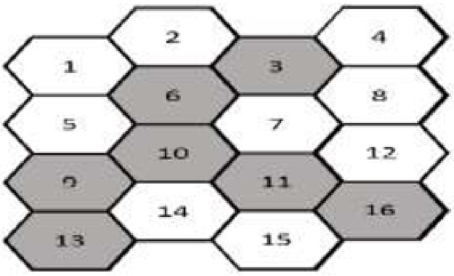


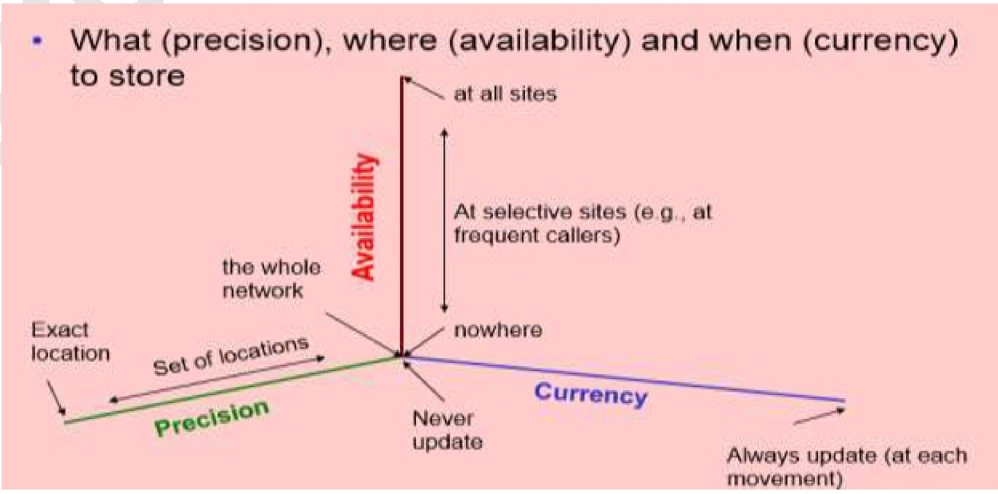
Figure 5.3: Example Reporting Cell Configuration

6 Location Management: Summary

- Our target is objects capable of changing their location
- We are interested in objects with identity
- We store user locations in multiple databases (DBs)
- Main questions:
  - How do we *update* data when user moves?
  - How do we *locate* user in DBs when it is required?

Locating Moving Objects

- Example of moving objects
  - mobile devices (cars, cellular phones, palmtops, etc)
  - mobile users (locate users independently of the device they are currently using)
  - mobile software (e.g., mobile agents)
- How to find their location - Two extremes
  - **Store** their current location everywhere
    - **Search locally**
    - **Cost of updates**
  - **Search** everywhere
    - No information is stored anywhere; search is expensive
    - No cost of updates
- Searching verses Update Cost
- **Availability** – either at all sites or at selective sites (frequently visited sites)
- **Currency** – Stored location is always updated (it may not make sense if user moves very frequently)
- **Precision** – Exact location verses set of possible locations



7 Effect of mobility on protocol stack

- At the data link layer, mobility based on wireless networks brings problems of bandwidth, reliability, and security, for which compression, encryption, and error correction techniques are needed. Other problems include fixed or dynamic channel allocation algorithms, collision detection and avoidance measures, QoS resource management, etc.
- At the network layer, mobility of mobile nodes means that new routing algorithms are needed in order to change the routing of packets destined for a moving node to its new point of attachment in networks. How to track a node's movement and how to keep the moving node's connectivity are two basic issues at the network layer. This in turn forms the two main operations of mobility management.
- At the transport layer, an end-to-end connection may mix wired and wireless links. This makes congestion control a complex task due to the different characteristics of wired and wireless networks, since packet loss is caused mainly by high error rates and handoff in wireless networks instead of because of congestion the situation on wired links. Retransmission mechanism based on increasing interval may lead to an unnecessary drop in the date rate. Function distribution between the transport and the data link layer is a new problem caused by mobility.
    - Application
        - new applications and adaptations
    - Transport
        - congestion and flow control
    - Network
        - addressing and routing
    - Link
        - media access and handoff
    - Physical
        - transmission errors and interference

8 Application Adaptations for Mobility

- System-transparent, application-transparent
    - **the conventional, "*unaware*" client/server model**
- System-aware, application-transparent
    - the client/proxy/server model
    - the disconnected operation model
- System-transparent, application-aware
    - dynamic client/server model
    - data broadcasting/caching
- System-aware, application-aware
    - the mobile agent model

9 World Wide Web and mobility

- Protocol (HTTP, Hypertext Transfer Protocol) and language (HTML, Hypertext Markup Language) of the Web have not been designed for mobile applications and mobile devices, thus creating many problems!
- Typical transfer sizes
    - HTTP request: 100-350 byte
    - responses avg. <10 kbyte, header 160 byte, GIF 4.1kByte, JPEG 12.8 kbyte, HTML 5.6 kbyte
    - but also many large files that cannot be ignored
- The Web is no file system
    - Web pages are not simple files to download
    - Static and dynamic content, interaction with servers via forms, content transformation, push technologies etc.

- many hyperlinks, automatic loading and reloading, redirecting
- a single click might have big consequences!

❖ **HTTP 1.0 and mobility**

➢ **Characteristics**
- stateless, client/server, request/response
- needs a connection oriented protocol (TCP), one connection per request (some enhancements in HTTP 1.1)
- primitive caching and security

➢ **Problems**
- designed for large bandwidth (compared to wireless access) and low delay
- large and redundant protocol headers (readable for humans, stateless, therefore large headers in ASCII)
- uncompressed content transfer
- using TCP
  - huge overhead per request (3-way-handshake) compared with the content, e.g., of a GET request
  - slow-start problematic as is without having to deal with the wireless problem
- DNS lookup by client causes additional traffic and delays

➢ **Caching**
- quite often disabled by information providers to be able to create user profiles, usage statistics etc.
- dynamic objects cannot be cached
  - numerous counters, time, date, personalization, ...
- mobility quite often inhibits caches
- security problems
  - caches cannot work with authentication mechanisms that are contracts between client and server and not the cache
- today: many user customized pages, dynamically generated on request via CGI, ASP, ...

➢ **POSTing (i.e., sending to a server)**
- can typically not be buffered, very problematic if currently disconnected

❖ **HTML and mobile devices**

➢ HTML
- **designed for computers with "high" performance, color high**-resolution display, mouse, hard disk
- typically, web pages optimized for design, not for communication

➢ Mobile devices
- often only small, low-resolution displays, very limited input interfaces (small touch-pads, soft-keyboards)

➢ **Additional "features"**
- animated GIF, Java AWT, Frames, ActiveX Controls, Shockwave, movie clips, audio, ...
- many web pages assume true color, multimedia support, high-resolution and many plug-ins

➢ Web pages ignore the heterogeneity of end-systems!
- e.g., without additional mechanisms, large high-resolution pictures would be transferred to a mobile