Adaptive Location Management in Mobile Environments

Ratul kr. Majumdar[1], Krithi Ramamritham[1], and Ming Xiong[2]

[1] Laboratory for Intelligent Internet Research
Department of Computer Science and Engineering.
Indian Institute of Technology Bombay
Mumbai, India 400076
{ratul,krithi}@cse.iitb.ac.in
[2] Lucent Bell Laboratories
Murray Hill, NJ 07974
xiong@research.bell-labs.com

Abstract. Location management in mobile environments consists of two major operations: location update and paging. The more up-to-date the location information, the less paging becomes necessary and vice versa. The conventional approach is the location area based approach (LA-based approach), where a location area (LA) consists of multiple cells. When the mobile station (MS) enters a new location area, the MS immediately updates its location information at the new location's visitor location register (VLR) and this update is propagated to the MS's home location register (HLR). The major drawback of the LA-based approach is that it does not consider any mobility patterns, or call arrival patterns. Moreover, MS updates frequently when it roams only within the boundary cells of different location areas, resulting in unnecessary location updates. So, there is a need for an efficient algorithm which can eliminate the drawbacks of the LA-based approach. To this end, an adaptive location management algorithm is described in this paper: an MS dynamically determines whether or not to update when it moves to a new LA, so that each location update becomes a necessary location update, i.e., in each updated location area at least one call is made. We have conducted experiments to capture the effect of mobility and call arrival patterns on the new location update strategy. We have also tested our algorithm with SUMATRA (Stanford University Mobile Activity TRAces), which has been validated against real data on call and mobility traces. Experimental results show that our adaptive location management algorithm considerably reduces the location management cost, by avoiding unnecessary location updates.

1 Introduction

Location management in a personal communication services (PCS) network environment deals with the problem of tracking mobile users. A cellular wireless service area is composed of a collection of cells, each serviced by a base station (BS). A number of base stations are wired to a *mobile switching center* (MSC). The backbone of a PCS network consists of all these BSs, MSCs and the existing wired network (such as PSTN, ISDN etc.) between them. This MSC acts as a gateway for the base stations.

In order to route an incoming call to a PCS user, his (or her) whereabouts in the cellular network needs to be determined within a fixed time, which may be specified as a QoS (quality of service). Location management as a whole involves 2 kinds of activities. An Mobile Station (MS) can report his presence as soon as it enters a new location area. This activity, initiated by the MS to limit the search space is called a *location update*. The system can initiate the search for an MS by paging, i.e., simultaneously polling all the cells where the MS can be possibly found, when a new call is made for the MS.

Both paging and location update consume scarce resources like wireless bandwidth, and each has a significant cost associated with it. The mechanism of paging involves the MSC sending a page message over a forward control channel to all the cells where the MS is likely to be present. The MS listens to the page message and sends a response message back over a reverse control channel. The location update mechanism involves the MS sending a message regarding its current location over the reverse control channel, which helps in limiting the search to route a call to the MS.

1.1 Location Update Strategy in the LA-based approach

In this section we will discuss the location update strategy, focusing on the different entities of Global System for Mobile Communication (GSM) systems.

Elements and Architecture of GSM The area covered by the mobile radio network is divided into zones, which in turn, are broken into cells, each supported by a Base Transceiver Station (BTS). Several BTSs form what is termed a location area (LA). The GSM network consists of three basic subsystems:

- Mobile Station (MS)
- Base Station Subsystem (BSS)
- Network Subsystems (NSS)

The BSS is further split into base transceiver stations (BTSs) and base station controllers (BSC). In the NSS, two main elements can be identified, mobile switching center (MSC) and location registers (HLRs and VLRs).

Figure 1 shows the main entities of GSM systems. A brief description of functions carried out by the different elements in the GSM network is given below.

BTS: This unit comprises the radio transceivers that make the link with the MS possible. A BTS includes a number of transceivers (TRX) which serve a single cell (omni-directional coverage) or several cells (sectorized coverage).

BSC: The element is responsible for most of the control functions related to a set of base stations. The BSC manages the radio sources of the system, assigning channels, deciding on hand-overs, etc.

MSC: It performs functions similar to those of a conventional telephone exchange, such as call routing. It also acts as an interface with the Public Switched Telephone Network (PSTN) or Integrated Services Digital Network (ISDN). MSC also acts as a bridge between BSCs for call hand-overs. In addition, the MSC is the main element involved in the mobility management of MSs. It also gives access to/from fixed telephone networks. To carry out mobility management and routing functions, MSCs are complemented with the two databases (registers) HLR and VLR.

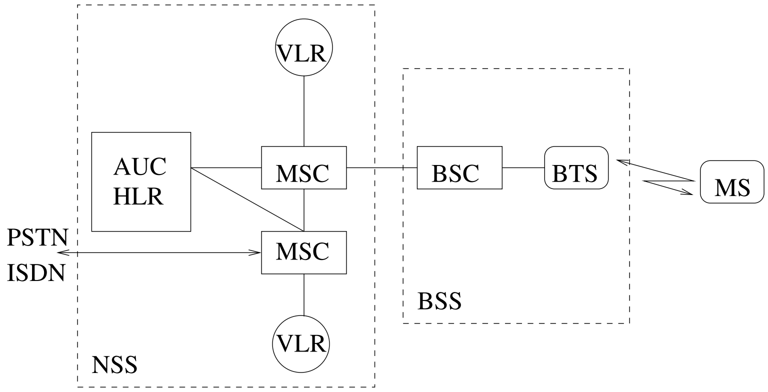


Fig. 1. Main Entities in GSM

HLR: It contains information about the network subscribers and the region in which they are located. This location register is updated when the MS is identified in the coverage of a different MSC. The authentication center (AUC) associated with a HLR contains the required security data for the user authentication process.

VLR: It contains information on all MSs located within its associated LA, so that these MSs may be paged and call may be routed towards their actual position in the network. Information about a visiting MS is deleted from the database when the MS moves to a new LA.

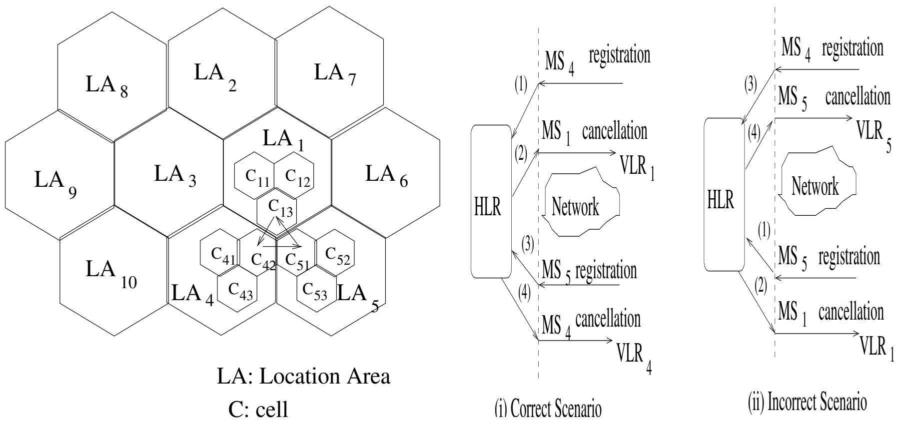
Location Update Location update is initiated whenever one of the following events occurs.

- When an MS moves from an LA to another LA.
- At the time of connection/disconnection (ATTACH/DETACH).

BSS at regular intervals broadcasts the ID of its LA. An MS verifies its Location Area ID (LAI) periodically. In the event of an change, a location update is initiated. If the MS is turned off, it sends a “DETACH” message. When it is switched on again, it sends an “ATTACH” message.

Location update is initiated by the MS when, by monitoring Broadcast Control Channel, it notices that the LAI broadcast is not the same as that previously stored in the MS’s memory. An update request and the International Mobile Subscriber Identity (IMSI) or previous Temporary Mobile Subscriber Identity (TMSI) is sent to the new VLR via the new MSC. A Mobile Station Roaming Number (MSRN) is allocated and sent to the MS’s HLR (which keeps the most current location) by the new VLR.The MSRN helps to route the call to the new VLR and subsequently translated to the TMSI of the MS. The HLR sends back a cancellation message to the previously visited VLR, so that the previous MSRN can be reallocated. Finally, a new TMSI is allocated by HLR and sent to the MS, to identify it in future paging or call initiation requests.

1.2 Why do we need an alternative to the conventional approach?



(a) Location Area and Cells within it (b) Location Update problem

Fig. 2. A Sample Mobility Scenario

Consider Figure 2(a). Let us assume an MS starts roaming from location area LA_1 to location area LA_4 and then to location area LA_5. An MS will receive a call if HLR stores the location information about the MS and VLR stores correct location of the MS. In the Figure 2(b)(i) the correct scenario is as follows.

1. While the MS enters location area LA_4 from location area LA_1, MS sends location information to VLR_4. VLR_4 sends location information to HLR.
2. HLR updates this location information and sends cancellation message to VLR_1 via MSC.
3. While the MS enters LA_5 from LA_4, MS sends location information to VLR_5.
4. HLR updates this location information and MSC sends cancellation message to VLR_4 via MSC.

Because of delays in network connecting VLRs and HLR, HLR may receive location update message from MSC of LA_5 before receiving message from MSC of LA_4. As HLR processes the messages in the order of messages received, HLR shows that MS is in location area LA_4 but actually MS is in location area LA_5. Thus, HLR and VLR may contain incorrect location information of an MS (we can refer to these as orphan records), and because of this MS may not receive calls. The order of events of an incorrect scenario is shown in the Figure 2(b)(ii).

Let us once again consider Figure 2(a). It shows several location areas and the cells within them. Location area consists of multiple cells. So, if an MS roams within the boundary cells, we now show that there might be a significant number of unnecessary location updates.

Let us consider the movement of an MS within the cells C_{13} in LA_1, cell C_{51} in LA_5 and the cell C_{42} within LA_4 as in Figure 2(a). We define a location update in an

LA to be *necessary* if at-least one call is made in that LA. Since the MS is roaming around the boundary cells of the different LAs frequently, each location update in LA_1, LA_2 and LA_3 may not be necessary, as the MS may not receive a call when it is in an LA, resulting in increase in the number of unnecessary location updates. If an MS is in a cell, which is well inside an LA, this problem does not arise.

So, we address specifically 2 problems (1) avoidance of orphan records from HLR and VLR and (2) optimized location management avoiding unnecessary location updates.

Our contribution lies in the development of 1) the MS-counter approach which ensures the avoidance of orphan records from HLR and VLR and 2) the MS-dynamically-adjustable-counter (MSD) approach, which dynamically determines whether or not to update in each LA so that each location update becomes a necessary location update.

2 Related Work

Most of the practical cellular mobile systems partition the geographical region into location areas (LA) and users update on entering a new LA. In this section we will discuss proposed location update strategies and paging methods.

Work proposed in [9] presents a mobility management scheme that integrates the location area approach with the location prediction idea. It is based on results from traffic flow theory. [5] presents a look-ahead strategy for distance-based location tracking so that the rate of location update can be reduced without incurring extra paging costs. A mobile tracking scheme that exploits the predictability of user mobility patterns in wireless PCS networks is proposed in [1].

A location management algorithm is proposed in [6], which uses the mobility history of individual subscribers to dynamically create individualized location areas, based on previous movement from cell to cell. and the average duration spent in each visited cell.

Paging schemes proposed in [8] provide a way of partitioning the service areas and accordingly minimizing the paging costs. The paging scheme proposed in [2] is based on the semi-real time movement information of individual users.

A mobile user location management mechanism is introduced in [4], that incorporates a distance based location update scheme and a paging mechanism that satisfies predefined delay requirements. Given the respective costs for location update and terminal paging, the average total location update and terminal paging cost is determined. An iterative algorithm is then used to determine the optimal location update threshould distance that results in minimum cost.

A static location Update strategy for PCS networks and its implementation using a genetic algorithm is proposed in [7] . The location update strategy proposed in [7] determines whether or not to update in each LA, and optimizes the average location management cost derived from user-specific mobility model and call generation patterns. But the optimization using genetic algorithms is computationally intensive

Also the ratio of cost of update (C_u) and cost of paging (C_p) is required to be known beforehand as this ratio C_u/C_p is important to determine whether to update or not to update in an LA. As the ratio C_u/C_p may also vary depending on mobile communication network topology it is difficult to get the value of C_u/C_p beforehand. So, we need an algorithm which can capture user's mobility patterns and call arrival patterns and adjust the location update frequency without the knowledge of the ratio C_u/C_p.

3 Counter based Algorithms

In the LA-based approach, an MS updates its position when it enters a new location area. The MS-counter approach is a simple implementation of this approach, which handles the first problem, i.e., it deletes orphan records from HLR and VLR. Then MS-counter approach is then extended to develop the MS-Dynamically-Adjustable-Counter (MSD) approach to handle the second problem, reducing location management cost.

3.1 MS-counter

In this approach, the MS updates its position as and when it enters a new Location area. An MS maintains a *c*ounter and increments the *c*ounter on entering each new LA. After the *c*ounter reaches a maximum value, it is reset to 0. In this approach, both location registers HLR and VLR maintain information about the value of the *c*ounter that each MS sends in the process of location updates. On entering a new LA, MS sends an update request with the value of its *c*ounter to the new VLR via the new MSC. A Mobile Station Roaming Number (MSRN) is allocated and sent to the MS's HLR (which always maintains the current location) by the new VLR. If HLR receives a registration message from a VLR with a *c*ounter value less than what it has received earlier, the HLR ignores this registration message and sends a cancellation message to the VLR.

Illustration:

Consider Figure 2(a), which shows that an MS starts roaming from location area LA_1 to location are LA_4 and then to location area LA_5.

Suppose the current *c*ounter value is 3. When the MS enters a new location area LA_4 , it increments the *c*ounter value to 4 and sends it to VLR_4, and VLR_4 sends this to HLR via MSC.

When the MS moves to another location area LA_5, it increments the *c*ounter value to 5 and sends this *c*ounter value to VLR_5, and VLR_5 sends this to HLR via MSC.

Let us consider the following situation, where update information from location area LA_5 reaches HLR before update information comes from location area LA_4 because of delays in the network connecting VLRs and HLR. Normally, the HLR will simply update its database with what it gets from the location area LA_4, but the location update information that HLR stores is wrong.

But in our approach HLR compares the *c*ounter values. Since HLR receives *c*ounter value 5 before *c*ounter value 4, it simply ignores the information regarding the *c*ounter value 4 and sends the cancellation message to VLR_4 via MSC.

3.2 MS-counter(k) Approach

In the MS-counter(k) approach, MS updates its location information after moving through k location areas. The value of k is pre-defined.

Between two successive location updates by MS, it is the HLR's responsibility to find out the location information of MS by paging in the following way.
HLR sends paging message to the last updated LA of the MS (this is the location area where the MS updates the last). If HLR does not receive any response within time-out from MS, HLR finds location of MS following the scheme as described below.

– HLR sends paging message to the most probable LA, if HLR does not receive any response within time-out, it multicasts to all other probable LAs (all the neighbouring LAs of the last updated LA) serially ring by ring till HLR receives response from the MS. We define a ring of distance d as a set of LAs, which are separated by *(*d-1) number of intermediate rings. So, the ring of distance one is the set of LAs which are immediate neighbouring LAs of the last updated LA.

Subsequently, the probability of movement to the boundary cell of different location areas are calculated in the following manner. HLR maintains in the user profiles a table, in which each tuple has the following attribute: ($Previous_{LA}$, $Current_{LA}$, $Future_{LA}$ and probability associated with $Future_{LA}$).

We can consider each LA as a state. While an MS moves from one LA to another we can view this as a state transition. So, we can use a learning automaton to find out the probabilities associated with $Future_{LA}$s.

Generally, the operation of the learning automaton is based on the state transition matrix, which contains the one step transition probabilities P_{ij} from the current state *i* to the next state *j*. When the automaton selects the right response, the positive transition is rewarded while the probabilities of state transition that were not selected are penalized to keep probability 1.

The probabilities associated with the $future_{LA}$s are calculated following linear-reward-penalty scheme:

If HLR does not get any response from MS in its last updated LA, say LA_i and also MS has not sent a "DETACH" message (MS sends this message to HLR when it is switched off), HLR needs to find the location of MS by sending paging message over forward control channel. The HLR pages LA_j for finding MS. If transition from LA_i to LA_j receives a positive feedback (if response comes from the MS) then equations for evaluating the new probabilities of finding MS in the location areas are

P_{ij}= P_{ij}(1-A) + A and P_{ik}= P_{ik}(1-A), if $k \neq j$.

P_{ij} are the probabilities of transition from LA_i to LA_j and P_{ik} are the probabilities of transition from LA_i to LA_k. Here A is a tuning parameter which speeds up this linear-reward-penalty scheme. The value of A between 0 and 1.

If transition from LA_i to LA_j receives a negative feedback (if response does not come from the MS) then equations for evaluating the new probabilities of finding MS in the location areas are

P_{ij}= P_{ij}(1+A) - A and P_{ik}= P_{ik}(1+A), if $k \neq j$.

If P_{ij} becomes negative then it is set to 0. And if P_{ik} becomes greater than 1 then it is set to 1.

If the probability P_{ij} attains the maximum value, MSC pages all the cells in the location area, LA_j, simultaneously as in location area paging method. Otherwise, the MSC starts paging in the probable location areas simultaneously.

Consider Figure 2(a). Let us assume the last updated LA is LA_1. LA_2, LA_3, LA_4, LA_5, LA_6 and LA_7 are the neighbouring LAs to LA_1. The probabilities of transition from LA_1 to LA_2, LA_3, LA_4, LA_5 LA_6 and LA_7 are P_{12}, P_{13}, P_{14}, P_{15}, P_{16} and P_{17} respectively. In order to route a call to an MS, HLR sends a paging message to LA_1. If HLR does not get any response from LA_1, HLR sends a paging message to the most probable LA, say LA_2. If MS is in LA_2, HLR gets a positive feedback and changes P_{12} to P_{12}(1-A) +A. HLR multiplies transition probabilities P_{13}, P_{14}, P_{15}, P_{16} and P_{17} by (1-A) so that P_{12}, P_{13}, P_{14}, P_{15}, P_{16} and P_{17} sum to 1.

HLR gets a negative feedback (If MS is not in LA_2), it changes P_{12} to P_{12}(1+A)-A. HLR multiplies other transition probabilities by (1+A).

4 Performance of Counter-based Algorithms

In this section we will discuss the simulation environment and metrics used to evaluate the proposed algorithms.

4.1 Simulation Environment

A service area is considered to be made of eight LAs. The model for the network is taken as a graph with eight nodes and eleven edges as depicted in Figure 3(a). This model is also considered in [7]. The diameter of the graph is three. Each node in the graph is considered as an LA and edge as an interconnection between the LAs. We also consider a second network topology, which is considered to be made up of eleven nodes and nineteen edges as depicted in Figure 3(b). The diameter of this graph is four.

Call inter-arrival rate is considered as exponentially distributed with a mean rate λ. varied from .5 to 2.5 with increment of .5. The total number of calls in each experimental set is 1000.

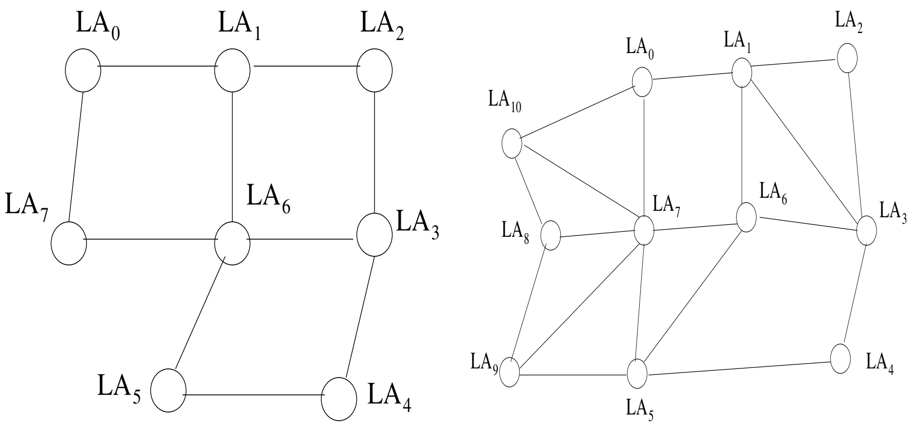
Movement of MS is modeled as discrete time random-walk model with some directional component. This model has been taken as an MS has certain directional component in its movement resulting in higher location probabilities in certain LAs than others.

- At discrete time t, an MS moves to one of the neighbouring LAs with probability q or stays at the current LA with the probability 1-q.

As compared to fluid flow model reported in [3], the random walk model with directional component is appropriate especially when most of the mobile subscribers in a PCN are likely to be pedestrians. Fluid flow model is more suitable for vehicle traffic such that continuous movement with infrequent change in speed and direction is expected.

Location area based paging wherein all the cells in a LA are simultaneously paged is used in the experiments. Paging cost has been assumed to be the same in all the LAs.

As cost of location update (C_u) and cost of paging cost (C_p) are usually not known, we consider different ratios of C_u and C_p from 0.1 to 2 with increment of 0.1 for our experiments.



(a) First Network Topology (b) Second Network Topology

Fig. 3. Location Area Graph

4.2 Metrics

1. Location Management Cost (LMC) vs. call-to-mobility-rate (CMR) for constant C_u/C_p.
2. Location Management Cost (LMC) vs. C_u/C_p for constant call-to-mobility-rate (CMR).

The meaning of all symbols used in the metrics are given in Table-1.

4.3 Experimental Results

Experiments have been done with counter(k) with k = 1, 2, 3, 4 and 5. Figure 4 shows that MS-counter(k) for k = 1 produces the least cost for all CMR values. We see the significant variations for k = 2, 3, 4 and 5. Since Figure 4 does not indicate a trend in the location management cost vs. CMR for different values of k, it implies that a static value of k cannot optimize the location management cost. Thus, we need to vary k depending on CMR. MS-dynamically-adjustable-counter approach, which we discuss next, dynamically finds the value of counter, responding to different call arrival patterns and mobility patterns.

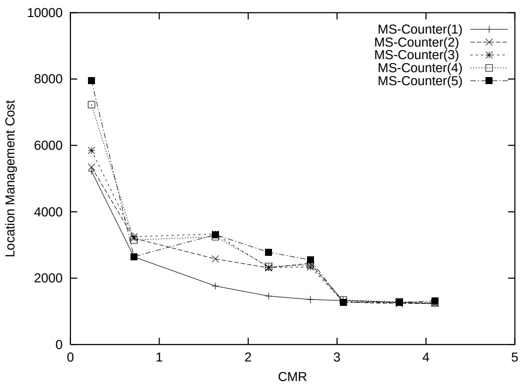


Fig. 4. Location Management Cost of MS-counter(k) for Cu/Cp = 1

Symbol	Meaning
C	Number of calls processed for each set
M	Total number of movements of MS from one LA to a new LA during C number of calls processed
n_u	Number of location updates by MS during C number of calls processed
n_p	Number of LAs paged during C number of calls processed
C_u	Location update cost in a location area
C_p	Paging cost in a location area.
CMR	Call to mobility rate, which is Ratio of C and M
TLUC	Total location update cost, $n_u C_u$
TPC	Total paging cost, $n_p C_p$
LMC	Location Management cost, TULC + TPC

Table 1. Meaning of Symbols used in Metrics

5 MSD: MS-dynamically-Adjustable-Counter approach

In previous 2 counter based algorithms, the counter whose value is used to decide when to update, takes a static value. But in the MSD approach the counter changes dynamically. The basic principle underlying this approach is to predict the number of useless updates based on history of mobile user's movement pattern.

Let us consider the following example of the combined call and movement pattern of (C, C, M, C, C, C, C, C, M, M, M, M, C) where C denotes a call and M denotes a movement of mobile user from a LA to a new LA. If between two calls there is a single movement then the location update in the new LA is useful. We see that from the combined call and movement pattern that there is single M between Cs, so the number of useless updates is 0. Since for the second sequence of Ms there are four Ms between two Cs, the number of useless updates is three. The protocol is as follows.

MS maintains one variable *c*ounter and one flag *c*all-came.

The value of counter is bounded within ($counter_{min}$, $counter_{max}$). The value of $counter_{min}$ is 1. The value of the $counter_{max}$ is dynamically set to the number of predicted useless updates in future, as calculated by the following formula.

$counter_{max}$ = max(U_{min}, min(U_{max}, U_{est})

Where U_{min} is the minimum number of the useless updates observed so far, U_{max} is the maximum number of the useless updates observed so far, and U_{est} is the estimated useless updates. The above formula keeps the value of $counter_{max}$ between U_{min} and U_{max}. If the value of U_{est} is greater than U_{min} and less than U_{max} then the value of $counter_{max}$ is set to the value of U_{est}. All the values of U_{max}, U_{min} and U_{est} dynamically change. But if U_{min} is set to 0, there will not be any change in its value. U_{est} is calculated from the following formula.

U_{est} = (pU_{recent} + mU_{avg} + (1-p-m)U_{min})

where U_{recent} is the most recent number of useless updates, U_{avg} is the average of three most recent number of useless updates and *p* and *m* are the tuning parameters.

The prediction of number of useless updates by MS in future needs to track the pattern of changes of number of useless updates. U_{recent} and U_{avg} help tracking these changes in the future.

If a call is not made in an updated LA (where MS updates its location information) counter gets incremented by 1. MS increments counter up to the value of dynamically calculated $counter_{max}$, which actually denotes the predicted number of useless updates in future.

MS starts the first registration with counter being equal to 1. But if a call is not made in this location area, MS increments its counter by 1 on entering a new location area. MS updates its location information after moving through $counter_{current}$ - 1 number of location areas from its last updated location area, or in other words MS skips location updates in the next $counter_{counter} - 1$ number of location areas. $counter_{current}$ is the present value of counter. In this way MS increments its counter linearly, if call is not made in its updated location area till counter reaches $counter_{max}$.

If a call is made in a location area but MS has not updated its location information in this location area then MS sets its counter to 1.

Between two updates by MS, HLR finds the location of MS to route the call as given in MS-counter(k) Approach. (section 3.2).

Illustration:

If $counter_{current}$ is 1 before entering a new location area, MS updates its location information on entering new location area. If call is not made in this updated location area MS increments its *c*ounter value by 1. Thus the value of $counter_{current}$ becomes 2. MS will update its location information after moving through $counter_{current}$ -1 or one location area or in other words MS will skip location update in the next new location area.

If $counter_{current}$ is 3 and call is made in a location area where MS has not updated its location information, then MS sets the *c*ounter to 1. So, MS will update its location information on entering the next new location area.

6 Performance of MSD approach

For the evaluation of adaptive algorithms the same simulation environment and metrics have been used as mentioned in subsection 4.1 and 4.2.

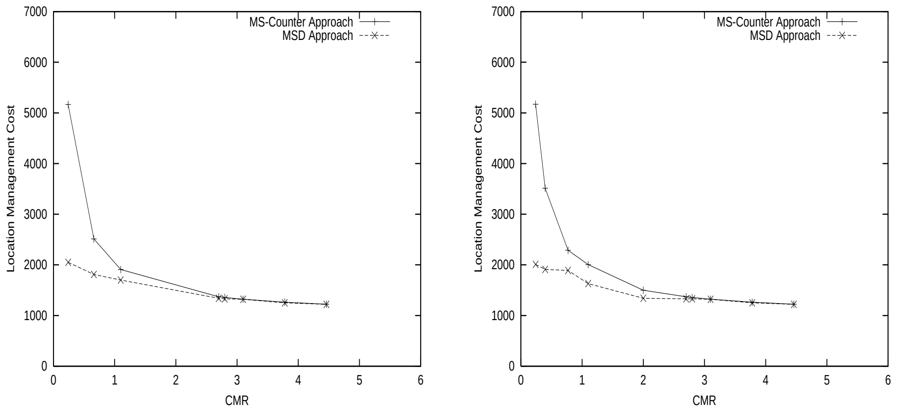
6.1 Experimental Results with Trace (Random Walk with Directional Component)

Figure 5(a) shows the comparison of location management cost vs. CMR for C_u/C_P =1 for the first network topology as depicted in Figure 3(a). We see from the Figure 5(a) that we can save location management cost in MSD approach for CMR $<$ 2.7. We also see from the Figure 5(a) that we can save as much as 60 % location management cost in MSD approach in the range of CMR from .24 to 1.1. Figure 5(b) shows the comparison of location management cost for the second network topology as depicted in Figure 3(b). We see from Figure 5(b) that we can save location management cost, which is as much as 58%, for CMR $<$ 2.6.

Figure 6(a) shows the comparison of LMC vs. C_u/C_p for CMR =.661. We see from the Figure 6(a) that for C_u/C_p $<$ 0.3 both the MSD approach and MS-counter incur almost the same cost. But in the range of C_u/C_p from 1 to 2 we can save significant cost as much as 60 %.

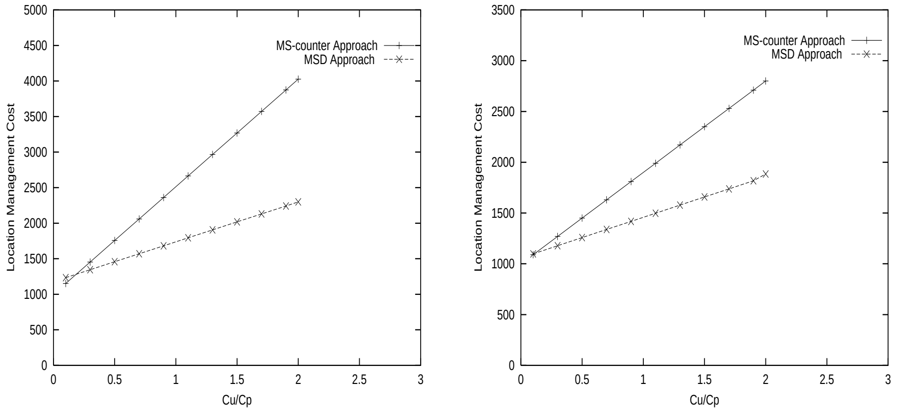
Figure 6(b) shows the comparison of LMC vs. C_u/C_p for CMR =1.1. We see from the Figure 6(b) that for C_u/C_p $<$ 0.35 both the MSD approach and MS-counter incur the same cost. But in the range of C_u/C_p from 1 to 2 we can save as much as 60% location management cost.

So, from the Figures 6(a) and 6(b) we can conclude that for C_u/C_p $<$ 0.35 MSD approach is as good as MS-counter approach. For larger values of C_u/C_p we can save location management cost as much as 60%.



(a) First Network Topology (b) Second Network Topology

Fig. 5. Cost vs. CMR for C_u/C_p = 1



(a) CMR =.661 (b) CMR=1.1

Fig. 6. Cost vs. Cu/Cp for the First Network Topology

6.2 Experimental Results with SUMATRA Trace

SUMATRA (Stanford University Mobile Activity TRAces) is a trace generator that encompasses several calling and mobility models. The unique aspect of SUMATRA is that it is validated against real data on calling and mobility traces.

We specifically use a trace named BAL1-2 (Bay Area Location Information (real-time) available at [10].The trace duration is 24 hours. Table 2 shows the information about the trace.

Figure 7 shows LMC vs. CMR for C_u/C_p =2. We see from the Figure 7 that we can save LMC using the MSD approach for CMR from 1.59 to 7.34. The saving of LMC can be as much as 16 % in the range of CMR from 1.59 to 4.1. If CMR is less than 1.59, then the saving of cost may be more than 16% .

#Zones	Users per zone	Calls per hour	Moves per hour	Callee Distribution
90	1, 100	2	0.3	80 % local, 20 % random

Table 2. Value of Trace parameter

Parameter	Definition	Value
C	Number of calls processed for each set	5000
p	weight associated with U_{recent} in MSD approach	0.19 - .2
m	weight associated with U_{avg} in MSD approach	.8 - .85

Table 3. Parameters used in Experiment

Figure 8(a) shows LMC vs. C_u/C_p for CMR = 1.59. We see from the Figure 8(a) that for $C_u/C_p < 0.5$ both the MSD approach and MS-counter incur the same cost. But in the range of C_u/C_p from 1 to 2 significant cost saving seen: as much as 18 %.

Figure 8(b) shows LMC vs. C_u/C_p for CMR =4.1. We see from the Figure 8(b) that for $C_u/C_p < 0.5$ both the MSD approach and MS-counter incur the same cost. But in the range of C_u/C_p from 1 to 2 we can save significant cost, which is as much as 13 %.

So, from the Figures 8(a) and 8(b) we can conclude that for $C_u/C_p < 0.5$ MSD approach is as good as MS-counter approach. For larger values of C_u/C_p, we can save as high as 16% of the location management cost.

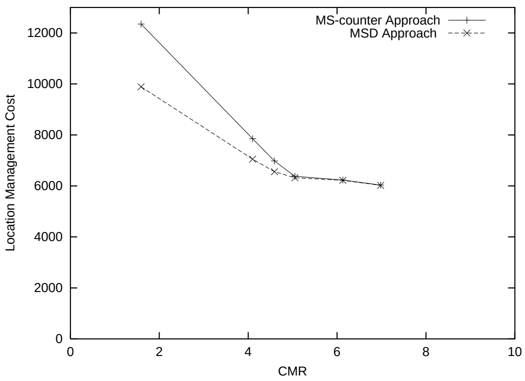
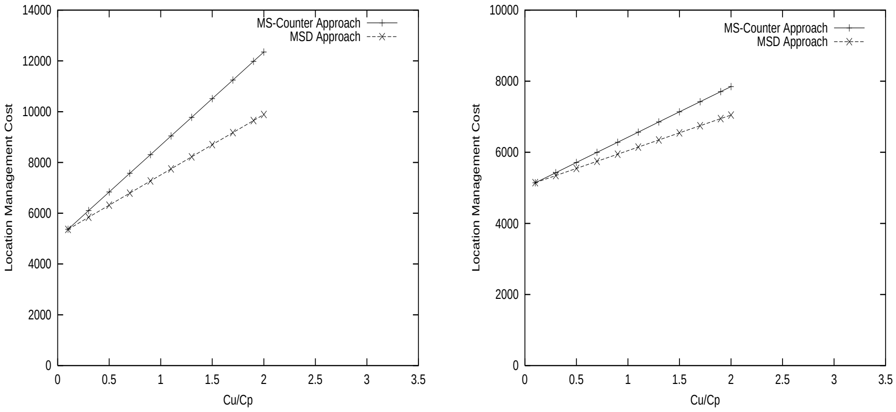


Fig. 7. Cost vs. CMR for C_u/C_p = 2; SUMATRA Trace

6.3 Summary

Typically, location area consists of multiple cells. All MSs update only at the boundary cells of the location area. In the LA-based approach, the update traffic is significant at



(a) CMR= 1.59; SUMATRA Trace (b) CMR = 4.1; SUMATRA Trace

Fig. 8. Cost vs. Cu/Cp

the boundary cells of the location area. The wireless traffic is generated in the following cases at the boundary cells of location areas.

- Location update traffic.
- call setup traffic, when the call is generated from the boundary cells of the location area.

Due to this considerable amount of wireless traffic at the boundary cells of the location area, the LA-based approach does not distribute the wireless traffic in different cells evenly. This is the one of the bottlenecks of the LA-based approach. Our simulation experimental results show that we can save as much as 60% location management cost for CMR from .24 to 1.1. The expermental results on SUMATRA show that we can save location management cost as much as 16% for CMR from 1.59 to 4.1. So, we can conclude that MSD approach considerably reduces the location management cost by avoiding unnecessary location updates.

7 Effect on Existing Location Management Protocols

The MSD approach is simply a refinement of the LA-based approach, aimed at eliminating unnecessary location update traffic.

The flow of messages in location update strategy approach from an MS required for the implementation of the MSD approach is the same as the existing strategy in GSM as discussed in section 1.1.2.

In MSD approach, HLR sends the paging message in the same manner as in LA-based approach, (i.e., page all the cells in the Location Area). HLR sends the message to the MSC of the future location area to page all the cells of the $future_{LA}$ whose probability of transition from the $current_{LA}$ is maximum. If MS is not found there, HLR sends a message to MSCs to page the neighbouring LAs of the $current_{LA}$ ring by ring serially until the intended MS is found.

MSD approach can be implemented in the existing mobile communication networks by software up-gradation in both HLR and MS. We can implement MSD approach in existing mobile communication networks to save significant amount of wireless traffic, a scarce resource, and transmitting power of MS, by paying the price for software up-gradation.

8 Conclusions and Future Work

Our MS-counter approach ensures the deletion of orphan records in HLR and VLR in LA-based approach. In the LA-based approach unnecessary location updates consume lot of wireless bandwidth and transmission power of MS. Thus, we motivated the need for an efficient algorithm which dynamically evaluates the condition that determines whether or not to update in each LA. MSD approach behaves like MS-counter approach when at-least one call is made in each updated LA. In case of frequent movement of MS between different location areas it saves considerable amount of wireless bandwidth by adjusting the update frequency. From the experiments it can be concluded that MSD approach reduces total location management cost avoiding unnecessary location updates. MSD approach saves location management cost as much as 16% for CMR from 1.59 to 4.1 and 60% for CMR from .24 to 1.1.

We can improve the performance of our approach by predicting 1) number of useless updates in future more accurately using the standard prediction techniques and 2)the most probable LA using hidden markov model (HMM). Exploring these possibilities is left as a future work.

References

1. Ben Liang and Zygmunt J. Haas, Predictive Distance-Based Mobility Management for PCS Networks, *IEEE INFOCOM'99, March 21-25, 1999.*
2. Guang Wan and Eric Lin, Dynamic Paging Scheme for Wireless Communication Systems , *Mobile Computing and Networking, 1997.*
3. H.Xie, S. Tabbane and D.Goodman, Dynamic Location Area Management and Performance analysis , *In Proceedings of IEEE, VTC, May 1993.*
4. Ian F. Akyildiz and Joseph S.M. Ho, A Mobile User Location Update and Paging Mechanism Under Delay constraints, *ACM Wireless Networks, 1995.*
5. I-Fei Tsai and Rong-Hong Jan, The Look-ahead Strategy for Distance-Based Location Tracking in Wireless Cellular Networks , *IEEE Real-Time Systems Symposium, Madrid, Spain, December 2-4, 1998.*
6. John Scourias and Thomas Kunz, A dynamic Individualized Location Management Algorithm , *10th International World Wide Web Conference, Hong Kong, May 2001.*
7. Sanjoy k. Sen , Amiya Bhattacharya and Sajal K. Das, A Selective Location Update Strategy for PCS users, *IEEE Network Magazine, 14(3):6-14, May 2000.*
8. Wenye Wang, Ian F. Akyildiz and Gordon L. Stuber, Effective Paging Schemes with Delay Bounds as QoS Constraints in Wireless Systems , *In proceedings of 1996 USENIX Technical Conference, January 1996.*
9. Yigal Bejerano, Israel Cidon, Location Management Based on Moving Location areas, *IEEE Infocom 2001.*
10. http://www-db.stanford.edu/sumatra/.