Multi HLR Architecture for Improving Location Management in PCS Network

Rajeev R. Kumar Tripathi
SSSIST
Sehore (M.P) India

G. S. Chandel
SSSIST
Sehore (M.P) India

Ravindra Gupta
SSSIST
Sehore (M.P) India

ABSTRACT

This paper proposes a multi HLR scheme for location management in PCS network instead of having a single HLR. In conventional architecture we have a single HLR that acts as centralized database to store the user profile along with the MT's location information. This approach of single HLR suffers from two major disadvantages: bottleneck and call misrouting during the peak load. In multi HLR architecture, we store the users profile and MT's location information zone wise or area wise. By doing so, we minimize the possibility of both bottleneck and call misrouting. In conventional architecture, we use explicit de-registration scheme for de-registration of an MT from a VLR on move. Several de-registration schemes has proposed previously in spite of explicit de-registration scheme as: implicit de-registration scheme, polling de-registration scheme, timeout de-registration scheme and group de-registration scheme. Performance analysis of all these schemes shows that group de-registration scheme is best among existing and proposed schemes. In the proposed multi HLR architecture group de-registration scheme is implemented instead of explicit de-registration scheme. Result shows that proposed architecture with group de-registration scheme is more efficient than explicit de-registration scheme.

Keywords

Bottleneck, Call misrouting, Implicit de-registration scheme, Polling de-registration scheme, Timeout de-registration scheme, Group de-registration scheme, Explicit de-registration scheme.

1. INTRODUCTION

Current approach to support user mobility in GSM and IS-41 standard, we use a two-level database architecture, one is called home location register (HLR) and another is called visitor location register (VLR). This HLR-VLR architecture is centralized in nature [6]. In GSM and IS-41 standard we have one HLR and several VLRs. In HLR we store the user's information on permanent basis while in VLRs this storage is temporarily. HLR acts as the master database for the entire system. VLRs cache the user's profile from this master database. As we have physically and logically only one copy of the HLR hence its management is easy but it suffers from bottleneck [1, 2]. As per move, information stored in HLR and VLR changes. When an MT leaves its RA and enters into new RA being served by the same VLR, this information is changed only in the VLR. But when an MT leaves its RA and enters into the new RA that is being served by the different VLR, the MT's information is updated in HLR and this information is deleted from the old VLR [1, 2, 3, 4, and 5].

Now we can differentiate the user's mobility in two ways in conventional HLR-VLR architecture.

1.1. Intra-VLR Move: This move occurs when MT's RA changed but VLR is same. In this case, MT's current location information is updated in the VLR. This move does not affect the HLR.

1.2. Inter-VLR Move: This move occurs when MT changes its RA and new RA is being served by the new VLR. This move is shown in fig1.It has six steps.
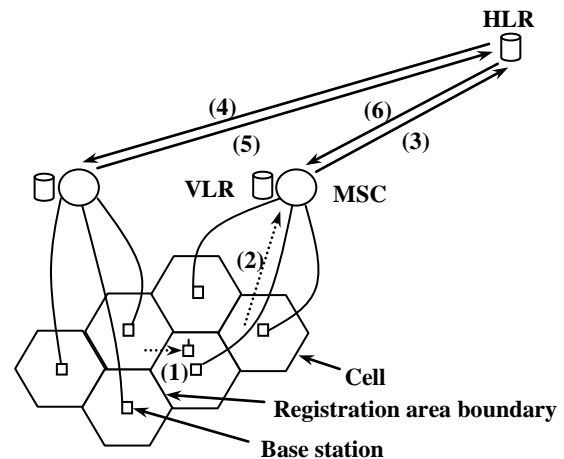

Fig 1: Explicit de-registration

Step 1: When mobile terminal enters into a new RA, it sends a location update to the nearby base station.

Step 2: The base station sends this information to the base station controller (BSC) through which it is being governed. The base station controller sends this information to the MSC/VLR.

Step 3: The new MSC/VLR updates its VLR which shows that the mobile terminal is residing in its service area and informs the HLR by sending a registration message to the HLR.

Step 4: The HLR sends a registration acknowledgment message to the new MSC/VLR together with a copy of the subscriber's user profile. This information is used to validate the subscriber.

Step 5: The HLR sends a registration cancellation message to the old MSC/VLR

Step 6: The old MSC/VLR removes the record of the mobile terminal at its associated VLR and sends a cancellation acknowledgment message to the HLR.

Various HLR implementations are available. SONOFON GSM report tells that if HLR supports 300,000customers and a user perform 20 HLR operations per day including location update, routing, authentication and network attachments, we have total 600,000 operations per day. During peak hour's 12-13% operations for location update, network attachment and

authentication come and total load becomes 800,000 queries per hour (222 transactions per second). This peak load even may be 50% higher. Such a heavy load cannot be supported by standalone HLR. As a consequence HLR suffers from bottleneck and cannot guarantee the quality of service [6].

In conventional HLR-VLR scheme, de-registration of an MT from a VLR is always explicit. Explicit in the sense that stale entries of VLRs are removed with the help of HLR. Actually HLR sends de-registration message to the VLR to remove the stale entries when an MT changes its VLR. This explicit de-registration increases the total cost by increasing the traffic load. To reduce the traffic load following de-registration strategies were proposed.

(A) Distance Based De-registration Scheme.
(B) Time-Based De-registration Scheme.
(C) Polling-Based De-registration Scheme.
(D) Group De-registration Scheme.
(E) Movement-Based De-registration Scheme.

Performance analysis of de-registration strategies in Personal Communication Network shows that the group de-registration scheme is best scheme among time and polling based de-registration schemes. This paper along with the limitations of the conventional HLR-VLR approach motivates to implement the group de-registration scheme in the multi HLR architecture [1].

2. REPLICATION OF HLR

Chen and Lee proposed an idea of replicating the HLR to avoid the bottleneck. In this approach we maintain several replicas of the HLR. In case of high load, replicas are used instead of the master-HLR [6]. If probability of failure in an HLR is p then availability of an HLR under high load condition will be $(1-p^n)$, if we have n HLRs (including 1 master-HLR and (n-1) replicas of HLR)[7]. However to maintain consistency of information in the existing replicas will introduce the extra costs. Again it is to be noted that this architecture is even centralized not distributed having only one advantage of removal of bottleneck but in case of high load when replicas are not correctly updated, misrouting of calls will take place.

3. PROPOSED MULTI HLR ARCHITECTURE

In the proposed architecture, we have several HLRs zone wise or circle wise instead of a single HLR. It reduces the storage overhead of the HLR. Each HLR can serve more than one VLR and each VLR can serve more than one RAs. Simply we can say that this architecture contains several conventional HLR-VLR architectures. For each MT we define two types of HLRs: a resident-HLR and a serving-HLR. Resident-HLR is the HLR where MT often resides. While on move, it can enter into the RA being served by another HLR (serving-HLR). When MT will be served by the HLR other than resident-HLR, we will refer it as roaming. In the proposed architecture we define following types of move as:

3.1. Intra-VLR-Resident-HLR Move: In this type of move, the MT changes its RA and the new RA is still being served by the same VLR. The serving VLR is being served by the resident-HLR. Now it is obvious that the location update is taking place only at VLR not at resident-HLR.

Intra-VLR-Resident-HLR move is shown in fig (2). An MT residing in registration area RA_1 moves to another registration area RA_2. RA_1 & RA_2 are being served by the same VLR, VLR_1. The VLR_1 is being served by the resident-HLR. Due to movement of MT, this location update is changed at VLR_1 not at resident-HLR.
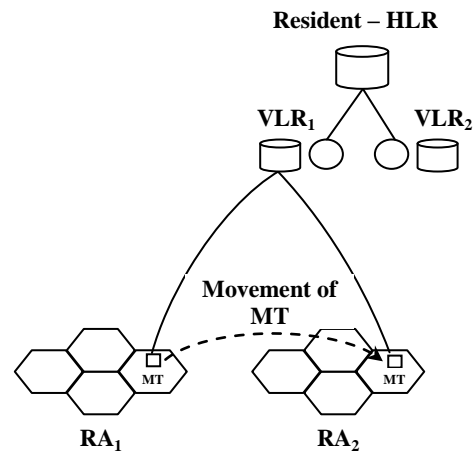

Fig 2: Intra-VLR-Resident-HLR move

3.2. Intra-VLR-Serving-HLR Move: In this type of move, the MT changes its RA and the new RA is still being served by the same VLR. The serving VLR is being served by the serving-HLR. Again this information is only updated at VLR not at serving-HLR and resident-HLR.
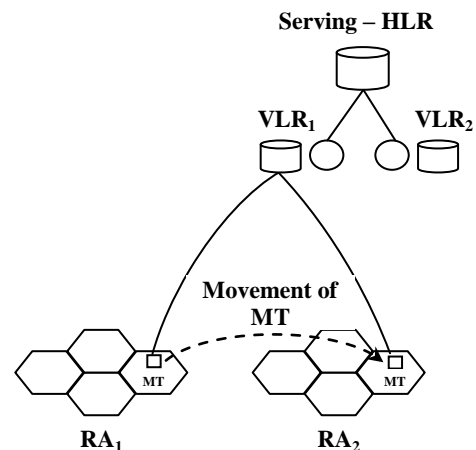

Fig 3: Intra-VLR-Serving-HLR move

Intra-VLR-Serving-HLR move is shown in fig (3). An MT residing in registration area RA_1 moves to another registration area RA_2. RA_1 & RA_2 are being served by the same VLR, VLR_1. The VLR_1 is being served by the serving-HLR. Due to movement of MT, this location update is changed at VLR_1 not at serving-HLR and resident-HLR.

3.3. Inter-VLR-Resident-HLR Move: In this type of move, the MT changes its RA and the new RA is being served by the new VLR. The serving VLR is being served by the resident-HLR. Now in this case registration of MT will take place at new VLR, de-registration of MT will take place at old VLR and finally resident-HLR will update this information in its database.

Inter-VLR-Resident-HLR move is shown in fig (4). An MT is residing in registration area RA_1 which is being served by the

VLR, VLR$_1$. This VLR$_1$ is being served by the resident-HLR. On move MT changes its registration area and comes in RA$_2$. This RA$_2$ is being served by the VLR$_2$. Now the registration of MT will take place at VLR$_2$ and de-registration will take place at VLR$_1$. This change in location update will take place on resident-HLR.
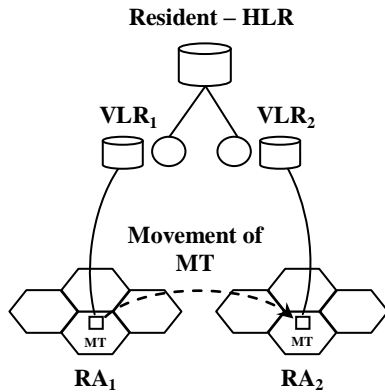

Fig 4: Inter-VLR-Resident-HLR move

3.4. Inter-VLR-Serving-HLR Move: In this type of move, the MT changes its RA and the new RA is being served by the new VLR. The serving VLR is being served by the serving-HLR. Now in this case registration of MT will take place at new VLR, de-registration of MT will take place at old VLR and finally serving-HLR will update this information in its database. However resident-HLR will not be updated.
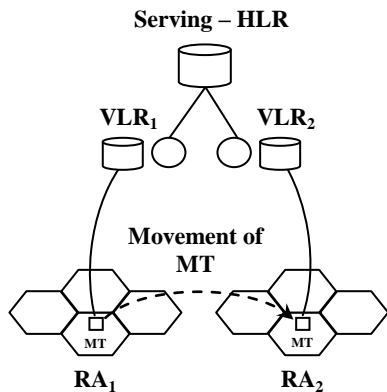

Fig 5: Inter-VLR-Serving-HLR move

Inter-VLR-Serving-HLR move is shown in fig (5). An MT is residing in registration area RA$_1$ which is being served by the VLR, VLR$_1$. This VLR$_1$ is being served by the serving-HLR. On move MT changes its registration area and comes in RA$_2$. This RA$_2$ is being served by the VLR$_2$. Now the registration of MT will take place at VLR$_2$ and de-registration will take place at VLR$_1$. This change in location update will take place on serving-HLR. Resident-HLR will remain unaffected.

3.5. Inter-VLR-Inter-HLR Move: This type of move occurs into two cases.

1) This type of move occurs when an MT comes back to its resident-HLR from serving-HLR. In this move MT's registration occurs at serving VLR of resident-HLR and this information is sent back to the serving-HLR.

2) On reception of this message serving-HLR de-registers this MT from its database and informs to its VLR (where MT was registered previously) for de-registration.

This case is shown in fig (6). MT is residing in the registration area RA$_2$ of VLR$_2$ of serving-HLR. On move MT comes under VLR$_3$ of resident-HLR. In this case both VLRs and HLRs change. The MT is now in its resident-HLR. Registration of MT will take place at VLR$_3$ and resident-HLR while it's de-registration will take place at serving-HLR and its corresponding VLR where MT was previously resided.
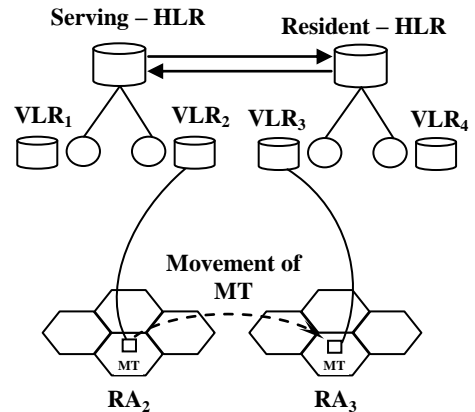

Fig 6: Inter-VLR-Inter-HLR move

3) When an MT moves to another serving-HLR. In this case registration of MT takes place at new serving-HLR and its VLR under which MT comes, de-registration occurs at the previous serving- HLR and its associated VLR from where MT is coming and resident-HLR is updated.

4. GROUP DE-REGISTRATION SCHEME IN PCS NETWORK

In group de-registration scheme, HLR maintains an OML (old mobile list) for each VLR. When an MT changes its VLR on move, HLR keeps this MT into the OML of previously serving VLR. When a location registration request comes for an MT, HLR sends this OML to the VLR as a part of registration acknowledgement. VLR uses this OML to remove the entries of MTs who have already left it. This scheme can be described into four steps:

1) When an MT changes its VLR, the new serving VLR sends a registration request to the HLR.
2) On reception of this request, HLR updates the location of MT and puts this MT's identification into the OML of previous VLR from where MT is coming.
3) HLR sends registration acknowledgement to the VLR along with its OML. Sent OML is always kept empty.
4) On reception of this acknowledgement, VLR finds the information of those MT who have left it and remove their information for its database.

Thus the invalid MT's identifications are removed from the VLR every time a new MT enters the RA [1, 3].

5. PERFORMANCE ANLAYSIS OF VARIOUS DEREGISTRATION STRATETGIES IN PCS NETWORK

Performance analysis of various de-registration strategies like explicit de-registration, implicit de-registration, timeout de-registration and group de-registration strategies are evaluated [1].

Result shows that cost incurred in the group de-registration strategy is greater than timeout and polling de-registration scheme but less than explicit de-registration strategy. However in case of group de-registration strategy no stale entry regarding the mobile terminal resides in the VLRs. In timeout de-registration strategy, removal of stale entries of MTs are dependent on the time. In this scheme, MTs periodically register themselves to VLRs so that it can ensure its presence into the VLRs. This scheme is a refinement to the polling de-registration scheme in which VLR broadcasts polling messages to collect the acknowledgement of the MTs residing in the RAs associated with the VLR. The polling scheme is also dependent on the time. Process of de-registration in explicit de-registration scheme is based on mobility of the terminals. Group de-registration strategy provides us to de-register the mobile terminals on mobility basis having less cost than the explicit one.

In this proposed architecture, we are implementing the group de-registration strategy in multi HLR architecture and evaluating its performance against explicit de-registration scheme.

6. PERFORMANCE ANALYSIS

An analytical model to evaluate the performance of multi HLR architecture has been presented here along with the group de-registration strategy implemented in same architecture. Here all HLRs are located at the same layer and they are communicating each other in point-to-point basis. In this analysis, hierarchal trees of R layers are being used. The layer R contains the roots of all trees and leaves of all trees are at the level 1. It means both roots and leaves reside on the same layer.

Following terms are being used in the performance analysis:-

$m_{x,y}$ Layer of the closest common node to RA x and RA y.
p Probability that the MT move is intra-VLR.
n New RA of the MT.
a Old RA of the MT.
α Probability of an MT to reside in its resident-HLR.
θ Probability of an MT to reside in first serving-HLR.
β Probability of an MT to reside in second serving-HLR.

$$\theta = \beta = (1 - \alpha) \dots\dots\dots\dots\dots (1)$$

P ($m_{x,y}$=i) is defined as the probability that the closest common node to LA x (RA x) and LA y(RA y) is in layer i. This probability can be given by the following equation.

$$P(m_{a,n} = i) = p(1-p)^{i-1} \text{ for } i = 1,2\dots\dots\dots.R\text{-}1$$
$$(1-p)^{i-1} \text{ for } i = R \dots\dots\dots\dots (2)$$

We furthermore denote the costs of various operations used in this analysis as follows:

T (i, j): Cost of transmitting a message over a link between two adjacent layers i and j.
C_m (i): Cost of accessing or updating a database in layer i.
$M_{multiHLR\text{-}VLR \text{ (explicit)}}$: Estimated cost of a location update in the explicit multi HLR-VLR scheme.
$M_{mutiHLR\text{-}VLR \text{ (group)}}$: Estimated cost of a location update using group de-registration scheme in multi HLR-VLR architecture.

Estimated cost of location update in explicit stand alone HLR-VLR scheme is given as:

$$M_{HLR-VLR(explicit)} = \left[P(m_{a,n=} 1) \times C_m(1) \right] + \sum_{i=2}^{R} P(m_{a,n=} i) \times \{2 \times C_m(1) + C_m(R) + 4T(1,R)\} \dots\dots (3)$$

The first part of Eq. (3) is the cost of location update in intra-VLR move. The second part illustrates the scenario after an inter-VLR move. T (1, L) =T (1, 2) +T (2, 3) +……..+T (L-1, L) is equal to the cost of traversing links between a node of layer 1(i.e., VLR) and the node of layer R (i.e., where an HLR is located). This cost is multiplied by 4 because new VLR sends registration request to the HLR, the latter sends cancellation request to the old VLR, old VLR sends an acknowledgement in response to the HLR and finally HLR confirms the registration of new MT at the new VLR.

Transmission cost of the message is described as follows:
T(1,L)=T(1,2)+T(2,3)+……..+T(L-1,L)
T (1, 2) will give the result 2; T (2, 3) will give the result 3 and so on.

Estimated cost of location update with group de-registration scheme is given as follows:

$$M_{HLR-VLR(group)} = \left[P(m_{a,n=} 1) \times C_m(1) \right] + \sum_{i=2}^{R} P(m_{a,n=} i) \times \{2 \times C_m(1) + 3 \times C_m(R) + 2T(1,R)\} \dots\dots (4)$$

The first part of this Eq. No.(4) is the cost of location update in intra-VLR move. The second part illustrates the scenario after an intra-VLR move. When an MT leaves its RA and enters into new RA the new VLR sends a registration request to the HLR. HLR keeps the identification of the MT into the OML of the old VLR. After performing the MT's profile update by accessing its database HLR sends the acknowledgement message along with the OML of new VLR. We see that HLR database is being consulted three times. The first access is done for putting the MT's identification into the old VLR's OML, second time for updating the MT's current location information and third time for emptying the OML of new VLR, further the entries of this OML is sent back with the acknowledgement. At the VLR side database is being consulted twice, first for the registration of new MT and second for de-registration of the entries sent by the HLR.

Using equ. No. (3) and (4) we can derive the expressions for location management schemes for multi HLR-VLR architectures as:

$$M_{multiHLR-VLR(explicit)} = \alpha x M_{HLR-VLR(explicit)} + \beta x [M_{HLR-VLR(explicit)} + 2] + \theta x [M_{HLR-VLR(explicit)} + 4] \dots\dots (5)$$

$$M_{multiHLR-VLR(group)}$$
$$= \alpha x M_{HLR-VLR(group)}$$
$$+ \beta x [M_{HLR-VLR(group)} + 2]$$
$$+ \theta x [M_{HLR-VLR(group)} + 4] \dots\dots (6)$$

First part of equ. (5) is the cost when MT is not leaving its resident-HLR. In this case mobility of MT is local to the HLR; it may change its serving VLR and can enter into new VLR being served by resident-HLR. The second part of eqn (5) is the cost when an MT leaves resident-HLR and enters into new RA being served by another HLR called serving-HLR. A cost of 2 is being added only to inform the resident-HLR that its MT has changed its location and resident-HLR acknowledges the current serving-HLR. In part 3 of this equation, we are generalizing the movement of an MT when MT leaves its resident-HLR and enters into new serving-HLR and then after it again changes its serving-HLR to another serving-HLR. A cost of 4 is being added to it because of the following reasons: Let an MT leaves its resident-HLR and enters into a serving-HLR say serving-HLR1. Again the same MT leaves this serving-HLR1 and enters into another serving-HLR say serving-HLR2. In registration process of MT in serving-HLR2, it sends location update to resident-HLR (cost incurred is 1), on reception of this message, resident-HLR updates the location information of MT and sends a location cancellation message to serving-HLR1 (cost incurred is 1). On reception of this message, the serving-HLR1 deletes the location information of this MT. The serving-HLR1 acknowledges the resident-HLR about the location cancellation message (cost incurred is 1). Finally the resident-HLR acknowledges the serving-HLR2 (cost incurred is 1) and location registration takes place at the serving-HLR2. As all HLRs are the same level hence message exchange cost between the two HLRs is T (1, 1) =1.

In eqn(6), each part can be explained in the same manner as in eqn(5). Only difference is that in eqn (5) explicit de-registration scheme is being used while in eqn (6) group de-registration scheme is being used.

7. RESULTS

In this section the numerical values of explicit de-registration scheme and group de-registration scheme implemented in the multi HLR-VLR architecture are evaluated and compared. Fig (6) and (7) show the performance of location update schemes with R=5 and R=3 respectively.
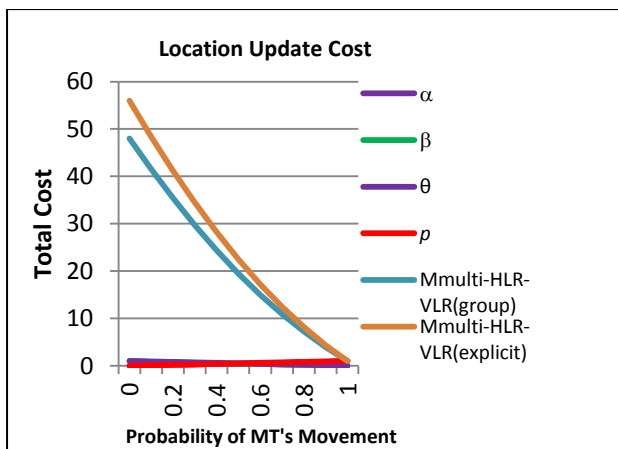

Fig 6: Location update cost when R=5.

Figures (8) & (9) show the relative cost in percentage $(((M_{multiHLR-VLR\ (group)}/\ (M_{multiHLR-VLR\ (explicit)})*100)$ when R=5 and 3. Probability α shows that the MT will reside in its resident-HLR. Probability β and θ indicate that MT will be in serving-HLRs. When α has value 1 then MT is not changing its resident-HLR. Probability p defines the MT's intra VLR move.

If p equals to 1 then MT is not changing its location. When p tends to 0, it means MT's move is not local. When α and p are equal to 1, MT is in its resident-HLR and not changing its location and hence location update cost is 1. When α and p equals to 0, it means MT is not in its resident-HLR and its move is not local with respect to serving-HLR. When α and p equal to 0, it shows the maximum degree of movement, in this case group de-registration scheme is 72.73% efficient than the conventional explicit de-registration scheme when R=5 and 85.71% in case when R=3.
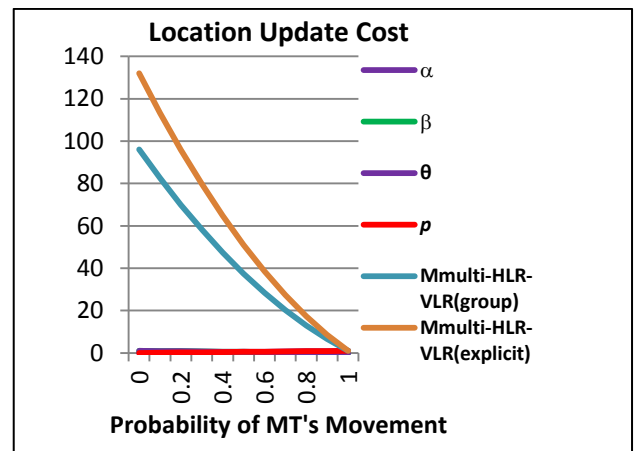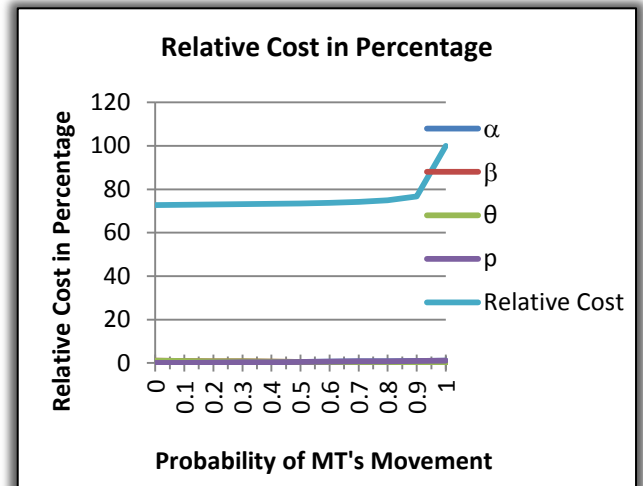

Fig 7: Location update cost when R=3.

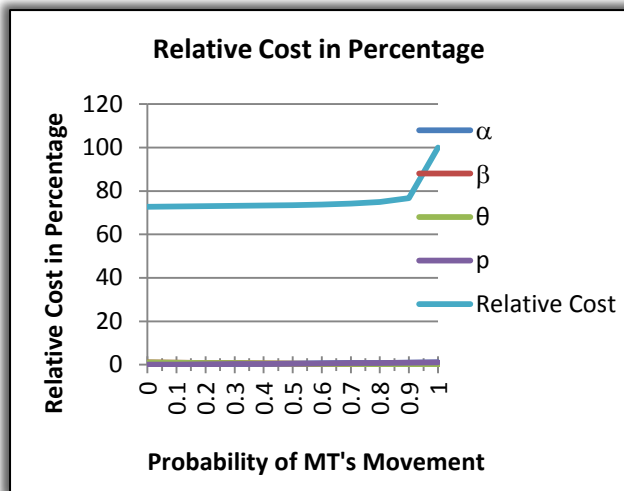**Fig 8: Relative location update cost when R=5.**

Fig9: Relative location update cost when R=3.

8. CONCLUSION

Conventional architecture has a single HLR and that's why it suffers from call misrouting and bottleneck during peak load. To remove this, several conventional architectures are group together to form multi HLR architecture. In this scheme we store the user profile in HLRs zone wise. This approach reduces the storage load on HLR and hence minimizes the possibility of bottleneck. Now in this architecture even in high load appropriate information is fetched from HLRs and we significantly minimize the possibility of call misrouting.

Analysis done in the last section shows that total cost incurred into the location management in the proposed multi HLR-VLR architecture using group de-registration scheme is efficient than the explicit de-registration scheme. The proposed architecture is free from the problem of bottleneck as we are not entirely [8]

relying on one HLR. We have not any stale entry of MT in any VLR as we have associated the de-registration process of the MT with its movement and saved sufficient cost by implementing group de-registration scheme instead of conventional explicit de-registration.

9. REFERENCES

[1] Rajeev R. Kumar Tripathi, Sudhir Agrawal and Swati Tiwari, "Performance Analysis of De-registration Strategy in Personal Communication Network," IJCA, vol.24-no.1, June 2011.

[2] Rajeev R. Kumar Tripathi, Sudhir Agrawal and Swati Tiwari, "Modified HLR-VLR Location Management Scheme in PCS Network," IJCA, vol.6-no.5, Sept 2010.

[3] Z. Mao, C. Douligeris, "A novel deregistration strategy for mobile network," IEEE VTS 2750-2754, vol.6, Aug 2002.

[4] Haider Safa and Samuel Pierre, "A New Architecture for Improving Location management in PCS Network," Journal of Computer Science1 (2):249-258, 2005.

[5] Sarvpal H. Singh, Sudhir Agrawal and Vijay Kumar "Performance Evaluation of Location Management Techniques in PCS Networks," IJCA, vol.15, no-8, Feb-2011.

[6] Guan-Chi Chen and Suh-Yin Lee, "Evaluation of Distributed and Replicated HLR for Location Management in PCS Network," Journal Of Information Science And Engineering 19, 85-101 (2003).

[7] George Couloris, Jean Dollimore and Tim Kindberg, "Distributed Systems Concepts and Design," Pearson Education, 4e.