Client Location Tracking with K-Step Prediction and Cache Policy in Ubiquitous Information Service Network

Jang, Hung-Chin (jang@cs.nccu.edu.tw), Lien, Yao-Nan (lien@cs.nccu.edu.tw)
Dept. of Computer Science, National Cheng-Chi University, Taipei, Taiwan, R.O.C.

Huang, Jyh-Shyan (frank@cs.nccu.edu.tw)
Dept. of Applied Mathematics, National Cheng-Chi University, Taipei, Taiwan, R.O.C.

Abstract

The objective of this paper is to present a feasible and efficient methodology that improves client mobility management under ubiquitous information service network. We base our system on the Open Integrated Service Network Architecture[4] and propose a hybrid infrastructure with facilities of both a centralized Network Management Center (NMC) and a distributed Home Base Node (HBN). We use LSDMk for choosing a HBN with minimum cost by computing the expected registration cost and location tracking cost according to users' future k-step movement and called frequencies. LSDM[9] is a special case of LSDMk when k is equal to 0. In addition, we use a cache strategy to save location tracking cost.

Running simulations, results show that our methodology is able to save up to 86% and 54% of the cost comparing to that of IS-41 in best and average cases, respectively, when the value of Call-Mobility Ratio (CMR) is in the range of (1, 100), which means either clients are called with high frequencies or are moving with low mobilities. As the value of CMR goes down to the range of (0.01, 1), which means either clients are called with low frequencies or are moving with high mobilities, we may save up to 83%, 60%, and 10% costs in best, average, and worst cases, respectively.

1. Introduction

Ubiquitous information service network (UISN) is ideally to provide an information environment such that any user may access information resource via either wired or wireless transportation network anytime and anywhere. The technique we had before allows us to retrieve information from a single resource. If an information system provides us with facility to integrate information from multiple resources, we shall have more precise forecast and make better decisions.

Lien [4] proposed an open service network architecture, which is appropriate for information providers to build up information for their clients. This architecture follows:

Rule 1: separation of physical transportation network and logical network (service network).
Rule 2: adhering to open system architecture.
Rule 3: decentralized service management for the purpose of reducing workload of centralized service management.

To maximize utilization of UISN, users are expected to be given a variety of mobilities, eg, terminal mobility, personal mobility, physical network mobility, agent mobility and client mobility (or logical network mobility). Client mobility provides logical network independence to users. A client needs only an unique ID number to travel around different logical networks, say Internet and PSDN. No matter where he is, other clients may always reach him by this number. For instance, if client A wants to reach client B and thus, sends a message to him. If B is on a cellular phone, the service network will forward this message to the telephone; while B is on Internet, this message would be sent to him through e-mail.

Current communication network provides no way for a mobile client to communicate with the Internet resource, like Internet phone, through his phone number. The client mobility allows a client, when he was in a car, to use his PDA to send out a message through wireless phone for hotel reservation that is provided by hotel reservation service network. At the time when he gets to office, he can retrieve the reservation result from Internet that is connected to his office PC. What he experienced is doing one thing using different communication devices and transportation networks at different time and different places.

In the upcoming network information age, everyone should go with computer network to do many things in his daily life. Client mobility would be more important than others, that's why we focus on management of client mobility in this paper. Most of the current researches of mobile communication technology focus on client mobility management of mobile telephone network, like management of mobile stations of cellular phones. However, less researches have referred to management of logical network. In this paper, our management of client mobility is accomplished through the following steps: First, through analyzing a client's moving frequency and called frequency, we compute both the registration cost and location tracking cost for each HBN, then we choose to register to the HBN with minimum joint cost.

The organization of this paper is as follows: we introduce ubiquitous information service system architecture in section two. In section three, we introduce several decision models to both optimize registration cost and reduce location tracking cost. In section four, we compare the total costs of registration and location tracking of our models to that of IS-41. Section five is conclusion and future work.

2. System Architecture

A feasible and effective mobility management system architecture should be associated with a variety of network management facilities. Our system architecture is equipped with the following facilities for real implementation:

(1) Network Management Center (NMC)
NMC is a central facility that supports all non-distributable management functions as well as distributed functions if it is needed. Typical OA&M (operation, administration, and maintenance) functions are client and server registration, authentication, name server, coordination, or client specific services. Although, it looks like a single node, it may actually be a number of nodes distributed over a network. For simplicity, we assume a NMC is a single node. In general, a commercial service network usually needs a NMC for such functionalities as billing, authentication, security, etc.

(2) Home Base Node (HBN)
Even though a mobile client may change its locations from time to time, it usually has a home location and a most frequently used Internet access point such as a personal account on an Internet-connected system or a PC, called Home Base Node. This infrastructure proposes to make use of HBN to share the OA&M workload. Since a mobile terminal device is very vulnerable in a hostile environment, it is not wise to use a mobile device as the primary information repository. A more reasonable working model for most users is to use their mobile computing devices only for temporary cache and have their HBNs as primary working systems. Further, when a client subscribes to a service network, he can choose to use or not to use his own HBN. A lot of overhead in managing a service network can be saved by using HBNs. A HBN can also offer auxiliary computing resources to help mobile terminals to cope with their residential resource limitation. For example, applications that a HBN can perform are : location register, fax server, answering machine, news filter, etc.

(3) Location Directory (LD)
Every record in LD has two entries of data, which are user-ID and user address on the network, respectively. As a client moved, we should update the data of user-address to make it consistent with the current address of the client.

(4) Location Server (LS)
The main functions of LS is to provide services of location registration and location tracking. Location registration is to update the user-address and make it consistent with the current address of the client. Location tracking is to localize where the client to be called is. One of the services provided by HBNs is to keep track of the clients' locations. Once a HBN is chosen to provide this service, it is also called a LS.

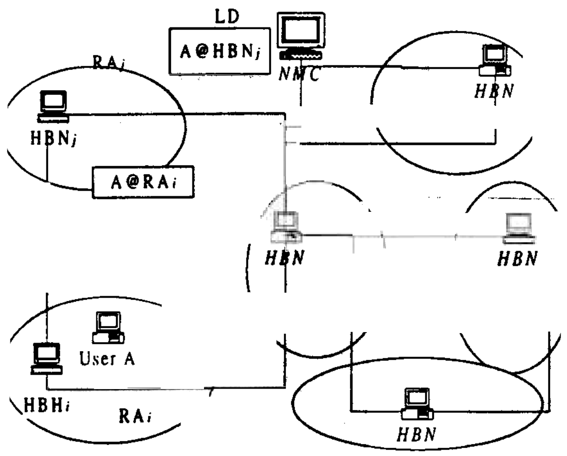

Fig. 2-1 System architecture.

(5) Register Area (RA)
A register area is a geographical region partitioned by NMC according to the distribution of all LSs. Every register area is assigned at least one LS. A mobile client might change a new LS only when he moves to other RA.

Fig. 2-1 is the system architecture which shows a variety of facilities distributed over a network. Where HBNs reside in RAs, mobile clients are allowed to move across RAs, and LDs are stored in both NMC and HBNs for location tracking. In this figure, user A is in RA_i, HBN_j is the LS of user A, and the LD of HBN_j records the address of user A as $A@RA_i$. The LD of NMC records HBN_j, which is the LS of user A. Comparing to those facilities of IS-41, NMC corresponds to HLR (Home Location Register) and LS to VLR (Visitor Location Server).

3. Location Server Decision Model (LSDM)

In our system, a client and his LS are allowed to reside in different RAs, e.g., user A and his LS, HBN_j, in Fig. 2-1. Also, all HBNs on the network are possible candidates of a given client's LS. In this section, we propose a decision model to choose one of the HBNs to serve as this given client's LS.

In section 3.1, we define some denotations of matrices and vectors to model a user's moving behavior and called behavior. In section 3.2, we represent a cost function of registration cost and location tracking cost of the user in terms of those matrices and vectors. In section 3.3, we use that cost function to choose a HBN, with minimum total cost, as the user's LS.

3.1 Denotations

In the following definitions, we use matrices and vectors to model length of time a user stays in a RA, probability of a user moves from one RA to another, time spent on network communication between two RAs, cost of using a NMC every unit time, user's called frequency every unit time, and the RA where the caller is in, etc.

N : a set of RAs. One register area belongs to N if and only if either a user A has ever stayed in this area or some other users have ever called user A in this area.

the total number of RAs in N.

RA_i : a register area, with index i, that belongs to N, where N={ $RA_1, RA_2, \ldots, RA_n$ }.

HBN_i : a home base node, with index i, that resides in RA_i. At a given time, each user would choose a HBN to serve as his LS. Note that the user and his LS are not required to reside in the same RA.

S : a RA-dimension vector, $S=(S_1, S_2, \ldots, S_n)$. Where S_i, a real number, represents the expected time a user stays in RA_i.

C a $n \times n$ matrix, $C=[C_i$ $_{n\times}$ Where C_i an integer, represents expected number of calls come from RA_i per unit time when a user stays in RA_j.

M : a $n \times n$ matrix, $M=[M_{i,j}]_{n \times n}$. Where $M_{i,j}$, a real number, represents the probability of a user's movement from RA_i to RA_j.

T : a $n \times n$ matrix, $T=[T_{i,j}]_{n \times n}$. Where $T_{i,j}$ represents the network communication time between HBN_i and HBN_j.

T_{NMC} : a n-dimension vector, $T_{NMC} = (T_{NMC,1}, T_{NMC,2}, \ldots, T_{NMC,n})$. Where $T_{NMC,i}$ a real number, represents the time spent on network communication between NMC and HBN_i.

W : a $n \times n$ matrix, $W=[W_{i,j}]_{n \times n}$. Where $W_{i,j}$, a real number, represents the cost spent on network communication between HBN_i and HBN_j every unit time.

W_{NMC} : a n-dimension vector, $W_{NMC} = (W_{NMC,1}, W_{NMC,2}, \ldots, W_{NMC,n})$. Where $W_{NMC,i}$, a real number, represents the cost spent on network communication between NMC and HBN_i.

All the above matrices and vectors are recorded in the mobility profile of a mobile user, say user A. Besides, this profile also includes the following information:

(1) user-IDs, calling times, and calling locations of all those users who communicate with user A.

(2) a names directory of user A. He might choose only respond to those users' calls whose names are listed in the directory.

3.2 Location Registration, Location Tracking and Cost Evaluation

In this section, we propose a strategy for mobile users' location registration and location tracking; and then use a cost function to compute the total cost.

3.2.1 Initial Location Registration

When a mobile client sends a " ask register " request to a service network at the first time, either NMC or the client himself will choose and set up a HBN as his LS. The client will then sends his user-ID together with user-address to that LS. Receiving these information, the LS forwards client's user-ID and LS-address to NMC..

The strategy proposed for mobile users' location registration and location tracking is described in section 3.2.2 and 3.2.3.

3.2.2 Evaluation of Registration Cost

Assuming that user A resides in RA_i with HBN_m as his LS. When he moves to RA_j and changes his LS to HBN_n, then the registration cost can be evaluated in two cases [3]. In the first case, when new LS is different from old LS, that is, $HBN_n \neq HBN_m$, where $n \neq m$; the other case is both of these LSs are the same, that is, $HBN_n = HBN_m$, where n=m.

Case1 : $n \neq m$, user A moves from RA_i to RA_j and changes his LS from HBN_m to HBN_n.

Step0 : user A sends "ask register" message to HBN_j and writes his user-ID and user-address into HBN_j's LD. Since both user A and HBN_j are in the same RA, communication cost between user A and HBN_j can be ignored.

Step1 : HBN_j sends "ask register" message to HBN_n and writes user A's user-ID and user-address (A@ RA_j) into HBN_n's LD.

Step2 : HBN_n sends "ask register" message to NMC and writes it's LS-address (A@ HBN_n) and user-ID into NMC's LD.

Step 3 : NMC sends "ask de-register" message to HBN_m to delete user A's registration record (A@ RA_i).

Summing up all the costs from step0 to step3, we have:

$$T_{j,n} \times W_{j,n} + T_{NMC,n} \times W_{NMC,n} + T_{NMC,m} \times W_{NMC,m}$$

Case 2 : m=n, user A moves from RA_i to RA_j with the same LS ($HBN_n = HBN_m$).

Step 0 : similar to the step0 of Case1, cost of this step can be ignored.

Step1 : similar to the step1 of Case1.

Step2 : since user A doesn't make any change to the record (A@ HBN_n) in NMC's LD.

Step3: since HBN_n and HBN_m are the same, cost of this step can be ignored.

Summing up all the costs from step0 to step3, we have:

$$T_{j,n} \times W_{j,n}$$

In summary, the registration cost (register_cost) of user A spent on moving from RA_i to RA_j and changing his LS from HBN_m to HBN_n, is as follows:

when $m \neq n$:

$$register\_cost(n,\ j,\ m,\ T,\ W) = T_{j,n} \times W_{j,n} + T_{NMC,n} \times W_{NMC,n} + T_{NMC,m} \times W_{NMC,m}$$

when $m = n$:

$$register\_cost(n,\ j,\ m,\ T,\ W) = T_{j,n} \times W_{j,n}$$

where *m*, *n*, and *j* represent user A's old LS, current LS, and current RA indices, respectively.

3.2.3 Evaluation of Location Tracking Cost

Assuming that user A is in RA_j with current LS, HBN_n , and user B is in RA_i. Once user B wants to track the current location of user A, there are three steps to be taken [3]:

Step 0 : user B sends "ask query" message to HBN_i. HBN_i checks it's LD to see if there is user A's current address. If it is found, then there are two possible cases:

Case1 : the record in HBN_i's LD shows that user A is not in RA_i, and user A sets HBN_i as his current LS. The address found in HBN_i's LD is user A's current location.

Case2 : the record in HBN_i's LD shows that user A is in RA_i. At this time, HBN_i broadcasts to every user that is in RA_i. If user A is actually in RA_i, then user A is found ; otherwise, it indicates that user A had ever stayed in RA_i, but moved to other RA. HBN_i would then delete this record and go to step1.Since HBN_i needs not communicate with other HBNs and NMC, the cost of step0 can be ignored.

Step1 : HBN_i sends "ask query" message to NMC to check user A's current LS. NMC would then check and send the address of user A's current LS back to HBN_i.

Step2 : now, user B has the address of user A's current LS, HBN_n. At this time, user B sends "ask query" message from HBN_i to HBN_n to retrieve user A's current location. As HBN_n received this query, it returns user A's current location back to HBN_i.

In summary, the cost of location tracking (location_cost) is:

when $i \neq j$ and $i \neq n$

$$location\_cost(i, j, n, T, W) = 2 \times T_{NMC,i} \times W_{NMC,i} + 2 \times T_{i,n} \times W_{i,n}$$

when $i = j$ or $i = n$

$$location\_cost(i, j, n, T, W) = 0$$

where i and j are indices of user B's and A's current RAs, respectively ; n is the index of user A's HBN.

3.2.4 Total Cost of Registration and Location Tracking

In the previous subsections, we evaluate the costs of registration and location tracking. Here, we define a total cost function as follows: the registration cost is the cost that a user, say user A, spent on moving from old RA to new RA, and the location tracking cost is the cost spent by other users who call user A, when he moves to a new RA. The registration cost can be evaluated by the same register_cost function, but the location tracking cost can't be just apply location_cost function since it depends on all the calls made by all callers instead of a single caller.

In Fig. 3-1, we assume that user A moves into RA_k and changes his LS from HBN_j to HBN_l, the registration cost is

$$register\_cost(l, k, j, T, W) \cdots\cdots\cdots\cdots(1)$$

N is the collection of all RAs in Fig. 3-4, so we have $N = \{RA_i, RA_j, RA_k, RA_l, RA_m, RA_n\}$. S_k is expected time user A stays in RA_k. $C_{i,k}, C_{j,k}, C_{l,k}, C_{m,k}, C_{n,k}$ represent expected number of calls every unit time made by callers from $RA_i, RA_j, RA_k, RA_l, RA_m, RA_n$, respectively. Altogether, we have $(C_{i,k} + C_{j,k} + C_{l,k} + C_{m,k} + C_{n,k}) \times S_k$ number of calls arrived during the time when user A stays in RA_k. Specifically, $C_{i,k} \times S_k$ represents number of calls come from RA_i, and the cost of location tracking by calling from RA_i is

$C_{i,k} \times S_k \times location\_cost(i, k, l, T, W)$.

The costs from other RAs can be computed similarly. Summing up all the costs, we have user A' location tracking cost as:

$$\sum_{RA_x \in N} C_{x,k} \times S_k \times location\_cost(x, k, l, T, W) \cdots\cdots(2)$$

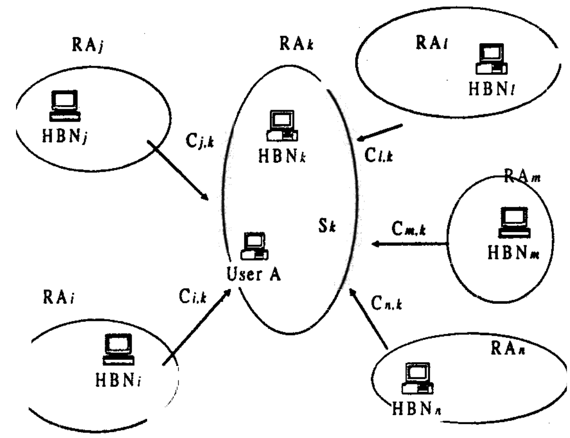

Fig. 3-1 Number of calls arrived every unit time.

From (1) and (2), we have user A's total cost function as follows:

$$Total\_cost(l, k, j, T, W, C, S) = register\_cost(l, k, j, T, W) + \sum_{RA_x \in N} C_{x,k} \times S_k \times location\_cost(x, k, l, T, W)$$

where l, k, and j are indices of user A's current LS, current RA; and old LS.

3.3 Decision Method for Choosing HBN to Register

3.3.1 Location Server Decision Method (LSDM)

When a user moves from RA_i to RA_j and changes his LS form HBN_m to an unknown HBN_x. The total cost is.

Total_cost(x, j, m, T, W, C, S)

Where j and m are constants and T, W, C, S are constant matrices and vectors that recorded in users' mobility profiles. Hence, *Total_cost* function can be defined in terms of x.

F(x)=Total_cost(x, j, m, T, W, C, S)

Applying all HBNs of RAs belonging to N into $F(x)$, we would have the total costs of each HBN that serves as user A's new LS. Our decision method would choose the

HBN, say HBN_{opt}, that satisfies

$F(opt) = Min\{F(x) | RA_x \in N\}$, as user's new LS.

3.3.2 LSDM with K-Step Prediction (LSDMK)

In the previous subsection, we choose the location of one user's LS through LSDM. In this subsection, we rebuild LSDM to consider user's k-step look ahead movement. The new decision model is thus called LSDMK.

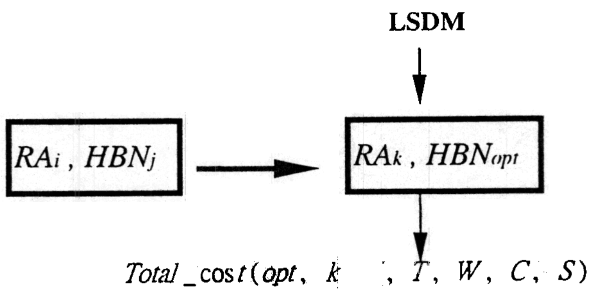

Fig. 3-2 Single step cost.

In Fig. 3-2, RA_i and HBN_j, in the left box, represent one user's previous state together with his LS, respectively. The right hand side of Fig. 3-2 shows that we use LSDM to choose HBN_{opt} as the user's current LS, while he is in RA_k, and the cost is $Total\_cost(opt, k, j, T, W, C, S)$. In the following, we employ LSDMK to take into consideration of the user's k-step look ahead movement and help the user to choose a proper HBN as his LS.

In Fig. 3-3, the user's old state is RA_i with LS set to HBN_j, as he moves to RA_k and chooses HBN_l as his LS, the expected total cost after x more step movement is $Total\_cost(l, k, j, T, W, C, S) + Y_{l,k}^{(x)}$.

According to the value iteration method, $Y_{l,k}^{(x)}$ represents the total cost when user's current state and LS are RA_k and HBN_l, respectively; and with x more steps to go. $Y_{l,k}^{(0)}$ equals to zero, where l, k are indices of RAs that belong to N. $Y_{l,k}^{(x)}$ can be expressed as

$$Y_{l,k}^{(x)} = \sum_{n,m \in N} [Total\_cost(n, m, l, T, W, C, S) + Y_{n,m}^{(x-1)}] \times M_{k,m}.$$

LSDMK decision model is used to choose the index of user's LS, l, to minimize $Total\_cost(l, k, j, T, W, C, S) + Y_{l,k}^{(x)}$.

That is, LSDMK is used to choose HBN_{optk} as user's LS such that

$$Total\_cost(optk, k, j, T, W, C, S) + Y_{optk,k}^{(x)} = \min_{l,k,j \in N} \{Total\_cost(l, k, j, T, W, C, S) + Y_{l,k}^{(x)}\}$$

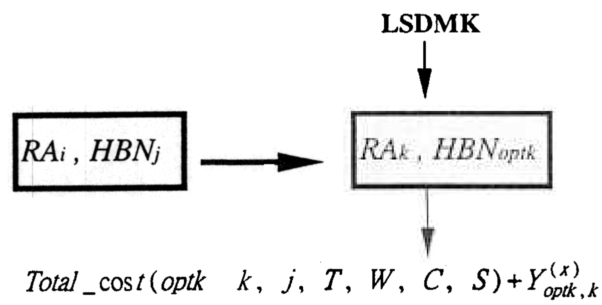

Fig. 3-3 K-step cost.

3.3.3 LSDM with Cache Policy

From the previous sections, we know that the higher the value of CMR the more expensive the total location tracking cost. In this subsection, we use a cache policy trying to reduce the cost when the value of CMR is high. For those users residing in the same RA with user B call user A with high frequency, there supposed to have many messages sent to NMC in high frequency to get the location of user A's LS asking for user A's current location. It results in high location tracking cost in this matter. Our proposed policy is to use a cache to cache user's LS in HBN for the sake of reducing NMC access frequency and accordingly saving lots of location tracking cost.

In our cache policy, every HBN has a cache consisting of many records. Every record contains two entries, one is user-ID and the other user's LS location. After tracking user A's location, HBN knows not only user A's address but also his LS location. Therefore, callers are not necessary to send "ask query" to NMC to access user A's LS location if this location remains the same. Once this location changes, the next caller who tracks user A will find that the location recorded in cache is not actually the user A's current LS location. The caller will therefore invoke location tracking policy to find user A's current address together with the location of LS, and update user A's LS record in the cache to make it consist with the location of user A's current LS.

From Fig. 3-4, we see that there is a cache in HBN_i with one $A@HBN_k$ record. It means that user A's LS is HBN_k which is observed by the latest user who called user A from RA_i and recorded this information in the cache of HBN_i. Assuming that user A is in RA_j with current LS set to HBN_n and user B is in RA_i. Once user B wants to track the current location of user A, there are four steps to take (see Fig. 3-4):

Step 0 : user B sends "ask query" message to HBN_i.
HBN_i checks it's LD to see if there is user A's current address. If it is found, then there are two possible cases:
Case1 : user A sets HBN_i as his current LS and the record in HBN_i's LD shows that user A is not in RA_i. The address found in HBN_i's LD is user A's current location. Go to step3.
Case2 : the record in HBN_i's LD shows that user A is in RA_i. At this time, HBN_i broadcasts to every user that is in RA_i. If user A is actually in RA_i, then user A is found and go to step 3; otherwise, it indicates that user A had ever stayed in RA_i but moved to other RA. HBN_i would then delete this record and go to step1.
Since HBN_i needs not communicate with other HBNs and NMC, the cost of step 0 can be ignored.

Step 1 : HBN_i checks it's cache to see if there is user A's LS location, HBN_k. If it is found, HBN_i sends "ask query" to HBN_k then there are two possible cases:
Case1 : if HBN_k is user A's current LS or user A is in RA_k then user A's current address is found. Go to step3.
Case2 : Otherwise, HBN_k sends back a "miss" message. Go to step2.
Cost of step1 is $T_{i,k} \times W_{i,k} + T_{k,i} \times W_{k,i}$.

Step 2 : use location tracking policy. Step2-1 and step2-2 represent the step1 and step2 in Fig. 3-3, respectively. After user A's current address being found, we go to step3.
Cost of step2 is *location(i, j, n, T, W)*.

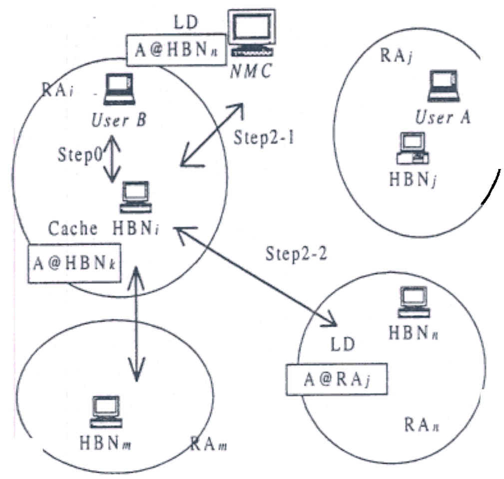

Fig. 3-4 User B tracks the location of user A with cache policy.

Step 3 : If user A's current LS is known, we check to see if the location of user A's LS that recorded in HBN_i's cache is consist with that of user A's current LS. If not, we update the location of LS to make them consist with each other. Since HBN_i needs not communicate with other HBNs and NMC, the cost of step3 can be ignored.

In summary, we have total cost of cache policy (cache_cost) in different cases:

Case I: When $i=j$ or $i=n$, *cache_cost(i, j, n, T, W)=0*
Case II: When ($i \neq j$ and $i \neq n$) and ($k=j$ or $k=n$), *cache_cost(i, j, n, T, W)*= $T_{i,k} \times W_{i,k} + T_{k,i} \times W_{k,i}$
Case III: When ($i \neq j$ and $i \neq n$) and ($k \neq j$ and $k \neq n$), *cache_cost(i, j, n, T, W)*=
$T_{i,k} \times W_{i,k} + T_{k,i} \times W_{k,i}$ +*location_cost(i, j, n, T, W)*

where *i* and *j* are indices of user B's and A's current RAs, respectively ; *n* is the index of user A's HBN; and *k* is the index of user A's HBN that recorded in HBN_i's cache.

4. Implementation and Result Analysis

4.1 Simulation

In the simulation of decision model, we assume that every RA has exactly one HBN. Let $W_{i,j}=1$, where *i*,

i=1, 2,.., n. S_i is chosen from a uniform random variable between 1 and 100 ; $C_{i,j}$ is chosen from a uniform random variable between 1 and 10.

The distributions of HBNs are on a xy-plane. The x and y coordinates range from -1000 to 1000, respectively. The (0,0) is chosen as NMC's coordinate. Let the network communication time between two HBNs be the distance between them. For example, HBN_i locates at (X_i,Y_i) and HBN_j locates at (X_j,Y_j), then $T_{i,j} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$, similarly, we have $T_{NMC,i} = \sqrt{X_i^2 + Y_i^2}$. In the following, we compute the total cost of registration and location tracking of a user's ten-step movement.

Total cost is defined in two different ways. The first one, Decesion_cost, is the cost spent on ten-step movement using LDSM. The second one, IS-41_cost, is the cost spent on ten-step movement using the method employed by IS-41.

4.2 Result Analysis

Fig. 4-1 illustrates the ratio of Decision_cost to IS-41_cost after ten-step movement. Y-axis represents the ratio of Decision_cost to IS-41_cost; and the x-axis represents the value of user's *CMR* [3]. The maximum, minimum, and average ratios of Decision_cost to IS-41_cost are 0.738, 0.554, and 0.609, respectively. The zigzag line in Fig. 4-1 represents the real ratios of Decision_cost to IS-41_cost, and the curve of function $y = 4\times10^{-6}X^2 - 0.0006 + 0.6258$ is the zigzag line's quadric asymptote. From the curve, we note that when the values of user's *CMR* are in the range of (1,100), changes to the values of *CMR* don't make significant changes to the ratios of Decision_cost to IS-41_cost. Hence, the ratios are very stable.

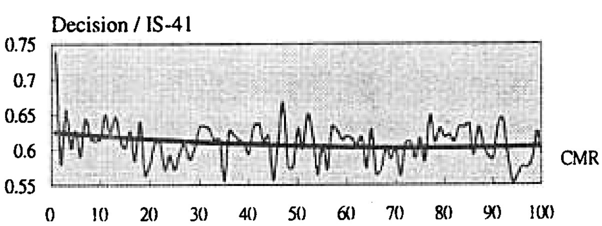

Fig. 4-1 Ratio of Decision_cost to IS-41_cost (CMR=1_100).

In Fig. 4-2, the maximum, minimum, and average ratios of Decision_cost to IS-41_cost are 0.602, 0.408, and 0.495, respectively. The zigzag line in Fig. 4-2 indicates the real ratio of Decision_cost to IS-41_cost, and the curve of function $y = 3\times10^{-7} - 2\times10^{-4}X^2 + 0.4795$ is the zigzag line's quadric asymptote. From the curve, we note that when the values of user's *CMR* are in the range of (0.01,1), changes to the values of *CMR* don't make significant changes to the ratios of Decision_cost to that of IS-41_cost. Hence, the ratios are very stable.

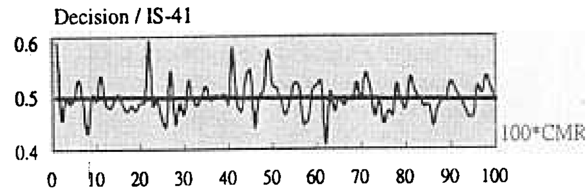

Fig. 4-2 Ratio of Decision_cost to IS-41_cost (CMR=0.01..1).

From these simulation results, we found that use of LSDM can save us at least 40% cost in comparing to that of IS-41. We don't count the computing time in making decision, which is of complexity $O(n^2)$, n is the total number of RAs in N, because the time is much less than the total network communication time, and all the computations are done on users' side.

4.3 Result Analysis (II) — LSDMK

In simulating the model of LSDMK, we fix the value of k to 2 and define total costs in terms of both LSDM_cost and LSDMK_cost. In Fig. 4-3, the maximum, minimum, and average ratios of LSDMK_cost to LSDM_cost are 1.000, 0.856, and 0.995, respectively. The zigzag line indicates the real ratio of LSDMK_cost to LSDM_cost, and the curve of function $y = -10^{-5}x^2 + 0.0015x + 0.9599$ is the zigzag line's quadric asymptote.

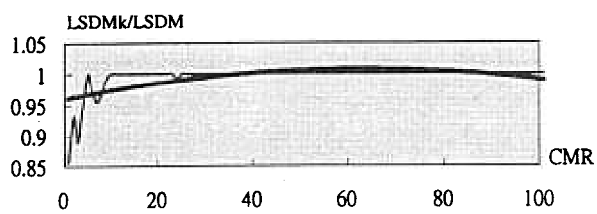

Fig. 4-3 Ratio of LSDMK_cost to LSDM_cost (CMR=1..100).

From the curve, we note that when the values of user's *CMR* are in the range of (1,100), changes to the values of *CMR* don't make significant changes to the ratios of LSDMK_cost to LSDM_cost. Where the ratio closes to 1 as the value of *CMR* is greater than 25.

In Fig. 4-4, the maximum, minimum, and average ratios of LSDMk_cost to LSDM_cost are 1.507, 0.401, and 0.820, respectively.

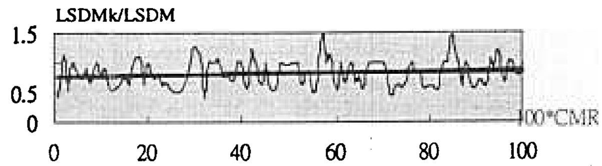

Fig. 4-4 Ratio of LSDMK_cost to LSDM_cost (CMR=0.01..1).

The zigzag line represents the real ratios of LSDMK_cost to LSDM_cost, and the curve of function $y=-10^{-5}x^2+0.0023x+0.7529$ is the zigzag line's quadric asymptote. From the curve, we note that when the values of user's *CMR* are in the range of (0.01,1), the ratio of LSDMK_cost to LSDM_cost is stable around 0.8.

Form above, we see that the ratio of LSDMK_cost to LSDM_cost closes to 1 as the value of *CMR* is greater than 25. In the best, average, and worst cases, the ratios of LSDMK_cost to IS-41_cost are 0.164 (0.401*0.408), 0.406 (0.820*0.495), and 0.907 (1.507*0.602), respectively, with the value of *CMR* is in (0.01,1).

4.4 Result Analysis (III) — LSDM with Cache Policy

In simulating the cache policy model, we define total costs in terms of both LSDM_cost and Cache_cost. Where LSDM_cost is the cost spent on ten-step movement using LDSM, and Cache_cost is that using cache policy. Fig. 4-5 and 4-6 illustrate the ratios of Cache_cost to LSDM_cost after ten-step movement with different ranges of CMR values. Y-axis represents the ratio of Cache _cost to LSDM_cost, and x-axis represents the value of user's *CMR*.

In the case when CMR is in the range of (1,100), the maximum, minimum, and average ratios of Cache_cost to LSDM_cost are 3.075, 0.258, and 0.749, respectively.

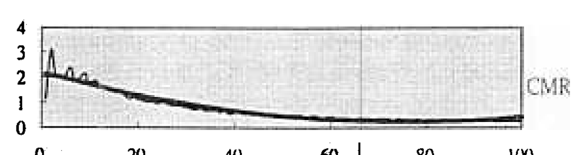

Fig. 4-5 Ratio of Cache_cost to LSDM_cost (CMR=1..100).

The zigzag line in Fig. 4-5 represents the real ratios of Cache_cost to LSDM_cost, and the curve of function $y=0.0003X^2-0.0506X+2.1799$ is the zigzag line's quadric asymptote. From the curve, we note that when the values of user's *CMR* are in the range of (1,100), the values of ratios of Cache_cost to LSDM_cost decrease as the values of CMR increase.

Table 4-1 shows varied ratios of Cache_cost to LSDM_cost when the value of CMR varied from 1 to 100. Column one represents the range of CMR; column two, three, and four represent the maximum, average, and minimum values of ratios of Cache_cost to LSDM_cost, respectively. It shows that, we may save much cost with cache policy as the value of CMR increases.

CMR	max	average	min
1 — 20	3.075007	1.721632	1.115412
21 — 40	1.078389	0.827424	0.572918
41 — 60	0.624073	0.508214	0.410561
61 — 80	0.419229	0.366386	0.312988
81 — 100	0.330884	0.2911	0.257954

Table 4-1 The maximum, average, and minimum ratios of Cache_cost to LSDM_cost when the value of CMR varied from 1 to 100.

In Fig. 4-6, the maximum, minimum, and average ratios of Cache_cost to LSDM_cost are 1.997, 0.948, and 1.168, respectively. The zigzag line represents the real ratios of Cache_cost to LSDM_cost, and the curve of function $y=-2\times10^{-5}X^2+0.0022X+0.9879$ is the zigzag line's quadric asymptote. From the curve, we note that when the values of user's *CMR* are in the range of (0.01,1), most of the ratios of Cache_cost to LSDM_cost is greater than 1. So, when one user's CMR is less then 1, HBN need not keep the location of the user's LS into its cache.

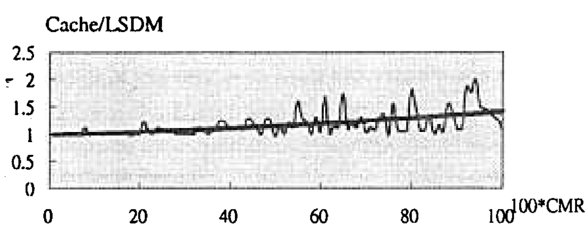

Fig. 4-6 Ratio of Cache_cost to LSDM_cost (CMR=0.01..1).

From Fig. 4-5 and Fig. 4-6 we found that the ratios of Cache_cost to LSDM_cost are not less then 1 until CMR is greater then 20. Thus, the right time to use cache policy is when the value of CMR is greater then 20. In the best, average, and worst cases with the value of *CMR* is in (1,100), the ratios of Cache_cost to IS-41_cost are 0.143 (0.258*0.554), 0.456 (0.749*0.609), and 2.269 (3.075*0.738) when the value of CMR is in (1,20) or 0.769 (1.078*0.738) when the value of CMR is in (21,100), respectively.

In the previous subsection, we observe that LSDMK can save cost only when CMR is in the range of (0.01,1). Therefore we conclude that it is better to use cache policy to manage user's mobility when the value of user's CMR is in the range of (21,100), and to use LSDMK when the value of user's CMR is in the range of (0.01,1).

5. Conclusion and Future Work

In comparing to other strategies of mobility management, the main difference is that we use LDSM, LSDMK, and Cache Policy to choose LSs with least costs, where a user and his LS are allowed to reside in different RAs. From the simulation results, we see that LSDM, LSDMK, and Cache Policy save us up to 40%, 60%, and 54% costs than that of IS-41, respectively. This is because of the following reasons:

(1) changing frequencies of LSs in LSDM (LSDMK) are much less than those in IS-41.
(2) in choosing a LS, LSDM (LSDMK) always selects the HBN with minimum location cost, but IS-41 doesn't.
(3) in LSDM (LSDMK), when either caller and callee; or caller and callee's LS are in the same RA, the location cost approximates to zero. In IS-41, only if caller and callee are in the same RA, the location cost approximates to zero. Thus, LSDM (LSDMK) saves location tracking cost.
(4) using a cache to cache user's LS in HBN may reduce NMC access frequency and thus save lots of location tracking cost.

Our system architecture currently has only one NMC. The more the numbers of users or the larger the regions covered, the higher and better performance demands on the service. In the future work, we would consider using multiple NMCs [5,6] or hierarchical architecture [8] to meet users' need.

Reference :

[1] A.Bar-Noy and I.Kessler, "Tracking Mobile Users in Wireless Communication Networks," IEEE Trans. on Info. Theory, Jan. 1994, pp.45-65.

[2] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile users: To update or not to update?" Wireless Networks, Vol. 1, 1995, pp. 175-186.

[3] Hung-Chin Jang, Y.N Lien and J.S. Huang, "Client Location Tracking in Ubiquitous Information Service Network" ICPADS'97 1997 International Conference on Parallel and Distributed System.

[4] Yao-Nan Lien, "Client and Agent Mobility Management," Second International Mobil Computing Conference, Hsinchu, Taiwan, R.O.C., March 25-27, 1996, pp. 141-151.

[5] Y.-B. Lin, "Location Tracking with Distributed HLRs and Pointer Forwarding," Proc. 1995 International Symposium on Communications, 1995, pp. 31-37.

[6] K.L. Sue, C.C. Tseng, and Y.S. Lai, "Reducing Call-Setup Time for Location Tracking with Distributed HLR and Pointer Forwarding," Joint Conf. of 1996 International Computer Symposium, Dec. 19-21, Kaohsiung, Taiwan, R.O.C., pp. 113-120.

[7] H.Wada et al., "Mobile Computing Environment Based on Internet Packet Forwarding," Proc. Winter USENIX, San Diego, CA, Jan. 1993, pp.503-17.

[8] S.Y. Yi and H. Shin, "A New Locating Scheme for Mobile Objects Based on Their Mobile Characteristics and Replication of location Information," Second International Mobil Computing Conference, Hsinchu, Taiwan, R.O.C., March 25-27, 1996, pp. 123-130.