Database and Location Management Schemes for Mobile Communications

Anna Hać, Senior Member, IEEE, and Bo Liu

Abstract—Signaling traffic incurred in tracking mobile users and delivering enhanced services causes an additional load in the network. Efficient database and location management schemes are needed to meet the challenges from high density and mobility of users, and various service features. In this paper, the general location control and management function is treated as the combination of two parts, the global and local scope. New schemes and methods are proposed, and improvements achieved over established basic schemes are shown by using simulations.

Index Terms—Broadcast, cost, GSM, mobility management, routing, partition, switching.

I. INTRODUCTION

THE Global System for Mobile Communications (GSM) allows users universal and worldwide access to information and the ability to communicate with each other independently of their location and mobility.

Tracking mobile users and routing calls are basic functions of the network system [1], [2]. The system needs to update and provide, upon request, information about the location of mobile users. Queries can be formulated by both the users and the network resource management facility (e.g., aggregate information for adaptive channel allocation) [6].

The criterion for efficient update is low signaling cost incurred by relocation of users between cells. The cost should be kept small enough not to affect network performance. Personal Communication Network (PCN) cellular technology is expected to progress to much smaller cells for greater bandwidth sharing and reuse. The signaling load for location updates is higher due to more frequent relocation for small cells.

Providing information about user location is coupled with updating. Because of frequent relocation of mobile users, especially for small cells, it is impractical to keep track of their exact locations [7] at all times. The more general approach is to keep the location information updated upon query, which requires extra search work. Keeping more complete information updated requires higher signaling cost, while less search work is needed upon query. Finding efficient methods or schemes can balance the effects of these factors and achieve optimal overall performance.

An architectural model of the cellular system [3], [4] is shown in Fig. 1. The part of the system directly serving the mobile users can be viewed as a network extension providing wireless access to the fixed information network. The network extension essentially consists of three elements: wireless mobile terminals (MT's), base stations (BS's), and mobile switching centers (MSC's). Each BS serves a group of MT's currently residing in a cell. An MSC typically connects from up to 100 BS's to trunks of the Public Switched Telephone Network (PSTN) or Integrated Services Digital Network (ISDN). Neighboring MSC's can also connect to each other through direct links. The databases related to location control and management are usually integral parts of their MSC's, but databases may also be stand-alone entities. Typical databases are the home location register (HLR), visitor location register (VLR), and equipment identity register (EIR). These facilities combined with their counterparts in the fixed information network, normally at routers or switches, provide routing and location management functionality for the system.

The switches and databases that perform the functions of tracking users and delivering enhanced services to the users communicate by using Signaling System no. 7 (SS7) [5] in current implementations. However, because SS7 was designed before there were large scale mobile networks, the ability of the existing systems to meet the emerging needs should be examined. In a simple model, it is often convenient to use a
single abstract notion to denote these functional entities, which we call a location server. In some cases, a location server can include an MSC and the HLR, and VLR directly attached to it. It is also possible for one VLR to control the service areas of several neighboring MSC’s. In such cases, a location server can include a major switch node of PSTN, the home location server (HLS) and visitor location server (VLS), and those MSC’s under the control of the VLR.

The system elements are as follows. MT’s terminate the radio link on the user side of the user-to-network interface. They are either vehicle-mounted or hand-held terminals used by subscribers to send and receive information. BS’s terminate the radio link on the network side of the user-to-network interface. The basic role of a BS is to manage radio channels and process information directly related to the establishment, maintenance, and release of radio links. The MSC performs the bulk of wireless and wired call processing in existing cellular systems. For call connections, the MSC processes MT registrations, queries location registers, manages paging processes, selects a BS to handle a call, and establishes connections between the PSTN and selected BS. When MT crosses cell boundaries during a call, the MSC also coordinates handoff activities. The HLR contains master records of the information for a group of MT’s within a given (regional) network. The record includes: 1) a Mobile Identification Number (MIN) to which the MT responds; 2) the Electric Series Number (ESN) burned into the MT during manufacturing; 3) the user’s profile indicating the service features available; 4) the MT’s state (active or not); and 5) a pointer to the last VLR which has reported the user’s whereabouts. The VLR is used when a mobile user roams out of his home area. It contains the information for all subscribers currently residing in its service area. The information includes copies of users’ profiles obtained from their HLR’s when these users enter and register in the VLR’s service area. The VLR assists in keeping track of a visiting MT and delivering calls destined for the MT.

The total signaling load generated by mobile communications in GSM results from the following four system activities.

1) Initiating Calls from Mobile Users: The procedures are similar to those for an ISDN call originating in the fixed network [2]. The MSC currently serving the MT first checks the Mobile Terminal Identifier and the authentication information of the MT through the VLR. If the information is not available, the VLR gets it from the MT’s HLR. A Temporary Mobile Terminal Identifier (TMTI) is assigned to the MT. The MSC sends out the initial address message (IAM) to transit switching nodes to set up the call.

2) Delivering Calls Destined for Mobile Users: Besides the basic procedures for an ISDN call originating in the fixed network, there are additional procedures. First, the system determines the registration area where the mobile user currently resides. Then the network pages the mobile user to determine the current cell of the mobile user. Following that, the system authenticates the MT to ensure that its access to the network is valid. Then the network assigns a cipher key to the radio link and may assign a new temporary local number (TMTI) to the MT. The MSC obtains from the VLR the user profile, which contains the information about features and special services to which the user has subscribed. The signaling traffic incurred is much higher than in the fixed network.

3) Updating the Location Information Upon Users’ Moves: A location update is initiated when a level boundary is crossed by an MT. There are several different situations. If the only cell boundary crossed is inside the service area of an MSC, a local update may or may not be performed, depending on the location management scheme used. If an MT enters the area of a new MSC but under the control of the same VLR, an update throughout the registration area may or may not be performed, depending on the management strategy. If a boundary of areas controlled by different VLR’s is crossed by an MT, the current VLR must obtain the authentication parameters from the previous VLR. The VLR authenticates the MT and allocates a new temporary number (TMTI). Then the user’s HLR is updated about the new VLR, and the HLR informs the previous VLR that the MT is no longer in its control area.

4) Handovers (Handoffs): When an MT crosses the cell boundary during a connection, the new base station takes over the task of maintaining the radio link between the MT and the system.

The total contributions of the above activities to the signaling volume in terms of the number of bytes per call can be expressed as follows:

\[
S_{volume_{percall}} = S_{volume_{m.o}} \times f_{m.o} + S_{volume_{m.t}} \times f_{m.t} + S_{volume_{u}} \times r_{u.c} + S_{volume_{h}} \times r_{h.c}
\]

(1)

where \(S_{volume_{m.o}}\) is the signaling volume (in bytes) for a mobile user originating a call, while \(S_{volume_{m.t.}}\), \(S_{volume_{u}}\), and \(S_{volume_{h}}\) are signaling volumes (in bytes) for a call destined for an MT, one location update (on the average), and one handover, respectively. The other factor in each term is the corresponding frequency of occurrence. In particular, \(f_{m.o}\) and \(f_{m.t.}\) are the percentages in total calls for calls originated by an MT and calls destined for an MT, respectively, and \(r_{u.c}\) and \(r_{h.c}\) are the average number of updates per call and handoffs per call, respectively.

The frequency of occurrence for each activity is determined by users’ behavior, either by their mobility patterns (for \(r_{u.c}\) and \(r_{h.c}\)) or by communication patterns (calling patterns of mobile users for \(f_{m.o}\), and calling patterns of their communication partners for \(f_{m.t.}\)). The signaling volume per call for each activity (\(S_{volume_{m.o.}}, S_{volume_{m.t.}}, S_{volume_{u.}}, \) and \(S_{volume_{h}}\)) is the summation of the signaling volumes for all the basic procedures taken in the activity.

II. THE TWO-SCOPE CONCEPT

A location server can include an MSC, and the HLR and VLR directly attached to it. It is also possible for one VLR to control the service areas of several neighboring MSC’s. In such cases, a location server can include a major switch node
of the PSTN, the HLS and VLS, and MSC’s under the control of VLR.

When a user moves within the area under the control of the same location server, we say the relocation or movement is a “bound roaming.” Otherwise it is an “unbound roaming.” Location control and management can be viewed as the combination of two different scopes, global scope and local scope. For example, to locate an unbound roamer, first the locating mechanism of global scope needs to find the location server for the area in which the roamer currently resides. Then the location server finishes the job in local scope. In local scope, the signaling traffic occurs within the service area of a location server. In global scope, the signaling traffic also occurs between location servers.

A. Local Scope

In local search, the location server has the database (typically a VLR) containing the whereabouts (ideally the exact cell location) of all mobile users currently residing in the area. When an MT makes a move within the area from cell to cell, a message can be sent from an MT to the location server to update the location records. As explained in [6], the cost of the user informing the base station and/or location server of the move is higher than the cost of a base station or location server contacting the user, due to the contention among users and the required authentication procedure. That is, the cost to update location is several times higher than the cost of a simple call delivery with neither the authentication procedure nor searching. This makes keeping constant track of users’ exact locations uneconomical. Limiting the signaling volume for location update decreases cost and keeps location information incomplete but reasonably accurate. However, the searching (e.g., broadcasting) cost incurred due to lack of complete location information must be balanced with the updating cost to make the total cost optimal with consideration of the response time for queries [6]–[8].

Three basic schemes [6] are as follows.

1) Broadcasting: When the exact location of an MT is not known, the location server broadcasts the searching message to a group of base stations which includes the BS currently providing a radio link to the MT.

2) Forwarding Pointers: When an MT moves into another cell, it leaves a pointer to the new cell at the previous cell. Searching is done by following a chain of pointers from the last known position to the current position.

3) List: When an MT moves following a certain pattern, a list of next possible locations with known likelihood can be stored in order. Searching is done from the last known position by trying the new locations successively on the list.

B. Global Scope

The global scope focuses on providing efficient and timely call delivery to a user who roams out of the area of his home location server (with HLR).

When an unbound roamer moves from the serving area of one location server to another, a message is sent to the roamer’s home location server to update its location information in the HLR. The HLR keeps an updated record of the location server which controls the area in which the roamer resides. The HLR is always involved in delivering calls to roamers outside their home area in the most straightforward call delivery scheme, called “dogleg routing” in [10], and “triangle routing” in [11] (these two terms will be used interchangeably). Basically, to set up a call destined for a roamer outside its home area, the call request is first routed to its home location server, and then to the location server currently serving the area where it resides. Fig. 2 shows a scenario of establishing a connection for a call destined for an MT roaming out of its home location server using dogleg routing.

Although in some circumstances, this method, which is used in current mobile systems, can be acceptable, there are many situations where the overhead of passing through the home location server is too high. For example, this happens when the caller is located geographically very close to the roamer location and far from the roamer’s home area. Another situation is when communicating partners frequently exchange high volume of data with a user when he roams away from his home area. For these situations, dogleg routing introduces an unreasonable penalty to the network cost.

The general approach to improve efficiency of call delivery over dogleg routing is to distribute the location information (replication) in the network and to eliminate certain overheads of passing through the home location server (with HLR). Under certain conditions, caching the location information obtained from previous calls at selected network switches can be an efficient way to deliver subsequent calls to the same roaming user [11]. The location information distributed networkwide needs to be selectively updated as users roam over the boundaries of areas served by different location servers, in order to achieve optimal efficiency [5], [9]. The cost of updating should be evaluated together with the gain on

![Fig. 2. Scenario of “dogleg (triangle)” routing.](Image)
call delivery for a net improvement. To limit the updating cost, partitioning location information in a hierarchical database structure spread geographically over the network is a good solution [7], [13], [14].

III. LOCATION MANAGEMENT IN LOCAL SCOPE

Several protocols are proposed in [8] for mobile inter-networking. The Mobile Internetworking Control Protocol (MICP) is used by Mobile Support Stations (MSS’s) to exchange control information, the information about mobile user location, and is used by Mobile Hosts (MH’s) to signal their MSS’s when entering a new cell. A method of combining caching and forwarding pointers is evaluated for efficient location management. Several locating strategies are compared in [6].

Two general characteristics of users’ mobility, spatial locality and temporal locality, can be exploited for the basic principle of limiting the location updating volume and achieving efficiency. The concept of spatial locality means that a limited number of consecutive moves of a user, corresponding to a set of contiguous cells, can be covered by a local zone centered at the starting point of the first move. The updating actions can be limited to only those times when a user crosses the current zone boundary. With the radius of a zone suitably determined, the searching cost (e.g., broadcasting cost) is also limited. The concept of temporal locality is actually another interpretation of spatial locality. That is, within a certain period of time, a user can only roam a limited distance away from the starting point, due to the finite speed. We can expect an upper bound for an update frequency.

Making use of the location information obtained from the previous call helps to lower the updating cost. When a roamer is connected to the network during the call, the location information is updated with much lower cost than purely user-initiated location updating.

Because usually a particular location management scheme performs well under certain conditions and not as well under other conditions, designing a combination of schemes promises a better overall performance than possible using any one scheme.

A. Method of Static Partitioning and Method of Dynamic Neighbor Zone

The service area of an MSC can be divided into partitions composed of a number of cells in the static partitioning method. The service area of a location server is partitioned and location update events are limited to cases of crossing partition boundaries. On call delivery, the location server broadcasts only to the base stations within the partition which covers the latest known position of the user.

In a static partitioning method, the boundaries never change. Savings on both updating volume and searching cost are shown in our simulations, in which we used 15 partitions. A new resource cost is incurred in this method due to the additional computation. The partition information is stored in each MT (in addition to the copy at the server), because an MT has to decide if it should initiate a necessary location update, depending on whether a boundary crossing has occurred. The extra memory and computation cost of each MT is reasonably considered worthwhile. Because technology drives CPU processing costs and memory costs down, and more importantly, because the network is often the performance bottleneck, especially in the GSM environment, using computation and memory costs for savings in network signaling and database loads is cost/performance efficient.

We propose a new scheme to further reduce update costs. Each MT keeps a local zone comprised of a set of base stations. The center of the zone is located at the cell which is the last position of the user reported to the location server. Whenever a call is delivered to the user or the current zone boundary is crossed by the user, the current position is updated and this new position becomes the new center of the zone. The zone boundary is also updated. To determine the zone boundary (the set of cells) for the new center position in a consistent way, the identification codes of base stations reflect their geographic relations. An MT can determine if the current zone boundary was crossed since the latest move by receiving an identification code from the new base station. An implementation example in our simulations uses an identification code similar to the two-dimensional coordinates shown in Fig. 3.

The dynamic neighbor zone algorithm is as follows.

(1) For each move:
   (2) Compare identification code of new cell with current zone boundary.
   (3) If crossing occurs,
       (4) Initiate location update for location server;
       (5) Update zone center as current cell;
       (6) Calculate new zone boundary.

(7) For each call received:
   (8) Update zone center as current cell;
   (9) Calculate new zone boundary.
B. Simulation Model

The basic activities of mobile users with MT’s and of the system are as follows:

1) user requests for calls (more generally, connections) to other hosts,
2) system updating the location information upon users’ moves,
3) system making connections to mobile users that possibly involve searching,
4) call handovers handled by the system when a cell crossing occurs during a call connection.

The first activity on the mobile users’ side does not add complexity to the fixed network, except for the connection made through exchanging radio signals between an MT and a BS. We focus on two types of events, “location updating” and “call delivery.” “Updating” events are systematic activities initiated purely for tracking mobile users. In addition to these events, location information is obtained through the first activity—mobile user-initiated calls—and is also updated through the fourth activity—handovers. The first activity is on the users’ side and independent of the “location updating.” To predict the occurrence pattern of mobile user-initiated calls and make use of the pattern in a systematic way involves additional information about the users’ behavior, and we consider this effect separately. The duration pattern of connections affects the fourth activity. Because we do not want to make our model dependent on the regularity of users’ behavior and the availability of this information, we analyze the additional impact that the effects of the first and fourth activities have on the results from our model.

As discussed in [6] and [12], the occurrence pattern of a user’s move and the calls destined for the user are the main factors determining both the updating and searching costs for a particular management scheme. Typically, the call/mobility ratio [6] is critical in quantitative performance analysis. We define “calls per move” as the number of calls received by a mobile user in the period between two consecutive moves.

The performance criterion is minimizing the combination of the updating and searching costs. We define “cost per event” (an event can be a move or a call) as the total cost (equal to the sum of updating and searching costs) divided by the total number of moves and calls delivered. Another criterion, “cost per call,” is defined as the total cost divided by the total number of calls delivered and describes the impact of mobility on call costs.

In the local scope, the system model consists of the location server under consideration connected to the PSTN and other location servers. The service area of the location server is composed of 121 cells symmetrically distributed around the position of the server, as shown in Fig. 3. There are 121 cells so that their coordinate-like identification codes change from 0 to 10 or −10. A mobile user moves according to “bound roaming” within this area. The characteristic of user’s call-mobility behavior is described by the parameter “calls per move.” A move is defined as a relocation of the user from one cell to a neighboring cell. With the same number of calls per move, there can be different details in the user’s moving pattern, that is, moving around in a small group of contiguous cells, or crossing a number of cells in one direction. This can also have an impact on the management cost. We uniformly generate all the moves. To compare different management schemes, the same set of moves is used for each simulation run.

We select the following cost metrics.

1) Cost of call delivery when the location of the MT (i.e., current cell) is known and the authentication procedure has been done. This is defined as one basic unit, “one message” cost.
2) Cost of location updating. Because the contention among users and the authentication procedure involved require bidirectional messages sent between a user and the base station, and correspondingly, between the base station and the location server, the total cost is defined as the factor α multiplied by the basic unit “message.” When a forwarding pointer is left at the previous base station by the user upon his moving, for location update, there is the cost incurred, which is defined as factor β multiplied by the basic unit “message.”
3) Cost of searching. We consider two basic searching mechanisms, broadcasting and pointer forwarding. The location server broadcasting to a set of γ base stations yields a maximum cost of γ “messages.” In pointer forwarding, one forwarding operation yields a cost of “one message.” A forwarding chain of length k costs k “messages.”

In the figures showing the simulation results, the signaling cost (e.g., “cost per event,” “cost per call”) is measured in a basic unit of “messages” as defined in this section.

C. Performance Evaluation of the Dynamic Neighbor Zone Method

For static partitioning, we divided the service area of the location server into 15 partitions of cells, as shown in Fig. 4.

In the dynamic partitioning method, a zone is a partition of fixed number of cells, centered at the last known position (cell) of the mobile user and moving with the user after a location update. In the dynamic neighbor zone method, we use a pair of integers (similar to two-dimensional x-y coordinates) as the identification code for each base station. The boundary cells are those with distance to the center cell equal to a predefined radius.

Figs. 5 and 6 show the results for the broadcast scheme and the forwarding pointer scheme, respectively, with α = 4 and β = 2 [11], [14]. The results in Figs. 5 and 6 are average values from 20 runs with different random seeds for generating moves and different starting points in the area. From 100,000 moves per run, we find that the standard deviations are negligible.

For both schemes, the dynamic neighbor zone method yields a substantial improvement over the method of static partitioning in the entire range of “calls per move.” The cost saving is obtained by further reducing the updating events.

From Fig. 5, we can see that the limited broadcast scheme performs better than the forwarding pointer scheme only at the
low “calls per move” end. Broadcast cost is usually several times higher than updating cost. If “calls per move” increases, the broadcast cost accounts for a larger portion of the total cost due to the higher number of call delivering events. Unless “calls per move” is small, the forwarding chain is short because the dynamic neighbor zone algorithm does more updates.

In the GSM environment, users’ mobility impacts location management costs. At the low “calls per move” (i.e., high mobility) end, the cost of location updating (or setting the forwarding pointers) accounts for the main portion of the total cost for location management. Because one of the key design goals of the network system is to accommodate increasing demands on call delivery capacities, the portion of the updating costs should be kept small. Also, the number of users making network connections in the same period of time is limited. The users in connectionless state (but still active) account for a certain portion of the total subscribers. Thus, even a small improvement at the low “calls per move” end can lead to significant performance improvements. Fig. 7 shows how the two schemes perform in this range with “cost per call” as the criterion, which makes the mobility impact more visible. We construct an algorithm by combining the limited broadcast scheme with the forwarding pointers scheme. This way we can reduce the updating costs at the low “calls per move” end and obtain the optimal performance in the entire range. Further explanations of Figs. 7 and 8 with the combination scheme follow in the next section.

D. Combining the Limited Broadcast and Forwarding Pointer Schemes

The forwarding pointers scheme performs well in a wide range. However, there is performance degradation at the low
“calls per move” end. The pointer chain is longer when no boundary crossing occurs, and there is no call delivery.

We propose an algorithm combining the limited broadcast and forwarding pointers schemes. At the low “calls per move” end, the algorithm switches to the limited broadcast scheme from the forwarding pointers scheme to improve performance. A value for the maximum chain length can be determined from the following equation:

$$\frac{\gamma}{\beta + 1}$$

where $\gamma$ is the broadcast cost and $\beta \times k_m$ is the accumulated cost of forwarding pointers. To determine from (2) when to switch from forwarding pointers to limited broadcast, an MT keeps a count of the pointers left at the previous base stations for the current chain. The chain is reset whenever a boundary crossing or call delivery occurs. The algorithm is as follows.

(assume the forwarding pointers scheme is in use)

FOR EACH MOVE, DO

IF CURRENT MOVE CAUSES ZONE BOUNDARY CROSSING

BEGIN

INITIATE LOCATION UPDATE;
SET NEW ZONE CENTER AS NEW POSITION;
CALCULATE CORRESPONDING BOUNDARY CELLS;
RESET POINTER COUNT $k = 0$;
END
ELSE

BEGIN

DETERMINE $k_m$ FROM EQUATION (2);
IF $k > k_m$

INFORM SERVER OF SWITCHING TO BROADCAST;
ELSE

INCREMENT $k$;
END

END

FOR EACH CALL RECEIVED, DO

BEGIN

SET NEW ZONE CENTER AS NEW POSITION;
CALCULATE CORRESPONDING BOUNDARY CELLS;
RESET POINTER COUNT $k = 0$;
END

When the current scheme is limited broadcast, we test the instantaneous value of “calls per move” to determine if switching to forwarding pointers scheme should be done

$$M < \frac{\gamma}{\beta + 1}$$

(3)

where $M$ is the number of moves in the period from last call to the current call and is equal to the reciprocal of the instantaneous value of “calls per move.” Note that pointer count $k$ is used in the forwarding pointers scheme.

If (3) holds, we switch the schemes. Test (3) has the following reasoning. Without taking into account the effect of zone boundary crossing, we have the following approximations from our model:

$$\text{COST}_{l.b.} \approx \gamma$$

(4)

$$\text{COST}_{f.p.} \approx M \times (\beta + 1)$$

(5)

Note that (5) is an approximation of the cost from (3) and (4) when switching the schemes. Dependence (3) provides some margin to make it safer for the switching to be taken in the range where forwarding pointers scheme performs better. This helps to reduce unnecessary switching.

When limited broadcast is used, an MT keeps count $M$ for moves and resets $M$ to 0 when a call is delivered. This part of the algorithm is as follows.

(assume the limited broadcast scheme is in use)

FOR EACH MOVE, DO

BEGIN

INCREMENT $M$;
IF CURRENT MOVE CAUSES ZONE BOUNDARY CROSSING

BEGIN

INITIATE LOCATION UPDATE;
SET NEW ZONE CENTER AS NEW POSITION;
CALCULATE CORRESPONDING BOUNDARY CELLS;
RESET POINTER COUNT $k = 0$;
END
ELSE

BEGIN

DETERMINE $k_m$ FROM EQUATION (2);
IF $k > k_m$

INFORM SERVER OF SWITCHING TO POINTERS SCHEME AND ITS NEW POSITION;
ELSE

INCREMENT $k$;
END

END

FOR EACH CALL RECEIVED, DO

BEGIN

IF (3) HOLDS

INFORM SERVER OF SWITCHING TO POINTERS SCHEME AND ITS NEW POSITION;
RESET $M = 0$;
END
Fig. 8 shows a performance comparison of the limited broadcast, forwarding pointers, and our proposed scheme. The criterion is “cost per event.” The performance improvement in the proposed scheme is significant. The peak cost occurs at the point where the probability of switching from forwarding pointers to limited broadcast is the highest. This is due to the fact that the MT informs the server of the switch from forwarding pointers to limited broadcast.

Transmitting any message costs the MT heavily in terms of energy. However, after MT informs the server that it is switching from forwarding pointer to broadcast, it does not need to send a message to the server on every move as long as broadcast mode is in use. Of course, the location can be updated when switching from forwarding pointer to broadcast with negligible extra cost. The algorithm is not symmetric for the two directions of switching. Informing the server of switching from broadcast to forwarding pointer is done when a call is received and practically no extra cost is incurred (only one more bit during the message exchange).

E. Impact of Zone Size on Performance

There are more boundary crossings for the same set of moves in a small zone than in a large zone. The overall cost of location updating in a small zone is also higher. The cost of searching is expected to be higher in a large zone than in a small zone. In the limited broadcast scheme there are more base stations, to which to broadcast, in large zones. In the forwarding pointers scheme, the large zones mean longer pointer chains. The impact of increasing zone size on searching cost should be much smaller for the forwarding pointers scheme than for the limited broadcast scheme. The number of cells in the zone is proportional to the square of the zone radius while the length of the pointer chain increases more slowly than the zone radius. The net impact is the combination of all these factors.

For simulation purposes, the dynamic neighbor zone is defined in three different ways, as shown in Fig. 9. The small zone includes only immediate neighbors. The medium zone, which has a few more neighbors, can be defined as those cells with identification code \((x,y)\) satisfying

\[
|x - x_0| + |y - y_0| \leq \text{Radius} \ (=2) \tag{6}
\]

where \((x_0,y_0)\) is the center cell of the current zone. The large zone contains several farther neighbor cells as well.

Fig. 10 shows a performance evaluation of our combination scheme with different zone sizes. When “calls per move” >1, the variation in zone size makes little difference. In this range, most of the time the forwarding pointer scheme is activated. This scheme is not as sensitive to zone size as the broadcast scheme is. Zone updating is mostly performed at the time of call delivery. The chance of zone crossing is negligible with large values of “calls per move.” This means that we should concentrate on the low “calls per move” end, which impacts the overall performance of the location management scheme.

In Figs. 10 and 11, the zone of medium size tends to give the optimal performance in a wide range. We expect the performance to deteriorate when the zone size is either too small or too large. With too-small zones, the incurred increase in updating cost dominates and cannot be balanced by the savings obtained in searching cost. With too-large zones, since the number of cells in the zone increases faster than the zone radius, the incurred increase in broadcast cost dominates and cannot be balanced by savings in updating cost.

At the low “calls per move” end, we can see that larger zones can yield a net improvement over medium zones when “calls per move” decreases. This feature is shown in Fig. 11, where we use “cost per call” to demonstrate mobility impact on call delivery cost. This result gives us an indication of how to achieve further performance improvements at the low
“calls per move” end. Larger neighbor zones are appropriate for those users who keep their MT’s active but get few calls over a long period of time.

F. Other Factors Involved in Location Management

Let us consider the effects of a user’s call initiation. We assume the current server has all necessary data of users’ profiles and this information is independent of the location management scheme in local scope. In general, this has a positive effect on reducing the updating cost defined in our model due to the fact that a user’s location information is also updated when he initiates a call. The neighbor zone can be updated when the user initiates a call. The resulting saving is the same for both broadcast and forwarding pointers schemes. In addition, the chain is reset if the forwarding pointers scheme is active. That scheme can benefit more from a mobile user initiating calls than the broadcast scheme, especially at relatively low “calls per move” end. The result is that the switching point between the two schemes for our proposed combination scheme, or equivalently, the cost peak of the performance curve shifts to the left (lower end) because of this difference.

When some zone boundary crossings occur during the periods of call connection, the location updates are performed automatically. The number of effective moves (relocations without simultaneous updating) is lower than the total number of moves. The effect of handovers is to make the effective “calls per move” higher. The curves in our simulation should shift to the left (low “calls per move” end).

Finally, let us consider the effect of response time upon call deliveries. In addition to the broadcast scheme and the forwarding pointers scheme, the cost of the list scheme was studied in [6], where calls were first delivered to the users’ previous position and then a list of possible new locations was tried in order of decreasing probability. We do not consider this scheme, mainly for the reason that the response time tends to be high. The limited broadcast scheme has shorter response time on the average than the forwarding pointers scheme.

However, since we prevent long chains with our proposed method, the difference is small.

IV. LOCATION MANAGEMENT IN GLOBAL SCOPE

There are two approaches to location management in global scope. One approach keeps the addressing scheme of Internet Protocol (IP) suite currently used in the fixed network, which means that each mobile user (for his MT) has a private (static) address assigned in a way consistent with his local (home) network. Naturally, the HLR keeps constantly updated location information about the MT, at least the location server that serves the area in which the MT currently resides. This information can be retrieved by the home location server, as in the case of dogleg routing. To handle the situations where pure dogleg routing is too expensive, replicas of the location information are distributed networkwide [11], [12].

In particular, [11] and [12] distinguish from the rest of the network the portion that has frequent communications with a certain user. In addition to keeping an accurate location information in the HLR for the user, the information is also distributed in this portion of the network (called the “friend network” in [12]) with a possibly variable degree of accuracy and update frequency. In [14], the performance of three updating policies was compared. These policies are: “complete update,” in which each relocation is recorded in all location servers; “selective update,” in which each relocation is recorded only in a “friend network”; and “lazy update,” in which other location servers query the home location server when the information is needed. In [11], an on-line adaptive updating algorithm was studied by using simulations. For each communicating partner of a mobile user, the algorithm predicts dynamically the cost of updating and the cost of searching without update, based on the on-line statistics of the user’s communication and recorded mobility pattern.

The second approach to location management in global scope does not use a static address for each MT compatible with the current IP addressing scheme. For a particular user, no location server is assigned to be the user’s home location server. The distribution of the location information for a mobile user is determined by his mobile history and the current location. The model of hierarchical location server architecture or similar structures is discussed in [7], [13], [14], and no dogleg routing is used. The savings on updating location information and delivering calls to roamers is achieved by properly partitioning and selectively updating the information in the hierarchy.

We observe that it is beneficial to keep the home address for each MT, relative to the roaming address, which is dynamically assigned by the location server when an MT roams out of its home area and resides in this server’s area. This provides a compatibility with the current addressing scheme of the IP protocol suite. Also, with a home address and home location server assigned to mobile users, the updated location information can be stored in the home location servers (with the HLR), which makes the cost lower for the hosts communicating with a mobile user within the home area. A number of partners in a user’s home area may frequently

Fig. 11. Impact of zone size with “cost per call” as performance criterion.
communicate with him. In this case, the first approach to location management in global scope is preferred.

Due to different users’ behavior (calling behavior and mobile behavior) and various geographic situations of an MT with respect to its home location server and its communication partners, a simple location management scheme without adaptive features for varying cases cannot provide a good overall performance. Thus, a compound strategy which incorporates different schemes efficient in different situations yields higher overall efficiency.

Updating the location information of a roamer locally within a certain distance from his current location is used in the hierarchical location server architecture. This can be useful for situations where a number of hosts in an area frequently communicate with a roamer residing in the same area but far from his home area. The method of locally updating makes the updating cost relatively small and provides high efficiency for these hosts to communicate with the roamer.

The portion of the network that communicates frequently with a mobile user should be distinguished from the rest of the network for the purpose of location information updating. In [11], an adaptive updating strategy is used to distribute location information of an MT at the source hosts which frequently initiate calls to the MT. Another possible way is to update the information at the location servers of these hosts or, alternatively, only at some common nodes on the route to the MT’s previous location, in order to further reduce the updating cost. Confining the distribution of users’ location information to a small number of points not only helps to reduce the updating cost, but also helps to preserve the privacy of the mobile users’ whereabouts.

We choose a nonhierarchical network model similar to the network in [11], [12], but we take a different approach. In [12], the “friend network” for an MT is statically known and is composed of all hosts that communicate with the MT frequently. In [11], an MT dynamically records the frequency with which each source communicates with the MT and the frequency of the MT’s relocation in areas of different location servers. The two frequencies are then used to calculate the average cost of triangle routing and the average cost of direct routing with constant updating. The scheme with a lower average cost is chosen. In this approach, a user’s moving pattern and his communication partners’ calling patterns do not change rapidly. Also, the two frequencies are not updated in each step, which reduces the effectiveness. In our model, we take a more general adaptive approach with well-defined information about the users’ behavior.

A. Simulation Model

Our network model consists of 68 nodes and 108 location servers geographically distributed. Some servers are directly linked to corresponding backbone nodes, and the remaining are serving smaller “administrative domains” which have higher user/call densities (i.e., fewer calls per user).

The results are shown for one MT in the network. The source hosts which communicate with the MT are fixed hosts. We first consider a simple case where each $SH$ is communicating with the MT independently from the other $SH$’s. Then we consider additional effects for the situation where a group of source hosts communicates with the MT during the same period of time.

The service areas of all location servers provide geographically complete coverage. A location server does not necessarily have a direct link to all location servers with whose service areas it has common boundaries. On the average, a location server is directly connected to five other servers [11], [12], [14]. Each move of the MT is defined as a boundary crossing between service areas. Different schemes are compared by using the same set of movements.

There are two possible methods to measure the network cost for packets to be sent (or call delivered) from one host to another [11]. We use $Cost(H_1, H_2)$ to denote the cost from host $H_1$ to host $H_2$. One method uses the distance as the cost metric. This is often a good approximation because establishing a connection over a long distance often involves more transmission and processing costs. When using a distance as the cost metric, each MT can store the distances (presumably the shortest paths) between each location server pair. An MT needs the cost information for an adaptive location management scheme. Additional network traffic incurred from obtaining this information should be minimized.

Another method to measure the network cost uses the number of hops traversed by the packets. This metric is more practical. Due to fluctuation of the network traffic, the packets are not necessarily sent along the shortest path because of possible local congestions. Thus, the metric of hop count is a more accurate measure of the current cost. This information can be obtained by an MT from a packet field similar to the IP’s “time to live” (TTL) field. However, the cost information for a particular source host stored in an MT can only be updated when a call is delivered from that source.

To avoid additional network traffic for obtaining the cost information, we use an approximate estimate from the prestored distance parameter in an MT and the previous hop count for the source host in the period while no active connection is maintained. We assume that each MT has reasonably accurate information about the current network cost for the purpose of using adaptive location management schemes. We use distance as the cost metric. For each location server, the shortest-path spanning-tree is calculated with Dijkstra’s algorithm to obtain its distance to other servers. We also assume that the distance (cost) from any MT or source host to its current location server is one basic unit. Note that the proposed algorithm is not affected by the choice of cost metric.

As in the local-scope example, we use “calls per move” as the parameter for performance evaluation. The performance criterion is “cost per call,” where the cost includes both the cost of delivering the call and the possible updating cost.

B. Comparison of Basic Schemes

We consider three basic schemes as follows.

1) Dogleg routing: No updating upon the MT’s move.
2) Basic updating: Direct routing with updating at each source host upon the MT’s move.
3) **Pointers:** Similar to the forwarding pointer scheme. An MT leaves a pointer to its current location server at the previous server. To deliver a call to the MT, a pointer chain is followed. The (start of) chain is reset after a call is delivered.

In this section, only one source is considered communicating with the MT. Further results for a set of source hosts communicating with the MT are presented in the section on “friend network.”

We simulated four situations as follows.

**Case I: The Source Host is Not Close to the MT’s Home Server:** The source host communicating with an MT is not close to the MT’s home server (with respect to the mean distance from the outer servers to the center of the coverage of the cellular network). The MT moves freely and can travel into the area of every server in the set of 100,000 moves generated randomly for each run. A typical result for a certain MT source pair is shown in Fig. 12.

When the “calls per move” is not small (≥0.3), the pointers scheme performs well. Because Cost(source, home) is not small and Cost(MT, home) is not small either on the average, the dogleg scheme is not cheap in a wide range of “calls per move.” Only at the low end of “calls per move” (≤0.2), the costs of the pointers scheme and the basic updating scheme soar well above the dogleg line. For basic updating, any cost of updating is wasted if there is no connection over many moves. For the pointers scheme, the performance deteriorates rapidly with the increase of the chain length.

**Case II: The Source Host is Close to the MT’s Home Server:** The source host φ communicating with an MT is not close to the MT’s home server (with respect to the mean distance from the outer servers to the center of the coverage of the cellular network). The MT moves freely and can travel into the area of every server in the set of 100,000 moves generated randomly for each run. A typical result for a certain MT source pair is shown in Fig. 13.

The simulation result shows that when “calls per move” is large enough (≥1.0) the performance differences are much smaller than for the last case of the three schemes. In a wide range of “calls per move” (<1), dogleg performs significantly better. The adaptive compound scheme, as defined in the next section, performs better in these situations.

**Case III: An MT is Moving in Areas Close to the Source but Far from Its Home Server:** We consider a special case where an MT is moving in areas close to the source host but far from its home server. This is an important subset of Case I. The result for a certain MT source pair is shown in Fig. 14.

Except for the extremely small “calls per move” end (which is not shown), the basic updating scheme has the best performance. This is consistent with our reasoning for the benefit of “updating locally” at the source hosts close to the MT. We also expect that combining basic updating with the pointers scheme can improve the performance of the pointers scheme because the chain length can be reset to avoid fast deterioration.

**Case IV: An MT is Moving in Areas Close to Its Home but Far from the Source:** An MT is moving in the areas close to its home server but far from the source host. This is an important subset of Case I. The result for a certain MT source pair is shown in Fig. 15.

The dogleg scheme has the best performance. From the MT-home-source triangle, we can see that the geometric feature for this case guarantees that the overhead incurred by dogleg is very small or negligible.
The compound scheme is composed of two modes, dogleg mode (mode 1) and pointers-updating mode (mode 2). Because the latter is the base mode, the current mode always starts with mode 2. There are two conditions under which the dogleg scheme is activated, “looking forward,” and “looking back.” The “looking forward” condition refers to dogleg mode and geographical prediction facility. The “looking back” condition refers to pointers-updating mode and accumulated data.

Let us consider the “looking forward” condition. When an MT is in Case II or IV described in the previous section, dogleg is the choice during the coming several moves. In particular, an MT checks the following inequality after each of its moves and switches to mode 1 if the inequality holds:

\[ \text{Cost} (SH, HM) + \text{Cost} (HM, LS_{\text{curr}}) < \min (\text{Cost}_{\text{pointer}}, \text{Cost}_{\text{updating}}) \]  

(8)

where the left side is the cost for delivering the next call by using the dogleg scheme, while the right side is the cost if the current mode is mode 2.

The “looking back” condition is even simpler. In Case I, the “cost per call” goes up for the pointers scheme and the basic updating scheme at the low “calls per move” end. An MT can record the updating cost since the last call was delivered to the MT from the source host. If the accumulated cost has reached a certain point before the next call delivery, we should activate the dogleg scheme. We adopt the following condition for the mode switching:

\[ \text{Cost}_{\text{u.a}} + \text{Cost}_{\text{updating}} > \text{Cost}_{\text{dogleg}} \]  

(9)

where \( \text{Cost}_{\text{u.a}} \) is the accumulated updating cost.

We also consider the condition for switching from mode 1 back to mode 2. We distinguish the situations where the previous switching is due to (8) or due to (9). The MT can record the type of previous switching (from mode 2 to mode 1) in “switching-type” as either “looking-forward” or “looking-back.” When a call is delivered to the MT, it switches back to the pointers mode if “switching-type” is “looking-back.” Otherwise, the MT checks inequality (10), and if it holds, the MT makes the switch:

\[ \text{Cost} (MT, HM) + \text{Cost} (HM, SH) > \delta \times \text{Cost} (MT, SH) \]  

(10)

where \( \delta \) is a constant. (10) gives the upper bound for the overheads allowed when using the dogleg scheme. Based on the simulation experiments (Figs. 16–19), a reasonable value for \( \delta \) is between 1.3 and 1.6. With smaller \( \delta \), there are more unnecessary switchings which add cost. With larger \( \delta \), there is more chance for dogleg to be unnecessarily active.
Our proposed algorithm is as follows.

(CURRENT MODE IS SET TO THE POINTERS MODE 2 AT THE START)

FOR EACH MOVE OF THE MT, DO
BEGIN
IF MODE = MODE 2
BEGIN
IF INEQUALITY (8) HOLDS
BEGIN
MODE := MODE 1;
SWITCHING_TYPE := "LOOKING FORWARD;"
INFORM SOURCE HOST OF MODE SWITCHING AND UPDATE RECORD OF CURRENT SERVER OF MT;
END
IF INEQUALITY (9) HOLDS
BEGIN
MODE := MODE 1;
SWITCHING_TYPE := "LOOKING BACK;"
INFORM SOURCE HOST OF MODE SWITCHING AND UPDATE RECORD OF CURRENT SERVER;
END
END /* MODE WAS MODE2 UPDATING POINTERS */
IF MODE = MODE2
IF INEQUALITY (7) HOLDS
BEGIN
UPDATE RECORD OF CURRENT SERVER AT THE SOURCE HOST;
INCREASE ACCUMULATED UPDATING COST BY THE NEW COST;
RESET POINTER CHAIN;
END
ELSE
LEAVE THE NEW POINTER AT THE LAST SERVER; /* DO NOTHING */
END

FOR EACH CALL DELIVERED TO THE MT FROM THE SOURCE HOST, DO
IF MODE = MODE 1
IF SWITCHING_TYPE = "LOOKING BACK;"
MODE := MODE 2;
ELSE
BEGIN
IF (10) HOLDS
MODE := MODE 2;
END
ELSE
BEGIN
RESET POINTER CHAIN;
RESET ACCUMULATED UPDATING COST TO ZERO;
END
Fig. 16. Performance of the compound scheme in Case I.

Fig. 17. Performance of the compound scheme in Case II.

The simulation results for the proposed algorithm are shown in Figs. 16–19. Each result corresponds to Cases I–IV shown in Figs. 12–15. Except for the very low “calls per move” end (≤0.1), the compound adaptive scheme has the optimal performance in various situations. At this end, the cost is reduced to less than one-third over the dogleg line, instead of going up. This improvement is significant; the extra mode switching in this range (and the additional cost incurred) cannot be avoided since the next call cannot be predicted precisely.

D. The Environment of the “Friend Network”

We use the concept of “friend network,” mainly composed of the set of source hosts for the MT; it can also include the servers frequently involved in related location management. In the preceding sections, only one source communicates with the MT. The implied assumption is that each source host communicates with the MT without any connection to the other source hosts. This is not a realistic case. However, we
can show that all the results obtained under this assumption are still valid with one modification to the definition of the update cost. Then we explore the possibility of further improvement in the environment of a “friend network” and propose a method for the real situation.

When a set of source hosts communicates with the MT, a subset of them receives updates upon each move. Ideally, if the minimum spanning-tree starting at the current server of the MT can be computed, the updating cost for the source host $SH$ is the cost from its parent in the tree to the $SH$. More generally, the updating cost is defined as the cost from the server (between the MT and the $SH$) that relays the updating information. We claim that the general features and qualitative conclusions in previous results still hold (with some numerical variations due to the modification.) In particular, the larger number of source hosts often results in the lower cost per source host, while increasing the total network load. The question now is how to further reduce the total network load.

We consider a group of source hosts residing in the areas of several location servers close together. The entire set of source hosts is geographically distributed in partitions, instead of being scattered sparsely. We introduce the concept of “image home” for such a group of source hosts when they are far from the home server of the MT. If we choose the server serving the $SH$ in the group that communicates with the MT most frequently as an “image home” server, then the other $SH$’s can view it as the real home server and their costs can be reduced dramatically. The original “calls per move” through the “image home” should be high enough to avoid extra cost for constant updating. However, this condition can be relaxed. As long as the cost for additional updating at the “image home” is relatively small with respect to the savings in the source hosts, the method of “image home” performs well. The simulation result for such a group of $SH$’s is shown in Fig. 20 with the average saving for one $SH$ in the group using the “image home.”

V. CONCLUSION

We proposed a two-scope concept to manage the extra burden incurred to track mobile users. The general location control and management functionality was treated as the combination of two parts, the global and local scopes.

We incorporated two new ideas into the basic local-scope location management schemes. First, we introduced a “dynamic neighbor zone” scheme to further reduce the updating cost over static partitioning. Second, by combining the limited broadcast scheme with the forwarding pointers scheme, we provided an overall optimal performance in the entire range of call/mobility ratio.

For the global-scope location management, we investigated the performance of three basic schemes in four typical situations. The simulation results show how different schemes can be chosen due to signaling cost under different situations. Based on these results, we proposed a compound scheme which activates the most efficient component scheme depending on the situation. By using our general network model, the compound scheme shows the optimal performance for all considered situations except for the very low call/mobility ratio end. At this end, the signaling cost was reduced to one-third
over the cost of dogleg. This is cost saving, considering that it is impossible to activate the dogleg at the exact time of the next call.

We emphasized the importance of performance improvement in the range of low call/mobility ratio. The mobile users with active MT’s and low call/mobility ratio account for a large portion of the total subscribers. Our scheme provides performance improvement in this range.

A good location management scheme must adapt to various situations. We made the model simple and general, without such assumptions as the regularity of users’ behavior and the availability of the information on behavior patterns.

REFERENCES


Anna Hać (M’85–SM’89) received the M.S. and Ph.D. degrees in computer science from the Department of Electronics, Warsaw University of Technology, Poland, in 1977 and 1982, respectively. She is a Professor in the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu. She was a Visiting Scientist at the Imperial College, University of London, U.K., a Postdoctoral Fellow at the University of California at Berkeley, an Assistant Professor of Electrical Engineering and Computer Science at The Johns Hopkins University, a Member of Technical Staff at AT&T Bell Laboratories, and a Visiting Professor in the Center for Wireless Communications at the University of California at San Diego. Her research contributions include system and workload modeling, performance analysis, reliability, modeling process synchronization mechanisms for distributed systems, distributed file systems, distributed algorithms, congestion control in high-speed networks, reliable software architecture for switching systems, telecommunications and wireless networks, and network management. She has published 120 papers in archival journals and international conference proceedings.

Dr. Hać is on the Editorial Advisory Board of Wiley’s International Journal of Network Management.

Bo Liu received the B.S. degree in radio and electronics from Peking University, Beijing, China, in 1986, and the M.S. degrees in physics and astronomy and in electrical engineering from the University of Hawaii at Manoa, Honolulu, in 1992 and 1996, respectively.

He is a Software Engineer at Quest Software, Newport Beach, CA, working on database systems.