A Comparative Cost Analysis of Degradable Location Management Algorithms in Wireless Networks

ING-RAY CHEN[1] AND BAOSHAN GU

Department of Computer Science, Virginia Tech, 7054 Haycock Road, Falls Church, VA 22043, USA
Email: irchen@cs.vt.edu

In this paper, we develop a uniform framework to provide a cost analysis of location update and search operations for a class of degradable location management algorithms in personal communication service (PCS) networks for tracking mobile users in the two-tier HLR (Home Location Register)–VLR (Visitor Location Register) structure. Depending on the algorithm employed, the PCS may be in a degraded state in maintaining the location of a mobile user. We classify existing location management algorithms based on how well the location information is maintained in terms of the costs associated with location updates and develop a two-level hierarchical modeling framework to analyze the performance characteristics of these algorithms. Specifically, the high-level model calculates the total cost incurred to the PCS network as a result of location-update and call-delivery operations during the period between two consecutive calls. The low-level model is a stochastic model that estimates the values of high-level model parameters. We show that by utilizing simple Markov models at the low level, we can assess and compare the performance characteristics of degradable location management algorithms easily. The basic scheme used in the standard IS-41 and GSM protocols, the paging and location updating algorithm (PLA), the forwarding and resetting algorithm (FRA) and the local anchoring algorithm (LAA) are used as examples to demonstrate the applicability of our approach. We also show how the modeling approach developed can be extended to the analysis of algorithms for handling service handoffs in the two-tier HLR–VLR architecture.

Received 5 June 2001; revised 3 November 2001

1. INTRODUCTION

In a personal communication service (PCS) network, a location management scheme must handle two operations efficiently: location update and call delivery. The former operation occurs when a mobile user moves to a new location; the latter operation occurs when there is a call for the mobile user and the network must deliver the call. Since it is possible for a mobile user to move from one place to another while it is being called, the PCS network must track the location of the mobile user in order to correctly deliver calls. A well-known basic and simple scheme is to update the location of each mobile user whenever it moves to a new registration location. This location management scheme exists in IS-41 [1] in the United States and GSM [2] in Europe, and is commonly known as the basic HLR (Home Location Register)/VLR (Visitor Location Register) scheme based on the two-tier HLR–VLR structure.

In recent years, various location management strategies for reducing the location management cost have been proposed in the literature [3, 4, 5, 6, 7, 8]. Most of the strategies studied are based on the HLR–VLR two-tier structure to improve IS-41 and GSM. There are also location management strategies being designed based on tree architectures with databases stored in tree nodes [9, 10, 11, 12]. The intent is to design the best location management scheme that can minimize the location update cost (minimizing the cost in location updates) without increasing by too much the cost of call delivery. Therefore the goal is to minimize the total cost incurred in servicing location-update and call-delivery operations. For the two-tier HLR–VLR architecture, it has been separately reported that when the frequency of incoming calls is higher than the mobile user's mobility, that is when the call-to-mobility ratio (*CMR*) is high, the location cache scheme [13] is effective, while when *CMR* is low the forwarding and resetting algorithm (FRA) [14, 15], the paging and location update algorithm (PLA) [16] and the local anchor algorithm (LAA) [17] are effective. There is no systematic way to categorize these location management algorithms and to compare their performance characteristics [8]. In particular, it is not clear how these algorithms will fare under identical workload conditions.

We are motivated to develop a uniform framework to analyze various update and search schemes. This paper will focus on the two-tier HLR–VLR architecture. We first

[1]Corresponding author.

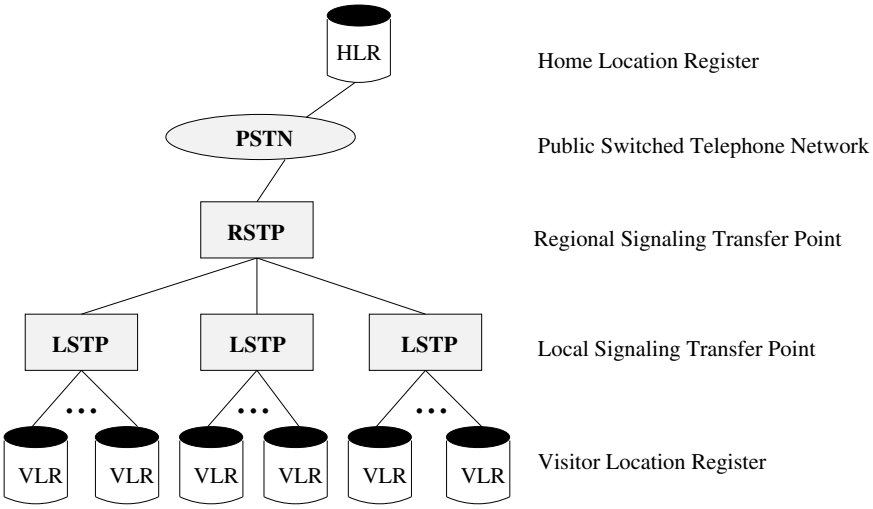

FIGURE 1. A hierarchical PCS network.

discuss the notion of degradable location management as a basis for classifying location management schemes. Then, we propose a two-level hierarchical performance model as a uniform framework for assessing and comparing the performance characteristics of location management algorithms in the two-tier HLR–VLR architecture. At the high level comes a model that calculates the total cost incurred to the PCS network as a result of location-update and call-delivery operations during the period between two consecutive calls. At the low level comes a stochastic model that estimates the values of high-level model parameters. The high-level model is generic, while the low-level model developed is algorithm-specific so as to capture the behavior of the PCS network operating under a particular algorithm. We show that, by utilizing simple Markov models at the low level, we can assess and compare the performance characteristics of degradable location management algorithms easily. Unlike previous performance studies which have concentrated on only a particular scheme or have been based on simulations, the uniform framework developed is based on analytical modeling and can be applied uniformly to analyze location management location schemes in the two-tier HLR–VLR architecture. In this paper, we illustrate our approach with three algorithms in the two-tier HLR–VLR architecture: PLA, FRA and LAA. Another contribution of this paper is that we show how the framework developed can be extended to the analysis of algorithms for handling service handoffs in the two-tier HLR–VLR architecture.

The rest of the paper is organized as follows. Section 2 discusses the notion of degradable location management algorithms and our system model. Section 3 exemplifies the notion of degradable location management algorithms with PLA, FRA and LAA. Section 4 describes our proposed two-level hierarchical framework for analyzing degradable location management algorithms. Low-level analytical models for describing the behaviors of the PCS system operating under PLA, FRA and LAA are developed to parameterize the generic high-level model based on the proposed two-level hierarchical modeling method. Section 5 compares these degradable location management algorithms under identical network conditions and provides a physical interpretation of the results. Section 6 discusses how to apply the methodology developed in the paper to analyze service handoffs. Finally, Section 7 summarizes the paper and outlines the applicability of the analysis results along with some possible future research areas.

2. DEGRADABLE LOCATION MANAGEMENT

No assumption is made regarding the structure of the PCS network. Conceptually, the HLR of a mobile user is at a higher level, while all VLRs that the mobile user wanders into from time to time are at the lower level. There may be some network switches connecting the HLR to VLRs in the mobile network. For IS-41, all service areas are divided into registration areas each corresponding to a VLR. Therefore, when a mobile user moves to a new registration area, the mobile user can send the registration information to the new VLR which in turn can perform appropriate update actions, depending on the scheme being used. A VLR or even the mobile unit itself may also keep a small database for location management. The type of database kept by each VLR or mobile unit will be different depending on the location management scheme employed. Obsolete database entries will be purged periodically. Thus, no explicit deregistration messages are needed so as to reduce the communication cost [18].

Figure 1 illustrates a possible hierarchical PCS network as discussed in [13]. The HLR and VLRs each contain a database for location management. The intermediate switches such as RSTPs (Regional Signaling Transfer Points) and LSTPs (Local Signaling Transfer Points) are only used for connecting VLRs and the HLR.

Under our proposed notion of degradable location management, a PCS network operating under a location

management algorithm can be in a 'strong' or 'weak' state. The degree of weakness depends on the extent to which the location information has been degraded since the last location update operation was performed by the HLR. These strong and weak states represent the capability of the system to service call-delivery operations. When the system is in the strong state, it means that the HLR can find the mobile user by directly consulting its own database and it can thus service the call delivery efficiently. Conversely, when the system is in a weak state, it means that the HLR must consult location databases stored elsewhere to find the mobile user and it will thus take a longer time to locate the user. How fast the system's state goes from a strong state to a weak state depends on the particular location algorithm employed. We call a location management algorithm *degradable* if the state of the PCS network regarding the *exact* location of a mobile user can evolve over time.

The spectrum of degradable location algorithms encompasses all existing location management algorithms, with the difference being in how fast and in what way the system's state degrades over time as the user moves across VLR boundaries. At the one end of the spectrum lies the basic scheme currently adopted by IS-41 in the United States and GSM in Europe. It represents the extreme case in which the state of the PCS is in the *strongest* state all the time. In this case, the HLR is updated whenever the mobile user makes a move across the VLR boundary. Thus, the exact location of the mobile user is known to the HLR all the time and the HLR can find a called mobile user efficiently by directly consulting its up-to-date database. At the other side of the spectrum lies the paging method [19] in which the location of a mobile user is not updated to the HLR at all. In this case, the performance of the PCS network degrades rapidly as more moves are made since it has to spend more time to page a called mobile user. Most existing schemes lie in the middle part of the spectrum.

An example is the FRA(k) scheme [14] where k is a design parameter. The basic idea under FRA is that, whenever a mobile user moves to a new VLR area, only a pointer is set-up between the two involved VLRs and there is no need to inform the HLR about the location change. Therefore, when a call is delivered, the PCS network must follow a chain of VLR database pointers to locate the current VLR. Since the HLR does not know the exact location of the mobile user and has to follow a chain of VLRs' databases in order to locate the user, the PCS network's performance in locating the mobile user degrades over time as more and more moves are made by the mobile user. In this paper, we refer to such an algorithm as FRA(k) where k is the maximum length of the forwarding chain at which an update operation is performed by the HLR.

Another example is the PLA(n) scheme [16]. The basic idea under PLA is that the HLR records a VLR which serves as the agent of the mobile user in an $(n-1)$-distance local region where an i-distance region covers all VLRs within a distance i from the agent. When the mobile user makes a move within the local region, there is no need to do any update operation at all. When a call arrives, the system performs a search in the local region by means of outward paging, starting from the agent until the mobile user is found. Therefore, the performance of the PCS network also degrades over time since the location of the mobile user becomes fuzzier to the HLR as more and more time has elapsed since the last update operation was performed. Also, the effectiveness of the search operation depends largely on whether the mobile user is near the agent. In this paper, we will call such an algorithm PLA(n) where n defines how large the $(n-1)$-distance region is.

TABLE 1. Location update cost when $CMR = 0.1$.

Algorithm (with a parameter)	Location update cost relative to IS-41
IS-41	1.0
LAA(2)	0.60
FRA(2)	0.58
PLA(2)	0.57
LAA(3)	0.48
FRA(4)	0.41
PLA(3)	0.29
Paging	0.0

The LAA(n) scheme [17] is another example of degradable location management. The basic idea is the same as the IS-41 scheme except that instead of reporting all location changes to the HLR, all updates are reported to a VLR called the local anchor (LA) of the mobile user. The LA is replaced and made known to the HLR when the mobile user moves across a regional switch boundary. Searching for the mobile user is essentially a three-step process under LAA: the HLR first, then the LA, and finally the VLR which currently covers the mobile user. We will call such an algorithm LAA(n) where n refers to the size of the n-level region covered by a regional switch in the mobile network.

We envisage that the whole spectrum of degradable location management algorithms can be classified by using the following performance metric: the network cost due to location update between two successive calls to a mobile user. This metric represents the extent to which an algorithm allows the PCS network to be degraded regarding the exact location of the mobile user after it services a call, but before it services the next call. In this spectrum, the basic IS-41 algorithm has the highest location update cost; the paging algorithm has the least location update cost (actually 0); and all other algorithms fall within these two ends. This paper discusses ways of quantifying the location update cost for a given location management algorithm with respect to the basic scheme, thus allowing us to determine which part of the spectrum a given algorithm falls into. This classification is useful for those systems designed to satisfy certain design goals such as servicing calls within a time bound for a particular type of mobile user while still being able to limit the location update cost within a tolerance limit.

Table 1 shows a snapshot of the spectrum that covers some algorithms considered in this paper for the case when

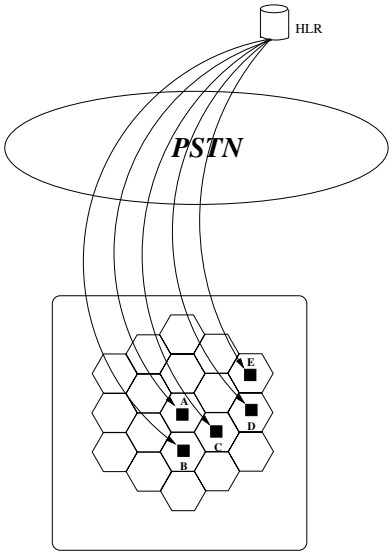

FIGURE 2. Update operations performed under IS-41.

(a) the call-to-mobility ratio (*CMR*) is equal to 0.1, i.e. the frequency of moves is about 10 times of that of calls; (b) the ratio of the average communication cost between two VLRs to the average communication cost between the HLR and a VLR is 0.3. The data shown in Table 1 will vary as the values of the above two parameters vary. The algorithms listed are ordered in descending order of the location update cost between two consecutive calls. In this paper, we will show how the data shown in Table 1 can be obtained. The results obtained will be useful in classifying mobile users into priority classes based on their quality of service (QoS) requirements regarding how fast they want their connection calls to be delivered. The assignment can be such that the QoS requirements are proportional to the location update costs shown in Table 1 such that mobile users in separate QoS classes are being served by separate location management schemes. This will result in per-class-based location management which will be simpler to implement and more scalable than the per-user-based counterpart [8].

3. DEGRADABLE LOCATION MANAGEMENT ALGORITHMS

In this section, we exemplify the notion of degradable location management algorithms by giving a more detailed description of some existing location management algorithms. In the next section, we develop analytical models based on hierarchical modeling to describe the performance behaviors of these algorithms and to obtain the location cost of the PCS network between two consecutive call operations (as in Table 1) so as to classify and compare these algorithms.

3.1. Basic HLR/VLR

Under the basic HLR/VLR scheme, a mobile user is permanently registered under a location register called the HLR. When the mobile user enters a new VLR area, it reports to the new VLR which in turn informs the HLR by means of a location update operation. When there is a call asking for the mobile user, the PCS network checks with the HLR of the mobile user to know the current VLR of the mobile user and then the call is delivered to the current VLR. In Figure 2, when a mobile user moves from VLR A to VLR B, the HLR is informed to point to VLR B. All subsequent moves to C, D and E behave similarly. That is, the HLR is updated to point to C, D and E, respectively, in these subsequent moves.

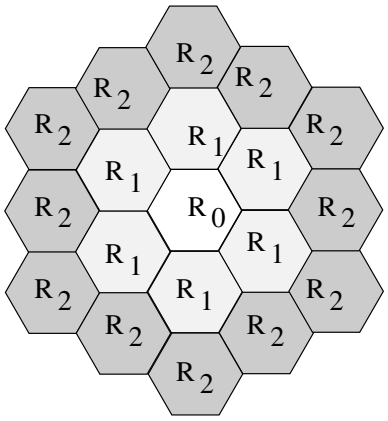

FIGURE 3. SDF partitioning under the hexagonal model.

3.2. PLA

PLA(n) is a movement-based location update scheme discussed in [16]. Under this scheme, a mobile user performs a location update to the HLR only when the distance between the agent and the current VLR is greater than or equal to a predefined distance value n. The VLR which performs the last update operation to the HLR is called the *agent* of the mobile user. Figure 3 illustrates a shortest-distance-first (SDF) partitioning scheme under the hexagonal network coverage model where each hexagon denotes a VLR. The SDF partitioning scheme divides the VLR areas into ring areas, starting from ring 0 (labeled R_0), ring 1 (labeled R_1) and so on, where ring i covers all VLRs which are within a distance of i from the agent. If the mobile user makes a movement for which the distance away from the agent is still smaller than the specified distance value n, no location update operation to the agent or the HLR is required. Such a movement is called a local movement. Otherwise, a location update operation to the HLR must be performed and the new VLR becomes the agent. A movement which causes the HLR to be updated is called a regional movement. When a call arrives, the HLR searches for the mobile user using an outward paging method, i.e. starting from ring 0 (the agent itself), ring 1 and so on until the mobile user is found.

In Figure 4, assume that $n = 3$, VLR A is the agent initially and the SDF partitioning scheme is used to divide the registration areas into ring areas such that VLR A is in

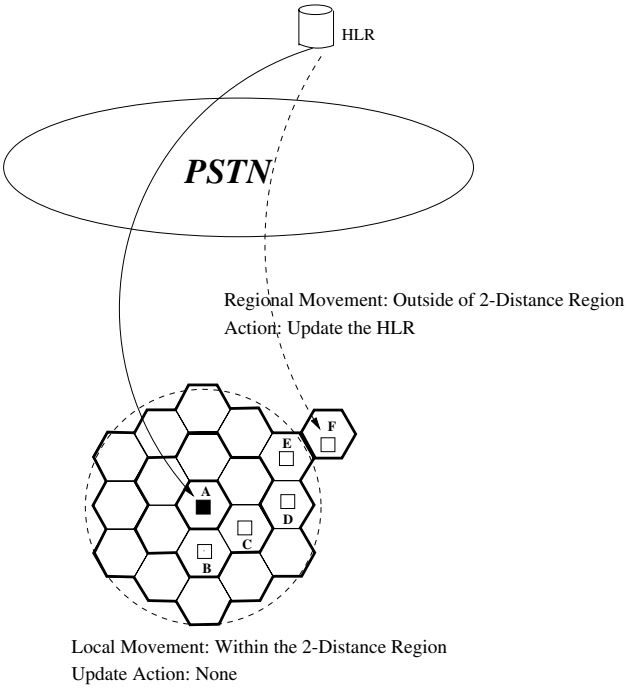

FIGURE 4. Update operations performed under PLA.

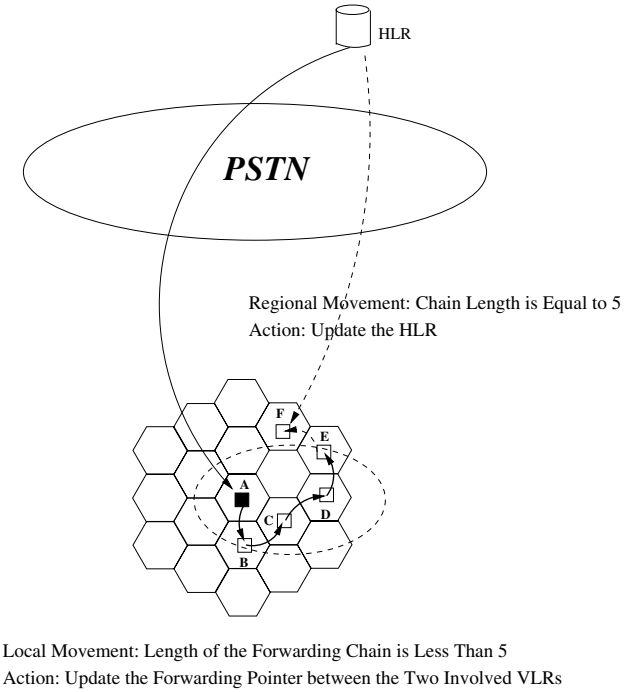

FIGURE 5. Location updates performed under FRA.

ring 0, VLRs B and C are in ring 1 and VLRs D and E are in ring 2. When the mobile user makes a local movement with a distance away from A of less than $n = 3$, e.g. to B, C, D or E, no location update operation is performed and the agent remains unchanged. However, when the mobile user makes a regional move, e.g. to VLR F, the distance from A (the agent) to F is now equal to $n = 3$. An update operation must be performed by the HLR in this case, after which VLR F becomes the new agent.

3.3. FRA

The advantage of forwarding over non-forwarding mechanisms in the mobile network was discussed in [14]. Basically, instead of informing the HLR every time the mobile user crosses a VLR boundary, only a forwarding pointer is set up between involved VLRs. All the time the HLR only points to the VLR at the beginning of the forwarding chain, say V_0, which points to V_1, V_2 and so on. As the length of the chain grows, the search cost increases since a long chain has to be followed to locate the mobile users. As a result, a periodic reset operation must be performed to balance the update and search costs. One way to view forwarding algorithms is to treat the area bounded within a length of k as a local region as shown in Figure 5. In this view, as long as the mobile user moves within the local region, there is no need to inform the HLR and only a pointer set up between two VLRs is necessary. When the mobile user makes a regional movement, i.e. when the length of the forwarding chain is k (a parameter to be determined), a reset operation is then performed to inform the HLR of the new V_0. In this sense, V_0 is being replaced dynamically and behaves like the agent of the mobile user.

In Figure 5, assume that VLR A is at the beginning of the forwarding chain, thus behaving as the local agent of the mobile user. When the mobile user makes a local movement so that the length of the forwarding chain is still smaller than a predefined value k, only forwarding pointers are set up. In Figure 5, A–B–C–D–E is the forwarding chain with a length of 4. Suppose $k = 5$ in Figure 5. Then, when the mobile user goes from VLR E to VLR F, a regional movement occurs, thus triggering a reset operation to be performed by the HLR, after which VLR F becomes the local agent. An interesting research problem in the FRA scheme is how often the reset operation (which defines the size of a local region) should be performed, and we have addressed this in a previous work [20].

3.4. LAA

In [17], Ho and Akyildiz proposed a scheme called LAing. The basic idea is that location registration operations should be as localized as possible so as to reduce the number of registration messages to the HLR. The VLR which performs the last registration operation with the HLR is called the LA of the mobile user. There is one LA per region where the size of the region is a parameter to be determined. Continuing with the hexagonal network coverage model (Figure 3), the number of VLRs covered by the LA in the local region under LAA is $3n^2 - 3n + 1$, where n is a design parameter, e.g. $n = 2$ means that seven VLRs are covered by a LA in the local region and $n = 3$ means that 19 VLRs are covered instead. Note that when $n = 1$, only one VLR is covered. When the mobile user crosses a VLR boundary but is within the local region, the new VLR only informs the LA without informing the HLR; when the mobile user makes a regional

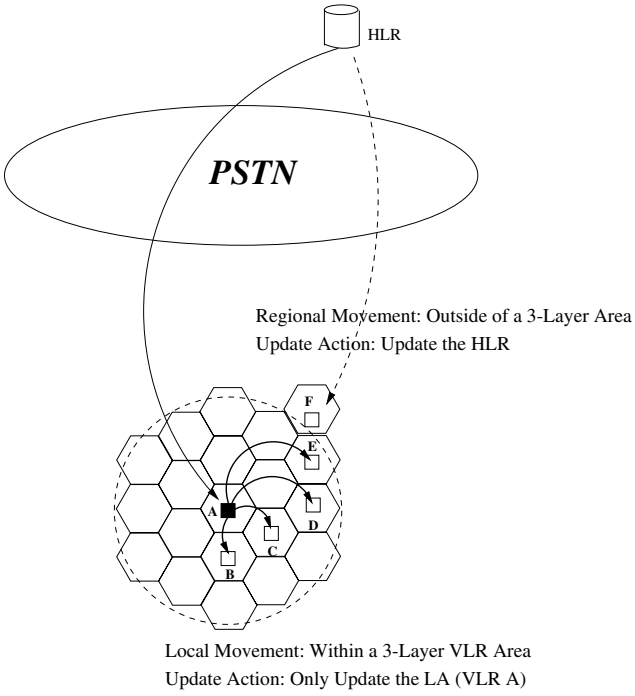

FIGURE 6. Location updates performed under LAA.

move, the new VLR informs the HLR and also becomes the new LA of the mobile user. At all times the HLR only knows the address of the LA.

In Figure 6, assume that $n = 3$ at which a network switch covers 19 VLRs; VLR A is the LA initially. When the mobile user makes a local movement from VLR A to VLR B, a location update operation to the LA is performed, but not to the HLR. Similarly, when the mobile user subsequently moves to VLR C, the LA's database is updated to point to VLR C. If there is a call looking for the mobile user, the search path starts from the HLR, then VLR A (the LA) and finally reaches VLR C. The LA is changed only when the mobile user makes a regional movement, in which case a location update operation to the HLR is required. In Figure 6, this happens when the mobile user moves to VLR F, which is outside of the 19 VLRs covered by VLR A. Note the basic difference between LAA and PLA: in LAA, an update operation is performed to the LA upon a local movement. In PLA, a local movement does not trigger any location update operation to the agent and thus does not incur any update cost at all.

4. MODELING

In this section, we develop analytical models to describe the behavior of a mobile user under various degradable location management algorithms. Our intent is to assess and compare degradable location management algorithms and to be able to classify them based on the location update cost required. Our approach is based on hierarchical modeling. At the high level comes a cost model for defining the cost of the PCS network in servicing 'location update' and 'location search' operations for a mobile user. At the low level comes a Markov model for parameterizing the cost model defined at the high level. Due to the use of Markov models at the low level, we implicitly assume that all times are exponentially distributed. We note that this assumption may not be justified for the residence time in a location area [8], but it can be relaxed by using Markov regenerative stochastic Petri net (MRSPN) models instead [21] at the low level in which times are generally distributed. The modeling approaches proposed in this paper in defining Markov models and in calculating location update and search costs can then be similarly applied.

4.1. High-level model

The high-level model adopts a cost model proposed in [22] and includes two components: (a) update cost, the cost of updating the location of the mobile user due to user movements; and (b) query cost, the cost of searching for the user in response to a call. For a location management scheme X, let X_{update} be the *average* cost of the PCS network to service a location update operation due to a user movement crossing VLR registration boundaries. Note that, for some location management algorithms, a user movement may not cause any update cost at all, e.g. a local movement under the PLA scheme causes zero update cost. Therefore, X_{update} here stands for the 'average' cost of a user movement over the lifetime of the mobile user, covering both the local and regional moves. Similarly, let X_{search} be the *average* cost to locate the mobile user in a location search operation. Furthermore, let X_{cost} be the average cost of the PCS network to service the above two types of operations between two consecutive calls. Then,

$$X_{\text{cost}} = X_{\text{update}} \times \sigma/\lambda + X_{\text{search}} \tag{1}$$

where σ is the rate at which the mobile user moves across VLR boundaries and λ is the rate at which the mobile user is being called, as defined in Table 2.

Equation (1) is obtained because between two consecutive calls, the number of mobility moves across VLR registration boundaries by the mobile user is equal to σ/λ on average. One can imagine that a mobile user moves across registration boundaries a number of times (σ/λ on average) before receiving a call and then the same pattern repeats again. The X_{cost} parameter above gives the total cost incurred by the PCS network in each such repeated period accounting for both the location update and search costs, thus providing a uniform cost measure so as to be able to fairly compare all location management schemes.

4.2. Low-level model

In this section, we develop three separate Markov models for PLA, FRA and LAA, respectively. The objective is to parameterize Equation (1). We first separate the model parameters into two classes. Tables 2 and 3 show these two classes of parameters, respectively. Note that the *per-mobile-user* parameter class is inherently associated with a mobile user, while the *network* parameter class

TABLE 2. Per-mobile-user parameters.

σ	the rate at which the mobile user moves across VLR boundaries
λ	the rate at which the mobile user is being called
CMR	λ/σ, the call-to-mobility ratio of the mobile user

TABLE 3. Mobile network parameters.

T	the average VLR–HLR round-trip communication cost
τ	the average VLR–VLR round-trip communication cost

depends on the mobile network structure. To make our presentation concrete, we consider the hexagonal network coverage model as shown in Figure 3. Note that T and τ defined in Table 3 represent the *average* communication costs. Their values can be calculated by means of a network coverage model (e.g. hexagonal) characterizing the underlying wireless network as in [20].

4.2.1. Modeling a PCS network operating under PLA

For notational convenience, we introduce the following additional parameters as we model PLA. Note that σ_i, β_i, μ_r and μ_i can each be expressed as a function of the per-mobile-user and network parameters discussed earlier. We will explain how these functions are obtained at a later time.

- n: the n parameter used by PLA to specify the $(n-1)$-distance region within which a user move causes no update cost.
- σ_i: the mobility rate of the mobile user moving from ring i to ring $i+1$.
- β_i: the mobility rate of the mobile user moving from ring $i+1$ to ring i.
- μ_r: the execution rate to perform a location update to the HLR.
- μ_i: the execution rate to locate the mobile user currently located in ring i.
- $P_{(i,j)}$: the probability that the system is in a particular state in equilibrium.

For a hexagonal model as shown in Figure 3, it can be shown [16] that the probability of the mobile user moving from ring i to ring $i+1$, $i \geq 0$, when a random move has been made by the mobile user, is given by

$$\begin{cases} \dfrac{2i+1}{6i}, & \text{if } i \geq 1, \\ 1, & \text{if } i = 0. \end{cases}$$

The special case is that the probability is 1 when moving from ring 0 (containing only the agent itself) to ring 1. Similarly, the probability of moving from ring $i+1$ to ring i, $i \geq 0$, given that a random move had been made by the mobile user, can be derived as

$$\frac{2(i+1)-1}{6(i+1)}.$$

Hence, the mobility rate of the mobile user moving from ring i to ring $i+1$, σ_i, is given by

$$\sigma_i = \begin{cases} \dfrac{(2i+1)\sigma}{6i}, & \text{if } i \geq 1, \\ \sigma, & \text{if } i = 0, \end{cases} \tag{2}$$

and the mobility rate of the mobile user moving from ring $i+1$ to ring i, β_i, is given by

$$\beta_i = \frac{(2i+1)\sigma}{6(i+1)}, \quad i \geq 0. \tag{3}$$

Furthermore, since it takes an average of T time to do a round-trip VLR–HLR communication, the execution rate to perform a location update from a new VLR agent to the HLR, μ_r, can be parameterized as

$$\mu_r = \frac{1}{T}. \tag{4}$$

Note that the above includes a one-way communication cost from the new VLR agent to the HLR to update the HLR's database and another one-way communication time from the HLR to the new agent to acknowledge the request. The update action is triggered by the new VLR agent outside of the $(n-1)$-distance region.

The time it takes to locate the mobile user located in ring i includes the following communication costs: (a) from the HLR to the agent; (b) from the agent (in ring 0) to the current VLR (in ring i); and finally (c) from the current VLR back to the HLR. The last step also updates the HLR's database such that the current VLR becomes the new agent. This is so because the search event is triggered by the HLR which expects to receive the user's location information from the current VLR. Hence, the execution rate to locate the mobile user in ring i, μ_i, can be parameterized as

$$\mu_i = \frac{1}{T + \frac{1}{2}(3i^2+3i)\tau}. \tag{5}$$

Here, the second term in the denominator accounts for the fact that on average the agent will have to query one-half of the VLRs in the i-distance region to find the current VLR in ring i.

Figure 7 shows a low-level Markov model for describing the behavior of a mobile user under PLA. Here a state is represented by (a, b) where a is either 0 (standing for IDLE) or 1 (standing for CALLED), and b indicates the current distance between the mobile user and the local agent. Of course, $0 \leq b \leq n-1$, where n is the distance value. Initially, the mobile user is in state $(0, 0)$; meaning that it is not being called and the mobile user resides in the area of the current VLR agent. In the following, we explain briefly how we construct the Markov model.

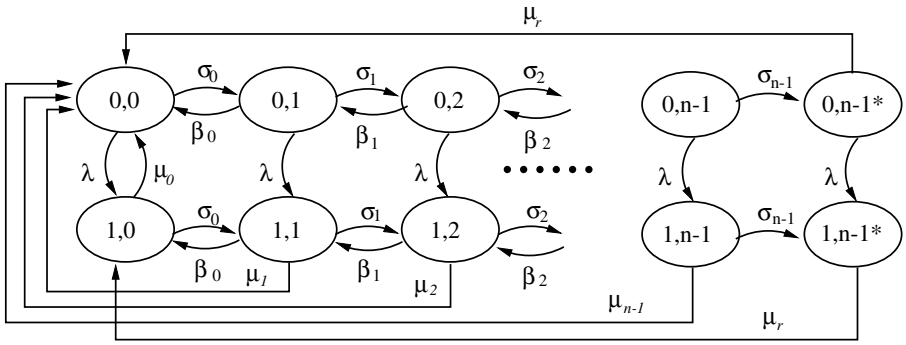

FIGURE 7. The Markov model for the PCS network under PLA.

(1) If the mobile user is in state $(0, i)$ and a call arrives, then the new state is $(1, i)$, i.e. the mobile user is now in the state of being called. This behavior is modeled by the (downward) transition from state $(0, i)$ to state $(1, i)$, $0 \leq i \leq n-1$, with a transition rate of λ.

(2) If the mobile user is in state $(1, i)$ and another call arrives, then the mobile user will remain in the same state, since the mobile user remains in the state of being called. This behavior is described by a hidden transition from state $(1, i)$ back to itself with a transition rate of λ. This self-transition is not shown in the model since it does not need to be considered as we solve the Markov model for the steady state probability $P_{(i,j)}$.

(3) If the mobile user is in ring i while it is being called, the network serves the call with a service rate μ_i whose magnitude depends on i. This is modeled by a transition from state $(1, i)$ to $(0, 0)$ with rate μ_i. Note that, in the above transition, the new state is $(0, 0)$ because all calls can be serviced at once and after the HLR is informed of the location update the new VLR which the mobile user currently resides under becomes the new agent. The parameter μ_i must account for the cost involved in the above action.

(4) Regardless of whether the mobile user is in the state of being called or not, if the mobile user moves from ring j to ring $j+1$, $0 \leq j \leq n-2$, the distance between the mobile user and the current local agent is increased by 1. This is described by a transition from state (i, j) to $(i, j+1)$ with rate σ_j. This also describes the behavior of the user in migrating from inner rings to outer rings.

(5) If the mobile user moves from an outer ring $j+1$ to an inner ring j, the distance between the mobile user and the current local agent is reduced by 1. This is described by a transition from state (i, j) to state $(i, j-1)$ with rate β_{j-1}, $1 \leq j \leq n-1$. This also describes the behavior of the user in migrating from outer rings to inner rings.

(6) If the mobile user is in state $(i, n-1)$, where n is the distance value, and it makes a move to a new outer ring n, then a regional movement occurs and a reset operation must be invoked to update the new local agent to the HLR. This behavior is described first by a transition from $(i, n-1)$ to $(i, n-1)^*$ with rate σ_{n-1}, after which a transition occurs from $(i, n-1)^*$ to $(i, 0)$ with rate μ_r, representing the time it takes for the PCS network to execute the reset operation and update the HLR with the new agent information.

If the call arrival rate λ is much higher than the mobility rate σ, then the probability that the system is found to stay in state $(1, i)$ would be much greater than in state $(0, i)$. Let pla_{update} denote the average cost of the PCS network for executing a location update operation and let pla_{search} denote the average cost for executing a location query operation. Then,

$$pla_{\text{update}} = \left(\sum_{i=0}^{1} (P_{(i,n-1)} + P_{(i,n-1)^*}) \times (1/\mu_r) \right) \quad (6)$$

$$pla_{\text{search}} = \left(\sum_{i=0}^{1} \sum_{j=0}^{n-1} P_{(i,j)} \times (1/\mu_j) \right) + (P_{(0,n-1)^*} + P_{(1,n-1)^*}) \times (1/\mu_0) \quad (7)$$

where $P_{(i,j)}$ is the percentage of time the system is found to be staying at state (i, j) in equilibrium.

Therefore, based on Equation (1),

$$pla_{\text{cost}} = pla_{\text{update}} \times \sigma/\lambda + pla_{\text{search}}. \quad (8)$$

Note that the right-hand side expression in Equation (6) can be used to classify PLA(n) (as shown in Table 1) for a given set of per-mobile-user and network parameter values.

4.2.2. Modeling a PCS network operating under FRA

In [20], we developed a Markov model to describe the behavior of the PCS network under the FRA scheme. In addition to analyzing the performance characteristics of the system under FRA, we also used the Markov chain to determine the best time to reset the forwarding chain. In this paper, we develop a modified Markov model to account for the following two additional behaviors in order to compare FRA with other algorithms fairly: (a) the forwarding chain will be reset after a location query operation is performed; (b) when the mobile user moves back to the previously visited VLR in the chain, i.e. from V_i to V_{i-1}, the length of

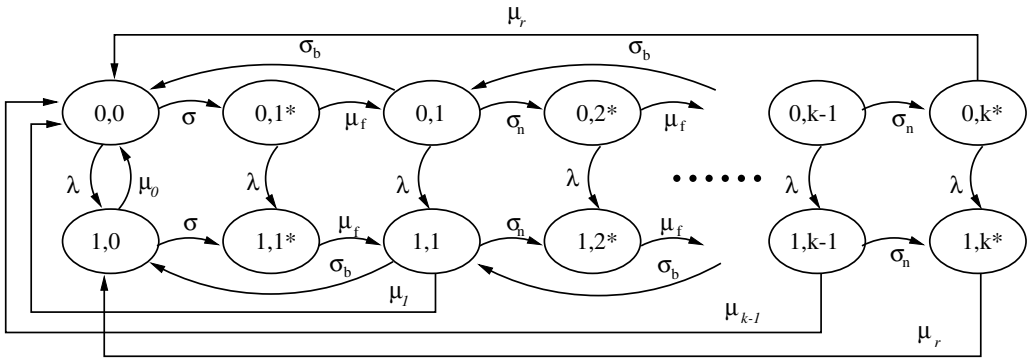

FIGURE 8. The Markov model for the PCS network under FRA.

the forwarding chain is reduced by 1 and no pointer deletion operation is required between V_i and V_{i-1}. The second point is based on the assumption that obsolete pointers will be purged automatically after a period of time much greater than the average reset period. The modified Markov model will account for the looping behavior involving two neighbor VLRs in the forwarding chain. Note that in the above scenario if the mobile user subsequently moves from V_{i-1} to V_i again, a pointer set-up operation is still required. As in [20], we assume that when the mobile user moves across a VLR, the forwarding pointer information will be updated before it crosses another VLR. This assumption has the implication that the dwell time of the mobile user in a VLR is much larger than the forwarding pointer update operation time.

For FRA, we also introduce a set of parameters as follows to facilitate discussion.

- k: the length of the forwarding chain at which a reset operation is performed.
- σ_n : the mobility rate of the mobile user moving to a new VLR.
- σ_b : the mobility rate of the mobile user moving to the previous VLR.
- μ_r : the execution rate to reset a forwarding chain, i.e. to update the HLR.
- μ_f : the execution rate to set-up or traverse a pointer between two VLRs.
- $\mu_i : 0 \leq i \leq k-1$, the execution rate to locate the mobile when the length of forwarding pointer is i.

Again, based on the hexagonal structure and random movement, σ_n and σ_b can be parameterized as follows:

$$\sigma_n = \tfrac{5}{6}\sigma; \quad \sigma_b = \tfrac{1}{6}\sigma.$$

The time to update the HLR from a VLR is T. Hence,

$$\mu_r = \frac{1}{T}.$$

Similarly, the time to set up a pointer between two VLRs is τ. Hence,

$$\mu_f = \frac{1}{\tau}.$$

Finally, the time to locate the mobile user when the length of the forwarding chain is i includes the time from the HLR to the first VLR on the chain, the time to traverse the chain from the first VLR to the ith VLR and finally the time to go from the ith VLR to the HLR. Hence,

$$\mu_i = \frac{1}{T + i\tau}.$$

Figure 8 shows a Markov model describing the behavior of a mobile user wherein a state is represented by (s_1, s_2) where s_1 is either 0 (standing for IDLE) or 1 (standing for CALLED), while the other component s_2 indicates the current length of the forwarding chain. Initially, the mobile user is in state $(0, 0)$, meaning that it is not being called and the number of forwarding steps is zero. A symbol '*' is put in a state if the mobile user just enters a new VLR but the forwarding pointer operation is not yet performed. For example, in state $(0, 1)^*$ the mobile user has just crossed V_1 from V_0 but the forwarding operation between V_0 and V_1 is not yet performed. Of course, after the forwarding pointer operation is performed, the length of the forwarding chain will be 1 in this case.

We briefly discuss the meaning of the Markov chain as follows. First, if the mobile user is in state $(0, i)$ or $(0, i)^*$ and a call arrives, then the new state is $(1, i)$ or $(1, i)^*$ in which the number of forwarding steps remains at i but the mobile user is now in the state of being called. This behavior is modeled by the (downward) transition from state $(0, i)$ to state $(1, i)$ or from state $(0, i)^*$ to state $(1, i)^*$, $0 \leq i \leq k$, with a transition rate of λ. Second, if the mobile user is in state $(1, i)$ or $(1, i)^*$ and another call arrives, then the mobile user will remain in the same state. Third, if the mobile user is in state $(1, i)$, the signaling network can service all pending calls simultaneously with a service rate of μ_i. After the service, the new state is $(0, 0)$ since all calls are serviced and the reset operation has also been performed. This behavior is described by the state transition from state $(1, i)$ to state $(0, 0)$ with a transition rate of μ_i. Note that this rate depends on the length of the forwarding chain. Fourth, if the mobile user moves back to the previously visited VLR on the forwarding chain, a transition from state (i, j) to state $(i, j-1)$ will take place, where $1 \leq j \leq k-1$. Note that obsolete pointers will be deleted implicitly, so there is

no need to take time to perform the pointer delete operation. Lastly, regardless of whether the mobile user is in the state of being idle or having been called, if the mobile user moves across a new VLR boundary, a pointer connection or a reset operation must be performed in response to the move event. This behavior is first modeled by a transition from state (i, j) to state $(i, j+1)^*$ with a mobility rate of σ_n where $0 \leq i \leq 1$ and $0 \leq j \leq k-1$, after which one of the following two cases may occur.

(1) If $0 \leq j \leq k-2$, then the new VLR simply sets up a forwarding pointer connection. This behavior is modeled by a transition from state $(i, j+1)^*$ to state $(i, j+1)$ with rate μ_f.
(2) If $j = k-1$, however, then the length of the forwarding chain has reached k, so the new VLR must perform a reset operation. This latter behavior is modeled by a transition from state $(i, k)^*$ to state $(i, 0)$ with rate μ_r.

Now let $P_{(i,j)}$ represent the probability that the system is found to be staying at state (i, j) in equilibrium. Let fra_{update} be the average cost to perform a location update operation. Let fra_{search} be the average cost to perform a location search operation. Then,

$$\begin{aligned} fra_{\text{update}} = {} & (P_{(0,k-1)} + P_{(1,k-1)} + P_{(0,k)^*} + P_{(1,k)^*}) \\ & \times (1/\mu_r) \\ & + \left(\sum_{i=0}^{k-2} (P_{(0,i)} + P_{(1,i)}) + \sum_{i=1}^{k-1} (P_{(0,i)^*} + P_{(1,i)^*}) \right) \\ & \times (1/\mu_f) \end{aligned} \tag{9}$$

$$fra_{\text{search}} = \left(\sum_{i=0}^{k-1} (P_{(0,i)} + P_{(1,i)} + P^*_{(0,i)} + P_{(1,i)^*}) \times (1/\mu_i) \right). \tag{10}$$

Hence, based on Equation (1),

$$fra_{\text{cost}} = fra_{\text{update}} \times \sigma/\lambda + fra_{\text{search}}. \tag{11}$$

Equation (11) above yields the average cost of the signaling network as a function of k. For a given set of parameter values, we can first compute the values of $P_{(i,j)}$ for all states and then use Equation (11) to compute the average cost. The optimal k value is the one that minimizes the cost measure defined in Equation (11). Once the optimal k value is determined, it can be used to compute fra_{update} so as to determine where the optimal FRA algorithm will fall in the spectrum of degradable location management algorithms for a given set of per-mobile-user and network conditions.

4.2.3. Modeling a PCS network operating under LAA

Under the LAA scheme, if the mobile user makes a move under the same network switch, i.e. a local movement, then the new VLR only informs the local agent without updating the HLR. However, if the mobile user crosses a network switch boundary, a registration operation must be initiated to update the HLR and the new VLR will become the local agent.

We introduce some more parameters below to ease our discussion.

- n: the n parameter to specify the n-layer VLR region which covers $3n^2 - 3n + 1$ VLRs.
- P_1: the probability that when the mobile user moves it remains under the same network switch.
- σ_1: the mobility rate of the mobile user moving under the same network switch, i.e. $\sigma_1 = P_1\sigma$.
- σ_r: the mobility rate of the mobile user crossing a network switch boundary, i.e. $\sigma_r = (1 - P_1)\sigma$.
- μ_g: the search execution rate when the mobile user is located in the agent's area.
- μ_a: the search execution rate when the mobile user is not located in the agent's area.
- δ_1: the execution rate to update the agent, i.e. to set up a pointer between the new VLR and the agent.
- δ: the execution rate to update the HLR.

Here we also consider a hexagonal network coverage model, in which each network switch covers an n-layer VLR region, where n can be either 2 or 3. In this case, it can be found easily (see [20]) that the probability of the mobile user staying under the same switch when making a move is equal to

$$P_1 = \frac{3n^2 - 5n + 2}{3n^2 - 3n + 1}.$$

Hence,

$$\sigma_1 = \frac{3n^2 - 5n + 2}{3n^2 - 3n + 1}\sigma$$

and

$$\sigma_r = \frac{2n - 1}{3n^2 - 3n + 1}\sigma.$$

When the mobile user is under the area directly covered by the agent, the average time needed to find the mobile user is the VLR–HLR round-trip communication time. Hence,

$$\mu_g = \frac{1}{T}.$$

Otherwise, the agent must also communicate with the mobile user's current VLR via a pointer to locate the mobile user. Thus,

$$\mu_a = \frac{1}{T + \tau}.$$

The average time to update the agent from a new VLR within the same switch is the VLR–VLR round-trip communication time. Hence,

$$\delta_1 = \frac{1}{\tau}.$$

On the other hand, the average time to update the HLR from a new VLR when a new switch area is entered is the VLR–HLR round-trip communication time, Hence,

$$\delta = \frac{1}{T}.$$

Figure 9 shows a Markov model describing the behavior of a mobile user with the state representation (a, b, c).

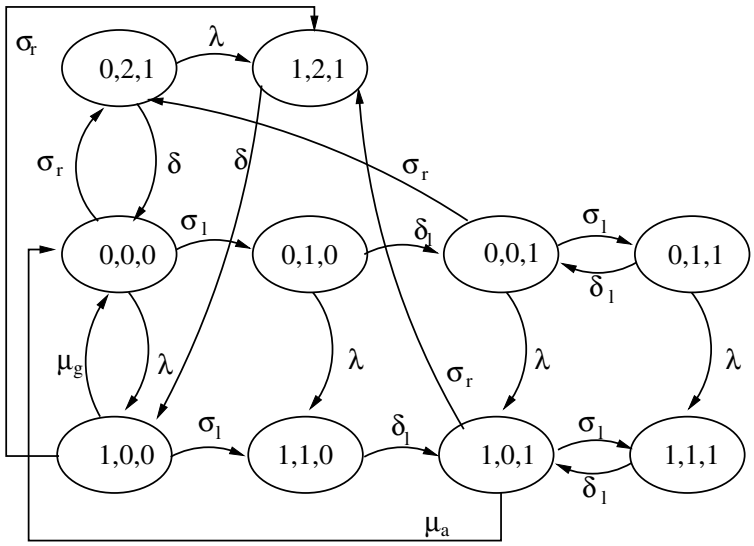

FIGURE 9. The Markov model for the PCS network under LAA.

The first component a indicates whether the mobile user is in the state of being called, with 0 standing for idle and 1 standing for busy. The second component b indicates whether the mobile user makes a move, with 0 meaning that it does not, 1 meaning that it just makes a local move and 2 meaning that it just makes a regional move. The third component indicates whether the agent currently covers the mobile user, with 0 meaning yes and 1 meaning no. Initially, the mobile user is in state $(0, 0, 0)$, meaning that the mobile user is not being called, has not yet made any move and is located in the agent's VLR area. Below, we explain briefly how we construct the Markov model.

(1) First, if the mobile user is in state $(0, i, j)$ and a call arrives, the new state will be $(1, i, j)$, i.e. the mobile user is now in the state of being called. This behavior is modeled by the transition from state $(0, i, j)$ to state $(1, i, j)$, $0 \leq i \leq 1$, or from state $(0, 2, 1)$ to state $(1, 2, 1)$, with a transition rate of λ.
(2) Calls will be serviced with a rate of μ_g when the system is in state $(1, 0, 0)$ since in this case the HLR database points to the agent which directly covers the mobile user. This is modeled by a transition from state $(1, 0, 0)$ to $(0, 0, 0)$ with a transition rate of μ_g. On the other hand, calls will be serviced with a rate of μ_a when the system is in state $(1, 0, 1)$ since in this case the HLR database points to the agent which does not cover the mobile user; therefore, we must follow the pointer stored at the agent's database to locate the VLR that currently covers the mobile user, after which the VLR also becomes the new agent. This latter case is modeled by a transition from state $(1, 0, 1)$ to $(0, 0, 0)$ with a transition rate of μ_a.
(3) Regardless of whether the mobile user is in the state of being called or not, if the mobile user moves across a VLR boundary, a location update operation will be performed either to the agent or to the HLR, depending on whether a network switch is crossed. We therefore distinguish the following two cases.

(a) If the mobile user has moved across a network switch, a reset operation must be performed to update the HLR. This behavior is modeled first by a transition from state $(i, 0, j)$ to state $(i, 2, 1)$, $0 \leq i, j \leq 1$, with a transition rate σ_r, after which a transition occurs from $(i, 2, 1)$ to $(i, 0, 0)$, $0 \leq i \leq 1$, with a transition rate δ to account for the HLR update time. Here, after an update is done to the HLR, the HLR database points to the new agent which now covers the mobile user who just enters into the new agent's VLR area.
(b) If the mobile user just makes a local move, then only a pointer set up between the local agent and the new VLR is required. This behavior is modeled first by a transition from state $(i, 0, j)$ to state $(i, 1, j)$, $0 \leq i, j \leq 1$, with a transition rate σ_1, after which a transition occurs from $(i, 1, j)$ to $(i, 0, 1)$, $0 \leq i, j \leq 1$, with an execution rate δ_1 to account for the update time to the agent.

Now, from the Markov chain,

$$laa_{\text{update}} = \left(\sum_{i=0}^{1} \sum_{j=0}^{1} P_{(i,0,j)} \times [P_1 \times (1/\delta_1) + (1 - P_1) \times (1/\delta)] \right) + \left(\sum_{i=0}^{1} \sum_{j=0}^{1} P_{(i,1,j)} \times (1/\delta_1) \right) + (P_{(0,2,1)} + P_{(1,2,1)}) \times (1/\delta) \quad (12)$$

$$laa_{\text{search}} = \left(\sum_{i=0}^{1} (P_{(i,0,0)} + P_{(i,2,1)}) \times (1/\mu_g) \right) + \left(\sum_{i=0}^{1} (P_{(i,1,0)} + P_{(i,0,1)} + P_{(i,1,1)}) \times (1/\mu_a) \right). \quad (13)$$

Therefore, based on Equation (1),

$$laa_{\text{cost}} = laa_{\text{update}} \times \sigma/\lambda + laa_{\text{search}}. \quad (14)$$

Again, Equation (12) above can be used to classify LAA for a given set of per-mobile-user and network conditions.

5. ANALYSIS AND COMPARISON

All the data reported here were obtained by solving the Markov models using the SHARPE software package [23] to obtain $P_{(i,j)}$'s for all states and then computing the location update, search or total cost based on the equations derived in Section 4. We report a case in which the ratio of the VLR-to-VLR cost to the VLR-to-HLR cost is equal to 0.3, i.e. $T = 1$ and $\tau = 0.3$. This selection reflects a reasonable ratio between T and τ. The exact ratio of T to τ depends on the wireless network employed and can be computed using the approach described in [20] by means of a network coverage model. Here we report only the results for this case since the main objective of the paper is to demonstrate how our two-level hierarchical modeling method can be effectively used to compare degradable location management algorithms when given a set of per-mobile-user and network parameters that characterize a wireless network environment. For completeness, we will first show the individual performance characteristics of these algorithms compared with the basic HLR/VLR IS-41 algorithm. Then we will compare these algorithms head to head under identical per-mobile-user and network conditions.

The average cost of the PCS network for location management under IS-41 is given by

$$IS\text{-}41_{\text{cost}} = IS\text{-}41_{\text{update}} \times \sigma/\lambda + IS\text{-}41_{\text{search}} \quad (15)$$

where $IS\text{-}41_{\text{update}} = T$ and $IS\text{-}41_{\text{search}} = T$. Note that here we ignore the common overhead cost applicable to all algorithms, i.e. the initial cost incurred from the caller to the HLR and the cost of paging the mobile user from within the current VLR. Recall that IS-41 lies at one extreme of the spectrum, for which the average update cost per move is equal to T.

The average cost of PLA, FRA and LAA, as defined by Equations (8), (11) and (14), are shown in Figures 10, 11 and 12, respectively, along with that of IS-41 for comparison purposes. These figures show that for the case when $\tau = 0.3T$ and when the *CMR* is small, PLA, FRA and LAA can all significantly outperform IS-41. Note again that the cost measure defined in Equation (1) accounts for the network cost incurred due to location management in a *repeated* period between two consecutive calls, so even a cost difference of 0.1 T between two location management

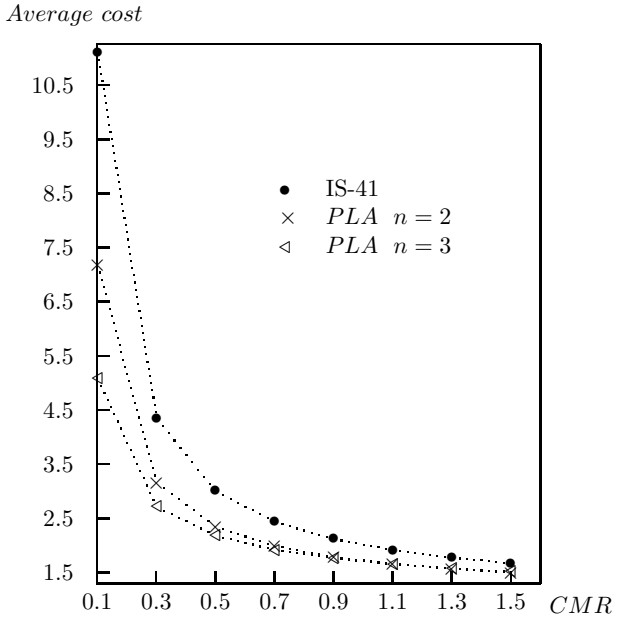

FIGURE 10. Comparison of PLA under different n-distance values.

schemes should be considered significant since the effect is cumulative.

Figure 10 shows a plot of the PCS network cost under PLA(n) for various n values. When the *CMR* value is small, the performance of PLA with $n = 3$ is better than that of PLA with $n = 2$. This behavior can be explained as follows. Recall that a larger n value means that a local agent can cover a larger area and thus there is a smaller probability for the mobile user to cross a regional boundary. Consequently, the number of update operations to the HLR is reduced as n increases. This is reflected in Figure 10 where we see that the performance of PLA with $n = 3$ is better than that of PLA with $n = 2$ at low *CMR* values where the cost of location updates dominates that of location queries. As the *CMR* value increases, however, the larger location query cost attributed by the larger cover area starts to dominate the reduced location update cost. Therefore, after the *CMR* value exceeds a threshold, PLA with $n = 2$ becomes better than PLA with $n = 3$.

Figure 11 demonstrates that FRA has a better performance with a long forwarding chain when *CMR* is small, again due to the fact that at a low *CMR* value the location update cost dominates the location query cost, so a longer chain is favored at low *CMR* since it reduces the location update cost. Again, as *CMR* increases, the higher cost associated with location search operations which happen frequently starts to offset the benefit of lower cost associated with location update operations which are not as frequent. In general, for any combination of mobile user and network conditions, there exists an optimal k value for which the network cost is minimized. The model presented in this paper can be used for that purpose.

Figure 12 exhibits the same trend as that in Figure 10, that is, when the *CMR* is small, LAA with $n = 3$ is better than LAA with $n = 2$, but when the *CMR* is large, it is the other

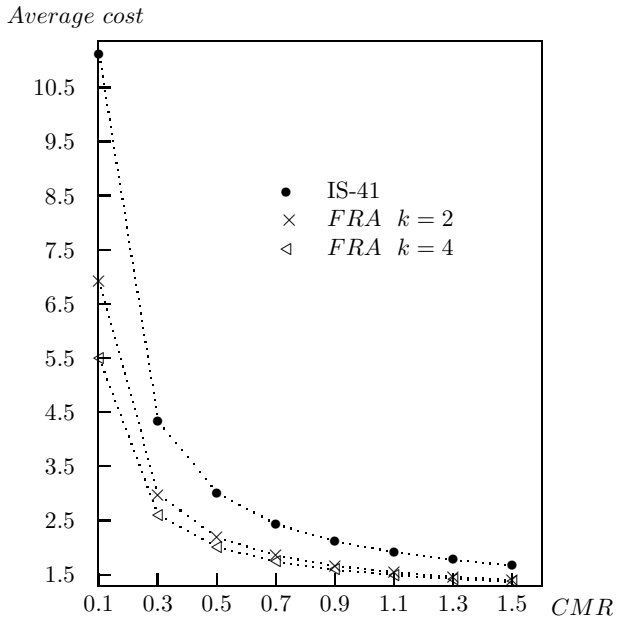

FIGURE 11. Comparison of FRA under different k values.

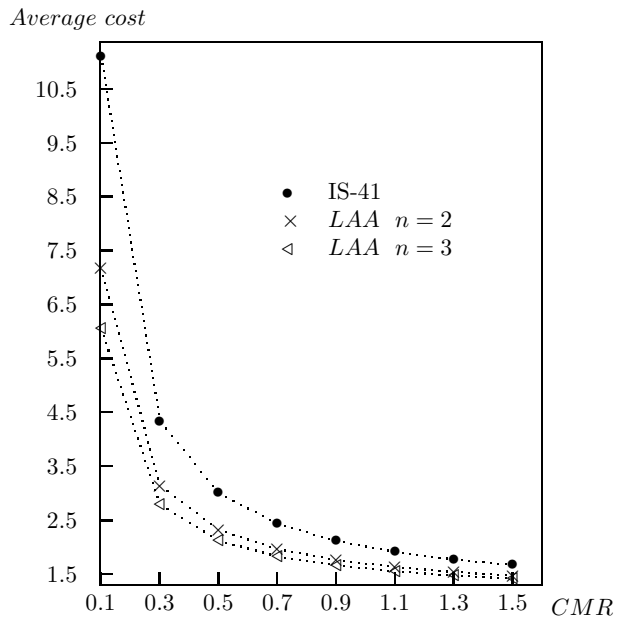

FIGURE 12. Comparison of LAA under different distance values.

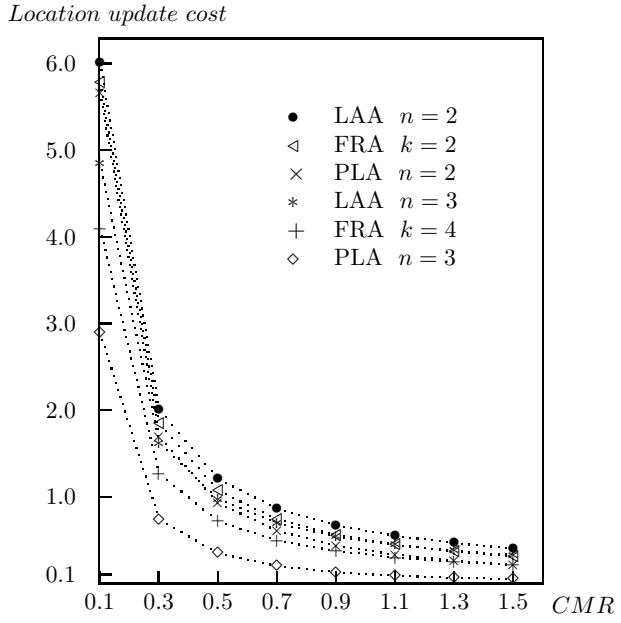

FIGURE 13. Comparison of the location update cost only.

way around. The same physical interpretation regarding the trade-off between the location update cost and the location query cost applies.

We compare these three algorithms head to head under the selected set of network conditions in Figures 13–15.

Figure 13 shows only the location update cost part, that is, the $X_{\text{update}} \times \sigma/\lambda$ part (or X_{update}/CMR) in Equation (1). The data shown in Table 1 earlier were generated from Figure 13 for the special case when $CMR = 0.1$, representing the update cost per move for an algorithm relative to IS-41 for a mobile user with $CMR = 0.1$. This figure provides us with a basis for classifying existing degradable location management algorithms based on the update cost per move relative to IS-41 for a wide range of *CMR* values. Specifically, the IS-41 HLR/VLR algorithm can be considered as an algorithm which keeps the system in the strong state all the time. LAA(2) is the one next to it among the six algorithms listed in terms of maintaining the location information in a good state. From Figure 13, we see that PLA(3) incurs the least amount of update overhead among the six algorithms listed, since under PLA(3) it is likely that the mobile user makes only local movements because the size of the local region is large. Consequently, it hardly does any updating at all. We therefore expect that PLA(3) will have to spend more time searching for the mobile user when a call arrives. This is confirmed in Figure 14 which displays only the location query cost part, that is, the X_{search} part in Equation (1), in which it shows indeed that PLA(3) incurs the most overhead to deliver a call. Figures 13 and 14 thus clearly demonstrate the trade-off between the location update and search operations.

We also observe that the increasing order in location search cost under these degradable location management algorithms shown in Figure 14 is not exactly the same as the decreasing order in location update cost shown in Figure 13, suggesting that certain algorithms can actually perform better under the same network conditions. Figure 15 compares all the six algorithms listed by combining both parts of the network cost, i.e. based on Equation (1). For the six algorithms listed under this selected network condition, we clearly see that PLA(3) is the best when *CMR* is 0.1, FRA(4) is the best when *CMR* is larger than 0.3, PLA(2) is the worst when *CMR* is small and lastly PLA(3) is the worst when *CMR* is large.

Finally, we examine the cumulative cost incurred due to multiple users. We compare the cases when all users

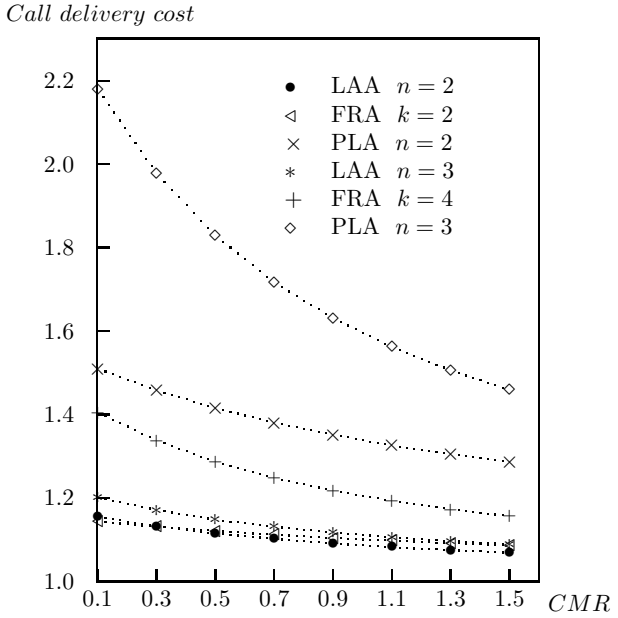

FIGURE 14. Comparison of the search cost only.

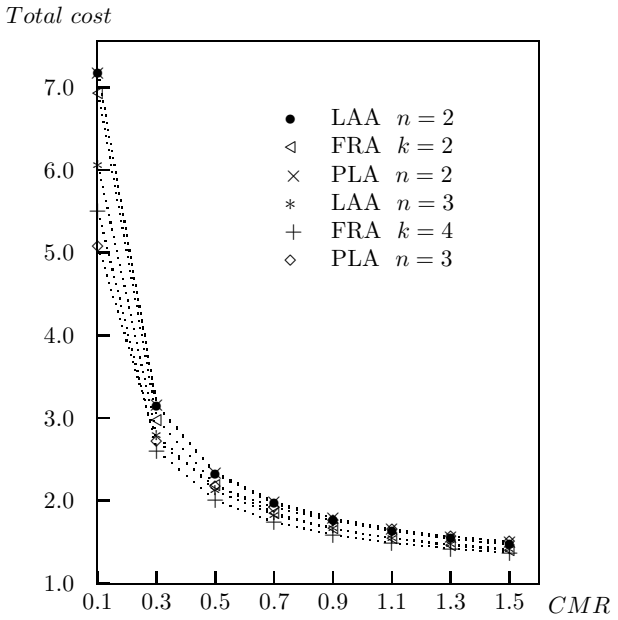

FIGURE 15. Comparison of the total communication cost for location management.

are served under a single algorithm against the case when individual users are served by their respective per-user best algorithms identified in the paper. Figure 16 shows the cumulative cost as a function of the number of mobile users. The curve labeled *Selective* stands for the per-user selective case. The *CMR* of each user is randomly generated in the range [0.1, 1.5]. The cumulative cost is obtained by simply summing all the costs incurred by individual users at their respective randomly generated *CMR* values under various algorithms. From Figure 16 we see that, as the number of users increases, the total cost difference increases because of the cumulative effect. If all users operate under a single

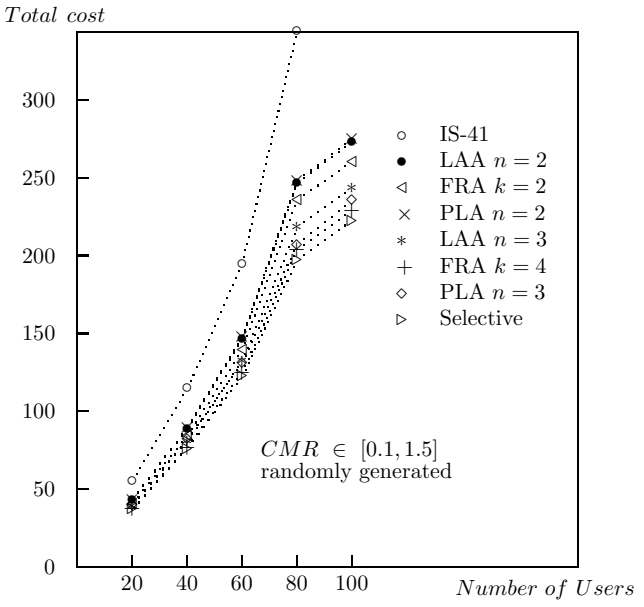

FIGURE 16. Comparison of the total communication cost for multiple users.

algorithm, FRA with $k = 4$ is the best among all single algorithms, followed by PLA with $n = 3$ and LAA with $n = 3$. All algorithms perform significantly better than the basic IS-41 algorithm for the multiple-user case. At the expense of increased software complexity and maintenance cost, if we allow the best algorithms to be applied on a per-user basis, Figure 16 shows that the per-user selective strategy can outperform all single-algorithm cases, the effect of which is more and more pronounced as the number of users increases.

6. SERVICE HANDOFF

In this section, we show how the methodology developed in the paper can be applied to the analysis of algorithms for handling service handoffs [24]. Service handoffs refer to the process of transferring or migrating the service of a client from one server to another in client–server applications. A service handoff is analogous to a location handoff but it occurs relatively infrequently. A major difference between location handoffs and service handoffs is that when a service handoff is performed the context information associated with the ongoing service needs to be transferred from one server (or a server proxy) to another or alternatively a pointer needs to be set up so that the new server can continue with the service. The context transfer includes both static context information, such as user profile and authentication data, as well as dynamic context information, such as files opened, objects updated, locks and timestamps.

Since service handoffs are mostly triggered by user movements, it has been suggested that the location management network be integrated with the service management network so that when a user moves into a new service area, a service handoff event can be detected and handled properly [24]. Thus, a service area can cover several VLRs.

Using the LAA scheme as an example, here we apply the methodology developed in the paper to analyze service handoffs. We assume that a service area corresponds to a network switch area. Thus when a user moves across a switch boundary, a service handoff is triggered. This will incur a context transfer communication cost C_s to transfer the context information from the old server to the new server. Suppose that the mobile user communicates with the server by means of operations. We are then interested in analyzing which LAA scheme ($n = 2$ or $n = 3$) is better in terms of lowering the average communication cost per operation when C_s and *CMR* are given. Assume that the location and service networks are integrated. The average communication cost per operation will be the sum of: (1) the communication cost between the server and the LA, the cost of which is τ; (2) the communication cost between the LA and the VLR in which the mobile user currently resides if the LA is not the current VLR, the cost of which is also τ; and (3) the communication cost of migrating the service context from the old server to the new server if, during the time of access, the mobile user happens to cross a service boundary and thus triggers a service handoff, the cost of which is C_s. We will call these three as the first, second and third cost factors, respectively. Note that the second and third cost factors may not be required. Recall that when modeling LAA in Figure 9, we used a state representation (a, b, c) with component a indicating whether the mobile user is in the state of being called (with 0 standing for idle and 1 standing for busy), component b indicating whether the mobile user makes a move (with 0 meaning that it does not, 1 meaning that it just makes a local move and 2 meaning that it just makes a move across a switch) and component c indicating whether the LA is the current VLR in which the mobile user resides (with 0 meaning yes and 1 meaning no). Therefore we can calculate the average cost per service operation by assigning: (1) a reward of $C_s + \tau$ to states in which component b is 2, i.e. (0, 2, 1) and (1, 2, 1); (2) a reward of τ to states in which component c is 0, i.e. (0, 0, 0), (1, 0, 0), (0, 1, 0) and (1, 1, 0); and (3) a reward of 2τ to states in which component c is 1 but component b is not equal to 2, i.e. states (0, 0, 1), (1, 0, 1), (0, 1, 1) and (1, 1, 1) in Figure 9.

Figure 17 shows the average communication cost per service operation in the LAA scheme as a function of C_s under various per-user *CMR* values. For ease of presentation, the cost is normalized with respect to $\tau = 1$. We see that when the context transfer cost C_s is not large relative to the average VLR–VLR communication cost, LAA with $n = 2$ can perform better than LAA with $n = 3$, especially at low *CMR* values. The reason behind this is that when *CMR* is low the probability of the mobile user crossing the switch boundary is low, so the contribution of the context transfer cost (the third cost factor) to the overall cost is low for both LAA schemes. Since there are more VLRs being covered by LAA with $n = 3$ (19 VLRs) than by LAA with $n = 2$ (seven VLRs), the contribution of the second cost factor is higher in LAA with $n = 3$ due to the fact that the probability of the LA not being the current

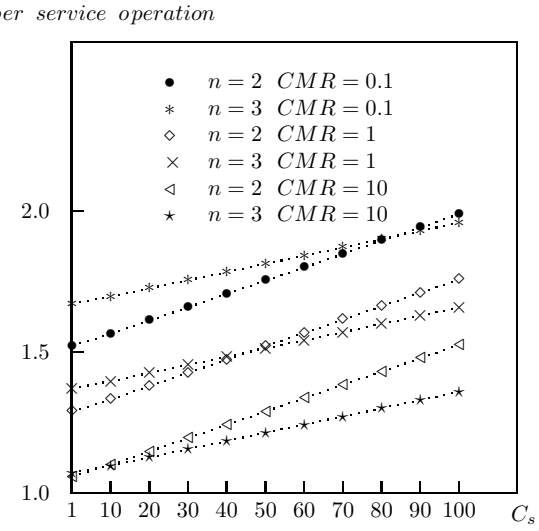

FIGURE 17. Cost per service operation in LAA under different *CMR* and C_s values.

VLR is higher in LAA with $n = 3$. From Figure 17 we also observe that there exists a crossover point after which LAA with $n = 2$ starts to yield a higher cost per service operation than LAA with $n = 3$ as C_s increases. Furthermore, the crossover point shifts towards the left as *CMR* increases. The reason for this is that as the context transfer cost increases relative to τ, LAA with $n = 3$ will be favored over LAA with $n = 2$ because it has a smaller probability of crossing switch boundaries and consequently a smaller context transfer overhead. Furthermore, as the *CMR* value increases and thus the probability of switch boundary crossing events decreases, the advantage of LAA with $n = 3$ over LAA with $n = 2$ becomes manifest even at a relatively small C_s value. The analysis performed here thus allows us to determine the coverage area of a server to minimize the average network communication cost per service operation, when values of *CMR* and C_s are given.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we discussed the notion of a degradable location management algorithm used in a PCS for locating users. We classified existing location management algorithms based on how well the user's location information is maintained. Methods were developed to quantitatively obtain the location update cost for location management algorithms, thus providing a basis for classifying them. We tested our method by modeling three existing algorithms: PLA, FRA and LAA. We demonstrated the applicability of our modeling method by classifying and comparing these algorithms under one chosen workload condition and revealed conditions under which one scheme can be superior.

The information obtainable via our proposed two-level hierarchical modeling method regarding how much location update cost the system is willing to invest so that a

user location query can be answered quickly is useful in classifying mobile users based on their QoS requirements. This will facilitate the deployment of per-class-based location management schemes which are much simpler to implement and more scalable than per-user-based schemes. It is also possible to design a dynamic location management algorithm (whose location update/search cost has been analyzed using the method proposed here) which can tune the system to a set of 'strong' and 'weak' states, depending on the network workload condition detected in real-time. In this scheme, when the network is under a heavy workload and there is a limited bandwidth for the mobile user to do update registration with the HLR, it can keep the PCS network operating under a 'weak' state to save the communication cost to reduce the network workload. When the network is under a lighter workload, it can keep the PCS under a 'strong' state to increase the service quality.

Finally, we also showed how the modeling methodology developed can be applied to determine the coverage area of a server in a replicated environment to support service handoffs in order to minimize the cost per service operation between a mobile client and its server. In mobile environments where location and service handoffs are tightly integrated, the analysis method developed in this paper can be used to assess the overall communication cost incurred due to both location and service managements.

Some future research areas related to this paper include: (a) introducing a real-time component into the design and deriving conditions under which user location queries can be satisfied in real-time while minimizing the location update cost; (b) considering users with different priority classes and discovering an optimal way to design location management algorithms so that a global design objective such as maximizing the system total reward can be best satisfied; and (c) investigating the applicability of the uniform framework developed in this paper to the analysis of tree-based location management algorithms.

REFERENCES

[1] EIA/TIA (1997) *Cellular Radio Telecommunication Intersystem Operations*. Technical Report TIA/EIA-41-D, Telecommunications Industry Association, Washington, DC.

[2] Mouly, M. and Pautet, M. B. (1992) *The GSM System for Mobile Communications*. Telecom Publishing, Olympia, WA.

[3] Akyildiz, I. F., McNair, J., Ho, J., Uzunalioglu, I. and Wang, W. (1999) Mobility management in next generation wireless systems. *Proc. IEEE*, **87**, 1347–1384.

[4] Bhattacharya, A. and Das, S. K. (1999) LeZi-Update: an information theoretical approach to track mobile users in PCS networks. In *Proc. 5th Ann. Int. Conf. on Mobile Computing and Networking (MobiCom'99)*, Seattle, Washington, August 17–19, pp. 1–12. ACM Press, New York.

[5] Chen, I. R., Chen, T. M. and Lee, C. (2001) Agent-based forwarding strategies for reducing location management cost in mobile networks. *ACM/Baltzer J. Mobile Networks Applic.*, **6**, 105–116.

[6] Krishnamurthi, G., Azizoglu, M. and Somani, A. (1998) Optimal location management algorithms for mobile networks. In *Proc. 4th Ann. Int. Conf. on Mobile Computing and Networking (MobiCom'98)*, Dallas, TX, October 15–30, pp. 223–232. ACM Press, New York.

[7] Sen, S. K., Bhattacharya, A. and Das, S. K. (1999) A selective location update strategy for PCS users. *ACM/Baltzer Wireless Networks*, **5**, 313–326.

[8] Wong, V. W. S. and Leung, V. C. M. (2000) Location management for next-generation personal communications networks. *IEEE Network*, **14**, 18–24.

[9] Cho, G. and Marshall, L. F. (1995) An efficient location and routing scheme for mobile computing environments. *IEEE J. Selected Areas Communi.*, **13**, 868–879.

[10] Ho, J. S. M. and Akyildiz, I. F. (1997) Dynamic hierarchical database architecture for location management in PCS networks. *IEEE/ACM Trans. Networking*, **5**, 646–660.

[11] Krishna, P., Vaidya, N. H. and Pradhan, D. K. (1996) Static and dynamic location management in mobile wireless networks. *Comput. Communi.*, **19**, 81–88.

[12] Pitoura, E. and Samaras, G. (2001) Locating objects in mobile computing. *IEEE Trans. Knowledge Data Eng.*, **13**, 571–592.

[13] Jain, R., Lin, Y. B., Lo, C. and Mohan, S. (1994) A caching strategy to reduce network impacts of PCS. *IEEE J. Selected Areas in Commun.*, **12**, 1434–1444.

[14] Jain, R., Lin, Y. B., Lo, C. and Mohan, S. (1995) A forwarding strategy to reduce network impacts of PCS. In *Proc. 14th Ann. Joint Conf. of the IEEE Computer and Communications Societies (IEEE INFOCOM '95)*, Boston, MA, April 2–6, pp. 481–489. IEEE Computer Society Press, Los Alamitos, CA.

[15] Rao, S. and Gopinath, B. (1993) *Optimizing Call Management of Mobile Units*. WINLAB-TR 63, Wireless Information Network Laboratory, Rutgers University.

[16] Akyildiz, I. F., Ho, J. S. M. and Lin, Y.-B. (1996) Movement-based location update and selective paging for PCS network. *IEEE/ACM Trans. Networking*, **4**, 629–638.

[17] Ho, J. S. M. and Akyildiz, I. F. (1996) Local anchor scheme for reducing signaling costs in personal communications networks. *IEEE/ACM Trans. Networking*, **4**, 709–725.

[18] Bellcore (1994) *Generic Criteria for Version 0.1 Wireless Access Communications Systems (WACS) and Supplement*. Technical Report TR-INS-001313, Issue 1, Bellcore.

[19] Rose, C. and Yates, R. (1997) Ensemble polling strategies for increased paging capacity in mobile communication networks. *ACM/Baltzer Wireless Networks*, **3**, 159–167.

[20] Chen, I. R., Chen, T. M. and Lee, C. (1998) Performance evaluation of forwarding strategies for location management in mobile networks. *Comp. J.*, **41**, 243–253.

[21] Choi, H., Kulkarni, V. G. and Trivedi, K. S. (1994) Markov regenerative stochastic Petri nets. *Perform. Eval.*, **20**, 337–357.

[22] Lin, Y. B. (1997) Reducing location update cost in a PCS network. *IEEE/ACM Trans. Networking*, **5**, 25–33.

[23] Sahner, R., Trivedi, K. S. and Puliafito, A. (1996) *Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package*. Kluwer Academic, Norwell, MA.

[24] Jain, R. and Krishnakumar, N. (1994) Network support for personal information services to PCS users. In *IEEE Conf. Networks for Personal Communications*, Long Branch, NJ, March 16–18, pp. 1–7. IEEE Computer Society Press, Los Alamitos, CA.