Agent-Based Forwarding Strategies for Reducing Location Management Cost in Mobile Networks

Ing-Ray Chen
Department of Computer Science
Virginia Tech
irchen@cs.vt.edu

Tsong-Min Chen and Chiang Lee
Computer Science and Information Engineering Dept.
National Cheng Kung University
{chentm,leec}@server2.iie.ncku.edu.tw
Tainan, Taiwan

Abstract

An important issue in location management for dealing with user mobility in wireless networks is to reduce the cost associated with location updates and searches. The former operation occurs when a mobile user moves to a new location registration area and the network is being informed of the mobile user's current location; the latter operation occurs when there is a call for the mobile user and the network must deliver the call to the mobile user. In this paper, we propose and analyze a class of new agent-based forwarding schemes with the objective to reduce the location management cost in mobile wireless networks. We develop analytical models to compare the performance of the proposed schemes with existing location management schemes to demonstrate their feasibility and also to reveal conditions under which our proposed schemes are superior to existing ones. Our proposed schemes are particularly suitable for mobile networks with switches which can cover a large number of location registration areas.

Index Terms — Location management algorithms, personal communication service networks, two location algorithm, agent-based forwarding algorithm, Markov chains.

1 Introduction

Personal Communication Services (PCS) networks provide wireless communication services to mobile users at any time and any place. In order to search a mobile user (MU) in response to a call, an efficient location management scheme is important. A location management scheme must handle two issues efficiently: location updates and searches. The former operation occurs when a MU moves to a new Visitor Location Register (VLR) area and the network is being informed of where the MU is currently located; the latter operation occurs when there is a call for the MU and the network must route the call to the MU.

Under the basic IS-41 [3] scheme, a MU is permanently registered under a location register called the Home Location Register (HLR). Whenever a MU moves into a new VLR, the MU's HLR is informed of the location change so that it keeps track of the current VLR exactly. When there is a call asking for the MU, the system must query the HLR to get the location of the called MU's current VLR. This scheme is also known as the basic HLR/VLR scheme.

In recent years, several location algorithms have been proposed to reduce the location update cost in mobile wireless networks. When the frequency of the incoming calls is higher than the mobile user's mobility rate, that is, when the call-to-mobility ratio (CMR) is high, the location cache scheme [5] is proposed to reduce the number of searching operations. When CMR is low, on the other hand, it is reported that the forwarding and resetting algorithm (FRA) [2, 6], the alternative location strategy (ALS) [12] and the two location algorithm (TLA) [8, 9] can be used to reduce the location update cost. The CMR value is presumably a per-MU measure since the mobility pattern and the call frequency vary from one MU to another. Therefore, different location algorithms can be adopted depending on the CMR value of the MU user in question. To compare the performance of location management algorithms, one should compare them under the same network environment setting for the same CMR value.

In this paper, we propose and analyze a class of agent-based forwarding strategies that extend and enhance the TLA scheme proposed by Lin [8] by applying the local agent concept [1, 12] and the forwarding concept [2, 6]. Our strategies are intended for MUs with relatively low CMR values, as TLA, ALS and FRA were designed for. The TLA scheme originally proposed by Lin [8] was introduced to reduce the location update cost by recording two most recently visited VLRs in both the HLR database and the MU database. In some cases such as the MU moves back and forth between two VLRs, the location database in the HLR does not have to be updated, thus saving the network cost due to location updates. The saving is especially significant when the CMR value of the MU is low. Of course, a penalty has to be paid when there is a "location miss" since the HLR's database is not up-to-date all the time and two searches instead of just one may be needed to locate the MU. Therefore, if the CMR value is large enough, the cost saving in location updates will eventually be outweighed by the high cost in location searches.

In this paper, we show that the performance of our proposed agent-based forwarding strategies can perform better than the TLA scheme which has been shown to perform very well at low CMR values [8, 9]. We present two strategies in this context, namely, the Agent-Based Two-Location Algorithm (ATLA) and the Cross-Update Agent-Based Two-Location Algorithm (CATLA) strategies. The basic idea is to use a VLR as the local agent on behalf of the HLR while applying the TLA scheme to the local agent instead of to the HLR. This effectively reduces the VLR-HLR communication cost to the VLR-VLR communication cost. The local agent of a MU may change as the MU moves across the VLR boundary, depending on the rules being applied. CATLA is different from ATLA by virtue of one rule: the local agent is replaced whenever the MU moves to a new VLR covered by a different network switch from the one that covers the current VLR. Our performance analysis is based on analytical modeling. Specifically, we develop three separate Markov models to describe the behavior of the network under TLA, ATLA, and CATLA, respectively. We then give values to model parameters for the same network setting (by means of a network coverage model) and workload condition, and then assess if our scheme works better. In this paper, we use the combined cost due to location searches and updates between two successive calls to a MU as a metric to compare our schemes with existing ones, as having been done in [9].

The rest of the paper is organized as follows. Section 2 introduces the background. We first describe existing schemes, including the IS-41 basic HLR/VLR and TLA schemes. Then we describe our proposed ATLA and CATLA schemes in detail to deal with mobility-induced location management issues. In Section 3, we develop three Markov models to describe the behavior of the network operating under TLA, ATLA and CATLA separately. Section 4 parameterizes these Markov models based on a hexagonal network coverage model so as to compare the performance of ATLA and CATLA schemes with IS-41 and TLA under identical conditions. Section 5 summarizes the paper and outlines some possible future research areas.

2 Background

A Personal Communication Services (PCS) network architecture is shown as in Figure 1. The regions covered by the wireless communicable area are divided into many registration areas (RAs)

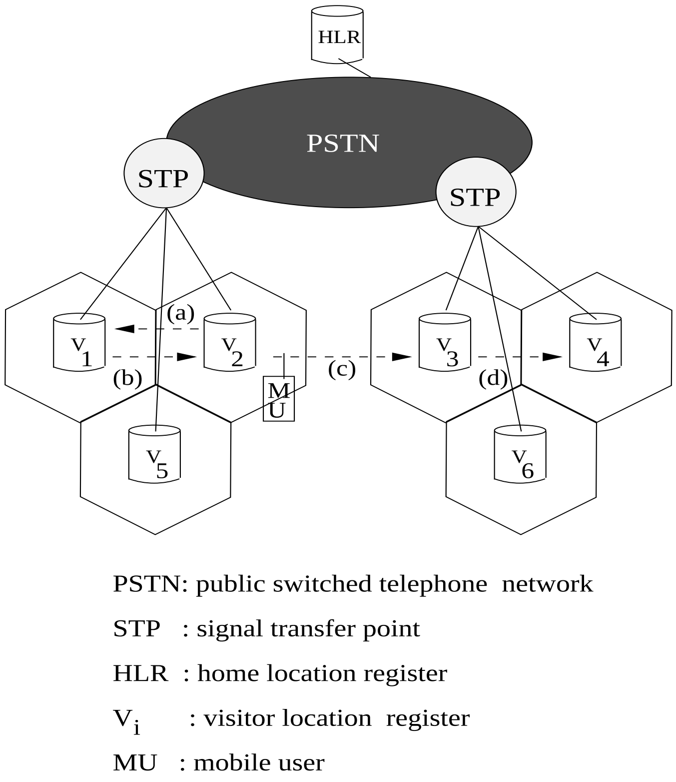

Figure 1: A Mobile Communication Network.

[4]. Assume that each RA has its own VLR. Typically, in a network as such there are switches connecting the HLR to VLRs and also between VLRs. For example, $V1$, $V2$ and $V5$ are three VLRs under one switch transfer point (STP), while $V3$, $V4$ and $V6$ are under another STP. STPs are connected by the public switch telephone system (PSTN). Figure 1 also shows a hierarchical PCS network as discussed in [5] in which there is only one HLR for each MU, but the MU may go to different registration areas under different VLRs (marked V_0, V_1, V_2, etc. in Figure 1). As we can see, when the HLR and VLRs are in different segments of the network, the location update cost can be substantial under the basic HLR/VLR scheme, especially when the MU moves frequently since every move involves a connection cost between the HLR and the new VLR which the MU moves into.

2.1 IS-41 Basic HLR/VLR Scheme

This basic HLR/VLR scheme exists in IS-41 [3] in the United States and GSM [10] in Europe. Under the basic HLR/VLR scheme, a MU is permanently registered under a location register called the home location register (HLR). When the MU moves to a new location area, it reports to the new VLR which in turn sends a location update message to the HLR that the MU is now under its area. When there is a caller asking for the MU, the caller first checks the local VLR for the MU's profile. If the profile is not in the local VLR, a message is sent to the HLR querying the MU's location. The MU's HLR verifies the current VLR location and then returns the location to the caller.

2.2 Two Location Algorithm (TLA)

Under the two location algorithm (TLA) scheme, a MU and its HLR each keep a *location table* to store two recently visited VLRs. When a MU moves to a new VLR which is not one of the two recently visited VLRs recorded in the table, an update operation is initiated by the mobile user so that both the location tables in the mobile unit and in HLR are updated. Figure 1 shows a MU moving from V_2 to V_1 and then back to V_2; after moving back to V_2, the MU moves to V_3 and finally to V_4. In Table 1, we illustrate TLA by showing the contents of the databases maintained by the HLR and MU upon movements (a) through (d). We use "*" to mark the current VLR being visited by the MU. Initial, assume that the MU is in V_2. When the MU makes movement (a), an update operation is performed with the HLR and the location tables of HLR and MU are both updated to (V_1^*, V_2), with V_1 being the current VLR. For the HLR's database, this sequence also corresponds to the search sequence. After movement (b), the MU moves back to the previously visited VLR, i.e., V_2, but the HLR database does not need to be updated since V_2 is already in the HLR's location table. Note that in this state, the HLR's database is inconsistent, i.e., it still keeps V_1 as the current VLR of the MU and thus will search V_1 first to service a call. In this case, if a call arrives then there will be a location miss first at V_1, and a double search cost will incur, although the MU will eventually be found at V_2. Continuing with our movement example, the MU subsequently makes movements (c) and (d), each of which also requires a location update to the

HLR database.

Table 1. Databases Stored in the HLR and MU under TLA.

Movement	*HLR Location Table*	*MU Location Table*	*Update Cost*
(a) $V2 \rightarrow V_1$	(V_1, V_2)	(V_1^*, V_2)	VLR-HLR
(b) $V1 \rightarrow V_2$	(V_1, V_2)	(V_2^*, V_1)	No
(c) $V2 \rightarrow V_3$	(V_3, V_2)	(V_3^*, V_2)	VLR-HLR
(d) $V3 \rightarrow V_4$	(V_4, V_3)	(V_4^*, V_3)	VLR-HLR

2.3 ATLA

In this section, we describe a proposed new scheme called ATLA. This agent-based forwarding strategy exploits the locality properties of the MU's movement and access history pattern. The idea is as follows: The HLR at all times only points to a single (but changeable) VLR as in the IS-41 basic HLR/VLR scheme. We will call this VLR the local agent of the MU. As in IS-41, this local agent which the HLR points to can be replaced as the MU moves from one VLR to another. However, during its tenure period serving as a local agent for the MU, it executes the TLA scheme on behalf of the HLR.

Specifically, when the MU moves from the local agent to a nearby VLR, only a forwarding pointer is setup between the local agent and the new VLR, and no location update operation is performed to the HLR. We also refer to this cost as the VLR-VLR "binding" cost. After that, the MU can move back and forth between the two VLRs, including the local agent, without informing the local agent or the HLR. When a call arrives, the local agent will first attempt to locate the MU in its area. If the MU cannot be found, then it will follow the forwarding pointer to the next VLR to search for the called MU, thus increasing the search cost. Note that, however, this search cost is still less than that of the TLA scheme because on the second search attempt, the system only has to pay the VLR-VLR communication cost in the ATLA scheme by following the VLR-VLR link, as opposed to the VLR-HLR communication cost in the TLA scheme.

Note that in the ATLA scheme, the MU must also record two recently visited VLRs in the table as in TLA, including the local agent. We will call the non-agent VLR in the table as the 2nd VLR. When the MU moves from the local agent VLR to a new VLR, it will inform the local agent to update its forwarding pointer to point to the new VLR which it just entered, thus making the new

VLR as the 2nd VLR, without notifying the HLR. However, If the move is from the 2nd VLR to a new VLR, then an update operation to the HLR must be performed, thus replacing the local agent with the new VLR just entered while keeping the 2nd VLR unchanged. Table 2 illustrates ATLA by showing the contents of the databases stored in the HLR and MU based on the ATLA scheme. After making movement (a), the HLR still records V_2 as the agent, the agent (i.e., V_2 at this point) sets up a pointer to V_1, and the MU's location table stores (V_1^*, V_2), with V_1 being marked with "*" because it is the current VLR. In this case, the MU only sends a VLR-VLR binding message to the local agent without informing the HLR about the location change. After movement (b), the MU moves back to the previous VLR, V_2, so it does not need to do anything except updating its own database to (V_2^*, V_1). When the MU makes movement (c), it discovers that it moves into a new VLR V_3, so it sends a VLR-VLR binding message to the local agent V_2. Finally, when the MU moves into V_4 from the 2nd VLR V_3, it performs an update operation to the HLR, thus replacing V_2 with V_4 as the new local agent.

Another note that is worth mentioning is the concept of state inconsistency. In Table 2, we see that after movement (a) or (c), the system is in an inconsistent state, i.e., the pointer from the HLR points to a VLR that is not the current VLR, while after movement (b) or (d), the system is in a consistent state. This can also be observed from Table 2 by combining columns 2 and 3 and see if it is the same as column 4 in sequence.

Table 2. Databases Stored in the HLR, Local Agent and MU under ATLA.

Movement	*HLR Pointer*	*Local Agent Pointer*	*MU Table*	*Update Cost*
(a) $V_2 \rightarrow V_1$	V_2	V_1	(V_1^*, V_2)	VLR-VLR
(b) $V_1 \rightarrow V_2$	V_2	V_1	(V_2*, V_1)	No
(c) $V_2 \rightarrow V_3$	V_2	V_3	(V_3^*, V_2)	VLR-VLR
(d) $V_3 \rightarrow V_4$	V_4	V_3	(V_4*, V_3)	VLR-VHR

2.4 CATLA

The CATLA strategy is the same as the ATLA strategy except that an update operation to the HLR will occur whenever the MU moves across a network switch boundary. The idea is to reduce the average VLR-VLR communication cost between the local agent and the 2nd VLR. In this scheme, if the MU enters a new VLR covered by another network switch (e.g., a STP in Figure

1), the MU will update the HLR directly because otherwise the VLR-VLR communication cost crossing the STP switch boundary is expensive and such cost would strike twice: once when the pointer is set-up between the local agent and the 2nd VLR (which are under two different STPs), and once when a call arrives subsequently and a location miss results. Because of this force-update rule, we expect that CATLA may gradually perform better than ATLA as the CMR value of the MU increases. Table 3 illustrates CATLA by showing the contents of the databases maintained by the HLR, local agent, and MU based on CATLA for the same movement scenario discussed earlier. Movements (a) through (b) are the same as in the ATLA scheme, since no switch boundary is crossed. Movement (c), however, incurs a location update operation to the HLR since V_2 and V_3 are under two separate STPs (see Figure 1), after which V_3 becomes the local agent and the 2nd VLR is nil. Movement (d) is also different from that in the ATLA scheme because V_3 now is the local agent, not the 2nd VLR as in the ATLA scheme. Therefore, when the MU enters V_4 from V_3, only a pointer is setup from V_3 to V_4 since V_4 is under the same STP as V_3.

Table 3. Databases Stored in the HLR, Local Agent and MU under CATLA.

Movement	*HLR Pointer*	*Local Agent Pointer*	*MU Table*	*Update Cost*
(a) $V_2 \rightarrow V_1$	V_2	V_1	(V_1^*, V_2)	VLR-VLR
(b) $V_1 \rightarrow V_2$	V_2	V_1	($V_2{}^*$, V_1)	No
(c) $V_2 \rightarrow V_3$	V_3	nil	(V_3^*, nil)	VLR-HLR
(d) $V_3 \rightarrow V_4$	V_3	V_4	($V_4{}^*$, V_3)	VLR-VLR

3 Modeling the Location Strategies

In this section, we develop three separate Markov models to describe the performance characteristics of the PCS network operating under TLA, ATLA and CATLA. We will later apply the result obtained from these models to compare TLA, ATLA, CATLA and IS-41 under identical conditions based on a hexagonal network structure.

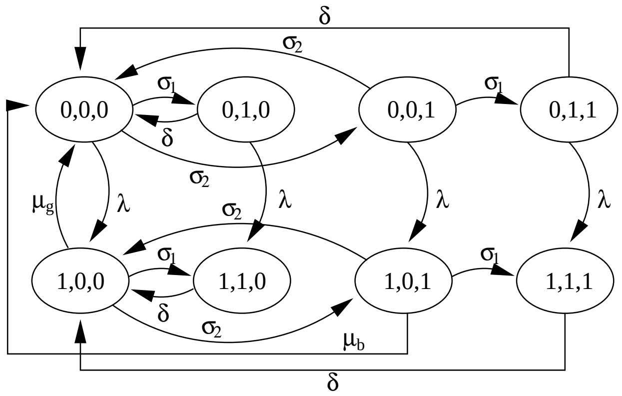

Figure 2: Markov Model for PCS Network under Two Location Algorithm.

3.1 Modeling TLA

Table 4. Notation Used for the TLA Model.

λ	arrival rate of calls to the MU.
σ	mobility rate of the MU.
θ	probability of the MU moving back to the previous VLR
CMR	call-to-mobility ratio of the MU, i.e., CMR=λ/σ
σ_1	mobility rate of the MU moving to a new VLR, i.e., $\sigma_1 = (1-\theta)\sigma$
σ_2	mobility rate of the MU moving to the previous VLR, i.e., $\sigma_2 = \theta\sigma$
δ	execution rate in updating the location table stored in the HLR
μ_g	execution rate in locating the MU when the location table in the mobile unit is consistent with that in the HLR
μ_b	execution rate in locating the MU when the location table in the mobile unit is inconsistent with that in the HLR

The state of a MU as it crosses database boundaries while being called can be described by a 3-component state description vector (a, b, c). Component a is a binary quantity indicating whether or not the mobile unit is in the state of being called, with 0 standing idle and 1 standing busy. Component b is also a binary quantity indicating if the MU has just moved to a new VLR area, with 1 meaning yes and 0 meaning no. The third component, c, indicates if the location table maintained by the mobile unit is inconsistent with that maintained by the HLR, with 0 meaning consistent and 1 meaning inconsistent.

Figure 2 shows the Markov model for describing the PCS operating under TLA. Table 4 shows the notation used for the TLA model. Initially, the MU is in the state of $(0, 0, 0)$, meaning that it is not being called, the MU has not yet made any move across any VLR boundary, and the location table stored in the HLR is consistent with that stored in the MU. Below, we explain briefly how we construct the Markov model.

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

Second, if the MU is in the state of $(1, i, j)$ and another call arrives, then the MU will remain at the same state, since the MU remains in the state of being called. This behavior is described by a hidden transition from state $(1, i, j)$ back to itself with a transition rate of λ. This type of transition is not shown in Figure 2 since it does not need to be considered when solving a Markov chain [7]. Note that this implies that in state $(1, i, j)$ the number of requests accumulated to locate the MU may be greater than one.

Third, if the MU is in the state of $(1, 0, 0)$, it means that the location table stored in the HLR is consistent with that stored in the mobile unit and the mobile unit is in the state of being called. Therefore, the PCS network can service all pending calls simultaneously with a service rate of μ_g. After the service, the new state is $(0, 0, 0)$.

Four, if the MU is in the state of $(1, 0, 1)$, it means that the location table stored in the HLR is inconsistent with that stored in the mobile unit but there are pending calls waiting to be serviced. Therefore, the PCS network has to spend twice as much time to locate the mobile unit. This behavior is modeled by using a different service rate of μ_b from state $(1, 0, 1)$ to state $(0, 0, 0)$. After the service, the new state is $(0, 0, 0)$ because the location table stored in the HLR is updated after the call delivery service and is therefore consistent with that stored in the mobile unit again.

Lastly, regardless of whether or not the MU is in the state of being called, the MU can move across a VLR boundary. There are two cases:

1. If the mobile unit moves to a new VLR then an update operation has to be performed to the table stored in the HLR. This behavior is modeled by a transition from state $(i, 0, j)$ to $(i, 1, j)$, $0 \leq i, j \leq 1$, with a transition rate σ_1, after which the system transmits from state $(i, 1, j)$ to state $(i, 0, 0)$ with a transition rate of δ.

2. If the MU moves back to the previously visited VLR, then there is no update operation required to update the HLR associated with this event. This is modeled by a transition from state $(i, 0, 0)$ to state $(i, 0, 1)$, $0 \leq i \leq 1$, with rate σ_2. This results in an inconsistent state, i.e., after the transition the location table stored in the HLR is consistent with that stored in the MU. Another possible transition is the reverse of the above, that is, from state $(i, 0, 1)$ to state $(i, 0, 0)$, $0 \leq i \leq 1$, also with a transition rate σ_2. This also results in a consistent state. Note that the update time of the location table stored in the mobile unit with respect to the PCS network is zero (although the mobile unit's location table is updated itself). Therefore, there is no need to model the time needed to perform the update operation in this case.

The Markov chain shown in Figure 2 is ergodic [7], which means that all states have a non-zero probability. The probability that the system is found in a particular state in equilibrium depends on the relative magnitude of the outgoing and incoming transitions rates. Let $P_{(i,j,k)}$ is the probability that the system stays in state (i,j,k) in equilibrium. Let TLA_{update} be the average cost of the PCS network in servicing a location update operation and let TLA_{call} be the average cost in locating the MU. Furthermore, let TLA_{cost} be the average cost of the PCS network in servicing the above two types of operations between two consecutive calls. Then,

$$\begin{aligned} TLA_{update} = & \left(\sum_{i=0}^{1} (P_{(0,0,i)} + P_{(1,0,i)}) \times (1-\theta) \times (1/\delta) \right) \\ & + \left(\sum_{i=0}^{1} (P_{(0,1,i)} + P_{(1,1,i)}) \times (1/\delta) \right) \end{aligned} \tag{1}$$

$$\begin{aligned} TLA_{call} = & \left(\sum_{i=0}^{1} \sum_{j=0}^{1} P_{(i,j,0)} \times (1/\mu_g) \right) + \\ & \left(\sum_{i=0}^{1} \sum_{j=0}^{1} P_{(i,j,1)} \times (1/\mu_b) \right) \end{aligned} \tag{2}$$

$$TLA_{cost} = TLA_{update} \times \sigma/\lambda + TLA_{call} \tag{3}$$

Equation (3) is obtained above because between two consecutive calls, the number of mobility moves across VLR boundaries by the MU is equal to σ/λ on average. Note that the number of moves corresponds to the number of update operations, although some of which may not cause any

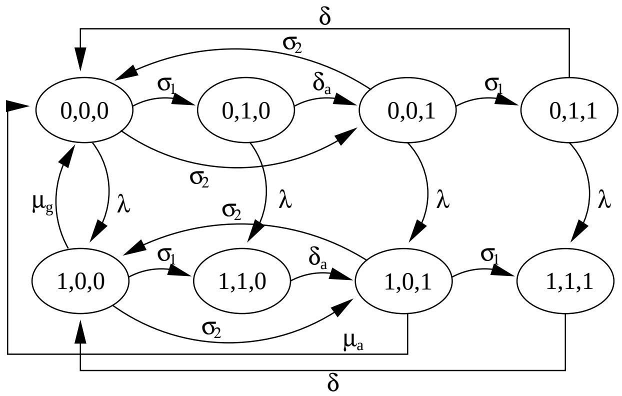

Figure 3: Markov Model for PCS Network under ATLA.

update cost to the PCS network depending on whether or not the location table in the HLR needs to be updated.

3.2 Modeling ATLA

Table 5. Additional Parameters Used in the ATLA Model.

δ_a	execution rate in updating the location table stored in the agent
μ_a	execution rate in locating the MU when the local agent is not the current VLR.

To model ATLA, we use the same set of parameters in Table 4 less the last parameter μ_b, and introduce two more parameters listed in Table 5. We describe ATLA by also using a 3-component vector (a, b, c). The meanings of components a and b are the same as before. The third component, c, indicates if the MU currently resides under the agent area, i.e., if the local agent is the current VLR, with 0 meaning that it is, and 1 meaning that it is not. Conceptually, the third component indicates if the HLR points to the current VLR, with 0 meaning yes and 1 meaning no. If yes, the system is in a consistent state in which the state information maintained by the HLR is consistent and the search operation can be done efficiently; otherwise, the system is in an inconsistent state and must follow a forwarding pointer to locate the MU.

Figure 3 shows the Markov model for describing the PCS operating under ATLA. Initially, the MU is in the state of $(0, 0, 0)$, meaning that it is not being called, has not moved across a VLR boundary, and the MU is currently located under the local agent (note that the HLR always points to the local agent in the ATLA scheme). Below, we explain briefly the differences between TLA and ATLA as we construct the Markov model for ATLA. The ATLA model is almost the same as the TLA model, except that we replace the transitions from state $(i,1,0)$ to state $(i,0,0)$ by those from state $(i,1,0)$ to state $(i,0,1)$, $0 \leq i \leq 1$, and replace μ_b by μ_a.

If the MU is in the state of $(1,0,1)$, it means that there are pending calls waiting to be serviced in this state but the MU is not in the cover area of the local agent, i.e., the local agent is not the current VLR. In this case, the system needs to follow the forwarding pointer from the local agent to locate the current VLR. This behavior is modeled by using a service rate of μ_a from state $(1,0,1)$ to state $(0,0,0)$. After the service, the new state is $(0,0,0)$ because the HLR database is updated after the call delivery service and is therefore consistent with that stored in the MU again. This rate μ_a is different from μ_b in Figure 2 for the TLA model since only a VLR-VLR communication cost incurs in the second search attempt by following the forwarding pointer. Recall that for the TLA model, the second search attempt also involves a VLR-HLR communication cost.

When the MU moves across a VLR boundary, there are three cases to be considered:

1. If the MU moves from the local agent to a new 2nd VLR, then a binding operation has to be performed to bind the new 2nd VLR to the agent by means of a forwarding pointer. This behavior is modeled first by a transition from state $(i,0,0)$ to $(i,1,0)$, $0 \leq i \leq 1$, with a transition rate σ_1, after which the system goes from state $(i,1,0)$ to state $(i,0,1)$ with a transition rate of δ_a. The first transition models the move event while the second models the binding (setting up pointer) event.

2. If the MU moves from a VLR, V_i, to a new VLR, V_j, where V_i and V_j are not the agent, then an update operation has to be performed to the table stored in the HLR. After the update operation, the location table of the HLR records the new VLR as the agent. This behavior is modeled by a transition from state $(i,0,1)$ to $(i,1,1)$, $0 \leq i \leq 1$, with a transition rate σ_1, after which the system transits from state $(i,1,1)$ to state $(i,0,0)$ with a transition rate of δ. Note that the originating state in this case is $(i,0,1)$, $0 \leq i \leq 1$, with the 2nd state component being 0 (meaning it has not moved to a new VLR) and the 3rd state component being 1 (meaning the agent is not the current VLR).

3. If the MU moves back to the previously visited VLR, then there is no update operation required to perform the HLR associated with this update event. This is modeled by a transition from state $(i,0,0)$ to state $(i,0,1)$ with a transition rate σ_2, after which the location table stored in the HLR is inconsistent with that stored in the MU, or from state $(i,0,1)$ to state $(i,0,0)$, $0 \leq i \leq 1$, also with a transition rate σ_2, after which the location table stored in the HLR becomes consistent with that stored in the MU. Note that again the update time of the location table of the MU with respect to the PCS network is zero.

Let $ATLA_{update}$ be the average cost of the PCS network in servicing an update operation and let $ATLA_{call}$ be the average cost in locating the MU. Furthermore, let $ATLA_{cost}$ be the average cost of the PCS network in servicing the above two types of operations between two consecutive calls. Then,

$$\begin{aligned}
ATLA_{update} &= \left(\sum_{i=0}^{1}(P_{(i,0,0)}) \times (1-\theta) \times (1/\delta_a)\right) \\
&+ \left(\sum_{i=0}^{1}(P_{(i,1,0)}) \times (1/\delta_a)\right) + \left(\sum_{i=0}^{1}(P_{(i,1,1)}) \times (1/\delta)\right) \\
&+ \left(\sum_{i=0}^{1}(P_{(i,0,1)}) \times (1-\theta) \times (1/\delta)\right)
\end{aligned} \tag{4}$$

$$\begin{aligned}
ATLA_{call} &= \left(\sum_{i=0}^{1} P_{(i,0,0)} \times (1/\mu_g)\right) + \\
&\left(\sum_{i=0}^{1} P_{(i,1,0)} \times (1/\mu_a)\right) + \left(\sum_{i=0}^{1}\sum_{j=0}^{1} P_{(i,j,1)} \times (1/\mu_a)\right)
\end{aligned} \tag{5}$$

$$ATLA_{cost} = ATLA_{update} \times \sigma/\lambda + ATLA_{call} \tag{6}$$

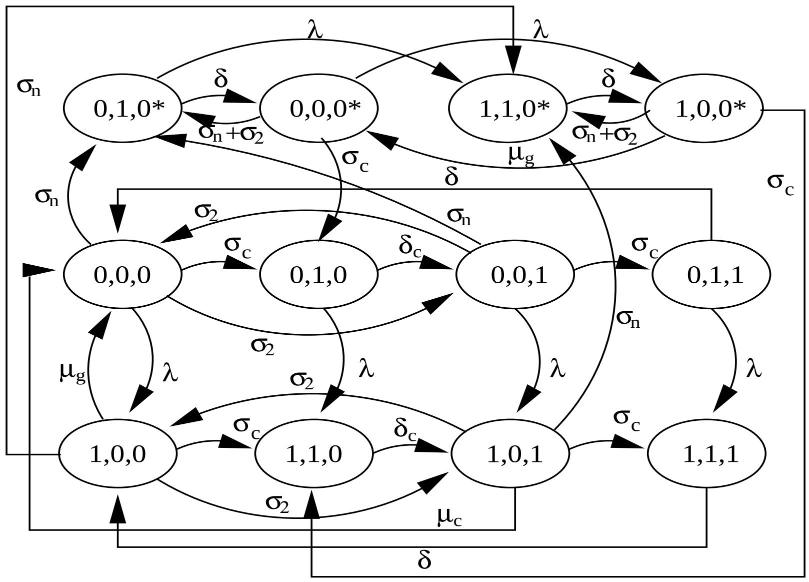

Figure 4: Markov Model for PCS Network under CATLA.

3.3 Modeling CATLA

Table 6. Additional Parameters Used in the CATLA Model.

δ_c	execution rate in updating the location table of the local agent
μ_c	execution rate in locating the MU when the local agent is not the current VLR.
γ	probability of the MU moving within the same network switch.
σ_n	mobility rate of the MU moving to a new VLR not covered by the same network switch, i.e., $\sigma_n = (1-\gamma)(1-\theta)\sigma = (1-\gamma)\sigma_1$
σ_c	mobility rate of the MU moving to a new VLR covered by the same network switch, i.e., $\sigma_c = \gamma(1-\theta)\sigma = \gamma\sigma_1$

To model CATLA, we again use the same set of parameters in Table 4 except μ_b, along with three more parameters, δ_c, μ_c and γ, listed in Table 6. For notational convenience, let $\sigma_n = (1-\gamma)\sigma_1$ and $\sigma_c = \gamma\sigma_1$.

Figure 4 shows a Markov model for describing the PCS operating under CATLA. The same 3-component state description as defined in the ATLA model is used here. Recall that under CATLA, whenever the MU moves across a network switch boundary, the HLR database is updated, in which case the new agent does not have a forwarding pointer in its local database, i.e., the 2nd VLR is nil. We differentiate this particular state by using the symbol "*". For example, in state $(0,0,0)^*$

the forwarding pointer stored in the agent is nil, while in state $(0,0,0)$ the forwarding pointer does exist. Note that in both cases, the system is in a consistent state, i.e., the local agent is the current VLR. The Markov model for CATLA is self-explanatory. Below we discuss the behavior of the MU as it moves to a new VLR. There are three possible cases:

1. If the MU moves to a new VLR covered by a different network switch (the mobility rate of which is σ_n), then an update operation is performed to the database stored in the HLR. This behavior is modeled first by a transition from state $(i,0,j)$ to $(i,1,0)^*$, $0 \leq i,j \leq 1$, with a transition rate σ_n, after which the system goes from $(i,1,0)^*$ to $(i,0,0)^*$ with a transition rate of δ to update the HLR's database to point to the new VLR. After the update, the new VLR becomes the local agent of the MU, but in its database the forwarding pointer to the 2nd VLR is nil because in this case the 2nd VLR does not exist.

2. After the MU just crosses a network switch, it is possible that the MU may subsequently cross a network switch boundary again. This event comes in two forms: (a) the MU subsequently moves back to the previously visited VLR, the mobility rate of which is σ_2; (b) the MU simply enters another VLR not covered by the same switch, the mobility rate of which is σ_n. In either case, another update operation will be triggered to update the HLR database. This behavior is modeled first by a transition from state $(i,0,0)^*$ to state $(i,1,0)^*$, $0 \leq i \leq 1$, (the top row of Figure 4) with a transition rate $\sigma_2 + \sigma_n$, after which the system goes from state $(i,1,0)^*$ to state $(i,0,0)^*$ with a transition rate of δ. After the MU just crosses a network switch, another likely event is that it subsequently goes to another VLR also within the same network switch. This is modeled first by a transition from state $(i,0,0)^*$ to state $(i,1,0)$, $0 \leq i \leq 1$, with a transition rate σ_c, after which the system goes from state $(i,1,0)$ to state $(i,0,1)$ with a transition rate of δ_c to update the pointer stored in the local agent. In this event, the local agent of the MU remains unchanged but the target state becomes inconsistent.

3. When the local agent and the 2nd VLR are not nil, if the MU moves to a VLR within the same network switch that covers the current VLR (which can be the local agent or the 2nd VLR), the system behaves the same as in the ATLA model. This behavior is described by

the state transitions in the last two rows of Figure 4. Note that here the third component in the state description tells us exactly whether or not the local agent is the current VLR. The target VLR which the MU moves into in this case can be the previously visited VLR (the mobility rate of which is σ_2) or a new VLR (the mobility rate of which is σ_c). In both cases, the MU stays within the same network switch.

Again, let $CATLA_{update}$ be the average cost of the PCS network in servicing an update operation and let $CATLA_{call}$ be the average cost in locating the MU. Furthermore, let $CATLA_{cost}$ be the average cost of the PCS network in servicing the above two types of operations between two consecutive calls. Then,

$$\begin{aligned} & CATLA_{update} = \\ & \tfrac{1}{\delta_c}\left(\sum_{i=0}^{1}(1-\theta)\gamma(P_{(i,0,0)}+P_{(i,0,0)^*})+P_{(i,1,0)}\right) + \\ & \tfrac{1}{\delta}\left(\sum_{i=0}^{1}(1-\theta)(1-\gamma)P_{(i,0,0)}\right) + \\ & \tfrac{1}{\delta}\left(\sum_{i=0}^{1}(1-\theta)P_{(i,0,1)}+P_{(i,1,0)^*}+(1-\gamma)P_{(i,0,0)^*}\right) \end{aligned} \tag{7}$$

$$\begin{aligned} & CATLA_{call} = \\ & \left(\sum_{i=0}^{1}(P_{(i,0,0)}+P_{(i,1,0)^*}+P_{(i,0,0)^*})\times(1/\mu_g)\right) + \\ & \left(\sum_{i=0}^{1}(P_{(i,1,0)}+P_{(i,0,1)}+P_{(i,1,1)})\times(1/\mu_c)\right) \end{aligned} \tag{8}$$

$$CATLA_{cost} = CATLA_{update}\times\sigma/\lambda + CATLA_{call} \tag{9}$$

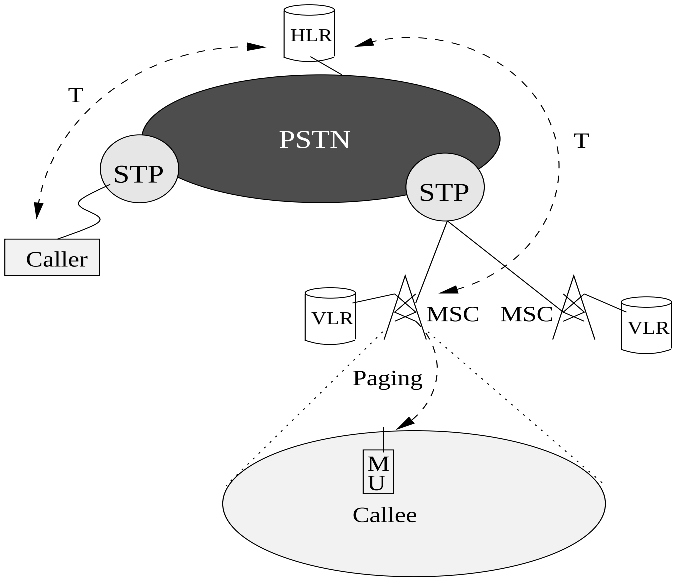

Figure 5: A Search Operation in the Mobile Network.

4 Application

Table 7. Communication Cost Parameters for All Models.

U	the average cost for locating the MU under the basic scheme; it is also the locating cost under the TLA scheme when the HLR's location table is consistent with that stored in the mobile unit, as well as the locating cost under the ATLA and CATLA scheme to find the local agent.
T	the average VLR-HLR communication cost between HLR and VLR; it is also the update cost under the basic scheme and the cost to update the location table in the HLR under the TLA, ATLA and CATLA schemes. In Figure 5, we illustrate the relationship between U and T as follows: $U = 2T + P_c$ where P_c is the average paging cost needed to page a mobile user within a VLR.
τ	the average VLR-VLR communication cost for two VLRs in the network; it is also the cost of a binding operation under the ATLA scheme.
τ'	the average VLR-VLR communication cost for two VLRs under the same STP switch; it is also the cost of a binding operation under the CATLA scheme.

In this section, we apply the result obtained in the last section to compare TLA, ATLA, CATLA and IS-41 under identical conditions. The network cost parameters listed in Table 7 will apply to all location management schemes. Specific values of these network communication cost parameters can be obtained by considering specific network coverage models. We will show how to do so with a hexagonal coverage model. In this paper, we fix the (relative) value of U at 1.0 as used in [8], but use different values of T, τ and $\tau^{'}$ to study their effects. We note that the relation among these last three parameters is

$$\tau = \gamma\tau^{'} + (1-\gamma)T \tag{10}$$

This is so because the STP switches are connected by the PSTN. Therefore, for any two VLRs not under the same STP, the VLR-VLR communication cost is comparable to the VLR-HLR communication cost. Here, γ is as defined in Table 6.

4.1 Parameterization of the TLA Markov Models

There are six parameters in the TLA Markov model (see Figure 2), i.e., σ, λ, θ, δ, μ_g, and μ_b (see Table 4 for their meanings). Of these six parameters, σ, λ and θ are MU dependent parameters and will be studied in the paper by changing their values; on the other hand, δ, μ_g, and μ_b are network structure dependent and can be parameterized as

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

$$\mu_g = \frac{1}{U}$$

and

$$\mu_b = \frac{1}{U+T+P_c} = \frac{1}{2U-T}$$

where U, T and P_c are as defined in Table 7.

4.2 Parameterization of the ATLA Markov Model

In ATLA, we again have three MU dependent parameters, i.e., σ, λ, and θ, which will be studied by changing their values. There are four network structure dependent parameters used in the ATLA Markov model (see Figure 3), i.e., δ, μ_g, δ_a and μ_a (see Tables 4 and 5 for their meanings). The parameterizations of δ and μ_g are the same as in TLA; δ_a and μ_a can be parameterized as follows:

$$\delta_a = \frac{1}{\tau}$$

and

$$\mu_a = \frac{1}{U + \tau + P_c} = \frac{1}{2U - 2T + \tau}$$

4.3 Parameterization of the CATLA Markov Model

In the CATLA Markov model (see Figure 4), we have five network structure dependent parameters, i.e., δ, μ_g, δ_c, μ_c and γ (see Tables 4 and 6 for their meanings) and the same three MU dependent parameters (σ, λ, and θ). Of these five network structure dependent parameters, δ and μ_g can be parameterized the same as before; γ can be derived given a specific network coverage model for which we will show how it can be done with a hexagonal coverage model; δ_c and μ_c can be parameterized as follows:

$$\delta_c = \frac{1}{\tau'}$$

and

$$\mu_c = \frac{1}{U + \tau' + P_c} = \frac{1}{2U - 2T + \tau'}$$

4.4 Comparing TLA, ATLA, CATLA and IS-41

We use IS-41 as the baseline model against which our proposed agent-based forwarding strategies will be compared. The average network cost of the PCS network for location management under IS-41 is given by

$$IS - 41_{cost} = IS - 41_{update} \times \sigma/\lambda + IS - 41_{call} \tag{11}$$

where $IS - 41_{update} = T$ and $IS - 41_{call} = U$, as defined in Table 7.

As in [6], we first study a case in which the VLR-VLR communication cost is one half of the VLR-HLR communication cost, i.e., $\tau/T = 0.5$. To derive γ, we consider a hexagonal network structure model wherein the number of VLRs under a n-layer STP is given by:

$$3n^2 - 3n + 1$$

Figures 6(a) and 6(b) show the number of VLRs covered by a 3-layer STP and a 2-layer STP, respectively, based on this hexagonal network coverage model.

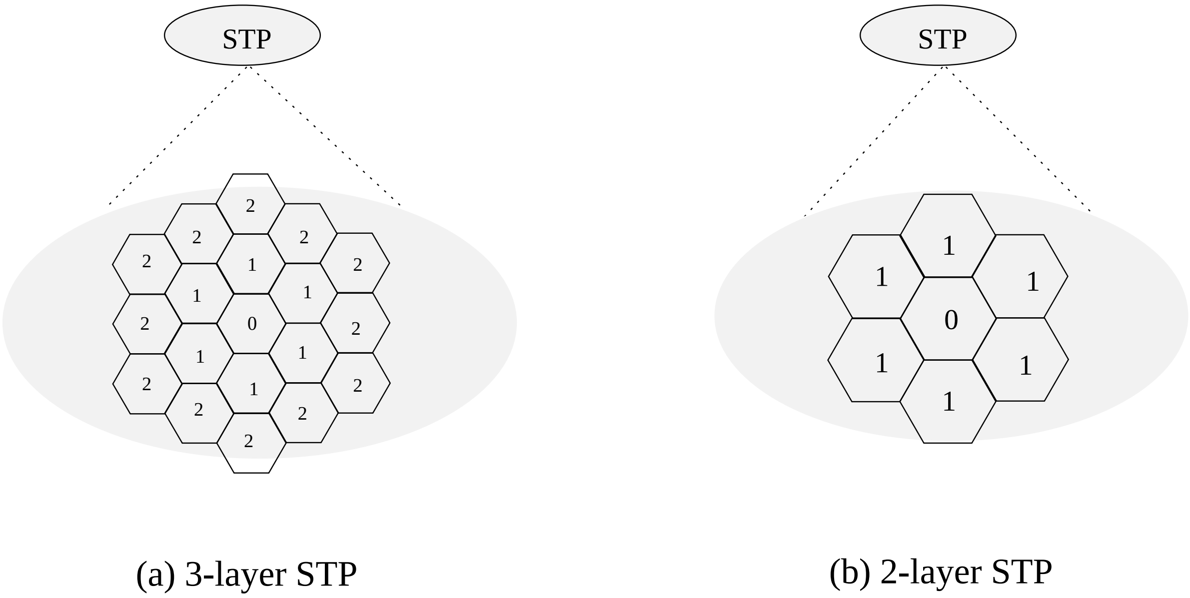

Figure 6: n-Layer STPs under the Hexagonal Network Coverage Model.

Here we consider the case when $n = 2$, i.e., a STP covers 7 VLRs; later, we will study other cases to analyze the effect of n. For the case when $n = 2$, it can be shown that $\gamma = 0.57$ (see [2]). Therefore, when τ/T is 0.5, τ'/T is 0.125 based on Equation 10. Figure 7 shows the average cost of the PCS network due to location management under TLA, ATLA, CATLA and IS-41, as a function of CMR (CMR $= \lambda/\mu$), for the case when $T = 0.4$ and $\theta = 0.3$ (with U fixed at 1). The data on the diagram were obtained by first solving a Markov model using the SHARPE software package [11] to obtain $P_{(i,j,k)}$ for each state (i, j, k) and subsequently computing the average cost based on equations developed in the paper (e.g., Equation 3 for the TLA scheme). Figure 7 illustrates that when the CMR value is small, TLA, ATLA and CATLA all can outperform IS-41, even with θ as small as 0.3. For example, when CMR is 0.1, TLA, ATLA and CATLA schemes reduce the cost of IS-41 by 22%, 35% and 31%, respectively. In addition, ATLA and CATLA perform significantly better than TLA. CATLA is worse than ATLA when CMR is small, but is better as CMR's value increases.

Figure 8 shows a similar condition as in Figure 7 except that $\theta = 0.7$. In this particular case, TLA, ATLA and CATLA significantly outperform IS-41 and also their own respective counterparts

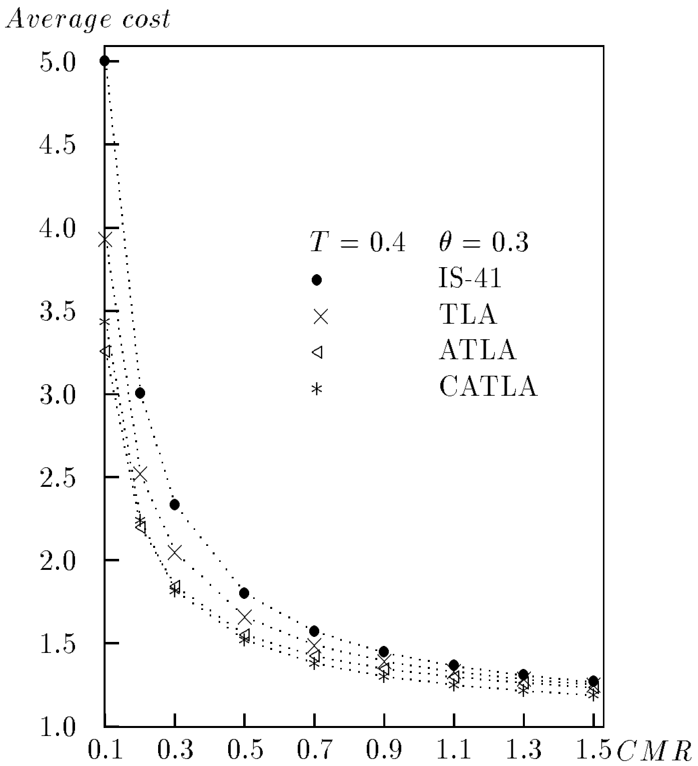

Figure 7: Comparison of IS-41, TLA, ATLA and CATLA with T=0.4 and θ=0.3.

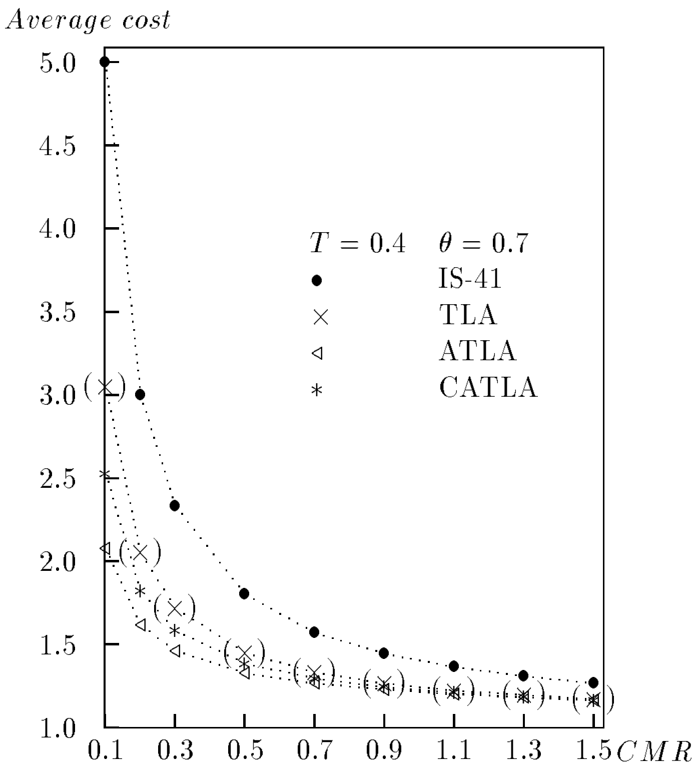

Figure 8: Comparison of IS-41, TLA, ATLA and CATLA with T=0.4 and θ=0.7.

in Figure 7 with $\theta = 0.3$. This is because the probability of the MU moving back to the previously visited VLR is high ($\theta = 0.7$) and hence the saving in the update cost is significant. This is especially the case when CMR is small partly because it costs less to update the user's location (since most moves are local) and partly because call arrivals are rare, so the high search cost can be amortized. Figure 8 also shows that in this case ATLA and CATLA can perform much better than TLA, because the VLR-VLR binding cost is less than the VLR-HLR update cost. Also, in this case since there is a high possibility of the MU moving back and forth, the advantage of CATLA over ATLA disappears even at high CMR values. This is conceivable since CATLA involves the VLR-HLR communication cost whenever a switch boundary is crossed and it is possible that the two most recently visited VLRs by the MU are covered by two separate network switches.

Both Figures 7 and 8 are based on the condition that T is 0.4 of U, that is, the VLR-HLR communication cost is a large fraction of U (recall that $U = 2T + P_c$). In Figure 9, we show an extreme case in which $T = 0.2$ while all other parameters remain the same as in Figure 7. In this extreme case, since T is small compared to U, the advantage of TLA, ATLA and CATLA over IS-41 is not significant. In fact, the advantage exists only at low CMR values. This is due to two reasons. First, at $\theta = 0.3$, the benefit of the local agent running the two-location algorithm on behalf of the HLR is not pronounced because the local agent will be replaced frequently, thus causing a high location update cost. Second, at $T = 0.2$, the HLR/VLR communication cost is relatively low, so there is little advantage for these algorithms to reduce the VLR-HLR communication cost. That is, whenever the MU moves into a new VLR, the location table of the HLR will be updated and the update cost becomes more affordable with a low T value. The benefit of this more frequent update operation is that it can keep a more precise view of the current VLR. Consequently, when a call arrives, it can serve the call with a lower cost. When CMR is larger than 0.5, the saving in update costs in ATLA and CATLA has a price. That is, ATLA and CATLA maintain a less precise state information than IS-41 and TLA, e.g., when the MU moves into a new VLR from the local agent, ATLA and CATLA do not update the HLR database, and thus can enter an inconsistent state more often than TLA. As a result, when calls do arrive frequently, ATLA and CATLA must pay a higher search cost.

The classic tradeoff between the update and search costs is affected significantly by the MU's CMR. Figure 9 shows that at low CMR values, ATLA and CATLA perform better than TLA but the converse is true at high CMR values. In fact, Figure 9 also shows that at relatively large CMR values, the basic IS-41 HLR/VLR scheme performs the best among all. The same physical interpretation regarding the tradeoff between the search and update costs applies. This, however, occurs only under extreme conditions when the probability of going back and forth is low ($\theta = 0.3$) and the VLR-HLR communication cost ($T = 0.2U$) is a small fraction of U. For most PCS networks, we do expect that the VLR-HLR communication cost accounts for a large percentage of U (as in Figure 8), and therefore our proposed ATLA and CATLA schemes will outperform both TLA and IS-41. An important conclusion from our result is that at low CMR values, ATLA and CATLA always perform better than TLA.

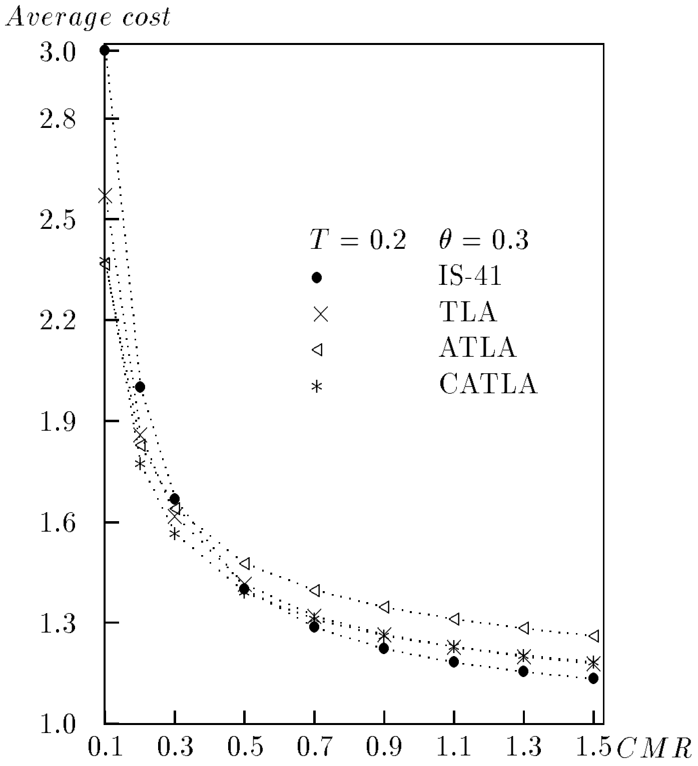

Figure 9: Comparison of IS-41, TLA, ATLA and CATLA with T=0.2 and θ=0.3.

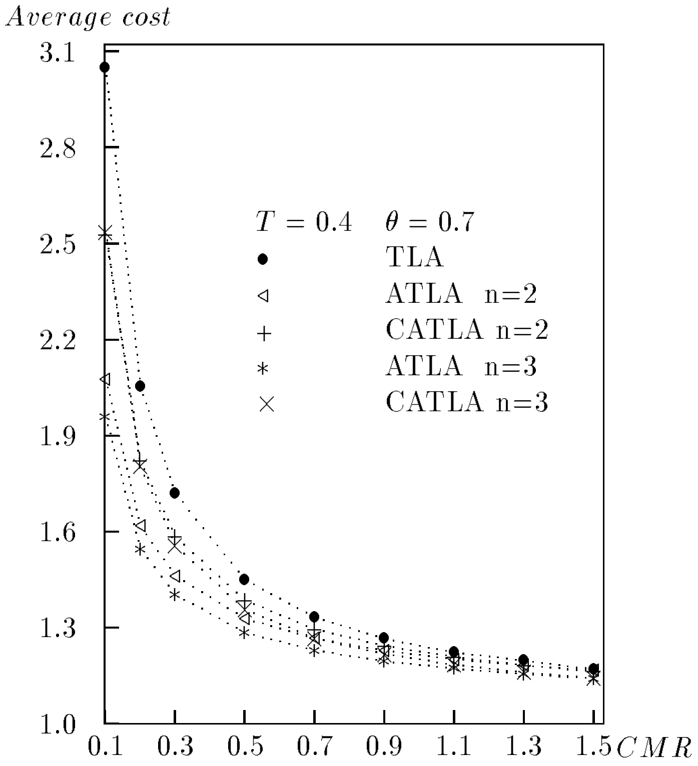

Figure 10: The Effect of n-Layer Network Structure on TLA, ATLA and CATLA.

4.5 Effect of Network Structure

Below we discuss the effect of network structures on the performance of agent-based forwarding strategies. Again suppose that the network structure is the hexagonal coverage model. Each STP is like a n-layer hierarchical structure covering $3n^2 - 3n + 1$ VLRs, where n can be either 2 or 3. It can be shown that the probability of the MU moving within the same STP (i.e., the γ parameter in Table 6) is equal to 0.57 for n=2 and 0.74 for n=3 [2]. Furthermore, as n becomes larger, the average VLR-VLR communication cost (i.e., τ) becomes lower relative to the VLR-HLR communication cost (i.e., T) since most MU movements will be likely to be under the same STP, thus lowering the binding cost in the ATLA and CATLA schemes. The price to pay is to use more sophisticated STP switches to cover more VLRs under the same switch. The switch structure will affect TLA to a smaller extent because the second search attempt under TLA must always involve a VLR-HLR communication cost.

Below we consider a case when $T = 0.4$ and $\theta = 0.7$ as in Figure 8. For $n = 2$, we again have

$\tau/T = 0.5$ and $\tau'/T = 0.125$ as derived earlier. For $n = 3$, however, the value of τ is decreased due to the fact that a STP now covers more VLRs. To study the effect of n, let us fix $\tau'/T = 0.125$ since any two VLRs under the same STP will have the same VLR-VLR communication cost regardless of the number of VLRs under a STP. Therefore, based on Equation 10, τ/T is 0.36 when n=3.

Figure 10 shows the effect of n by comparing our proposed agent-based forwarding schemes against TLA for $n = 2$ and $n = 3$ for the case when $T = 0.4$ and $\theta = 0.7$. We see that as n increases, the performance gain of our proposed agent-based forwarding schemes over TLA becomes more and more pronounced as a result of a lower and lower VLR-VLR communication cost relative to the VLR-HLR communication cost. This means that our proposed schemes are especially attractive for network structures using sophisticated STPs that can cover many VLRs under one network switch.

5 Conclusion and Future Work

In this paper, we proposed the concept of of agent-based forwarding to reduce the location management cost in mobile networks. In particular, we developed the ATLA and CATLA schemes which extend and improve over the TLA scheme. Markov models were used to analyze the performance characteristics of these algorithms. The exact condition under which one scheme is superior to the others and by how much can be assessed by using our Markov models. Our performance analysis data showed that our proposed schemes consistently outperform the basic IS-41 and TLA schemes for the same condition, especially when the CMR value of the MU is low. The advantage of our approach is especially pronounced for a hierarchically structured network in which the VLR-VLR communication cost is much lower than the VLR-HLR communication cost by means of sophisticated STPs which can cover many VLRs under one switch.

Some future research areas related to this paper include (a) investigating the possibility of combining the agent-based forwarding scheme with other approaches such as caching to further reduce the network location management cost for a wider range of parameter values, e.g., covering both high and low CMR values; (b) designing "service handoff" management algorithms in mobile systems and applying similar modeling techniques to assess their performance characteristics.

References

[1] G. Cho, L. F. Marshall, "An Efficient Location and Routing Scheme for Mobile Computing Environments," *IEEE Journal on selected areas in communications,* Vol. 13, No. 5, June 1995, pp. 868-879.

[2] I.R. Chen, T.M. Chen and C. Lee, "Performance evaluation of forwarding strategies for location management in mobile networks," *The Computer Journal*, Vol. 41, No. 4, Aug. 1998, pp. 243-253.

[3] EIA/TIA, *Cellular Radio Telecommunication Inter system Operations,* Technical Report IS-41 (Revision B), EIA/TIA, July 1991.

[4] B. Jabbari,G. Golombo, A. Nakajima, and J. Kulkarni. "Network Issues for Wireless communications," *IEEE Communication Magazine,* Jan. 1995.

[5] R. Jain, Y.B. Lin, C. Lo and S. Mohan, "A caching strategy to reduce network impacts of PCS," *IEEE Journal on Selected Areas in Communications,* Vol. 12, No. 8, Oct. 1994, pp. 1434-1444.

[6] R. Jain, Y.B. Lin, C. Lo and S. Mohan, "A forwarding strategy to reduce network impacts of PCS," *14th Annual Joint Conference of the IEEE Computer and Communications Societies, (IEEE INFOCOM '95),* 1995, Vol. 2, pp. 481-489.

[7] L. Kleinrock, *Queuing Systems, Vol. 1: Theory*, John Wiley and Sons, 1975.

[8] Y.B. Lin, "Reducing location update cost in a PCS network," *IEEE/ACM Transactions on Networking,* Vol. 5, No. 1, Feb. 1997, pp. 25-33.

[9] Phone Lin and Wen-Nung Tsai, "A Simulation Study of Multiple-Location Scheme for PCS Mobility Management," *3rd Workshop on Mobile Computing*, Taiwan, 1997, pp. 37-45.

[10] M. Mouly and M.B. Pautet, *The GSM System for Mobile Communications*, 49 rue Louise Bruneau, Palaiseau, France, 1992.

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

[12] S. Tabbane, "An Alternative Strategy for Location Tracking," *IEEE Journal on Selected Area in Communication,* Vol. 13, No. 5, June 1995.