Tolerating Visitor Location Register Failures in Mobile Environments*

Saad Biaz†   Nitin H. Vaidya

Department of Computer Science, Texas A&M University
College Station, TX 77843-3112, USA
email addresses : {saadb,vaidya}@cs.tamu.edu

Abstract

For mobile users who move frequently but receive relatively rare calls, a forwarding scheme has been shown to outperform the normal IS-41 location management scheme. But the forwarding scheme is more vulnerable to failure of intermediate Visitor Location Registers (VLRs) than the IS-41 scheme. We propose two simple variations to the forwarding scheme to address the fault tolerance weakness. One is based on the idea of maintaining two paths from the home location server to the last VLR. The second scheme is based on the knowledge of the neighbors of the faulty VLR. We evaluate and compare the performance of these location management schemes.

1. Introduction

In Personal Communications Services (PCS), a user is able to receive calls at any location in the PCS coverage area. To provide this "anytime anywhere" service, provisions must be made to be able to *locate* a mobile user (or mobile terminal) whenever a call is to be delivered. This is achieved using an appropriate *location management* strategy.

A simple location management strategy would use a fixed database to store the current location of a mobile terminal (MT). This approach is used in the North American standard IS-41 [1]. As elaborated later in the paper, the IS-41 location management strategy is based on a two-tier system consisting of a Home Location Register (HLR) and Visitor Location Registers (VLR). Each mobile terminal (MT) is associated with a unique HLR – identity of the appropriate HLR is determined based on the identifier of the MT. Each VLR serves a predefined area. In IS-41, whenever a

Mobile Terminal (MT) enters a new area served by a different VLR, the HLR is updated with the new VLR address, so that future calls to the mobile terminal can be correctly delivered. In this scheme, a considerable signaling traffic must be sustained to keep the HLR database updated with the current mobile terminal location. The rapid growth in personal communications services incurs increasing loads on the databases and the network signaling resources. Simple data management strategies, such as IS-41, would not handle these loads efficiently [13, 12]. Therefore, several approaches have been investigated to try to reduce network loads by exploiting specific patterns of mobility and call arrival [11, 9, 1, 6, 2].

This paper considers a "forwarding" strategy proposed by Jain and Lin [11]. This strategy is used for users who move frequently, but receive calls relatively infrequently. As elaborated later, for such users, the *forwarding* scheme avoids the update of the Home Location Register (HLR) by setting a pointer from the previous VLR to the new VLR. This strategy reduces the load on the signaling network between the VLR and the HLR, and avoids HLR database update. However, this scheme is more vulnerable to failures, in comparison to IS-41. In IS-41, success of call delivery requires the HLR and the callee's current VLR to be failure-free. In the forwarding scheme, success of call delivery also requires intermediate VLRs (maintaining forwarding pointers) to be failure-free. In this paper, we present two schemes to tolerate the failure of the VLRs, when using forwarding pointers, and compare their performance. We assume that the callee's HLR is failure-free. Our schemes are designed to tolerate only VLR failures. Schemes for recovery from HLR failure could be used in conjunction with the proposed schemes (e.g., [5]).

This paper is organized as follows: Related work is summarized in Section 2. Section 3 presents an architecture of the Personal Communications Services. Section 4 describes IS-41 location management, the simple forwarding scheme, and then the proposed fault tolerance schemes. Section 5 presents performance analysis. Section 6 presents some numerical results, and conclusions are presented in Section 7.

2. Related Work

There is limited work on the issue of fault tolerance in mobile systems. The most relevant work to this paper is by Lin [8] in which he studies HLR database restoration after a database crash. His approach is based on periodic checkpointing of location databases. Lin derives the optimal interval for checkpointing. A recent paper by Chang et al.[5] presents the standard GSM database failure recovery and proposes a *VLR identification algorithm* which identifies a superset of VLRs to be contacted by an HLR after a failure. The fault tolerance schemes presented in our paper are related to the work on fault tolerant linked lists (for instance, [7]). However, we investigate this problem in the context of location management of mobile hosts. In [15], Rangarajan and Dahbura consider an architecture where each base station maintains the location directory of all mobiles in the network. They present a fault-tolerant protocol to maintain this directory despite base station failures and mobile disconnections. No work, to our knowledge, considers delivery of calls despite a VLR failure.

3. The PCS architecture

The PCS architecture consists of two networks [1] : the Public Switched Telephone Network (PSTN) and a signaling network. The PSTN is the traditional telephone system carrying voice, while the signaling network, meant for management purposes, uses the SS7 (Signaling System no 7 [14]). For location management, the signaling network carries messages for two purposes:

- to track the location of the mobile terminals (registration/deregistration), and

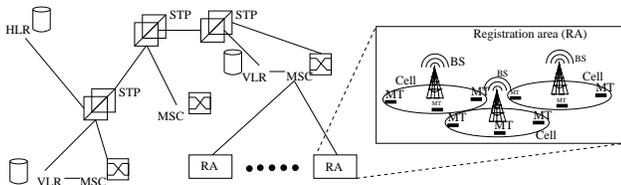- to provide information to the PSTN switches to establish a circuit between a caller and a mobile callee.

Figure 1. Simplified PCS architecture

Figure 1 gives a schematic view of the PCS architecture [1]. The mobile terminals (MT) get access to the PSTN through base stations using wireless links. The area covered by a base station is called a *cell*. A set of geographically close cells defines a *Registration Area (RA)*. All mobile terminals roaming in a registration area have a record in a database

called a *Visitor Location Register (VLR)*. A VLR is responsible for a group of RAs. Each mobile terminal is registered permanently with a *Home Location Register (HLR)*. The HLR keeps the user profile and the information needed to locate the mobile terminal. The HLR and the VLRs communicate through the *SS7 signaling network* to keep track of the mobile terminal position. Messages on the SS7 network are routed through the *Signal Transfer Points (STP)* which are installed in pairs for reliability purpose. The *Mobile Switching Center (MSC)* provides normal switching functions and plays an active role in registration and call delivery. Each VLR can be associated with one or more MSCs. For simplicity, we assume that each VLR is associated with one MSC. The MSC and the VLR can be connected through the STP (SS7 signaling network) routers or with a direct X25 link. We omit some details in this architecture which are beyond the scope of this paper (please refer to [1, 10] for a good description).

4. Location Management Schemes

Any location management scheme must provide two basic operations :

- MOVE : to register the new location when a mobile terminal enters a new RA

- FIND : to determine the current location for a given mobile terminal (to deliver a call)

In this section, we describe the IS-41 location management strategy, the forwarding scheme by Jain and Lin [11], and two ways of incorporating fault tolerance in the forwarding scheme.

4.1. The IS-41 location management scheme

In IS-41, whenever a mobile terminal (MT) enters an RA covered by a different visitor location register, the HLR is updated with the address of the new VLR (referred hereafter as a *pointer*). When a call is issued to an MT, the MT's HLR is contacted to obtain the address of the VLR that covers the MT's current location, and the call is delivered. When a host moves from one registration area (RA) to another, there are two possibilities:

- The MT moves to another RA covered by its current VLR. The MT's HLR need not be updated. In Figure 2(a), an MT moves from registration area $RA1$ to $RA2$, where both $RA1$ and $RA2$ are covered by VLR C. The MT's record at VLR C is updated when the move occurs. However, the MT's HLR is not updated.

- The MT moves to an RA covered by another VLR. The MT's HLR is updated to point to the new VLR. For instance, in Figure 2(b), the mobile terminal moves from
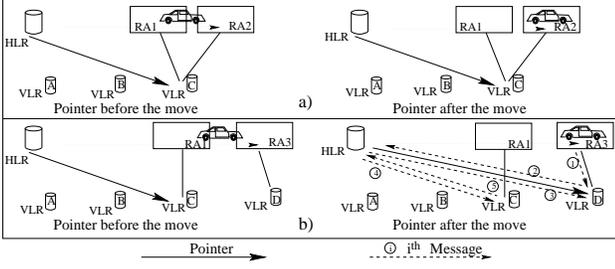
Figure 2. Movement across RAs with IS-41

the registration area $RA1$ to registration area $RA3$, which are covered by the VLRs C and D, respectively. When the move occurs, VLR D is informed of the MT's arrival (see message 1 in Figure 2(b) – broken arrows denote messages, whereas solid arrows denote pointers). VLR D then sends a registration request to the MT's HLR (message 2). HLR updates its record for the MT, to point to VLR D, and sends an acknowledgement to VLR D (message 3), and a cancellation message to VLR C (message 4). VLR C then sends an acknowledgement (for the cancellation) to the HLR (message 5).

When an MT enters an RA covered by a new VLR, four messages involve the HLR. The load can be significant on the signaling network. This MOVE operation also requires a database update at the HLR. These operations are wasteful if many moves occur without any call being made to the MT.

Now, we describe call delivery in IS-41. When a call is issued for an MT, a FIND operation is invoked to locate the MT. Assume that MT a is calling MT b. Let V_a and V_b denote the VLRs covering current locations of a and b, respectively, and let M_a and M_b be the mobile switching centers (MSCs) associated with V_a and V_b. Let H_b be the HLR for b. The first step is to determine the VLR V_b covering currently the MT b. Second, the MSC associated with VLR V_b will deliver a *Temporary Local Directory Number* (TLDN) to HLR H_b. This $TLDN$ will be used to set the circuit between caller and callee on the PSTN network[1].

When MT a dials the call, the call will reach the mobile switching center M_a. M_a then queries MT a's current visitor location register V_a. There are two possibilities:

- VLR V_a may determine that MT b is in its coverage (i.e., $V_a = V_b$). In this case, MT b's HLR is not needed to locate b's current location.

- b is not under V_a's coverage. In this case, the following steps are performed. The mobile switching center M_a queries b's HLR H_b. Then H_b looks up its record for MT b, and determines that b is presently under the coverage of VLR V_b. H_b then sends a call request to V_b.

Thus, the call request is delivered to the appropriate VLR (V_b). This VLR V_b then forwards the call request to its associated mobile switching center M_b. The MSC M_b provides a *TLDN* used to set up the call over the PSTN network[1].

In the first of above two cases, FIND operation is not needed as both MTs are covered by the same VLR. In this case, the location management scheme used is not relevant. Therefore, in further discussion and analysis, we consider only the case where $V_a \neq V_b$.

4.2. Forwarding Strategy

The forwarding strategy, proposed by Jain and Lin [11], modifies the IS-41 scheme such that it does not involve the HLR at each MOVE operation, even if the MT enters a new RA covered by a new VLR. Instead, at each move involving a new VLR, a pointer is established from the old VLR to the new VLR. The forwarding strategy is intended to reduce HLR database updates. However, when a mobile is called, a chain of VLRs must be queried before reaching the current VLR. To bound the call set-up time, the chain length is limited to be less than some value K. Let us assume, in the following examples, that K is set to 3. Note that the chain length is the number of hops from the *first VLR* to the current VLR covering an MT. This definition of the chain length is convenient because the overhead of the forwarding scheme is the cost to traverse this chain. The cost to reach the *first VLR* is the same as the cost of a FIND operation in the IS-41 scheme.

Figure 3(a) shows a mobile terminal that has been in RAs under the coverage of VLRs A and B, in that order. As shown in Figure 3(b), when the MT moves from the RA covered by VLR B to another RA covered by VLR C, a pointer (for this mobile MT) will be set at VLR B pointing to VLR C. This MOVE operation does not involve the HLR, because the length of forwarding pointer chain from the HLR to the new VLR is 2. This length is smaller than the threshold value $K = 3$. (Recall that we measure the length from the first VLR, in this case VLR A). Now, as shown in
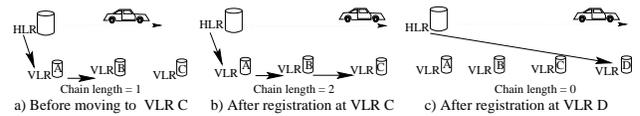

a) Before moving to VLR C   b) After registration at VLR C   c) After registration at VLR D

Figure 3. Move operation using forwarding

Figure 3(c), the MT moves to an RA covered by VLR D. In this case, the pointer chain length (from VLR A to the new VLR D) would be equal to threshold $K = 3$. Therefore, the home location register is informed of the new VLR, and a pointer is set at the HLR to point directly to VLR D [11]. Therefore, the chain length becomes 0.

For the FIND operation, the chain of VLRs will be queried until getting the current VLR covering the mobile terminal. In Figure 4(a), assume that a mobile terminal a is calling mobile terminal b. The MSC serving a will query b's Home Location register HLR_b. The query will then be forwarded through VLRs A, B, and C. After mobile terminal b is found, the forwarding pointer chain is compressed as shown in Figure 4(b). That is, after the FIND operation, the HLR for b is updated. While the forwarding scheme has

a) During a FIND operation   Pointer   Message   b) After a successful FIND operation
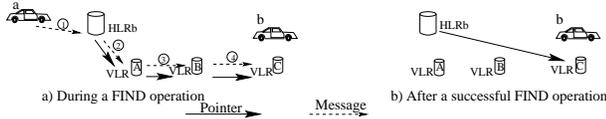
Figure 4. Find operation using forwarding

performance advantages [11], it is more vulnerable to VLR failures as compared to IS-41. For instance, in Figure 4(a), to be able to deliver the call, VLRs A, B and C must be failure-free. With IS-41, only VLR C would need to be failure-free. In the following, we consider two schemes to tolerate a visitor location register failure and compare their performance.

4.3. Scheme 1: Bypass Forwarding Strategy

When a VLR on the forwarding pointer chain fails, scheme 1 attempts to "bypass" the faulty VLR by forwarding a request to all its "neighbors". A VLR X is a neighbor of VLR Y if a mobile terminal may move from a registration area covered by VLR X to another registration area covered by VLR Y. Let $neighbors(X)$ denote the set of neighbors of VLR X. The MOVE operation, with this scheme, is identical to that in forwarding scheme.

Now, consider the FIND operation. If no intermediate VLR on the forwarding chain fails, the procedure is same as FIND operation for Jain and Lin's forwarding scheme. Now, consider the case where a VLR fails. For instance,
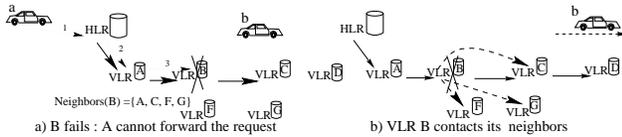

Neighbors(B)={A, C, F, G}
a) B fails : A cannot forward the request   b) VLR B contacts its neighbors

Figure 5. Find operation with a VLR failure

assume that VLR B in Figure 5 failed after the forwarding pointer from B to C was set. Assume that VLR B has come back up when the FIND operation is being performed. In this case, when a FIND operation is performed, the request is forwarded along the forwarding chain, until it reaches VLR B. VLR B cannot forward the request because it failed after the forwarding pointer was set (the

pointer is lost). Therefore, VLR B forwards the request to all its neighbor VLRs (except VLR A). VLR B has this information available in $neighbors(B)$. Let us assume that $neighbors(B) = \{A, C, F, G\}$. Thus, the request will be forwarded to C, F and G. Now, two possibilities can occur:

- The mobile terminal has moved to an area covered by another VLR (say, VLR D in Figure 5(b)): In this case, on receiving the request from B, VLR C will forward the request to VLR D (as C has a forwarding pointer for the MT), and will send a *positive-reply* message to VLR B. VLRs F and G will send a *negative-reply* to B, as they do not have a pointer for the MT.

- The mobile has not moved from the RA covered by VLR B. In this case, none of the VLRs to whom the request was forwarded (C, F and G in our example) will have an entry for the callee MT. Therefore, VLR B will receive *negative-reply* from each of these neighbors. Thus, B can presume that the mobile terminal is in its coverage area.

This scheme will tolerate failures as long as no two consecutive visitor location registers in the forwarding chain fail. Note also that if the mobile moves to a VLR which is not a "usual" neighbor of the current VLR, this scheme will fail.

4.4. Scheme 2: Two-Path Forwarding Strategy

The idea here is to establish two independent paths (if possible) from the MT's HLR to its current VLR. Figure 6 illustrates the idea. In this case, the HLR maintains two pointers for each MT. In Figure 6(a), a mobile terminal is initially in a registration area covered by VLR A. At this time, both pointers at the MT's HLR point to VLR A. When the MT subsequently moves to registration area covered by VLRs B, C and D, the pointers are updated as shown in Figures 6(b), (c) and (d) respectively. Observe that the two pointers at the HLR lead to two paths that do not share any VLR except the MT's current VLR.

To see this in more detail, assume that the current state is as shown in Figure 6(c). Now, the MT enters a new RA covered by VLR D (see Figure 6(d)). At this time, the MT registers with VLR D by passing the addresses of its two previous VLRs, namely, B and C. VLR C then sends messages to VLRs B and C to set their pointers to itself (i.e, VLR D). The resultant pointers are shown in Figure 6(d). We only consider single VLR failure in our work. To find an MT, the HLR will first follow pointers along one path (similar to FIND operation in the basic forwarding scheme by Jain and Lin). If this FIND operation fails due to loss of a pointer (due to VLR failure) along the path, the HLR will start again with the other path. As the two paths do not share an intermediate VLR, single intermediate VLR failure can be tolerated.
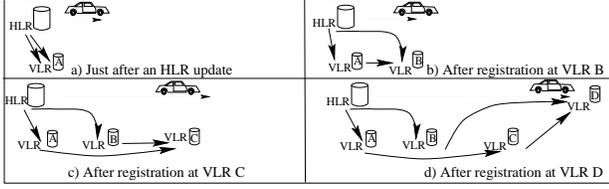
Figure 6. Pointers update for MOVE

Both the paths fail (under single failure assumption) if and only if the current VLR of the MT had failed. Thus, if on both paths, the same VLR, say VLR X, is unable to forward the request, then it can be concluded that the MT is in the coverage area of X. Thus, X can complete the call.

Similar to the previous schemes, the forwarding pointer chains are compressed when the first chain followed by the HLR becomes of length K. Note that the length of each chain will be approximately half that when using scheme 1. Thus, in scheme 2, the forwarding chain will be compressed half as frequently as scheme 1. Also, the chain length grows at half the rate of scheme 1. Although these factors may reduce the failure-free overhead, another factor contributes to an increase in failure-free overhead. Specifically, on each MOVE, scheme 2 requires twice the messages required by scheme 1. In the next section, we will compare the total *cost* of using the proposed fault-tolerance schemes.

5. Performance Analysis

In this section, we evaluate the "cost" of using each fault tolerant schemes. The *cost* could be in terms of bandwidth usage, time required, amount of money, etc. The actual meaning of the term *cost* does not affect our cost analysis. This approach is similar to that used in previous work [11, 9, 6]. We will use the following notations in this section.

- K : length of a forwarding chain can at most be $K - 1$

- $1/\lambda_m$ is the expected residence time of a user in an RA. The residence time is exponentially distributed

- λ_c is the call arrival rate (Poisson process)

- $p = \lambda_c/\lambda_m$ is the call-to-mobility ratio

- λ_v is the failure rate of a VLR (Poisson process)

- M_{IS41} is the cost of a MOVE operation for IS-41

- F_{IS41} is the cost of a FIND operation for IS-41 scheme

- S is the cost of setting a pointer between VLRs

- T is the cost of traversing a pointer

- M_i is the expected cost of all MOVE operations between two consecutive calls when using scheme i

- F_i is the expected cost for one FIND when using scheme i, in the absence of failures.

- P_f is the probability that a FIND operation encounters a faulty VLR.

- F_i^f is the expected cost for one FIND operation which encounters a faulty VLR using scheme i.

- C_i is the expected cost per call when using scheme i

Our performance metric for scheme i is the *expected cost per call*, denoted as C_i. C_i consists of two components:

- Expected cost of all MOVE operations before the call, since the previous call. This cost is denoted as M_i.

- Expected cost of a FIND operation. Special steps need to be taken if a forwarding pointer is corrupted due to a VLR failure. Thus, there are two possibilities:

  - The FIND operation encounters a VLR failure. The probability that this event will occur is denoted as P_f. In this case, the cost is denoted as F_i^f.
  - The FIND operation does not encounter a VLR failure (with probability $1 - P_f$). In this case, the cost of a FIND operation is denoted as F_i.

  Thus, the expected cost of a FIND operation is given by $(1 - P_f) F_i + P_f F_i^f$.

The expected cost per call C_i is obtained by adding the above two components, as $C_i = M_i + (1 - P_f) F_i + P_f F_i^f$. In the next two subsections, we outline expressions for M_i, P_f, F_i and F_i^f for the two fault tolerant schemes.

5.1. Expected cost per call C_1 for scheme 1

For brevity, where possible, we will use previously known results without proof.

5.1.1 Expected cost of MOVE operations per call

The expression for M_1 can be obtained using the analysis presented by Jain and Lin [11]. The analysis presented by Jain and Lin does not consider failures. However, since MT movements are not influenced by failures, we can use their results to obtain M_1.

$p = \lambda_c/\lambda_m$ is the call-to-mobility ratio. It is easy to show that $1/p$ is the expected number of moves between two calls to a mobile terminal. The forwarding scheme nominally sets a forwarding pointer on each MOVE (the cost of setting a pointer is S). However, on every K-th

MOVE, instead of setting a forwarding pointer, the HLR is updated, i.e., pointer chain is *compressed*. The expected number of compressions between two calls can be calculated as $\frac{1}{(1+p)^K-1}$ [11, 4]. Each chain compression requires an HLR update, incurring the same cost as a move operation in IS-41 (this cost is denoted as M_{IS41}). Thus, the cost S of setting the pointer is incurred, on average, for $\left(\frac{1}{p} - \frac{1}{(1+p)^K-1}\right)$ MOVEs per call, and the cost M_{IS41} of chain compression is incurred for $\frac{1}{(1+p)^K-1}$ MOVEs per call. Thus, the total cost M_1 per call using scheme 1 is

$$M_1 = S\left(\frac{1}{p} - \frac{1}{(1+p)^K-1}\right) + \frac{M_{IS41}}{(1+p)^K-1}$$

5.1.2 Expected cost of a FIND operation

In this section, we evaluate P_f, F_1 and F_1^f for Scheme 1. While the analysis presented by Jain and Lin [11] was useful to determine M_1, it is not useful to obtain the cost of a FIND operation using scheme 1. To derive F_1 and F_1^f, we model chain length for a mobile terminal with the Markov model in Figure 7. In the Markov model, the state is i or i'
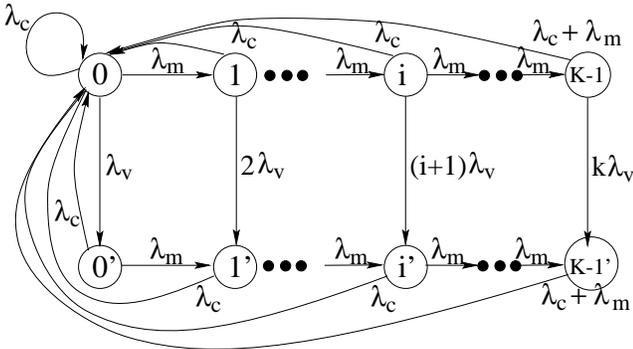

Figure 7. Markov model for scheme 1

$(i = 0, \cdots K - 1)$ when the forwarding chain is of length i. State i implies that a FIND operation performed (due to a call) while in that state will *not* encounter failure. On the other hand, a FIND operation performed when in state i' *will* encounter a failure. Maximum chain length can only be $K - 1$. The rate of transition from state i to $i + 1$ $(0 \le i \le K - 2)$ is λ_m (note that λ_m is the rate at which MOVEs occur). The rate of transition from state i to state i' is $(i + 1)\lambda_v$, as failure of any of the $(i + 1)$ VLRs on the chain will be encountered by a future FIND operation. If a call with rate λ_c occurs in any state, the new state becomes state 0 (due to chain compression). We make certain assumptions regarding failures, that are realistic in telecommunication domain. We assume that the maximum possible failure rate in any state, i.e., $K\lambda_v$, is sufficiently small that the probability of more than one failure occurring *between*

two calls is negligible. Therefore, no new failure may occur in state i' $(i = 0, \cdots K - 1)$. We also assume that downtime of a VLR following its failure is negligible. When a VLR comes back up after failure, its previous database is lost.

Let P_i denote the steady state probability of being in state i $(i = 0, \cdots, K - 1)$, and let Q_i denote the steady state probability of being in state i' $(i = 0, \cdots, K - 1)$. Details for solving the Markov Model can be found in our report [3]. If a call occurs in state i, then it is completed without encountering a corrupted pointer (due to VLR failure). Whereas, if a call occurs in state i', then it encounters a corrupted pointer. Thus, the probability that FIND operation will encounter a corrupted pointer (denoted as P_f) is given by

$$P_f = \frac{\frac{\lambda_c}{\lambda_c+\lambda_m}\sum_{i=0}^{K-1}Q_i}{\frac{\lambda_c}{\lambda_c+\lambda_m}\sum_{i=0}^{K-1}Q_i + \sum_{i=0}^{K-1}\left(\frac{\lambda_c}{\lambda_c+\lambda_m+(i+1).\lambda_v}\right).P_i}$$

Now, we evaluate cost F_1. This cost includes the cost to query the HLR and then to traverse the pointer chain. The first component corresponds to a FIND operation for IS-41. The second component depends on the expected length of the chain given that no VLR failure occurred in this chain. Thus, F_1 is obtained as

$$F_1 = F_{IS41} + T\sum_{i=0}^{K-1}i\left(\frac{P_i}{\sum_{i=0}^{K-1}P_i}\right)$$

where T is the cost to traverse a pointer, F_{IS41} is the cost of a FIND operation in IS-41 and $\left(\frac{P_i}{\sum_{i=0}^{K-1}P_i}\right)$ is the probability that the chain is of length i given that it contains no faulty VLR.

Now, we consider the expected cost F_1^f of a FIND operation which encounters a corrupted pointer due to a failed VLR. As noted before, we assume that at most one VLR visited by the MT since the last call may fail at any time. F_1^f has three components :

- The cost to query the HLR – this cost is equal to F_{IS41}.

- The cost to traverse the forwarding pointer chain, given that a VLR on the chain has failed. This cost is obtained as product of T and expected chain length given that a VLR failure has occurred on the chain. Thus, this cost is obtained as $T\left(\sum_{i=0}^{K-1}i\left(\frac{Q_i}{\sum_{j=0}^{K-1}Q_j}\right)\right)$.

- The cost to "bypass" the faulty VLR. If we assume that each VLR covers an hexagonal region, then we can approximate the cost to "bypass" a VLR with $5.T$ (request forwarded to five neighbors). The assumption of a hexagonal region is reasonably accurate since VLRs cover large regions [1].

Thus, we have

$$F_1^f = F_{IS41} + 5.T + T \left(\sum_{i=0}^{K-1} i \cdot \left(\frac{Q_i}{\sum_{j=0}^{K-1} Q_j} \right) \right)$$

Now, using the expressions derived above, the expected cost per call C_1 for scheme 1 can be obtained as $C_1 = M_1 + (1 - P_f).F_1 + P_f.F_1^f$

5.2. Expected cost per call C_2 for scheme 2

5.2.1 Expected cost of MOVE operations per call

The key difference here (from scheme 1) is that the HLR is updated only every $2K$ MOVEs, as length of each of the two chains maintained in this scheme increases only on alternate MOVEs. Thus, similar to scheme 1, cost M_2 is obtained as :

$$M_2 = 2S \left(\frac{1}{p} - \frac{1}{(1+p)^{2K} - 1} \right) + \frac{M_{IS41}}{(1+p)^{2K} - 1}$$

5.2.2 Expected cost of a FIND operation

Recall that in scheme 2, the HLR first follows one forwarding pointer chain. The second chain is used only in the event of a failure. We use the Markov model in Figure 8 to evaluate scheme 2. This model is slightly different because the length of the chain followed first by the HLR increases only after every *two* moves: the states ia and ib denote a chain of length i with no faulty VLR ($i = 0, \cdots K - 1$). Similarly, states ia' and ib' denote a chain of length i with a faulty VLR.
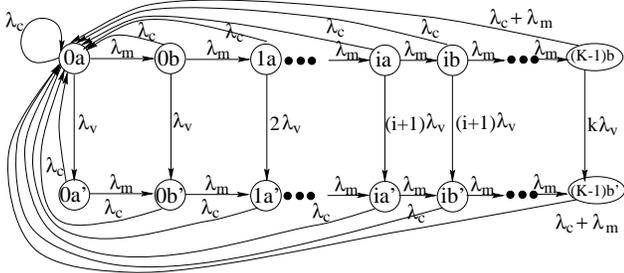

Figure 8. Markov model for scheme 2

For lack of space, we omit here the derivation of the expected cost for the FIND operation. Details can be found in [3].

6. Numerical Results

We are interested in the ratio $\frac{C_1}{C_2}$ to determine the conditions under which one scheme outperforms the other. The costs M_{IS41} and F_{IS41} are comparable. Therefore, we normalize them to be 1. Costs S and T are comparable. For the numerical plots, we assume $S = T$. Because the operation of setting or traversing a pointer involves less work than IS-41 MOVE and FIND, the costs S and T are fractions of M_{IS41} and F_{IS41}. For the numerical plots, we consider two values $S = T = \frac{1}{2}$ and $\frac{1}{32}$. Since VLR failures are rare, we set $\lambda_v = 10^{-6}$. Note that, at realistic failure rates, the failure-free cost of a scheme dominates its total cost.

Figures 9 and 10 plot the ratio C_1/C_2 versus call-to-mobility ratio p for two different values of λ_c (0.001 and 100), assuming that $K = 6$. Figures 9 and 10 assume $S = 1/32$ and $S = 1/2$, respectively. In each of these figures, observe that it is hard to distinguish between the curves for the two values of λ_c. This shows that, for a given call-to-mobility ratio $p = \lambda_c/\lambda_m$, the C_1/C_2 ratio is not very sensitive to the chosen λ_c. Observe that the curves
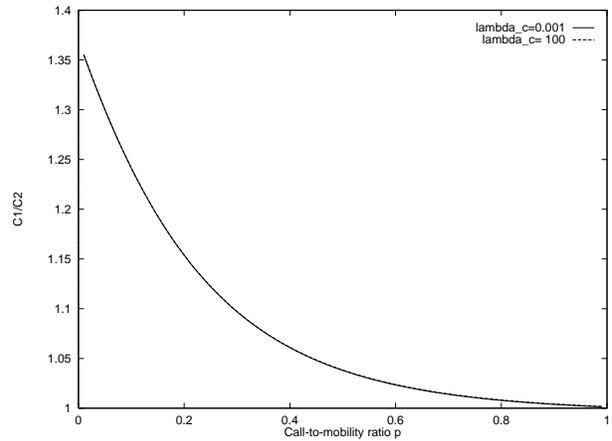

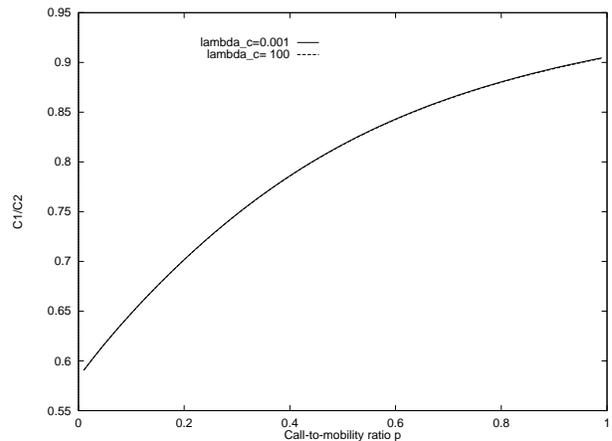**Figure 9. Ratio $\frac{C_1}{C_2}$ with $K = 6$ and $S = \frac{1}{32}$**

Figure 10. Ratio $\frac{C_1}{C_2}$ with $K = 6$ and $S = \frac{1}{2}$

in Figure 9 and Figure 10 have different shapes. We now

explain why this phenomenon occurs.

For $K = 6$ and $S = \frac{1}{32}$ (Figure 9), the ratio $\frac{C_1}{C_2}$ is greater than 1, and it decreases with increasing call-to-mobility ratio p. When S is small relative the cost of HLR updates is the dominant component of the overall cost. Let us consider two values of p, $p_1 = 0.01$ and $p_2 = 0.5$. With p_1, we have, on average, 100 moves per call. Thus, on average, scheme 1 will perform $100/K \approx 16$ HLR updates between calls, while scheme 2 will perform half as many. Since the cost of an HLR update is relatively large (when S is small), the overhead of scheme 2 $-$ in setting an extra pointer per move $-$ is negligible in comparison of the cost saving due to reduced HLR updates. Therefore, scheme 2 performs significantly better than scheme 1 when $p = 0.01$. Now, as p becomes larger, the average number of moves between calls decreases (2 moves per call when $p_2 = 0.5$). Thus, as p increases, it becomes increasingly unlikely that the forwarding chain length will reach the threshold ($K = 6$). Therefore, the number of HLR updates performed due to a FIND operation (when a call arrives) becomes larger, and the number of HLR updates due to chain compression becomes smaller (with increasing p). Note that the number of HLR updates due to FIND operation is identical for both fault-tolerant schemes(as these updates occur when calls occur, and call arrivals are independent of the choice of fault-tolerant scheme). Thus, relative performance of scheme 2 (compared to scheme 1) degrades with increasing p, because the cost of setting two pointers per MOVE becomes an increasing fraction of the cost of scheme 2.

Observe that when p becomes large, the cost of the two schemes becomes identical. This is intuitive – with very large p, many calls occur between moves. Thus, the total cost is determined primarily by the cost of a call, given that no move occurred since the previous call. This cost is independent of the fault-tolerant scheme used, therefore, when $p \to \infty$, the two schemes result in the same cost (i.e., $C_1/C_1 \to 1$).

Now, in Figure 10, with $K = 6$ and $S = \frac{1}{2}$, ratio $\frac{C_1}{C_2}$ is less than 1, and it increases with increasing p. When S is large, the cost of setting a pointer is not negligible compared to the cost of updating the HLR. As scheme 2 sets two pointers on each move, its cost tends to be higher than that of scheme 1 (therefore, $C_1/C_2 < 1$). Now, when p is small, the total cost is dominated by the cost of moves. Therefore, for small p, cost of scheme 2 is approximately twice that of scheme 1 (i.e., $C_1/C_2 \approx 0.5$). Note that the average chain length when using scheme 2 is smaller than that for scheme 1. Therefore, the cost of a FIND operation with scheme 2 is smaller. This saving increases as p increases, therefore, the ratio C_1/C_2 increases with increasing p.

We now study the impact of the threshold K, using the plots in Figures 11 and 12, for $S = 1/32$ and $1/2$, respectively. Observe that, for $S = 1/32$ and $K < 20$, the ratio

C_1/C_2 decreases with increasing p (the reasons are similar to the case of $K = 6$ and $S = \frac{1}{32}$ discussed above). Also, when $S = 1/2$ (Figures 12), the ratio C_1/C_2 increases with p for all values of K considered here (for reasons similar to those discussed above for the case of $K = 6$ and $S = \frac{1}{2}$).
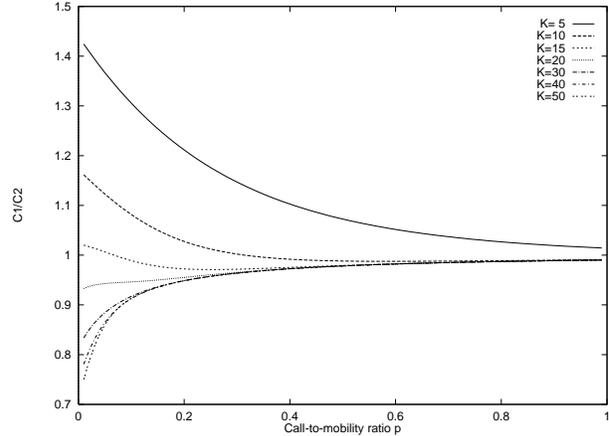

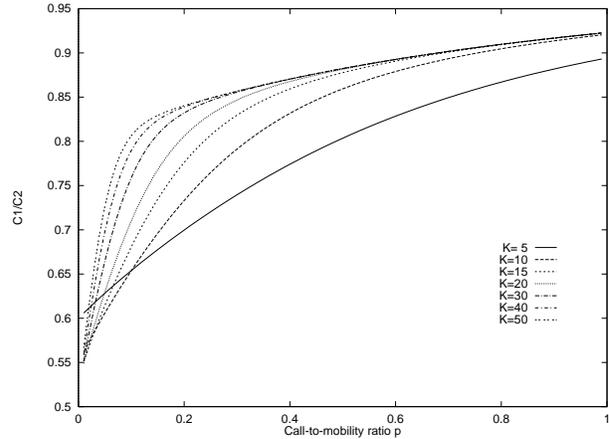**Figure 11. Ratio $\frac{C_1}{C_2}$ with $\lambda_c = 1$ and $S = \frac{1}{32}$**

Figure 12. Ratio $\frac{C_1}{C_2}$ with $\lambda_c = 1$ and $S = \frac{1}{2}$

Now, for $S = 1/32$ and $K \geq 20$ (see Figure 11), the ratio C_1/C_2 is smaller than 1, and it increases with p. This trend is different from that observed previously (when $K = 6$). To explain this trend, let us assume that p is small and K is large. In this case, it is unlikely that the chain length when using either fault tolerant scheme will reach the threshold K. Since move cost for scheme 2 is twice that for scheme 1, and calls are infrequent compared to moves (due to small p), scheme 1 will perform better (i.e., $C_1/C_2 < 1$). As p increases, the reduction in FIND cost due to scheme 2 increases, therefore, C_1/C_2 increases.

7. Conclusions

We have described and analyzed two fault-tolerant variations to the forwarding scheme for location management. Our schemes are able to tolerate single VLR failures. Analytical results show that the relative performance of the two schemes depends mainly on the threshold value K, and the relative value of cost S (when compared to cost M_{IS41}). Each scheme is superior for a range of parameter values, therefore, the actual choice of fault-tolerant scheme depends on the system parameters. However, as the *call-to-mobility ratio p* increases, the two schemes result in the same cost.

8. Acknowledgment

The authors would like to thank the referees for their insightful remarks and constructive criticism.

References

[1] I. F. Akyildiz and J. S. Ho. On location management for personal communications networks. *IEEE Communications Magazine*, pages 138–145, Sept. 1996.

[2] A. Bar-Noy, I. Kessler, and M. Sidi. Mobile users: To update or not to update? *Wireless Networks journal*, 1(2):175–186, 1995.

[3] S. Biaz and N. H. Vaidya. Tolerating location register failures in mobile environments. Technical Report 97-015, Dep. of Comp. Science, Texas A&M, Dec. 1997. Revised version April 1998.

[4] S. Biaz and N. H. Vaidya. Performance analysis of a fault-tolerant scheme for location management of mobile hosts. In *IPPS'98 Workshop FTPDS*, Apr. 1998.

[5] M.-F. Chang, Y.-B. Lin, and S.-C. Su. Improving the fault tolerance of GSM networks. *IEEE Network*, 12(1):58–63, Jan. 1998.

[6] P. Krishna, N. H. Vaidya, and D. K. Pradhan. Static and adaptive location management in mobile wireless networks. *Computer Communications*, Apr. 1996.

[7] Q. Li and S. Vlaovic. Redundant linked list based cache coherence protocol. In *IEEE Workshop of FTPS*, June 1994.

[8] Y. Lin. Failure restoration of mobility databases for personal communication networks. *ACM-Baltzer Wireless Networks*, pages 365–372, 1995.

[9] Y. Lin. Reducing location update cost in a PCS network. *IEEE/ACM Trans. on Networking*, Feb. 1997.

[10] Y. Lin and S. DeVries. PCS network signalling using SS7. *IEEE Personal Communications*, June 1995.

[11] Y. Lin and R. Jain. Performance modeling of an auxiliary user location strategy in a PCS network. *ACM-Baltzer Wireless Networks*, pages 197–210, June 1995.

[12] C. Lo and R. Wolff. Estimated network database transaction volume to support wireless personal data communications applications. In *Proc. Conf. on Communications*, 1993.

[13] K. Meier-Hellstern and E. Alonso. The use of SS7 and GSM to support high density personal communications. In *Int'l Conf. on Communications*, 1992.

[14] A. Modaressi and R. Skoog. Signaling system no. 7. *IEEE Com. Mag.*, July 1990.

[15] S. Rangarajan, K. Ratnam, and A. T. Dahbura. A fault-tolerant protocol for location directory maintenance in mobile networks. In *FTCS-25*, June 1995.