Tolerating 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
E-mail: saadb@cs.tamu.edu
Phone : (409) 845-5007
Fax : (409) 847-8578

Technical Report 97-015

December 5,1997

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.

Key Words: Personal Communications Services – Location Management – Visitor Location Register Failures– IS-41 – Forwarding Scheme

1 Introduction

In Personal Communications Services (PCS), a user is able to receive calls at any location in the PCS coverage area [8, 9]. To provide this "anytime anywhere" service, provisions must be

*Research reported is supported in part by a Fulbright Scholarship, and by Texas Advanced Technology Program grant 009741-052-C.

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 [2]. 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 [17, 16]. Therefore, several approaches have been investigated to try to reduce network loads by exploiting specific patterns of mobility and call arrival [15, 13, 2, 3, 10, 6].

This paper considers a "forwarding" strategy proposed by Jain and Lin [15]. This strategy is used for those 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 also 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 HLR is failure-free – if the HLR is not failure-free, a mobile terminal cannot typically be authenticated, and therefore, services cannot be provided to the mobile terminal.

This paper is organized as follows : Related work is summarized in Section 2. Section 3 presents a broad 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 the discussion and conclusion.

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 [12] 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. The fault tolerance schemes presented here are also related to work on fault tolerant linked lists (for instance, [11, 5]). However, we investigate this problem in the context of location management of mobile hosts.

In [20], 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. In their scheme, all base stations are updated after each mobile moves from one base station to another. The main problem addressed is avoiding old updates from overwriting current information. Alagar et al. [4] consider the problem of base station failures. They present two schemes to tolerate such failures which are based on secondary base stations. The idea is to use replication (optimistic or pessimistic) on the secondary base stations. Pradhan et al. [19] consider recovery from failures of the mobile host. They present different schemes for message logging and checkpointing. Acharya et al. [1] consider the checkpointing of distributed applications. They consider specifically schemes to tolerate the failure of a mobile host or its disconnection. No work, to our knowledge, considers delivery of calls despite a VLR failure.

3 The PCS architecture

The PCS architecture consists of two networks [2] : 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 [18]). 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.

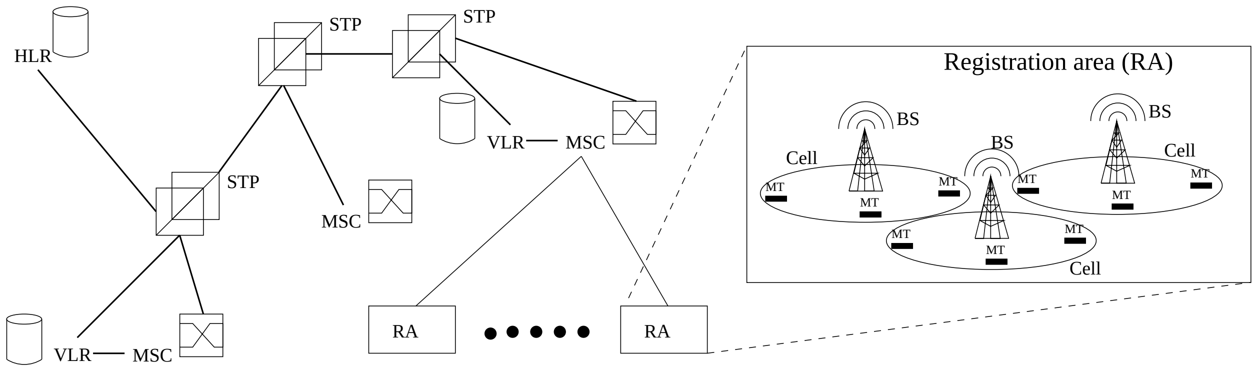


Figure 1: Simplified PCS architecture

Figure 1 gives a schematic view of the PCS architecture [2]. 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 define 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 nature of information stored at the HLR depends on the location management strategy used). 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 [2, 14] 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 [15], and two ways of incorporating fault tolerance in the forwarding scheme. To make the paper readable, we have illustrated the various schemes by means of examples, instead of presenting formal descriptions.

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. 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. We now illustrate the IS-41 scheme with examples. When a host moves from one registration area (RA) to another, there are two possibilities:

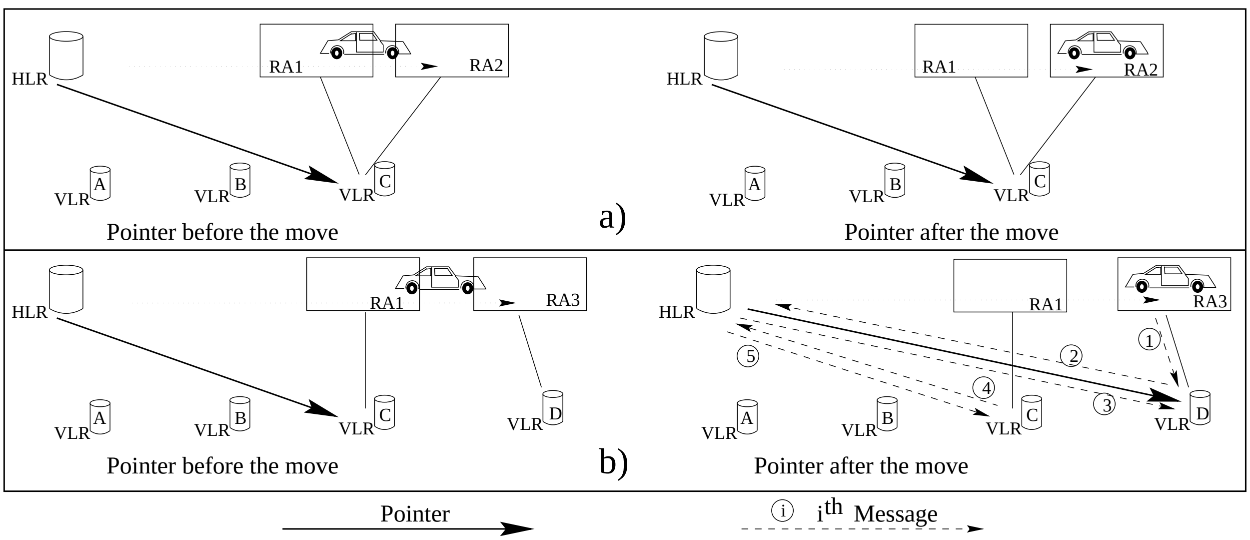


Figure 2: Movement between two RAs when using IS-41

- The MT moves to another RA covered by its current VLR. In this case, the MT's HLR need not be updated. For instance, in Figure 2(a), an MT moves from registration area $RA1$ to $RA2$, where both $RA1$ and $RA2$ are covered by VLR C. In this case, 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. In this case, the MT's HLR is updated to point to the new VLR. For instance, in Figure 2(b), the mobile terminal moves from 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, specially if the HLR is "far away" from the VLRs. 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 the caller and the callee on the PSTN network [2].

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 (called FIND operation). 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 appropriate VLR in both above cases. This VLR V_b then forwards the call request to its associated mobile switching center M_b. The MSC M_b provides a "Temporary Local Directory Number (TLDN)". This TLDN is used to set up the call over the PSTN network [2].

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 [15], 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. With this scheme, 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).

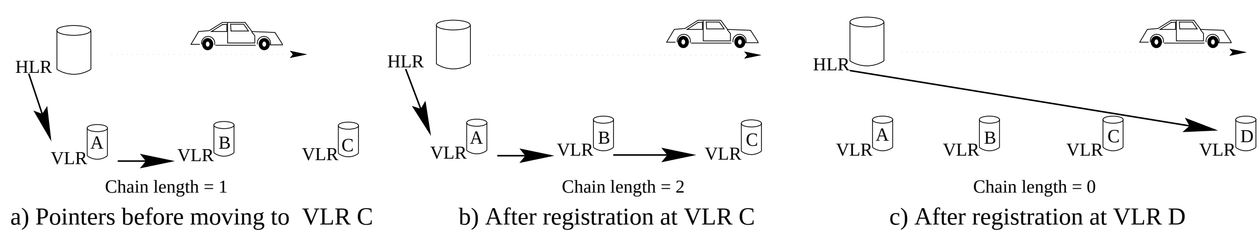


Figure 3: Move operation in the forwarding scheme

Now, as shown in 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 [15]. 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. 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 (see Figure 4(a)) 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.

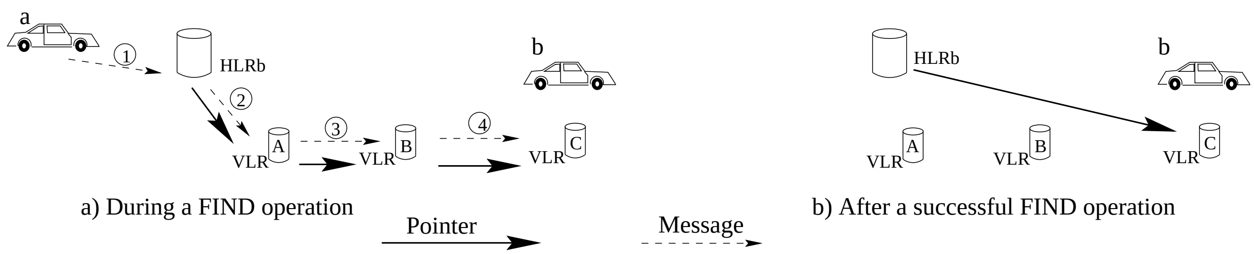


Figure 4: Find operation in the forwarding scheme

While the forwarding scheme has performance advantages [15], 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: Fault-Tolerant 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 Jain and Lin's forwarding scheme in section 4.2.

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, 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 the case where VLR B is still down, the VLR A will not get a positive acknowledgement to its forwarding request. In this case, VLR A will act as a proxy for VLR B. VLR A

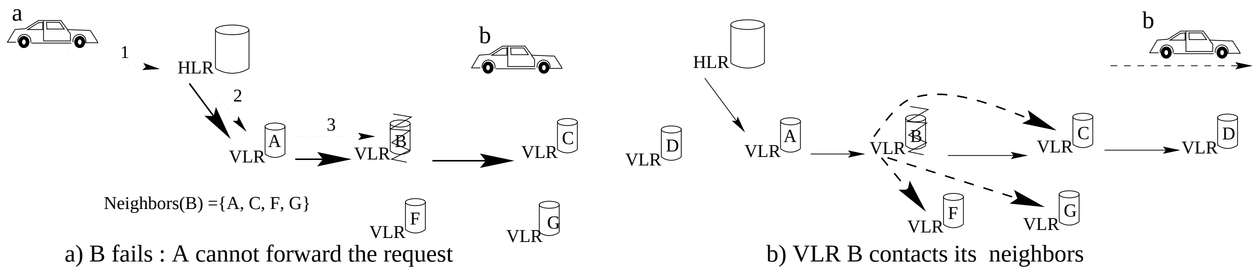


Figure 5: Find operation with a VLR failure

will forward the request to all B's neighbors and also will ask B's MSC to page the mobile. In this, case, the call will be delivered. This variation does not have an impact on our analysis.

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 corrupted or lost). Therefore, VLR B forwards the request to all its neighbor VLRs (except VLR A). VLR B has this information available in its $neighbors(B)$ set. 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 is still in an RA covered by VLR B. In this case, VLR B will receive *negative-reply* from all its neighbors (except A). 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.

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 it is responsible for. As shown 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).

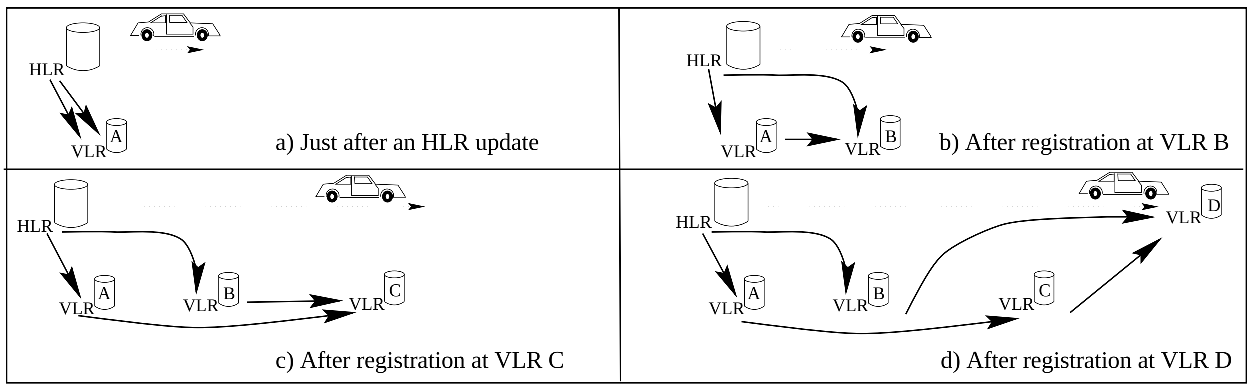


Figure 6: MOVE operation and pointers update

We only consider single VLR failure in our work. Also, as noted before, HLRs are assumed failure-free.

As the HLR has two different pointers leading to the current VLR, 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, intermediate VLR failure can be tolerated.

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.[1]

Note that in this case, length of the two chains may differ by at most 1. As in the case of

[1]Similar to scheme 1, the fault tolerance procedure for scheme 2 can be modified to tolerate a VLR that is still down when the FIND operation is being performed. The discussion here assumed that the VLR failed, lost its database, and has come back up.

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. Although this factor 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 these two 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 [15, 13, 3, 10]. We will use the following notations in this section. The notation may become clearer when we present details of the analysis.

- 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 (call arrival is a Poisson process)
- $p = \lambda_c/\lambda_m$ is the call-to-mobility ratio
- λ_v is the failure rate of a VLR (VLR failures governed by a Poisson process)
- M_{IS41} is the cost of a MOVE operation for IS-41 scheme (described in Section 4.1)
- 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 fault tolerance scheme i ($i = 1$ or 2)
- F_i is the expected cost for one FIND when using scheme i, in the absence of failures.

- F_i^f is the expected cost for one FIND operation using scheme i in presence of a VLR failure
- C_i is the expected cost per call when using scheme i
- P_f is the probability that a FIND operation encounters a faulty VLR. P_f is different for the two schemes.

Our performance metric for scheme i is the *expected cost per call*, denoted as C_i. Cost 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: The FIND operation traverses forwarding pointers. 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 (to be evaluated later in the paper). In this case, the cost of FIND operation 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 derive expressions for M_i, P_f, F_i and F_i^f for the two fault tolerant schemes (i.e., $i = 1, 2$).

5.1 Expected cost per call C_1 for scheme 1

5.1.1 Expected cost of MOVE operations per call using scheme 1

The expression for M_1 can be obtained using the analysis presented by Jain and Lin [15]. 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 $\frac{1}{(1+p)^K-1}$ [15, 7]. Each chain compression requires an HLR update, incurring the same cost as a move operation in IS41 (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 using scheme 1

In this section, we evaluate P_f, F_1 and F_1^f. While the analysis presented by Jain and Lin [15] 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 a chain length for a mobile terminal with a Markov model on Figure 7.

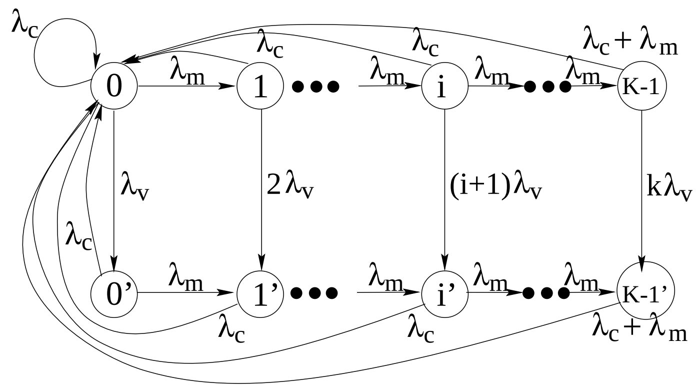


Figure 7: Markov model for scheme 1

In the Markov model, the state is i or i' ($i = 0, \cdots K-1$) 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 $K-1$, because the chain is compressed on the next MOVE. The rate of transition from state i to $i+1$ ($i \leq K-2$)

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 occurs in any state, the new state becomes state 0 (due to chain compression). Therefore, from all states, a transition to state 0 occurs with rate λ_c. From state $K-1$, a MOVE also causes a transition to state 0 (due to chain compression). Therefore, rate of transition from state $K-1$ to state 0 is $\lambda_m + \lambda_c$.

We make certain assumption 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 occuring *between two calls* is negligible. Therefore, no new failure may occur in state i' $(i = 0, \cdots K-1)$. We assume that VLR failures corrupt the VLR database, but the downtime is negligible compared to mean time to failure of a VLR.

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' $(i = 0, \cdots, K-1)$. Solving the Markov model (for more details refer to the appendix 7.1), we obtain the following expressions:

$$P_0 = \frac{1}{\sum_{i=0}^{k-1} \left(1 + \frac{(i+1)\lambda_v}{\lambda_c+\lambda_m}\right) \alpha(i)}, \text{ where } \alpha(i) = \begin{cases} 1 & \text{for } i = 0 \\ \frac{\lambda_m^i}{\prod_{j=1}^{i} (\lambda_c+\lambda_m+(j+1).\lambda_v)} & \text{for } i = 1, \ldots, K-1. \end{cases}$$

$$P_i = \alpha(i)\ P_0, \qquad Q_i = (i+1).\frac{\lambda_v}{\lambda_c+\lambda_m}\alpha(i).P_0, \qquad i = 0, \ldots, K-1$$

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 compounds :

- 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 accurate since VLRs cover large regions [2].

Thus, we have

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

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 for scheme 2

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 :

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

5.2.2 Expected cost of a FIND operation for scheme 2

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 now use the Markov model in Figure 8. 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. We consider the length of the chain being used by the HLR in a FIND operation.

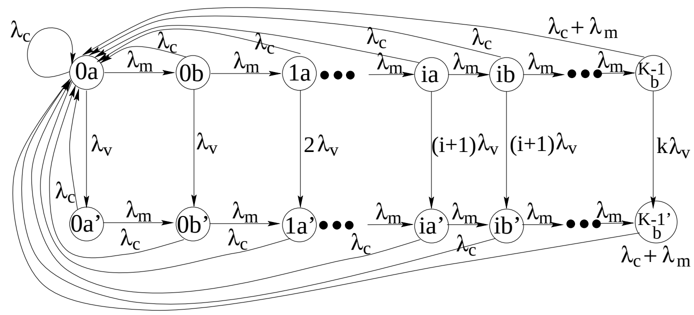


Figure 8: Markov model for scheme 2

Now, let P_i denote the steady state probability of being in one of the two states ia or ib ($i = 0, \cdots K-1$) in Figure 8. Similarly, let Q_i be the steady state probability of being in one of the two states ia' or ib' ($i = 0, \cdots K-1$). Note that these definitions of P_i and Q_i are slightly different from those for scheme 1. The expressions for P_i and Q_i (obtained by solving the Markov model) are different for the two schemes (please refer to the appendix 7.1).

Similar to F_1, FIND cost in absence of failure is obtained as

$$F_2 = F_{IS41} + T\left(\sum_{i=0}^{K-1} i.\left(\frac{P_i}{\sum_{j=0}^{K-1} P_j}\right)\right)$$

Now, we consider the expected cost F_2^f of a FIND operation which encounters a failed VLR. F_2^f has three components :

- Cost to query the HLR (this cost is equal to F_{IS41})
- Cost to traverse the first chain that contains a corrupted pointer.
- Cost to traverse the other chain (this chain does not contain a failed VLR, due to single failure assumption). Cost of this component is same as F_2.

Thus,

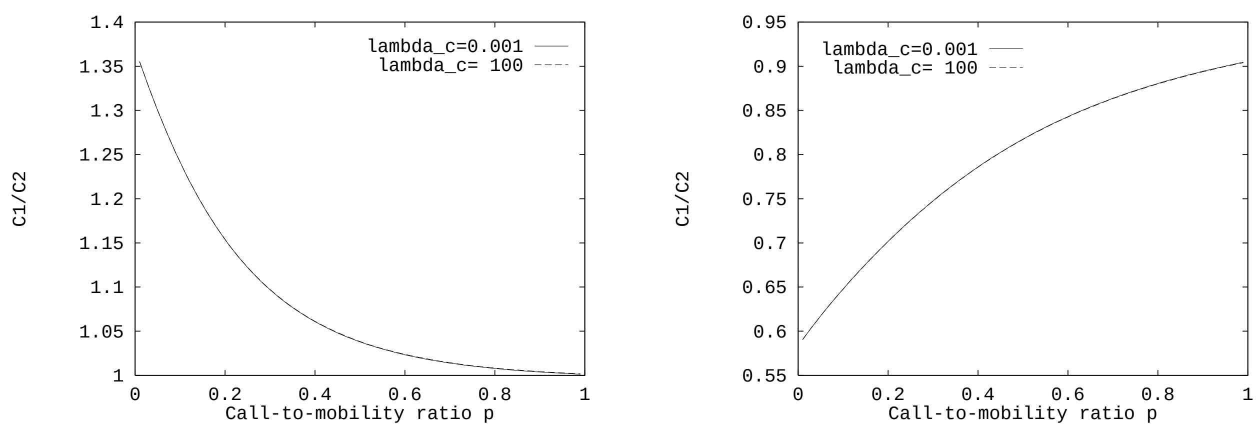
$$F_2^f = F_{IS41} + T\ \left(\sum_{i=0}^{K-1} i\left(\frac{Q_i}{\sum_{j=0}^{K-1} Q_j}\right)\right) + F_2$$

Finally, the expected cost per call C_2 is obtained as $C_2 = M_2 + (1 - P_f).F_2 + P_f.F_2^f$

6 Discussion and Conclusion

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 equal to 1. The 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 IS41 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 the failure rate $\lambda_v = 10^{-6}$.

To understand these results, the key question is whether the HLR updates are due to calls (FIND) or to the fact that the chain length reached the threshold value K. With $\lambda_c = 0.001$ (4 calls per hour) and $\lambda_c = 100$ (100 calls per second!!!), Figures 9(a) and (b) show that the ratio $\frac{C_1}{C_2}$ does not depend on the absolute value of λ_c. We get similar results for different values of K (i.e, insensitive to λ_c). But, the ratio $\frac{C_1}{C_2}$ depends on K and p.



(a) $S = \frac{1}{32}$ (b) $S = \frac{1}{2}$

Figure 9: Ratio $\frac{C_1}{C_2}$ with $K = 6$

For $K = 6$ and $S = \frac{1}{32}$ (see Figure 9(a)), the ratio $\frac{C_1}{C_2}$ decreases with the call-to-mobility ratio p. This is due to the fact that it is the average number of HLR updates (with cost $M_{IS41} = 32.S$) which determines the overall cost. Let us pick two values p_1 and p_2 with $p_1 \ll p_2$ (e.g., $p_1 = 0.01$ and $p_2 = 0.5$). With p_1, we have on the average 100 moves per call.

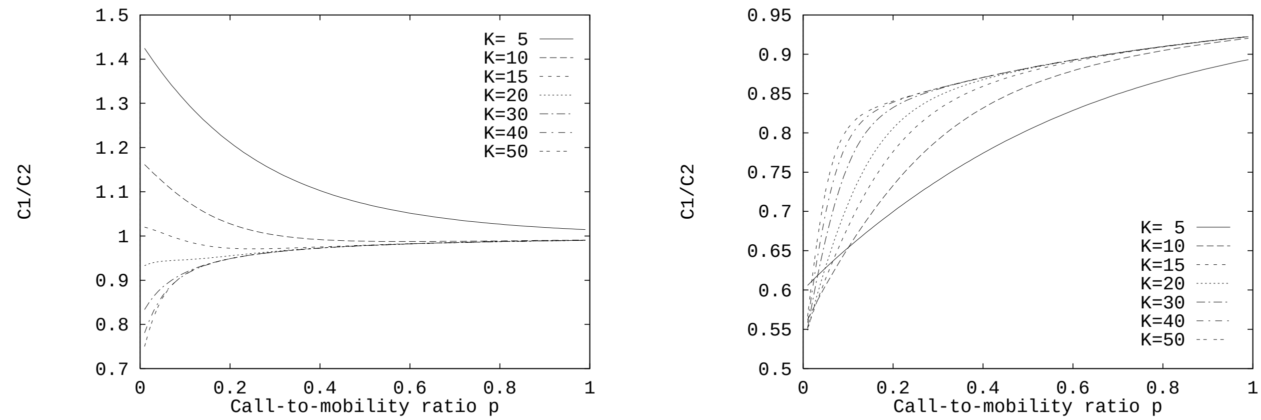
On the average, scheme 1 will perform 16 $\left(i.e, \frac{100}{6}\right)$ HLR updates while scheme 2 will perform twice less. Since the cost of one HLR update (i.e, $M_{IS41} = 32.S$) is much larger than S, the overhead of scheme 2 -in setting an extra pointer per move- is negligible in comparison of the number HLR updates avoided by scheme 2. So, scheme 2 performs better than scheme 1 with $p = p_1 = 0.01$.

Now, with $p_2 = 0.5$, we have on the average 2 moves per call. This means that HLR update is due mainly to the calls because it is unlikely that the chain length will reach the threshold ($K = 6$). So, both schemes perform the same number of HLR updates. Since scheme 2 sets two pointers at each move, the cost for scheme 2 is almost twice that of scheme 1. Note also that since the chain is on the average less than 2, then scheme 2 does not profite from its lower cost for the FIND operation. Therefore, scheme 1 performs better than scheme 2 for $p = p_2 = 0.5$. In conclusion, for $S = \frac{1}{32}$ and $K = 6$, the ratio $\frac{C_1}{C_2}$ decreases with p.

Now, with $S = \frac{1}{2}$ and $K = 6$ (see Figure9(b)), the ratio $\frac{C_1}{C_2}$ increases with p. This is due to the fact that at each move, scheme 2 sets two pointers with a cost $2.S = 1 = M_{IS41}$ (cost of an HLR update!). So, it is not the number of HLR updates which is critical. It is the cost of the FIND operation which is determinant. So, with p small, the cost for scheme 2 will be twice that of scheme 1 (rare calls). As p increases, the relative number of calls (FIND operations) increases. Since the expected chain length with scheme 2 is half that with scheme 1, the cost of FIND operation for scheme 2 is half that of scheme 1. So, scheme 2 performs better with more frequent calls, as p increases, and balances the overhead incurred by the MOVE operations. In conclusion, for $S = \frac{1}{2}$ and $K = 6$, the ratio $\frac{C_1}{C_2}$ increases with p.

Now, we set $\lambda_c = 1$ and study the impact of the threshold K with the plots on Figure 10. For $S = \frac{1}{32}$ and $K \leq 20$, the ratio $\frac{C_1}{C_2}$ decreases with p (see case $S = \frac{1}{32}$ and $K = 6$ above). Now, for $S = \frac{1}{32}$ and $K > 20$ (see Figure 10(a)), the ratio $\frac{C_1}{C_2}$ increases with p. Let us pick $p = 0.1$ and a large value for K (e.g, $K = 50$). The key here is that it will be less likely that the chain length in both schemes will reach the threshold $K = 50$. Since moves cost for scheme 2 is twice the moves cost for scheme 1 and calls are rare, then scheme 1 will perform better for small p. Now, let us pick $p = 0.5$. We have two moves per call. Since calls are frequent, the FIND cost is determinant. Note also that the FIND cost for scheme 2 is cheaper. Then scheme 2 outperforms scheme 1. Therefore, the ratio $\frac{C_1}{C_2}$ increases with p for $S = \frac{1}{32}$ and $K > 20$.

Now, for $S = \frac{1}{2}$ (see Figure 10(b)), the ratio increases with p. Please refer to the

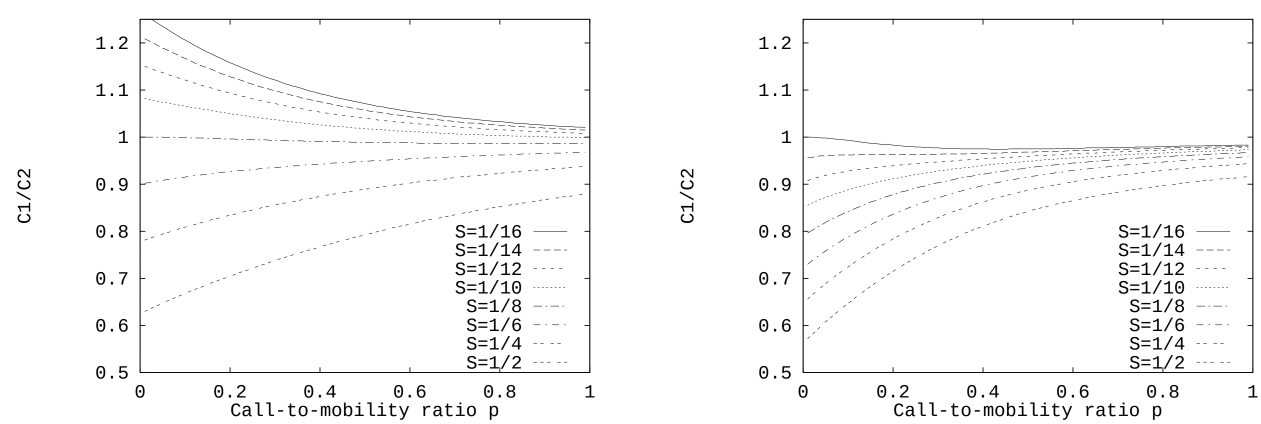


(a) $S = \frac{1}{32}$ (b) $S = \frac{1}{2}$

Figure 10: Ratio $\frac{C_1}{C_2}$ with $\lambda_c = 1$

explanation above for Figure 9(b).

With large areas covered by the VLRs, it is unreasonable to take threshold values larger than 6. So, given K, the determinant factor will be the fraction S with respect to M_{IS41}. We study on Figure 11 the impact of the parameter S for $K = 4$ and $K = 8$.



(a) $K = 4$ (b) $K = 8$

Figure 11: Ratio $\frac{C_1}{C_2}$ with $\lambda_c = 1$

When $S \leq \frac{1}{15}$, scheme 2 will always outperform scheme 1. When $S \geq \frac{1}{5}$, scheme 1 outperforms always scheme 2. For $\frac{1}{15} < S < \frac{1}{5}$, their relative performance depends on K. For small values of K ($K = 2, 3,$ *or* 4), scheme 2 always outperforms scheme 1. For larger values of K ($K = 5, \cdots, 10$), scheme 1 always outperforms scheme 2. In conclusion, if the threshold

K is set, the choice of a scheme depends on S. We summarize in the table 1 our results. Given a value for K and a value for S, we give the scheme wich is the best. For instance, for $S \geq \frac{1}{5}$, we put 1 as scheme 1 always outperforms scheme 2.

K/S	$S \leq 1/15$	$1/15 < S < 1/5$	$S \geq 1/5$
$K = 2$	2	2	1
$K = 4$	2	depends on p	1
$K = 6$	2	1	1
$K = 8$	2	1	1

Table 1: Relative Performance depending on K and S

We have described and analyzed two variations to the forwarding scheme to tolerate VLR failures. Our results show that the relative performance of the two schemes depends mainly on the threshold value K and the fraction S with respect to M_{IS41}. The parameter S depends on the system design.

References

[1] A. Acharya and B. Badrinath, "Checkpointing distributed applications on mobile computers," in *IEEE Conf. on Parallel and Distributed Information Systems*, Oct. 1994.

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

[3] I. F. Akyildiz, J. S. Ho, and Y.-B. Lin, "Movement-based location update and selective paging for PCS networks," *IEEE/ACM Transactions on Networking*, pp. 629–638, 1996.

[4] S. Alagar, R. Rajagopalan, and S. Venkatesan, "Tolerating mobile support station failures," in *Proceedings of IEEE Conf. on Fault Tolerant Systems*, pp. 225–231, Dec. 1995.

[5] N. M. Amato and M. C. Loui, "Checking linked data structures," in *International Symposium on Fault-Tolerant Computing FTCS-24*, pp. 164–173, June 1994.

[6] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile users: To update or not to update?," *Wireless Networks journal*, vol. 1, no. 2, pp. 175–186, 1995.

[7] S. Biaz and N. H. Vaidya, "Performance analysis of a fault-tolerant scheme for location management of mobile hosts." submitted to 1998 Workshop on Fault-tolerant Parallel and Distributed Systems.

[8] D. Cox, "Wireless personal communications: What is it?," *IEEE Personal Communications Magazine*, pp. 20–35, Apr. 1995.

[9] T. Imielinski and B. R. Badrinath, "Wireless mobile computing : Challenges in data management," *Communications of the ACM*, pp. 19–27, Oct. 1994.

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

[11] Q. Li and S. Vlaovic, "Redundant linked list based cache coherence protocol," in *1994 IEEE Workshop of Fault-Tolerant Parallel and Distributed Systems*, June 1994.

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

[13] Y. Lin, "Reducing location update cost in a PCS network," *IEEE/ACM Transactions on Networking*, pp. 25–33, Feb. 1997.

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

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

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

[17] K. Meier-Hellstern and E. Alonso, "The use of SS7 and GSM to support high density personal communications," in *Proceedings Int'l Conference on Communications*, 1992.

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

[19] D. Pradhan, P. Krishna, and N. Vaidya, "Recovery in mobile environments: Design and trade-off analysis," in *26th Int'l Symposium on Fault-Tolerant Computing*, June 1996.

[20] S. Rangarajan, K. Ratnam, and A. T. Dahbura, "A fault-tolerant protocol for location directory maintenance in mobile networks," in *25th Int'l Symposium on Fault-Tolerant Computing*, June 1995.

7 Appendix

7.1 Scheme 1: Solution for the Markov Model

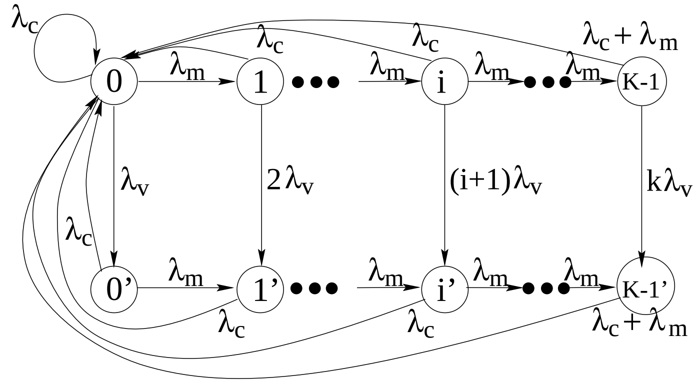


Figure 12: Markov model for scheme 1

In the steady state, we have the following equations for the states i :

$$\begin{aligned}
\lambda_c \sum_{i=0}^{K-1} (P_i + Q_i) + \lambda_m\ (P_{K-1} + Q_{K-1}) &= (\lambda_c + \lambda_m + \lambda_v)\ P_0 \\
(\lambda_c + \lambda_m + 2\ \lambda_v)\ P_1 &= \lambda_m\ P_0 \\
(\lambda_c + \lambda_m + 3\ \lambda_v)\ P_2 &= \lambda_m\ P_1 \\
(\lambda_c + \lambda_m + 4\ \lambda_v)\ P_3 &= \lambda_m\ P_2 \\
&\cdots \\
(\lambda_c + \lambda_m + (i+1)\ \lambda_v)\ P_i &= \lambda_m\ P_{i-1} \qquad (1)
\end{aligned}$$

From the system of equations (1) we get :

$$P_i = \alpha(i)\ P_0, \text{ where } \alpha(i) = \begin{cases} 1 & \text{for } i = 0 \\ \frac{\lambda_m^i}{\prod_{j=1}^{i} (\lambda_c + \lambda_m + (j+1).\lambda_v)} & \text{for } i = 1, \ldots, K-1. \end{cases} \qquad (2)$$

Now, for all states i', we can write :

$$Q_i = \left(\frac{(i+1)\ \lambda_v}{\lambda_c + \lambda_m} \right) P_i \quad for \ \ i = 0, \cdots, K-1 \qquad (3)$$

Since, the system is always in one of these states, we can write :

$$\sum_{i=0}^{K-1} (P_i + Q_i) = 1 \qquad (4)$$

Now, replacing in the equation (4) P_i and Q_i by their expressions (2) and (3), we can solve for P_0 and get :

$$P_0 = \left(\sum_{i=0}^{k-1} \left(1 + \frac{(i+1)\lambda_v}{\lambda_c + \lambda_m} \right) \alpha(i) \right)^{-1}$$

7.2 Scheme 2: Solution for the Markov Model

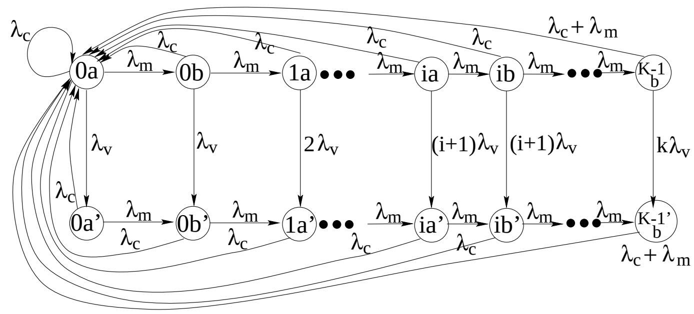


Figure 13: Markov model for scheme 2

Let P_{2i} and P_{2i+1} be the probabilities to be respectively is states ia and ib ($i = 0, \cdots, K-1$). For the states ia and ib, we can write the following equations :

$$
\begin{aligned}
\lambda_c \sum_{i=0}^{K-1} ((P_{2i} + P_{2i+1}) + (P_{Qi} + Q_{2i+1})) + \lambda_m \ (P_{2K-1} + Q_{2K-1}) &= (\lambda_c + \lambda_m + \lambda_v) \ P_0 \\
(\lambda_c + \lambda_m + \lambda_v) \ P_1 &= \lambda_m \ P_0 \\
(\lambda_c + \lambda_m + 2 \ \lambda_v) \ P_2 &= \lambda_m \ P_1 \\
(\lambda_c + \lambda_m + 2 \ \lambda_v) \ P_3 &= \lambda_m \ P_2 \\
(\lambda_c + \lambda_m + 3 \ \lambda_v) \ P_4 &= \lambda_m \ P_3 \\
(\lambda_c + \lambda_m + 3 \ \lambda_v) \ P_5 &= \lambda_m \ P_4 \\
\cdots & \\
(\lambda_c + \lambda_m + (i+1) \ \lambda_v) \ P_{2i} &= \lambda_m \ P_{2i-1} \\
(\lambda_c + \lambda_m + (i+1) \ \lambda_v) \ P_{2i+1} &= \lambda_m \ P_{2i} \\
\cdots &
\end{aligned} \tag{5}
$$

Replacing recursively each P_j with its expression in the system (5), we get :

$$
\begin{aligned}
P_{2i} &= \frac{\alpha(2i)}{(\lambda_c + \lambda_m + (i+1) \ \lambda_v)} \ P_0, \\
P_{2i+1} &= \frac{\alpha(2i+1)}{(\lambda_c + \lambda_m + (i+1) \ \lambda_v)^2} \ P_0, \quad \text{for} \quad i = 0, \cdots, K-1 \ where \\
\alpha(r) &= \frac{\lambda_m^r (\lambda_c + \lambda_m + \lambda_v)}{\prod_{j=1}^{\lfloor \frac{r}{2} \rfloor} (\lambda_c + \lambda_m + (j+1) \ \lambda_v)^2} \quad \text{for} \quad r = 0, \ldots, 2K-1.
\end{aligned}
$$

Now, for all states ia' and ib', we can write :

$$Q_{2i} = \left(\frac{(i+1)\ \lambda_v}{\lambda_c + \lambda_m} \right) P_{2i} \tag{6}$$

$$Q_{2i+1} = \left(\frac{(i+1)\ \lambda_v}{\lambda_c + \lambda_m} \right) P_{2i+1} \quad for \ \ i = 0, \cdots, K-1 \tag{7}$$

Since, the system is always in one of the states, we can write :

$$\sum_{i=0}^{K-1} (P_{2i} + P_{2i+1}) + (Q_{2i} + Q_{2i+1}) = 1 \tag{8}$$

Now, replacing in the equation (8) P_r and Q_r by their expressions (6) and (7), we can solve for P_0 and get :

$$P_0 = \left(\sum_{i=0}^{K-1} \frac{1}{\lambda_c + \lambda_m + (i+1)\ \lambda_v} \left(1 + \frac{(i+1)\lambda_v}{\lambda_c + \lambda_m} \right) \left(\alpha(2i) + \frac{\alpha(2i+1)}{\lambda_c + \lambda_m + (i+1)\ \lambda_v} \right) \right)^{-1} \tag{9}$$