Fast Recovery Protocol for Database and Link Failures in Mobile Networks*

Govind Krishnamurthi[1], Stefano Chessa[2], and Arun K. Somani[1]
[1]Department of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011
&
[2]Dipartimento di Informatica,
Universitá di Pisa, 56125 Pisa, Italia
Phone: (515) 294-0941, Fax: (515) 294-8432
e-mail: govindk@andes.ee.iastate.edu, ste@di.unipi.it, arun@iastate.edu

Abstract

An important issue in the design of future Personal Communication Services (PCS) networks is the efficient management of location information. The current IS-41 standard PCS architecture uses a centralized database, the Home Location Register (HLR), to store service and location information of each mobile registered in the PCS network. If the HLR fails, all incoming calls to a mobile from hosts which are not in the same location area as the mobile are lost. Location updates from mobiles to the HLR are also lost. Once the HLR is functional it can not direct calls to mobiles immediately as mobiles could have changed their location during the HLR's failure. Fast recovery from a failure of the HLR is hence important. A link failure in the network could partition the network resulting in a loss of location updates from mobiles affected by the failed link. We present a new protocol for fast recovery of the HLR after a HLR failure or an intermediate link failure. The protocol does not require use of wireless bandwidth during the recovery process, has a bounded recovery period and is simple to implement making it an appealing choice in the design of future mobile networks. We analyze the protocol in order to find a medium between protocol cost and the recovery interval.

Keywords: PCS networks, database failures, recovery protocol, location management

*This research was supported in part by the David C. Nicholas Professorship of Electrical and Computer Engineering at Iowa State University.

1 Introduction

Cellular communication technology has advanced rapidly in the last decade with significant developments in the capabilities of mobile networks. The third generation networks, called the Personal Communications Services (PCS) networks, will be capable of providing different types of services to a large population. A large amount of research activities is presently various design problems in PCS networks. In this paper we focus on one of the problems, namely the Location Management Problem.

1.1 The Location Management Problem

In PCS networks, a location tracking mechanism is needed to locate the position of the mobile hosts[1] in order to establish connections. Current methods require a mobile to report its location to the network when necessary [1]. Some of the location management schemes use a time-based update strategy. Other schemes require a mobile to update its location only if its present location is at least a pre-determined distance away from its location at the previous update. The network stores the location of the mobile in location-information-databases (LIDs) and this information is retrieved during call delivery. When a LID fails calls may have to be dropped as the location information of mobiles registered in the LID is unavailable. The failure of these databases therefore negatively affects the performance of the network. It is therefore imperative that the up-time of the databases be very high and in case of a failure the recovery period is very small. An intermediate link failure partitions the network resulting in the loss of location updates

[1]We refer to mobile hosts as mobiles in the rest of the paper.

from mobiles. Any location management protocol must therefore take into account the databases or links being in a state of failure at the time of update. In this paper, we present a new protocol for location management in presence of database and link failures. We analyze the protocol in terms of the average number of calls lost during the time of failure of the database and also during the time when the database is recovering from failure.

The rest of the paper is organized as follows. In Section 2 we describe the system and failure model. In Section 3 we present previous research in the area. In Section 4 we describe a new protocol for database recovery from database and intermediate link failures. We present an analysis of the protocol in Section 5. We discuss the results in Section 6 and conclude the paper in Section 7.

2 System and Fault Model

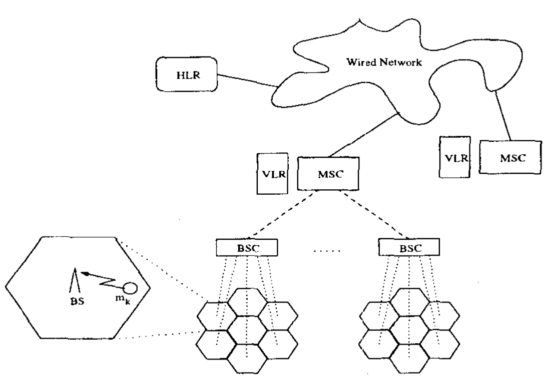

Figure 1. PCS Architecture

Most PCS networks use a cellular architecture for bandwidth efficiency [1], [2]. In this architecture, each cell has a base station to which the mobiles of the cell communicate through a wireless link. A set of base stations is controlled by a Base Station Controller (BSC). The primary function of a BSC is to manage the radio resources of its base stations, by performing hand-off and allocating radio channels. Each BSC is connected to a Mobile Switching Center (MSC) through a wired network. A MSC typically provides switching functions and coordinates location registration and call delivery. The MSC has access to the location information databases in the network, which are used to store location and service information for each registered mobile of the PCS network. PCS architectures which adopt the IS-41 standard [6] (Figure 1) use a two level hierarchy of LIDs for location management. These are the Home Location Register (HLR) and the Visitor Location Register (VLR). The HLR is a global database in which information about all mobiles registered in the PCS network is stored. A VLR is a local database usually associated with the MSC and stores information about mobiles visiting the MSC's Location Area (LA).

A single fault model (network link(s) or the HLR) is assumed in the paper. The HLR is assumed to be fail-stop and have a stable storage which can be recovered after a failure (this can be easily achieved using a redundant array of disks [3]). The HLR's volatile memory is periodically backed-up to the disk and the HLR logs the transactions on the disk between backups [4]. This ensures that we can retrieve the contents of the volatile memory of the HLR at the time of the HLR's failure from the contents of the disk backup and the log. Once a backup is done the log is deleted and a fresh logging process is initiated.

2.1 Location Update and Call Delivery

The Location Update (LU) protocol is used to update a mobile's position in LIDs when the mobile moves into a different LA. We describe the operation of LU and Call Delivery processes in the context of the IS-41 architecture. When a mobile m_1 moves from a LA with MSC MSC_1 to another LA with MSC MSC_2, it sends a LU to MSC_2 (Step 1 in Figure 2). MSC_2 delivers the LU to the HLR (Step 2 in Figure 2) and confirms the receipt of the LU received from MSC_2. The HLR also informs MSC_1 about the relocation of m_1. MSC_1 deletes the mobile's entry in VLR_1 (Steps 3 and 4 in Figure 2)on receiving this message. MSC_2 also updates VLR_2 with m_1's entry.

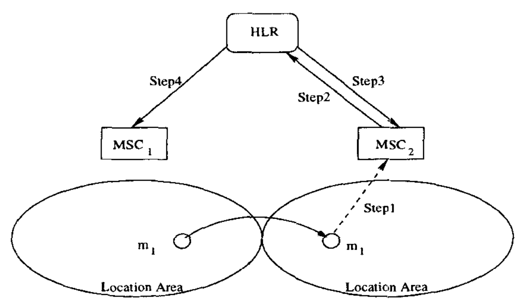

Figure 2. Processing a Location Update

The Call Delivery protocol is used to establish a call between two mobiles. When mobile m_1 calls another mobile m_2 it sends a Call Delivery Request (CDR) to MSC_1 (Step 1 in Figure 3). MSC_1 then queries the local VLR about for the position of the called mobile. If the information is retrieved, the MSC establishes a connection (if the mobile can receive a call) between the two mobiles. If the information is not present in the VLR, MSC_1 queries

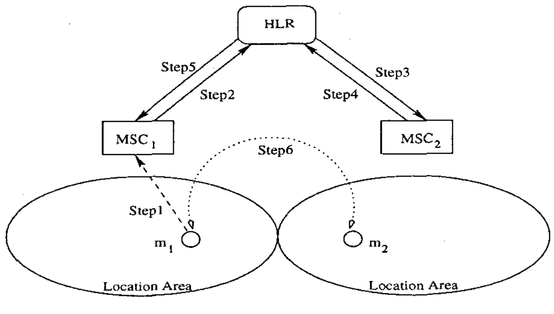

Figure 3. Processing a CDR

the HLR for the called mobile's information (Step 2 in Figure 3). The HLR determines the LA of the called mobile and sends a route request message to the MSC of the called mobile's LA (MSC_2) (Step 3 in Figure 3). MSC_2 then determines a Temporary Location Directory number for m_2 (if the called mobile can receive a call) and transfers the information to the HLR (Step 4 in Figure 3). The HLR delivers this information to MSC_1 (Step 5 in Figure 3), and establishes a connection between the mobiles (Step 6 in Figure 3). A similar protocol involving the HLR is used when a call originates from a fixed host. If a mobile calls a fixed host (a host in a wired network) the MSC establishes the connection with the fixed host. This operation does not depend on the state of the HLR.

3 Previous Research

In [5] during the normal operation of the HLR, the HLR's volatile memory is backed up on disk periodically. When recovering from a failure the HLR recovers the data from the non-volatile backup and sends an Unreliable Roamer Data Directive to its associated VLRs. The VLRs remove records of the mobiles associated with the HLR. Subsequently, when a mobile confirms its location by making a call or through a LU the HLR reconstructs its database in an incremental fashion. [6] also recommends periodic location updates from mobiles to reduce the recovery period. During the time the HLR is unable to determine a mobile's location all calls to the mobile are lost. An analysis of [5] and [6] is presented in [7]. It is shown that when the system is at least moderately reliable (period between HLR failures > 100 time units) the call origination rate and the rate of LA crossing has only a minor effect on the system cost provided the frequency of the frequency of the periodic update is comparable to the value of the call inter-arrival time. The IS-41 standard [6] recovery process has the following drawbacks. A periodic location update from each mobile is sent to the HLR irrespective of the state of the HLR, requiring the usage of wireless bandwidth. This problem is magnified if the population of mobiles is large. This also results in a power drain for the mobiles. To be effective the frequency of these periodic updates must be comparable to the frequency of incoming calls. Any reduction in the recovery period can only be achieved by increasing the frequency of the periodic location update which translates to a greater use of the wireless bandwidth. Frequent update messages also increases the load on the HLR.

An aggressive restoration procedure for the HLR is proposed in [8]. Once the HLR becomes functional after a failure, it requests the location information from the VLR's. To reduce the amount of information to be exchanged, the HLR requests only the information about mobiles which have moved after its last checkpoint. If the HLR is not checkpointed frequently, this approach may require a high communication overhead. The clocks of the HLR and the VLR also need to be synchronized. This approach is attractive only if very few records of information is sent from the VLR's to the HLR.

4 Fast Recovery Protocol

In this section we present a protocol to recover from HLR failures and intermediate link failures. As described in the previous section, when a mobile moves into a different LA it updates its location in the HLR through a location update message. We assume that the mobile can time-stamp (attach a serial number) the LUs. When a MSC receives a LU from the mobile it attempts to communicate the message to the HLR. If the MSC does not receive an acknowledgment from the HLR (failed HLR or link(s)), the MSC times-out and continually sends a LU retry (LU_r) with a period s (we assume that the latency of the wired network is much smaller than the retry period). The s value in retrying for a mobile should be chosen to trade-off protocol cost and lost incoming calls to the mobile during the recovery period. We discuss the factors which decide the choice of s in Section 6. Each LU_r has the same time-stamp as the LU it represents. The LU_r reaches the HLR once the HLR (link) becomes functional. If the HLR had failed, then it initiates the recovery process using the messages given below.

Messages sent while recovering from a HLR failure. The following messages are sent by the HLR to the MSC(s) when it initiates its recovery. In the following discussion we use HLR_{id} to accommodate the possibility of mobiles registered with different HLRs being present in the same LA.

- ALL-FREEZE(HLR_{id}): Broadcast by HLR_{id} to its registered MSCs, when the HLR becomes functional after being in a failed state. This freezes the entries of the mobiles belonging to HLR_{id} in the MSC's VLR. A frozen entry cannot be used until the HLR validates the entry.

This message prevents the MSC from routing calls to non-existent mobiles in its LA (when the calls arise from mobiles in its own LA).

- UN-FREEZE(HLR_{id}): Broadcast by HLR_{id} to its registered MSCs at the end of its recovery period (after a HLR failure). This unfreezes the frozen entries of mobiles registered with HLR_{id} in the VLR.
- DEL(HLR_{id},MSC_{id},m_k): Sent by the HLR to MSC_{id} when the HLR determines that m_k does not reside in MSC_{id}'s LA. On receiving this message the MSC removes the entry of mobile m_k (belonging to HLR_{id}) from the VLR. The MSC stops retrying for the mobile.
- UPDATE(HLR_{id},MSC_{id},details of m_k): Sent by HLR_{id} to MSC_{id} when it determines that the mobile resides in the MSC's LA. When this message is received the MSC adds m_k's entry to the VLR if it is not already there. Otherwise the MSC un-freezes the entry in the VLR (This happens if the mobile had moved out of the LA during the failure of the HLR and then returned to the LA during the recovery period of the HLR. This entry can now be used for call delivery).
- UPDATE-FREEZE(HLR_{id},MSC_{id},details of m_k): Sent by HLR_{id} to MSC_{id}, when the HLR receives a LU_r from the MSC. The VLR entry for mobile m_k is updated with the data in the message, but is kept in a frozen state. The MSC stops retrying for the mobile.

When the HLR is in its failed state (*off period*), all incoming calls to a registered mobile needing location information from the HLR are lost. If a mobile moves to a new LA say LA_i during this period, then any update sent from MSC_i to the HLR is also lost. Recall that we have assumed that the HLR has stable storage. The disk therefore can be retrieved after the HLR becomes functional. The HLR after the off period enters a period of time when it rebuilds its database (*recovery period*). The HLR determines whether any of its member mobiles have moved to a different LA during its off period. It then rebuilds itself incrementally with a mobiles' new position when it receives the appropriate message from the mobile. We show that this period is bounded by a time s (LU_r period) after which the location of every mobile is known.

We now describe the behavior of the HLR during its recovery period. We will discuss the number of messages sent, the length of the recovery period, and the number of incoming calls lost in each of these cases, pertaining to a single mobile m_k. Two messages are always sent, ALL-FREEZE at the beginning of the recovery and UN-FREEZE at the end of the recovery. An UPDATE-FREEZE is sent to the MSCs which retry for mobile m_k during the recovery period (an UPDATE is not sent here as the HLR is unsure about the number of LAs visited by the mobile during its failure). In each of the following cases incoming calls to the mobile arriving during the off and recovery period are lost.

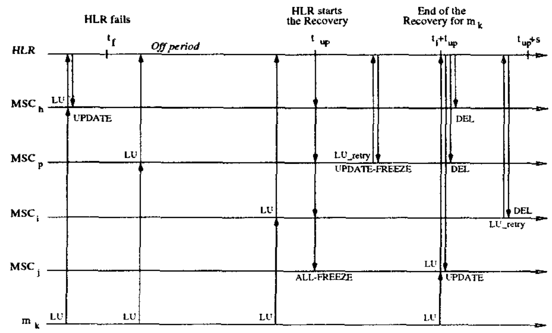

Figure 4. Working of the protocol in Case 1

Case1. *A LU (CDR) arrives from MSC_j at time t_l (t_c) $< s$ after the off period, from m_k (Figure 4)*

- Explanation: The mobile has moved to a new LA (has made a call) during the recovery.
- Recovery period for the mobile (t_{rec}) is t_l (t_c).
- UPDATE(MSC_j, HLR_{id}, m_k details) sent to MSC_j. DEL(MSC,HLR_{id},m_k) sent to the MSCs which have retried for m_k during the recovery period and unless the MSC is MSC_j. A DEL message is also sent to the mobile's MSC at the time of the HLR's failure unless the MSC is MSC_j.
- If any retry message is received after t_{rec}, a DEL message is sent to the MSC unless it is MSC_j.

Case2. *At time s after the off period $LU_r(s)$ has(ve) arrived, and no LU (CDR) has arrived from the mobile (Figure 5)*

- Explanation: The mobile had moved to at least one LA(s) during the off-period of the HLR and it did not move or make a CDR during the recovery period.
- The HLR compares the time-stamps, sends an UPDATE to the MSC with the largest time-stamp and multi-casts a DEL message to the other MSCs which have retried for m_k. A DEL message is sent to the MSC of the LA where the mobile resided at the time of the HLR failure, unless it is the same as the MSC where the UPDATE is sent.

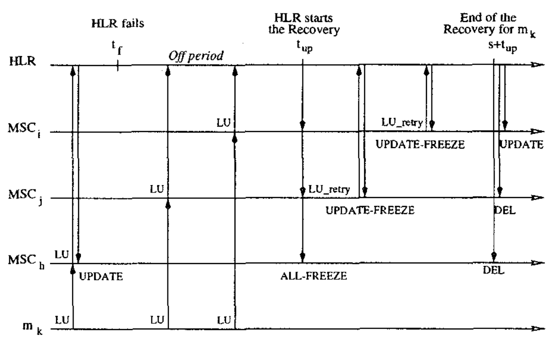

Figure 5. Working of the protocol in Case 2

- The recovery period lasts for a time s after the off-period.

Case 3. *The HLR does not receive any message regarding m_k during a time s after the off-period*

- Explanation: The mobile did not move to a new LA during off-period, nor does it move or make a CDR during the recovery period.
- The recovery period (t_{rec}) for m_k lasts for a time s, which is the maximum time for recovery.

A LU_r received during the HLR's recovery is logged before sending an UPDATE-FREEZE. If the HLR fails again during the recovery period, MSCs which have not received the UPDATE-FREEZE keep retrying. When the HLR becomes functional again, a new recovery process is started taking its logs from the previous recovery process into account.

4.1 Recovery from a Link Failure

If a link between the MSC and the HLR fails[2] we handle it as follows. The HLR now receives a LU_r when it is not in its *recovery period*. The HLR recognizes the failure scenario and initiates the recovery procedure. We present two approaches to recover from a link failure.

Optimistic approach: If the link failures last for a short period, then very few mobiles update more than once during the time the link is failed. In this case the HLR treats each LU_r as a LU and sends an UPDATE to the MSC (if the time-stamp of the LU is the latest it has received from the mobile). This ensures that the recovery period depends only on the processing time of the messages. The HLR could, however, misdirect a call to a wrong MSC if any mobile does update more than once during the link failure. If the failure time for the link is small then the probability of this misdirection is very low.

[2] In case of a link failure between the mobile's BS and the MSC a similar approach of retrying the message ensures that the LU is delivered at the MSC.

Pessimistic approach: If the duration of link failures is long, the HLR adopts the same recovery procedure as it would when recovering from its own failure.

In the case that a link failure occurs during the recovery of the HLR (after a HLR failure), and a mobile is unable to update its position, the HLR has an incorrect entry for the mobile's location at the end of the recovery period. This is however unavoidable as the mobile is disconnected from the network. The mobile's position is recovered when the faulty link becomes active again as explained previously in this section.

5 Analysis of the Protocol

In this section, we present an analysis of the protocol we described in the previous section in the context of HLR failures. We derive a closed form expression for the average cost of HLR failure in terms of calls lost per mobile and wasted communication (updates lost). The loss of outgoing calls from a mobile to another mobile is included in calculating the loss of incoming calls to the called mobile. Calls from a mobile to a fixed host do not depend on the state of the HLR as these calls are delivered by the MSC directly to the fixed host through the wired network.

We make the following assumptions. Calls to the mobile arrive at the HLR as a Poisson process with parameter λ_a. The inter-arrival time between successive LU's is assumed to be exponentially distributed with parameter λ_l. The time of HLR failure is assumed to be exponentially distributed with parameter λ_r, and is denoted by a random variable R. The mean time to failure (MTTF) of the HLR is T_f. The update retry interval is s. CDR's from the mobile (calls made by the mobile) arrive at the HLR as a Poisson process with parameter λ_c. The parameters are illustrated in Figure 6.

5.1 Expected Length of Recovery Period

As described in Section 4 the recovery period is upper bounded by s. If the HLR receives a LU or CDR from a mobile within a time s then the recovery period is shorter. We now describe the quantities used in our analysis. Let *Rec* be a random variable representing the length of this interval (recovery period). Let the inter-arrival time between two CDRs from the mobile be T_c and let T_l be the inter arrival time between two LUs. T_c and T_l are exponentially distributed by our initial assumption. Let t_c and t_l be the times at which an incoming

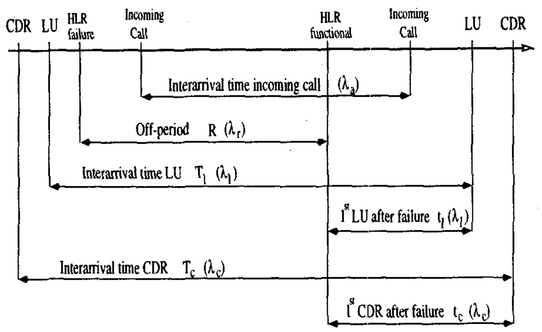

Figure 6. Illustration of the various parameters used in the analysis

call or a LU arrives after the HLR becomes functional. By the memoryless property of the exponential distribution, t_c and t_l are exponentially distributed. The length of the recovery period is given by $min(s, t_c, t_l)$. The probability density function f_{Rec} being

$$f_{Rec}(t) = \begin{cases} (\lambda_c + \lambda_l)e^{-(\lambda_l+\lambda_c)t} & 0 \le t < s \\ e^{-(\lambda_c+\lambda_l)t}\delta(t-s) & t = s \\ 0 & \text{otherwise} \end{cases}$$

where $\delta()$ is the Dirac Delta function. The average length of the recovery period E_{Rec} is

$$E_{Rec} = \frac{1 - e^{-(\lambda_c+\lambda_l)s}}{\lambda_c + \lambda_l}. \qquad (1)$$

5.2 Expected Loss of Incoming Calls

We now derive expressions for the expected loss of incoming calls to the mobile. These involve calls which arrive during the off period and during the recovery period.

Let X_a be the random variable defining the number of call arrivals to a mobile. The probability that k calls arrive for a given off period r is

$$P(X_a = k | R = r) = \frac{(\lambda_a r)^k e^{-\lambda_a r}}{k!}.$$

Consequently

$$\begin{aligned} P(X_a = k) &= \int_0^\infty \frac{(\lambda_a r)^k e^{-\lambda_a r} \lambda_r e^{-\lambda_r r} dr}{k!} \\ &= \frac{{\lambda_a}^k \lambda_r}{(\lambda_a + \lambda_r)^{k+1}}. \end{aligned}$$

The expected number of calls lost during the off period is

$$\begin{aligned} E_{Aoff} &= \sum_{k=1}^{\infty} \frac{k\lambda_a^k \lambda_r}{(\lambda_a + \lambda_r)^{k+1}} \\ &= \frac{\lambda_a}{\lambda_r}, \qquad \lambda_r > 0. \qquad (2) \end{aligned}$$

Similarly the expected number of incoming calls lost during the recovery period is $\lambda_a E_{Rec}$, where E_{Rec} is the mean length of the recovery period as shown in (1). The expected loss of incoming calls is

$$E_A = \frac{\lambda_a}{\lambda_r} + \lambda_a E_{Rec}, \qquad \lambda_r > 0. \qquad (3)$$

5.3 Expected Loss of Location Updates

If a MSC happens to send a LU during the off period, it times-out and repeatedly tries again with a period s, until it receives an ack from the HLR. This section analyses the average number of updates lost during the off period.

Let X_l be the random variable denoting the number of retries lost. If the first LU arrives at the HLR at time t after the HLR fails (during the off period), then the probability that there are k retries during the off period is given by

$$\begin{aligned} P(X_l &= k \mid \text{1st update at time t}) \\ &= \int_{t+ks}^{t+(k+1)s} \lambda_r e^{-\lambda_r r} dr \\ &= e^{-\lambda_r t}\left[e^{-\lambda_r ks}(1 - e^{-\lambda_r s})\right]. \end{aligned}$$

Factoring out the time t we have,

$$\begin{aligned} P(X_l = k) &= \int_0^\infty \lambda_l e^{-\lambda_l t} e^{-\lambda_r t}\left[e^{-\lambda_r ks}(1 - e^{-\lambda_r s})\right] dt \\ &= \frac{\lambda_l}{\lambda_r + \lambda_l}\left[e^{-\lambda_r ks}(1 - e^{-\lambda_r s})\right]. \end{aligned}$$

The expected loss of communication during the off period noting that on an average[3] $\frac{\lambda_l}{\lambda_r}$ LUs arrive during the off period and if for a LU k retries are made we loose $k+1$ updates is:

$$E_L = \frac{\lambda_l^2}{\lambda_r(\lambda_r + \lambda_l)(1 - e^{-\lambda_r s})}, \qquad \lambda_r > 0. \qquad (4)$$

5.4 Expected Cost of a HLR failure

We associate costs of C_A, C_L for the loss of an incoming call and LU, respectively. C_A is a cost

[3] Using a similar argument as in (2)

associated with the QoS of the system, while C_L is a cost associated with the system management. We have placed emphasis on the QoS in this paper, and hence we associate a higher cost to the loss of incoming calls rather than wasted communication. We however realize that the cost of wasted communication is important in designing a practical system, and hence include it in an appropriate way in our discussion of the protocol. If T_f is the MTTF then C_{tot} the average cost associated with the protocol is given by

$$C_{tot} = \frac{C_A E_A + C_L E_L}{T_f} . \quad (5)$$

The cost function does not include the cost of the ALL-FREEZE and UN-FREEZE messages. This is a fixed cost and does not play any role in determining the minimum of the cost function. It is also important to note that the protocol does not send any more messages for each LU_r it receives in the recovery period than what it would send for each LU it receives in a fully functional state. Hence we do not include the cost of protocol messages during recovery in calculating the total cost of the protocol.

6 Discussion

In our discussion we analyze the cost of the protocol in terms of incoming calls and LUs lost with respect to a single mobile, with various values of λ_a, λ_c, λ_l and s, and fixed values of $\lambda_r = 1, C_a = 1, C_l = 0.1$. The value of T_f does not play a very important factor in deciding the optimal value of s as it is just a scaling factor. We fix $\lambda_r = 1$ as λ_r just serves as a time scale.

Case 1: High λ_a, High λ_l. In Figure 7, when λ_c is low, the value of s is critical, and the minimum cost is achieved for $s \approx 0.1$. This is about 10% of λ_r. When λ_c is comparable to λ_a, the protocol incurs a high cost when s is small. The cost is lower for larger values of s, but does not vary much with increasing s.

This behavior can be explained as follows. For larger values of λ_c, λ_c determines the length of the recovery period. At lower values of λ_c, however the value of s plays an important part in deciding how long the recovery phase is, and hence a smaller s is better. However a very small retry interval increases the communication cost.

Case 2: Low λ_a, High λ_l. As seen in Figure 8, when λ_c is high, a large value of s is preferred. This is because the recovery period is dominated by λ_c, so the recovery period is almost always lesser than s. Hence in this case it is better not to loose $LU_r^\prime s$ during the HLR failure as will be the case when we choose a small s. On the other hand when λ_c is very low, the optimal value of s is small. This is because the cost associated with a lost call is much higher than a wasted update, and as λ_c is low the chance that the length of the recovery process being s is high, and hence higher is the probability of calls being lost during recovery.

Case 3: High λ_a, Low λ_l. In this case (Figure 9), the recovery period is determined by λ_c and s as λ_l is low. In other words the recovery period can be considered to be $min(s, t_c)$. As the value of λ_a is high, irrespective of what the λ_c is, we use the lowest possible value for s, taking into account the number of LU_r's generated, to minimize the recovery period.

Case 4: Low λ_a, Low λ_l. The behavior of the system (Figure 10) is essentially the same as in the previous case. The overall cost is lower due to a low λ_a. The optimal value of s is low, but is larger than the value of s shown in the previous case.

We are able to upper bound the recovery period by s in the worst case and to a much smaller value than s when the values of λ_c and λ_a are comparable or λ_l is high. In the case our initial assumption of negligible network latency is not true, the upper bound of the recovery period is s + *maximum latency* of the network. In future mobile networks, one can assume that mobiles registered in different HLR's reside in the same LA. The failure of one HLR should not negatively affect the QoS provided to the other mobiles. A recovery protocol therefore should minimize the usage of wireless bandwidth. The protocol proposed in this paper is simple to implement and does not use any wireless bandwidth while bounding the recovery period which makes it a feasible choice in designing future mobile networks.

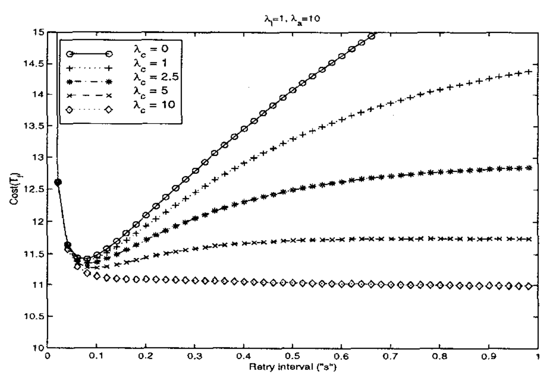

Figure 7. Cost vs (λ_c, s) for high λ_l and λ_a

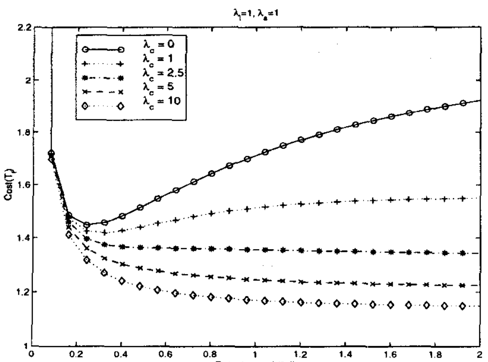

Figure 8. Cost vs (λ_c, s) for high λ_l and low λ_a

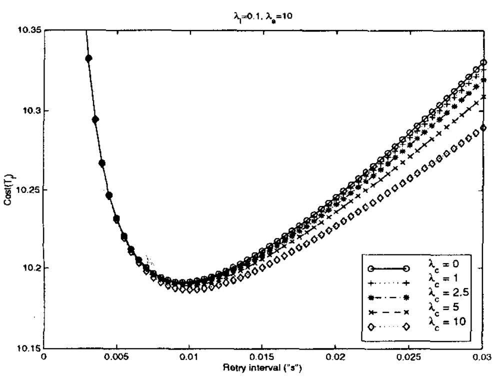

Figure 9. Cost vs (λ_c, s) for low λ_l and high λ_a

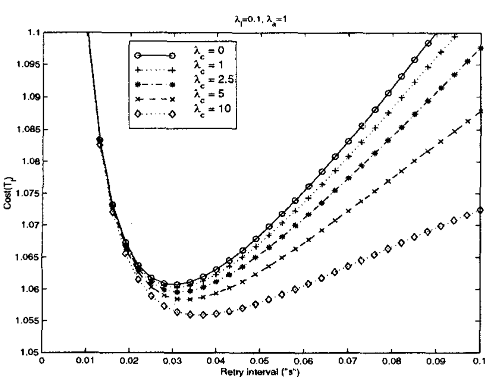

Figure 10. Cost vs (λ_c, s) for low λ_l and λ_a

7 Conclusions

We have presented a new protocol to recover from database and link failures in mobile networks. The protocol is simple and is easy to implement in current networks. The protocol does not require use of wireless bandwidth for recovery messages. [6] requires the mobiles to inform the HLR of its location periodically (irrespective of the state of the HLR) to reduce the recovery period after a HLR failure. The recovery protocol presented does not send any message when the HLR has not failed. We have shown by analysis that the length of the recovery phase is upper bounded by the retry interval s. We have presented a methodology to analyze the system, and have shown the values of s which minimize the cost in different scenarios which could exist in the system. The centralized LID architecture assumed in the paper is not essential for the protocol to work. The protocol presented can be easily extended to accommodate distributed LID architectures. An optimal recovery protocol for multiple database failures in a distributed architecture, and protocols to deal with multiple types of failures are open problems.

References

[1] I. F. Akyildiz, J. S. M. Ho, "On Location Management for Personal Communication Networks", IEEE Communications Magazine, Vol. 34, pp. 138-145, Sept. 1996.

[2] S. Tabbane, "Location Management for Third-Generation Mobile Systems", IEEE Communications Magazine, Vol. 35, pp. 72-84, Aug. 1997.

[3] D. A. Patterson, G. A. Gibson, R. H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)", In Proceedings of ACM SIGMOD International Conference on Management of Data, ACM Press, New York, pp. 109-116, May 1989.

[4] R. E. Strom, S. Yemini, "Optimistic Recovery in Distributed Systems", ACM Transactions on Computers, pp. 204-226, Aug. 1985.

[5] EIA/TIA, "Cellular Radio-Telecommunications Intersystems Operation, PN-2991", Nov. 1995.

[6] ETSI/TC-SMG, "Digital cellular telecommunications system; Restoration procedures", Version 5.0.0, GSM Technical Specification (GSM 03.07), ETSI, Nov. 1996.

[7] Z. Haas, Y.-B. Lin, "On Optimizing the Location Update Costs in the Presence of Database Failures," to appear in the ACM/Baltzer Wireless Networks Journal, 1998.

[8] Y.-B. Lin, "Failure Restoration of Mobility Databases for Personal Communication Networks", Wireless Networks 1 (1995), pp. 365-372.