行政院國家科學委員會專題研究計畫　期中進度報告

行動使用者位置資料最佳備份週期之研究(1/2)

計畫類別：個別型計畫
計畫編號：NSC94-2213-E-009-109-
執行期間：94 年 08 月 01 日至 95 年 07 月 31 日
執行單位：國立交通大學資訊工程學系(所)

計畫主持人：張明峰

計畫參與人員：張弘鑫、江俊賢、曾至鴻、許詩賢

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 95 年 5 月 30 日

行政院國家科學委員會補助專題研究計畫 □ 成 果 報 告
■期中進度報告

行動使用者位置資料最佳備份週期之研究(1/2)

計畫類別：■ 個別型計畫　　□ 整合型計畫
計畫編號：NSC94-2213-E-009-109-
執行期間：94 年 8 月 1 日至 95 年 7 月 31 日

計畫主持人：張明峰
共同主持人：
計畫參與人員：

成果報告類型(依經費核定清單規定繳交)：■精簡報告　□完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：交通大學 資訊工程系

中 華 民 國 九十五 年 五 月 三十 日

行政院國家科學委員會專題研究計畫成果報告

行動使用者位置資料最佳備份週期之研究(1/2)

The Analysis of Location Record Checkpointing Interval for Mobility Aatabase

計畫編號：NSC94-2213-E-009-109-

執行期限：2005.08.01 至 2006.07.31

主持人：張明峰　交通大學資工系

計畫參與人員： 張弘鑫、江俊賢、曾至鴻、許詩賢 交通大學資工系

中文摘要

使用者行動管理是個人通訊系統領域的重要議題。為了快速建立撥接到行動手機的電話，個人通訊系統必須維護用戶行動資料庫以記錄行動手機目前所在的位置。行動資料庫失效時，撥打到手機的電話就可能無法接通。行動資料庫失效後可從備份資料庫回復。由於個別用戶的行為模式不同（註冊頻率不同，來話流量不同），因此個別用戶位置資料為單位的備份方式可提供最佳的效能。本計畫分析比較三種個別用戶位置備份方法，我們考慮的成本函數包含位置資料備份的成本，以及呼叫手機的代價。我們以數學分析及計算機模擬的方法研究資料備份週期對成本函數的影響，並設法找出最佳的資料備份週期以得到最小的系統成本。我們目前的研究成果顯示如果註冊週期是指數分佈，我們探討的三種別用戶位置備份方法都得到相同最佳備份週期。如果資料備份成本高於手機呼叫成本，則用戶之位置資料不需備份；反之，如果手機呼叫成本高於資料備份成本，則當用戶註冊時同時備份其位置資料。我們也利用計算機模擬的方式探討 Gamma 分佈的註冊週期。基本上我們得到相同的結論，但是當註冊週期的變異（variance）高時，且資料備份成本高與手機呼叫成本相當時，則使用固定週期資料備份計時器之預期成本最低。.

關鍵詞: 行動通訊，行動資料庫，資料備份，個別用戶備份。

Abstract

Mobility database that stores the users' location records is very important to connect calls to mobile users on personal communication networks. If the mobility database fails, calls to mobile users may not be set up in time. This project studies failure restoration of mobility database. We study per-user location record checkpointing schemes that checkpoint a user's record into a non-volatile storage from time to time on a per-user basis. When the mobility database fails, the user location records can be restored from the backup storage. Numeric analysis, as well as computer simulation, has been used to choose the optimum checkpointing interval so that the overall cost is minimized. The cost function that we consider includes the cost of checkpointing a user's location record and the cost of paging a user due to an invalid location record. Our results indicate that when user registration intervals are exponentially distributed and the checkpointing timer duration is fixed or exponentially distributed, the user record should never be checkpointed if checkpointing costs more than paging. Otherwise, if paging costs more, the user record should be always checkpointed when a user registers. Computer simulation has also been used to study a more general case where user registration interval has a gamma distribution. Similar results as above on the optimal checkpointing frequency have been obtained, except when the variance of

user registration intervals is large, and the checkpointing cost and the paging cost almost balance, checkpointing timer of fixed duration should be used.

Keywords: Personal Communications Services (PCS), Mobility Database, Per-user Checkpoint.

1. Introduction

To set up a call in time to a mobile user in a cellular network, such as GSM (Global System for Mobile Communications) and UMTS (Universal Mobile Telecommunications System), it is necessary to constantly keep track of the mobile user's location. In GSM and UMTS, user location records are stored in a two-level database that consists of HLR (Home Location Register) and VLR (Visitor Location Register) [1]. The HLR resides in the user's home network and maintains mobile users' profile information and the current visited VLRs. For each visiting user in the location areas managed by a VLR, the VLR stores the user's subscription information and current location. When a mobile user crosses a location area, the user needs to register to the VLR and/or the HLR. Thus, the mobility database, HLRs and VLRs, are frequently modified for location tracking and queried for call delivery. If the mobility database fails, calls to mobile users may not be set up in time because of invalid location records.

Many mobility database restoration schemes have been studied. ETSI (European Telecommunications Standards Institute) recommends periodically autonomous registration [2] where a mobile user is required to register its location with the mobility database periodically even if the user does not cross a location area. Therefore, after a location database fails, an invalid location record can be restored sooner by the autonomous registration, and the number of calls lost due to invalid location records is reduced. Haas and Lin [3] considered the tradeoff between the cost of autonomous registration and the penalty of lost calls due to invalid location records. They suggested that the autonomous registration interval should be chosen to be approximately equal to the call inter-arrival time. Fang, et al. [4] considered the same cost function, and their study concluded that the optimal choice of autonomous registration interval may not be unique. They also showed that the optimal value can be found under certain traffic conditions. In addition, Fang, et al. [5] showed that the optimal autonomous registration interval depends on the weighting ratio between the registration signaling cost and the lost-call cost. To further reduce the time to restore invalid location records, Haas and Lin [6] proposed a demand re-registration scheme where mobile users are requested to re-register after the database fails. This scheme reduces the time to restore the location database. However, since user registration requires radio contact, this demanded re-registration from a large number of mobile users may cause repeated channel collisions, and thus waste wireless resources. Lin and Lin [7] studied a similar problem, the registration interval of badge-based location tracking system. Their results indicated that the channel collisions can be reduced by using exponential registration intervals without increasing the probability of losing calls due to invalid location records.

Checkpointing and rollback-recovery has long been used to reduce the expected execution time of long-running computation and to enhance the reliability of a database in presence of failures [8-11]. UMTS recommends that the mobility database is periodically checkpointed to a non-volatile storage [12]. After a mobility database failure, the user location information can be restored from the non-volatile storage. Checkpointing mobility database is more cost-effective than autonomous registration, because accessing a local non-volatile storage is in general cheaper and faster than accessing a radio channel. If a user's location record is not checkpointed every time when it is updated, the restored record may be out-of-date. In this case, to set up a call, the network can page the user at the location areas around the out-of-date location. Lin [13] derived the optimal checkpointing interval to balance the checkpointing cost against the paging cost, and showed that a user record need not be checkpointed if the checkpointing frequency is higher than 10 times or lower than 0.1 times of the user's moving rate. Wang, et al. [14] proposed an aperiodic checkpointing scheme where checkpointing of location database is not performed periodically but is triggered by a threshold on the number of uncheckedpointed location records. They also showed that aperiodic checkpointing outperforms periodic checkpointing when the threshold value is not large. Lin [15] proposed a per-user checkpointing algorithm where a user record is checkpointed only if the user record is modified when the checkpointing timer for the user expires. Otherwise, checkpointing is performed when the user registers for the next time. Since mobile users exhibit different characteristics in terms of registration and calling behavior, per-user checkpointing schemes can serve each user better than a whole-system scheme, but the system has to maintain a checkpointing timer for each user. This timer maintenance job seems to be a large overhead to the system, but the hashed and hierarchical timing wheels, designed by Varghese and Lauck [16], take constant ($O(1)$) time to maintain n outstanding timers, i.e., the time complexity is independent of the number of timers.

In summary, per-user checkpointing schemes can serve each user best without much overhead. However, no analysis has been done on the choice of the checkpointing intervals for per-user checkpointing scheme. In this project, we study three per-user checkpointing schemes and consider a cost function consisting of the checkpointing cost and the paging cost. Numeric analysis was used to derive the optimal checkpointing frequency when user registration interval is exponentially distributed. In addition, computer simulation was used to study a more general case where user registration interval has a gamma distribution.

The rest of the report is organized as follows. Section 2 describes three per-user checkpoint algorithms, and their analytic models are presented in Section 3. Numeric and simulation results are discussed in Section 4. Conclusions are given in Section 5.

2. Three Per-User Checkpointing Algorithms

To simplify our discussion, all the events that lead to location update of a mobile user, such as registration, call origination, and crossing of location areas, will be referred to as registration. Since accessing a radio channel is more expensive than accessing a local storage, we assume that no autonomous registration is performed. Note that for a per-user database checkpointing algorithm, the checkpointing timer and the registration interval for each user may be different.

Three per-user database checkpointing algorithms are depicted in Figure 1. The notation used in the figures is described as follows. t_r denotes the interval between two consecutive registrations and T_C denotes the checkpointing timer. In general, when T_C expires, the user record is checkpointed if it has been updated.

● Periodically checkpointing a modified record (FIXED)

The first scheme is essentially the same as the UMTS checkpointing method except that it is performed on a per-user basis and that only a modified location record is checkpointed. It works as follows,

1. When a user record is checkpointed, a timer, T_C, of fixed expiration interval is set on (see t_0 in Figure 1.a).
2. When T_C expires, if the user record has been modified, the user record is checkpointed (see t_3 and t_6 in Figure 1.a). Otherwise, if the user record has not been modified when T_C expires, T_C is restarted (see t_4 in Figure 1.a) and the process repeats.

   This scheme will be referred to as FIXED, because a timer of fixed expiration interval is used.

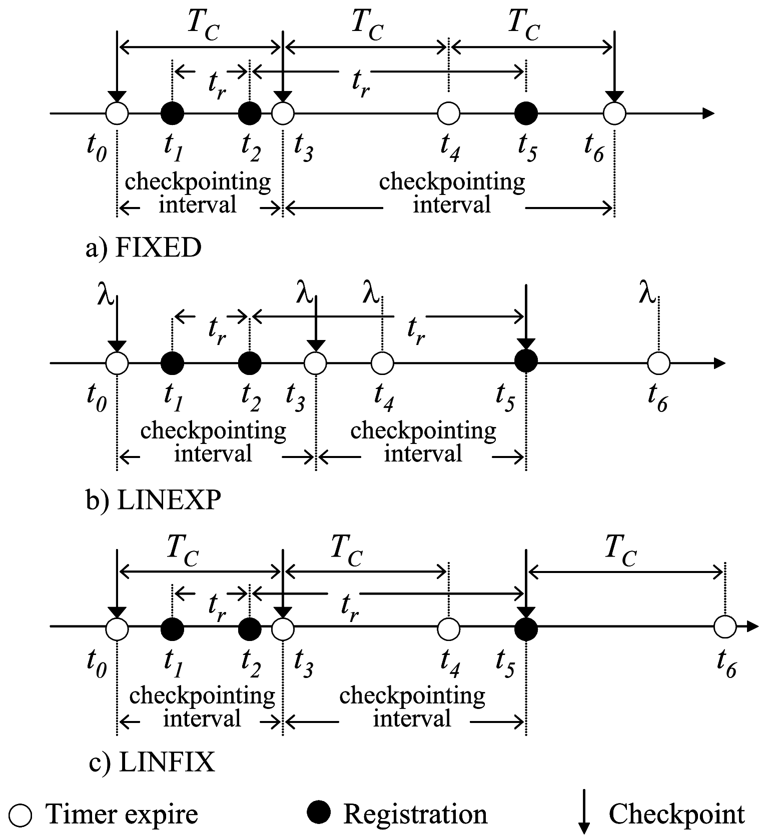

Figure 1. Three per-user checkpointing algorithms.

● Lin's per-user checkpointing algorithm with an exponential timer (LINEXP)

Lin presented a per-user checkpointing algorithm [15], which is illustrated in Figure 1.b. His algorithm assumes that timer T_C is exponentially distributed with mean $1/\lambda$. The algorithm is described as follows,

1. T_C is started when a user record is checkpointed (see t_0 in Figure 1.b).
2. When T_C expires, if the user record has been updated (see t_3 in Figure 1.b), it is checkpointed. Otherwise, if the user record has not been updated (see t_4 in Figure 1.b), the user record is

checkpointed at the next user registration (see t_5 in Figure 1.b).

Lin's algorithm differs from the FIXED scheme in that when the timer expires and the user record is not modified, the user record is checkpointed at the next user registration, but scheme FIXED waits until the timer expires after the next user registration. This algorithm will be referred to as LINEXP.

- Lin's per-user checkpointing algorithm with a fixed checking interval (LINFIX)

To study the effects of exponential timers and fixed timers, we apply fixed timers to Lin's per-user checkpointing algorithm. The algorithm is identical to LINEXP except that timer T_C is of fixed expiration interval. An example user registration and checkpointing scenario can be found in Figure 1.c. This algorithm will be referred to as LINFIX.

3. Numeric Analysis

The cost function we consider in the report includes the cost of paging a user with an invalid location record and the cost of checkpointing a user's location record. Let P_{ib} denote the probability that a user record in the backup database is invalid when the main database fails. When an invalid user record is encountered by an incoming call, the network pages the user. Let t_f denote the average database failure interval. It can be shown that the paging cost is proportional to P_{ib}/t_f. Let I denote the expected length of the checkpointing interval. The checkpointing cost is proportional to $1/I$. Let c_b denote the cost of checkpointing a location record, and c_p denote the expected cost to page a user with an invalid location record due to mobility database failure. For the cost function we consider, the checkpointing cost equals to $c_b\left(1/I\right)$, and the paging cost equals to $c_p\left(P_{ib}/t_f\right)$. The cost function is given as follows

$$C = c_b \cdot 1/I + c_p\left(P_{ib}/t_f\right) \quad \text{..........}(1)$$

We will study the effects of changing the expiration interval of T_C on the total cost, and try to find the optimal timeout interval to minimize the total cost. For our analytic models, t_r is assumed to be exponentially distributed with mean $1/u$. However, later in the computer simulation, t_r can have a gamma distribution with mean $1/u$ and variance σ.

- FIXED

Let T denote the expiration interval of timer T_C. Consider two consecutive checkpoints, checkpoints A and B, as shown in Figure 2. At checkpoint A, the user record is checkpointed and timer T_C is activated. Since the user registers after T_C expires for the (i-1)th time and before the i th time, the user record is checkpointed when T_C expires for the ith time, at checkpoint B.

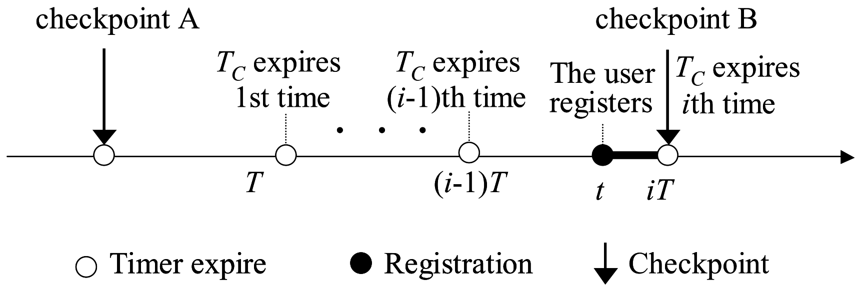

Figure 2. Two consecutive checkpoints (FIXED).

Let Q_i denote the probability that the interval between two consecutive checkpoints is of length iT, i.e., the user registers between time (i-1)T and iT. We have

$$Q_i = \int_{(i-1)T}^{iT} ue^{-ut}dt = e^{-u(i-1)T}(1-e^{-uT})$$

The expected checkpointing interval can be obtained as follows.

$$I_{FIXED} = \sum_{i=1}^{\infty} iTQ_i = \frac{T}{1-e^{-uT}} \qquad (2)$$

Since the inter-registration interval has an IID (independent identically distribution). The user registrations can be modeled as a renewal process. The behavior of checkpointings is also a renew process; because at each checkpoint, timer T_C is restarted and the registration interval is exponentially distributed. For a reliable mobility database, we expect the interval between two consecutive database failures is significantly larger than the user registration interval and the checkpointing interval. In this situation, the time when the database fails can be seen as a random observer to the renew process of user registration and that of checkpointing. The backup user record is invalid only after the user registers and before the record is checkpointed. If the main database fails during this period, the system restores an invalid backup record. Thus, we have

$$P_{ib\_FIXED} = \left(\sum_{i=1}^{\infty} \int_{(i-1)T}^{iT} ue^{-ut}(iT-t)dt \right) \Big/ I_{FIXED} = 1 - \frac{1-e^{-uT}}{uT} \qquad (3)$$

From (1), (2), and (3) the cost function can be obtained as follows,

$$C_{FIXED} = c_b \cdot \frac{1-e^{-uT}}{T} + \frac{c_p}{t_f}\left(1 - \frac{1-e^{-uT}}{uT}\right) = \frac{c_p}{t_f} + \frac{1}{u}\left(uc_b - \frac{c_p}{t_f}\right)\left(\frac{1-e^{-uT}}{T}\right) \qquad (4)$$

Our goal is to minimize the cost by choosing an appropriate T.

$$\frac{d}{dT}\left(\frac{1-e^{-uT}}{T}\right) = \left(-1 + \frac{1+uT}{e^{uT}}\right) \Big/ T^2$$

Since $e^{uT} = 1 + uT + (uT)^2/2! + (uT)^3/3! + \ldots$ and uT>0, we have $\frac{1+uT}{e^{uT}} \leq 1$ and $\left(-1 + \frac{1+uT}{e^{uT}}\right) \Big/ T^2 \leq 0$ for $T \geq 0, u \geq 0$. This leads to that $\frac{1-e^{-uT}}{T}$ is a monotonic decreasing function of T. From (4), we can draw the conclusions below,

1. If $uc_b < \frac{c_p}{t_f}$, C_{FIXED} is a monotonic increasing function of T. C_{FIXED} can be minimized when $T = 0$, i.e., the expiration interval of the timer is of length 0. At each user registration, since the timer must have expired, the user record should be checkpointed. In this case, $C_{FIXIED} = uc_b$.

2. If $uc_b > \frac{c_p}{t_f}$, C_{FIXED} is a monotonic decreasing function of T. C_{FIXED} can be minimized when $T = \infty$, i.e., the expiration interval of the timer is of infinite length. Since the timer never expires, the user record should never be checkpointed. In this case $C_{FIXIED} = \frac{c_p}{t_f}$.

3. If $uc_b = \frac{c_p}{t_f}$, $C_{FIXIED} = uc_b = \frac{c_p}{t_f}$; C_{FIXED} is a constant independent of T. T can be any value, i.e., at a user registration, the user record can be either checkpointed or not checkpointed. The cost of checkpointing the record and the cost of not checkpointing (the expected paging cost) are the same.

Note that the minimum cost that scheme FIXED can achieve equals to $Min\left(uc_b, \frac{c_p}{t_f}\right)$

● LINEXP

The analysis of the LINEXP is similar to that of the FIXED. Considering two consecutive checkpoints, there are two possible conditions as shown in Figures 3. For Case I, shown in Figure 3.a, the user registers before timer T_C expires, so that the checkpointing interval is equal to the expiration interval of timer T_C (s in Figure 3.a). For Case II, shown in Figure 3.b, the user registers after timer T_C expires, so that the checkpointing interval is equal to the user registration interval (t in Figure 3.b).

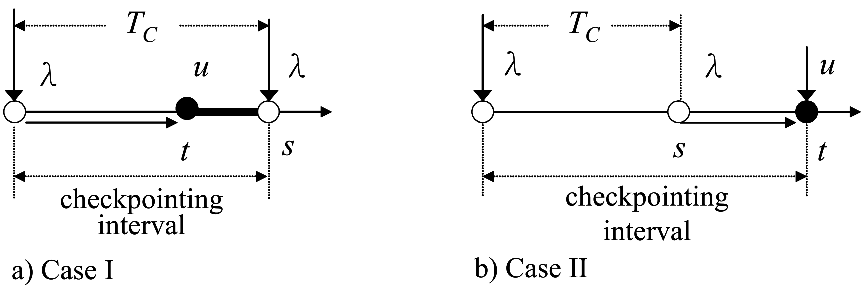

Figure 3. Two possible cases of checkpointing (LINEXP).

Since the registration interval and the checkpointing timer are both exponentially distributed, the expected length of checkpointing interval can be obtained by adding the intervals of both conditions.

$$I_{LINEXP} = \int_0^{\infty}\int_{t=0}^{s} s \cdot ue^{-ut} \cdot \lambda e^{-\lambda s} dtds + \int_0^{\infty}\int_{t=s}^{\infty} t \cdot ue^{-ut} \cdot \lambda e^{-\lambda s} dtds = \frac{1}{\lambda} + \frac{\lambda}{u(\lambda + u)} \text{ ....(5)}$$

For Case I, the backup user record is invalid only after the user registration at time *t*. For Case II, the backup user record is always up-to-date because when the user registers, the record is also checkpointed. From the random observer property, P_{ib} can be obtained as follows.

$$P_{ib\_LINEXP} = \left(\int_0^{\infty}\int_{t=0}^{s} (s-t) \cdot ue^{-ut} \cdot \lambda e^{-\lambda s} dtds\right) \Big/ I_{LINEXP} = \frac{u^2}{u^2 + \lambda u + \lambda^2} \text{ ....(6)}$$

From (1), (5), and (6), the cost function can be obtained as follows,

$$C_{LINEXP} = c_b \cdot \left(\frac{1}{\lambda} + \frac{\lambda}{u(\lambda+u)}\right) + \frac{c_p}{t_f} \cdot \frac{u^2}{u^2 + \lambda u + \lambda^2}$$

$$= uc_b + \left(\frac{c_p}{t_f} - uc_b\right) \cdot \frac{u^2}{u^2 + \lambda u + \lambda^2} \text{ ....(7)}$$

Since $\frac{d}{d\lambda}\left(\frac{1}{u^2 + \lambda u + \lambda^2}\right) = \frac{-(u + 2\lambda)}{(u^2 + \lambda u + \lambda^2)^2} \le 0$ for $\lambda \ge 0, u \ge 0$, $\frac{1}{u^2 + \lambda u + \lambda^2}$ is a monotonic decreasing function of *λ*. We also obtain the following results, which are essentially the same as those obtained from FIXED. Note that the expected timeout interval of the exponential timer is $1/\lambda$.

1. If $uc_b < \frac{c_p}{t_f}$, C_{LINEXP} is a monotonic decreasing function of λ. The optimum $C_{LINEXP} = uc_b$, when $\lambda = \infty$, i.e., the timeout interval of the checkpointing timer is of length 0.
2. If $uc_b > \frac{c_p}{t_f}$, C_{LINEXP} is a monotonic increasing function of λ. The optimum $C_{LINEXP} = \frac{c_p}{t_f}$, when $\lambda = 0$, i.e., the timeout interval of the checkpointing timer is of infinite length.
3. If $uc_b = \frac{c_p}{t_f}$, $C_{LINEXP} = uc_b = \frac{c_p}{t_f}$, a constant independent of λ. λ can be any value.

- LINFIX

Since this algorithm is identical to algorithm LINEXP except that it utilizes a checkpointing timer with fixed expiration interval. The two checkpointing cases of LINEXP shown in Figure 3 can also be used to analyze LINFIX. For Case I, the checkpointing interval is equal to the expiration interval of the timer, which is *T*. For the Case II, the checkpointing interval is equal to the user registration interval (*t*). The expected length of checkpointing interval can be obtained as follows.

$$I_{LINFIX} = \int_0^{T} T \cdot ue^{-ut} dt + \int_T^{\infty} t \cdot ue^{-ut} dt = T + \frac{e^{-uT}}{u} \text{ ....(8)}$$

P_{ib} equals to the probability that the main database fails in Case I after the user registration.

$$P_{ib\_LINFIX} = \left(\int_0^T (T-t) \cdot ue^{-ut} dt \right) \Big/ I_{LINFIX} = 1 - \frac{1}{uT + e^{-uT}} \text{ ....(9)}$$

From (1), (8) and (9), the cost function can be obtained as follows,

$$C_{LINFIX} = \frac{c_b}{T + \frac{e^{-uT}}{u}} + \frac{c_p}{t_f} \cdot \left(1 - \frac{1}{uT + e^{-uT}}\right) = \frac{c_p}{t_f} + \left(uc_b - \frac{c_p}{t_f}\right) \cdot \frac{1}{uT + e^{-uT}} \text{ ....(10)}$$

Since $\frac{d}{dT}\left(\frac{1}{uT + e^{-uT}}\right) = \frac{-u(1-e^{-uT})}{(uT + e^{-uT})^2}$ and $e^{-uT} \le 1$ for $T \ge 0, u \ge 0$, we get $\frac{-u(1-e^{-uT})}{(uT + e^{-uT})^2} \le 0$.

Thus, $\frac{1}{uT + e^{-uT}}$ is a monotonic decreasing function of *T*. We can obtain same results as in the FIXED and LINEXP.

1. If $uc_b < \frac{c_p}{t_f}$, C_{LINFIX} is a monotonic increasing function of *T*. The optimum $C_{LINFIX} = uc_b$, when $T = 0$.
2. If $uc_b > \frac{c_p}{t_f}$, C_{LINFIX} is a monotonic increasing function of *T*. The optimum $C_{LINFIX} = \frac{c_p}{t_f}$, when $T = \infty$.
3. If $uc_b = \frac{c_p}{t_f}$, $C_{LINFIX} = uc_b = \frac{c_p}{t_f}$. *T* can be any value.

It is important to note that the analyses of three algorithms all lead to the same conclusions. If the checkpointing cost out-weights the paging cost ($uc_b > \frac{c_p}{t_f}$), we should never checkpoint a user record. On the other hand, if $\frac{c_p}{t_f} > uc_b$, we should use a duplicated database.

4. Numeric Results

Without loss of generality, we let the expected user registration rate, *u*, to be 1 per unit-of-time. This can be interpreted as one registration per *x* minutes. A small *x* means the user registers frequently. First we consider exponential registration interval and examine the effects of the timeout interval (*T*) on the expected checkpointing interval (*I*) and on the probability of invalid backup record at database failure (P_{ib}). The expiration interval of timer T_C used in FIXED and LINFIX varies in the range 0.2-8 unit-of-time. In addition, the expected expiration interval of the exponential timer in LINEXP also varies in the range 0.2-8. The curves in Figure 4.a are obtained from Equations (2), (5), and (8), and those in Figure 4.b from Equations (3), (6), and (9). The results indicate that all three algorithms obtain similar results; both the expected checkpointing interval and the probability of invalid backup record increase as the timeout interval increases. The differences between the three algorithms are small, but for a given timeout interval, LINFIX has the smallest P_{ib} at the cost of the shortest checkpointing interval, *I*. When the timeout interval is larger than 4 (i.e., four times the registration interval), all checkpointing algorithms act much the same. This is because when a long checkpointing timer expires, the user record is most likely modified and needs to be checkpointed for all algorithms.

Figure 5 shows the cost functions at various paging costs; the user registration interval is exponential distributed. The curves are obtained from Equations (4), (7), and (10). Without loss of generality, let $u = 1$, $c_b = 1$ and $\frac{c_p}{t_f}$ vary in the range of 0.5-1.5. The results indicate when $uc_b = \frac{c_p}{t_f}$, the cost of all algorithms equals to 1 ($= \frac{c_p}{t_f} = uc_b$), and the total cost is independent of the timer expiration interval. Furthermore, when $\frac{c_p}{t_f} > uc_b$, the cost increases as the timeout interval, *T*, increases, and when $uc_b > \frac{c_p}{t_f}$, the cost decreases as *T* increases.

Computer simulation has been used to study the effects of changing registration interval variance on P_{ib} and *I*. The registration interval is assumed to have a gamma distribution with mean 1 unit-of-time and variance σ. In order to speedup the simulation, the database failure rate is chosen to be $1/500$, which may be unrealistically large but is small enough (compared to the user registration rate) to obtain correct simulation results, i.e., the random observer property still holds. In each computer simulation, the database fails for at least 10,000 times until stable results are obtained. The results in Figure 6 indicate that for all algorithms, *I* increases as σ increases. This is because as σ increases, there are more short registration intervals that are shorter than the checkpointing timer, and no checkpointing is needed at registration. In addition, P_{ib} drops as σ increases. This is because as σ increases, there are also more long registration intervals during which the backup record is always valid. As a random observer, the database failure is more likely to occur at long registration intervals. As a result, P_{ib} drops.

Figure 7 depicts the cost functions for various $\frac{c_p}{t_f}$ values and different variance of registration intervals. The results in Figure 7.a, c, d, and f indicate that the optimal choice of the checkpointing timer is determined by the weighting ratio of the paging cost and the checkpointing cost; it is independent of the variance of registration intervals. The optimal expiration interval of the checkpointing timer is either of length 0 or infinity. Only when the checkpointing cost and the paging cost almost balance, i.e., when $uc_b \approx \frac{c_p}{t_f}$, the variance of registration intervals affects the choice of checkpointing algorithms. The results in Figure 7.b indicate that when $\sigma\,(=0.5)$ is small, a duplicated database should be used (T =0). The results in Figure 7.e indicate that when σ ($=2$) is large, all algorithms can outperform a duplicated database scheme or a non-checkpointing database scheme. This is because as σ increases, there are more short registration intervals; By setting appropriate timer length, all three checkpointing schemes can skip the short registration intervals (shorter than the checkpointing timer) without checkpointing, and thus reduce the overall cost. Our results indicate FIXED is best at skipping short registration intervals, and the optimum expiration interval of the checkpointing timer can be obtained from computer simulation.

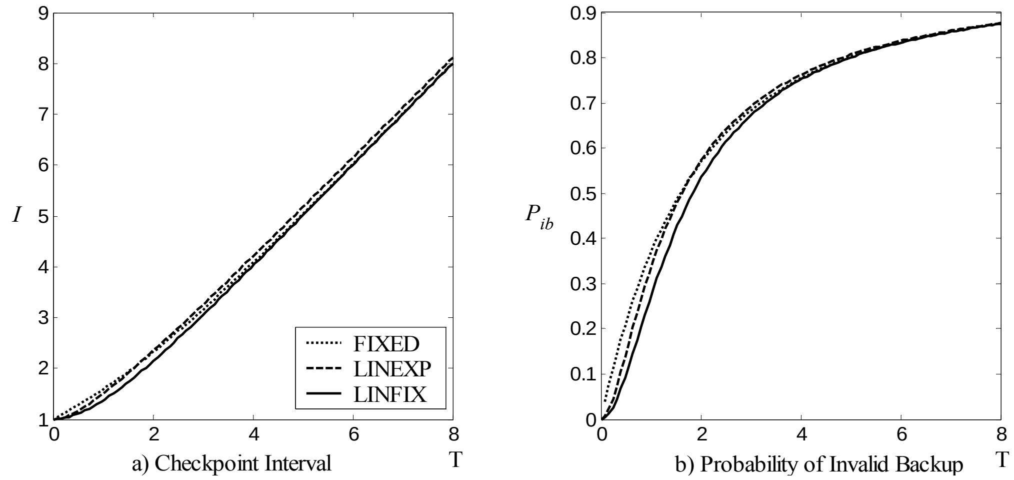

Figure 4. Comparison of checkpointing algorithms for exponential registration interval.

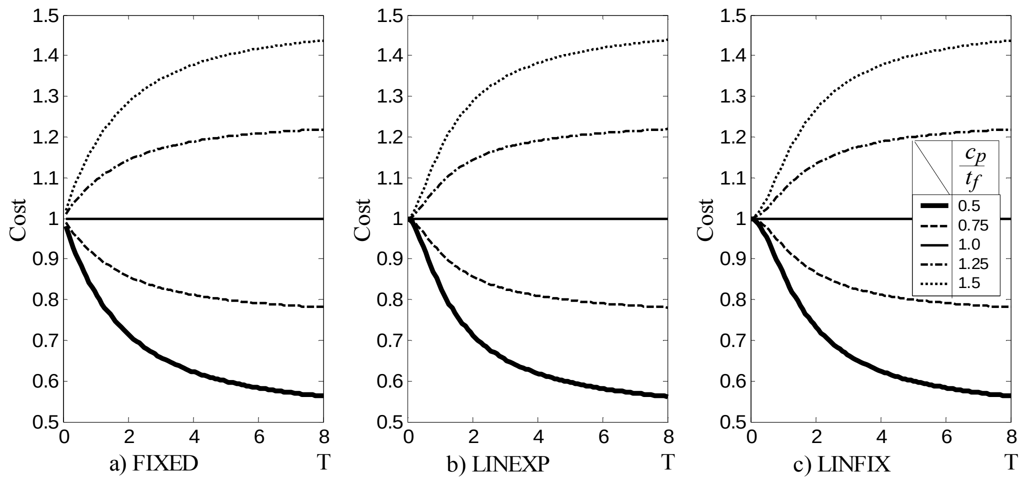

Figure 5. Cost functions for various paging costs (exponential registration interval).

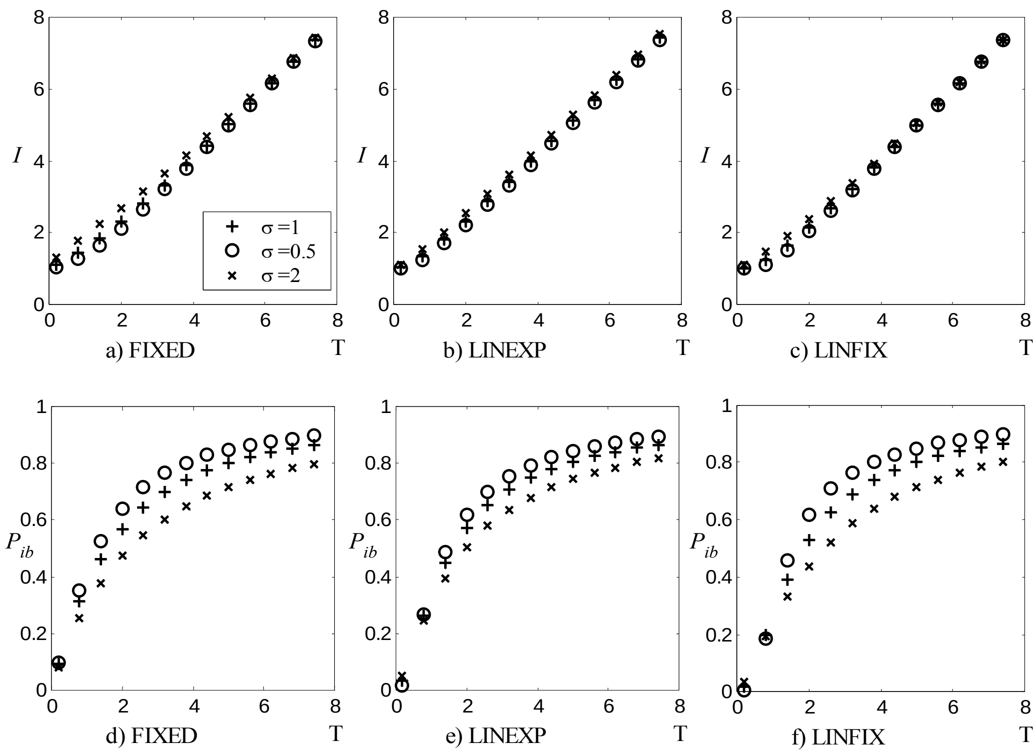

Figure 6. The effects of registration interval variance.

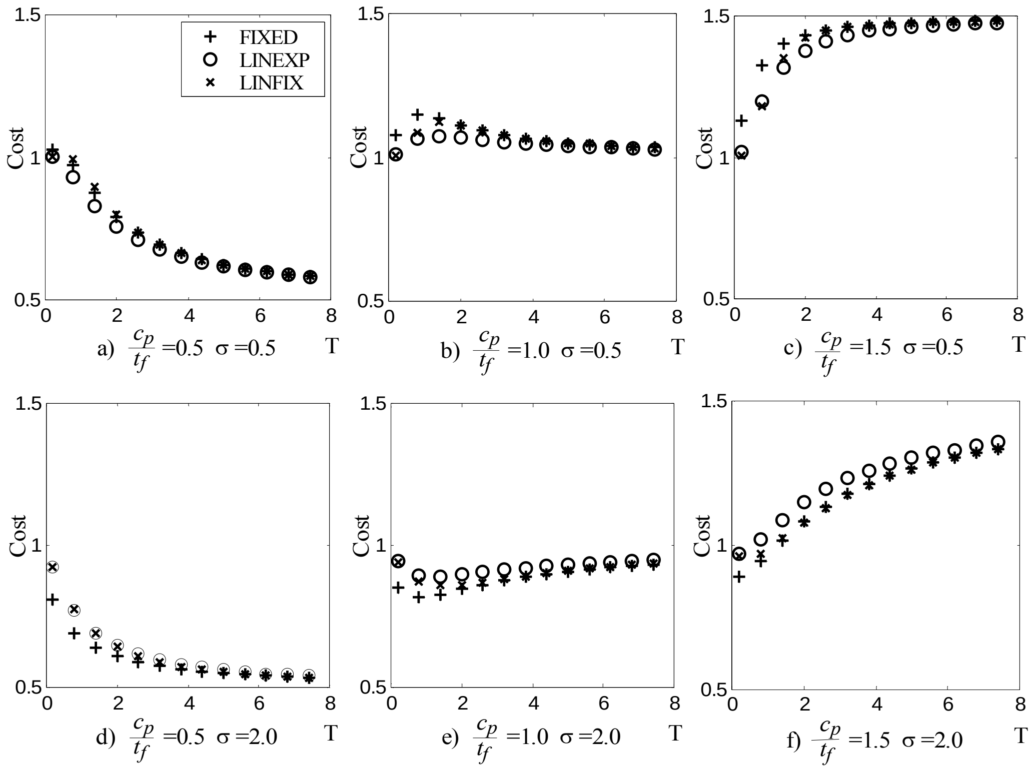

Figure 7. Cost functions for various paging costs and different variance of registration interval.

5. Conclusions

Checkpointing can be used to enhance the reliability of the location database of PCS networks, Since each user exhibits an unique calling and moving behavior, per-user checkpointing schemes can serve the users, as well as the operators, better. In this report, we have analyzed three per-user location database checkpointing algorithms using numeric analysis and computer simulation. The costs that we considered include the checkpointing cost and the paging cost. Our results indicate that when inter-registration times are exponentially distributed, a user location record should either be always checkpointed at registration, or be never checkpointed at all, depending on the weighting ratio between the checkpointing cost and the paging cost. If the checkpointing cost is of more concern, the user record should never be checkpointed; otherwise, the user record should be always checkpointed (duplicated) at registration. We have also studied the effects of the variance of registration interval using computer simulation. When the checkpointing cost and the paging cost almost balance, and the variance of registration interval is large, a simple checkpointing algorithm using a fixed checkpointing timer is preferred and the optimal choice of the checkpointing timer can be determined by computer simulation. In this report, we did not investigate the effects of incoming call arrivals on the optimal choice of the checkpointing frequency directly; we assumed that the expected paging cost is known. Further study is needed to obtain the paging cost in the PCS networks. In addition, if paging a user with an invalid location record cannot be done in time, the caller may hang up and the call is lost. It may be meritorious to consider a cost function consisting of the checkpointing, paging and lost-call costs.

Reference

[1] EIA/TIA, Cellular radio-telecommunication intersystem operation: Automatic roaming, *BEIA/TIA Technical Report IS-41.3* (1991).

[2] ETSI/TC, Restoration procedures, version 4.2.0, *ETSI Technical Report Recommendation GSM 03.07* (1993).

[3] Z. J. Haas and Y.-B. Lin, On the optimizing the location update costs in the presence of database failures, ACM-Baltzer Wireless Networks 4 (August 1998) 419-426.

[4] Y. Fang, I. Chlamtac and H. B. Fei, An active mobility database failure recovery scheme and performance analysis for PCS networks, in: Conference of the IEEE Communications Society (March 2000) pp. 757-764.

[5] Y. Fang, I. Chlamtac and H. B. Fei, Analytical results for optimal choice of location update interval for mobility database failure restoration in PCS networks, IEEE Transactions on Parallel and Distributed Systems 11 (June 2000) 615-624.

[6] Z. J. Haas and Y.-B. Lin, Demand re-registration for PCS database restoration, in: Proceedings Military Communications Conference 2 (October-November. 1999) pp. 887-892.

[7] Y.-B. Lin and P. Lin, Performance modeling of location tracking systems, ACM Mobile Computing and Communications Review 2 (July-August 1998) 24-27.

[8] J. W. Young, A first order approximation to the optimum checkpoint interval, ACM communications 17 (September 1974) 530-531.

[9] E. Gelenbe, On the optimum checkpoint interval, Joural of the Association for Computing Machinery 26 (April 1979) 259-270.

[10] J. S. Plank and W.R. Elwasif, Experimental assessment of workstation failures and their impact on checkpointing Systems, in: 28th International Symposium on Fault-Tolerant Computing (June 1998) pp. 48-57.

[11] J. S. Plank, K. Li and M. A. Puening, Diskless Checkpointing, IEEE Transactions on Parallel and Distributed Systems 9 (October 1998) 972-986.

[12] 3GPP. 3rd generation partnership project: Technical specification group services and system

aspects; General packet radio service (GPRS); Service description; Stage 2, Technical Specification 3G TS 23.060 version 4.1.0 (June 2001).

[13] Y.-B. Lin, Failure restoration of mobility database for personal communication networks, ACM-Baltzer Wireless Networks 1 (March 1995) 365-372.

[14] T.-P. Wang, C.-C. Tseng and W.-K. Chou, An aggressive approach to failure restoration of PCS mobility databases, ACM Mobile Computing and Communications Review 1 (September 1997) 21-28.

[15] Y.-B. Lin, Per-user checkpointing for mobility database failure restoration, IEEE Transactions on Mobile Computing 4 (March 2005) 189-194.

[16] G. Varghese and A. Lauck, Hashed and hierarchical timing wheels: Efficient data structures for implementing a timer facility, IEEE/ACM Transactions on Networking 5 (December 1997) 824-834.