

# On Failure Recoverability of Client-Server Applications in Mobile Wireless Environments

Ing-Ray Chen, *Member, IEEE*, Baoshan Gu, Sapna E. George, and Sheng-Tzong Cheng, *Member, IEEE*

**Abstract**—Analytical results for the Cdf of the failure recovery time for client-server applications in mobile wireless environments characterized by logging, and mobility handoff strategies for facilitating failure recovery are reported in the paper. The results can be applied to determine if a mobile application can satisfy its recoverability requirement upon a mobile host failure when operating under a set of parameter values characterizing the mobile application, the underlying client-server environment, and the logging & mobility handoff strategies adopted by the mobile application. Model parameters which affect the shape of the failure recovery time Cdf for two mobility handoff strategies, namely, Eager and Lazy, are identified, and their effects are analyzed, with numerical data and result interpretations given. A tradeoff analysis between the cost invested by these two mobility handoff strategies for maintaining the logging and checkpoint information before failure versus the return of investment in terms of improved failure recoverability is given, and the best checkpoint interval period that would yield the best return of investment for the eager mobility handoff strategy over the lazy strategy is identified.

**Index Terms**—Client-server mobile applications, failure recoverability, mobile wireless networks, mobility handoff.

## ACRONYMS<sup>1</sup>

Cdf	cumulative distribution function
MH	mobile host

## NOTATION

$F_r(t)$	Cdf of the recovery time
$f_f(t)$	Pdf of the failure time of the MH
$N(t)$	The number of log entries accumulated since the previous checkpoint
$T_1$	Time required to load a log entry (through a wireless channel)
$T_2$	Time required to execute a log entry
$T_3$	Time required to load the last checkpoint (through a wireless channel)
$T_f$	The time duration from the beginning to the MH failure time
$T_R$	Failure recovery time
$T_c$	Checkpoint interval

Manuscript received March 15, 2003; revised August 15, 2003. This work is supported in part by a National Science Foundation grant #9987586, a Microsoft Research grant, and an Intel grant. Associate Editor: J.-C. Lu.

I.-R. Chen, B. Gu, and S. E. George are with the Department of Computer Science, Virginia Tech, Northern Virginia Center, Falls Church, VA 22043 USA (e-mail: irchen@vt.edu; bgu@vt.edu; sgeorge@vt.edu).

S.-T. Cheng is with the Department of Computer Science & Information Engineering, National Cheng Kung University, Tainan, Taiwan (e-mail: stcheng@mail.ncku.edu.tw).

Digital Object Identifier 10.1109/TR.2004.837518

<sup>1</sup>The singular and plural of an acronym are always spelled the same.

$t_f$	Failure time of the MH
$r$	Ratio of the transfer time in the wired network to the transfer time in the wireless network
$\lambda$	Log arrival rate
$\delta$	MH failure rate
$\sigma$	MH mobility rate

## I. INTRODUCTION

WITH THE proliferation and ubiquity of handheld devices and wireless networks, it is anticipated that mobile applications will become a norm in a very near future. This paper concerns failure recovery of mobile client-server applications for the case in which the client mobile host crashes or fails due to battery power, memory exhaustion, or lack of resources (e.g., bandwidth) in mobile wireless environments. Unlike a workstation in a fixed network, a MH engaged in a client-server application in a wireless network can easily fail because of limited resources.

The subject of failure recovery of mobile client-server applications has received considerable attention in recent years [1]–[6] due to its potential applicability. The center of attention, however, has been in the design and validation of checkpoint and message logging protocols for saving the execution state of a mobile application, such that when a MH recovers from a failure, the mobile application can roll back to the last saved state, and restart execution with recovery guarantees. Less attention has been paid to study the reliability and real-time recoverability behaviors of these protocols. The existing protocols discern themselves in the issue of where to store checkpoint and logging information, with most of them assuming that the MH's disk storage is not stable (i.e., no hard disk), and thus checkpoint and logging information will be stored at base stations in the wireless network [1], [2].

Two broad categories of mobile checkpoint protocols have been proposed in the literature thus far, namely, coordinated and uncoordinated. In the coordinated protocols presumably involving multiple MH, the participating MH must coordinate their local checkpoints to ensure a consistent and recoverable global checkpoint. A consistent global checkpoint is selected from all these local states during the recovery process [3]. In the uncoordinated protocol, presumably more applicable to mobile applications involving only a single client MH, the MH can independently checkpoint its local state. Pradhan, Krishna, and Vaidya [1] proposed two uncoordinated checkpoint protocols: No-logging versus Logging. The No-logging approach requires the MH to create a new checkpoint every time it receives a message or a user input (called a write-event) which modifies the state of the application. The Logging approach, on the

other hand, creates checkpoints only periodically, and logs all write-events which occur in between two checkpoints to save the checkpoint cost, at the expense of incurring more recovery time in the event of failure. When a MH recovers from a failure, it will retrieve the checkpoint along with log entries saved from the base stations to start the recovery. Performance analysis of Logging versus No-logging was reported [1], but no analysis was given on the recovery time behavior.

In this paper we specifically address the failure recoverability aspect of mobile applications. In particular, we are interested in obtaining the failure recovery time probability distribution function analytically, based on system parameters characterizing a mobile client-server application, reflecting the design of specific protocols. The resulting recovery time function, once obtained, can be utilized to compute the failure recoverability, as a function of time for time-sensitive mobile applications based on a particular protocol.

While performing such an analysis is generally applicable to all protocols, we illustrate the analysis techniques to the logging approach discussed above as we perceive that the cost of no-logging is often too high to realize a mobile application. Also, because a MH can move from one base station to another while it executes a client-server application, we also specifically consider the effect of two different mobility handoff strategies, namely Eager and Lazy, on the failure recoverability. Under the Eager mobility handoff strategy [1], we always keep the logging and checkpoint information in the base station under which the MH currently resides. Thus, when the MH moves from one base station to another during the execution of a mobile application, all the checkpoint and logging information must be moved to the current base station as well. The advantage of this approach is fast failure recovery. Under the Lazy strategy, on the other hand, we do not move the checkpoint and logging information as the MH moves. Rather, a forwarding pointer is established from the current base station to the last base station so that when a failure occurs, the checkpoint and logging information of the mobile application can be recovered from all the base stations on the forwarding chain by following the links. Obviously, how effective these two strategies would be depends on various system parameters, including the checkpoint rate, logging message arrival rate, user mobility rate, failure rate, and bandwidth, among others. We aim to quantify the effects of these model parameters on the failure recoverability of mobile applications by obtaining the recovery time probability distribution function expressed in terms of these system parameters.

The rest of the paper is organized as follows. Section II presents our system model, and states the assumptions used in the paper. Section III gives a probabilistic analysis for Eager and Lazy mobility handoff strategies, when combined with the logging strategy, and derives expressions for the failure recovery time Cdf for a mobile application operating under these strategies. Section IV presents some numerical results to analyze the effect of identified system parameters on the recoverability of mobile applications operating under Eager and Lazy mobility handoff strategies. Section V analyzes failure recoverability versus cost tradeoff. Finally, Section VI discusses the applicability of analytical results obtained, and outlines some future research areas.

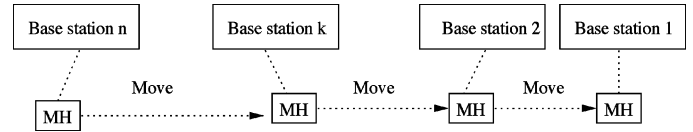


Fig. 1. A MH moving from one base station to another during a mobile application execution.

## II. SYSTEM MODEL

Our system model considers a MH, and a server, engaging in a client-server distributed application in a mobile wireless environment. To ensure failure recoverability, the application performs “logging”, and “checkpoint” activities. A checkpoint activity occurs when the application takes a snapshot of its state to save the values of state variables in a persistent storage. Between two checkpoints, the application records incremental state changes by performing logging activities. A logging activity occurs when a “write-event” occurs which changes the state of the application, i.e., when the MH receives an input from the user or a response from the server. Typically, logging activities create log entries in a persistent storage to incrementally record state changes since the last checkpoint caused by write-events.

We make the assumption that the storage space at the MH is volatile, and unsafe. Thus, the persistent state information (i.e., checkpoint and logging information) will be kept at the base stations. The MH periodically checkpoints its state, resulting in new checkpoint information stored at the current base station. Between two checkpoint events, each write-event is logged at the base stations. When the MH fails, and subsequently recovers, it reads this persistent information stored at the base stations to roll back to a state saved at the last checkpoint, and then re-executes write-events through reading log entries saved after the last checkpoint.

Fig. 1 shows a scenario in which a MH moves from one base station to another during the execution of a mobile application. In the paper, we will use “cells” and “base stations” interchangeably. A “mobility handoff” event occurs as the MH move across a cell boundary, which necessitates the new base station to know how to access the checkpoint and logging information kept at previous base station(s) for failure recovery purposes. When a mobility handoff occurs, two mobility handoff strategies are considered regarding the storage of the last checkpoint and logging information afterward as follows:

- **Eager Strategy:** When the MH moves from cell  $k$  to cell  $k - 1$ , all the persistent information (the last checkpoint and message logs afterward) are transferred into the new base station at cell  $k - 1$ . So when the MH fails, the persistent information can be found at the current base station. Consequently, there is only one base station involved in failure recovery.
- **Lazy Strategy:** When the MH crosses a base station boundary, say from cell  $k$  to cell  $k - 1$ , no persistent information is transferred between the two base stations involved. Instead, a forwarding link is setup between base station  $k - 1$  and base station  $k$ . When the MH moves from base station  $n$  to base station 1, a linked list is formed with the length of  $n - 1$ . Consequently, there will be  $n$  base stations involved in failure recovery, and the persistent information will be scattered in the base stations on the forwarding chain.

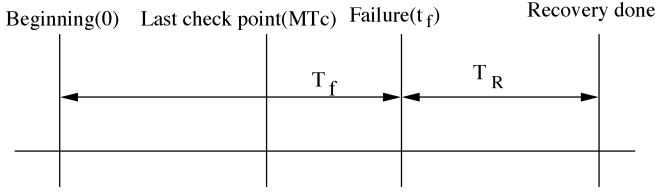


Fig. 2. Time diagram for eager mobility handoff strategy.

### III. ANALYSIS OF FAILURE RECOVERY PROBABILITY

We follow the scenario shown in Fig. 1 as we unfold our analysis. Namely, the MH starts from base station  $n$ , and moves across several cell boundaries to reach base station 1, at which a failure of the MH occurs. We consider three types of events during the execution of the mobile application before the MH fails:

- Boundary crossing events which occur when the MH moves from one cell to another.
- Checkpoint events which occur periodically.
- Message logging events which occur when an application write-event arrives.

#### A. Eager Mobility Handoff Strategy

Under the Eager mobility handoff strategy, the persistent information is always stored at the current base station, because the persistent information is either transferred from the previous base station or created in the current base station during the execution of the mobile application. When a failure occurs, the recovery time is the sum of (a) the loading time of the checkpoint and log entries from the current base station, and (b) the execution time of the log entries at the MH.

Let the number of log entries accumulated in the current base station since the last checkpoint be  $N(t)$ , where  $t$  is the time of failure of the MH. We first note that after a checkpoint operation is performed, all log entries before the checkpoint will be purged, and the number of log entries  $N(t)$  will be reset to 0. Thus, if the failure time interval  $T_f$  is longer than the checkpoint interval  $T_c$ , the MH will only re-execute those log entries accumulated past the last checkpoint. Let  $F_r(T)$  be the Cdf of the recovery time of the mobile application. Then,

$$\begin{aligned}
 F_r(T) &= \Pr\{T_R \leq T\} \\
 &= \int_0^{+\infty} \Pr\{T_R \leq T | t_f = t\} f_f(t) dt \\
 &= \sum_{M=0}^{+\infty} \int_{MT_c}^{(M+1)T_c} \Pr\{T_R \leq T | t_f = t\} f_f(t) dt \\
 &= \sum_{M=0}^{+\infty} \int_{MT_c}^{(M+1)T_c} \Pr\{N(t)[T_1 + T_2] + T_3 \leq T\} f_f(t) dt
 \end{aligned}$$

Here the summation from  $M = 0$  to infinity accounts for the fact that  $T_f$  probabilistically may be longer than the  $T_c$  by a multiple of  $M$  checkpoint intervals, as illustrated in Fig. 2. Within each such interval, the number of log entries accumulated  $N(t)$  will be relative to the  $M^{\text{th}}$  (and last) checkpoint which occurs at  $MT_c$ . We note that in the above probability expression, the term

$N(t)[T_1 + T_2]$  accounts for the time to load and execute all log entries past the last checkpoint, while  $T_3$  accounts for the time to load the last checkpoint.

Let  $t' = t - MT_c$ . We can rewrite the expression of  $F_r(T)$  as:

$$\begin{aligned}
 F_r(T) &= \sum_{M=0}^{+\infty} \int_0^{T_c} \Pr\{N(t' + MT_c)[T_1 + T_2] + T_3 \leq T\} \\
 &\quad \times f_f(t' + MT_c) dt' \\
 &= \sum_{M=0}^{+\infty} \int_0^{T_c} \Pr\{N(t')[T_1 + T_2] + T_3 \leq T\} \\
 &\quad \times f_f(t' + MT_c) dt' \\
 &= \sum_{M=0}^{+\infty} \int_0^{T_c} \Pr\left\{N(t') \leq \frac{T - T_3}{T_1 + T_2}\right\} f_f(t' + MT_c) dt'
 \end{aligned}$$

The above expression can be further simplified when given specific information regarding the distributions of random variables. For the special case in which the message logs arrive as a Poisson process with arrival rate  $\lambda$ , and the MH failure time is exponentially distributed with failure rate  $\delta$ . Then,

$$\begin{aligned}
 \Pr\left\{N(t') \leq \frac{T - T_3}{T_1 + T_2}\right\} &= \sum_{n=0}^{\lfloor \frac{T - T_3}{T_1 + T_2} \rfloor} \Pr\{N(t') = n\} \\
 &= \sum_{n=0}^{\lfloor \frac{T - T_3}{T_1 + T_2} \rfloor} \frac{e^{-\lambda t'} (\lambda t')^n}{n!} \\
 f_f(t' + MT_c) &= \delta e^{-\delta(t' + MT_c)}
 \end{aligned}$$

Thus, in this special case, the Cdf of the recovery time,  $F_r(T)$ , is given as:

$$\begin{aligned}
 F_r(T) &= \sum_{M=0}^{+\infty} \int_0^{T_c} \sum_{n=0}^{\lfloor \frac{T - T_3}{T_1 + T_2} \rfloor} \frac{e^{-\lambda t'} (\lambda t')^n}{n!} \cdot \delta e^{-\delta t'} \cdot e^{-\delta MT_c} dt' \\
 &= \frac{\int_0^{T_c} \sum_{n=0}^{\lfloor \frac{T - T_3}{T_1 + T_2} \rfloor} \frac{e^{-\lambda t} (\lambda t)^n}{n!} \cdot \delta e^{-\delta t} dt}{1 - e^{-\delta T_c}}
 \end{aligned} \tag{1}$$

#### B. Lazy Mobility Handoff Strategy

With the Lazy handoff strategy, a forwarding chain of base stations visited by the MH will be formed as a result of the MH moving across cell boundaries during the execution of the mobile application. As a design tradeoff compared with the Eager strategy, the Lazy strategy simplifies the handoff process at the expense of slower failure recovery should a failure occur during the mobile application execution. To recover from a MH failure, we have to transfer all log entries distributed over multiple base stations past the last checkpoint along the forwarding chain, and the last checkpoint to the MH. Fig. 3 shows a scenario in which the MH crosses a cell boundary after the last checkpoint before it fails. In this case, the last checkpoint will be stored in the last base station (before the crossing), and all log entries will be distributed between the last, and current base stations.

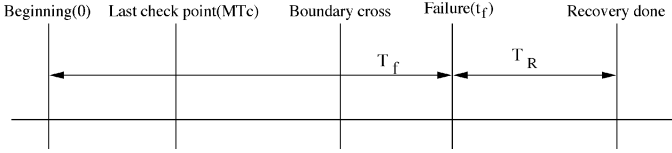


Fig. 3. Time diagram for lazy mobility handoff strategy.

Suppose that the number of base stations visited by a MH is  $k$  since the last checkpoint. The recovery process will involve (a) transferring the last checkpoint and log entries distributed among the  $k$  base stations to the last base station (that is, station 1 as shown in Fig. 1) via the wired network, (b) transferring the last checkpoint with all log entries from the last base station to the MH, and (c) the re-execution of the log entries. The last two steps are the same as in the Eager strategy. When a checkpoint is made, all the old log entries stored in the base stations along the forwarding chain will be purged with the forwarding chain length reset to 0. Therefore, the number of base stations on the forwarding chain at the recovery time depends on the amount of failure time past the last checkpoint. Let  $t$  be the failure time of the MH. Let  $MT_c$  be the  $M^{\text{th}}$  and last checkpoint instant relative to the failure time. Then the number of base stations on the forwarding chain at the recovery time depends on  $t - MT_c$ . Thus,

$$\begin{aligned}
 F_r(T) &= \Pr\{T_R \leq T\} \\
 &= \int_0^{+\infty} \Pr\{T_R \leq T | t_f = t\} f_f(t) dt \\
 &= \sum_{M=0}^{+\infty} \int_{MT_c}^{(M+1)T_c} \Pr\{T_R \leq T | t_f = t\} f_f(t) dt \\
 &= \sum_{M=0}^{+\infty} \int_{MT_c}^{(M+1)T_c} \sum_{k=0}^{+\infty} \Pr\{T_R \leq T | t_f = t, k_b = k\} \\
 &\quad \times \Pr\{k_b = k | t_f = t\} f_f(t) dt
 \end{aligned}$$

Here  $k_b$  is a random variable representing the number of base stations crossed by the MH past the last checkpoint given that the failure time is  $t$ . Let  $t' = t - MT_c$  be the amount of failure time past the  $M^{\text{th}}$  and last checkpoint. Then  $F_r(T)$  can be rewritten as

$$\begin{aligned}
 F_r(T) &= \sum_{M=0}^{+\infty} \int_0^{T_c+M T_c} \sum_{k=0}^{+\infty} \Pr\{T_R \leq T | t_f = t' + MT_c, k_b = k\} \\
 &\quad \times \Pr\{k_b = k | t_f = t' + MT_c\} f_f(t' + MT_c) dt' \\
 &= \sum_{M=0}^{+\infty} \int_0^{T_c+M T_c} \sum_{k=0}^{+\infty} \Pr\{r[kT_1 + N(t')T_1 + T_3] \\
 &\quad + N(t')[T_1 + T_2] + T_3 \leq T\} \\
 &\quad \times \Pr\{k_b = k | t_f = t' + MT_c\} f_f(t' + MT_c) dt' \\
 &= \sum_{M=0}^{+\infty} \int_0^{T_c+M T_c} \sum_{k=0}^{+\infty} \Pr\left\{N(t') \leq \frac{T - T_3 - rT_3 - rkT_1}{rT_1 + T_1 + T_2}\right\} \\
 &\quad \times \Pr\{k_b = k | t_f = t' + MT_c\} f_f(t' + MT_c) dt'
 \end{aligned}$$

where in the second expression it is given that

- 1) there are  $k$  base stations and  $N(t')$  log entries accumulated past the last checkpoint; and
- 2) the recovery time  $T_R$  is the sum of
  - a. the time to inform all  $k$  stations on the forwarding chain of the recovery process ( $rkT_1$ ),
  - b. the time to transfer all log entries past the last checkpoint and the last checkpoint itself to the current base station ( $rN(t')T_1 + rT_3$ ),
  - c. the time to load the last checkpoint to the MH ( $T_3$ ), and
  - d. the time to load and execute all log entries past the last checkpoint at the MH ( $N(t')T_1 + N(t')T_2$ ).

The expression for  $F_r(T)$  can be further simplified if we have knowledge about specific distributions of certain random variables. Consider the special case in which the message logs arrive as a Poisson process with arrival rate  $\lambda$ , the MH failure time is exponentially distributed with failure rate  $\delta$ , and the residence time is exponentially distributed with rate  $\sigma$ . Then,

$$\begin{aligned}
 \Pr\left\{N(t') \leq \frac{T - T_3 - rT_3 - rkT_1}{rT_1 + T_1 + T_2}\right\} \\
 &= \sum_{n=0}^{\lfloor \frac{T - T_3 - rT_3 - rkT_1}{rT_1 + T_1 + T_2} \rfloor} \Pr\{N(t') = n\} \\
 &= \sum_{n=0}^{\lfloor \frac{T - T_3 - rT_3 - rkT_1}{rT_1 + T_1 + T_2} \rfloor} \frac{e^{-\lambda t'} (\lambda t')^n}{n!} \\
 f_f(t' + MT_c) &= \delta e^{-\delta(t' + MT_c)}
 \end{aligned}$$

and

$$\Pr\{k_b = k | t_f = t' + MT_c\} = \frac{e^{-\sigma t'} (\sigma t')^k}{k!}$$

Consequently, in this special case, the Cdf of the recovery time,  $F_r(T)$ , is given by

$$\begin{aligned}
 F_r(T) &= \sum_{M=0}^{+\infty} \int_0^{T_c+M T_c} \sum_{k=0}^{\lfloor \frac{T - T_3 - rT_3 - rkT_1}{rT_1 + T_1 + T_2} \rfloor} \sum_{n=0}^{\lfloor \frac{T - T_3 - rT_3 - rkT_1}{rT_1 + T_1 + T_2} \rfloor} \\
 &\quad \times \frac{e^{-\lambda t'} (\lambda t')^n}{n!} \cdot \frac{e^{-\sigma t'} (\sigma t')^k}{k!} \cdot \delta e^{-\delta t'} \cdot e^{-\delta M T_c} dt' \\
 &= \frac{\int_0^{T_c+M T_c} \sum_{k=0}^{\lfloor \frac{T - T_3 - rT_3 - rkT_1}{rT_1 + T_1 + T_2} \rfloor} \sum_{n=0}^{\lfloor \frac{T - T_3 - rT_3 - rkT_1}{rT_1 + T_1 + T_2} \rfloor} \frac{e^{-\lambda t'} (\lambda t')^n}{n!} \cdot \frac{e^{-\sigma t'} (\sigma t')^k}{k!} \cdot \delta e^{-\delta t'} dt'}{1 - e^{-\delta T_c}}
 \end{aligned} \tag{2}$$

#### IV. NUMERICAL RESULTS

In this section, we report numerical results generated by MATLAB based on (1) and (2) derived to analyze the effects of various parameters on failure recoverability of mobile applications operating under the Eager and Lazy handoff strategies in combination with the logging strategy. We vary the values of application-dependent model parameters, such as the log arrival rate ( $\lambda$ ), MH failure rate ( $\delta$ ), checkpoint interval ( $T_c$ ), MH mobility rate ( $\sigma$ ), and transmission ratio between wired and

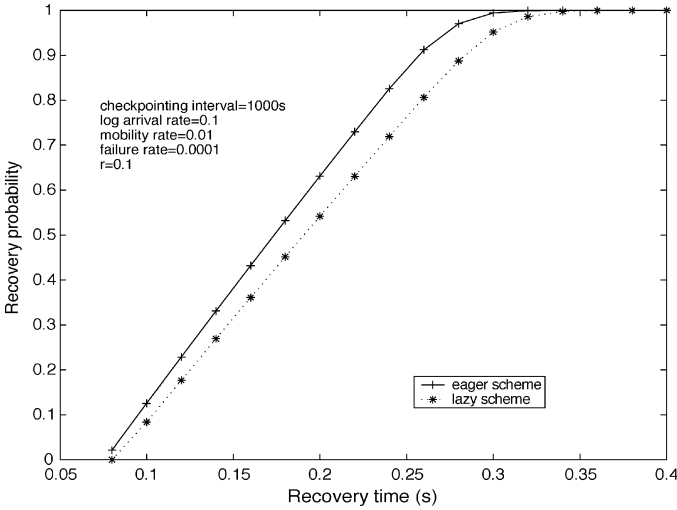


Fig. 4. Recovery probability of eager and lazy strategies.

wireless communication ( $r$ ), to analyze their effects. Without loss of generality, we consider a mobile application for which the time required to execute a log entry,  $T_2$ , is 0.0001 second, the size of a log entry is 50 bytes, and the size of a checkpoint is 2000 bytes. Also consider that the data transfer rate of the wired network in the network infrastructure is 2 Mbps (thus the wireless network data rate is  $2r$  Mbps). Thus, the values of  $T_1$ , and  $T_3$  are calculated as  $0.0002/r$ , and  $0.008/r$  seconds, respectively.

We first compare the Cdf of the failure recovery time (or the recovery probability) of the mobile application operating under Eager and Lazy mobility handoff strategies in Fig. 4 where  $\lambda = 0.1$ ,  $\delta = 0.0001$ ,  $T_c = 1000$ ,  $\sigma = 0.01$ , and  $r = 0.1$ . As expected, the failure recovery probability under the eager strategy is always better than that under the lazy strategy for the same recovery time elapsed, due to the fact that an extra step is required to transfer distributed log entries and checkpoint information to the current base station for recovery for the eager strategy. However, we note that the recovery probability of Eager is much higher than that of the Lazy strategy when the recovery time  $T$  is in the range  $[0.15, 0.3]$ , and the recovery probability difference between the two strategies is negligible when  $T < 0.15$  or  $T > 0.3$ . This indicates that when given enough recovery time ( $T > 0.3$ ) in this case, the failure recoverability offered by the less costly lazy strategy is just as good as the more costly eager strategy.

Fig. 5 shows the effect of log arrival rate  $\lambda$  on failure recoverability for the case when  $T = 0.24$  s, with other parameters kept at their values as before. As observed, the system recovery probability decreases dramatically as the log arrival rate  $\lambda$  increases. For the eager strategy, the recovery probability is close to 1.0 when the arrival rate is less than or equal to 0.05. As the log arrival rate increases to 0.15, the recovery probability reduces to 0.56. A similar trend is observed for the lazy strategy. This indicates that the recovery probability is very sensitive to the log arrival rate. Another observation is that the recovery probability difference between the two handoff strategies is negligible when the log arrival rate is low, because not too many log entries would have accumulated past the last checkpoint.

Another observation exhibited in Fig. 5 is that the effect of mobility rate ( $\sigma$ ) on the recovery probability is marginal in this

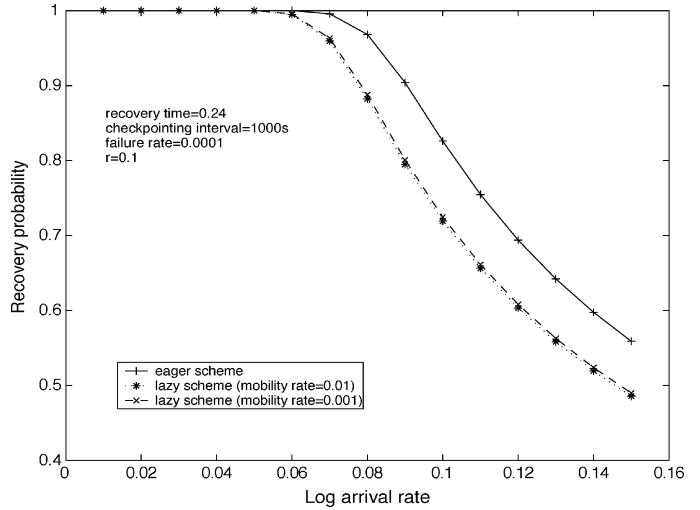


Fig. 5. Effect of log arrival rate and mobility rate.

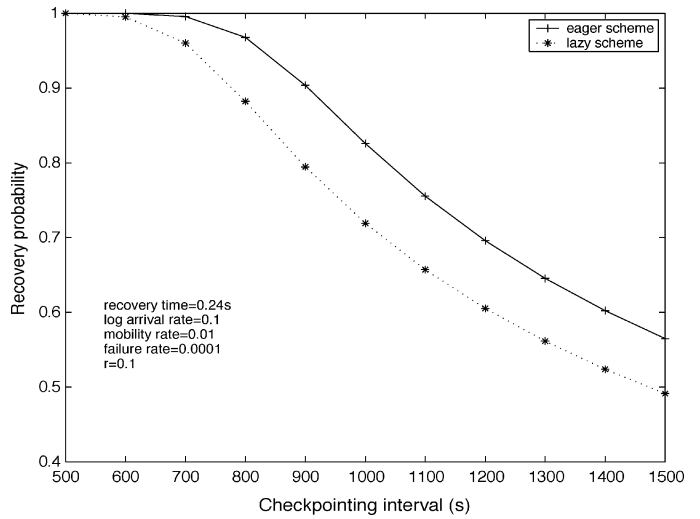


Fig. 6. Effect of checkpoint interval.

case. The effect of the mobility rate on the recovery probability is manifested through the number of base stations on the forwarding chain storing all log entries past the last checkpoint. That is, the higher the mobility rate, the longer the forwarding chain, and the worse the failure recoverability. The effect of mobility rate is not sensitive in Fig. 5 due to two reasons: (1) the data transfer time in the wired network is relatively small compared to the data transfer time in the wireless network as indicated by the small  $r$  value ( $r = 0.1$ ), and (2) the checkpoint interval is small ( $T_c = 1000$ ). With a small  $T_c$ , the MH would not have crossed too many base stations when a failure occurs. Thus the forwarding chain length would be short, and the data transfer time via the wired network would be small. We expect that the effect of mobility rate would be more significant with a large  $T_c$ .

In Fig. 6, we show the effect of the checkpoint interval  $T_c$  on the recovery probability with all other parameters kept at their previous values. Here we observe that the recovery probability is also very sensitive to the checkpoint interval. As the checkpoint interval increases, more log entries would arrive in between two checkpoint intervals. This greatly increases the time for the MH

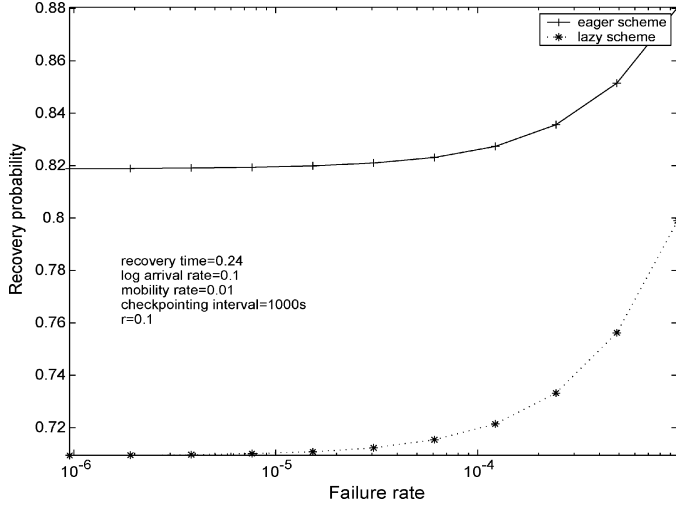


Fig. 7. Effect of failure rate.

to do recovery. Fig. 6 also shows that the difference in recovery probability at a particular recovery time between the Eager and Lazy strategy becomes more significant as the checkpoint interval increases, due to the extra time required to transfer a larger amount of log entries, and the last checkpoint from base stations on the forwarding chain to the current base station.

Finally, Fig. 7 shows the effect of failure rate ( $\delta$ ) on the recovery probability. Here we see that the higher the failure rate (toward the right of the X-coordinate), the less there will be log entries accumulated when a MH failure occurs, and thus less time is required to recover from the failure. Consequently, the recovery probability actually increases as the failure rate increases. Quite counter-intuitively, as the failure rate increases in the setting as shown in Fig. 7, the difference between the Eager and Lazy mobility handoff strategies becomes less significant, even though the Eager strategy is meant to deal with frequent failures to facilitate fast failure recovery. The reason is that if the failure rate of the MH is much higher compared with other rates, then the MH does not have too much time to accumulate log entries past the last checkpoint (or past the last recovery point), so in this case the Eager and Lazy mobility handoff strategies behave about the same in terms of failure recoverability. Of course, the difference between these two handoff strategies is dictated by the relative magnitude of the failure rate with respect to other event rates. The analysis given in the paper provides a theoretic basis for a system designer to assess the failure recovery time Cdf of a mobile application operating under a particular mobility handoff strategy, when given model parameter values characterizing the underlying mobile application and network environment, especially for those mobile applications with a time-sensitive recoverability requirement.

## V. FAILURE RECOVERABILITY VERSUS COST TRADEOFF ANALYSIS

The analysis in Section IV indicates that the Eager strategy is always better than the Lazy strategy in failure recoverability for the same recovery time  $T$  elapsed due to a higher cost invested to force the checkpoint and logging information to be placed in the current base station. In this section we analyze the tradeoff

involved between the cost invested for maintaining the checkpoint and logging information (due to the transfer cost in response to mobility handoff events during the failure-free period) versus the resulting recovery probability gained when a failure occurs. Our objective is to identify conditions under which the cost invested for the maintenance of the checkpoint and logging information by the Eager strategy will be the most effective in terms of the resulting recovery probability gained over that by the Lazy strategy. When given a set of model parameter values characterizing the mobile application and network environment, we wish to identify the best checkpoint interval under which the system should perform checkpoint activities, such that the cost invested by the eager mobility handoff strategy has the best investment return, i.e., the difference in recovery probability between the Eager and Lazy strategies is the highest.

We define a *Failure Recoverability versus Cost Ratio (FRCR)* parameter as the ratio of the difference in recovery probability to the difference in cost invested by these two strategies. That is, let  $C_{eager}$  and  $C_{lazy}$  be the costs invested; and  $P_{eager}$  and  $P_{lazy}$  be the corresponding recovery probabilities gained, respectively, by the Eager, and Lazy strategies, when given a set of parameter values. Then,

$$FRCR = \frac{P_{eager} - P_{lazy}}{C_{eager} - C_{lazy}}$$

Note the physical meaning of  $FRCR$  is equal to the slope of the recovery probability gained versus the cost invested, so a high  $FRCR$  means that the return investment in recovery probability with respect to the cost invested by the Eager strategy over the Lazy strategy is high, and vice versa.

When given a set of parameter values, we can calculate  $P_{eager}$  and  $P_{lazy}$  (and thus  $P_{eager} - P_{lazy}$ ) easily at a given recovery time  $T$  using the Cdf expressions derived in Section III, e.g.,  $P_{eager} = F_r(T)$  for the Eager strategy based on (1). For computing  $C_{eager} - C_{lazy}$ , on the other hand, we note that the cost difference between the Eager and Lazy strategies is due to the transfer cost of log entries, and the last checkpoint, which incurs when the mobile user moves across a boundary. More specifically, when a MH move event occurs, the Eager strategy will move all log entries after the last checkpoint, together with the last checkpoint, to the current base station, while the lazy strategy will only establish a link between the last base station, and the current base station without moving logs, and the last checkpoint.

We note that the average number of checkpoints before failure is given by:

$$\frac{1}{\delta T_c}$$

The average number of moves crossing base station boundaries before two consecutive checkpoints is given by:

$$T_c \times \sigma$$

The number of log entries accumulated between two consecutive moves is given by:

$$\frac{1}{\sigma} \frac{1}{\lambda}$$

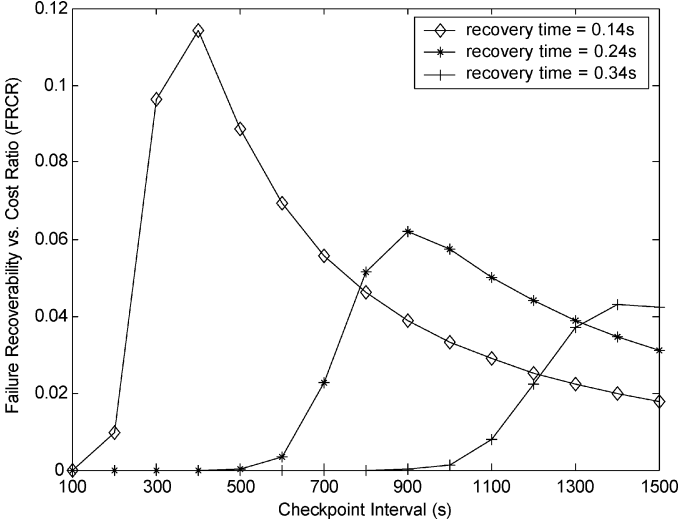


Fig. 8. Failure recoverability versus cost ratio (FRCR) as a function of checkpoint interval.

Thus, the total number of log-entry transfer operations required by the eager mobility handoff strategy between two consecutive checkpoints is given by:

$$\sum_{n=1}^{T_c \times \sigma} \frac{1}{\lambda} \times n$$

In the above expression,  $n$  represents the  $n$ th move among a total of  $T_c \sigma$  moves which occur between two consecutive checkpoints. Summarizing above, the cost invested by the eager mobility handoff strategy before failure is given by:

$$C_{\text{eager}} = \frac{1}{T_c} \times \left( r \times T_3 \times T_c \times \sigma + r \times T_1 \times \sum_{n=1}^{T_c \times \sigma} \frac{1}{\lambda} \times n \right) \quad (3)$$

Here, the first term inside the parenthesis represents the communication cost for transferring the last checkpoint, and the 2nd term represents the cost for transferring the log entries since the last checkpoint as a result of  $T_c \sigma$  moves which occur between two consecutive checkpoints.

For the Lazy strategy, the cost invested before failure is simply the cost per move, namely, the communication cost for setting up the link, denoted by  $C_p$ , multiplied with the total number of moves before failure, *viz.*,

$$C_{\text{lazy}} = \frac{1}{T_c} \times (T_c \times \sigma \times C_p) \quad (4)$$

Utilizing (1) and (2) for calculating  $P_{\text{eager}} - P_{\text{lazy}}$ , and (3) and (4) for calculating  $C_{\text{eager}} - C_{\text{lazy}}$ , we can obtain  $FRCR$  when given a set of model parameter values characterizing the mobile application and network environment.

Fig. 8 shows  $FRCR$  as a function of the checkpoint interval  $T_c$ , when given the same set of parameter values as in Section IV along with  $C_p = 0.0004$  sec. representing the communication cost to set up a forwarding link upon a move event under the Lazy mobility handoff strategy (assuming a 10-byte packet is used for this purpose), for several recovery time values.

We observe that there exists a best checkpoint interval  $T_c$  under which the Eager strategy is most cost-effective over the

Lazy strategy in terms of the amount of improvement in failure recoverability with respect to the cost invested for maintaining the logging and checkpoint information. The reason is that when the checkpoint interval is very short, all log entries since the last checkpoint as well as the last checkpoint itself are likely to reside in the current base station, making the failure recoverability of both strategies virtually the same. As the checkpoint interval increases, the number of log entries accumulated between two consecutive checkpoints becomes more substantial, thus resulting in an increase in  $FRCR$  due to a larger difference in failure recoverability between the eager and lazy strategies with respect to the difference in the cost invested for maintaining the logging and checkpoint information. When the checkpoint interval is very long, however, the improvement in failure recoverability cannot catch up with the increase in the cost investment difference, thus resulting in a decline in  $FRCR$ . Here we also observe that the best cost-effective checkpoint interval for the eager strategy increases as the recovery time increases, because a larger value of recovery time relaxes the failure recoverability requirement, thus allowing a longer checkpoint interval to be used by either strategy to satisfy the requirement. Consequently, a larger value of recovery time dictates a larger checkpoint interval for the cost-effectiveness of the eager strategy over the lazy strategy to be observed.

## VI. APPLICABILITY AND FUTURE RESEARCH

We have derived closed-form expressions for the failure recovery time distribution of a client-server application in mobile wireless environments for both Eager, and Lazy, mobility handoff strategies; and exemplified the utility of the derived functions using Poisson distributions as a special case. The failure recovery time distributions derived, nevertheless, allows a system designer to clearly see the effects of these two handoff strategies on failure recoverability of a mobile application in terms of the time taken to reach a target recovery probability, and also to examine the effects of various parameters on failure recoverability.

Our numerical data obtained showed that if the target recovery probability is either very low or very high, if the log arrival rate is very low relative to the checkpoint rate or failure rate, or if the failure rate is very high, then the improvement on the recovery probability brought by the eager strategy is insignificant. In view of the high cost invested under the eager strategy to move checkpoint and logging information from one base station to another whenever the MH crosses a base station boundary, the system designer may decide to adopt the lazy mobility handoff strategy, especially if after inspecting the failure recovery time distribution of the lazy strategy derived in the paper it is found that the lazy mobility handoff strategy can also satisfy the target failure recoverability requirement. In light of this result, we have analyzed the tradeoff involved between the cost invested for maintaining checkpoint and logging information (that is, the transfer cost in the event of mobility) versus the recovery cost, thereby revealing conditions under which the eager mobility handoff strategy should be applied over the lazy mobility handoff strategy to obtain the largest cost investment return in terms of improved failure recoverability.

A future research direction is to consider adopting a more sophisticated probabilistic model (e.g., a Stochastic Petri net

model) to analyze the combined effect of various checkpoint strategies, including cumulative-log-based, and mobility-based (or distance-based); and mobility handoff strategies for maintaining logging and checkpoint information (such as a hybrid strategy in addition of eager and lazy), on the failure recoverability of specific mobile applications, such as those based on various architectural models for executing mobile transactions.

#### REFERENCES

- [1] D. K. Pradhan *et al.*, "Recovery in mobile environments: design and trade-off analysis," in *26th IEEE Fault-Tolerant Computing Symp.*, 1996, pp. 16–25.
- [2] A. Acharya and B. R. Badrinath, "Checkpoint distributed applications on mobile computers," in *Proc. 3rd Int. Conf. Parallel and Distributed Information Systems*, Austin, Texas, 1994, pp. 73–80.
- [3] N. Neves and W. K. Fuchs, "Adaptive recovery for mobile environments," *Commun. ACM*, vol. 40, no. 1, pp. 69–75, 1997.
- [4] R. Prakash and M. Silghal, "Low-cost checkpoint and failure recovery in mobile computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 7, no. 10, pp. 1035–1048, 1996.
- [5] B. Yao *et al.*, "Message logging in mobile computing," in *29th IEEE Fault-Tolerant Computing Symp.*, 1999, pp. 294–301.
- [6] C. Pedregal-Martin and K. Ramamritham, "Support for recovery in mobile systems," *IEEE Trans. Comput.*, vol. 51, no. 10, pp. 1219–1223, 2002.

**Ing-Ray Chen** received the B.S. degree from the National Taiwan University, Taipei, Taiwan, and the M.S. and Ph.D. degrees in Computer Science from the University of Houston, Texas. He is currently an Associate Professor in the Department of Computer Science at Virginia Tech. His research interests include mobile computing, pervasive computing, multimedia, distributed systems, real-time intelligent systems, and reliability and performance analysis. Dr. Chen has served on the program committee of numerous conferences, including being as Program Chair of the 29th IEEE Annual Computer and Software Application Conference in 2005, 14th IEEE International Conference on Tools with Artificial Intelligence in 2002, and the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology in 2000. Dr. Chen currently serves as an Associate Editor for IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *The Computer Journal*, and *International Journal on Artificial Intelligence Tools*. He is a member of the IEEE/CS and ACM.

**Baoshan Gu** received the B.S. degree from University of Science and Technology of China, Hefei, China, in 1992 and the M.S. degree in Computer Science from Institute of Computing Technology, Chinese Academia of Science, Beijing, China, in 1995. From 1995 to 2000, he was a Research and Development Engineer with the Institute of Computing Technology, Chinese Academia of Science. He is currently pursuing his Ph.D. degree in the Department of Computer Science, Virginia Tech, where he is a Research Assistant in the Systems and Software Engineering Laboratory. His research interests include next-generation wireless system architectures, design and evaluation of location and service management schemes in mobile computing environments, and mobile multimedia systems.

**Sapna E. George** received the Bachelors of Technology degree in Electronics Engineering from Cochin University of Science and Technology, India, and the M.S. degree in Computer Science at Virginia Tech. She is currently a Ph.D. student in the Department of Computer Science, Virginia Tech. Her research interests are in mobile transactions, algorithms, and modeling and evaluation of computer systems.

**Sheng-Tzong Cheng** received the B.S. in 1985 and M.S. in 1987 in Electrical Engineering from the National Taiwan University, Taipei, Taiwan. He received the M.S. in 1993 and Ph.D. in 1995 in Computer Science from the University of Maryland, College Park, MD. He was an Assistant Professor of Computer Science and Information Engineering at the National Dong Hwa University, Hualien, Taiwan, in 1995, and became an Associate Professor in 1996. He is currently an Associate Professor in the department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. His research interests are in design and performance analysis of mobile computing, wireless communications, multimedia, and real-time systems.