Region-based Recovery for Reliable Mobile Computing Environments

Taesoon Park

***Abstract*—Fault tolerance is a very important design issue to build a mobile computing system. In this paper, a checkpointing recovery scheme suitable for a mobile computing system is proposed. The proposed scheme considers the movement patterns of mobile hosts in the system. For each pattern, the migration of the recovery information is restricted, partially allowed or fully allowed. As a result, the proposed scheme tries to balance the cost for the failure-free operation and the cost for the failure recovery. The performance of the proposed scheme is evaluated with the extensive simulation study.**

***Index Terms*—checkpointing, logging, mobile computing system, region-based management, rollback-recovery.**

I. Introduction

Many distributed applications are nowadays extended to continue their services in the mobile computing environment [1]. However, simple extension of the existing distributed algorithms is not well fixed to the mobile environment, since the mobile system introduces a new challenge in the handling of mobile hosts. The mobile host (MH) keeps moving from a cell to another cell; and it is connected to a mobile support station (MSS) via a wireless network with the low bandwidth and the very fragile connection. The MH also carries small memory and disk spaces, and its low battery capacity sometimes requires the disconnected operation for the power saving.

Checkpointing-recovery is a distributed service for a system to cope with failures. Considering the MHs vulnerable to failures, it is desirable for the mobile computing system to be equipped with a proper recovery facility. However, many distributed recovery schemes cannot be directly used for the mobile environment. For example, coordinated checkpointing schemes [2, 3] consume the network bandwidth with too many coordination messages and the frequent checkpointing of communication-pattern based schemes [4] may not be affordable with the low computing power of MHs. Communication-induced checkpointing [5, 6] may be the one with less checkpointing overhead, however, considering the recovery, all of these checkpointing-only schemes have a problem of recursive rollbacks, unless the rollbacks of the related MHs are fully synchronized.

Considering the MHs frequently disconnected from the network without a failure and the expensive coordination cost, asynchronous recovery without any coordination must be sought. For the asynchronous recovery, message logging can be used with independent checkpointing; and for the logging, pessimistic [7, 8], optimistic [9], and causal schemes [10] are used. For the most logging schemes, the stable storages of MSSs are used to store checkpoints and message logs of a MH, due to the lack of spaces in MHs. Hence, as an MH moves around the cells, the storages of the recovery information become dispersed over a number of MSSs, and in case of a failure, the MH must locate the MSSs carrying the proper checkpoint and message logs.

For the instant recovery from a failure, checkpoints and logs should be near the current location of an MH, however, migration of checkpoints and logs distributed over the network may cause some severe network traffic. Hence, efficient management of the distributed recovery information becomes an important design issue to implement checkpointing-recovery schemes for mobile environments. For fast recovery, an MH in [7] carries checkpoints and logs as it moves, and the suggestion made in [8] utilizes the home of each MH as a centralized recovery information manager.

In this paper, a distributed storage management scheme using the region-based scheme is presented. The region-based recovery schemes presented in [11] consider the movements of MHs within a region and between the regions. Hence, frequent migration of recovery information between the regions causes some severe network costs during the failure-free operation, however, this guarantees fast recovery in case of a failure. On the other hand, no migration of checkpoints and logs within a region guarantees low overhead during the normal operation, however, it causes very slow recovery for the failed MH.

Hence, in the proposed scheme, unlike the previous region-based schemes, movements of MHs are categorized into three patterns; moving within a region, moving around neighbor regions, moving across regions. When an MH moves within a small range, the migration of recovery information is restricted or partially allowed. However, when an MH moves far from the previous region, the recovery information is migrated to the region near the MH. As a result, the cost for migration of recovery information during the failure-free operation and the recovery cost can be balanced.

Manuscript received December 10, 2013. Taesoon Park is with the Department of Computer Engineering, Sejong University, Seoul, 143-747, KOREA (phone: +822-3408-3240; fax: +822-3408-3321; e-mail: tspark@sejong.ac.kr).

The rest of this paper is organized as follows: Section II explains the system model and Section III presents the protocol for the distributed recovery information management. Section IV describes the simulation environments and compares the performance results of the proposed scheme with those of previous schemes. Section V concludes the paper.

II. System Model

A mobile computing system considered in this paper follows the model presented in [4] and [12]. The system consists of a set of static mobile support station, called *MSS*s, and a set of mobile hosts, called *MH*s. All MSSs in the system are connected with a high speed wired communication link and a wireless communication link can be established between an MH and an MSS. The service area covered by an MSS is called a *cell* and an MH residing in a cell can be connected to the MSS servicing the cell. An MH can communicate with another MH only through the local MSSs. The links in the wireless network support the FIFO communication in both directions however there is no assumption on the message delivery order by wired links.

Distributed computation performed by a set of processes running concurrently on a set of MHs is assumed to follow the piece-wise deterministic model[10], in which a process always produces the same sequence of computational states during the execution if the same sequence of message receipt events would happen at the process. Failures assumed in this paper are transient and fail-stop; that is, a process does not likely fail again at the same execution point when it recovers from a failure; and in case of a failure, the process stops its execution and does not perform any malicious action. When a failure occurs, the contents stored in the volatile memory of the MH or MSS would be lost, however, the stable storage survives the failure. In this paper, we focus on the failure-recovery of MHs, since MSSs can perform the self-recovery using the checkpoint in its own stable storage. The message transmission delay in static and dynamic networks is assumed to be finite and arbitrary.

For an MH to leave a cell and enter into another cell, it first has to end its current connection by sending a *leave(r)* message to the old MSS, where *r* is the sequence number of the last message received from the MSS, and then establish a new connection by sending *join(MH-id, previous MSS-id)* message to the new MSS. Leaving a cell and entering into another cell happens simultaneously when the MH crosses the boundary between two cells and it is called a *hand-off*.

During the hand-off, the location management activity of an MH is performed, based on the two-level data hierarchy consisting of *home location register (HLR)* and *visitor location register (VLR)*. Each MH in the system has a *home* carrying the HLR which keeps track of the MH's current location, and also, it has one or more MSSs which carry the VLR, the location information for the MH. During the hand-off of an MH, the new MSS sends a registration query to the VLR. If the MH has moved out of the area managed by the previous VLR, the new VLR sends the location update message to the HLR on behalf of the MH. With this mechanism, the HLR maintains the current location of the MH, and hence, a message sent for an MH can be delivered based on the information in HLR and VLR.

III. Region based Recovery Scheme

A. Checkpointing and Mobility Tracking

A mobile host MH_i periodically saves its current state as a checkpoint and then transfers the checkpoint to the MSS_p in which MH_i is currently residing. Let C_i^a denote the *a*-th checkpoint of MH_i. A checkpoint, C_i^a, is identified by a pair of integers, *(i,a)*, and the identifier of C_i^a is transferred with the checkpoint. Each MH_i maintains a variable, $m_i^{rcv\_seq}$, to count the number of messages MH_i has received, and the value of $m_i^{rcv\_seq}$ right before the latest checkpointing is also carried with the checkpoint. The counter value, $m_i^{rcv\_seq}$, carried with a checkpoint is used to decide the correct position of the latest checkpointing with respect to the logged messages. On the receipt of a new checkpoint and the related information, MSS_p saves them into the stable storage.

A mobile support station, MSS_p, also maintains the message log for the MHs residing in its cell. Since all the messages delivered to a mobile host MH_i in the cell are routed through MSS_p, message logging for MH_i incurs no extra communication overhead between MSS_p and MH_i. Let M_i^a denote the *a*-th message received by MH_i. The pair of integers, *(i,a)* is used as the identifier for M_i^a and each M_i^a is saved with its identifier. For the pessimistic logging, every message headed to MH_i in the cell is first saved with its message identifier into the stable storage of MSS_p, and then delivered to MH_i.

For MH_i moving from a cell to another cell to recover from a failure, it has to locate the latest checkpoint and a sequence of logged messages since the last checkpointing. To locate the MSSs carrying the latest checkpoint and the logged messages of MH_i, a data structure, called a $Trace_i$, is maintained. $Trace_i$ is an array of two integer variables, cp_{seq} and cp_{loc}, and a list, log_{set}. The *a*-th entry of $Trace_i$ corresponds to the *a*-th checkpoint interval; that is, $Trace_i[a].cp_{seq}$ and $Trace_i[a].cp_{loc}$ include the sequence number for the checkpoint, C_i^a and the identifier of the MSS carrying C_i^a, respectively. $Trace_i[a].log_{set}$ includes a set of MSSs carrying the messages logged after C_i^a was taken.

During the hand-off of MH_i, $Trace_i$ is transferred from the old MSS to the new MSS, say MSS_p, and saved into the stable storage of MSS_p. When a first message is sent from MSS_p to MH_i, MSS_p logs the message for MH_i and it includes its identifier into $Trace_i[a].log_{set}$. When MSS_p saves a checkpoint for MH_i, it creates a new entry into $Trace_i$; puts the checkpoint sequence number into $Trace_i[a].cp_{seq}$ and its identifier into $Trace_i[a].cp_{loc}$; and initializes $Trace_i[a].log_{set}$ as an empty list. For the pessimistic logging, $Trace_i$ maintains the latest entry only, since the rollback to the latest checkpoint guarantees the consistent recovery.

B. Distributed Recovery Information Management

Checkpoints and message logs in the lazy scheme for the recovery information management are dispersed over a large number of MSSs, and hence, during the recovery, a large number of messages have to be exchanged. On the other hand, the pessimistic recovery scheme requires a large size of messages have to be exchanged during the hand-off time, to carry the recovery information into the current MSS of a MH [7].

A home based scheme transfers any checkpoint or log entry to the home of an MH during the hand-off [8]. This scheme requires one message transfer carrying the recovery information for each hand-off, however, the size of each message is much smaller than that of the pessimistic scheme since checkpoints and logs are not accumulated. Also, on the recovery, MH_i needs to contact with its home only, which requires much less number of messages compared to the lazy scheme. The home based scheme, however, may not work well when the MH is far away from its home.

One way to solve this problem is to make the home near the MH. Hence, in this paper, a region-based scheme is proposed. In the proposed scheme, a number of cells are grouped into a region and one MSS in the region is assigned as a *recovery manager* (RM), which takes the role of the home for the MHs in the region. Considering the failure-free operation cost and the failure-recovery cost, variations of the region-based scheme can be considered for the implementation [11].

(1) An MH traversing within a region may or may not transfer its recovery information to the current RM.

(2) When an MH moves out of a region, it may or may not transfer the recovery information to the new RM.

Within a region, the transfer of recovery information to the RM during the hand-off can be achieved in a low cost, and in case of a recovery, the MH can retrieve the recovery information from the RM with one message exchange. However, if the MH moving in a region would fail and collect the recovery information from the current RM, there would be two moves of the recovery information, one from the MSS to the RM; and another from the RM to the failed MH. Hence in the proposed scheme to avoid any redundant migration of the recovery information, only the $Trace_i$ of MH_i is transferred to the RM during the hand-off of MH_i. When the MH moves out of a region, the RM collects the checkpoints and/or logs dispersed in the MSSs in the current region, using the information in the $Trace_i$.

Also, when an MH moves out of a region, the recovery information maintained by the previous RM can be transferred to the new RM to reduce the recovery cost, as in the pessimistic scheme; or the information can remain as it is to reduce the failure free operation cost as in the lazy scheme. If a MH would move around a small number of regions back and force, it is desirable to restrict the log transfer, considering the relatively high log transfer cost between the regions. However, if a MH traverses too far without carrying the recovery information, the recovery cost should be too high.

Hence, in the proposed scheme, the transfer of recovery information between the regions is considered depending upon the regions in which the MH moves around. For this, each RM maintains a list of identifiers of its neighbor RMs. Two RMs are called *neighbors* if any two cells in their region are adjacent. In the proposed scheme, when an MH moves out of a region, the old RM transfers the recovery information to the new RM only if they are not neighbors. Considering the fact that the message logs saved before the latest checkpoint can be eliminated in the pessimistic logging, this kind of somewhat slow recovery information transfer of the proposed scheme can reduce the unnecessary transfer cost.

To explain the region-based scheme in more detail, the following notations are used:

Let RM_m denote the recovery manager for a region with an identifier, *m*. For a mobile host, MH_i, connected to a support station, MSS_p, MSS_p first saves checkpoints and/or message logs; and updates the information in $Trace_i$ as explained before. Let $CL_{(i,p)}$ denote a set of checkpoint and message logs of MH_i managed by MSS_p. When MH_i leaves MSS_{old} and joins into another MSS, say MSS_{new}, a hand-off procedure begins by MSS_{new} sending a *hand-off_request* for MH_i (denoted by *HO_RQST(i)*) to MSS_{old}. MSS_{new} includes the identifier of its RM_{new} in *HO_RQST(i)*, so that MSS_{old} can decide if MH_i leaves its region. While the hand-off procedure is performed, the transfer of recovery information is performed as follows:

The protocol is initiated by MSS_{new} on the receipt of *join(i, MSS_{old})* from MH_i.

(1) MSS_{new} sends *HO_RQST(i)* to MSS_{old} with the identifier of its RM_{new}.

(2) On the receipt of the message, if RM_{new}=RM_{old}, the normal hand-off procedure without any log transfer is taken.

(3) If RM_{new} is not RM_{old}, MSS_{old} sends *HO_RQST(i)* and its $Trace_i$ to RM_{old} for the log transfer. Also, MSS_{old} includes the identifier of RM_{old} in the $Trace_i[a].cp_{loc}$ and $Trace_i[a].log_{set}$, before its sends $Trace_i$ to MSS_{new} for the hand-off.

(3.1) On the receipt of the message, if RM_{new} is in the neighbor list of RM_{old}, RM_{old} sends a *log-transfer-request* to every MSS_r in $Trace_i[a].cp_{loc}$ and $Trace_i[a].log_{set}$, unless MSS_r is not in the neighbor list of RM_{old}.

(3.2) On the receipt of the request, MSS_r transfers $CL_{(i,r)}$ to RM_{old}.

(3.3) On the receipt of $CL_{(i,r)}$ from every MSS_r, RM_{old} saves the information into the stable storage; and sends an *save-done(i)* to MSS_r.

(3.4) RM_{old} now deletes the identifier of every MSS_r from $Trace_i[a].cp_{loc}$ and $Trace_i[a].log_{set}$; includes its identifier in $Trace_i[a].log_{set}$; and updates $Trace_i[a].cp_{loc}$ with its identifier if it has received the latest checkpoint from any MSS_r.

(3.5) On the recipe of *save-done(i)*, MSS_r discards $CL_{(i,r)}$ and $Trace_i$.

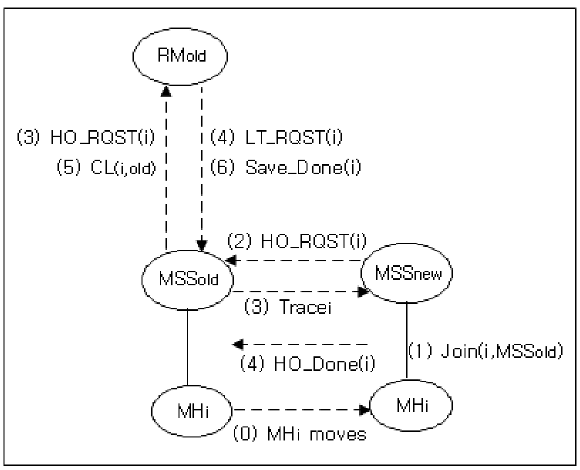
Note here that RM_{old} sends a *log-transfer-request* to every MSS_r which is not in the neighbor list of RM_{old}. Then, MSS_r is either an MSS visited by MH_i in the current region or an RM which is not a neighbor of RM_{old}. As a result, the recovery information from the RMs which are not neighbors of RM_{old} can be accumulated in RM_{old}; and the recovery cost can be reduced in case of a failure of MH_i. Also, when RM_{old} send the *log-transfer-request* to MSS_r, it sends the $Trace_i[a].cp_{seq}$ with the request so that the MSS_r can discard any old and unnecessary recovery information.

Now, we consider the case that MH_i is disconnected from the network for a while and connected to a new region which is not neighbor of the old RM. In this case, the recovery information is transferred to the new RM for the fast recovery, since the distance between two RMs can be very far.

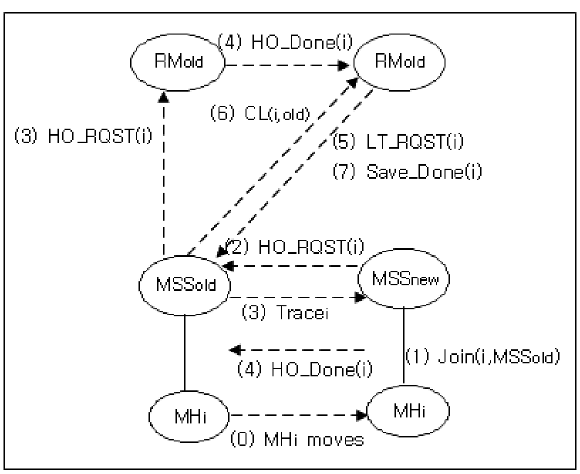
(4) On the receipt of *HO_RQST(i)* message from MSS_{old}, if RM_{new} is not in the neighbor list of RM_{old}, RM_{old} sends *HO_RQST(i)* to RM_{new} with $Trace_i$.

(4.1) RM_{new} then sends a *log-transfer-request* to every MSS_r in $Trace_i[a].cp_{loc}$ and $Trace_i[a].log_{set}$, unless MSS_r is not in the neighbor list of RM_{new}.

(4.2) RM_{new} and other related MSSs then complete the steps in (3.2)-(3.5).



(a) Recovery information retrieval within a region.



(b) Recovery information retrieval between regions.

Fig. 1. A region-based recovery information management scheme.

Fig. 1 shows the protocol for recovery information transfer during the hand-off. Fig. 1(a) is the case that RM_{old} and RM_{new} are neighbors and Fig. 1(b) is the case that they are not neighbors. From the figures, it can be noticed that the transfer of recovery information can be performed asynchronously with the normal hand-off procedure, in order to reduce the delay in hand-off. One possible problem of the asynchronous log transfer is that MSS_{old} may not send the correct $Trace_i$, sicne MSS_{old} may not know which one of RM_{old} and RM_{new} would retrieve the recovery information. This problem can easily be fixed if MSS_{old} include both of the identifier of RM_{old} and RM_{new} in $Trace_i$ so that MH_i can contact to both locations in case of any problem situation.

For the simplicity, Fig. 1 describes the recovery information transfer of MSS_{old} only, however, every MSSs in $Trace_i[a].log_{set}$ follows the steps (4)-(6) in Fig. 1(a) and the steps (5)-(7) in Fig. 1(b). Table I summarizes the symbols used for the protocol description.

TABLE I
SYMBOLS FOR PROTOCOL DESCRIPTION

Symbol	Description
MH_i	A mobile host with an identifier *i*.
MSS_p	A mobile support station with an identifier *p*.
RM_m	A region manager with an identifier *m*.
C_i^a	The *a*-th checkpoint of MH_i.
M_i^a	The *a*-th log entry for the message received by MH_i.
$m_i^{rcv\_seq}$	A variable to count the number of messages received by MH_i.
$Trace_i$	This includes the location information for C_i^a and M_i^a of MH_i. $Trace_i[a].cp_{loc}$ has the identifier of MSS carrying C_i^a and $Trace_i[a].log_{set}$ has a set of identifiers of MSSs carrying log entries saved after C_i^a.
$CL_{(i,p)}$	A checkpoint and/or a set of log entries of MH_i, which are managed by MSS_p.
HO_RQST(i)	A message to inform the hand-off of MH_i.
HO_Done(i)	A message to inform the completion of the hand-off process of MH_i.
LT_RQST(i)	A log transfer request message for MH_i.
Save_Done(i)	A message to inform the completion of recovery information transfer for MH_i.

When MH_i fails, it first contacts with its current MSS, say MSS_p, for the recovery. MSS_p then contact with its RM, say RM_p. The recovery information saved in the MSSs visited in the current region can be retrieved from MSS_p and the recovery information saved in the neighbor RMs can be retrieved from RM_p. RM_p may also have the recovery information collected from some distant RMs. As a result, MH_i can correctly recover from the failure.

IV. PERFORMANCE STUDY

A mobile computing system with an *N* x *N* mesh cell configuration [13] has been simulated. A square shaped cell has eight neighbors and the homogeneous size cells are assumed. To simulate the region-based schemes, the system is divided into the homogeneous size regions and one region includes *n* x *n* cells.

One hundred MHs are initially distributed over the cells and

TABLE II
SYMBOLS FOR PERFORMANCE STUDY

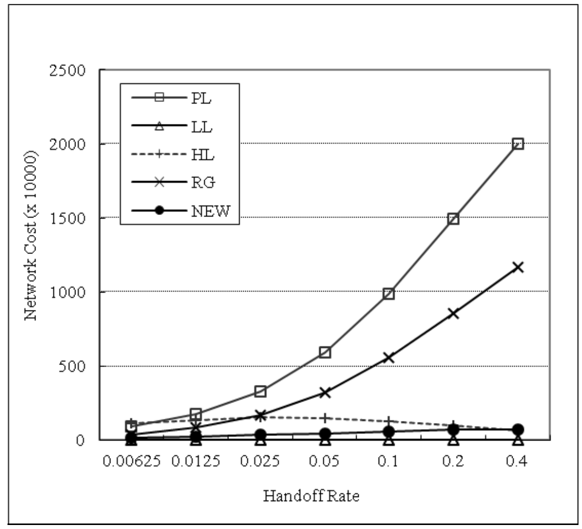
Symbol	Description
L_h	Hand-off rate of a mobile host
L_c	Message sending rate of a mobile host
C_c	Checkpoining interval of a mobile host
L_f	Failure rate of a mobile host
C_m	Average cost of transferring a control message over one hop of the wired network
$a*C_m$	Average cost of transferring an application message over one hop of the wired network
$b*C_m$	Average cost of transferring a checkpoint over one hop of the wired network
r	Ratio of wireless network cost to wired network cost

the MSS initially connected to an MH is assigned as a home of the MH. Each MH takes the next move into one cell randomly selected out of eight neighbors and the time interval between two consecutive hand-offs follows the exponential distribution with a mean $1/L_h$. The message sending rate of a process running on a MH follows a Poisson process with the rate L_c, and the recipient of each message is selected randomly. Each process takes a checkpoint with a fixed checkpointing interval, C_c, and the failure rate of each MH follows a Poisson process with a rate L_f.

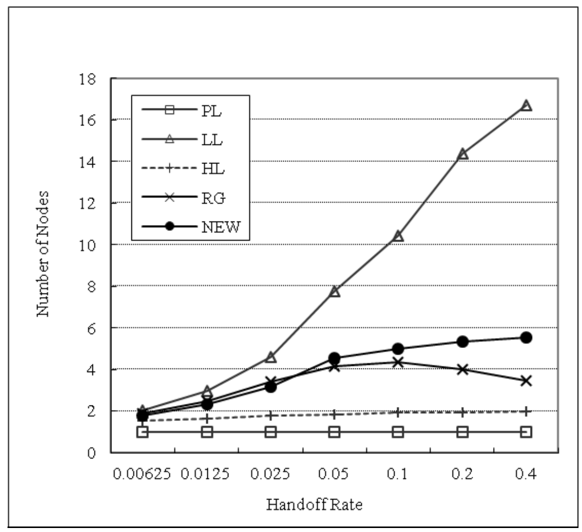
Two performance measures are used in the experiment. One is the number of nodes participated in the failure-recovery, which indicates the cost to retrieve the recovery information from the stable storage. The other is the network cost for the checkpoint transfer and the log transfer during the hand-off and the recovery. The network bandwidth usage is measured as the relative cost: Let C_m be the average cost of transferring a control message over one hop of the wired network. For the cost of transferring an application message and a checkpoint over one hop of the wired network, $a*C_m$ and $b*C_m$ are used, respectively. For the wireless network cost, r is multiplied to the cost. Table II summarizes the symbols used in the experiments and their description.

Five schemes are evaluated; pessimistic scheme (PL), lazy scheme (LL), home based scheme (HL), region based scheme (RG) proposed in [11] and the scheme proposed in the paper (NEW). In the pessimistic scheme, the checkpoint and the log entries are transferred to the new MSS when an MH moves to a new cell. In the lazy scheme, each MSS visited by an MH carries the recovery information created while the MH resides in the corresponding cell. The recovery information in the home based scheme is transferred to the home of an MH during the hand-off of the MH. In the previous region based scheme, the recovery information transferred to the new region manager when an MH moves to a new region. Otherwise, each MSS visited by the MH maintains the recovery information created.

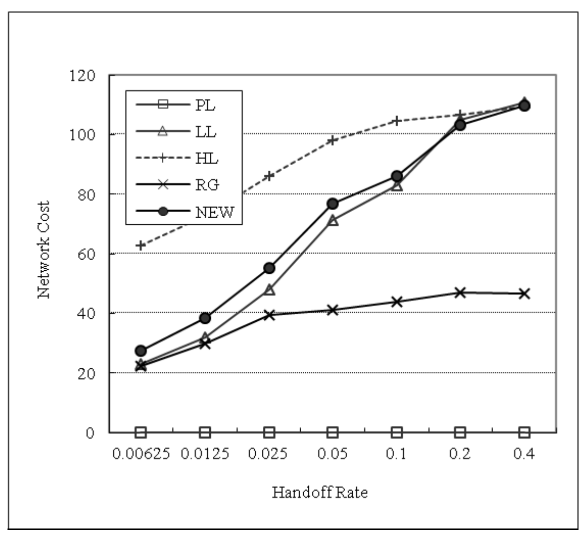
Fig. 2 first shows some simulation results with the 10 X 10 system and 5 X 5 regions: To obtain the performance, the following values are used: $L_c=10^{-1}$, $C_c=200$, $L_f=10^{-4}$, $C_m=1$, $a=1$, $b=10$ and $r=10$.



(a) Network cost during the hand-off operations.



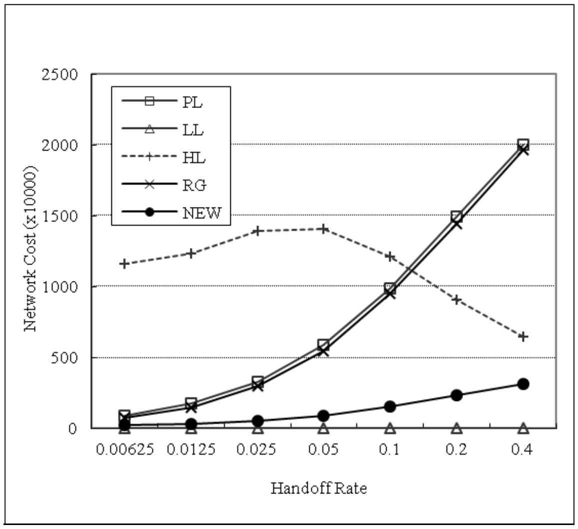
(b) Number of nodes participated for each recovery.



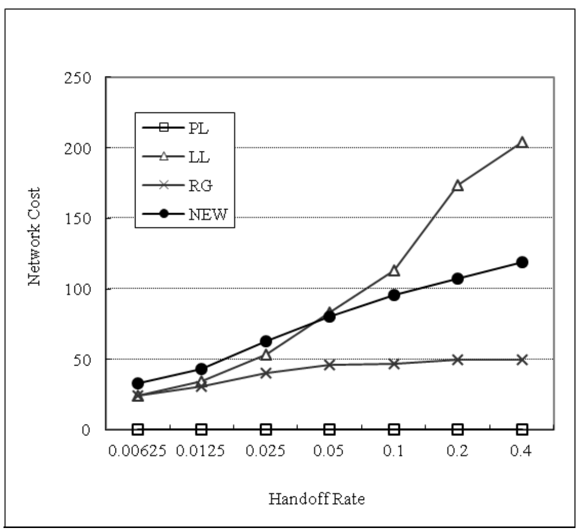
(c) Network cost during each recovery.

Fig. 2. Simulation results with the 10x10 system and 5x5 regions.

Fig. 2(a) compares the network cost of five schemes. The scheme PL requires the highest network cost due to the frequent transfer of the accumulated recovery information. The scheme RG also shows the higher network cost as the handoff rate of an MH becomes higher. Since the frequent handoff increases the possibility of region change, the frequency of recovery information transfer to the new region manager and the network cost becomes increased.



(a) Network cost during the hand-off operations.



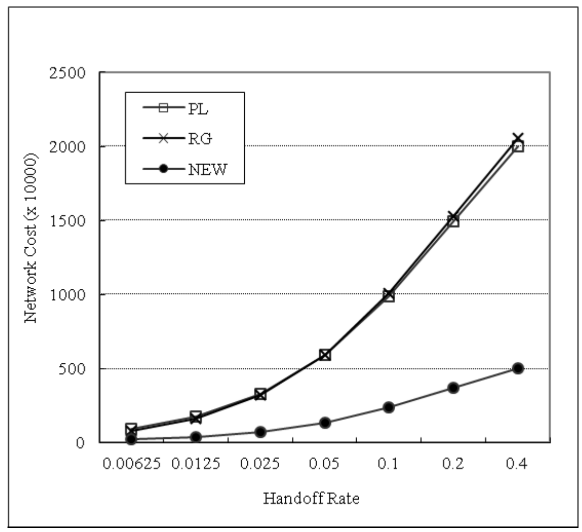
(b) Network cost during each recovery.

Fig. 3. Simulation results with the 100x100 system and 5x5 regions.

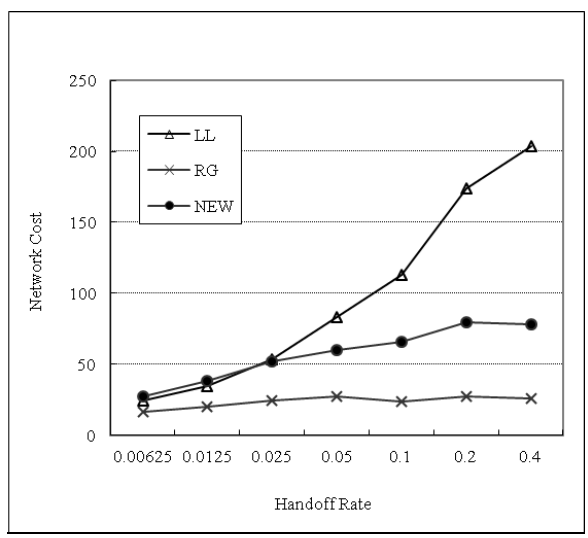
Fig. 2(b) shows the number of nodes carrying the recovery information required for the failure-recovery. When an MH fails, these nodes should retrieve the information from the stable storage, which causes the stable storage access cost. As shown in the figure, the recovery information is dispersed in the largest number of nodes in the scheme LL, since the migration and the collection of recovery information are not performed during the handoff. The scheme RG and the scheme NEW require the less number of participant nodes since the collection of recovery information may be done when the MH changes the region.

Fig. 2(c) compares the network cost required for each failure recovery. The scheme HL shows the worst performance since the distance from the current MSS of an MH and its home MSS may not be near. The scheme NEW requires the cost higher than that of the scheme RG since the recovery information in the scheme RG maintains by the current region manager and the MSSs in the current region, while the information in the scheme NEW can be maintained by the neighbor region managers. Instead, in the scheme NEW, the information transfer cost during the handoff can be much lower as shown in Fig. 2(a).

To examine the performance of the MH moving in a wide range, the simulation with a larger system size has been done. Fig. 3 shows the results for the system having 100 x 100 cells. From Fig. 3(a), it can be noticed that the scheme HL and the scheme RG shows the performance which is very different from that in Fig. 2(a). In the larger system, an MH moves very far from its home and hence the cost transferring the recovery information from current MSS to the home can be very higher.



(a) Network cost during the hand-off operations.

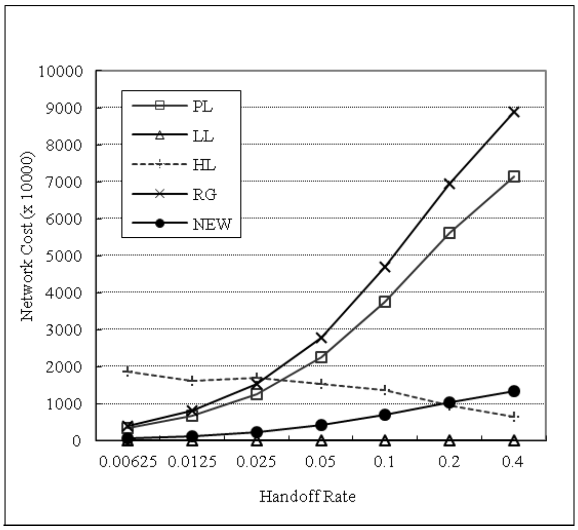


(b) Network cost during each recovery.

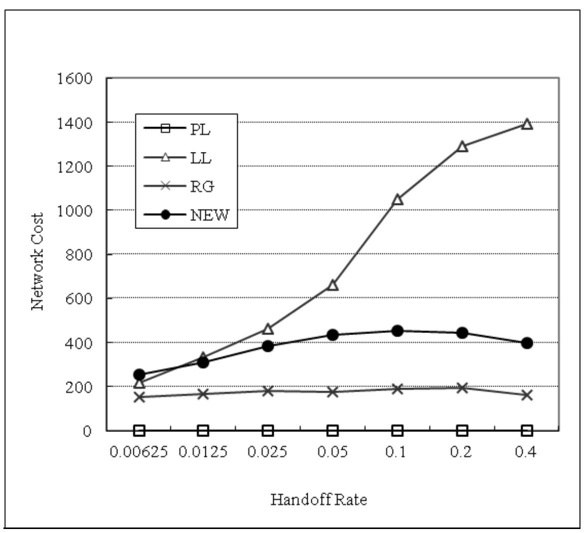
Fig. 4. Simulation results with the 100x100 system and 3x3 regions.

For the scheme RG, the MH has more chance to change its region in the larger system and hence the network cost becomes higher.

In Fig. 3(b), the recovery cost of the scheme LL becomes much higher than that of the scheme NEW, since in the larger system, the movement of an MH can include more number of cells. The scheme NEW proposed in the paper performs very well in the large system, since the maintenance of the recovery information is collaborated by a current region manager and its neighbor region managers. The cost of the scheme HL is not in Fig. 3(b) since the recovery cost of the scheme is too high compared to the other schemes.



(a) Network cost during the handoff operations.



(b) Network cost during each recovery.

Fig. 5. Simulation results with the 100x100 system, 5x5 regions and C_c=1000.

Fig. 4 shows the effect of the region size. For these results, the system with 3x3 regions has been simulated. As shown in Fig. 4(a), the cost during the handoff operations has slightly become larger for both of the scheme RG and the scheme NEW. The main reason is that the small region incurs more number of region changes. However, the recovery cost of both schemes can be much lower, since in the small region, the distance from the current MSS and the region manager or the neighbor region mangers can be nearer.

For the experiments in Fig. 5, the checkpointing interval, C_c, has been changed from 200 to 1000. For all the recovery schemes, the log entries collected are deleted when an MH takes a new checkpoint, since the MH need not be rolled back to the point before the latest checkpointing. Hence, the longer checkpointing interval means the transfer of more log entries during the handoff operations and the recovery.

Fig. 5(a) and 5(b) show the effect of more log entries. In Fig. 5(a), the network cost of the scheme PL and the scheme RG becomes much higher and in Fig. 5(b), the recovery cost of the scheme LL becomes much higher. The scheme HL does not show any performance differences, since in this scheme, the log entries are moved to the home for each handoff and they are not accumulated unlike the scheme PL and the scheme RG. However, the network cost of the scheme HL is not included in Fig. 5(b), since the cost is too high compared to the other schemes.

The scheme NEW proposed in the paper shows the performance better than those of the scheme PL, the scheme HL and the scheme RG, during the handoff operations. The main reason is that the scheme NEW reduces the unnecessary log transfer within neighbor regions. Also, the scheme NEW shows the performance better than those of the scheme LL and the scheme HL, during the recovery. The reason is that the log entries for the recovery are maintained by the current region manager and its neighbor region managers within a certain range.

V. Conclusion

This paper presents a region-based distributed storage management scheme for mobile environments. Unlike the previous schemes, the patterns of MHs' movement are categorized into three steps; moving within a region, moving around neighbor regions, moving across regions. When an MH moves within a region, the recovery information is not migrated. When the MH moves around the neighbor regions, each RM collects the information for fast recovery. Only when the MH moves far from the previous region, the recovery information is migrated to the current region manager.

The simulation results show that the proposed scheme shows the performance better that those of the pessimistic scheme, the home based scheme and the other region based scheme, since in the proposed scheme, the recovery information is not migrated while an MH moves around neighbor regions. Also, the proposed scheme shows the performance better than those of the lazy scheme and the home based scheme, since the distance from the current MSS and the region manager carrying the recovery information is restricted within the neighbor regions.

REFERENCES

[1] B.R. Badrinath, A. Acharya and T. Imielinski, "Structuring distributed algorithms for mobile hosts," In *Proc. of the 14th Int'l Conf. on Distributed Computing Systems*, pp. 21-28, Jun. 1994.

[2] G. Cao and M. Singhal, "Low-cost checkpointing with mutable checkpoints in mobile computing systems," In *Proc. of the 18th Int'l Conf. on Distributed Computing Systems*, pp. 464-471, May 1998.

[3] R. Prakash and M. Singhal, "Low-cost checkpointing and failure recovery in mobile computing," *IEEE Trans. on Parallel and Distributed Computing Systems*, Vol. 7, pp. 1035--1048, Feb. 1996.

[4] A. Acharya and B.R. Badrinath, "Checkpointing distributed applications on mobile computers," In *Proc. of the 3rd Int'l Conf. on Parallel and Distributed Information Systems*, pp. 73--80, Oct. 1994.

[5] D. Manivannan and M. Singhal, "Failure recovery based on quasi-synchronous checkpointing in mobile computing systems," *OSU-CISRC-7\/96-TR36, Dept. of Computer and Information Science, The Ohio State University*, 1996.

[6] N. Neves and W.K. Fuchs, "Adaptive recovery for mobile environments," *Communications of the ACM*, Vol. 40, No. 1, pp. 68--74, Jan. 1997.

[7] D.K. Pradhan, P. Krishna, and N.H. Vaiday, "Recoverable mobile environment: Design and trade-off analysis," In *Proc. of the 26th Int'l Symp. on Fault Tolerant Computing Systems*, pp. 16--25, Jun. 1996.

[8] B. Yao, K. Ssu and W.K. Fuchs, "Message logging in mobile computing," In *Proc. of the 29th Symp. on Fault Tolerant Computing Systems*, pp. 294--301, Jun. 1999.

[9] T. Park and H.Y. Yeom, "An asynchronous recovery scheme based on optimistic message logging for mobile computing systems," In *Proc. of the 20th Int'l Conf. on Distributed Computing Systems*, pp. 436-443, Apr. 2000.

[10] L. Alvisi and K. Marzullo, "Message logging: Pessimistic, optimistic and causal," In *Proc, of the 15th Int'l Conf. on Distributed Computing Systems*, pp. 229--236, May 1995.

[11] T. Park, N. Woo and H.Y. Yeom, "Efficient Recovery Information Management Schemes for the Fault Tolerant Mobile Computing Systems," In *Proc. of the 20th IEEE Symp. on Reliable Distributed Systems*, Oct. 2001.

[12] I.F. Ayildiz and J.S.M. Ho, "On location management for personal communications networks," *IEEE Communications Magazine*, pp. 138--145, Sep. 1996.

[13] J. Li, H. Kameda and K. Li, "Optimal dynamic location update for PCS networks," In *Proc. of the 19th Int'l Conf. on Distributed Computing Systems*, May 1999.