社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－MBL－36（54）
2006－UBI－10（54）
2006／2／17

Stable Storage for Wireless Multihop Access Networks

Tatsuya Hirakawa and Hiroaki Higaki
Department of Computers and Systems Engineering
Tokyo Denki University
E-mail: {tatuya,hig}@higlab.net

For implementation of a checkpoint recovery method in a mobile wireless network, a stable storage is required for storing state information. However, failure-independent storage devices are not realized in a mobile computer. This paper proposes a distributed stable storage realized by cooperation of multiple neighbor mobile computers. Our proposed protocol works with an existing 3-phase checkpoint protocol without additional control message transmissions.

無線マルチホップアクセスネットワークにおける安定記憶実現手法

東京電機大学 理工学部 情報システム工学科
平川達也　桧垣 博章
E-mail: {tatuya,hig}@higlab.net

モバイルネットワーク環境を対象としてチェックポイントリカバリを実行するためには、各コンピュータの状態情報を格納するための安定記憶を実現しなければならない。しかし、モバイルコンピュータは、故障独立した安定記憶を自身に実現してはいない。本論文では、モバイルコンピュータの状態情報を格納するための安定記憶を実現するために、複数の近隣モバイルコンピュータの揮発性記憶か安定記憶を実現している無線基地局かのいずれかに状態情報を格納する分散型安定記憶の実現手法を提案する。提案手法では、従来の 3 フェーズ型チェックポイントプロトコルに対して追加の制御メッセージを必要とせず、すべてのモバイルコンピュータに安定記憶を実現することができる。

1 Introduction

Recently, wireless LANs composed of mobile computers such as handheld PCs, palmtop PCs and personal digital assistants (PDAs) in which a wireless communication protocol such as IEEE802.11 [2] and Bluetooth [1] is implemented are researched and developed. Most of currently available wireless LANs are infrastructured networks. Here, a mobile computer exchanges wireless signals carrying application data only with a base station. However, in an infrastructured network, a mobile computer communicates with another computer only while it is within a wireless signal transmission range of a base station. Hence, it is difficult to achieve high connectivity. In order to achieve higher connectivity in a mobile wireless network, wireless multi-hop transmission is introduced [13]. There are two kinds of wireless networks with wireless multihop transmission; one is a wireless multi-hop access network composed of a wired network of fixed computers, mobile networks of mobile computers with wireless communication devices and base stations which is gateways between the wired network and the wireless ones and the other is an ad hoc network composed of only mobile computers.

In a wireless multi-hop access network, for supporting mission-critical network applications, checkpoint recovery is one of the most important methods for achieving a fault-tolerant and failure-resilient environment. In most of the proposed checkpoint recovery protocols designed for wired networks [5], a stable storage which is k-resilient where k is the possible maximum simultaneous failures is assumed to be implemented in each computer. However, in a mobile computer, it is difficult to implement such a stable storage.

In checkpoint protocols designed for infrastructured networks, state information of all mobile computers at their local checkpoints is stored into a stable storage in a base station [3, 6, 9–11] since all the mobile computers are within a wireless signal transmission range of one of the base stations. That is, only 1-hop wireless transmission is required for storing state information of each mobile computer. However, in a wireless multi-hop access network, hop counts to the nearest base station are not the same and based on a layout of mobile computers and the base station. Hence, a method storing state information of mobile computers always into a stable storage in a base station may require higher communication overhead. Hence, this paper discusses a method to store state information of a mobile computer into storage devices in its neighbor mobile computers for reduction of communication overhead in checkpointing and recovery and of storage overhead in a base station. That is, a k-resilient storage for storing state information of a mobile computer is realized distributedly on its multiple neighbor mobile computers. We design a checkpoint protocol supporting the distributed stable storage based on a 3-phase checkpoint protocol and evaluate its performance in simulation experiments.

2 Related Works

In a fixed computer F_i connected to a wired network, a k-resilient stable storage is available and state information S_i at a local checkpoint c_i is recovered even in a case of k-simultaneous failures. This is because F_i has $k+1$ failure-independent storage devices and S_i is stored into all of them. However, it is difficult for a mobile computer M_i to support a k-resilient stable storage in M_i since storage devices in M_i are usually failure-dependent. For example, these storage devices may be broken simultaneously due to falling of M_i. In an infrastructured network where a mobile computer exchanges wireless signals only with a base station, most checkpoint protocols are designed to store state information of mobile computers into a stable storage in a base station or in another fixed computer through a base station [3, 6, 9–11]. However, in a wireless multi-hop access network and an ad hoc network where a mobile computer exchanges wireless signals with any computers, i.e. both base stations and mobile computers, within its wireless signal transmission range and messages are transmitted by wireless multi-hop transmission to a base station or a destination mobile computer, a mobile computer may be multiple hops away from a base station as shown in Figure1. Hence, higher communication overhead is required to transmit state information of a mobile computer to a base station. The higher communication overhead is also required in recovery after a failure since the state information is transmitted back to the mobile computer by wireless multi-hop transmission. In addition, since state information of multiple computers is stored into a k-resilient stable storage in a base station or another fixed mobile computer, high-spec configuration is required in storage devices for store and recovery of the state information.

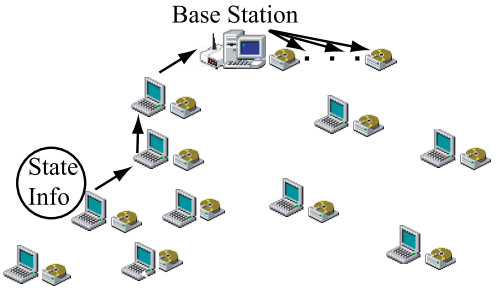

Figure 1: Centralized Stable Storage in Base Station.

In order to reduce the communication overhead to transmit state information of a mobile computer to a base station and storage overhead in a base station, a protocol achieving a distributed k-resilient stable storage in mobile computers has been proposed in [16]. Here, state information of each mobile computer is stored into storage devices in $k+1$ different mobile computers as shown in Figure2 or into a k-resilient stable storage in a base station only for mobile computers for which it is impossible to realize a distributed k-resilient stable storage. The protocol is base on a 3-phase checkpoint protocol such as in [4] and [8]. Here, there is a checkpoint coordinator computer Co determined dynamically or stably. In the 1st phase, a checkpoint request message $CReq$ is distributed from Co to all computers and each computer takes its local checkpoint by taking a snapshot of its current state information. In the 2nd phase, a checkpoint reply message $CRep$ is gathered from all the computers to Co for confirmation. In the 3rd phase, a checkpoint finish message $CFin$ is distributed from Co to all computers for commitment and achieving consistency [4]. In the proposed protocol in [16], since all information including state information of mobile computers is piggybacked to the message in a 3-phase checkpoint protocol, i.e. $CReq$, $CRep$ and $CFin$, no additional messages are required to realize stable storage distributed on multiple mobile computers.

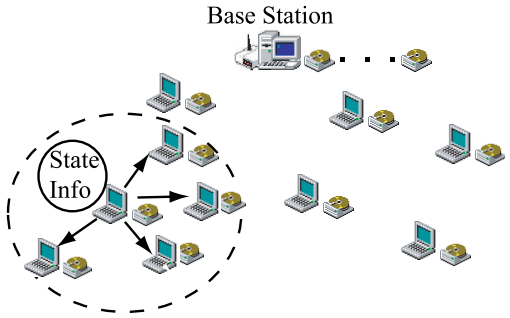

Figure 2: Distributed Stable Storage in Neighbor Mobile Computers.

The overview of the protocol in [16] is as follows. In the protocol, distributions of $CReq$ and $CFin$ messages in a mobile network with wireless multi-hop transmission are realized by flooding of them. Initiator of the flooding is served by base stations. By the flooding of a $CReq$ message, all mobile computers are informed to take local checkpoints. In addition, spanning trees on the mobile network are dynamically configured [14]. Each spanning tree includes a base station as a root. Gathering of $CRep$ messages in a mobile network is realized by transmission along the links of the spanning tree. Each mobile computer M_i takes its local checkpoint c_i when it receives the first $CReq$ message from a neighbor computer which is a base station or a mobile computer. M_i takes a snapshot of its current state information S_i and stores S_i into its storage device. Since M_i broadcasts a $CReq$ message after receipt of the first $CReq$ message, by designing a checkpoint protocol as that M_i broadcasts a $CReq$ message after taking c_i, S_i is piggybacked to the $CReq$ message. Hence, S_i is transmitted to all the mobile computers in a wireless signal transmission range of M_i, i.e. to all 1-hop neighbor mobile computers of M_i. Now, S_i is stored into storage in $|N1_i|+1$ failure-independent mobile computers where $N1_i$ is a set of 1-hop neighbor mobile computers of M_i. It depends on a mobile network topology and timing of transmission of a $CReq$ message whether $|N1_i|+1 > k$ is satisfied. If

$|N1_i| + 1 \leq k$, S_i should be stored into storage devices in additional mobile computers. Since the spanning tree determined by transmission of *CReq* is dynamically configured, M_i does not get a set of its descendant mobile computers before M_i broadcasts a *CReq* message. In addition, since communication among the descendant mobile computers of M_i is restricted to one along the links of the spanning tree, M_i cannot control a number of descendant mobile computers storing S_i into storage devices. Hence, S_i is stored into storage devices in ancestor mobile computers of M_i and their neighbor ones. If $|N1_i| \leq k$, S_i is piggybacked to a parent mobile computer M_j of M_i. Then, S_i is piggybacked to a *CRep* message sent to a grandparent mobile computer M_k of M_i. Since the *CRep* message is broadcasted in a wireless signal transmission range of M_j, all 1-hop neighbor mobile computers of M_j overhears it. Hence, by storing S_i into their storage devices, S_i is stored into storage devices in $|N1_i \bigcup N1_j|+1$ mobile computers. Until $|\bigcup_i N1_i| + 1 > k$is satisfied, the same procedure is repeated. That is, S_i is stored into storage devices in 1-hop neighbor mobile computers of an additional ancestor mobile computer. Only if a total number of ancestor mobile computers of M_i and their 1-hop neighbor mobile computers is less than k, S_i is stored into a stable storage in a base station. In this case, S_i is surely transmitted to the base station without additional messages since S_i is piggybacked to the *CRep* message forwarded by ancestor mobile computers of M_i.

3 Novel Checkpoint Protocol

As discussed in the previous section, in the proposal protocol in [15], state information S_i of a mobile computer M_i at its local checkpoint c_i is stored into storage devices in M_i and its ancestor mobile computers and their neighbor ones and into a stable storage in a base station only if the number of these mobile computers is less than $k + 1$. This means that much state information should be transmitted to a base station and stored into its stable storage in the following cases:

- k is large due to high failure ratio.
- A number of neighbor mobile computers is small due to low density of mobile computers (i.e. sparse distribution of mobile computers).
- A number of ancestor mobile computers is small due to location of a mobile computer.

In order to reduce the communication in checkpoint and recovery and storage overhead in a base station, this paper proposes an extended protocol in which state information S_i of M_i is stored into 2-hop neighbor mobile computers without any additional messages. As shown in Figure3, since a set $N2_i$ of 2-hop neighbor mobile computers of M_i includes a set $N1_i$ of 1-hop neighbor ones, less ancestor mobile computers and their 1-hop neighbor ones are required to store S_i and a number of state information required to be stored into a stable storage in a base station is reduced. Each mobile computer and base station determines its 1-hop neighbor mobile computers by broadcasting a control

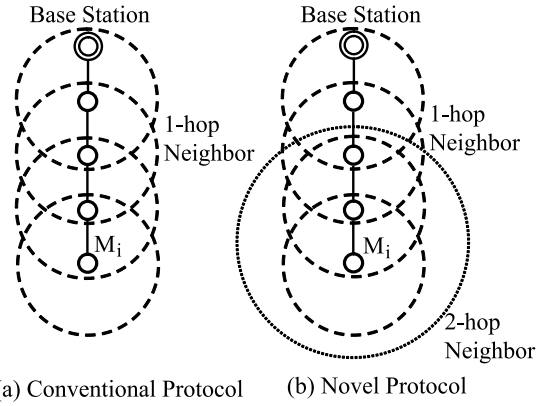

Figure 3: Candidates Storing $\mathbf{S_i}$.

message containing an *ID* of a sender computer repeatedly with a certain interval to all computers within its wireless signal transmission range. The protocol in [15] is base on this mechanism and it is assumed that each mobile computer M_i has a set $N1_i$ of its current 1-hop neighbor mobile computers. By broadcasting a control message containing not only an *ID* of sender computer but also a set of *IDs* of 1-hop neighbor mobile computers repeatedly with a certain interval to all computers within a wireless signal transmission range of a sender mobile computer, each mobile computer M_i achieves a set $N2_i$ of 2-hop neighbor mobile computers. For example, in an ad hoc routing protocol OLSR [7], each mobile computer M_i broadcasts a HELL message containing $N1_i$ repeatedly with a certain interval, M_i achieves $N2_i$. On the other hand, in another ad hoc routing protocol DSDV [12], which is based on a distance vector routing protocol such as RIP and BGP, each mobile computer maintains a routing table which is kept up-to-date by exchanging control messages containing a routing table of a sender mobile computer repeatedly. Hence, a mobile computer M_i achieves $N1_i$ and $N2_i$ from a routing table. That is, in a mobile network applying such an ad hoc routing protocol, M_i achieves $N1_i$ and $N2_i$ without any additional overhead. In the protocol in [15], a *CReq* message is distributed to all mobile computers by flooding. Here, each mobile computer broadcasts a *CReq* message to all its 1-hop neighbor mobile computers exactly once and receives a *CReq* message from each 1-hop neighbor mobile computer exactly once. Hence, by piggybacking state information of a sender mobile computer to surly transmitted to all its 1-hop neighbor mobile computers and each mobile computer surly achieves state information of its 1-hop neighbor mobile computers. In addition, each mobile computer transmits a *CRep* message to its parent computer on a spanning tree exactly once after it receives *CReq* messages from all its 1-hop neighbor mobile computers and *CRep* messages from all its child mobile computers. Hence, Before a mobile computer transmits a *CRep* message to its parent mobile computer on a spanning tree configured by flooding of a *CReq* message, it achieves state information of all its 1-hop neighbor mobile computers. Thus, by overhear-

ing a *CRep* message, each mobile computer achieves state information of all 2-hop neighbor mobile computers. Therefore, state information of a mobile computer is transmitted to all 2-hop neighbor mobile computers. In the following protocol, each *CRep* message is broadcasted to all 1-hop neighbor mobile computers of a sender one.

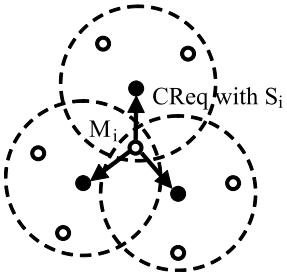

Figure 4: Transmission of $\mathbf{S_i}$ to 1-hop Neighborsby piggyback to **CReq**.

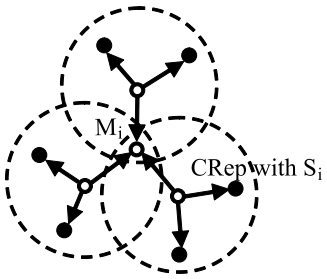

Figure 5: Transmission of $\mathbf{S_i}$ to 2-hop Neighbors by piggyback to **CRep**.

[Checkpoint Protocol]

(Flooding of *CReq*)

1) A fixed computer F_0 connected to a wired network serves a coordinator. F_0 takes its local checkpoint c_0 by storing its current state information into a stable storage composed of $k+1$ failure-independent storage devices. Then, F_0 sets a timer T_0 and boradcasts *CReq* messages to all fixed computers connected to the wired network.
2) If T_0 expires before F_0 receives *CRep* messages from all the fixed computers, F_0 resets T_0 and broadcasts the *CReq* messages again.
3) On receipt of a *CReq* message, a fixed computer F_i connected to the wired network takes its local checkpoint c_i by storing its current state information into a stable storage conposed of $k+1$ failure-independent storage devices. Then, F_i sends back a *CRep* message to F_0.
4) On receipt of a *CReq* message, a base station B_i connected to the wired network takes its local checkpoint c_i by storing its current state information into a stable storage conposed of $k+1$ failure-independent storage devices. Then, B_i sets a timer T_i and broadcasts the *CReq* message to all mobile computers within its wirress signal transmission range, i.e. to all neighbor mobile computers.
5) If T_i expires before B_i receives *CRep* messages from all the mobile computers whitin its wireless signal transmission range, B_i resets T_i and broadcasts the *CReq* message again.
6) On receipt of a *CReq* message sent from a neighbor computer N_i which is a base station or a mobile computer, a mobile computer M_j suspends transmissions of application messages during the following procedure.
   6–1) If M_j has not yet received the same *CReq* message from N_i, M_j stores state information $S_i$of N_i into its storage device if it is piggybacked to the received *CReq* message.
   6–2) If M_j has not yet received the same *CReq* message from any neighbor computers, M_j takes its local checkpoint c_j by storing its current state information S_j into its storage device. Let $M_j^{down} := \phi$. Then, M_j sets a timer T_j and broadcasts a *CReq* message to which S_j and an *ID* of N_i are piggybacked to all computers within its wireless signal transmission range.
   6–3) If M_j has already received the same *CReq* message from a neighbor mobile computer and an *ID* of M_j is piggybacked to the received *CReq*, $M_j^{down} := M_j^{down} \cup \{N_i\}$.
   6–4) If T_j expires before M_j receives *CRep* messages from all the mobile computers within its wireless signal transmission range, M_j resets T_j and broadcasts the *CReq* message again.

(Gathering of *CRep*)

1) A mobile computer M_i broadcasts a *CRep* message to all computers within its wireless signal transmission range if M_i has already broadcasted the *CReq* message and received the *CReq* messages from all the neighbor computers and $M_i^{down} = \phi$. The following information is piggybacked to the *CRep* message:
   - State information piggybacked to the received *CReq* messages.
   - A tuple $\langle N2_i, S_i \rangle$ if $|N2_i| < k$ where $N2_i$ is a set of 2-hop neighbor mobile computers of M_i.
2) On receipt of a *CRep* message, a mobile computer M_i stores state information piggybacked to the *CRep* message (including state information S_i piggybacked as a tuple $\langle stored_j, S_j \rangle$) into its storage device.
3) A mobile computer M_i broadcasts a *CRep* message to all computers within its wireless signal transmission range if M_i has already broadcasted the *CReq* message, received the *CReq* messages from all the neighbor computers and received *CRep* messages from all mobile computers in M_i^{down}. The following information is piggy-

backed to the *CRep* message:

- State information piggybacked to the received *CReq* messages.
- A tuple $\langle N2_i, S_i \rangle$ if $|N2_i| < k$ where $N2_i$ is a set of 2-hop neighbor mobile computers of M_i.
- A tuple $\langle stored_j \cup N1_j, S_j \rangle$ for each tuple $\langle stored_j, S_j \rangle$ piggybacked to the received *CRep* message if $|stored_j| < k$ where $N1_i$ is a set of neighbor mobile computers of M_i.

4) On receipt of a *CRep* message from a neighbor mobile computer, a base station B_i stores state information S_j into its $k - |stored_j|$ failure-independent storage devices for each tuple $\langle stored_j, S_j \rangle$ piggybacked to the received *CRep* message if $|stored_j| < k$.
5) B_i sends the *CRep* message to F_0 when B_i receives *CRep* messages from all the mobile computers within its wireless signal transmission range.

(Flooding of *CFin*)

1) On receipt of all the *CRep* messages from fixed computers connected to the wired network, F_0 broadcasts *CFin* messages to all the fixed computers.
2) On receipt of the *CFin* message, a fixed computer F_i terminates the checkpoint protocol.
3) On receipt of the *CFin* message, a base station B_i sets a timer T_i and broadcasts the *CFin* message to all mobile computers within its wireless signal transmission range.
4) If B_i receives the same *CFin* messages from all mobile computers within its wireless signal transmission range before T_i expires, B_i terminates the checkpoint protocol. Otherwise, i.e. if T_i expires, B_i resets T_i and broadcasts the *CFin* message again.
5) On receipt of *CFin* message sent from a neighbor computer N_i which is a base station or a mobile computer, a mobile computer M_j sets a timer T_j and broadcasts the *CFin* message to all the mobile computers within its wireless signal transmission range if M_j has not yet received a *CFin* message from any neighbor computers.
6) If T_j expires before M_j receives *CFin* messages from all the neighbor mobile computers, M_j resets T_j and broadcasts the *CFin* message again. □

4 Evaluation

In this section, performance of the proposed protocol is compared to one of the protocol in [16] and of a naive protocol in which all state information is transmitted to a base station. Here, amount of state information stored into a stable storage in a base station and average and maximum hop counts between a mobile computer and computers storing its state information are evaluated by simulation experiments. These are for estimation of storage and communication overhead for checkpointing and recovery. Simulation assumptions are as follows. A wireless signal transmission range of a mobile computer is a circle whose center is the mobile computer and radius is $100m$. 30–100 mobile computers and a base station are randomly located according to unique distribution in a $500m \times 500m$ field. Only layouts in which all mobile computers are reachable to a base station wireless in multi-hop transmission.

State information S_i of a mobile computer M_i is stored into $max(k - X_i, 0)$ failure-independent storage devices in a base station if S_i is stored into storage devices in X_i mobile computers. In a naive protocol, X_i is always 0. In a conventional protocol, since M_i and its ancestor mobile computers and their 1-hop neighbor ones are candidates to store S_i, $X_i = |\bigcup_j (N1_j \cup \{M_j\})|$ where M_j is M_i or its ancestor mobile computer and $N1_j$ is a set of 1-hop neighbor mobile computers of M_j. Finally, in the proposed protocol, since M_i, 2-hop neighbor mobile computers of M_i, ancestor mobile computers of M_i and their 1-hop neighbor mobile computers are candidates to store S_i, $X_i = |(N2_i \cup \{M_i\}) \cup (\bigcup_j (N1_j \cup \{M_j\}))|$ where $N2_i$ is a set of 2-hop neighbor mobile computers of M_i. Totally, amount of state information stored into a base station is $\sum_i max(k - X_i, 0)$ on an assumpiton that all state information require the same amount of storage. Figures 6 and 7 show ratio of amount of state information stored into a stable storage in a base station compared to the naive protocol. In Figure 6, the possible maximum number of simultaneous failures k is fixed to 10 and number of mobile computers is changed. Compared to the naive protocol, both the conventional and the proposed protocol require only 11.0–17.1% amount of storage in a base station. In addition, compared to the conventional protocol, the proposed protocol requires 19.3%, 1.7% and 8.5% less storage devices with 30 mobile computers, 100 mobile computers and averagely, respectively. In Figure 7, a mobile network consists of 60 mobile computers and k is changed. Here, 0.5%, 28.0% and 12.6% storage consumption is reduced

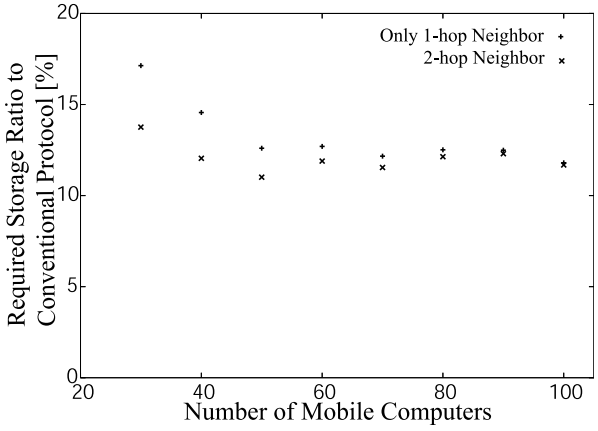

Figure 6: State Information in Base Station for 10-resiliency.

in cases of $k = 5$ and $k = 20$ and averagely, respectively. As discussed in Section 3, the proposed protocol improves the performance better in cases of lower density of mobile computers and higher failure ratio.

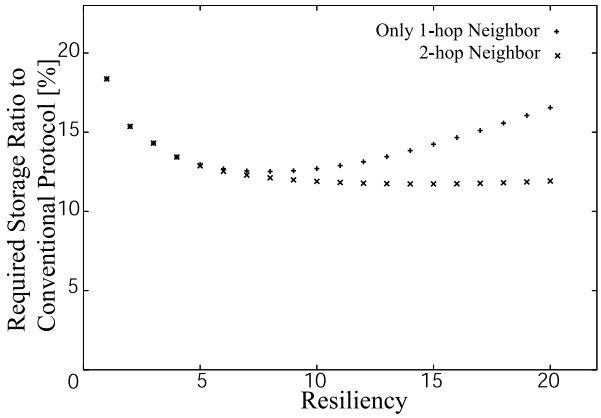

Figure 7: State Information in Base Station with 60 mobile computers.

5 Conclusion

This paper has proposed a method to realize k-resilient distributed stable storage in a checkpoint protocol for a wireless multi-hop access network. By using the method, state information of a mobile computer is stored into storage devices in neighbor mobile computers and less state information is stored into a stable storage in a base station. Hence, communication and storage overhead are reduced. The method is able to be applied to a 3-phase checkpoint protocol without any additional messages. For further reduction of the overhead, not only 1-hop neighbor but also 2-hop neighbor mobile computers are involved into achieving the distributed stable storages. We design a checkpoint protocol by using the proposed method. In simulation evaluation, only 10–20% storage overhead of a naive method storing all state information into a stable storage in a base station is required. Especially in case of sparser mobile computer distribution and higher failure ratio, our extension involving 2-hop neighbor mobile computers works efficiently for reduction of the overhead.

References

[1] "The Official Bluetooth Wireless Info Site," http://www.bluetooth.com .

[2] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) SpecificatiStandard IEEE 802.11 (1999).

[3] Acharya, A. and Badrinath, B.R., "Checkpointing Distributed Applications on Mobile Computers," Proc. of the 3rd International Conference on Parallel and Distributed Information Systems, pp. 73–80 (1994).

[4] Chandy, K.M. and Lamport, L., "Distributed Snapshots: Determining Global States of Distributed Systems," ACM Trans. on Computer Systems, Vol. 3, No. 1, pp. 63–75 (1985).

[5] Elnozahy, E.N., Alvisi, L., Wang, Y.M. and Johnson, D.B., "A Survey of Rollback-Recovery Protocols in Message-Passing Systems," ACM Computing Surveys, Vol. 34, No. 3, pp. 375–408 (2002).

[6] Higaki, H. and Takizawa, M., "Checkpoint-Recovery Protocol for Reliable Mobile Systems," Proc. of the 17th International Symposium on Reliable Distributed Systems, pp. 93–99 (1998).

[7] Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A. and Viennot, L., "Optimized Link State Routing Protocol for Ad Hoc Networks," IETF RFC3626 (2001).

[8] Koo, R. and Toueg, S., "Checkpointing and Rollback-Recovery for Distributed Systems," IEEE Trans. on Software Engineering, Vol. SE-13, No. 1, pp. 23–31 (1987).

[9] Miyazaki, M., Morita, Y. and Higaki, H., "Hybrid Checkpoint Protocol for Mobile Networks with Unreliable Wireless Communication Channels," Proc. of the 2nd Asian International Mobile Computing Conference , pp. 164–171 (2002).

[10] Morita, Y. and Higaki, H., "Checkpoint-Recovery for Mobile Computing Systems," Proc. of the 21st Internation Conference Distributed Computing Systems Workshops, pp. 479–484 (2001).

[11] Neves, N. and Fuchs, W.K., "Adaptive Recovery for Mobile Environments," Communications of the ACM, Vol. 40, No. 1, pp. 69–74 (1997).

[12] Perkins, C.E. and Bhagwat, P., "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computerss," ACM SIGCOMM' 94, pp. 234–244 (1994).

[13] Ueno, S., Tanaka, T., Usui, K., Yasui, Y., Shindo, Y. and Maruyama, H., "Development of Service-area Expomsion Technologies for Wireless IP Access System," NTT Technical Review, Vol. 3, No. 8, pp. 52–56 (2005).

[14] Umeshima, S. and Higaki, H., "Multiple–Route Adhoc Ondemand (MRAODV) Routing Protocol," Proceedings of the IASTED International Conference on Wireless and Optical Communications (WOC), pp. 610–615 (2003).

[15] Ono, M. and Higaki, H., "Checkpoint Protocol for Mobile Ad hoc Networks," IPSJ Technical Report, Vol. 2005, No. 58, pp. 13–18 (2005).

[16] Hirakawa, T., Ono, M. and Higaki, H., "Checkpoint Protocol for k-Resilient Wireless Multi-hop Networks," IPSJ Technical Report, Vol. 2004, No. 61, pp. 13–18 (2004).