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

2005－MBL－32（16）
2005－UBI－　7（16）
2005／3／17

Consistent Checkpoint Protocol for Mobile Ad hoc Networks

Masakazu Ono and Hiroaki Higaki
Department of Computers and Systems Engineering
Tokyo Denki University
E-mail: {masa,hig}@higlab.k.dendai.ac.jp

For achieving mission–critical network applications, checkpoint recovery protocols have been researched and developed. In conventional protocols for wired networks, stable storages to store state information are assumed and enough bandwidth is assigned to synchronize a sender and a receiver computers of a message in order to avoid that the message becomes inconsistent, i.e. neither orphan nor lost. In this paper, we propose a novel checkpoint protocol in ad hoc networks without stable storage and enough communication bandwidth. Here, a checkpoint request message is delivered by flooding. State information of a mobile computer is carried by this message and stored into neighbor mobile computers. A candidate of a lost message is detected and stored by intermediate mobile computer on its transmission route. Here, communication overhead for taking global checkpoint is reduced.

アドホックネットワークのためのチェックポイントリカバリプロトコル

東京電機大学 理工学部 情報システム工学科
小野真和　桧垣 博章
E-mail: {masa,hig}@higlab.k.dendai.ac.jp

ネットワーク環境においてミッションクリティカルアプリケーションを実現する手法として、チェックポイントリカバリプロトコルがある。従来の有線ネットワークを対象としたプロトコルでは、状態情報を格納するための安定記憶が存在することと、メッセージの送信元コンピュータと送信先コンピュータの同期によって一貫性のないメッセージ (紛失メッセージと孤児メッセージ) が検出、回避できる程度に十分な通信帯域幅が存在することが前提となっている。本論文では、これらの前提が成立しないアドホックネットワークにおけるチェックポイントプロトコルを提案する。状態情報は複数の隣接移動コンピュータに記憶する。このとき、紛失メッセージとなる可能性のあるメッセージを中継移動コンピュータの状態情報の一部として記憶することにより、状態情報とメッセージログを同一の移動コンピュータに同時に保存することができる。これによって、チェックポイントプロトコルの開始から終了までに要する時間を短縮することができる。

1 Introduction

Recently, wireless LANs composed of mobile computers such as notebook PCs and personal digital assistants (PDAs) within which a wireless communication protocol such as IEEE802.11 [3], HIPERLAN [1] and Bluetooth [2] are so highly developed are widely available. Not only infrastructured networks which are consist of mobile computers and fixed infrastructure network with base stations but also ad hoc networks only with mobile computers are required for achieving lower construction and maintenance overhead and higher flexibility. Examples of applications in an ad hoc network environment are temporally configured networks in conventions and for disaster rescue, cooperation among autonomous mobile robots in an area where it is difficult to set base stations, sensor network and so on. In an ad hoc network, mission-critical applications are also supported as in a conventional wired networks and checkpoint recovery in one of the important methods for achieving fault-tolerant environment. However, in traditional checkpoint protocols for fixed networks is assumed that all computers have some stable storages. In addition, network bandwidth are assumed to be so high that inconsistent messages, i.e. lost and orphan messages, are detected by synchronization between sender and receiver components with low communication overhead. Hence, it is difficult to apply these conventional protocols in a mobile ad hoc network since required cost for achieving stable storages in unstable mobile computers and overhead for achieving synchronization among mobile computers which are connected unreliable, unstable and narrow wireless communication links. Therefore, in this paper, we propose a novel checkpoint protocol which achieves a stable storage by cooperation among multiple mobile computers and avoiding communication overhead due to synchronization between sender and receiver mobile computers.

2 Related Works

2.1 Checkpoint Protocol

An ad hoc network $\mathcal{N} = (\mathcal{V}, \mathcal{E})$ is a network with a set $\mathcal{V}$ of mobile computers and a set $\mathcal{E}$ of bi–directional wireless communication links $\langle M_i, M_j \rangle$ between two mobile computers M_i and M_j between which messages are exchanged directly. Generally in a computer network both wired and wireless, a global checkpoint $C_{\mathcal{V}}$ which is a set of local checkpoints c_i each of which is taken by a mobile computer $M_i \in \mathcal{V}$ is consistent if the following condition is satisfied [6].

[Definition]

1) A message m which is transmitted from a source mobile computer M_s to a destination one M_d is a lost message for a global checkpoint $C_{\mathcal{V}}$ if $Send(m)$ precedes to c_s in M_s and c_d precedes to $Receive(m)$ in M_d. Here, $Send()$ and $Receive()$ are message sending and receipt events in an application layer, respectively.
2) A message m is a lost message for $C_{\mathcal{V}}$ if c_s precedes to $Send(m)$ in M_s and $Receive(m)$ precedes to c_d in M_d.
3) A global checkpoint $C_{\mathcal{V}}$ is consistent if there is neither lost nor orphan message for $C_{\mathcal{V}}$. □

Here, if all lost messages are retransmitted after recovery, it is possible for a system to keep consistency even with recovery. Hence, a consistent global checkpoint is re–defined as follows:

[Definition]

4) A global checkpoint $C_{\mathcal{V}}$ is consistent if there are no orphan messages and all lost messages are retransmitted after recovery. □

A checkpoint protocol in [8] is designed to store lost messages into a message log in a destination computer M_d for retransmission in recovery according to this definition of a consistent global checkpoint.

Almost all conventional checkpoint protocols are designed on an assumption that each lost and orphan message is detected in a destination computer M_d of the message. Hence, virtual synchronization among all computers in a system is required. For example in a checkpoint protocol in [11], a computer is prohibited to send any application message between receipt of a checkpoint request message *CReq* and receipt of a checkpoint finish message *CFin* for avoidance of orphan messages. However, in an ad hoc network, higher synchronization overhead is required than in a conventional wired network due to narrower bandwidth of wireless communication links, reduction of transmission power of wireless signal, contention and collision caused by multiple access and longer transmission delay hidden terminal problem in multihop transmission.

Thus, longer duration suspending execution of application programs is required and processing time for a checkpoint protocol is extended. In our protocol proposed in this paper, in an ad hoc network, not only a destination mobile computer but also an intermediate one along a message transmission route detects that a message m is possible to because to become a lost or orphan message and stores or delays m as needed to achieve a consistent global checkpoint. Here, only synchronization between two neighboring mobile computers and lower synchronization overhead is required.

2.2 Mobile Checkpoint Protocol

In [14], mobile computer networks are classified into the following 4 categories:

1) Centralized Networks

2) Cell-Dependent Infrastructured Networks
3) Cell-Independent Infrastructured Networks
4) Ad hoc Networks

An infrastructure network with fixed computers and base stations as an interface between wired and wireless networks is included in 1), 2) and 3). Hence, each local checkpoint is taken by storing state information for recovery into stable storage in a computer in wired network. In [10], hybrid checkpointing realized by combination of synchronous and asynchronous protocols. Protocols in [10] and [15] are designed for 1) and one in [14] is designed for 2). In addition, protocols in [4] and [16] are designed to store local state information of mobile computers into a stable storage in a fixed computer. However in 4), since a network consists of only mobile computers, it is difficult to achieve stable storage. Hence in our protocol, local state information of a mobile computer is stored into storage of neighbor mobile computer.

3 Checkpoint Protocol

Our proposed protocol is designed under the following assumptions:

1) Any pair of mobile computers in an ad hoc network are mutually reachable with multi–hop message transmission during processing of the checkpoint protocol.
2) A wireless communication link between two mobile computers is dynamically connected and disconnected due to their mobility.
3) Each mobile computer keeps a list of neighbor mobile computers within its message transmission range up–to–date.
4) All wireless communication links between two neighboring mobile computers are bi–directional and communication along them is half–duplex.

Now, we show an outline of our checkpoint protocol. Any mobile computer in an ad hoc network initiates a checkpoint protocol. Transmission of request for taking local checkpoints and virtual synchronization of them are realized by flooding [7] of copies of a checkpoint request message *CReq* as show in Figure 1.

On receipt of *CReq*, a mobile computer M_i takes its local checkpoint c_i by storing its state information S_i and broadcasts copies of *CReq* to all its neighbor mobile computers within its message transmission range. By applying this method to all mobile computers, all mobile computers in an ad hoc network take their local checkpoints due to assumption 1). Here, since it is difficult for each mobile computer alone to implement stable storage for storing S_i, M_i asks neighbor mobile computers of M_i to store S_i for achieving stable storage. Since each mobile computer broadcasts *CReq* after taking its local checkpoint, i.e. achieving

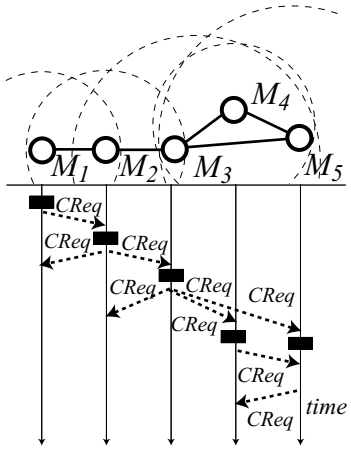

Figure 1: Checkpoint Protocol.

its local state information, no additional messages are required to transmit the state information by pigging back it to *CReq*.

[Ad hoc Checkpont Protocol (Outline)]

1) Any mobile computer M_0 initiates a checkpoint procedure. M_0 takes its local checkpoint c_0 by storing its local state information S_0 and broadcasts copies of a checkpoint request message *CReq* with an *ID* created by M_0 to which S_0 is pigged back to all mobile computers within a message transmission range of M_0.
2) M_0 sets a timer T_0.
3) A mobile computer M_j receives *CReq* from a neighbor mobile computer M_i.
4) If M_j has not yet received another *CReq* from M_i with the same *ID*, M_j stores local state information S_i of M_i carried by the received *CReq*.
5) If M_j has not yet received another *CReq* with the same *ID* from any other neighbor mobile computer, M_j takes its local checkpoint by storing local state information S_j and broadcasts *CReq* with the same *ID* assigned to the received *CReq* and to which S_j is pigged back to all mobile computers within a message transmission range of M_j. In addition, M_j sets a timer T_j.
6) If T_j times out before M_j receives *CReq* from all neighbor mobile computers, M_j broadcasts *CReq* to all mobile computers within a message transmission range again. □

Here, messages sent, forwarded and received during processing of a checkpoint protocol might be lost or orphan messages. Hence, each lost message is required to be stored in a certain mobile computer and to be retransmitted after recovery in order to keep a global state of an ad hoc network consistent. On the other hand, it is impossible to keep a global state in an ad hoc network consistent if there are some orphan

messages which is not surely retransmitted after recovery. Hence, an orphan message should be avoided in an usual message transmission protocol. If there is neither additional connections nor removed connections due to mobility of computers, during a checkpoint procedure, the following properties are satisfied:

[Properties]

1) One of the following properties is surely satisfied for any lost message m_l along its transmission route:
   1-a) In at least one of mobile computers M_i along a transmission route of m_l, $receive(m_l) \rightarrow c_i \rightarrow send(m_l)$ is satisfied. Here, $\rightarrow$ represents happen before relation between two communication events.
   1-b) For two mobile computers M_i and M_j which are included in a message transmission route of m_l and between which messages are directly exchanged, both $send(m_l) \rightarrow c_i$ and $c_j \rightarrow receive(m_l)$ are satisfied.
2) The following property is surely satisfied for any orphan message m_0 along it message transmission route:
   2-a) For two mobile computers M_i and M_j which are included in a message transmission route of m_0 and between which messages are directly exchanged, both $c_i \rightarrow send(m_0)$ and $receive(m_0) \rightarrow c_j$ is satisfied. □

According to property 1), an intermediate mobile computer along a message transmission route detects a message m_l which is possible to be a lost message. If this detection is realized only in a destination mobile computer M_d, e.g. only M_r detects m_l to be a lost message in [6] and [11], M_d has already broadcasts *CReq* to all mobile computers within a message transmission range of M_d before $Receive(m_l)$ where M_d receives m_l. In this case, m_l has to be broadcasted to all mobile computers within a message transmission range of M_d in order to be stored into these mobile computers. However, there occur the following problems:

(1) Messages broadcasted to all neighbor mobile computers for M_d to be stored for recovery is increased.

(2) When m_l is broadcasted, mobile computers which are included in a message transmission range of M_d and receive and store local state information S_d of M_d are not always within a transmission range of M_d due to their mobility. Hence, some mobile computer keep only S_0 and other mobile computers keep only m_l. Therefore, information required for recovery in M_d is distributed into multiple mobile computers and high communication overhead is required in recovery.

(3) Additional synchronization messages are required to be exchanged among mobile computers in order to detect that all lost messages required to be retransmitted in recovery for achieving a consistent global state are stored in some mobile computers and a checkpoint protocol terminates.

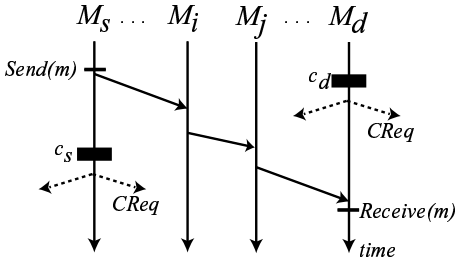

Figure 2: Delayed Lost Message Detection Problem.

Therefore, in our proposed checkpoint protocol, a message m_l which is possible to be a lost message (we call m_l an pseudo lost message) is detected by an intermediate mobile computer along a message transmission route of m_l before it sends *CReq*. For a message m_l satisfying property 1–a), a mobile computer M_i in property 1–a) detects m_l before M_i broadcasts *CReq*.

[Detection of Pseudo Lost Message (1)]
If a mobile computer M_i receives a message m_l which is sent out by another mobile computer and transmitted through M_i, i.e. forwarded by M_i, before broadcasting *CReq* and has not yet sent m_l before broadcasting *CReq*, m_l is a pseudo lost message. M_i broadcast *CReq* to which local state information M_i and m_l are pigged back to all mobile computers within a message transmission range of M_i. On receipt of this *CReq*, a neighbor mobile computer M_j of M_i stores S_i and m_l for recovery of M_i. □

On the other hand, for a message m_l satisfying property 1–b), a mobile computer M_j in 1–b) detects m_l to be a pseudo lost message. However, before receipt of m_l, M_j might have broadcasted *CReq*. Hence, in our checkpoint protocol, detection of a pseudo lost message is realized in M_j and M_i which has not yet broadcasted *CReq* before sending m_l stores m_l before broadcasting *CReq* as shown in Figure 3.

[Detection of Pseudo Lost Message (2)]
If a mobile computer M_j detects a message m_l from M_i to be a pseudo lost message which is sent before broadcast of *CReq* in M_i and received after broadcast of *CReq* in M_j, M_j informs M_i that m_l is a pseudo lost message by using an acknowledgment message $ack(m)$ of m_l. By receiving such $ack(m_l)$, M_i is informed that m_l is a pseudo lost message and in a next broadcast of *CReq*, m_l is pigged back to the *CReq*. In order for this method to be realized in our protocol, each mobile computer is prohibited to broadcast *CReq* between

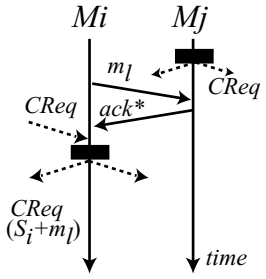

Figure 3: Lost Message Detection in Intermediate Mobile Computer.

transmission of a message and receipt of its acknowledgment message. □

An intermediate mobile computer only forward a message that is possible to be an orphan message. By delaying acceptance of the message in an application layer in a destination mobile computer, i.e. *Receive*() event, it is achieved to avoid orphan messages.

[Message Transmission Protocol]
In a mobile computer M_i, a message sending event *Send*() and a message receipt event *Receive*() in an application layer occur if an application program requests to send and receive a message, respectively. On the other hand, on receipt of a message from a network, a message receipt event *receive*() in a network layer occurs. A message sending event *send*() in a network layer occurs in *Send*() in a source mobile computer and in *receive*() in an intermediate mobile computer.

(*Send*(m) in an application layer)
1) $m.logged := false$.
2) If M_i has already sent a checkpoint request message *CReq*, $m.source\_creq\_sent := true$.
Otherwise, $m.source\_creq\_sent := false$.
3) M_i invokes *send*(m).

(*Receive*(m) in an application layer)
1) If $m \notin buffer_i$, M_i suspends this procedure until M_i finishes a procedure for *receive*(m).
2) If $m.source\_creq\_sent = true$ and M_i has not yet sent *CReq*, M_i suspends this procedure until M_i sends *CReq*.
3) M_i removes m from $buffer_i$.
4) If $m.logged = true$ and M_i has not yet sent *CReq*, m is included in a set of messages each of which is discarded even if it is received after recovery.

(*send*(m) in a network layer)
1) If M_i has already sent *CReq*, $m.creq\_sent := true$.
Otherwise, $m.creq\_sent := false$.
2) M_i blocks to send *CReq* even if it receives *CReq* until finish of this procedure.
3) M_i sends m.
4) M_i receives an acknowledgment message $ack(m)$ for m. If $m.logged = false$ and $ack(m).logged = true$, m is pigged back to a next transmitted *CReq*

(*receive*(m) in a network layer)
1) If $m.logged = false$, $m.creq\_sent = false$ and M_i has already set *CReq*, $m.logged := true$ and M_i sends back $ack(m)$ where $ack(m).logged := true$.
Otherwise, M_i sends back $ack(m)$ where $ack(m).logged := m.logged$.
2) If M_i is a destination of m, m is stored into $buffer_i$.
Otherwise, M_i invokes *send*(m) for forwarding m.
□

Messages which are to be pigged back to *CReq* from M_i are stored into a message log ML_i.

[Ad hoc Checkpoint Protocol]
1) Any mobile computer M_0 initiates a checkpoint procedure. M_0 takes its local checkpoint by storing its local state information S_0 and broadcasts a checkpoint request message *CReq* to which S_0 and a message log ML_0 of M_0 are pigged back to all mobile computers with a message transmission range of M_0.
2) M_0 sets a timer T_0.
3) A mobile computer M_j receive *CReq* from M_i.
4) If M_j has not yet received another *CReq* from M_i with the same *ID*, M_j stores local state information S_i of M_i and a message log ML_i carried by *CReq*.
5) If M_j has not yet received another *CReq* with the same *ID* from any other neighbor mobile computer, M_j takes its local checkpoint by soring its local state information S_j and broadcasts *CReq* with the same *ID* assigned to the received *CReq* and to which S_j and a message log ML_j of M_j are pigged back to all mobile computers within a message transmission range of M_j. In addition, M_j sets a timer T_j.
6) If T_j times out before M_j receives *CReq* from all neighbor mobile computers, M_j broadcasts *CReq* to all mobile computers within a message transmission range again.
7) Otherwise, $ML_i := \phi$ □

4 Concluding Remarks

This paper has proposed a novel ad hoc checkpoint protocol in which detection of lost messages is not based on end-to-end but hop-by-hop, i.e. an intermediate mobile computer detects and stores pseudo lost messages which are possible to be lost messages. By introducing this method, each mobile computer is required to broadcast a checkpoint request message only once in a checkpoint procedure. That is, lower communication and synchronization overhead is required. If

wireless communication links between mobile computers are connected and/or disconnected during a checkpoint procedure due to mobility, there may exist lost and/or orphan messages which do not satisfy properties 1) and 2). In future work, we design an extended protocol to achieve consist global checkpoint even with such messages.

References

[1] "Radio Equipment and Systems (RES); HIPERLAN," ETSI,Functional Specifications (1995).

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

[3] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," Standard IEEE 802.11 (1999).

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

[5] Callaway, E.M., "Wireless Sensor Networks," Auerbach Publications (2003).

[6] 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).

[7] Corson, M.S. and Ephremides, A., "A Distributed Routing Algorithm for Mobile Wireless Networks," ACM Journal of Wireless Networks, vol. 1, No. 1, pp. 61–81 (1995).

[8] Elnozahy, E.N. and Zwaenepoel, W., "On the use and Implementation of Message Logging," Proc. of the Fault-Tolerant Computing Symposium, pp. 298–307(1994).

[9] Gendelman, E., Bic, L.F. and Dillencourt, M.B., "An efficient checkpointing algorithm for distributed systems implementing reliable communication channels," Proc. of 18th International Symposium on Reliable Distributed Systems, pp. 290–291 (1999).

[10] 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).

[11] 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).

[12] Lampson, B.W., Paul, M. and Siegert, H.J., "Distributed Systems – Architecture and Implementation," Springer–Verlag, pp. 246–265 (1981).

[13] Lesser, V., Ortiz, C.L. and Tambe, M., "Distributed Sensor Networks," Kluwer Academic Publications (2003).

[14] 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).

[15] 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).

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

[17] Yao, B. and Fuchs, W.K., "Message logging optimization for wireless networks," Proc. of the 20th International Symposium on Reliable Distributed Systems, pp. 182–185 (2001).

[18] Ono, M. and Higaki, H., "Checkpoint Protocol for Mobile Ad–hoc Networks," IPSJ SIG MBL, Vol. 2003, No. 93, pp. 91–96 (2003).