

アドホックネットワークのためのチェックポイントプロトコル

東京電機大学 理工学部 情報システム工学科

小野 真和 森田 義徳 桧垣 博章

E-mail: {masa,mine,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、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)の検出、回避を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実現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。状態情報は複数の移動コンピュータに記憶し、マルチホップ配送されるメッセージが孤児メッセージとならないように転送タイミングを制御する。コンピュータの移動により孤児メッセージになる可能性のあるメッセージは、中継コンピュータによる判断を保留する。これによって、同期オーバーヘッドが大幅に削減される。

Checkpoint Protocol in Ad-hoc Networks with Localized Synchronizarion

Masakazu Ono, Yoshinori Morita and Hiroaki Higaki

Department of Computers and Systems Engineering

Tokyo Denki University

E-mail: {masa,mine,hig}@higlab.k.dendai.ac.jp

For achieving mission critical network applications, checkpoint recovery protocols have been researched and developed. In coventional 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.

1 はじめに

ノート型コンピュータやPDAなどの移動コンピュータや、IEEE802.11やHIPERLANなどの無線通信プロトコルの研究開発が進み、広く普及している。また、無線基地局を介して有線ネットワークと接続されたインフラストラクチャネットワークだけでなく、移動コンピュータだけで構成されるアドホックネットワークへの要求が高まっている。アドホックネットワークの適用例として、一時的に構築されるイベント会場や災害現場などでのネットワーク、危険地帯で無線基地局が設置できない箇所における小型ロボットの協調動作のためのネットワークがある。このようなアドホックネットワークにおいて、ミッションクリティカルアプリケーションの実行を考えたとき、耐故障性を実現するためのチェックポイントリカバリ手法を適用するこ

とが考えられる。しかし、従来のチェックポイント手法は安定記憶がネットワーク上に存在することを前提としており、また、一貫性のないメッセージ(孤児メッセージ、紛失メッセージ)を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検出することが可能な帯域幅がネットワークによって提供されているとしている。そのため、アドホックネットワーク上の移動コンピュータは安定記憶を持っていない問題や、狭帯域幅で低信頼なネットワークによる同期のための通信オーバーヘッドの拡大の問題を解決する必要がある。そこで、本論文ではアドホックネットワークにおいて安定記憶を実現し、送受信コンピュータ間における同期のための通信オーバーヘッドを回避する新たなチェックポイントプロトコルを提案する。

2 従来手法

2.1 チェックポイントプロトコル

チェックポイントプロトコルによって、各移動コンピュータ $M_i \in V$ が設定したローカルチェックポイント c_i の集合であるグローバルチェックポイント C_V が一貫性を持つとは、次の性質を満たすことをいう。

[定義]

- 1) M_s から M_r へ配送されるメッセージ m が紛失メッセージであるとは、グローバルチェックポイント C_V に対して、 $Send(m)$ が c_s に先行し、 c_r が $Receive(m)$ に先行することである。なお、 $Send()$ と $Receive()$ は、アプリケーションにおけるメッセージ送受信イベントである。
- 2) メッセージ m が孤児メッセージであるとは、 C_V に対して、 c_s が $Send(m)$ に先行し、 $Receive(m)$ が c_r に先行することである。
- 3) 紛失メッセージ、孤児メッセージを含まないグローバルチェックポイントは、一貫性が保たれているという。□

ただし、紛失メッセージをリカバリ回復時に再送信することができれば、システム状態の一貫性を維持することが可能である。そこで、一貫性のあるグローバルチェックポイントを以下のように再定義する。

[定義]

- 4) 一貫性のあるグローバルチェックポイントとは、孤児メッセージを含まず、すべての紛失メッセージをリカバリ回復時に再送信可能であるものである。□

従来のチェックポイントプロトコルにおいては、 m が C_V に対して紛失メッセージや孤児メッセージとなることを M_r でのみ判定することを前提としている。そのため、これらの発生を回避するには、システム全体での同期を必要としていた。例えば、Koo [2] のプロトコルにおいては、チェックポイント要求メッセージ $CReq$ を受信してから、チェックポイント終了メッセージ $CFin$ を受信するまでの間、アプリケーションメッセージの送信を禁止している。ところが、アドホックネットワークにおいては、無線通信の狭帯域幅、低信頼性といった性質により、同期のコストが大きく、アプリケーションの停止時間が長くなるという問題がある。そこで、アドホックネットワークのためのチェックポイントプロトコルには、以下の条件が満たされることが求められる。

[要求条件]

- 1) m が紛失メッセージあるいは孤児メッセージとなる可能性を、 m の配送経路上にある移動コンピュータが判定し、必要に応じて m を記憶したり、 m の転送を遅延させたりする。□

ここでは、隣接する移動コンピュータ間における同期のみが必要であることから、アプリケーションの停止時間を短縮することが可能である。また、チェックポイントプロトコルの制御メッセージの転送をアプリケーションの実行が待たせることがないことから、チェックポイントプロトコルによる移動コンピュータへの処理オーバーヘッドも小さいという特性を持つ。

2.2 モバイルチェックポイントプロトコル

モバイルチェックポイントプロトコルの実現にあたり、論文 [4] では、移動コンピュータを含むネットワークを以下の4つのモデルに分類している。

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

1)~3) は、ネットワークの構成要素に固定コンピュータを含んでいる。そこで、固定コンピュータに実現した安定記憶に移動コンピュータの状態情報を保存することにより、チェックポイントを設定することができる。論文 [1] では、同期チェックポイント手法と非同期チェックポイント手法を組み合わせた複合チェックポイント手法を提案している。[1, 3] および [4] において、それぞれ 1)、2) に対するプロトコルを設計している。ところが 4) においては、ネットワークの構成要素が移動コンピュータのみであることから、安定記憶の実現が困難である。したがって、複合チェックポイント手法で用いたメッセージログによる再実行は適切とはいえないことから、移動コンピュータのチェックポイントは、同期型チェックポイントプロトコルによって設定されることが必要である。そこで、以下の要求条件を満たすプロトコルを構成する。

[要求条件]

- 2) 各移動コンピュータのチェックポイントの設定は、複数の移動コンピュータに状態情報を記憶させることによって実現される。□

3 提案プロトコル

以下の条件のもとでプロトコルを構成する。

[前提条件]

- 1) アドホックネットワークに含まれるすべての移動コンピュータは、チェックポイントプロトコルの

実行中、マルチホップ配送により互いにメッセージ交換が可能である。

- 2) 隣接する移動コンピュータ間の通信リンクは、動的に切断および確立されることがある。
- 3) 各移動コンピュータは、隣接する移動コンピュータのリストを保持している。
- 4) 隣接する移動コンピュータ間の通信リンクは双方向であり、半二重通信が行なわれる。また、ユニキャスト通信は、受信確認と再送機構により、メッセージの紛失なく実現されているものとする。□

チェックポイントプロトコルの基本形を示す。チェックポイントプロトコルの開始は、任意の移動コンピュータが任意のタイミングで行なうことができる。チェックポイント設定要求の伝達と、チェックポイント間の同期は、チェックポイント設定要求メッセージ $CRReq$ のフラグディングによって実現される(図1)。 $CRReq$ を受信した移動コンピュータは、直ちにチェックポイントを設定するとともに、そのメッセージを隣接する移動コンピュータ群にブロードキャストする。これを繰り返すことによって、前提1)により、すべての移動コンピュータにおいて、ローカルチェックポイントを設定することができる。

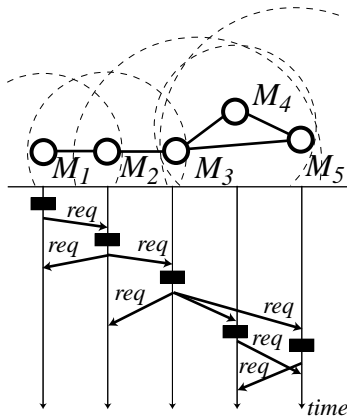


図1: チェックポイントプロトコル

[アドホックチェックポイントプロトコル(基本形)]

- 1) 任意の移動コンピュータ M_0 が、自身の状態情報 S_0 を保存するとともに、 S_0 を含むチェックポイント設定要求メッセージ $CRReq$ を自身の無線信号到達範囲内にブロードキャストする。このとき、タイマ T_0 をセットする。
- 2) 移動コンピュータ M_i が送信したチェックポイント設定要求メッセージ $CRReq$ を受信した移動コンピュータ M_j は、以下の処理を行なう。なお、こ

の処理を終えるまで、 M_i はアプリケーションメッセージを送信しないこととする。

- 2-1) M_i から同一の ID を持つ $CRReq$ を受信していないならば、受信した $CRReq$ に含まれる M_i の状態情報 S_i を保存する。
- 2-2) M_j が一度も $CRReq$ を受信していないならば、自身の状態情報 S_j を保存するとともに、 S_j を含む $CRReq$ を自身の無線信号到達範囲内にブロードキャストする。このとき、タイマ T_j をセットする。
- 3) 移動コンピュータ M_j が近隣する移動コンピュータリスト L_j に含まれるすべての移動コンピュータから $CRReq$ を受信する以前にタイマ T_j が時間切れとなったならば、 M_j は、同じ $CRReq$ を再度ブロードキャストする。□

ここで、チェックポイントプロトコルの実行と並行に送受信されたメッセージは、紛失メッセージや孤児メッセージとなる可能性がある。紛失メッセージは、いずれかの移動コンピュータに保存し、リカバリ回復時に、保存されたメッセージを再送信することによって、システム状態の一貫性を維持することができる。一方、孤児メッセージは、リカバリ再実行時に送信元移動コンピュータが同一のメッセージを再度送信する保障がないことから、その発生を回避しなければならない。移動コンピュータの移動による通信路の切断と接続が発生しない場合には、以下の性質が成り立つ(図2)。

[性質]

- 1) 紛失メッセージ m_l の配送経路上にある1台以上の移動コンピュータ M_i において、 $\langle receive(m_l), receive(CRReq), send(CRReq), send(m_l) \rangle$ の順に通信イベントが発生する。
- 2) 孤児メッセージ m_o の配送経路上にある1台以上の移動コンピュータ M_i において、 $\langle receive(CRReq), receive(m_o), send(m_o), send(CRReq) \rangle$ の順に通信イベントが発生する。□

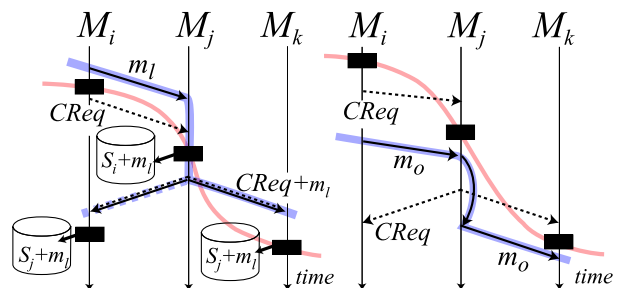


図2: 紛失メッセージと孤児メッセージの検出

性質1)により、 M_i において、紛失メッセージとなる可能性のあるメッセージ m_l を $CReq$ の送信前に検出することができる。 M_i は、 m_l を記憶するとともに、 m_l を $CReq$ にピギーバックする。 $CReq$ を受信した M_i の隣接する移動コンピュータにおいても、 m_l を記憶することによって、リカバリ回復時に m_l が再送されることが保障できる。また、性質2)により、 M_i において、孤児メッセージとなる可能性のあるメッセージ m_o を $CReq$ の送信前に検出することができる。そこで、 $CReq$ を受信した後、 $CReq$ を送信する前に受信した m_o の送信を $CReq$ の送信後まで延期することによって、孤児メッセージの発生を防ぐことができる。なお、いずれの場合も $CReq$ の送信を遅延させることがないことから、チェックポイントプロトコルの進行がアプリケーションメッセージの送受信によって遅延されることがない。これは、帯域幅が必ずしも十分ではなく、信頼性の低い移動ネットワーク環境において、チェックポイントプロトコルを短時間で終結させることによって、アプリケーションへの負荷を低減させるという点で重要な性質である。

[アドホックチェックポイントプロトコル (追加部分)]

- 1) M_j が $CReq$ を受信した時、転送待ちとなっているメッセージ m があるならば、 m を $CReq$ にピギーバックして送信する。
- 2) 1) の $CReq$ を受信した M_k は、 S_j とともに m も記憶する。
- 3) リカバリ回復時には、 M_j および M_k が m を再送信する。
- 4) M_j が $CReq$ を受信してから $CReq$ を送信するまでの間に受信したメッセージ m の転送は、 $CReq$ の送信後まで延期する。□

移動コンピュータ間のリンクが切断、確立することにより、図3に示すように、性質1)2)を満たさない紛失メッセージ、孤児メッセージが発生する。そこで、以下の条件を満たすメッセージは、紛失メッセージ、孤児メッセージとなる可能性のあるメッセージであると判定する。

[性質]

- 3) m_{pl} を M_j から受信した M_k において m_{pl} を受信する直前のチェックポイント c_k を設定したとき、 M_i が M_k の隣接する移動コンピュータリストに含まれていなかった場合、 m_{pl} は紛失メッセージとなる可能性がある。
- 4) m_{po} を M_k に送信した M_i において m_{po} を送信する直前のチェックポイント c_i を設定したとき、

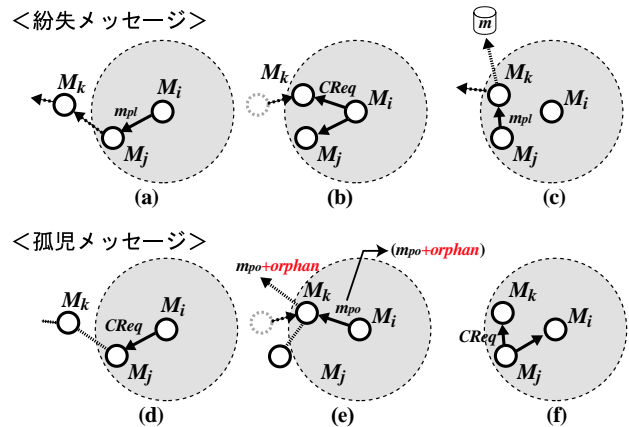


図3: 紛失/孤児メッセージになり得るメッセージ

M_k が M_i の隣接する移動コンピュータリストに含まれていなかった場合、 m_{po} は孤児メッセージとなる可能性がある。□

紛失メッセージとなる可能性のある m_{pl} は、 M_k によって保存される (図3(c))。一方、孤児メッセージとなる可能性のある m_{po} には、孤児メッセージとなる可能性があることを示す情報を添付し (図3(d))、送信先移動コンピュータで、ローカルチェックポイント設定手続きを終えるまでアプリケーションでの受信を遅延する。

4 まとめ

本論文では、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の概略を示した。紛失メッセージ、孤児メッセージとなる可能性を、エンド-エンドではなく、ホップバイホップで検証する機構を導入することにより、チェックポイントプロトコル動作に要するオーバーヘッドを削減することが可能である。

参考文献

- [1] Higaki, H. and Takizawa, M., "Checkpoint-Recovery Protocol for Reliable Mobile Systems," Proc. of the 17th SRDS, pp.93-99 (1998).
- [2] 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).
- [3] Morita, Y. and Higaki, H., "Checkpoint-Recovery for Mobile Computing Systems," Proc. of the 21st ICDCS Workshops, pp.479-484 (2001).
- [4] Miyazaki, M., Morita, Y. and Higaki, H., "Hybrid Checkpoint Protocol for Mobile Networks with Unreliable Wireless Communication Channels," Proc. of the 2nd AMOC, pp.164-171 (2002).