

複数通信モデル混在環境におけるマルチメディアチェックポイントの 一貫性評価手法

東京電機大学 理工学部 情報システム工学科

木下 稔也 小野 真和 桧垣 博章

E-mail: {kino, masa, hig}@higlab.net

ネットワークシステムの耐故障性を高める手法としてチェックポイントリカバリの研究が広く進められている。システムに故障が発生した後にチェックポイントが定めるグローバル状態から正しく処理を再開するためには、各プロセスが設定したチェックポイントの集合が一貫性をもつものであることが必要となる。これまでに、マルチメディアネットワークシステムのための一貫性評価が提案されているが、これらはすべてのプロセスが限られた通信モデルに従う場合のみを想定して設計されていた。本論文では異なる通信モデルが混在した環境においても正しく一貫性を評価するために、誘導孤児パケットを導入し、これを一貫性評価に含める手法を提案する。

Consistency Evaluation of Multimedia Checkpoints with Bulk and Stream Communication Events

Toshiya Kinoshita, Masakazu Ono and Hiroaki Higaki

Department of Computers and Systems Engineering

Tokyo Denki University

E-mail: {kino, masa, hig}@higlab.net

For achieving mission-critical property in multimedia communication network systems, the authors have designed a series of checkpoint protocols. A global state determined by a set of local checkpoints called a global checkpoint is required to be consistent for reasonable recovery. Due to much larger multimedia messages than conventional data messages, a local checkpoint is required to be taken even during a communication event. For supporting such situation, the authors have proposed a novel criteria for evaluation of consistency. Though it assumes that every communication event is instantaneous, some events are consecutive. In this paper, the authors classify multimedia communication events into instantaneous and consecutive and propose new criteria based on efficient of induced orphan packets reducing consistency of global checkpoint.

1 背景と目的

障害に対して耐性を持つネットワークアプリケーション実行環境を実現する手法のひとつとしてチェックポイントリカバリ手法が研究されている。ここでは、ネットワークアプリケーションを実行する各プロセスが正常動作時の状態情報を安定記憶に保存し、障害発生時にはシステム内のすべてのプロセスが保存されている状態から処理を再開する。ただし、各プロセスが設定したチェックポイントの集合は、リカバリ後のシステム動作に矛盾を発生させない一貫性のあるグローバル状態を定めることが求められる [1]。一方、メッセージサイズが大きく、ひとつのメッセージが複数のパケットの集合として送受信されるマルチメディアネットワークにおいては、メッセージの一部がリカバリ時に失われてもシステム動作に必ずしも矛盾を発生しないことに注目し、リカバリ時に失われるパケット群に基づいた新しい一貫性の評価方法が提案されている [2]。ただし、すべての通信イベントが、送信時にはメッセージを構成するすべてのパケットが送信開始時にアプリケーションによって決定される一括送信モデルに従い、受信時にはメッセージを構成するすべてのパケットを受信した時点でアプリケーションに受理されるという一括受信モデルに従うことを前提としている。本論文では、送信されるマルチメディアメッセージを構成するパケットがパケット送信時点で決定される順次送信イベント、各パケットを受信した時点でアプリケーションの受理を可とする順次受信イベントをも含む場合のグローバル状態の一貫性評価方法を提案する。

2 マルチメディアネットワーク

2.1 マルチメディアメッセージ

音声や動画像などのマルチメディアデータを対象とした配送と処理を行なうマルチメディアネットワークにおいて送受信されるメッセージは、これまでのデータ通信において送受信されるメッセージに比べてメッセージサイズが大きく、メッセージ送受信イベントを瞬時に終わることができないという特徴がある。

マルチメディアメッセージ m は複数のパケット $\langle pa_0, \dots, pa_{l-1} \rangle$ に分割されて配送される。プロセス p_i から p_j へ配送されるマルチメディアメッセージ m の送受信中におけるイベントは、以下の通りである。

- $sb(m)$: プロセス p_i が m の送信を開始
- $se(m)$: プロセス p_i が m の送信を終了
- $rb(m)$: プロセス p_j が m の受信を開始

- $re(m)$: プロセス p_j が m の受信を終了

p_i におけるメッセージ m の送信イベント $send(m)$ は、 $sb(m)$ で開始され、 $se(m)$ で終了する。また、 p_j におけるメッセージ m の受信イベント $receive(m)$ は、 $rb(m)$ で開始され、 $re(m)$ で終了する。

2.2 マルチメディア通信イベントモデル

2.2.1 送信イベントモデル

パケット列 $\langle pa_0, \dots, pa_{l-1} \rangle$ からなるマルチメディアメッセージ m がアプリケーションから送信される送信イベントには、一括送信イベントと順次送信イベントがある。

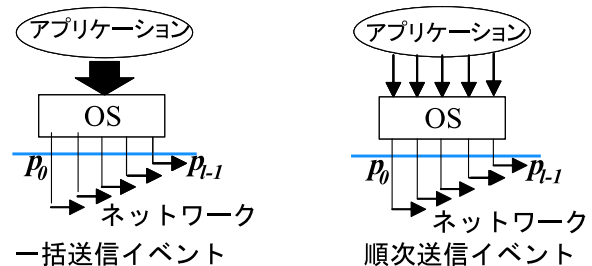


図 1: マルチメディア送信イベント

[一括送信イベント]

一括送信イベントでは、アプリケーションが送信イベント開始時に pa_0, \dots, pa_{l-1} をすべて決定している。その状態で、図 1 のように、 m の全体がオペレーティングシステムに渡され、オペレーティングシステムが pa_0, \dots, pa_{l-1} に分割してネットワークに送信する。このモデルでは、受信したメッセージやパケットが m の送信中にオペレーティングシステムからアプリケーションに渡されて受信処理がされたとしても、これらが m の内容に影響を与えることはない。□

[順次送信イベント]

順次送信イベントでは、 pa_0, \dots, pa_{l-1} が順次決定されている。アプリケーションは pa_i を決定するとともにオペレーティングシステムに渡し、オペレーティングシステムがネットワークに送信する。受信したメッセージやパケットが m の送信中にオペレーティングシステムからアプリケーションへ渡され、受信処理がされると、それ以降に決定される m の中のパケットに影響を与えることがある。□

2.2.2 受信イベントモデル

パケット列 $\langle pa_0, \dots, pa_{l-1} \rangle$ からなるマルチメディアメッセージ m がアプリケーションに受信される受信イベントには、一括受信イベントと順次受信イベントが

ある。

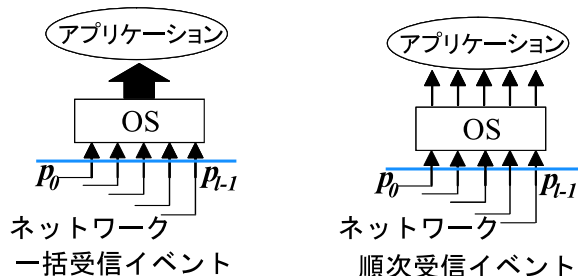


図 2: マルチメディア受信イベント

[一括受信イベント]

一括受信イベントでは、 m の全体をオペレーティングシステムが受信してからアプリケーションに渡され、受信処理がされる。このため、 pa_0 を受信してから pa_{l-1} を受信するまでの間にアプリケーションがオペレーティングシステムに送信要求したメッセージやパケットの決定に対して m の内容が影響を与えることはない。□

[順次受信イベント]

順次受信イベントでは、オペレーティングシステムが m 全体の受信を待つことなく、各パケットを受信するごとにアプリケーションに渡し、受信処理をおこなう。このため、 pa_0 を受信してから pa_{l-1} を受信するまでの間にアプリケーションがオペレーティングシステムに送信要求したメッセージやパケットの決定に対して、その決定以前にアプリケーションに渡されたパケット pa_i が影響を与えることがある。□

なお、各メッセージがいずれの送信モデルに従って送信され、いずれの受信モデルに従って受信されるかは、通信イベントが発生する各プロセスによって決定されるものであり、メッセージによって定まるものではない。したがって、一括送信イベントで送信されたメッセージが順次受信イベントで受信されることも、順次送信イベントで送信されたメッセージが一括受信イベントで受信されることも許される。

3 従来手法

コンピュータネットワークシステムにおいて耐故障性を実現する手法としてチェックポイントリカバリ手法が提案されている。チェックポイントリカバリ手法は、チェックポイントの設定手法とリカバリ手法から成る。アプリケーションを実行するプロセスは、正常動作時のアプリケーションの変数値やレジスタ値などの状態情報を安定記憶に保存し、チェックポイントを設定する。システム内に故障が発生した場合には、システム内の

すべてのプロセスが保存しておいた状態情報を読み込むことでチェックポイント設定時の状態を再現し、アプリケーションの実行を再開することでリカバリする。

ここで、安定記憶とは、故障独立した記憶装置を多重化することにより、故障が発生したとしても保存しておいた状態情報が失われないという記憶装置である。ただし、リカバリ後のシステムが正しく処理を再開するためには、プロセスが設定したチェックポイントの集合で定まるグローバル状態が一貫性を満たしたものである必要がある。

論文 [1] では、メッセージをプロセス間通信の単位とし、各プロセスのローカルチェックポイントをメッセージの送受信イベント中には設定できないという仮定のもとで、以下の紛失メッセージ、孤児メッセージが存在しないグローバルチェックポイントのみ一貫性があると定義している。

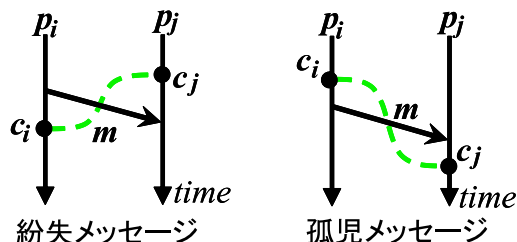


図 3: 紛失メッセージと孤児メッセージ

[紛失メッセージ]

チェックポイント c_i を設定したプロセス p_i からチェックポイント c_j を設定したプロセス p_j へ送信されたメッセージ m が、 $send(m) \rightarrow c_i$ かつ $c_j \rightarrow receive(m)$ を満たすとき、 m を紛失メッセージという。□

[孤児メッセージ]

チェックポイント c_i を設定したプロセス p_i からチェックポイント c_j を設定したプロセス p_j へ送信されたメッセージ m が、 $c_i \rightarrow send(m)$ かつ $receive(m) \rightarrow c_j$ を満たすとき、 m を孤児メッセージという。□

これに対して、メッセージを構成するパケットをプロセス間通信の単位とすることにより、各プロセスがローカルチェックポイントをメッセージの送受信中に設定することが可能となる。ただし、これによって以下の紛失パケット、孤児パケットが発生する。

[紛失パケット]

チェックポイント c_i を設定したプロセス p_i からチェックポイント c_j を設定したプロセス p_j へ送信されたメッセージ m がパケット $\langle pa_1, pa_2, \dots, pa_{l-1} \rangle$ で構成

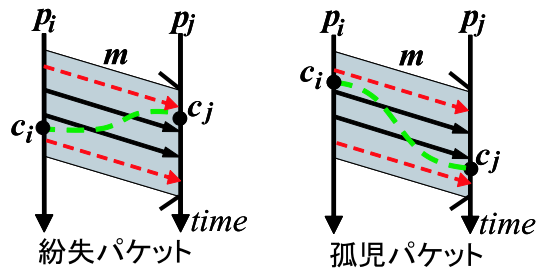


図 4: 紛失パケットと孤児パケット

されているとする。このとき、 $send(pa_k) \rightarrow c_i$ かつ $c_j \rightarrow receive(pa_k)$ を満たすとき、 pa_k を紛失パケットという。□

[孤児パケット]

チェックポイント c_i を設定したプロセス p_i からチェックポイント c_j を設定したプロセス p_j へ送信されたメッセージ m がパケット $\langle pa_1, pa_2, \dots, pa_{l-1} \rangle$ で構成されているとする。このとき、 $c_i \rightarrow send(pa_k)$ かつ $receive(pa_k) \rightarrow c_j$ を満たすとき、 pa_k を孤児パケットという。□

論文 [2] では、通信イベントが一括送信イベントと一括受信イベントのみからなる通信モデルのもとで、以下の性質が成り立つとしている。

[メッセージの一貫性]

孤児パケットはリカバリ回復時に再送信されることから、メッセージの一貫性を低下させない。紛失パケットはリカバリ回復時に再送信されないことから、メッセージの一貫性を低下させる。□

[メッセージの一貫性低下の独立性]

あるメッセージの孤児パケット、紛失パケットが他のメッセージの一貫性に影響を与えない。□

この性質に基づくと、各プロセスが送受信したメッセージの一貫性を紛失パケットの価値に基づいて他のプロセスとは独立に評価し、これらの統合によってグローバルチェックポイントの一貫性を評価できる [2]。

4 誘導孤児パケット

通信イベントが一括送信イベントと一括受信イベントのみからなるマルチメディアネットワークシステムでは、前章で述べた性質によって各メッセージの一貫性を各プロセス独立に評価したものを統合してグローバルチェックポイントの一貫性を評価することができる。しかし、順次送信イベント、順次受信イベントをも含むマルチメディアネットワークシステムにおいては、この性質が必ずしも成り立たない。例えば図 5 において、 p_1

は m_1 を一括送信、 p_2 は m_1 を順次受信、 m_2 を順次送信、 p_3 は m_2 を一括受信しているものとする。ローカルチェックポイント c_1, c_2, c_3 と m_1, m_2 に含まれるパケットの送受信イベントとの順序関係から、紛失パケット $lp_i \in m_1$ と孤児パケット $op_i \in m_2$ が存在する。ここで、 c_2 では m_1 が順次受信、 m_2 が順次送信されることから、 $p_{k_1} \in m_1, p_{k_2} \in m_2$ について $receive(p_{k_1}) \rightarrow send(p_{k_2})$ が成り立つとき、 p_{k_2} は p_{k_1} の影響を受けることがある。 lp_i はリカバリ回復後に再送信されないことから、 c_2 において $receive(lp_i) \rightarrow send(p_{k_2})$ を満たすパケット p_{k_2} はリカバリ回復後に再送信されないことや内容の異なるパケットが再送信されることがある。そのため、 $receive(lp_i) \rightarrow send(op_i)$ となる op_i は m_2 の一貫性を低下させる。これは、論文 [2] では想定されていない。以降では、このパケットを誘導孤児パケットとよぶ。誘導孤児パケットが発生するのは、以下の条件が成り立つ場合である。

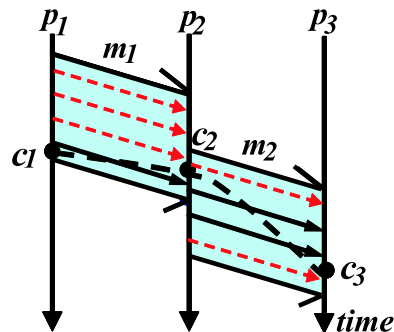


図 5: 誘導孤児パケット

[誘導孤児パケット発生条件]

- 1) p_1 は m_1 を一括送信または順次送信、 p_2 は m_2 を順次送信、 p_3 は m_2 を一括受信または順次受信する。
- 2) 次のいずれかが成り立つ
 - p_2 において $sb(m_2) \rightarrow re(m_1) \rightarrow se(m_2)$ が成り立つ。 m_2 の受信イベントは一括受信でも順次受信でもよい。(図 6(a) および図 6(b))
 - p_2 において $rb(m_1) \rightarrow se(m_2) \rightarrow re(m_1)$ が成り立つ。 m_2 の受信イベントは順次受信である。(図 6(c) および図 6(d)) □

先に述べたそれぞれの条件において、どのようにチェックポイントが設定された場合に誘導孤児パケットが発生するかについて述べる。図 6(a), (b) のようなメッセージの送受信を行なっているとき、

- p_2 における受信イベントが一括受信である場合:

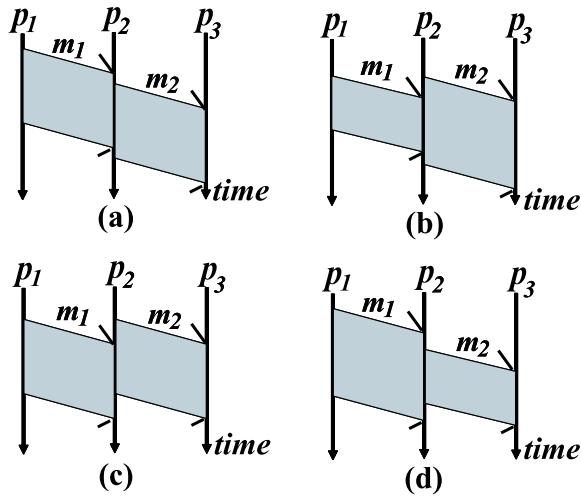


図 6: 誘導孤児パケットの発生条件

$\exists pa_i \in m_1$ について $send(pa_i) \rightarrow c_1$ かつ $sb(m_2) \rightarrow c_2 \rightarrow receive(pa_i)$ を満足し、 $pa_j \in m_2$ について $re(m_1) \rightarrow send(pa_j)$ 、 $c_2 \rightarrow send(pa_j)$ 、 $receive(pa_j) \rightarrow c_3$ を満足するとき、 pa_j は誘導孤児パケットである。

- p_2 における受信イベントが順次受信である場合:
 $\exists pa_i \in m_1$ について $send(pa_i) \rightarrow c_1$ かつ $sb(m_2) \rightarrow c_2 \rightarrow receive(pa_i)$ を満足し、 $pa_j \in m_2$ について $receive(pa_i) \rightarrow send(pa_j)$ 、 $c_2 \rightarrow send(pa_j)$ 、 $receive(pa_j) \rightarrow c_3$ を満足するとき、 pa_j は誘導孤児パケットである。

また、図 6(c) および (d) のようなメッセージの送受信を行なっているとき、

- p_2 における受信イベントが順次受信であり、 $\exists pa_i \in m_1$ について、 $send(pa_i) \rightarrow c_1$ かつ $sb(m_2) \rightarrow c_2 \rightarrow receive(pa_i)$ を満足し、 $pa_j \in m_2$ について $receive(pa_i) \rightarrow send(pa_j)$ 、 $c_2 \rightarrow send(pa_j)$ 、 $receive(pa_j) \rightarrow c_3$ を満足するとき、 pa_j は誘導孤児パケットである。

5 チェックポイントの一貫性評価

各メッセージの一貫性評価の統合によってグローバルチェックポイントの一貫性を評価するという分割統治型の手法は、分散型チェックポイントプロトコルの実現に対して有効である [2]。しかし、順次送受信イベントモデルの導入により、あるメッセージの一貫性低下が他のメッセージの一貫性低下の原因となることがある。この場合においても各メッセージの一貫性評価の統合としてグローバルチェックポイントの一貫性を評価するためには、誘導孤児パケットによるグローバル

チェックポイントの一貫性の低下をこれを含むメッセージの一貫性評価に反映することが必要である。論文 [2] では、メッセージ $m = \langle p_0, \dots, p_{l-1} \rangle$ の一貫性の評価 $MC(m)$ は、メッセージを構成するすべてのパケットの価値の総和に対するすべての孤児パケットおよびすべての紛失パケットの価値の総和の比 $value_ratio_{lost}$ 、 $value_ratio_{orphan}$ の関数として与えられる。ただし、以下の条件を満たすものとする。

- $value_ratio_{lost} = 0$ のとき $MC(m) = 1$
- $value_ratio_{lost} = 1$ のとき $MC(m) = 0$
- $MC(m)$ は $value_ratio_{lost}$ の単調減少関数
- $value_ratio_{orphan} \neq 1$ のとき $MC(m) = 1$
- $value_ratio_{orphan} = 1$ のとき $MC(m) = 0$

$value_ratio_{lost}$ と $value_ratio_{orphan}$ それぞれに対する $MC(m)$ を図 7 と図 8 に示す。

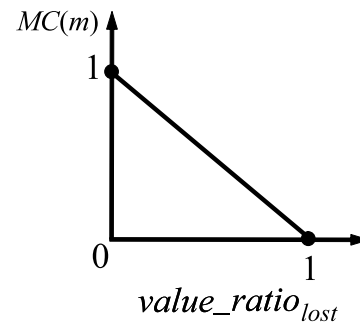


図 7: 紛失パケットによる一貫性の低下

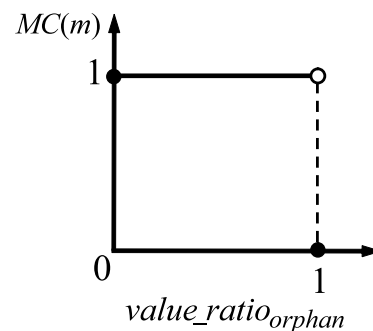


図 8: 孤児パケットによる一貫性の低下

誘導孤児パケットは孤児パケットでありながら、因果先行する紛失パケットが存在するために送信プロセスがリカバリ回復後に再送しないことや異なる内容のパケットを送信することがある。しかし、紛失パケットのように確実に再送されないものではないことから、同じパケットが誘導孤児パケットとなることによる一貫性の低下は紛失パケットとなることによる一貫性の

低下よりも小さい。したがって、以下の条件を満足することが求められる。ただし、メッセージを構成するすべてのパケットの価値の総和に対するすべての誘導孤児パケットの価値の総和の比を $value_ratio_{induced_orphan}$ とする。

- $value_ratio_{induced_orphan} = 0$ のとき $MC(m) = 1$
- $value_ratio_{induced_orphan} = 1$ のとき $MC(m) = 0$
- $MC(m)$ は $value_ratio_{induced_orphan}$ の単調減少関数
- $\frac{dMC(m)}{dvalue_ratio_{lost}} \leq \frac{dMC(m)}{dvalue_ratio_{induced_orphan}}$

これらを満足する時の $value_ratio_{induced_orphan}$ に対する $MC(m)$ を図 9 に示す。

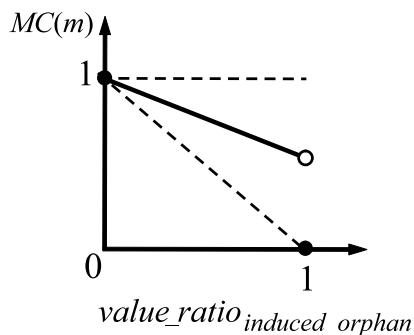


図 9: 誘導孤児パケットによる一貫性の低下

6 まとめと今後の課題

本論文では、順次送信イベント、順次受信イベント、一括送信イベント、一括受信イベントが混在するマルチメディアネットワークアプリケーションにおけるグローバルチェックポイントの一貫性評価手法を提案した。送信イベントと受信イベントが 1 つのプロセスで並行に発生する場合、孤児パケットの一部がリカバリ回復後に再送信されない誘導孤児パケットとなることがある。今後は、新しい評価法に基づくチェックポイントプロトコルを設計する。

参考文献

- [1] 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).
- [2] Osada, S. and Higaki, H., “Checkpoint Protocols in Multimedia Communication Networks,” Proceedings of the 2nd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 346–353 (2001).