

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

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

近年、コンピュータやネットワーク技術の急速な発達に伴ない、マルチメディア情報をネットワーク上で扱うマルチメディアネットワークシステムが普及している。また、ネットワーク上でのサービスにおいてシステムの耐故障性を高める手法である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。チェックポイントリカバリでは、システムに存在するプロセスが故障のないときの状態情報を保存し、チェックポイントを設定しておく。そして、システムに故障が発生した際には、すべてのプロセスが保存しておいた状態情報を読み込み、故障前の状態に復旧することにより耐故障性を保障している。しかし、システムが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満たしたものであることが必要となるため、この一貫性評価が重要である。ここで、これまで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、これらはシステム内のプロセスのすべてが、限られた通信モデルの場合であるときのみを想定して設計されていた。そのため、その他の通信モデルの混在した環境においては正しく一貫性を評価できないことがあるという問題がある。そのため、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、これまでの一貫性評価手法における問題点について述べ、それに対応した一貫性評価方法を提案する。

Global Consistency of Multimedia Checkpoints with Bulk and Stream Communication Events

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

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

コンピュータやネットワーク技術の発達によってストリーミング配信などの、音声や動画などのマルチメディアデータをネットワーク上で扱うマルチメディアネットワークシステムの開発が進められている。また、多数のコンピュータやルータ等の機器から構成されるネットワークでは、その一部が一時的に障害や保守等によって機能しなくなることが考えられる。しかし、ミッションクリティカルアプリケーションは、このような状況においてもサービスを正しく提供することが必要である。障害等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がある。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に動作してい

る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。このとき、リカバリ後のシステム動作に矛盾を発生させないためには、各プロセスが設定したチェックポイントの集合で定まるグローバル状態が論文 [1] で定められた一貫性を保つものであることが求められる。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させないことに注目し、論文 [2] では、リカバリ時に失われるパケット数に基づいた新しい一貫性の評価方法が提案されている。ただし、すべてのプロセスにおける通信イベントは、一括送信イベントおよび一括受信

イベントのみであることを前提としている。一括送信イベントでは、マルチメディア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決定される。また、一括受信イベントでは、マルチメディアメッセージを構成するすべてのパケットを受信してはじめて以降に送信されるメッセージに影響を与える。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以降に送信されるメッセージあるいはパケットの決定に影響を与え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

音声や動画などのマルチメディア情報をあつか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 (pa_0, \dots, pa_{l-1}) に分割され、送信される。ここで、プロセス p_i から p_j への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を定義する。

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

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 (pa_0, \dots, pa_{l-1}) 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。

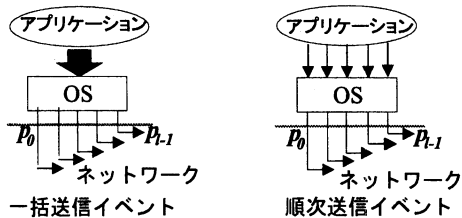


図1: マルチメディア送信イベント

[-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 (pa_0, \dots, pa_{l-1}) 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。

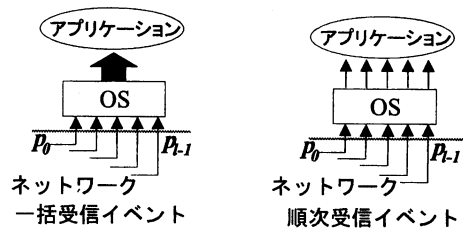


図2: マルチメディア受信イベント

[-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]

順次送信イベントでは、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。

ここで、これまでのマルチメディアシステムのための一貫性評価においてはすべてのプロセスの通信モデルが、一括送信モデル、一括受信モデルである場合のみを想定していた。

一方、例として、あるプロセスにおいて同時に送受信を行っており、そのプロセスにおける通信イベントが順次受信、順次送信であったとする。この場合、受信するメッセージ中のパケットを受信した以降に送信されるメッセージの内容に影響を及ぼしているという可能性がある。

このように、通信モデルにより、メッセージの内容が決定時期が異なる。

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

3.1 チェックポイント

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取得とリカバリによって成り立つ。

まず、アプリケーションを実行するプロセスが正常動作時のアプリケーションの変数やレジスタの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取得する。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によってリカバリする。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ぬ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。

3.2 データ通信における一貫性

これまでのデータ通信を対象とした一貫性評価においては、一貫性のある状態を、チェックポイント中に紛失メッセージ、孤児メッセージが存在しない状態とし、また、一貫性の無い状態を、紛失メッセージ、孤児メッセージが存在する状態としていた。そのため、チェックポイントの一貫性は以下のように評価されていた。

$$G_c = \begin{cases} 1 & \text{紛失、孤児メッセージが存在しない} \\ 0 & \text{それ以外} \end{cases}$$

また、紛失メッセージと孤児メッセージの例を図3に示す。

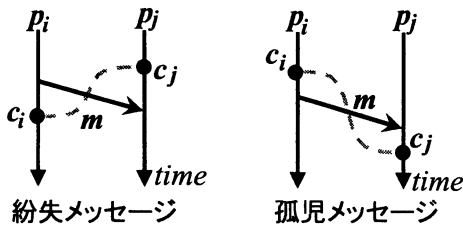


図3: 紛失メッセージと孤児メッセージ

ここで、メッセージ m の送信イベントを $send(m)$ とし、メッセージ m の受信イベントを $receive(m)$ とする。

[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。

□

この状態で、システムがチェックポイント c_i, c_j に復帰すると、 m は送信側では送信が完了している状態にもかかわらず受信側では受信されない。また、送信側ではすでに送信が完了している状態であるので、その後再送信されることはないため、一貫性のないメッセージである。

[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。

□

この状態で、システムがチェックポイント c_i, c_j に復帰すると、 m は送信側で再送信されるという決定性の要素が無いのに対して、受信側では m を受け取った状態で処理を再開させることとなる。このように、送信側がチェックポイントに復帰後に同じメッセージが再送信されるという保証がないため、一貫性のないメッセージである。

また、データ通信を対象としたチェックポイントリカバリの手法ではテキストベースのメッセージをやりとりするシステムを想定して設計されていたため、メッセージ通信イベント中はチェックポイントを取得しないと定められていた。

ここで、メッセージサイズが大きく、送受信時間が長いマルチメディアネットワークシステムにおいて、データ通信を対象としたこれまでのチェックポイントリカバリの手法を適用すると、メッセージの送受信が終了するまでチェックポイントが取得できないことからオーバーヘッドが大きくなってしまふ。このことから、論文 [2] ではメッセージ通信イベント中にもチェックポイントを取得できるようにすることができるマルチメディアチェックポイント手法と、マルチメディアシステムのための一貫性の評価が提案されている。

3.3 マルチメディア通信における一貫性

論文 [2] で提案されているマルチメディアシステムのための一貫性の評価では、グローバルチェックポイントの一貫性を G_c (Global consistency) として、システムに存在するすべてのチェックポイントより定めている。一方、チェックポイントはシステムに存在する各チャネルの両端で取得されることからチャネルの一貫性を C_c (Channel consistency) として定めている。ここで、各チャネルの一貫性はそのチャネルを通るメッセージの一貫性より定めている。以上のことよりシステムの一貫性を定めるためにはメッセージの一貫性 M_c (Message consistency) を定めることで、チャネルの一貫性を定め、チャネルの一貫性よりグローバルチェックポイントの一貫性を定めるとしている。

3.3.1 一貫性のないマルチメディアメッセージ

マルチメディアメッセージ m が図4に表すような紛失マルチメディアメッセージであったり、図5に表すような孤児マルチメディアメッセージであった場合、それらは一貫性のないマルチメディアメッセージであるという。紛失マルチメディアメッセージと孤児マルチメディアメッセージの定義を以下に示す。

[紛失マルチメディアメッセージ]

図4のように、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 へ送信されたマルチメディアメッセージ m が、 $se(m) \rightarrow c_i$

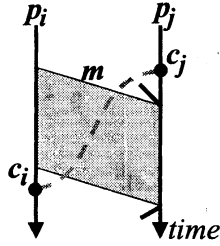


図 4: 紛失マルチメディアメッセージ

かつ $c_j \rightarrow rb(m)$ を満たすとき紛失マルチメディアメッセージである。□

この状態で、システムがチェックポイント c_i, c_j に復帰すると、 m は送信側では送信が完了している状態にもかかわらず受信側では受信されない。また、送信側ではすでに送信が完了している状態であるため、その後再送信されることはない。そのため、一貫性のないマルチメディアメッセージである。

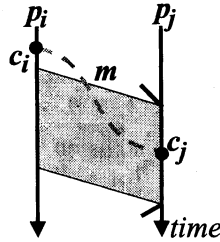


図 5: 孤児マルチメディアメッセージ

[孤児マルチメディアメッセージ]

図5のように、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 へ送信されたマルチメディアメッセージ m が、 $c_i \rightarrow sb(m)$ かつ $rb(m) \rightarrow c_j$ を満たすとき m を孤児マルチメディアメッセージである。□

この状態で、システムがチェックポイント c_i, c_j に復帰すると、 m が送信側で再送信されるという決定性の要素が無いのに対し、受信側では m の全体、もしくは一部を受け取った状態で処理を再開させることになる。このように、送信側がチェックポイントに復帰後に必ずしも同じメッセージを再送信される保証がないため、一貫性のないマルチメディアメッセージである。

これら一貫性のないマルチメディアメッセージを含むチェックポイントは一貫性がない。そのため、一貫性の評価値を0として評価する。

3.3.2 紛失パケットと孤児パケット

マルチメディアメッセージはテキストベースのメッセージに比べ、メッセージサイズが大きいという性質と、チェックポイントに復帰したときに、メッセージの一部を紛失したとしてもアプリケーションの処理が必ずしも正しく行われぬということがないという性質を持っている。このことより、論文 [2] で提案されているチェックポイントプロトコルにおいては、メッセージと

メッセージの間でチェックポイントを取得するだけではなくメッセージの送受信中にチェックポイントを取得するとしている。しかし、パケットを送信中にもチェックポイントを取得することによりデータ通信を対象としたチェックポイントプロトコルにおいては存在しなかった紛失パケットや、孤児パケットが発生する。紛失パケット、孤児パケットの定義を以下に示す。

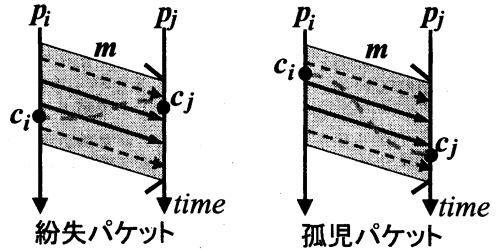


図 6: 紛失パケット、孤児パケット

[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 $(pa_1, pa_2, \dots, pa_{l-1})$ で構成されているとする。このとき、パケット pa_k を構成要素に含むとする。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。□

図6左図に示すような紛失パケットが存在した状態でチェックポイントに復帰した場合、このパケットはすでに送信側では送信が完了している状態のため、復帰した後に再送信されることはない。よって、紛失パケットは存在すればするほど一貫性を低下させる。

[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 $(pa_1, pa_2, \dots, pa_{l-1})$ で構成されているとする。このとき、パケット pa_k を構成要素に含むとする。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。□

図6右図に示すような孤児パケットが存在した状態でチェックポイントに復帰した場合、送信側ではメッセージの送信自体は既に開始されているため、孤児パケットとなっている部分のパケットは再送信される保証がある。このことより孤児パケットは一貫性があるパケットとされる。そのため、一貫性を低下させる要因とはならない。

これらのことから、マルチメディア通信のための一貫性評価においては、紛失パケットのみに注目し、一貫性を評価することとしていた。

マルチメディア通信におけるグローバルチェックポイントの一貫性 Gc は、紛失パケットに注目し $[0, 1]$ の範囲で評価していた。

3.3.3 メッセージの一貫性

[メッセージの一貫性]

メッセージの一貫性は、紛失パケットによって、メッセージ全体からどの程度の価値を失ったのかにより表現されている。メッセージ内の全パケット数に対する紛失パケット数の割合から算出していた。ここで、メッセージ

m 、メッセージを構成するパケットを pa_0, \dots, pa_{l-1} とする。このとき、 $\{pa_i, \dots, pa_j\} (0 \leq i < j < l)$ が紛失パケットであるとき、メッセージの一貫性を以下のようにして求めている。

$$Mc(m) = 1 - \frac{\sum_{k=i}^j \text{value}(pa_k)}{\text{value}(m)}$$

このことからメッセージの一貫性 $Mc(m)$ は次のような値をとる。

$$\begin{cases} Mc(m) = 0 & (\sum_{k=i}^j \text{value}(pa_k) = \text{value}(m)) \\ Mc(m) = 1 & (\sum_{k=i}^j \text{value}(pa_k) = 0) \\ 0 < Mc(m) < 1 & (0 < \sum_{k=i}^j \text{value}(pa_k) < \text{value}(m)) \end{cases}$$

ただし、 $\text{value}(m)$ はメッセージ m の価値、 $\text{value}(pa_k)$ は紛失パケット pa_k の価値である。

3.3.4 チャンネルの一貫性

システムに複数のプロセスが存在する場合、各プロセス間にはメッセージを送受信するためのチャンネルが存在する。また、チェックポイントはこのチャンネルの両端で取得されることからチャンネルの一貫性 (Cc) はそのチャンネルの両端で取得されたチェックポイントとチャンネルを通るメッセージの一貫性より定められる。ここで、そのチャンネルを通るメッセージのうちで一つでも一貫性が 0 のものが存在する場合チャンネルの一貫性も 0 とならなければならない。逆にチャンネルを通るすべてのメッセージの一貫性が 1 の場合も、チャンネルの一貫性は 1 とならなければならない。このことより、チャンネルの一貫性は以下のように定められていた。

[チャンネルの一貫性]

プロセス p_i, p_j がそれぞれチェックポイント c_i, c_j を取得したときに送受信されたメッセージを M_{ij} としたとき、このときのチャンネルの一貫性は以下のように定められていた。

$$Cc(p_i, p_j, c_i, c_j) = \prod_{M_{ij}} Mc(p_i, p_j)$$

3.3.5 グローバルチェックポイントの一貫性

システムに存在するプロセスはチャンネルを介してメッセージを送受信する。このとき、グローバルチェックポイントの一貫性 (Gc) はシステムに存在するすべてのチャンネルの一貫性より定められる。ここで、システムに存在するチャンネルのうちで一つでも一貫性が 0 のチャンネルが存在する場合、グローバルチェックポイントの一貫性も 0 とならなければならない。また、システムに存在するすべてのチャンネルの一貫性が 1 の場合もグローバルチェックポイントの一貫性が 1 とならなければならない。よって、グローバルチェックポイントの一貫性の値は以下のように定められていた。

[グローバルチェックポイントの一貫性]

システムに存在するプロセスの集合を \mathcal{P} 、各プロセスで取得されたチェックポイントの集合を $C_{\mathcal{P}}$ 、各プロセス

間に存在するチャンネルの集合を \mathcal{L} としたときグローバルチェックポイントの一貫性 $Gc(C_{\mathcal{P}})$ は以下のように表わされる。

$$Gc(C_{\mathcal{P}}) = \left(\prod_{\mathcal{L}} Cc(p_i, p_j, c_i, c_j) \right)^{1/|\mathcal{L}|}$$

グローバルチェックポイントの一貫性はチャンネルの一貫性のときとは異なり、チャンネルの一貫性をかけあわせたものを $1/(\text{チャンネルの数})$ することで定めている。これはこのようにすることによって、グローバルチェックポイントの一貫性の値がシステムの規模に依存しないようにするためである。

4 マルチメディアシステムのための一貫性評価の問題点

これまでのマルチメディアシステムのための一貫性評価においては、先に述べた一括送信イベント、一括受信イベントという限られた通信モデルのみを想定していたことにより、次のような性質が成り立つ。

[メッセージの一貫性低下の独立性]

あるメッセージにおいて発生した孤児パケット、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、各プロセスにおいて、送受信したメッセージの一貫性を全パケット数に対する紛失パケット数の比率から算出し、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、順次受信イベントも含む混在環境においては、この性質が必ずしも成り立たない。例えば、図 7 に示す場合を考える。ここで、 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。

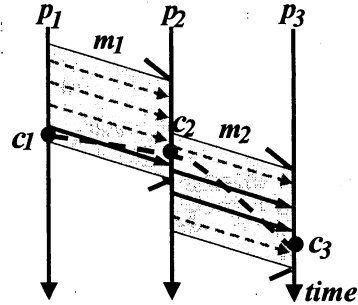


図 7: 誘導孤児パケット

図 7 を見ると、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 $opa_i \in m_2$ が発生している。このとき、 p_2 において、 m_1 が順次受信、 m_2 が順次送信されることから、 $pa_{k_1} \in m_1$ 、 $pa_{k_2} \in m_2$ について $receive(pa_{k_1}) \rightarrow send(pa_{k_2})$ が成り立つとき、 pa_{k_2} は pa_{k_1} の影響を受けていることがある。このとき、 lp_{a_i} はリカバリ回復後に再送信されないパケットであることから、 c_2 において $receive(lp_{a_i}) \rightarrow send(pa_{k_2})$ を満たすパケット pa_{k_2} はリカバリ回復後に再送信されないことがある。そのため、 $receive(lp_{a_i}) \rightarrow send(opa_i)$ となる opa_i は m_2 の一貫性を低下させる。

このようなパケットは、[2]において想定されていた通信モデルの環境においては発生しないため、一貫性の評価において考慮されていなかった。そのため、このようなパケットが存在した場合、この一貫性評価方法では一貫性を正しく評価することができない。以降では、このパケットを誘導孤児パケットとよぶ。

5 誘導孤児パケット発生条件

[送受信パターンの条件]

誘導孤児パケットが発生するメッセージの送受信パターンや通信モデルは以下のような場合である。

- 1) p_1 は m_1 を一括送信、 p_2 は m_2 を順次送信、 p_3 は m_2 を一括受信する。
- 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。このとき、 m_2 の受信イベントは、一括受信、もしくは順次受信である。(図 8(a) および図 8(b))
 - p_2 において $rb(m_1) \rightarrow se(m_2) \rightarrow re(m_1)$ が成り立つ。このとき、 m_2 の受信イベントは、順次受信である。(図 8(c) および図 8(d))

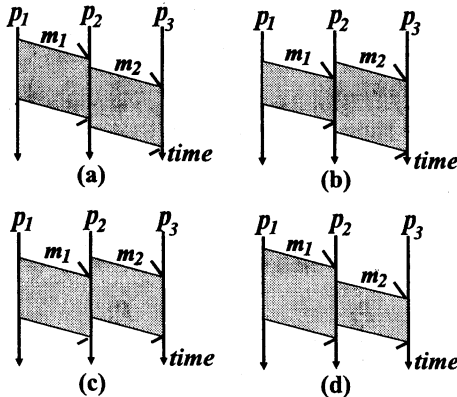


図 8: 誘導孤児パケットの発生条件

[チェックポイントの設定条件]

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 8(a), (b) のようなメッセージの送受信を行っているとき、

- p_2 における受信イベントが一括受信であった場合、 $pa_i \in m_1$ であり、 $send(pa_i) \rightarrow c_1$ かつ $sb(m_2) \rightarrow c_2 \rightarrow receive(pa_i)$ を満たすようにチェックポイントが設定されたとする。また、 $re(m_1) \rightarrow send(pa_j)$ かつ $pa_j \in m_2$ であり、 $c_2 \rightarrow send(pa_j)$ かつ $receive(pa_j) \rightarrow c_3$ を満たすようにチェックポイントが設定された場合、 pa_j が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った場合、 $pa_i \in m_1$ であり、 $send(pa_i) \rightarrow c_1$ かつ $sb(m_2) \rightarrow c_2 \rightarrow receive(pa_i)$ を満たすようにチェックポイントが設定されたとする。また、 $receive(pa_i) \rightarrow send(pa_j)$ かつ $pa_j \in m_2$ であり、 $c_2 \rightarrow send(pa_j)$ かつ $receive(pa_j) \rightarrow c_3$ を満たすようにチェックポイントが設定された場合、 pa_j が誘導孤児パケットである。

また、図 8(c), (d) のようなメッセージの送受信を行っているとき、

- p_2 における受信イベントが順次受信の場合のみであるため、 $pa_i \in m_1$ であり、 $send(pa_i) \rightarrow c_1$ かつ $sb(m_2) \rightarrow c_2 \rightarrow receive(pa_i)$ を満たすようにチェックポイントが設定されたとする。また、 $receive(pa_i) \rightarrow send(pa_j)$ かつ $pa_j \in m_2$ であり、 $c_2 \rightarrow send(pa_j)$ かつ $receive(pa_j) \rightarrow c_3$ を満たすようにチェックポイントが設定された場合、 pa_j が誘導孤児パケットである。

プロセス数、チャネル数が増えた場合、送受信を同時におこなっているプロセスの通信モデルに注目し、先に述べたような、誘導孤児パケットの発生する通信モデルであり、また先に述べたようなチェックポイントの設定条件が成り立つようにチェックポイントが設定されていた場合に誘導孤児パケットが発生している。

また、複数のメッセージ間に影響が及んでいる場合においては、上流のメッセージにおいて発生した紛失パケットの影響を受けているパケットが孤児パケットとなるようにチェックポイントが設定された場合、その孤児パケットは誘導孤児パケットである。

6 提案手法

誘導孤児パケットを含むメッセージの一貫性は、本論文の通信モデルにおいては、その数の増加に対して単調減少させ、評価するべきである。すなわち、メッセージ $m = (pa_0, \dots, pa_{l-1})$ において $\{pa_i, \dots, pa_j\} (0 \leq i < j < l)$ が誘導孤児パケットであるとき、 m の一貫性 $Mc(m)$ は次式で与えられることが考えられる。

$$Mc(m) = 1 - \frac{\sum_{k=i}^j \text{value}(pa_k)}{\text{value}(m)}$$

ただし、 $\text{value}(m)$ はメッセージ m の価値、 $\text{value}(pa_k)$ はパケット pa_k の価値である。

7 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におけるメッセージの一貫性の新しい評価手法を提案した。ここでは、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を示し、これによる一貫性低下を考慮する手法を示した。今後は、この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).