マルチメディア通信と分散処理 102－27
（２００１．３．２２）

QoS-based Checkpoint Protocol in Multimedia Network Systems

Shinji Osada, Yusuke Nakamura, Kengo Hiraga and Hiroaki Higaki
{shinji, yusuke, hira, hig}@higlab.k.dendai.ac.jp
Department of Computers and Systems Engineering
Tokyo Denki University

In order to achieve fault-tolerant distributed systems, checkpoint-recovery has been researched and many protocols have been designed. A global checkpoint taken by the protocols have to be consistent. For conventional data communication networks, a global checkpoint is defined to be consistent if there is no inconsistent message in any communication channel. For multimedia communication networks, there are different requirements for time-constrained failure-free execution and large-size message transmissions where lost of part of the message is acceptable. This paper proposes a new criteria for consistent global checkpoints in multimedia communication networks. In addition, a QoS-based checkpoint protocol is designed according to the criteria. This protocol is non-blocking and time-bounded for supporting realtime multimedia message transmissions. Finally, the protocol is evaluated in an MPEG-2 data transmission.

マルチメディアネットワークシステムにおけるQoSに基づいたチェックポイントプロトコル

長田 慎司　中村 祐介　平賀 研吾　桧垣 博章
東京電機大学 理工学部 情報システム工学科

分散システムにおいてフォールトトレラントを達成するために、様々なチェックポイントプロトコルが研究、開発されてきた。それらのプロトコルを用いて取得されたグローバルチェックポイントでは、一貫性を保つことが必要とされる。従来のネットワーク通信では、グローバルチェックポイントの一貫性は、一貫性のないメッセージが存在しない状態と定義されている。ところが、マルチメディアネットワークでは時間制約のあるアプリケーションを実行するために、たとえメッセージの一部が紛失したとしても、オーバヘッドの小さな手法を導入することが求められている。本論文では、マルチメディアネットワークの特性を考慮して、グローバルチェックポイントの一貫性に新たな基準を設けた。また、これに基づいたチェックポイントプロトコルを設計した。本論文で提案されたプロトコルは、マルチメディアメッセージのリアルタイム配送をサポートすることができる。最後に、MPEG-2データを送受信することによって提案プロトコルの有効性を評価した。

1 Introduction

Advanced computer and network technologies have lead to the development of distributed systems. Here, an application is realized by multiple processes located on multiple computers connected to a communication network such as the Internet. Each process computes and communicates with other processes by exchanging messages through a communication channel. Mission-critical applications are required to be executed fault-tolerantly. That is, even if some processes fail, execution of an application is required to be continued. One of the important methods to realize fault-tolerant distributed systems is checkpoint-recovery [3, 8]. During failure-free execution, each process takes local checkpoints by storing state information into a stable storage.If a certain process fails, the processes restart from the checkpoints by restoring the state information from the stable storage. For restarting the system correctly, a set of local checkpoints taken by all the processes and from which the processes restart should form a *consistent global checkpoint* [1]. A consistent global checkpoint is defined as that there is neither *orphan* nor *lost message*. However, it is not suitable to multimedia communication networks. In this paper, we propose a novel criteria for consistent global checkpoints based on properties of multimedia communication networks and design a non-blocking checkpoint protocol.

The rest of this paper is organized as follows: In section 2, we review consistent global checkpoints in a conventional data communication network. In section 3, we discuss properties of a multimedia communication network and requirements for a consistent global checkpoint. Section 4 proposes a novel criteria for a consistent global checkpoint in a multimedia communication network. In section 5, we design a checkpoint protocol based on QoS (Quality of Service) for consistency and timeliness. Finally in section 6, for an evaluation, the criteria of consistency and the checkpoint protocol are applied to an MPEG-2 data transmission [4]. The result shows that they work well in multimedia communication networks.

2 Conventional Consistency

Let $\mathcal{S} = \langle \mathcal{V}, \mathcal{L} \rangle$ be a distributed system where $\mathcal{V} = \{p_1, \dots, p_n\}$ is a set of processes p_i and $\mathcal{L} \subseteq \mathcal{V}^2$ is a set of communication channels $\langle p_i, p_j \rangle$ from a process p_i to another process p_j. Execution of an application in p_i is modeled by occurrence of a sequence of *events*. A *state* of p_i is changed at each event. There are two kinds of events: *local events* and *communication events*. At a local event, a state of p_i is changed by local computation without exchanging a message. At a communication event, p_i communicates with another process by exchanging a message and the state is changed. There are two kinds of communication events: a *message sending event* $s(m)$ and a *message receipt event* $r(m)$ for a message m.

In order to realize a fault-tolerant distributed system, checkpoint-recovery is widely available [3, 6, 8]. Here, during failure-free execution, each process p_i sometimes takes a *local checkpoint* c_i by storing state information into a *stable storage*. If p_i fails, p_i restarts from c_i by restoring the state information from the stable storage. If p_i restarts independently of the other processes, there may be two kinds of *inconsistent messages*: *lost messages* and *orphan messages* [1].

[**Inconsistent message**] Suppose that processes p_i and p_j take local checkpoints c_i and c_j, respectively. A message m transmitted through a communication channel $\langle p_i, p_j \rangle$ is *inconsistent* if m is a lost message or an orphan message for a set $C_{\{p_i,p_j\}} = \{c_i, c_j\}$ of local checkpoints. m is a *lost message* iff $s(m)$ occurs before taking c_i in p_i and $r(m)$ occurs after taking c_j in p_j. m is an *orphan message* iff $s(m)$ occurs after taking c_i in p_i and $r(m)$ occurs before taking c_j in p_j. □

In order to achieve correct recovery from a failure, there should be neither lost nor orphan message in any communication channel in $\mathcal{L}$. Thus, if a process p_i fails, not only p_i but also other processes are required to restart from local checkpoints. Hence, a *global checkpoint* $C_{\mathcal{V}} = \{c_1, \dots, c_n\}$ which is a set of local checkpoints of all the processes in $\mathcal{V}$ should be *consistent* [1].

[**Consistent global checkpoint**] A global checkpoint $C_{\mathcal{V}}$ in $\mathcal{S}$ is consistent iff there is no inconsistent message in any communication channel in $\mathcal{L}$. □

3 Multimedia Networks

Recently, distributed multimedia applications such as distance learning, tele-conference, telemedicine and video on demand are developed on communication networks [7]. Here, messages with multimedia data including text, voice, sound, picture and video are exchanged among processes for an application. These messages are so large that they are compressed before being sent and uncompressed (expanded) after being received. However, the messages are still larger than messages with conventional data. Hence, it takes longer time to transmit and receive the messages. As in Figure 1, the following four *primitive events* are defined for a multimedia message m transmitted from a process p_i to another process p_j [9]:

- $sb(m)$: p_i begins transmitting m.
- $se(m)$: p_i ends transmitting m.
- $rb(m)$: p_j begins receiving m.
- $re(m)$: p_j ends receiving m.

A message sending event $s(m)$ for m begins at $sb(m)$ and ends at $se(m)$ in p_i. A message receipt event $r(m)$ for m begins at $rb(m)$ and ends at $re(m)$ in p_j.

In a computer communication network, protocols are hierarchically composed, e.g. TCP/IP protocols [2]. A message in an upper layer is decomposed into multiple packets in a lower layer. For example, an IP datagram may be decomposed into multiple Ethernet frames in a sender process since an MTU (Maximum Transfer Unit) for an IP datagram is 64kbyte and one for an Ethernet frame is 1.5kbyte. These frames are reassembled in a receiver process. Thus, a multimedia message m is assumed to be decomposed into a sequence $\langle pa_1, \dots, pa_l \rangle$ of multiple *packets* for transmission as in Figure 1. Here, $s(pa_i)$ is a *packet sending event* and $r(pa_i)$ is a *packet receipt event* for a packet pa_i.

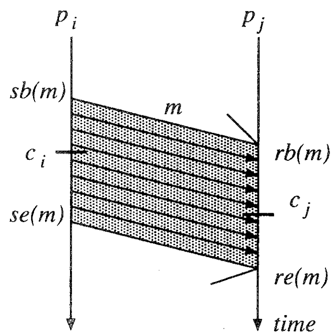

Figure 1: Multimedia message transmission.

In a distributed system exchanging a message m with conventional data, communication events $s(m)$ and $r(m)$ are assumed to be atomic. Here, each local checkpoint c_i of a process p_i is assumed to be taken only when no event occurs in p_i. However, since a multimedia message is much larger than a conventional data message, it takes longer time to transmit and receive the message. Thus, if a process is required to take a local checkpoint during a communication event, it has to wait until an end of the event. Hence, timeliness requirement in a checkpoint protocol is not satisfied and communication overhead for recovery is increased. Therefore, a local checkpoint should be taken immediately when a process is required to take it even during a communication event in order to reduce synchronization and communication overhead. That is, a process p_i sending a message $m = \langle pa_1, \dots, pa_l \rangle$ takes a local checkpoint c_i between $s(pa_s)$ and $s(pa_{s+1})$ and another process p_j receiving m takes a local checkpoint c_j between $r(pa_r)$ and $r(pa_{r+1})(1 \leq s, r < l)$. In addition, a part of a multimedia message can be lost in a communication channel for an application, e.g. MPEG-2 data transmission. Such an application requires not to retransmit lost messages but to transmit messages with shorter transmission delay and smaller jitter.

4 Novel Consistency

As discussed in section 2, since the conventional criteria of consistency for a global checkpoint is based on an architecture of conventional data communication networks, a novel criteria should be introduced into multimedia communication networks.

Global consistency $Gc(C_{\mathcal{V}})$ for a global checkpoint $C_{\mathcal{V}} = \{c_1, \dots, c_n\}$ denotes a degree of consistency for $C_{\mathcal{V}}$ in $\mathcal{S}$. In a conventional data communication network, $Gc(C_{\mathcal{V}})$ is defined as follows:

$$Gc(C_{\mathcal{V}}) = \begin{cases} 1 & \text{no inconsistent message for } C_{\mathcal{V}}. \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

In a multimedia communication network, a local checkpoint is taken even during a communication event and it is acceptable for an application to lose a part of a multimedia message. Hence, the domain of $Gc(C_{\mathcal{V}})$ is a closed interval $[0, 1]$ instead of a discrete set $\{0, 1\}$. Here, $Gc(C_{\mathcal{V}})$ should be compatible with the conventional criteria, i.e. (1) should be satisfied.

Consistency of a global checkpoint is determined by timing-relation between local checkpoints and messages transmitted through communication channels. Thus, $Gc(C_{\mathcal{V}})$ is calculated by *channel consistency* $Cc(\langle p_i, p_j \rangle, c_i, c_j)$ for all the communication channels $\langle p_i, p_j \rangle \in \mathcal{L}$. Furthermore, channel consistency for a communication channel $\langle p_i, p_j \rangle$ is calculated by *message consistency* $Mc(m, c_i, c_j)$ for all the messages m transmitted through $\langle p_i, p_j \rangle$. Finally, message consistency for a message m transmitted through $\langle p_i, p_j \rangle$ is induced by timing-relation between m and $C_{\{p_i, p_j\}} = \{c_i, c_j\}$.

4.1 Message Consistency

Message consistency $Mc(m, c_i, c_j)$ is a degree of consistency for a set $C_{\{p_i, p_j\}} = \{c_i, c_j\}$ of local checkpoints and a multimedia message m transmitted through a communication channel $\langle p_i, p_j \rangle$. Here, c_i and c_j are taken by processes p_i and p_j, respectively.

Now, we define an inconsistent multimedia message.

[Inconsistent multimedia message] A multimedia message m is inconsistent iff m is a lost message or an orphan message. m is a lost message iff a primitive event $se(m)$ occurs before taking c_i in p_i and another primitive event $rb(m)$ occurs after taking c_j in p_j. m is an orphan message iff $sb(m)$ occurs after taking c_i in p_i and $rb(m)$ occurs before taking c_j in p_j. □

If m is a lost message, all the packets of m are sent by p_i but none of them is received by p_j. If m is an orphan message, m might not be retransmitted after recovery due to non-deterministic properties of p_i even though p_j has already received a part of m.

[Consistency for inconsistent message] If a multimedia message m is inconsistent, $Mc(m, c_i, c_j) = 0$. □

Suppose a process p_i takes a local checkpoint c_i while p_i is transmitting a multimedia message m and/or another process p_j takes a local checkpoint c_j while p_j is receiving m as in Figure 1. As discussed in the previous section, m is decomposed into a sequence of multiple packets $\langle pa_1, \dots, pa_l \rangle$. Thus, $s(m)$ is composed of a sequence $\langle s(pa_1), \dots, s(pa_l) \rangle$ of packet sending events and $r(m)$ is composed of a sequence $\langle r(pa_1), \dots, r(pa_l) \rangle$ of packet receipt events. Suppose that local checkpoints c_i and c_j are taken between $s(pa_s)$ and $s(pa_{s+1})(1 \leq s < l)$ and between $r(pa_r)$ and $r(pa_{r+1})(1 \leq r < l)$, respectively. A *lost packet* and an *orphan packet* are also defined same as a lost message and an orphan message.

[Lost and orphan packets] pa_k is a lost packet iff $s(pa_k)$ occurs before taking c_i in p_i and $r(pa_k)$ occurs after taking c_j in p_j. pa_k is an orphan packet iff $s(pa_k)$ occurs after taking c_i in p_i and $r(pa_k)$ occurs before taking c_j in p_j.□

Clearly, if c_i takes p_i between $s(pa_k)$ and $s(pa_{k+1})$ and p_j takes c_j between $r(pa_k)$ and $r(pa_{k+1})(1 \leq k < l)$, a set $C_{\{p_i, p_j\}} = \{c_i, c_j\}$ of local checkpoints is consistent.

Suppose that p_i takes c_i between $s(pa_s)$ and $s(pa_{s+1})$ and p_j takes c_j between $r(pa_r)$ and $r(pa_{r+1})$ where $1 \leq s, r < l$.

If $s > r$, $\{pa_{r+1}, \dots, pa_s\}$ is a set of lost packets. These packets are not retransmitted after p_i and p_j restart from c_i and c_j, respectively. In some conventional checkpoint protocols applied in data communication networks, lost messages are stored in a stable storage with the state information at a local checkpoint and restored in recovery [5]. However, a checkpoint protocol in a multimedia communication network is required to be achieved with less overhead in failure-free execution since many applications require time-constrained execution. On the other hand, even if a part of a message is lost in recovery, an application accepts the message. The less packets of the message are lost, the higher message consistency we achieve. As discussed before, a multimedia message is usually compressed for transmission. Thus, value of packets for a message is not unique. For example in an MPEG-2 data transmission, value of a packet for an I-picture is higher than value of a packet for a B-picture. Therefore, lost consistency for a set $\{c_i, c_j\}$ of local checkpoints is defined as a ratio of value of the lost packets $\{pa_{r+1}, \ldots, pa_s\}$ in a message m to value of m. Hence, the message consistency for m is defined as follows:

[Message consistency] Let $value(pa_k)$ and $value(m)$ be value of a packet pa_k and a message m, respectively.

$$Mc(m, c_i, c_j) = Mc(m, s, r) = 1 - \frac{\sum_{k=r+1}^{s} value(pa_k)}{value(m)} \quad (2)$$

□

Here, the domain of $Mc(m, c_i, c_j)$ is an open interval $(0, 1)$.

If $s < r$, $\{pa_{s+1}, \ldots, pa_r\}$ is a set of orphan packets. An orphan message might not be retransmitted after recovery due to non-deterministic properties of a process. However, these packets are surely retransmitted after recovery since c_i and c_j are taken during transmission and receipt of m and the content of m being carried by a sequence $\langle pa_1, \ldots, pa_l \rangle$ of packets is not changed even after recovery. Hence, a set $\{c_i, c_j\}$ of local checkpoints is consistent, i.e. $Mc(m, c_i, c_j) = 1$.

Message consistency $Mc(m, s, r)$ for s and r is shown in Figure 3. l is a number of packets consisting of a message m, i.e. $m = \langle pa_1 \cdots pa_l \rangle$. Here, $s \leq 0$ ($r \leq 0$) means that p_i (p_j) takes a local checkpoint c_i (c_j) before $sb(m)$ ($rb(m)$) and $s > l$ ($r > l$) means that p_i (p_j) takes a local checkpoint c_i (c_j) after $se(m)$ ($re(m)$).

- $Mc(m, s, r) = 1$ if $s \leq 0$ and $r \leq 0$.
- $Mc(m, s, r) = 0$ if $s \leq 0$ and $r > 0$ since m is an orphan message.
- $Mc(m, s, r) = f_2(s)$ where $df_2(s)/ds \leq 0$, $\lim_{s \to 0} f_2(s) = 1$ and $\lim_{s \to l} f_2(s) = 0$ if $0 < s < l$ and $r \leq 0$.
- $Mc(m, s, r) = f_1(s, r)$ where $f_1(u, u) = 1 (0 < u < l)$, $df_1(s, r)/ds \leq 0$ and $df_1(s, r)/dr \geq 0$. $\lim_{s \to l} f_1(s, r) = f_3(r)$ and $\lim_{r \to 0} f_1(s, r) = f_2(s)$ if $0 < s < l$ and $0 < r < s$.
- $Mc(m, s, r) = 1$ if $0 < s < l$ and $s \leq r$.
- $Mc(m, s, r) = 0$ if $l \leq s$ and $r \leq 0$.

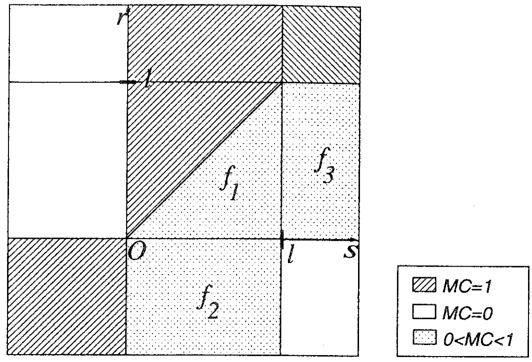

Figure 2: Message consistency.

- $Mc(m, s, r) = f_3(r)$ where $df_3(r)/dr \geq 0$, $\lim_{r \to 0} f_3(r) = 0$ and $\lim_{r \to l} f_3(r) = 1$ if $l < s$ and $0 < r < l$.
- $Mc(m, s, r) = 1$ if $l \leq s$ and $l \leq r$.

4.2 Channel Consistency

Based on the message consistency for a multimedia message m and local checkpoints c_i and c_j in processes p_i and p_j respectively, channel consistency $Cc(\langle p_i, p_j \rangle, c_i, c_j)$ is defined as a degree of consistency for a set $\{c_i, c_j\}$ of local checkpoints and a communication channel $\langle p_i, p_j \rangle \in \mathcal{L}$. For compatibility, if message consistency for every message is 1, channel consistency is also 1. In addition, if consistency for at least one message is 0, channel consistency is also 0. Thus, channel consistency for $\langle p_i, p_j \rangle$ is induced by multiplication of message consistency for all the messages transmitted through $\langle p_i, p_j \rangle$.

[Channel consistency] Let $\mathcal{M}_{ij}$ be a set of messages transmitted through a communication channel $\langle p_i, p_j \rangle$.

$$Cc(\langle p_i, p_j \rangle, c_i, c_j) = \begin{cases} 1 & \text{if } \mathcal{M}_{ij} = \phi. \\ \prod_{m \in \mathcal{M}_{ij}} Mc(m, c_i, c_j) & \text{otherwise.} \end{cases} \quad (3)$$

□

4.3 Global Consistency

Global consistency is determined according to timing-relations of all the sets of two checkpoints c_i and c_j where there is a communication channel $\langle p_i, p_j \rangle \in \mathcal{L}$. That is, global consistency is calculated by using channel consistency. For compatibility, global consistency $Gc(C_{\mathcal{V}})$ is induced by multiplication of $Cc(\langle p_i, p_j \rangle, c_i, c_j)$ for all the channels $\langle p_i, p_j \rangle \in \mathcal{L}$. $1/|\mathcal{L}|$ is a normalization factor where $|\mathcal{L}|$ is the number of channels in $\mathcal{S}$ for independence of system scale.

[Global consistency]

$$Gc(C_{\mathcal{V}}) = \prod_{\langle p_i, p_j \rangle \in \mathcal{L}} Cc(\langle p_i, p_j \rangle, c_i, c_j)^{1/|\mathcal{L}|} \quad (4)$$

□

5 Checkpoint Protocol

Here, we design a checkpoint protocol for a multimedia communication network according to the global consistency defined in (4). The protocol is based on a three-phase coordinated checkpoint protocol. In a data communication network, for avoiding inconsistent messages, each process is required to be blocked for a certain period. However, for the requirement of time-bounded execution of an application, our protocol does not require processes to block execution of an application during the checkpoint protocol. Each process p_i takes a local checkpoint c_i immediately when p_i is required to take c_i. That is, the protocol is non-blocking. In this protocol, there is a coordinator process p_c. Here, we make the following assumptions:

- A sequence number $seq(m)$ is assigned to a message m when m is transmitted. $seq(m)$ is piggied back to each packet pa_k of m.
- $rvalue(pa_k, m) = value(pa_k)/value(m)$ is carried by each packet pa_k.

According to the definition of the global consistency in multimedia network systems in the previous section, the less lost packets are, the higher global consistency we achieve. Hence, the following rules are applied:

- If a process p_i is sending a message, it is better to take a local checkpoint c_i immediately for high consistency.
- If p_i is receiving a message, it is better to take c_i later.

However, a timely checkpoint is also required. That is, the checkpoint protocol is required to be finished within a predetermined period. The checkpoint protocol is as follows [Figure 3]:

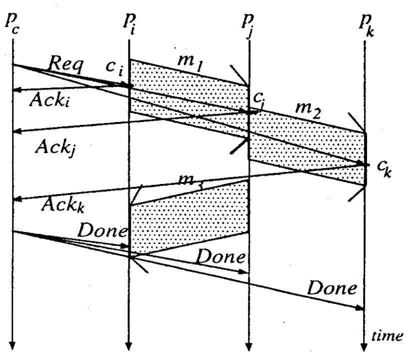

Figure 3: Multimedia checkpoint protocol.

[Checkpoint protocol]

1) Let RC ($0 \leq RC \leq 1$) be required global consistency and τ be a required time limit for a checkpoint protocol. RC and τ are QoS parameters determined by a coordinator process p_c. p_c sends checkpoint request messages *Reqs* to all the processes $p_i \in \mathcal{V}$.
2) Each process p_i takes a tentative local checkpoint tc_i according to the following rules.
   2-1) If p_i is sending a message, p_i takes tc_i on receipt of the *Req*.
   2-2) If p_i is receiving a message or is not communicating with other processes, p_i postpones taking tc_i for $\tau - 2\delta_i$ where the transmission delay of $\langle p_c, p_i \rangle$ is δ_i. During this period, if p_i begins sending a message, p_i takes tc_i immediately.
3) Each p_i sends back an acknowledgment message Ack_i to p_c. For every communication channel $\langle p_i, p_j \rangle$ ($\langle p_j, p_i \rangle$), $seq(m_{ij})$ ($seq(m_{ji})$), $tvalue(m_{ij}) = \sum rvalue(pa_k, m_{ij})$ ($tvalue(m_{ji}) = \sum rvalue(pa_k, m_{ji})$ for all the packets pa_k of the last message m_{ij} (m_{ji}) sent (received) before taking tc_i where $sb(m_{ij})$ occurred before taking tc_i ($rb(m_{ji})$ occurred before taking tc_i) are piggied back to Ack_i. That is, $Ack_i.seq_{ij} = seq(m_{ij})$, $Ack_i.tvalue_{ij} = tvalue(m_{ij})$, $Ack_i.seq_{ji} = seq(m_{ji})$ and $Ack_i.tvalue_{ji} = tvalue(m_{ji})$ are piggied back to Ack_i.
4) On receipt of all Ack_i, p_c calculates $Cc(\langle p_i, p_j \rangle, c_i, c_j)$ for every communication channel $\langle p_i, p_j \rangle \in \mathcal{L}$. The followings are induced from (3).
   4-1) $Cc(\langle p_i, p_j \rangle, c_i, c_j) = 0$, if $Ack_i.seq_{ij} < Ack_j.seq_{ij}$.
   4-2) $Cc(\langle p_i, p_j \rangle, c_i, c_j) = 1 - (Ack_i.tvalue_{ij} - Ack_j.tvalue_{ij})$, if $Ack_i.seq_{ij} = Ack_j.seq_{ij}$.
   4-3) $Cc(\langle p_i, p_j \rangle, c_i, c_j) = Ack_j.tvalue_{ij}$ $(1 - Ack_i.tvalue_{ij})$, if $Ack_i.seq_{ij} = Ack_j.seq_{ij} + 1$.
   4-4) $Cc(\langle p_i, p_j \rangle, c_i, c_j) = 0$, if $Ack_i.seq_{ij} > Ack_j.seq_{ij} + 1$.
5) p_c calculates $Gc(C_{\mathcal{V}})$ according to (4).
6) If $Gc(C_{\mathcal{V}}) > RC$, p_c sends *Done* messages to $p_i \in \mathcal{V}$. Otherwise, p_c sends *Cancel* messages to $p_i \in \mathcal{V}$.
7) On receipt of *Done*, each p_i changes tc_i to a stable local checkpoint c_i. On receipt of *Cancel*, each p_i discards tc_i. □

6 Evaluation

In order to evaluate the proposed criteria of consistency for a global checkpoint and the checkpoint protocol, we apply them to an MPEG-2 data transmission. MPEG-2 is a specification of video data compression. The amount of an original video data is 720 × 480 dots/frame and 29.97 frames/sec [1]. Each frame is encoded to three kinds of pictures; I-picture, P-picture and B-picture.

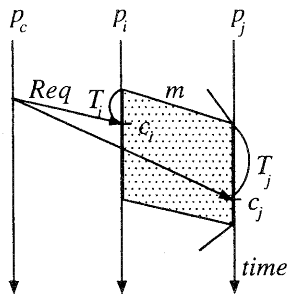

Figure 4: Evaluation parameters.

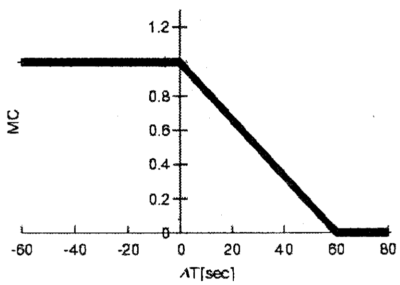

Figure 5: Consistency in MPEG-2 (60 [sec]).

Suppose there are two processes p_i and p_j connected by a communication channel $\langle p_i, p_j \rangle$ and a multimedia message m is transmitted through $\langle p_i, p_j \rangle$ as in Figure 4. In the proposed checkpoint protocol, *Req* messages are transmitted from a coordinator process p_c to p_i and p_j. On receipt of the *Req* messages, p_i and p_j take local checkpoints c_i and c_j, respectively. Let T_i be time duration from $sb(m)$ to $r(Req)$, i.e. taking c_i in p_i, and T_j be time duration from $rb(m)$ to $r(Req)$, i.e. taking c_j in p_j. Here, message transmission delay of communication channels $\langle p_c, p_i \rangle$ and $\langle p_c, p_j \rangle$ are not the same. Let $\Delta T = T_i - T_j$.

Figure 5 shows relation between ΔT and $MC = Mc(m, c_i, c_j)$ for a message m which includes 60sec MPEG-2 data. In MPEG-2, if a B-picture is lost, only one frame cannot be decoded. However, if an I-picture is lost, all the frames in the GOP (Group of Pictures) cannot be decoded. That is, $value(pa_k)$ is different for each pa_k. Thus, the mapping from ΔT to MC is not one-to-one but one-to-N. According to Figure 5, $MC(5.52) = [0.900, 0.907]$ and $MC(5.90) = [0.894, 0.900]$. Hence, if required consistency is 0.9 and $\Delta T < 5.52$, a global checkpoint $\{c_i, c_j\}$ is consistent. In addition, if $\Delta T < 5.90$, $\{c_i, c_j\}$ might be consistent. This depends on which pictures are lost due to difference of transmission delay for *Req* messages. Therefore, even if p_i and p_j are not completely synchronized, we can achieve QoS based consistent global checkpoint.

[1] This encoding is called MP@ML (Main Profile, Main Level).

7 Concluding Remarks

This paper proposes novel consistency of global checkpoints in multimedia communication networks. Unlike the conventional consistency, it allows for processes to take local checkpoints during communication events and to lose a part of a message in recovery. In addition, we show a checkpoint protocol based on the proposed consistency. The checkpoint protocol is non-blocking for supporting time-constrained applications. In addition, it is QoS-based where QoS parameters are consistency and timeliness. The evaluation shows that the consistency and the protocol work well in the system transmitting an MPEG-2 data.

References

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

[2] Douglas, E.C., "Internetworking with TCP/IP," Prentice-Hall (1991).

[3] Elozahy, E.N., Johnson, D.B. and Wang, Y.M., "A Survey of Rollback-Recovery Protocols in Message-Passing Systems," Technical Note of Carnegie Mellon University, CMU-CS-96-181 (1996).

[4] ISO/IEC 13818 and ISO/IEC 11172, "The MPEG Specification," http://www.mpeg2.de/.

[5] Johnson, D.B., "Efficient Transparent Optimistic Rollback Recovery for Distributed Application Programs,"the 12th International Symposium on Reliable Distributed Systems, pp. 86–95 (1993).

[6] 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).

[7] Mathew, E. H. and Russell, M. S., "MULTIMEDIA COMPUTING - Case Studies from MIT Project Athena," Addison-Wesley (1993).

[8] Pankaj, J., "Fault Tolerance in Distributed Systems," Prentice Hall, pp.185–213 (1994).

[9] Shimamura, K., Tanaka, K. and Takizawa, M., "Group Protocol for Exchanging Multimedia Objects in a Group," 2000 ICDCS Workshop on Group Computation and Communications, pp. 33–40 (2000).