Consistent Global Checkpoints in Multimedia Network Systems

Kengo Hiraga, Kenji Hasebe and Hiroaki Higaki
{hira,namu,hig}@higlab.k.dendai.ac.jp
Department of Computers and Systems Engineering
Tokyo Denki University

To achieve a 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 network, a global checkpoint is defined consistent if there is neither orphan nor lost message. For multimedia communication network, there are different requirements such as time-constrained failure free execution, large-size massages and allowance of certain amount of messages. This paper proposes a new criteria of consistency for supporting multimedia communication network. In addition, a checkpoint protocol with QoS based consistency is designed and applied to MPEG-2 data transmission.

1 Introduction

The 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. Some 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 important method to realize fault-tolerant distributed systems is a checkpoint-recovery [6, 17]. During failure-free execution, each process takes local checkpoints by storing state information into a stable storage [15]. If a certain process fails, the processes restart execution from the checkpoints by restoring the state information from the stable storage. For restarting execution of an application correctly, a set of local checkpoints taken by all the processes and from which the processes restart should form a *consistent global checkpoint* [3]. A consistent global checkpoint is defined as that there is neither *orphan message* nor *lost message*. However, the definition is too strict in a multimedia network system where messages carrying a large multimedia data are exchanged among processes. In this paper, we propose a novel consistent global checkpoint for multimedia network systems and design a checkpoint protocol.

The rest of this paper is organized as follows: In section 2, we review the conventional consistent global checkpoint in a conventional data communication network. In section 3, we discuss properties of a multimedia network system and requirements for a consistent global checkpoint. Section 4 proposes a novel consistent global checkpoint supporting a multimedia network system. According to this proposal, we design a checkpoint protocol which is based on QoS (Quality of Service) for consistency. Finally in section 6, for an evaluation, the consistency and the checkpoint protocol are applied to MPEG-2 data transmission[10]. The result shows they work well in multimedia network systems.

2 Conventional Consistency

A distributed system $\mathcal{S}$ is modeled by a tuple $\langle \mathcal{V}, \mathcal{L} \rangle$ where $\mathcal{V} = \{p_1, \ldots, p_n\}$ 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 a sequence of *events*. A *state* of p_i changes at each event. There are two kinds of events: *local events* and *communication events*. At a local event, p_i changes a state by local computation without exchanging a message. At a communication event, p_i communicates with another process by exchanging a message and changes a state. 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, there are two main kinds of methods: checkpoint-recovery and replication. In the replication[1, 8, 9, 14, 18], each process is replicated and placed on multiple computers. Even if a certain process fails, other replicated processes can continue to execute an application. On the other hand, in the checkpoint-recovery[2, 4, 6, 11, 13, 17, 21, 22], each process p_i sometimes takes a *local checkpoint* c_i by storing state information into a *stable storage*[15]. If a certain process p_i fails, p_i restarts execution from c_i by restoring the state information from the stable storage. If a process restarts independently of the other processes, there may be two kinds of *inconsistent messages*: *lost messages* and *orphan messages* [3]. Let processes p_i and p_j take local checkpoints c_i and c_j, respectively. Suppose a message m is transmitted from p_i to p_j through a communication channel $\langle p_i, p_j \rangle$.

[Inconsistent message] m is *inconsistent* iff 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 mes-

sage in any communication channel in $\mathcal{L}$. Thus, if a process p_i fails, not only p_i but also other processes p_j are required to restart execution from local checkpoints c_j. Hence, a *global checkpoint* $C_{\mathcal{V}} = \{c_1, \ldots, c_n\}$ which is a set of local checkpoints of all the processes in $\mathcal{V}$ should be *consistent*, i.e. satisfy the following condition [3]:

[Consistent global checkpoint] A global checkpoint $C_{\mathcal{V}}$ in $\mathcal{S}$ is consistent iff there is no inconsistent message, i.e. neither lost nor orphan message, in any communication channel in $\mathcal{L}$. □

3 Multimedia Networks

Recently, distributed applications such as distance learning, tele-conference, tele-medicine and video on demand are developed on communication networks [16], e.g. the Internet. That is, messages carrying multimedia data including text, voice, sound, picture and video are exchanged among processes to execute an application. These messages are so large that it takes time to transmit and receive the messages as shown in Figure 1. Here, the following four events are defined for a multimedia message m transmitted from a process p_i to another process p_j [19]:

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

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

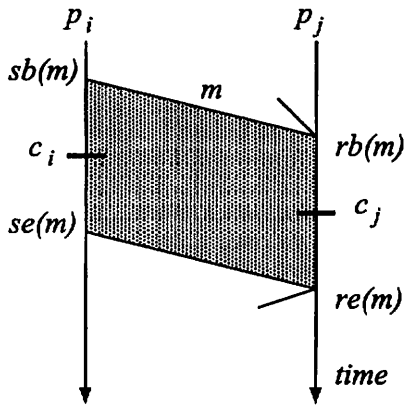

Figure 1: Multimedia message transmission.

In the conventional distributed systems with a message carrying conventional data, $s(m)$ and $r(m)$ are assumed to be atomic. Here, in the definition of a consistent global checkpoint [3], each local checkpoint is assumed to be taken only between two successive events. However, for achieving a fault-tolerant multimedia network system, each local checkpoint c_i is required to be taken even while a process p_i is transmitting and/or receiving a multimedia message m. That is, as in Figure 1, checkpoints c_i and/or c_j may be taken between $sb(m)$ and $se(m)$ and/or between $rb(m)$ and $re(m)$, respectively.

In addition, in a multimedia application, a certain amount of data in a multimedia message can be lost in a communication channel. Such an application requires not to retransmit lost messages but to transmit messages with shorter transmission delay and smaller jitter. Hence, an overhead for taking a checkpoint during a failure-free execution is required to be reduced.

In a computer communication network, protocols are hierarchically composed. For example, an IP datagram may be decomposed into multiple Ethernet frames in a sender process since the maximum size of an IP datagram is 64[kbyte] and that of an Ethernet frame is 1.5[kbyte]. These frames are gathered in a receiver process. Thus, a multimedia message m is assumed to be decomposed into a sequence $\langle pa_1, \ldots, pa_l \rangle$ of multiple *packets* for transmission in an underlying protocol as in Figure 2.

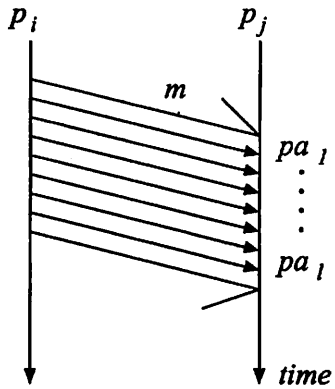

Figure 2: Packets for a multimedia message.

4 Novel Consistency

As discussed in the previous section, since a multimedia message is much larger than a conventional data message, it takes longer time to transmit and receive the message. In a conventional data communication network, a communication event, i.e. a message sending event and a message receipt one, is assumed to be atomic and completed instantaneously. Hence, in a conventional checkpoint-recovery protocol, a process takes a local checkpoint only between two successive events, not during an event. However in a multimedia communication network, since larger messages are transmitted among processes, local checkpoints are required to be taken even during a communication event in a process. By allowing taking a local checkpoint during a communication event, synchronization and communication overhead is reduced. If the conventional protocol is applied, a process is required to take a local checkpoint before starting or after finishing transmission and/or reception of a message. Hence, whole message is required to be retransmitted after recovery. However, by taking a local checkpoint during a communication event,

each process takes it immediately and only a part of a message, is required to be retransmitted after recovery. Since consistency of a global checkpoint has been defined for conventional data communication networks [3], it is required to define different consistency for multimedia communication networks.

Now, we introduce a *global consistency function* $Consistency(C_{\mathcal{V}})$ where a global checkpoint $C_{\mathcal{V}} = \{c_1, \ldots, c_n\}$ is a set of local checkpoints of all the processes in $\mathcal{V}$. $Consistency(C_{\mathcal{V}})$ denotes a degree of consistency of $C_{\mathcal{V}}$ in $\mathcal{S}$. In a conventional data communication network, $Consistency(C_{\mathcal{V}})$ is defined as follows:

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

In a multimedia communication network, a local checkpoint can be taken during a communication event, In addition, it is acceptable for an application to lose a part of a multimedia message. Hence, the domain of $Consistency(C_{\mathcal{V}})$ is required to be a closed interval $[0, 1]$ instead of a discrete set $\{0, 1\}$.

4.1 Message Consistency

First, we define a *message consistency function* $Mconsistency(m, c_i, c_j)$ as a degree of consistency for a set $\{c_i, c_j\}$ of local checkpoints and a multimedia message m where m is transmitted from a process p_i to another process p_j. Here, c_i and c_j are taken by p_i and p_j, respectively. For compatibility with the conventional definition in (1), the following should be satisfied:

$$Mconsistency(m, c_i, c_j) = 1 \quad \text{if } c_i \rightarrow sb(m) \text{ and } c_j \rightarrow rb(m). \quad (2)$$

$$Mconsistency(m, c_i, c_j) = 0 \quad \text{if } c_i \rightarrow sb(m) \text{ and } re(m) \rightarrow c_j. \quad (3)$$

$$Mconsistency(m, c_i, c_j) = 0 \quad \text{if } se(m) \rightarrow c_i \text{ and } c_j \rightarrow rb(m). \quad (4)$$

$$Mconsistency(m, c_i, c_j) = 1 \quad \text{if } se(m) \rightarrow c_i \text{ and } re(m) \rightarrow c_j. \quad (5)$$

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, a multimedia message m is decomposed into a sequence of multiple packets $\langle pa_1, \ldots, pa_l \rangle$. $s(pa_k)$ $(1 \leq k \leq l)$ is a *packet sending event* and $r(pa_k)$ is a *packet receipt event*. $s(m)$ is composed of a sequence $\langle s(pa_1), \ldots, s(pa_l) \rangle$ and $r(m)$ is composed of a sequence $\langle r(pa_1), \ldots, r(pa_l) \rangle$. 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 the definition of 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.□

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 $1 \leq s, r < l$. Clearly, if $s = r$, $Mconsistency(m, c_i, c_j) = 1$.

If $s > r$, $\{pa_{r+1}, \ldots, pa_s\}$ 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 [12]. 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. For example, storing a message log in a stable storage makes transmission delay and jitter larger in MPEG-2 data transmission. On the other hand, less than a certain threshold, a certain number of packets can be lost in recovery to execute an application. Here, we define lost consistency for a set $\{c_i, c_j\}$ of local checkpoints 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 function]

$$Mconsistency(m, c_i, c_j) = 1 - \frac{\sum_{k=r+1}^{s} value(pa_k)}{value(m)} \quad (6)$$

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

If $s < r$, $\{pa_{s+1}, \ldots, pa_r\}$ is a set of orphan packets. In a conventional data communication network, an orphan message might not be retransmitted after recovery due to non-deterministic property 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 after recovery. Hence, a set $\{c_i, c_j\}$ of local checkpoints c_i and c_j is consistent, i.e. $Mconsistency(m, c_i, c_j) = 1$.

Figure 3 shows a message consistency function $Mconsistency(s, r)$ for s and r. Let l be 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 $se(m)$ $(re(m))$.

- According to (2), $Mconsistency(s, r) = 1$ if $s \leq 0$ and $r \leq 0$.
- $Mconsistency(s, r) = 0$ if $s \leq 0$ and $r > 0$ since m is an orphan message. This case shows (3).
- $Mconsistency(s, r) = g_1(s)$ where $dg_1(s)/ds \leq 0$, $\lim_{s \rightarrow 0} g_1(s) = 1$ and $\lim_{s \rightarrow l} g_1(s) = 0$ if $0 < s < l$ and $r \leq 0$.

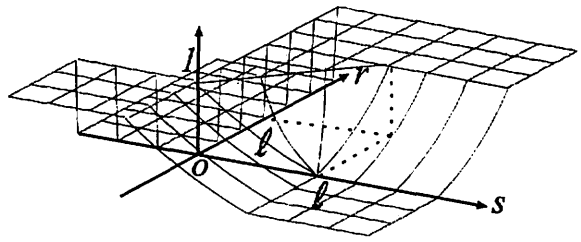

Figure 3: Message consistency.

- $Mconsistency(s, r) = g_2(s, r)$ where $g_2(u, u) = 1 (0 < u < l)$, $dg_2(s, r)/ds \leq 0$ and $dg_2(s, r)/dr \geq 0$, $\lim_{s \to l} g_2(s, r) = g_3(r)$ and $\lim_{r \to 0} g_2(s, r) = g_1(s)$ if $0 < s < l$, $0 < r < l$ and $s > r$.
- $Mconsistency(s, r) = 1$ if $0 < s < l$ and $s \leq r$.
- According to (4), $Mconsistency(s, r) = 0$ if $l \leq s$ and $r \leq 0$.
- $Mconsistency(s, r) = g_3(r)$ where $dg_3(r)/dr \geq 0$, $\lim_{r \to 0} g_3(r) = 0$ and $\lim_{r \to l} g_3(r) = 1$ if $l < s$ and $0 < r < l$.
- According to (5), $Mconsistency(s, r) = 1$ if $l \leq s$ and $l \leq r$.

4.2 Channel and Global Consistency

Based on the message consistency function $Mconsistency(m, c_i, c_j)$ for a multimedia message m and local checkpoints c_i and c_j in processes p_i and p_j respectively, a channel consistency function $Cconsistency(\langle p_i, p_j \rangle, c_i, c_j)$ is defined as a degree of consistency of a set $\{c_i, c_j\}$ of local checkpoints for a communication channel $\langle p_i, p_j \rangle \in \mathcal{L}$ where $\mathcal{M}_{ij}$ is a set of messages transmitted through $\langle p_i, p_j \rangle$.

[Channel consistency function]

$$Cconsistency(\langle p_i, p_j \rangle, c_i, c_j) = \begin{cases} 1 & \text{if } \mathcal{M}_{ij} = \phi. \\ \prod_{m \in \mathcal{M}_{ij}} Mconsistency(m, c_i, c_j) & \text{otherwise.} \end{cases} \quad (7)$$

In a distributed system $\mathcal{S} = \langle \mathcal{V}, \mathcal{L} \rangle$, a global consistency for a global checkpoint $C_{\mathcal{V}} = \{c_1, \ldots, c_n\}$ where each local checkpoint c_i is taken by a process $p_i \in \mathcal{V}$ is defined base on the channel consistency function defined in (7).

In a conventional data communication network, each local checkpoint is taken between two successive local events or communication events. In addition, at a communication event, i.e. a message sending event or a message receipt event, each process exchanges a message with other processes. In the multimedia data communication network, each local checkpoint may be taken during the occurrence of a communication event. In addition, a process may exchange multiple messages with multiple processes simultaneously. A global consistency is calculated according to relations of all the sets of two checkpoints c_i and c_j where there is a communication channel $\langle p_i, p_j \rangle$. That is, a global consistency is calculated by using channel consistencies. Therefore, a global consistency function $Consistency(C_{\mathcal{V}})$ is induced by a multiplication of $Cconsistency(\langle p_i, p_j \rangle, c_i, c_j)$ 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 $\mathcal{S}$.

[Global consistency function]

$$Consistency(C_{\mathcal{V}}) = \prod_{\langle p_i, p_j \rangle \in \mathcal{L}} Cconsistency(\langle p_i, p_j \rangle, c_i, c_j)^{1/|\mathcal{L}|} \quad (8)$$

The above definition is compatible with the conventional consistency definition. If there is at least one completely lost or orphan message in a communication channel $\langle p_i, p_j \rangle \in \mathcal{L}$, $Consistency(C_{\mathcal{V}}) = 0$ where $c_i, c_j \in C_{\mathcal{V}}$. This is because $Cconsistency(\langle p_i, p_j \rangle, c_i, c_j) = 0$.

5 Checkpoint Protocol

Here, we show a checkpoint protocol for a multimedia communication network according to the consistency defined in (8). The proposed protocol is based on a three-phase coordinated checkpoint protocol in [13]. However, our protocol does not require processes to block execution of an application during the checkpoint protocol. That is, it is a non-blocking protocol [7, 20]. In this protocol, there is a coordinator process p_c. Here, we make the following assumptions:

- A sequence number $seq(m)$ is assigned to each message m when m is transmitted. $seq(m)$ is piggied back to each packet pa_k of m.
- Each packet pa_k carries $value(pa_k)/value(m)$.

The checkpoint protocol is as follows [Figure 4]:

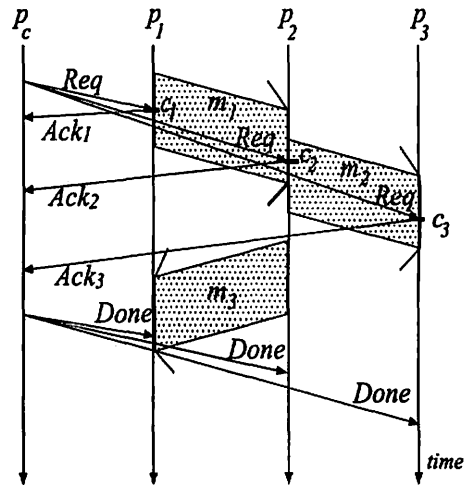

Figure 4: Multimedia checkpoint protocol.

[Checkpoint protocol]

1) A coordinator process p_c sends checkpoint request messages Reqs to all the processes $p_i \in \mathcal{V}$. Here, p_c determines a required consistency RC $(0 \leq RC \leq 1)$.
2) Each process p_i takes a tentative local checkpoint tc_i and sends back an acknowledgement message Ack_i to p_c. For every communication

channel $\langle p_i, p_j \rangle$ ($\langle p_j, p_i \rangle$), $seq(m_{ij})$ ($seq(m_{ji})$), $tvalue(m_{ij}) = \sum value(pa_k)/value(m_{ij})$ ($tvalue(m_{ji}) = \sum value(pa_k)/value(m_{ji})$) for all the packets pa_k of the lost message m_{ij} (m_{ji}) send (received) before taking c_i: where $sb(m_{ij}) \rightarrow c_i$ ($rb(m_{ji}) \rightarrow c_i$) 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.

3) On receipt of all the Ack_i messages from $p_i \in \mathcal{V}$, p_c calculates channel consistency $Cc_{ij} = Cconsistency(\langle p_i, p_j \rangle, c_i, c_j)$ for every communication channel $\langle p_i, p_j \rangle \in \mathcal{L}$.
   3-1) If $Ack_i.seq_{ij} < Ack_j.seq_{ij}$, $Cc_{ij} = 0$.
   3-2) If $Ack_i.seq_{ij} = Ack_j.seq_{ij}$, $Cc_{ij} = 1 - (Ack_i.tvalue_{ij} - Ack_j.tvalue_{ij})$.
   3-3) If $Ack_i.seq_{ij} = Ack_j.seq_{ij} + 1$, $Cc_{ij} = Ack_j.tvalue_{ij}(1 - Ack_i.tvalue_{ij})$.
   3-3) If $Ack_i.seq_{ij} > Ack_j.seq_{ij} + 1$, $Cc_{ij} = 0$.
4) p_c calculates global consistency $Gc = Consistency(C_{\mathcal{V}}) = \prod_{\langle p_i, p_j \rangle \in \mathcal{L}} Cc_{ij}^{1/\mathcal{L}}$.
5) If $Gc > RC$, p_c sends *Done* messages to $p_i \in \mathcal{V}$. Otherwise, p_c sends *Cancel* messages to $p_i \in \mathcal{V}$.
6) 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 consistency and the checkpoint protocol, we apply them to MPEG-2 data transmission. MPEG-2 is a specification of video data compression [11]. The amount of an original video data is 720 × 480 [dots/frame], 29.97 [frames/sec] [1]. Each frame is encoded to three kinds of pictures; I-picture, P-picture and B-picture. An I-picture is achieved by encoding an original frame with DCT (Discrete Cosine Transform). An original frame is achieved by decoding an I-picture alone. A P-picture and a B-picture are achieved by using the motion compensation. The sizes of a P-picture and a B-picture are about 1/3 and 1/6 of an I-picture, respectively. An original frame encoded to a P-picture is achieved by the P-picture and the previous frame encoded to an I-picture or a P-picture. If the previous I-picture or P-picture is lost, the original frame cannot be achieved. An original picture encoded to a B-picture is achieved by using the bidirectional prediction. Here, the previous and the following I-picture or P-picture is used. Thus, if one of the pictures is lost, the original frame cannot be achieved. A GOP is a unit of coding and decoding. A widely used GOP includes 15 pictures for 0.5[*sec*] video.

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 5. In the proposed checkpoint protocol, *Req* messages are transmitted from 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 a time duration from $sb(m)$ to $r(Req)$, i.e. taking c_i in p_i, and T_j be a 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 $\Delta T = T_i - T_j$.

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

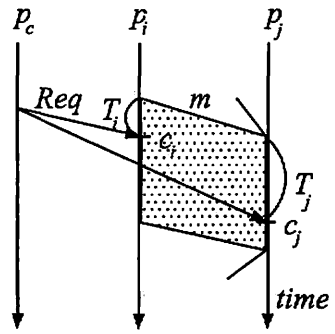

Figure 5: Evaluation parameters.

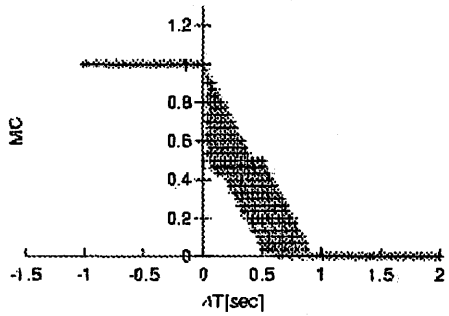

Figure 6: Consistency in MPEG-2 (1.0 [sec]).

Figure 6 and Figure 7 show relationship between ΔT and message consistency $MC = Mconsistency(m, c_i, c_j)$ for a message m which includes 1.0 [sec] and 60 [sec] 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 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 as shown in Figure 6. According to Fig-

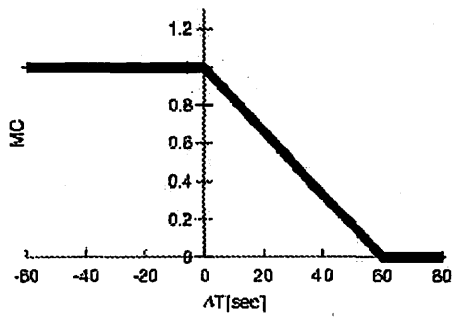

Figure 7: Consistency in MPEG-2 (60 [sec]).

ure7, $MC(5.52) = [0.900, 0.907]$ and $MC(5.90) = [0.894, 0.900]$. Hence, if a 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.

7 Concluding Remarks

This paper proposes novel consistency of global checkpoints in multimedia network systems. Unlike the conventional consistency, it allows for processes to take local checkpoints during communication events and to lose a part of a message in the recovery. In addition, we show a checkpoint protocol based on the proposed consistency. The evaluation shows that the consistency and the protocol works well in the system transmitting an MPEG-2 data. In our future work, by introducing a tradeoff between consistency and recovery time, we will design a QoS based checkpoint protocol in a multimedia communication network.

References

[1] Bernstein, P.A., and Goodman, N., "An Algorithm for Concurrency Control and Recovery in Replicated Distributed Databases," ACM Trans. on Database Systems, Vol. 9, No. 4, pp. 1197-1207 (1984).

[2] Bhargava, B. and Lian, S.R., "Independent Checkpointing and Concurrent Rollback for Recovery in Distributed Systems," The 7th International Symposium. on Reliable Distributed Systems. pp. 3-12 (1988).

[3] 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).

[4] Cristian, F. and Jahanian, F., "A Timestamp-Based Checkpointing Protocol for Long Lived Distributed Computations," Reliable Distributed Software and Database Systems, pp. 12-20 (1991).

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

[6] 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).

[7] Elnozahy, E.N., Johnson, D.B., and Zwaenepoel, W., "The performance of consistent checkpointing," International Symposium on Reliable Distributed Systems, pp. 39-47 (1992).

[8] Giffrod, D.K., "Weighted Voting for Replication Data," The 7th ACM Symposium on Operating Systems, pp. 150-162 (1979).

[9] Higaki, H., Nemoto, N., Tanaka, K. and Takizawa, M., "Protocol for Groups of Pseudo-Active Replication Objects," International Workshop on Object Oriented Realtime Distributed Systems, pp. 35-41 (1999).

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

[11] Juang, T.T.Y. and Venkatesan, S., "Efficient Algorithms for Crash Recovery in Distributed Systems." The 10th Conference on Foundations of Software Technology and Theoretical Computer Science, pp. 349-361 (1990).

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

[13] 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).

[14] Kumar, A., "Hierarchical Quorum Consensus: A New Algorithm For Managing Replicated Data," IEEE Trans. on Computers, Vol. 40, No. 9, pp. 996-1004 (1991).

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

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

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

[18] Pu, C.A., Noe, D.D. and Proudfoot, A., "Regeneration of Replicated objects: A Technique and its Eden Implementation," IEEE Trans. on Software Engineering, Vol. 14, No. 7, pp. 936-945 (1988).

[19] 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).

[20] Silva, L.M. and Silva, J.G., "Global Checkpointing for Distributed Programs," International Symposium on Reliable Distributed Systems, pp. 155-162 (1992).

[21] Venkatesh, K., Radhakrishnan, T. and Li, H.F., "Optimal and Local Recording for Domino-Free Rollback Recovery," Information Processing Letters, Vol. 25, . 295-303 (1987).

[22] Wood, W.G., "A Decentralized Recovery Protocol," The 11th International Symposium on Fault Tolerant Computing Systems, pp. 159-164 (1981).