5ZC-08

Group Communication for Multimedia Objects *

Kenichi Shimamura, Katsuya Tanaka, and Makoto Takizawa †
Tokyo Denki University ‡
Email : {ken, katsu, taki}@takilab.k.dendai.ac.jp

1 Introduction

In distributed applications like teleconferences, a group of multiple processes are cooperating. Various kinds of group communication protocols [1] are discussed so far. In the group communication, a *group* is first established among multiple processes and then messages sent by the processes are *causally* or *totally* delivered in the destination processes in the group [1]. A message m_1 *causally precedes* another message m_2 if a sending event of m_1 *happens before* a sending event of m_2 [2]. In the totally ordered delivery, even messages not to be causally preceded are delivered to every common destination of the messages in a same order. In the protocols, messages transmitted at the network level, not at application level, are ordered independently of how the applications use the messages.

In distributed applications, not only data but also various kinds of multimedia objects like image and video are exchanged among the processes in the group. Multimedia objects are larger and more complexed and structured than the traditional data units exchanged among the processes. In addition to causally delivering objects, quality of service (QoS) of a multimedia object like frame rate, bandwidth, and message loss ratio has to be satisfied in the destination processes. Some objects may have to be delivered to the applications in a predetermined time after the messages are sent.

In this paper, we newly define a causally precedent relation between objects which are exchanged among the processes. We discuss a protocol which supports the new types of causally precedent relations among the objects.

2 System Model

Distributed applications are realized by the cooperation of a group of application processes A_1, ..., A_n ($n \geq 1$) which are interconected in a *high-speed* network. Application processes exchange *data* including multimedia with the other processes in the group by using the network. A unit of data exchanged among the processes is referred to as *message object*, simply saying *object*.

An application process A_t is supported by a system process p_t (t = 1, ...,n). A system process p_s takes an object from the application process A_s and then delivers the object to the system processes supporting the destination application processes by using the basic communication service supported by the network. From here, let a term *process* mean a system process. A data unit exchanged by the processes in the network is referred to as *message*. We assume that the network supports the processes with asynchronous communication. That is, messages may be lost due to the congestions and unexpected delay and the delay time between a pair of processes is not bounded in the network. An object is decomposed into a sequence of messages and the messages are delivered to the destination processes. The destination process p_t assembles the messages received into an object and then delivers the object to the application process A_s. The cooperation of the processes supporting the group of the application processes is coordinated by a *group protocol* which supports the reliable, efficient communication service of multimedia objects by taking usage of the network service.

3 Causality of Multimedia Objects

3.1 Traditional messages

The *happen-before* relation among events occurring in a distributed system is defined by Lamport [2]. By using the *happen-before* relation, the causally precedent relation among messages exchanged among multiple processes is defined as follows [2]:

- A message m_1 *causally precedes* another message m_2 iff a sending event of m_1 happens before a sending event of m_2.

The traditional group protocols [1] discuss how to causally deliver *network-level* messages, independently of what kinds of data the messages carry. Therefore, a communication event is assumed to *atomically* occur in a process since it does not take a longer time to send and receive a message. That is, a process does not send a message while the process is receiving another message.

3.2 Multimedia objects

We discuss how a process sends and receives multimedia objects in a group G of processes p_1, ..., p_n ($n \geq 1$). Suppose that a process p_s sends an object o to another process p_t. Since a multimedia object is larger than a traditional message, it takes a longer time to send and receive the multimedia object. That is, a process may send and receive messages of an object while the process is sending and receiving other objects. Figure 1 shows three processes p_s, p_t, and p_u exchanging objects o_1 and o_2. In Figure 1(3), p_t starts to send messages of an object o_2 after receiving all the messages of another object o_1. According to the traditional causality theory, o_1 causally precedes o_2. In Figure 1(1), p_t starts to send a message of the object o_2 before receiving all the messages of o_1. Here, o_1 does not causally precede o_2. In Figure 1(2), p_t sends o_2 while receiving o_1. On the other hand, p_t sends o_2 after receiving all the messages of o_1. Here, o_1 does not causally precede o_2, either.

We discuss how a pair of the objects o_1 and o_2 can be causally preceded. We formally define the causalities among multimedia objects. Let $ss_t(o)$ and $es_t(o)$ denote events that p_t starts to send an object o and finishes the transmission of o, respectively. Let $sr_t(o)$ and $er_t(o)$ denote events that p_t starts and ends to

*マルチメディアオブジェクトにおけるグループ通信
†島村 健一, 田中 勝也, 滝沢 誠
‡東京電機大学

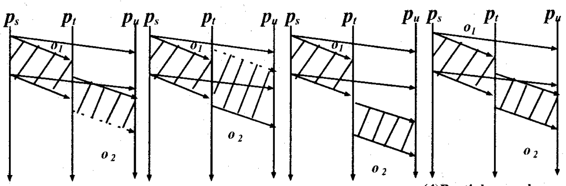

Figure 1: Multimedia messages.

receive the object o, respectively. For a traditional object o, a pair of starting events $ss_t(o)$ and events $es_t(o)$ for sending o simultaneously occur and a pair of receipt events $sr_t(o)$ and $er_t(o)$ also occur in a process at the same time. However, these events cannot be assumed to simultaneously occur in the communication of the multimedia objects.

[Definition] Following types of precedent relations are defined for a pair of objects o_1 and o_2 sent by processes p_s and p_t, respectively:

- o_1 *top-precedes* o_2 ($o_1 \rightharpoonup o_2$) iff $sr_t(o_1)$ happens before $ss_t(o_2)$ and $ss_t(o_2)$ happens before $er_t(o_1)$.
- o_1 *tail-precedes* o_2 ($o_1 \rightharpoondown o_2$) iff $er_t(o_1)$ happens before $es_s(o_2)$ and $ss_s(o_1)$ happens before $es_s(o_2)$.
- o_1 *fully precedes* o_2 ($o_1 \Rightarrow o_2$) iff $er_s(o_1)$ happens before $ss_t(o_2)$.□
- o_1 *partially precedes* o_2 ($o_1 \rightarrow o_2$) iff $o_1 \rightharpoonup o_2$, $o_1 \rightharpoondown o_2$, and o_1 is interleaved with o_2.

In Figure 1, $o_1 \rightharpoonup o_2$ in (1), $o_1 \rightharpoondown o_2$ in (2), and $o_1 \Rightarrow o_2$ in (3). Here, the process p_u is required to deliver the messages of objects o_1 and o_2 so that the causalities defined here are preserved. In Figure 1 (4), an object o_1 is *interleaved* with another object o_2 since $ss_t(o_2)$ happens before $er_t(o_1)$ and $ss_t(o_1)$ happens before $ss_t(o_2)$ in a source process p_t of o_2.

4 Protocol

We present a protocol for supporting the causally ordered delivery of multimedia objects. A group G is composed of multiple processes p_1, ..., p_n ($n > 1$).

A process p_t manipulates two variables $V = \langle V_1, ..., V_n \rangle$ and $A = \langle A_1, ..., A_n \rangle$ in order to transitively precede objects. V shows the vector clock. A is used to precede objects. Each element A_t takes integer value, not bit. Let $o.SA$ show a value of A_t when o is started to be transmitted and $o.EA$ show a value of A when the object o is ended to be transmitted.

A process p_t manipulates the variables V and A each time p_t sends an object o:

- $V_t := V_t + 1$;
- $A_t := A_t + 1$;

The process p_t manipulates the variable A when p_t finishes to send the object o:

- $A_t := A_t + 1$;

On receiving a top message of an object o from a process p_s, the process p_t manipulates the variables V and A as follows:

- $V_s :=$ max $(V_s, o.SV_s)$ ($s = 1, ..., n, s \neq t$);
- $A_s :=$ max $(A_s, o.SA_s)$ ($s = 1, ..., n, s \neq t$);

[Ordering rule] Suppose that a process p_s sends an object o_1 and another process p_t sends an object o_2 to the other processes.

- $o_1 \Rightarrow o_2$ iff $o_1.EA_v < o_2.SA_v$(v=1, ..., n, $v \neq$s).
- $o_1 \rightharpoonup o_2$ if $o_1.SV_v < o_2.SV_v$(v=1, ..., n, $v \neq$s).
- $o_1 \rightharpoondown o_2$ if $o_1.EA_v < o_2.EA_v$(v=1, ..., n, $v \neq$s).
- $o_1 \rightarrow o_2$ iff $o_1.EA_v > o_2.SA_v$, $o_1.EA_v < o_2.EA_v$, and $o_1.SV_v < o_2.SV_v$(v=1, ..., n, $v \neq$s).□

5 Evaluation

We evaluate the multimedia group protocol discussed here in terms of number of network-level messages to be causally ordered by comparing with the traditional network-level causality. Suppose that p_t receives messages m_{21}, ..., m_{2l} after sending m_1 and before sending m_2. Let N_G be the average number of $| d_t(m) |$ and N_{OG} be the average number of $| o_t(m) |$ for every message m. N_G and N_{OG} are calculated through the simulation.

We make the following assumptions on the simulation:

1. There are n processes p_1, ..., p_n in a group G.
2. Each process p_t sends one object at a time and sends totally m objects. Here, m = 1000.
3. Each object is sent to all the processes in the group G.
4. Each object is decomposed into h messages. τ is a random variable between $mint$ and $maxt$. $\bar{\tau}$ is $(mint + maxt)/2$.
5. Each process sends a message every τ time units.
6. It takes δ time units for a message to arrive at the destination.

Figure 2 shows the ratio of N_{OG} to N_G for number h of messages of an object. h shows the size of each object where $\delta/\bar{\tau} = 0.25$ and $n = 10$. N_{OG}/N_O shows how much the multimedia group protocol can reduce the computation and communication overhead. The larger an object is, the less ratio of messages are causally preceded in the multimedia group protocol than the traditional one.

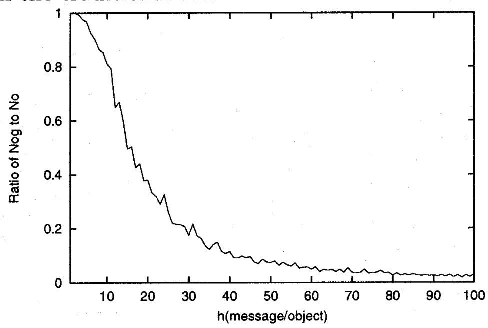

Figure 2: Evaluation.

6 Concluding Remarks

This paper has discussed a group protocol for exchanging multimedia objects. We have defined novel causally precedent relations among multimedia objects, i.e. top, tail, partially and fully precedent relations. We have shown how the multimedia group protocol can reduce the number of messages to be causally preceded.

References

[1] Birman, K., "Lightweight Causal and Atomic Group Multicast," *ACM Trans. on Computer Systems*, 1991, pp.272–290.

[2] Lamport, L., "Time, Clocks, and the Ordering of Events in a Distributed System," *Comm. ACM*, Vol.21, No.7, 1978, pp.558–565.