

INTERNET-DRAFT

[<draft-shiroshita-rmtpv2-00.txt>](mailto:draft-shiroshita-rmtpv2-00.txt)

T. Shiroshita, NTT

T. Sano, NTT

O. Takahashi, NTT

N. Yamanouchi, IBM Japan

1 September, 1997

Expires in six months

## Reliable Multicast Transport Protocol Version 2

### Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet- Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

### Abstract

This draft presents the specifications of the 'Reliable Multicast Transport Protocol Version 2,' which is to be used for information delivery such as newspaper delivery and software updates.

The protocol aims to realize the completely reliable multicast distribution of bulk data from a server to thousands of receivers. Version 2 added new functions of information access control, rate flow control, and suspending resume.



## Table of Contents

|                                                                                    |                    |
|------------------------------------------------------------------------------------|--------------------|
| <a href="#">1. Introduction.....</a>                                               | <a href="#">3</a>  |
| <a href="#">2. Description of Extension Features.....</a>                          | <a href="#">3</a>  |
| <a href="#">2.1 Common Extension Rules.....</a>                                    | <a href="#">3</a>  |
| <a href="#">2.2 Description of Individual Extended Features.....</a>               | <a href="#">3</a>  |
| <a href="#">2.2.1 Rate Control.....</a>                                            | <a href="#">3</a>  |
| <a href="#">2.2.2 Large Volume Delivery.....</a>                                   | <a href="#">4</a>  |
| <a href="#">2.2.3 Packet Scrambling.....</a>                                       | <a href="#">7</a>  |
| <a href="#">2.2.4 Group Management and Terminal Authentication.....</a>            | <a href="#">8</a>  |
| <a href="#">2.2.5 Suspend and Resume.....</a>                                      | <a href="#">9</a>  |
| <a href="#">2.2.6 Optional Information Transfer in Connection Establishment...</a> | <a href="#">20</a> |
| <a href="#">2.2.7 Abort Request to Individual Terminal.....</a>                    | <a href="#">20</a> |
| <a href="#">3. Packet Format.....</a>                                              | <a href="#">21</a> |
| <a href="#">3.1 Encoding Rules.....</a>                                            | <a href="#">21</a> |
| <a href="#">3.2 Packet Format.....</a>                                             | <a href="#">22</a> |
| <a href="#">References.....</a>                                                    | <a href="#">29</a> |
| <a href="#">Author's Addresses.....</a>                                            | <a href="#">29</a> |



## Reliable Multicast Transport Protocol Version 2 Part 2 (Extension Part)

### [1. Introduction](#)

#### (1) Structure of This Specification

This Part 2 is the specification of the extension of RMTP Protocol Specification Version 1. The Version 2 Part 1 (Base Part) is identical to RMTP Protocol Specification Version 1. The entire RMTP Protocol Specification Version 2 consists of Part 1 (Base Part) and this Extension Part.

#### (2) Extended Functions

In addition to reliable information distribution based on multicast specified in the RMTP Specification Base Part, this part specifies the following extended protocol features to improve services and performance: services in mobile environments, dynamic rate control that improves distribution performance, and scrambling for security.

This Extension Part document describes the differences from Version [1](#) (i.e., addition and changes to Version 1).

### [2. Description of Extension Features](#)

#### [2.1 Common Extension Rules](#)

Declaration about use of the extended features at connection establishment:

- Use of each extended feature is optional (both in implementation and in use) except for the following:
  - \* The terminal should return its terminal ID together with ACK.
- In the case where the server declares to use an option and a terminal does not support the option, the decision of whether to execute or not to execute the option should be made according to the nature of the option. It is not specified in this document.
- Use of each option is designated in the parameter field of the connection establishment request (CONN) packet.

#### [2.2 Description of Individual Extended Features](#)

##### [2.2.1 Rate Control](#)

###### (1) Overview

The server monitors the packet arrival conditions at the terminals and controls the transmission rate of the data packets.

## (2) Protocol Sequence

The server monitors the packet arrival conditions at the terminals and controls the transmission rate using the following steps.

### Step1 (Server)

The server transmits a DT packet with the flow measurement start bit turned ON.

### Step1 (Terminal)

A terminal records the DT sequence number on receipt of a DT packet with flow measurement bit ON, and starts counting the number of DT packets.

### Step2 (Server)

The server transmits a DT packet with the flow measurement stop bit turned OFF. It starts the report timer.

### Step2 (Terminal)

A terminal records the DT sequence number on receipt of a DT packet with flow measurement bit OFF, stops counting the number of DT packets, and records the total number of DT packets counted. The terminal returns the DT sequence numbers of the measurement start and measurement stop DT packets, and the total number of counted DT packets to the server via a flow control measurement report packet (FREPO packet).

### Step3 (Server)

The server collects the FREPO packets, summarizes the DT receipt status of the terminals, and reflects the result to the DT transmission rate. Timeout may be called by the report timer.

### (Note)

- The adjustment of the transmission rate is implementation dependent. It will be determined by such factors as the number of reporting terminals and the reported number of packets.

## (3) Packet Format

- The measurement start and stop bits in a DT packet.
- The flow control measurement report packet (FREPO).

### [2.2.2 Large Volume Delivery](#)

#### (1) Overview

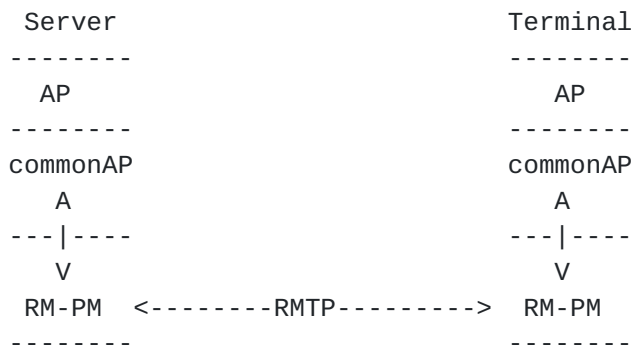
A common application layer divides a large volume of data into small messages which are delivered sequentially by RMTP.



## (2) Protocol Sequence

- The large volume delivery uses RMTP for delivering the divided messages.
- Except for the points described below, the transfer sequence of the divided messages is identical to the Base RMTP transfer sequence.
  - \* Omit the acks for the connection establishment (CAKs) in the middle of repeated transmissions except for the first transmission. The connection established for the transmission of the first divided message is retained through the transmission of the remaining divided messages.

This feature is defined as a common AP (both on the server and the terminals) of the RMTP Protocol Machine (RM-PM) that controls the underlying RM-TP.???



## Assumed Tables

This feature assumes a BC table that records success/failure of RMTP transmission for each divided message in both the server and the clients.

## Protocol Sequence (procedure) on Server

Event (CO\_VLDD): Received connection establishment request (first divided message of a large volume transfer).

This is usually the transmission start command from the upper layer AP.

Previous State: Before transmission (CLOSED\_VLDD).

Action: Request transmission of a connection establishment request packet CONN (first time) to RM-PM.

Proceed to the divided messages delivery in process (ON\_VLDD) state.



Event (BM-COMP): Received the transfer completion report of each divided message from RM-PM.

Previous State: ON\_VLDD.

Action: Record the completion result (success or failure) in the BC table.

If the server has more divided messages to send (not\_last), request CONN (intermediate) to RM-PM. Continue ON\_VLDD state.

If the server has no more divided messages to send (last), proceed to CLOSED\_VLDD.

#### Server State Transition Table

| Event State | CLOSED_VLDD    | ON_VLDD                                       |
|-------------|----------------|-----------------------------------------------|
| CO-VLDD     | CONN-->ON_VLDD |                                               |
| BM-COMP     |                | (not_last)-->ON_VLDD<br>(last) -->CLOSED_VLDD |

Blank: ignore.

#### Protocol Sequence (procedure) on Terminals

Event (CO-VLDD-R): Received connection establishment request (first divided message of a large volume transfer).

Previous State: Before delivery (CLOSED\_VLDD\_R)

Action: Create a BC table. Proceed to Large volume data transfer in progress (ON\_VLDD\_R).

Event (BM-COMP-R): Received the transfer completion report of each divided message from RM-PM.

Previous State: ON\_VLDD\_R.

Action: Record the completion result (success or failure) in the BC table.

If the server has more divided messages to be sent (not\_last), continue ON\_VLDD\_R state.

If the server has no more divided messages to be sent (last), proceed to CLOSED\_VLDD\_R.

#### Terminal State Transition Table

| Event State | CLOSED_VLDD_R    | ON_VLDD_R                                         |
|-------------|------------------|---------------------------------------------------|
| CO-VLDD-R   | CONN-->ON_VLDD_R |                                                   |
| BM-COMP-R   |                  | (not_last)-->ON_VLDD_R<br>(last) -->CLOSED_VLDD_R |

Blank: ignore.

(Note) Retransmission of the divided messages based on the

BC table is beyond the scope of this specification.

### (3) Packet Format

Following parameters are added to the CONN packets of the first and all subsequent transmission of divided messages.

- \* original message size (message size before dividing)
- \* divided message size
- \* sequence number of divided message

### [2.2.3](#) Packet Scrambling

#### (1) Overview

Packet scrambling scrambles the DT packet sequence numbers, transmits the packet to the terminals in a scrambled order, and has the terminal unscramble the sequence number to the original number based on the key delivered prior to the transmission. There is an optional "user unscramble" mode that users unscramble the data at the time of use while the RMTP terminal receives and stores the data without unscrambling.

#### (2) Protocol Sequence

##### Automatic Unscramble Mode

The server reorders the packet sequence numbers of a unit scramble block (constant size) into a fake (scrambled) sequence.

If the message cannot be divided evenly into the scramble block size, SC padding packets are added so that the message size is evenly divided into the scramble blocks.

The terminal receives the messages, replacing the fake sequence numbers into the original sequence numbers. The sequence numbers contained in the NACK and RRDY packets should be fake numbers.

After receiving the entire message, the SC padding packets are removed.

##### User Unscramble Mode

The terminal receives the DT packets according to the fake sequence numbers and passes them to the application. SC padding packets should be removed by the application.

(Note) Distribution of the unscrambling key to the terminals is beyond the scope of this specification.

### (3) Packet Format

Parameters of CONN packet:

The message size field should contain the following scrambled message length.

Shiroshita, Sano, Takahashi, Yamanouchi

[Page 7]

Scrambled message length

= message size+padding of the last DT packet+SC padding size

The "True Message Size" field of the CONN packet optional parameter should contain the original message size.

#### **2.2.4 Group Management and Terminal Authentication**

##### **(1) Overview**

This feature manages the terminal identity by their names instead of their IP addresses. To implement this feature a terminal name is added to the connection establishment acknowledgment packet CACK. Also, a terminal authentication information is added to CACK in order to authenticate the terminal name.

##### **(2) Protocol Sequence**

(Connection by terminal names)

The terminal replies to the server by CACK associated with the terminal name. The server identifies the connected terminal by this name and the sender address of the CACK IP packet.

The use of this feature is mandatory (not optional) in the communication based on RMTP version 2.

(Authentication)

The terminal also adds authentication information as well as the terminal name to the CACK. The server authenticates the terminal when it receives a CACK. At the same time the terminal's IP address included in the CACK packet is regarded as the valid terminal address during the connection.

If a name is not associated with CACK, the terminal is identified by the IP address. In a mobile environment where IP address is dynamically assigned, the use of CACKs associated with names is highly recommended.

(Notes)

- Distribution of the key necessary for generating authentication information is beyond the scope of this specification.
- The action to be taken when a terminal is found to be unauthorized (such as aborting communication) is beyond the scope of this specification.

##### **(3) Packet Format**

- Add terminal name and authentication information to CACK.



### **2.2.5 Suspend and Resume**

#### **(1) Overview**

In case of unstable communication links such as wireless links in a mobile environment, this function allows RMPT to complete the data retransmissions to the normal terminals first by suspending retransmissions to the unstable terminals. The retransmissions to the unstable terminals are performed at a later time.

While Busy Retransmission is an individual retransmission capability controlled by the terminal; Suspend/Resume is controlled by the server.

#### **(2) Protocol Sequence**

##### **Suspend:**

- 1) When the server detects a terminal whose receive performance is poor, it determines to suspend retransmission to the terminal and marks it for individual retransmission.
- 2) The server sends a suspend (SUSP) packet via unicast to the terminal to be suspended.

##### **Resume:**

After the main transmission phase completes, the following procedure is invoked in the individual retransmission phase.

- 1) The server sends a POLL to the suspended terminal via unicast.
- 2) The terminal replies with a NACK when it receives the POLL. The NACK should contain sequence numbers of the missing/unreceived data packets.
- 3) The server retransmits the data packets to the terminal via unicast according to the NACK information. Steps after this point is identical to the BUSY retransmission procedure.

##### **Exception handling:**

If the server receives any unexpected responses such as NACK, BUSY and RRDY from a SUSPended terminal, it should send a SUSP packet again to the terminal via unicast.

If the server receives an ACK from a SUSPended terminal, it processes the terminal as it has received all data packets normally, and records a normal completion on the log.

##### **(Notes)**

- 1) The criteria for determining that a terminal has an unstable link and needs individual retransmission is beyond this specification. Examples of such criteria include
  - (case 1) the number of POLL time out occurrences exceeds a predefined limit, and
  - (case 2) the number of unreceived packets reported in a NACK exceeds

a predefined limit.

2) The priority between BUSY retransmission and SUSP/RESUME retransmission is beyond this specification.

(3) Packet Format

Suspend notification packet (SUSP)



Additional Specification for Suspending Designation and Resume  
 Additional specification to part 1 [section 3](#) by this [section 2.2.5](#)  
 is described as follows.

Section in Part1: 3.1 State and event definition

3.1.1 State and event definition (Server side)

Major Transitions of Multicast on the Server Side

State Definition: no addition.

Event Definition:

| Event                       | Previous state | Action                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Suspending designation (ex) | OPEN           | Send a SUSP packet to the terminal.<br>Enter into CC tables.<br>(1) If there are terminals not BUSY nor Suspending, continue OPEN state<br>(2) If all terminals are BUSY or Suspending, proceed to Separate Retransmissions Phase to "BUSY terminals" state.                                                                                           |
|                             | ACK/NACK wait  | Enter into CC tables.<br>(1) If some responses not received, continue ACK/NACK wait state.<br>(2) All responses received, if NACK and BUSY, proceed to OPEN state.<br>(3) All responses received, if there is ACK, send REL and proceed to Connection Release Response wait state<br>(4) If all BUSY or RRDY, proceed to Separate Retransmission phase |

State Transition of Separate Retransmission Phase  
 to BUSY Terminals on the Server Side

No modification in the state definition and the event definition.



State Transition of Separate Retransmission Phase  
to Suspending Designated Terminals on the Server Side  
State definitions

Separate Retransmission wait(SOP): Following suspending designation the state of waiting for a separate retransmission request to the terminal occurs.

ACK/NACK wait(SAW): Following data transmission, the state of waiting for response from terminal (ACK/NACK/BUSY).

Connection release response wait(SRW): Following request for connection release, the state of waiting for response from the terminal.

Separate re-transmission completion(SCL): The state of separate re-transmission to the said terminal being completed.

Separate re-transmission phase to BUSY terminal(BS): Following BUSY reception, the phase in which re-transmission for each separate terminal is being conducted.

Event Definition: State Transition of Separate Retransmission Phase  
to Suspending Designated Terminals

| Event                                          | Previous state                    | Action                                                                                                                                                                         |
|------------------------------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Transmission request (separate, initial)(ex)   | Separate Retransmission wait      | Send POLL packet and proceed to ACK/NACK wait.                                                                                                                                 |
| Transmission request (separate, following)(ex) | Separate Retransmission wait      | Transmit retransmission data and proceed to ACK/NACK wait.                                                                                                                     |
| ACK reception                                  | ACK/NACK wait                     | MC table entry, send connection release packet, and proceed to Connection Release Response wait state.                                                                         |
| NACK reception                                 | ACK/NACK wait                     | MC table entry and proceed to Separate Retransmission wait state.                                                                                                              |
| Connection release response (RACK) reception   | ConnectionW Release Response wait | Release resources of separate retransmission to this terminal. Enter conclusion of retransmission into the MC table and conclude the separate retransmission to this terminal. |

continue to the next page



Event Definition: State Transition of Separate retransmission phase  
to BUSY Terminals (cont'nd)

| Event                                                                           | Previous state                   | Action                                                                                                                                                                                          |
|---------------------------------------------------------------------------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BUSY reception                                                                  | ACK/NACK wait                    | Release resources of separate retransmission to this terminal.<br>Proceed to Busy release wait phase to BUSY terminal.                                                                          |
|                                                                                 | Separate Retransmission wait     | Release resources of separate retransmission to this terminal.<br>Proceed to Busy release wait phase to BUSY terminal.                                                                          |
| TIMEOUT (ex)                                                                    | ACK/NACK wait                    | Send POLL packet and after reaching retry limits, release resources of separate retransmission to this terminal, enter into CC table and conclude the separate retransmission to this terminal. |
|                                                                                 | Connection Release Response wait | Release resources of individual retransmission to this terminal, enter conclusion of retransmission into MC table, enter into CC table and conclude separate retransmission to this terminal.   |
|                                                                                 | Busy Release wait                | Send POLL packet and after reaching retry limits, release resources of separate retransmission to this terminal, enter into CC table and conclude the separate retransmission to this terminal. |
| ABORT reception<br>(from terminals<br>conducting<br>separate<br>retransmission) | all time                         | Release resources of separate retransmission to this terminal, enter into CC table and conclude separate retransmission to this terminal.                                                       |

ABORT transmission requests from servers conducting separate retransmission and ABORT reception from terminals other than the said terminals conducting separate retransmission have been included in Multicast main transition.



## Section in Part1: 3.1.2 Definition of States and Events (terminal side)

State definition: terminal side

Addition.

|                                       |                                                            |
|---------------------------------------|------------------------------------------------------------|
| +-----+-----+-----+-----+-----+-----+ |                                                            |
| SUSPENDING                            | State of suspending communication by receiving SUSP packet |
|                                       | from a server                                              |
| +-----+-----+-----+-----+-----+-----+ |                                                            |

Event Definition: terminal side

Addition

|                                       |                |                                          |  |
|---------------------------------------|----------------|------------------------------------------|--|
| +-----+-----+-----+-----+-----+-----+ |                |                                          |  |
| Event                                 | Previous state | Action                                   |  |
| +-----+-----+-----+-----+-----+-----+ |                |                                          |  |
| SUSP                                  | CONNECTED      | Proceed to SUSPEND state.                |  |
|                                       |                |                                          |  |
| +-----+-----+-----+-----+-----+-----+ |                |                                          |  |
|                                       | RELEASE        | Send ACK and stay in RELEASE state.      |  |
|                                       |                |                                          |  |
| +-----+-----+-----+-----+-----+-----+ |                |                                          |  |
|                                       | BUSY           | Send BUSY packet and stay in BUSY state. |  |
|                                       |                |                                          |  |
| +-----+-----+-----+-----+-----+-----+ |                |                                          |  |
|                                       | other          | Notify error internally and stay in the  |  |
|                                       |                | previous state.                          |  |
| +-----+-----+-----+-----+-----+-----+ |                |                                          |  |



## Section in Part1: 3.2 State Transition Table

## 3.2.1 State Transition Table (Server side)

Symbol and Notation are the same as those in part 1.

Summary of additions

(1) Multicast main transition table

An event "Suspending designation" is added to the table.

The state which transited by BUSY or Suspending designation is expressed as "Separate retransmission phase BS/SS."

(2) State Transition Table of Separate Re-transmissions to BUSY Terminals  
No addition.

(3) State Transition Table of Separate Re-transmissions by Suspending  
Resume

A new state transition table is added. Busy may occur in the separate retransmission.

Note) Suspending resume in separate retransmission is out of the scope of this specification.



Multicast Main Transition Table (Server side)

| State     | CLOSED | Connection | OPEN  | ACK/NACK  | Connection | Separate Re- |
|-----------|--------|------------|-------|-----------|------------|--------------|
|           | CL     | Establish- | OP    | wait      | Release    | transmission |
|           |        | ment       |       | AW        | Response   | Phase        |
| Event     |        | Response   |       |           | wait       |              |
|           |        | wait       |       |           | RW         | BS/SS        |
| Trans-    | error  | error      | DT    | error     | error      | error        |
| mission   |        |            | ->AW  |           |            |              |
| request   |        |            |       |           |            |              |
| (ex)      |        |            |       |           |            |              |
| ACK       |        |            |       | mct, cct, |            |              |
| reception |        |            |       | (2)REL    |            |              |
|           |        |            |       | ->(1)AW,  |            |              |
|           |        |            |       | (2)RW     |            |              |
| NACK      |        |            |       | mct, cct, |            |              |
| reception |        |            |       | (3)REL    |            |              |
|           |        |            |       | ->(1)AW,  |            |              |
|           |        |            |       | (2)OP,    |            |              |
|           |        |            |       | (3)RW     |            |              |
| BUSY      |        | CACK       | cct   | cct,      |            |              |
| reception |        | completed  | ->    | (3)REL    |            |              |
|           |        | terminals  | (1)OP | ->(1)AW,  |            |              |
|           |        | cct->CW    | (2)BS | (2)OP,    |            |              |
|           |        |            |       | (3)RW,    |            |              |
|           |        |            |       | (4)BS     |            |              |
| Suspend-  |        |            | SUSP, | SUSP, cct |            |              |
| ing       |        |            | cct-> | (3)REL    |            |              |
| desig-    |        |            | (1)OP | ->(1)AW,  |            |              |
| nation    |        |            | (2)SS | (2)OP,    |            |              |
|           |        |            |       | (3)RW,    |            |              |
|           |        |            |       | (4)SS     |            |              |
| Time out  |        | (not last) |       | (not      | ce, cct    |              |
|           |        | (a)->CONN  |       | last)     | ->(a)CL,   |              |
|           |        |            |       | POLL->AW  | (b)OP,     |              |
| (ex)      |        | (last)ccp, |       | (last)    | (c)BS      |              |
|           |        | (1)ce->    |       | mct, cct, |            |              |
|           |        | (1)OP,     |       | cr->(1)OP |            |              |
|           |        | (2)CL,     |       | (2)BS,    |            |              |
|           |        | (3)BS      |       | (3)CL     |            |              |
| Separate  |        |            |       |           | rr, ap     |              |
| retrans.  |        |            |       |           | ->CL       |              |

|                                       |  |  |  |  |  |  |
|---------------------------------------|--|--|--|--|--|--|
| completed                             |  |  |  |  |  |  |
| (ex)                                  |  |  |  |  |  |  |
| +-----+-----+-----+-----+-----+-----+ |  |  |  |  |  |  |

continue to the next page

Multicast Main Transition Table (Server side)

(contn'd)

| State     | CLOSED | Connection   | OPEN         | ACK/NACK     | Connection   | Separate Re- |
|-----------|--------|--------------|--------------|--------------|--------------|--------------|
|           | CL     | Establish-   | OP           | wait         | Release      | transmission |
|           |        | ment         |              | AW           | Response     | Phase        |
|           |        | Response     |              |              | wait         | BS/SS        |
| Event     |        | wait CW      |              |              | RW           |              |
| ABORT     | error  | AB,cct,ap,rr | AB,cct,ap,rr | AB,cct,ap,rr | AB,cct,ap,rr | AB,cct,ap,rr |
| trans-    |        | ->CL         | ap,rr        | ->CL         | ->CL         | ->CL         |
| mission   |        |              | ->CL         |              |              |              |
| request   |        |              |              |              |              |              |
| (ex)      |        |              |              |              |              |              |
| ABORT     |        | cct          | mct,         | mct,cct      | cct,         | ABORT from   |
| reception |        | ->CW         | cct          | ->(1)AW,     | (a)(2)rr,ap  | terminals    |
|           |        |              | ->OP         | (2)RW,       | ->(a)(1)RW,  | other those  |
|           |        |              |              | (3)OP,       | (2)CL        | conducting   |
|           |        |              |              | (4)BS        | (b)(1)RW,    | separate     |
|           |        |              |              |              | (2)OP        | re-          |
|           |        |              |              |              | (c)(1)RW,    | transmission |
|           |        |              |              |              | (2)BS        |              |
| Connec-   | CONN   | error        | error        | error        | error        | error        |
| tion      | ->CW   |              |              |              |              |              |
| establish |        |              |              |              |              |              |
| request   |        |              |              |              |              |              |
| (ex)      |        |              |              |              |              |              |
| CACK      |        | cct,         |              |              |              |              |
| reception |        | (2)ce, mct   |              |              |              |              |
|           |        | set-up ->    |              |              |              |              |
|           |        | (1)CW, (2)OP |              |              |              |              |
|           |        | (3)CL, (4)B  |              |              |              |              |
| RACK      |        |              |              |              | cct, (a)     |              |
| reception |        |              |              |              | (2)rr,ap     |              |
|           |        |              |              |              | ->(a)(1)RW,  |              |
|           |        |              |              |              | (2)CL        |              |
|           |        |              |              |              | (b)(1)RW,    |              |
|           |        |              |              |              | (2)OP        |              |
|           |        |              |              |              | (c)(1)RW,    |              |
|           |        |              |              |              | (2)BS        |              |
| BUSY      |        | CACK         |              |              |              |              |
| release   |        | completed    |              |              |              |              |
| (RRDY)    |        | terminals    |              |              |              |              |
| reception |        | cct->CW      |              |              |              |              |

|         |         |         |         |         |         |         |
|---------|---------|---------|---------|---------|---------|---------|
| *1      |         |         |         |         |         |         |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |

\*1: in the case of BUSY packet missing

State Transition Table of Separate Retransmissions to Suspending  
Designated Terminals (Server side)

| State                                                                           | Separate<br>Retrans-<br>mission<br>wait<br>SOP | ACK/NACK<br>wait<br>SAW                                | Conne-<br>tion<br>Release<br>Response<br>wait<br>SRW | Retrans-<br>mission<br>to this<br>terminal<br>completed<br>SCL | Separate<br>retrans-<br>mission<br>phase<br>(to BUSY<br>terminals)<br>BS |
|---------------------------------------------------------------------------------|------------------------------------------------|--------------------------------------------------------|------------------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------------------------|
| Transmission<br>request (ex)<br>(initial)                                       | DT<br>->SAW                                    |                                                        |                                                      |                                                                |                                                                          |
| Transmission<br>request (ex)<br>(following)                                     | DT<br>->SAW                                    |                                                        |                                                      |                                                                |                                                                          |
| ACK<br>reception                                                                |                                                | mct, REL<br>->SRW                                      |                                                      |                                                                |                                                                          |
| NACK<br>reception                                                               |                                                | mct<br>->SOP                                           |                                                      |                                                                |                                                                          |
| BUSY<br>reception                                                               | irr<br>->BS                                    | irr<br>->BS                                            |                                                      |                                                                |                                                                          |
| Time out<br>(ex)                                                                | ccr, irr<br>->SCL                              | (not last)<br>POLL->SAW<br>(last)<br>cct, irr<br>->SCL | irr, cct<br>->SCL                                    |                                                                |                                                                          |
| ABORT reception<br>(from terminals<br>conducting<br>separate<br>retransmission) | irr, cct<br>->SCL                              | irr, cct<br>->SCL                                      | irr, cct<br>->SCL                                    |                                                                |                                                                          |
| RACK<br>reception                                                               |                                                |                                                        | irr, cct<br>->SCL                                    |                                                                |                                                                          |



**3.2.2 State Transition Table (Terminal side)**

|              | CLOSED<br>(CLS)       | CONNECTED<br>(CTD)                                                                                                                                 | RELEASE<br>(REL)              | BUSY                                 | RCV_RDY                                                                                                                                                | SUSPEND<br>(SPD)      | CLS_BSY                   |
|--------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|---------------------------|
| CONN         | Send<br>CACK<br>->CTD | Send<br>CACK<br>->CTD                                                                                                                              | Ignore                        | Ignore                               | Ignore                                                                                                                                                 | Ignore                | CACK<br>(NO)-><br>CLS_BSY |
| DT           | Ignore                | (not BUSY)<br>OK/NGentry<br>Not LAST<br>->CTD<br>LAST&alloK<br>Send ACK<br>->REL<br>LAST &<br>partly NG<br>Send NACK<br>Retrans.<br>CNT++<br>->CTD | Ignore                        | Ignore                               | (BUSY)<br>OK/NGentry<br>Not LAST<br>->RCV_RDY<br>LAST&alloK<br>Send ACK<br>->REL<br>LAST &<br>partly NG<br>Send NACK<br>Retrans.<br>CNT++<br>->RCV_RDY | Ignore                | Ignore                    |
| POLL         | Ignore                | Send NACK<br>->CTD                                                                                                                                 | Send<br>ACK<br>->REL          | Send<br>BUSY                         | Send RRDY<br>->RCV_RDY                                                                                                                                 | Send<br>NACK<br>->CTD | Ignore                    |
| REL          | Ignore                | Ignore                                                                                                                                             | RelProc.<br>SendRACK<br>->CLS | Ignore                               | Ignore                                                                                                                                                 | Ignore                | Ignore                    |
| ABORT        | Ignore                | RelProc.<br>->CLS                                                                                                                                  | RelProc.<br>->CLS             | RelProc.<br>->CLS                    | RelProc.<br>->CLS                                                                                                                                      | RelProc.<br>->CLS     |                           |
| SUSP         | Error<br>->CLS        |                                                                                                                                                    | Send ACK<br>->REL             | Send BUSY<br>->BUSY                  |                                                                                                                                                        | Error<br>->SPD        | Error-><br>CLS_BSY        |
| EX-<br>TMOUT | Never<br>occur        | Send NACK<br>Retrans.<br>CNT++<br>->CTD                                                                                                            | RelProc.<br>->CLS             | Never<br>occur<br>(timer<br>not set) | Send NACK<br>Retrans.<br>CNT++<br>->CTD                                                                                                                | RelProc.<br>->CLS     | Ignore                    |
| EX-<br>BUSY  | ->CLS_<br>BSY         | Send BUSY<br>->BUSY                                                                                                                                | Error<br>->REL                | Ignore                               | Send BUSY<br>->BUSY                                                                                                                                    | Send<br>BUSY          | Ignore                    |
| EX-<br>RRDY  | Error<br>->CLS        | Error<br>->CTD                                                                                                                                     | Error<br>->REL                | Send RRDY<br>->RCV_RDY               | Error<br>->RCV_RDY                                                                                                                                     | Error<br>->SPD        | ->CLS                     |
| EX-          | Error                 | Send                                                                                                                                               | Send                          | Send                                 | Send                                                                                                                                                   | Send                  | Ignore                    |

|         |         |          |          |          |          |         |
|---------|---------|----------|----------|----------|----------|---------|
| ABORT   | ABORT   | ABORT    | ABORT    | ABORT    | ABORT    |         |
|         | ->CLS   | RelProc. | RelProc. | RelProc. | RelProc. |         |
|         |         | ->CLS    | ->CLS    | ->CLS    | ->CLS    |         |
| +-----+ | +-----+ | +-----+  | +-----+  | +-----+  | +-----+  | +-----+ |

### **2.2.6 Optional Information Transfer in Connection Establishment**

#### **(1) Overview**

The server passes information about communication environment to the terminals in order to improve multicast communication performance during the connection establishment phase.

#### **(2) Protocol Sequence**

No additional procedure.

#### **(3) Packet Format**

The following parameter is added to the connection establish request packet CONN:

- RBOT (Range of BackOff Time): the response delay range for the backoff time algorithm in the terminal.  
Each terminal waits a randomly selected time in the range between 0 seconds and the RBOT value before transmitting responses to the server. This applies to sending CACK, ACK and RACK.

### **2.2.7 Abort Request to Individual Terminal**

#### **(1) Overview**

The server can terminate individual terminals' communication operation (Abort).

#### **(2) Protocol Sequence**

The server transmits an ABORT packet via unicast to each terminal to be terminated. Upon receipt the terminals should stop its RMTP communication and terminate.

#### **(Notes)**

Two ABORT functions are now defined in RMTP.

ABORT defined in the Base Part: The server or a terminal sends this ABORT to its party to notify that a server or a terminal is terminating.

ABORT defined in this Extended Part: The server notifies a terminal to terminate its process. The server will continue communication with the rest of the terminals.

#### **(3) Packet Format**

Use ABORT packet by UniCast.



### **3. Packet Format**

#### **3.1 Encoding Rules**

- Each packet has at the front common parameters of packet type, packet length, and connection ID.
- Packet lengths are all explicitly encoded, which show the length from packet type to the very end of the packet.
- User data is at fixed lengths, and padding is to be placed following the end of the message.
- Values indicating sizes, number of times and other parameters are to be encoded as integer toward the right side of the box.
- Reserved field and Option field may be prescribed in the future with the expansion of this protocol. As far as this version of the protocol is concerned, this may be freely used within the scope of the application or such protocol as the application may prescribe.
- Packets to be used in this protocol are to be transferred as user data of lower layer protocols. When UDP is used in lower layer, it is stored in the data octets portion of the UDP User Datagram and transferred.



### 3.2 Packet Formats

Legend UC: transferred by unicast MC: transferred by multicast

(1) Data packet (DT) :

[MC the first transmission and retransmission ],

[UC retransmission and separate retransmission after busy]

A server to terminals.

| 0                                                                | 8              | 16                         | 24 | 31 |
|------------------------------------------------------------------|----------------|----------------------------|----|----|
| +-----+-----+-----+-----+                                        |                |                            |    |    |
| Packet type (1)                                                  |                | Packet length (byte)       |    |    |
| +-----+-----+-----+-----+                                        |                |                            |    |    |
| Connection ID                                                    |                | Packet sequence number     |    |    |
| +-----+-----+-----+-----+                                        |                |                            |    |    |
| Flag                                                             | Retransmission | Message sequence number    |    |    |
|                                                                  | count          | (in large volume delivery) |    |    |
| +-----+-----+-----+-----+                                        |                |                            |    |    |
| User data                                                        |                |                            |    |    |
| (User data will be at fixed lengths, and padding is to be placed |                |                            |    |    |
| following the end of the message.)                               |                |                            |    |    |
| +-----+-----+-----+-----+                                        |                |                            |    |    |

Packet type: integer value begins at 1

Connection ID: integer (the same value as that used in CONN)

Packet sequence numbers: integer value 1 to  $2^{16}-2$

Flag: Separate re-transmission (16th bit is ON)

LAST (17th bit is ON)

Monitoring start (22th bit is ON)

Monitoring stop (23th bit is OFF)

Retransmission count: integer value 0 (initial transmission),

1 (1st time re-transmission), 2, ---

If message sequence number is not set, user data moves up.

(2) ACK : [UC] terminals to a server

Notification from terminals to a server of the reception of  
messages(all data packets)

| 0                               | 8 | 16             | 24       | 31 |
|---------------------------------|---|----------------|----------|----|
| +-----+-----+-----+-----+       |   |                |          |    |
| Packet type (2)                 |   | Packet length  |          |    |
| +-----+-----+-----+-----+       |   |                |          |    |
| Connection ID                   |   | Retransmission | Reserved |    |
|                                 |   | count          |          |    |
| +-----+-----+-----+-----+       |   |                |          |    |
| Message sequence number         |   |                |          |    |
| (only in large volume delivery) |   |                |          |    |
| +-----+-----+-----+-----+       |   |                |          |    |



(3) NACK : [UC] terminals to a server

| 0                                                                      | 8 | 16                         | 24 | 31 |
|------------------------------------------------------------------------|---|----------------------------|----|----|
| +-----+-----+-----+-----+                                              |   |                            |    |    |
| Packet type (3)                                                        |   | Packet length              |    |    |
| +-----+-----+-----+-----+                                              |   |                            |    |    |
| Connection ID                                                          |   | Retransmission             |    |    |
|                                                                        |   | count                      |    |    |
| +-----+-----+-----+-----+                                              |   |                            |    |    |
| Message sequence number                                                |   | Retransmission requirement |    |    |
| (only in large volume delivery)                                        |   |                            |    |    |
| +-----+-----+-----+-----+                                              |   |                            |    |    |
| The sequence number of the packet requiring retransmission(16bit x m), |   |                            |    |    |
| the scope of the sequence numbers (Number 16 bit, scope symbol[16 bit  |   |                            |    |    |
| all 1], Number 16 bit), or any combination of the foregoing.           |   |                            |    |    |
| +-----+-----+-----+-----+                                              |   |                            |    |    |
| Scope of sequence numbers "10 to 15" is encoded "10 scope symbol 15".  |   |                            |    |    |
| If message sequence number is absent, a retransmission requirement     |   |                            |    |    |
| field moves up.                                                        |   |                            |    |    |

(4) BUSY Notification (BUSY) : [UC] a terminal to a server

| 0               | 16 | 31 |
|-----------------|----|----|
| +-----+-----+   |    |    |
| Packet type (4) |    |    |
|                 |    |    |
| +-----+-----+   |    |    |
| Connection ID   |    |    |
|                 |    |    |
| +-----+-----+   |    |    |

(5) BUSY release (RRDY) : [UC] a terminal to a server

| 0                                                            | 16 | 31 |
|--------------------------------------------------------------|----|----|
| +-----+-----+                                                |    |    |
| Packet type (5)                                              |    |    |
|                                                              |    |    |
| +-----+-----+                                                |    |    |
| Connection ID                                                |    |    |
|                                                              |    |    |
| +-----+-----+                                                |    |    |
| The sequence number of the packet requiring re-transmission  |    |    |
| (16 bit x m), the scope of the sequence numbers              |    |    |
| (Number 16 bit, scope symbol [16 bit all 1], Number 16 bit), |    |    |
| or any combination of the foregoing.                         |    |    |
| +-----+-----+                                                |    |    |

(6) ABORT : [MC] a server to terminals, [UC] a terminal to a server

| 0               | 16 | 31 |
|-----------------|----|----|
| +-----+-----+   |    |    |
| Packet type (6) |    |    |
|                 |    |    |
| +-----+-----+   |    |    |
| Connection ID   |    |    |
|                 |    |    |
| +-----+-----+   |    |    |



(7) State inquiry (POLL) : [UC] a server to a terminal

|                     |               |    |
|---------------------|---------------|----|
| <u>0</u>            | 16            | 31 |
| +-----+-----+-----+ |               |    |
| Packet type (7)     | Packet length |    |
| +-----+-----+-----+ |               |    |
| Connection ID       |               |    |
| +-----+-----+-----+ |               |    |

(8) Connection establishment request (CONN) :

[MC the first time], [UC retransmission] a server to terminals

|                                               |                  |          |    |    |
|-----------------------------------------------|------------------|----------|----|----|
| <u>0</u>                                      | 8                | 16       | 24 | 31 |
| +-----+-----+-----+-----+                     |                  |          |    |    |
| Packet type (8)                               | Packet length    |          |    |    |
| +-----+-----+-----+-----+                     |                  |          |    |    |
| Connection ID                                 | Protocol ver.(2) | Reserved |    |    |
| +-----+-----+-----+-----+                     |                  |          |    |    |
| Message size (byte)                           |                  |          |    |    |
| +-----+-----+-----+-----+                     |                  |          |    |    |
| Block number                                  | Block size       |          |    |    |
| +-----+-----+-----+-----+                     |                  |          |    |    |
| Option (Length is variable in units of bites) |                  |          |    |    |
| +-----+-----+-----+-----+                     |                  |          |    |    |

- Protocol ver.: the version number of the protocol
- Message size: the size of data given by the application
- Block number: the number of packets consisting of the message, which equals to the maximum number of sequence numbers
- Block size: the size of user data field in DT, during the session it does not change.
- Message sizes at 32 bit may be designated up to about 4.2 Gbytes, and block sizes at 16 bit up to about 65 Kbytes
- Option field obeys the following formats.

[Format of the option field]

Option parameters

0. Application type (file bulk transfer, etc.)

1. Data name (file name, etc.)

2. Data type

16. Size of an original un-divided message (in large volume delivery)

17. Size of a divided message (in large volume delivery)

18. Sequence number of a divided message (in large volume delivery)

32. Scramble mode (Mode: none, decoded by a protocol, decoded by an application)

33. Message size before scrambled (in scramble delivery)

34. Authentication requirement (requirement of authentication code to terminals)



## 0. Application type

|                                                                        |                  |    |
|------------------------------------------------------------------------|------------------|----|
| <u>0</u>                                                               | 16               | 31 |
| +-----+-----+                                                          |                  |    |
| Parameter type (0)                                                     | Parameter length |    |
| +-----+-----+                                                          |                  |    |
| Application type                                                       |                  |    |
| +-----+                                                                |                  |    |
| Parameter type: integer value, Parameter length: integer value in byte |                  |    |
| Application type: integer value (0: bulk file transfer)                |                  |    |

## 1. Data name

|                                                                        |                  |    |
|------------------------------------------------------------------------|------------------|----|
| <u>0</u>                                                               | 16               | 31 |
| +-----+-----+                                                          |                  |    |
| Parameter type (1)                                                     | Parameter length |    |
| +-----+-----+                                                          |                  |    |
| Data name                                                              |                  |    |
| +-----+                                                                |                  |    |
| Data name: consists of ASCII characters (is up to 256 bytes in length) |                  |    |

## 2. Data type

|                                                               |                  |    |
|---------------------------------------------------------------|------------------|----|
| <u>0</u>                                                      | 16               | 31 |
| +-----+-----+                                                 |                  |    |
| Parameter type (2)                                            | Parameter length |    |
| +-----+-----+                                                 |                  |    |
| Data type                                                     |                  |    |
| +-----+                                                       |                  |    |
| Data type: integer value                                      |                  |    |
| 0:ASCII, 1:PDF, 2:RTF, 3:MHEG, 4:ODA, 8:binary (octet-stream) |                  |    |
| 16:JPEG, 17:MPEG1, 18:MPEG2, 19:MPEG4                         |                  |    |

## 3. Range of backoff time

|                                              |                  |    |
|----------------------------------------------|------------------|----|
| <u>0</u>                                     | 16               | 31 |
| +-----+-----+                                |                  |    |
| Parameter type (3)                           | Parameter length |    |
| +-----+-----+                                |                  |    |
| Range of backoff time                        |                  |    |
| +-----+                                      |                  |    |
| Range of backoff time: integer value in msec |                  |    |

## 16. Size of an un-devided original message (in large volume delivery)

|                                                                                |                  |    |
|--------------------------------------------------------------------------------|------------------|----|
| <u>0</u>                                                                       | 16               | 31 |
| +-----+-----+                                                                  |                  |    |
| Parameter type (16)                                                            | Parameter length |    |
| +-----+-----+                                                                  |                  |    |
| Size of an un-devided original message                                         |                  |    |
| +-----+                                                                        |                  |    |
| Size of an un-devided message: integer value in byte, if unknown, set to all 1 |                  |    |



## 17. Size of a devided message (in large volume delivery)

|                                                  |                  |    |
|--------------------------------------------------|------------------|----|
| <u>0</u>                                         | 16               | 31 |
| +-----+-----+-----+                              |                  |    |
| Parameter type (17)                              | Parameter length |    |
| +-----+-----+-----+                              |                  |    |
| Size of a devided message                        |                  |    |
| +-----+-----+-----+                              |                  |    |
| Size of a devided message: integer value in byte |                  |    |

## 18. Sequence number of a devided message (in large volume delivery)

|                                                               |                  |    |
|---------------------------------------------------------------|------------------|----|
| <u>0</u>                                                      | 16               | 31 |
| +-----+-----+-----+                                           |                  |    |
| Parameter type (18)                                           | Parameter length |    |
| +-----+-----+-----+                                           |                  |    |
| Sequence number of a devided message                          |                  |    |
| +-----+-----+-----+                                           |                  |    |
| Sequence number of a devided message: integer value 1, 2, --- |                  |    |

## 32. Scramble mode

|                                                                                                       |                  |    |
|-------------------------------------------------------------------------------------------------------|------------------|----|
| <u>0</u>                                                                                              | 16               | 31 |
| +-----+-----+-----+                                                                                   |                  |    |
| Parameter type (32)                                                                                   | Parameter length |    |
| +-----+-----+-----+                                                                                   |                  |    |
| Scramble mode                                                                                         |                  |    |
| +-----+-----+-----+                                                                                   |                  |    |
| Scramble mode: integer value, no scrambled 0, decoded by a protocol 1,<br>decoded by an application 2 |                  |    |

## 33. Real message size (in scramble delivery)

|                                          |                  |    |
|------------------------------------------|------------------|----|
| <u>0</u>                                 | 16               | 31 |
| +-----+-----+-----+                      |                  |    |
| Parameter type (33)                      | Parameter length |    |
| +-----+-----+-----+                      |                  |    |
| Real message size                        |                  |    |
| +-----+-----+-----+                      |                  |    |
| Real message size: integer value in byte |                  |    |

## 34. Authentication requirement (requirement of authetication code to terminals)

|                                                                       |                        |    |
|-----------------------------------------------------------------------|------------------------|----|
| <u>0</u>                                                              | 16                     | 31 |
| +-----+-----+-----+                                                   |                        |    |
| Parameter type (34)                                                   | Parameter length       |    |
| +-----+-----+-----+                                                   |                        |    |
| Authentication requirement                                            | Random value (8 Bytes) |    |
| +-----+-----+-----+                                                   |                        |    |
|                                                                       |                        |    |
| +-----+-----+-----+                                                   |                        |    |
| Authentication requirement: integer value, Not required 0, Required 1 |                        |    |

Random value: integer value, 8 bytes in length

Shiroshita, Sano, Takahashi, Yamanouchi

[Page 26]

256 or later. Application specified parameter

|                               |                  |    |
|-------------------------------|------------------|----|
| <u>0</u>                      | 16               | 31 |
| +-----+-----+                 |                  |    |
| Parameter type (X) X: 256 --- | Parameter length |    |
| +-----+-----+                 |                  |    |
| User defined parameter(s)     |                  |    |
|                               |                  |    |
| +-----+-----+                 |                  |    |

User defined parameters: which are used for and specified by the user application. The length and number of parameters, and nesting format are up to the user application. Mutual agreements about these parameters must be needed in advance.

(9) Connection establishment response (CACK):[UC] terminals to a server

|                 |                              |    |
|-----------------|------------------------------|----|
| <u>0</u>        | 16                           | 31 |
| +-----+-----+   |                              |    |
| Packet type (9) | Packet length                |    |
| +-----+-----+   |                              |    |
| Connection ID   | Response                     |    |
|                 | (Yes:integer value 0, No: 1) |    |
| +-----+-----+   |                              |    |
| Option          |                              |    |
|                 |                              |    |
| +-----+-----+   |                              |    |

- Option field obeys the following formats.

[Format of the option field]

0. Receiver name

|                    |                  |    |
|--------------------|------------------|----|
| <u>0</u>           | 16               | 31 |
| +-----+-----+      |                  |    |
| Parameter type (0) | Parameter length |    |
| +-----+-----+      |                  |    |
| Receiver name      |                  |    |
| +-----+-----+      |                  |    |

Receiver name: ASCII/JIS8 (up to 32 Bytes in length, and the length of each terminal name may not be the same)

1. Authentication ID

|                    |                  |    |
|--------------------|------------------|----|
| <u>0</u>           | 16               | 31 |
| +-----+-----+      |                  |    |
| Parameter type (1) | Parameter length |    |
| +-----+-----+      |                  |    |
| Authentication ID  |                  |    |
| +-----+-----+      |                  |    |

Authentication ID: integer value (128 bits)

Authentication ID is calculated by MD5 whose input is "Receiver name

+ random value G (8 Bytes) notified in CONN packet + pre-distributed  
key K (8 Bytes)"

(10) Connection release request (REL) :

[MC after multicast transmission and retransmission],  
[UC after unicast retransmission] a server to terminals

|                  |               |    |
|------------------|---------------|----|
| <u>0</u>         | 16            | 31 |
| +-----+-----+    |               |    |
| Packet type (10) | Packet length |    |
| +-----+-----+    |               |    |
| Connection ID    |               |    |
| +-----+          |               |    |

(11) Connection release response (RACK): [UC] terminals to a server

|                  |               |    |
|------------------|---------------|----|
| <u>0</u>         | 16            | 31 |
| +-----+-----+    |               |    |
| Packet type (11) | Packet length |    |
| +-----+-----+    |               |    |
| Connection ID    |               |    |
| +-----+          |               |    |

(12) Flow control measurement report packet (FREPO): [UC] terminals to a server

|                                   |                                    |    |
|-----------------------------------|------------------------------------|----|
| <u>0</u>                          | 16                                 | 31 |
| +-----+-----+                     |                                    |    |
| Packet type (12)                  | Packet length                      |    |
| +-----+-----+                     |                                    |    |
| Connection ID                     | Sequence number of DT[st]          |    |
| +-----+-----+                     |                                    |    |
| Sequence number of DT[en]         | The number of receipt DTs (4bytes) |    |
| +-----+-----+                     |                                    |    |
| The number of receipt DTs(contnd) |                                    |    |
| +-----+                           |                                    |    |

Sequence number of DT[st]: The sequece number of DT with monitoring start bit ON, integer value

Sequence number of DT[en]: The sequece number of DT with monitoring start bit OFF, integer value

The number of receipt DTs: The amount of receipt DTs from DT[st] reception to DT[en] reception, integer value

(14) Suspend notification packet (SUSP): [UC] a server to terminals

|                  |               |    |
|------------------|---------------|----|
| <u>0</u>         | 16            | 31 |
| +-----+-----+    |               |    |
| Packet type (14) | Packet length |    |
| +-----+-----+    |               |    |
| Connection ID    |               |    |
| +-----+          |               |    |



## References

- [SS96] Shiroshita, T., Sano, T., Takahashi, O., Yamashita, M., Yamanouchi, N., and Kushida, T., Performance evaluation of reliable multicast transport protocol for large-scale delivery, IFIP Fifth International Workshop on Protocols for High-Speed Networks (PfHSN'96), pp. 149-164, Oct. 1996.
- [ST96] Shiroshita, T., Takahashi, O., and Masahide, Y., Integrating Layered Security into Reliable Multicast, Proceedings of the 3rd International Workshop on Protocols for Multimedia Systems (PROMS'96), Oct. 1996.
- [SS97] Sano, T., Shiroshita, T., Takahashi, O., Yamashita, M., Monitoring-based Flow Controls for Reliable Multicast Protocols and its Evaluation, IEEE International Performance, Computing and Communication Conference (IPCCC'97), Feb. 1997.
- [SY97] Sano, T., Yamanouchi, N., Shiroshita, T., Takahashi, O., Flow and Congestion Control for Bulk Reliable Multicast Protocols, submitted to IEEE Infocomm'98, August 1997.

The above papers and related publications are available at  
<http://isserv.tas.ntt.co.jp/chisho/rmttp/rmtpprefe.htm> or  
<http://www.trl.ibm.co.jp/rmttp/rmtpprefe.htm>.

## Author's Addresses

Teruji Shiroshita, Tetsuo Sano, Osamu Takahashi  
{siro, sano, osamu}@isl.ntt.co.jp  
NTT Information and Communication Systems Laboratories  
1-1 Hikarinooka, Yokosuka, Kanagawa 239 Japan

Nagatsugu Yamanouchi  
yamanouc@trl.ibm.co.jp  
IBM Research, Tokyo Research Laboratory  
IBM Japan  
1623-14 Shimotsuruma, Yamato, Kanagawa 242 Japan