INTERNET-DRAFT
<draft-shiroshita-rmtp-spec-01.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

Status of this Memo

Abstract

This draft presents the specifications of the 'Reliable Multicast Transport Protocol,' which is used for information delivery such as newspaper delivery and software updates. The protocol aims to realize a full reliable multicast of bulk data from a server to thousands of receivers. Scalability of the protocol, flow/congestion control extension, and security issues are described as an addendum for the protocol specifications. This document defines RMTP version 1 specification and also constitutes a Part 1 of RMTP version 2 specification.

Table of Contents

1. Introduction

1.1 The Scenario of Distribution Type Services

Distribution of identical information over networks (hereafter, "Distribution Type Services") is a new form of communication service which shows significant promise. Specifically, the following types of services are expected.

a) Electronic newspaper delivery
Currently, newspapers are printed up late at night at several district centers, delivered by trucks to terminal sales units, and distributed to each subscriber by paper boys. Electronic delivery will substantially reduce the costs of printing, shipping and delivery. Furthermore, since there is a substantial amount of manpower involved in performing these traditional tasks, a tremendous savings in payroll expense can be realized.

b) Software delivery (including audio-visual items)
Already, electronic delivery is being attempted for certain items of computer software. There are examples of deliveries to the end user, and others to the sales outlets that have the necessary equipment to transcribe the contents onto CDROM and other media. The latter service can be expanded to cover audio and video, thus replacing current CD and video rentals. CD rental stores and convenience stores can set up terminal equipment with memory devices to which popular tunes can be delivered once every day. Customers can select the tunes desired and have a CD manufactured on the spot.

The main features of the delivery type service are:
a) Delivery of data can be handled more efficiently than through repetitious point-to-point communication; efficiency is achieved by multicast data transfer.
b) Error-free data delivery is ensured.
c) The service assumes charges, hence the data needs to be delivered without fail, and when delivery is not completed, a mechanism of recording delivery failure is necessary. And
d) In the event communication carriers are to furnish this distribution type service, a specific structure may be embedded within the carrier's network, and it must be possible to optimize the location of this structure.

To date, the technology needed to realize this reliable delivery type service has not been fully developed. Specifically, no multicast protocols endowed with adequate reliability related to data transmission within the environment of an extremely large number of terminals have been reported. This hinders charging for data delivery services.

The "Reliable Multicast Transport Protocol" (hereafter referred to
Protocol) that is provided for by these specifications is designed to
realize the "Distribution Type Service" just discussed.

Here we pick up a wide newspaper delivery as an example to illustrate how information distribution will be used. The pages are prepared at the head office, and sent to the servers at each district. Next, the server in each district distributes the data to terminals in each separate household. We can use RMTP as the delivery mechanism for server to terminal distribution. The number of terminals supported by one server will be about 5,000 in one example environment in Japan. Terminals are either dedicated terminals or PC based; selection would be based mainly on price and ease of use.

A rough estimate of the distributed data volume, considering the volume of data present in existing newspapers (including images) is about 2M Bytes The data is distributed daily by the Protocol. The maximum time allowable for data delivery from one server to all supported terminals is one hour.

The terminal stores the incoming data on magnetic discs and other forms of memory devices and awaits user readout. The user can conduct searches and other actions through the terminal on the delivered news.

Contents of the data distributed, based on current newspaper style would, we assume, include the texts of articles, headings, still pictures (photographs, diagrams partially in color). The data of these contents would be encoded by standard data/document formats (e.g. SGML, HTML, MHEG, ODA).

1.2 Considerations on Networks to be Used

The delivery system used would be for either:
a) Delivery directed at general users using the public switched network, or, b) Delivery directed at corporate users using dedicated LANs.

Technically, in the case of (a), this would involve one-to-one communication with the terminals connected to the public switched network, while for (b), it would be a distribution via Ethernet and other such physical networks making one-to-many communication possible.

The information distribution capability of the network itself (Broadcast: one-to-all data transfer; to all terminals connected to the network, and Multicast: one-to-many (specific) data transfer; to terminal groups) is the mechanism that supports realization of the distribution type system. Shared physical media, such as the Ethernet, Token Ring and others, in which the connected terminals share transmission channels and support the one-to-many transmission function at the same cost as one-to-one communication. In contrast, physical media based on one-to-one communication requires the addition of a data duplication capability at some point within the network.

Regarding (a) in particular, neither existing analog telephone lines nor N-ISDN provide an effective one-to-many mechanism. To realize multicast capability, data duplication at the transmission side, such as "repeating the call by telephone", or "calling multiple terminals simultaneously" is required. Also a device for connecting multiple lines to the server or router is required.

1.3 Quality of Networks and Terminals

The networks to be used for RMTP may consists of N-ISDN, B-ISDN, dedicated lines, and various types of LANs connected to each other. The protocol mechanism design assumes the following network quality and terminal performance.

Data transfer reliability is essential to charged data access services, and RMTP protocol provides re-transmission service to achieve the reliability. The re-transmission mechanism design assumes the quality of digital networks in the physical layer. Furthermore, it assumes unreliable networks such as the networks interconnected by Internet Protocol.

Requirements to the terminals for protocol design in general includes sufficient protocol processing speed and sufficient write access speed to the data memory/storage. In addition, the information delivery to large number of terminals require consideration of the possibility of data reception performance deterioration in a part of terminals due to heavy use of the terminals by other jobs.

2. Communication Model

(1) Outline

This specification defines RMTP (Reliable Multicast Transport Protocol) which realizes reliable, connection-oriented multicast communication between the server and multiple terminals.

At lower layers, an end-to-end datagram service without connections is assumed. Copying data packets, delivery to terminal groups, and detection of bit errors (check sum) are conducted at lower layers. As specific lower layer protocols, UDP and IP (IP multicast, hereafter IP-M) may be used.

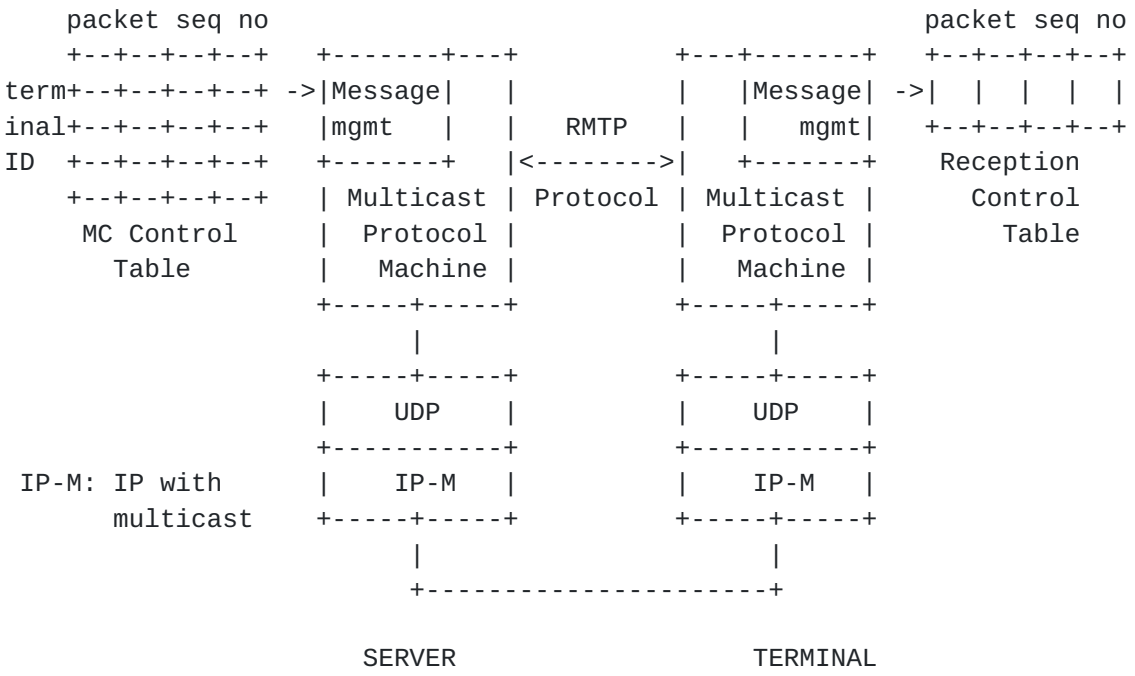
As for the network, there are no restrictions on the type and the form of LAN, WAN or others used. Multicast functions corresponding to IP-M are expected to be implemented (e.g. by the router).

The RMTP uses the UDP and IP-M to realize multicasting in one

direction from the server to multiple terminals. Reliable, bulk transfer type, multicast communication is realized by exercising

connection control, recovering all errors by re-transmission, and confirming the conclusion of reception.

This version of the specification limits the number of connections set up simultaneously to one, and does not distinguish multiple Applications(APs) on top of RMTP. Moreover, delivery of application data is conducted once with a single connection.



Communication Model

(2) Addressing

Addresses of the communication entities, the server and terminals, are to be designated by the TSAP address = NSAP address (IP address) + Transport Extension (Port Number). The IP address is designated in the IP packet, and the port number is designated in the UDP packet at the lower layers. Thus, there is no need to include addresses in the packets of this protocol.

Multicast delivery to a groups is realized by the layers below the Network layer. In the case of IP, delivery to the group is conducted by the IP multicasting with IP class D address for group identification.

Values of the IP class D address are not specified in this specification.

(3) Connection Identification

Multiple data deliveries from the identical sending entity to the same destionation group are identified by connections. A connection is identified by a connection ID

(4) Function Module

Examples of function modules realizing RMTP are shown in the following.

Multicast PM (Protocol machine):
Conducts RMTP multicast and unicast protocol processing. On the server side, based on the control table for message control, transmission of packetized messages and communication control of re-transmission of the same are conducted through UDP/IP-M. On the terminal side, based on the reception control table for message control, NACK/ACK/BUSY notifications are sent to the server and reception control is conducted.

Message Control: Manages the partitioning of mail, files and other application data (hereafter messages) received from AP and the re-transmission in units of packets. On the server side, messages are divided into packets, a control table is set up, and transmission to multicast PM is requested. The control table controls data transmission in units of packets and of terminals, and generates re-transmission messages by taking the OR of packet sequence numbers as reported from each terminal by NACK.

On the terminal side, reception control in units of packets is conducted by means of the reception control table. Based on this table, NACK/ACK/BUSY/RRDY (Receive Ready) packets are generated.

(5) Definition of Functions

The following communication functions are provided by RMTP.

Connection Establishment:

>From the server, to the multiple "n" terminals, 1:n connections are established by multicasting the connection establishment packet. At the time of establishing connections, confirmation of message size, block size and the number of blocks is conducted between the server and terminals. Connection is established with the terminal group corresponding to the group list given by AP.

Data Transfer:

Data packet transfer is conducted in a single direction from the server

to terminals. The server divides messages into blocks and transmits them as multiple packets. Initial transfer is provided by multicast;

subsequent re-transmissions continue using either multicast or unicast. For BUSY terminals, following a series of re-transmissions, a separate re-transmission is conducted. The data packet includes a sequence number and a re-transmission numbers (number of times) controlling subsequent re-transmission.

Delivery confirmation:

After all packets have been received without error, the terminal returns ACK once to the server. If any packets are missing or corrupted, a NACK including the sequence number of the packets to be re-transmitted is returned once.

Error detection:

Checksum in the UDP option is used for error detection and is not conducted by RMTP itself. RMTP is to confirm packet loss and to re-order packets according to the sequence numbers.

Re-transmission:

Following data transfer, re-transmission is conducted to the terminal(s) >from which NACK was sent.

- Re-transmission is conducted by multicast or unicast (selection may be determined by the ratio of NACK terminals to all connected terminals).
- The server retransmits the packets requested by one or more terminals.
- To determine the packet to be retransmitted, the server takes the OR of the failed packet information of all NACKs.
- Re-transmission continues until no terminal sends a NACK(Maximum number of re-transmissions will be restricted).
- During multicast re-transmission, the group remains unchanged (with the same IP class D address).
- When the terminal receives a data packet, it ignores it if already correctly received.

Separate re-transmission phase after BUSY:

>From the terminal, a request for temporary stoppage of reception is given (BUSY notification) to the server, and after BUSY is cleared, the server re-transmits the missing packets separately by unicast. However, the separate re-transmissions following BUSY are to be conducted after normal re-transmission is completed.

Flow Control:

In this version of the specification, setting the transfer speed at the server is a local matter.
This specification defines a primitive flow control scheme between a server and terminals, which identifies those terminals incapable of

receiving data by BUSY packets.

Connection release:

After sending or re-transmission, terminals returning ACK (successful) are released by explicit connection release just after the data is successfully received. Releases after multicast sending and multicast re-transmission are executed by the transfer of a connection release request packet. There are implicit releases (ABORT or TIMEOUT) unaccompanied by a connecion release request packet.

ABORT: Either the server or the terminal may discontinue the communication unilaterally by sending ABORT request packet.

Servers can request discontinuance to all terminals by multicasting ABORT packet. Immediately following transmission, the resources that are occupied for the communication are released and all delivery attempts are regarded as failures. Terminals can request discontinuance to the server by unicasting ABORT packet. Servers regard this terminal connection as incomplete and do not conduct re-transmission.

3. State Transition

3.1 State and event definition

3.1.1 State and event definition (Server side)

On the server side, multicast and subsequent re-transmissions are regarded as major multicast transitions. Separate re-transmissions to terminals after BUSY, are to be defined as separate transitions with due consideration of the state of each terminal.

(1) Types of main multicast transition state

CLOSED: State prior to request for connection establishment

Connection establishment response wait: Following request for connection establishment, the state of standing by for response from terminals.

OPEN: Following connection establishment, the state of standing by for communication request (initial or re- transmissions) coming from sources other than the server protocol machine.

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

Connection release response standby: Following request for connection release, the state of standing by for response from the terminal.

Separate re-transmission to BUSY terminal: Following retransmission

of multicast data, the phase in which re-transmission for each separate terminal is being conducted by unicast to BUSY terminals.

State transition by separate re-transmission to each terminal is shown in (2) to follow.

(2) State transition of separate re-transmission

The server contemplates the separate state transitions for each terminal that has notified BUSY and defines the state of each.

BUSY release(RRDY) wait: Following BUSY reception, the state of waiting for BUSY release.

Separate re-transmission wait: Following reception of BUSY release(RRDY), the state of waiting for separate re- transmission request in the server to the said terminal (waiting for multicast re-transmission or separate re-transmission to other terminals)

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

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

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

Assumption in the server side

(1) Tables
(1a) Connection control table ((CC table)
For each terminal in the list given by AP, this table manages the states of as connection established, not-established, release completed (succeeded or not succeeded), BUSY, RRDY(Receive Ready) classification.

(1b) Multicast control table (MC table)
This table manages the success or failure of data packet reception for each terminal and each packet. The state transition of separate re-transmission following BUSY shares the MC table of multicast main transition.

For each terminal: Classification of non-response, ACK, NACK, BUSY, RRDY, connection release conclusion.

For each cell of the MC table formed by the number of terminals by the number of packet sequence block numbers:
Achievement/non-achievement of packet reception.

(1c) Re-transmission counter
This is a counter for setting the number of times data packets, ACK,

NACK have been retransmitted.
In the case of re-transmission by multicasting, one re- transmission counter is to be set up for the entire system, and in the case of

re-transmission by unicasting and separate re-transmission to BUSY terminals, one such counter is to be set up for each re-transmission. In main state transitions, initial value is set to 0, and values are incremented each time protocol processing is moved to the OPEN state by a re-transmission. Regarding the state transitions of separate re-transmissions following BUSY, initial value is set to 0, and values are incremented each time protocol processing is moved to the re-transmission standby state by repeated re-transmissions.

(1d) Other counters
Re-transmission counter for requesting connection establishment, and re-transmission counter for inquiries (POLL) are to be devised to control CONN retransmission and POLL retransmission.

(2) Timers
(2a) Connection establishment response timer
The server is to start this timer upon transmitting the connection establishment requesting packet (CONN) and release the same upon reception of the connection establishment response packet from all terminals. (A CONN re-transmission number counter is used)

(2b) ACK, NACK timer
The server is to start this timer upon completing transmission or re-transmission of all data packets, and release the same upon reception of ACK, NACK or BUSY notification packets from all terminals. When a timeout occurs, a state inquiry packet(POLL) is transmitted and the timer re-started (a POLL re-transmission number counter is used for each timeout terminal).

(2c) BUSY release timer
The server is to start this timer upon receiving the BUSY notification packet and release the same upon receiving from the said terminals a BUSY release notification packet. When a timeout occurs, a state inquiry packet is sent the terminal and the timer is re-started (POLL re-transmission frequency counters are used for every timeout terminal).

(2d) Connection release timer
The server is to start this timer upon receiving the connection release request packet and release the same upon receiving a release response packet from all terminals targeted for release action. (This timer is a one-time item and no counters need be installed).

(3) Schedulers
In the case that unicast re-transmissions are to be conducted to multiple terminals (including separate re-transmission following BUSY), the scheduler controls the order in which re-transmissions are to be started from each terminal.

Event Definition: Main Transitions of Multicast on the Server Side
(except connection establishment and release)

Event	Previous state	Action
Transmission request (ex)	OPEN(MC table set, automatic start-up)	Begin sending of data packet (DT). After sending 1 unit DT, proceed to ACK/NACK wait state. Send all DT packets.
ACK reception	ACK/NACK wait	Enter into MC and CC tables. (1) If some responses not received, continue ACK/NACK wait state. (2) All responses received, REL transmission, proceed to Connection Release Response wait state.
NACK reception	ACK/NACK wait	Enter into MC and CC tables. (1) If some responses not received, continue ACK/NACK wait state. (2) All responses received, if all NACK, proceed to OPEN state. (3) All responses received, if there is ACK, REL transmission, proceed to Connection Release Response wait.
BUSY reception	ACK/NACK wait	Enter into MC and CC tables. (1) If some responses not received, continue ACK/NACK wait state. (2) All responses received, if NACK and BUSY, proceed to OPEN state. (3) All responses received, if there is ACK, REL transmission, proceed to Connection Release Response wait state (4) If all BUSY, proceed to Separate Retransmission phase to "BUSY terminals" state.
	OPEN	Enter into MC and CC tables. (1) If there are terminals not BUSY, continue OPEN state. (2) If all terminals BUSY, proceed to Separate Retransmissions Phase to "BUSY terminals" state.

continue to the next page

Event Definitions: Main Transitions of Multicast on the Server Side
(except connection establishment and release) (cont'nd)

Event	Previous state	Action
TIMEOUT (ex)	ACK/NACK wait (1) NACK reception (2) No NACK reception, there is BUSY reception (3) No NACK nor BUSY reception	Send state inquiry POLL and proceed to ACK/NACK wait state. Following POLL frequency limits,enter into MC table, evaluate terminal as being equivalent of connection unestablished and enter into CC table. (1) Release ACK terminal, proceed to OPEN state. (2) Release ACK terminal, proceed to Separate Retransmissions to "BUSY terminals" state. (3) Release connection for all terminals, proceed to CLOSED state.
All separate re-transmissions to BUSY terminals completed (ex)	In midst of retransmission to BUSY terminals	Release all resources, report to AP and proceed to CLOSED state.
Request ABORT transmission (ex)	Any time after connection establishment	Report results to AP and proceed to CLOSED state.
ABORT reception	Any time after connection establishment	Enter into MC and CC tables and release resources of terminal sending ABORT message.OPEN states to continue remaining OPEN, at ACK/NACK wait, same transfer as ACK reception.
BUSY Cancel,(RRDY) reception, (From terminal not receiving BUSY)	ACK/NACK wait	Enter into CC table, Same as BUSY reception (ACK/NACK wait)
	OPEN	Enter into CC table, Same as BUSY reception.

In the event definitions, if undefined packets or parameters are received by a server or a receiver, they are neglected.

Event Definition: Main Transitions of Multicast on the Server Side (Connection Establishment)

Event	Previous state	Action
Request for connection establishment (ex)	CLOSED	Transmit establishment packet (CONN) twice and proceed to Connection Establishment Response wait state.
Reception of connection establishment response (CACK)	Connection Establishment Response wait	(1) With terminals lacking response reception, continue Waiting for Connection Establishment Response wait state. (2) With responses from all terminals, connections are established for terminals with affirmative response, proceed to OPEN state (3) With responses from all terminals, no affirmative response nor BUSY, proceed to CLOSED state. (4) With responses from all terminals, no affirmative response but with BUSY, proceed to Separate Retransmission Phase to "BUSY terminals" state.
Reception of BUSY (from terminals concluding affirmative establishment response)	Connection Establishment Response wait	Enter into CC table, continue Connection Establishment Response wait state.
Reception of BUSY release (in the event of BUSY loss)	Connection Establishment Response wait	Enter into CC table, continue Connection Establishment Response Response wait.

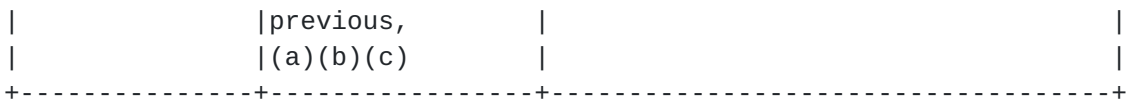
continue to the next page

Event Definition: Main Transitions of Multicast on the Server Side (Connection Establishment) (cont'd)

Event	Previous state	Action
TIMEOUT (ex)	Connection Establishment Response wait	(a) Transmit CONN by unicast to terminals without response (CONN retransmission) After reaching frequency limit of CONN retransmission, proceed as follows. (1) Connection is established for terminals with affirmative response only, enter into CC table, proceed to OPEN state. (2) With no response from any terminal, proceed to CLOSED state. (3) With no affirmative response and nothing but BUSY, proceed to Separate Retransmission Phase to "BUSY terminals" state.
Request ABORT transmission (ex)	Connection Establishment Response wait	Transmit ABORT, release all resources, Report results to AP, proceed to CLOSED state.
ABORT reception (from terminals concluding affirmative response)	Connection Establishment Response wait	Enter into CC table, with terminals lacking response reception, continue Connection Establishment Response wait state.

Event Definition: Server Side Multicast Main Transitions
(Connection Release)

Event	Previous state	Action
Connection release response (RACK) reception	Connection Release Response wait One state before previous, (a) NACK & BUSY absent (b) NACK present (c) NACK absent & BUSY present	CC table entry (a)(1) If there is any non-reception response, continue Connection Release Response wait state. (2) If responses are received from all terminals, release all resources, report results to AP and proceed to CLOSED state. (b)(1) If there is any non-reception response, continue Connection Release Response wait state. (2) If responses are received from all terminals, proceed to OPEN state. (Retransmission counter++) (c)(1) If there is any non-reception response, continue Connection Release Response wait state. (2) If responses are received from all terminals, proceed to Separate Retransmission Phase to "BUSY terminals" state.
TIMEOUT (ex)	Connection Release Response One state before previous, (a)(b)(c)	Release connection, CC table entry (a) CLOSED (b) OPEN state (Retransmissioncounter++) (c) proceed to Separate Retransmission Phase
ABORT transmission request (ex)	Connection Release Response wait .	ABORT transmission, release all resources, proceed to CLOSED state.
ABORT reception	Connection Release Response wait . One state before	Same action and transition connection release response reception.


|previous,
|(a)(b)(c)

Event Definition: State Transition of Separate retransmission phase to BUSY Terminals

Event	Previous state	Action
BUSY release (RRDY) reception	BUSY release wait	Confirm MC table corresponding to this terminal and proceed to Separate Retransmission state wait .
Transmission request (separate) (ex)	Separate Retransmission wait	Transmit retransmission data, and ACK/NACK state wait. (separate)
ACK reception	ACK/NACK wait	Transmit 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.
BUSY reception	ACK/NACK wait	Release resources of separate retransmission to this terminal. Proceed to Busy release wait state.
	Separate Retransmission wait	Release resources of separate retransmission to this terminal. Proceed to Busy release wait state.

continue to the next page

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

Event	Previous state	Action
TIMEOUT (ex)	ACK/NACK wait	Conduct inquiry of state 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	Conduct inquiry of state 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 (from terminals conducting individual 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.

3.1.2 Definition of States and Events (Terminal side)

(1)State definition: Terminal side

CLOSED	State without connection. (Not established/following release)
CONNECTED	Connection in place, state of capability of data reception.
RELEASE	State of undergoing processing of connection release
BUSY	State of difficulty in data reception due to circumstances on the terminal side.
RCV_RDY	State wherein BUSY state has been released but separate retransmission (BUSY retransmission) has not yet occurred. If separate reestablishment occurs, there will be transfer to a CONNECTED state.
CLS_BSY	A CLOSED state wherein a BUSY external event has occurred CONN packet is not accepted. If a release event should occur, a reversal to CLOSED state would take place.

(2)Tables

(2a) Data receive control table (RCV table)

For each data packet, classification of either "Normal reception completed" or "Not yet received" are to be noted.

(2b) Re-transmission counter

This counts the number of times of re-transmission. After receiving DT packet in a CONNECTED state, and when NACK is to be transmitted due to abnormal or lack of reception, it is incremented. It is reset when CONNECTION is released.

(3)Timer

(3a) Timer on data wait

The client following establishment of CONNECTION, proceeds to a CONNECTED state and awaits data arrival in control of the RCV table. In the event data packet (DT) is lost, the state at the RCV table is not one complying with "all data received" so the client waits with the

timer.

When proceeded to a CONNECTED , or when DT packet is received but the state of "everything in place" has not yet been achieved, the timer is set. When "all data received" has been achieved, the timer is stopped. When the following DT packet has arrived, the timer is reset once, and when the state of "all data received" has not yet been achieved, the timer is set once more. When a TIMEOUT occurs, all packets not yet received are classified as "errors" (lost). NACK is transmitted and the timer is released. In this case, the CONNECTED state is continued.
In the case of DT transmission in a state of RCV_RDY, the timer is established in the same manner as the foregoing. Note that when a state of RCV_RDY has been assumed, as DT re-transmission does not follows immediately, the timer should be set for a long time or not used at all.

(3b) Timer on REL wait

Upon receiving all data and sending ACK, the client proceeds to a RELEASE state awaiting a REL packet. When the REL packet has been lost, the connection cannot be released so the client waits the REL with the timer.
When proceeded to REL state, the timer is set. This is released upon arrival of the REL packet.
When timeout occurs, it is deemed that the REL packet has been lost. Connection release is processed and proceeded to a CLOSED state.

State Definition: Terminal side

Event	Previous state	Action
CONN	CLOSED	CONN packet is received, and proceed to CONNECTED state. Prepare a data reception control table.
DT	CONNECTED	DT packet is received. BUSY retransmission flag of DT not to be on. Result (reception completed) to be recorded in RCV table. When Dts are completely assembled, return ACK.
	RCV/RDY	DT packet is received. BUSY retransmission flag of DT not to be on. Result (reception completed) to be recorded in RCV table. When Dts are completely assembled, return ACK.

continue to the next page

State Definition: Terminal side (cont'd)

Event	Previous state	Action
POLL	CONNECTED	POLL packet is received. There should be data not yet received so NACK transmission.
	RELEASE	POLL packet is received. There should be no data not yet received so ACK transmission.
	BUSY	POLL packet is received. BUSY transmission to notify BUSY state.
	RCV_RDY	POLL packet is received. RRDY transmission to notify RCV_RDY state.
REL	RELEASE	REL packet is received. After release processing, RACK transmission proceeding to CLOSED state.
ABORT	All states	ABORT packet is received. After release processing, proceed to CLOSED state.
EX TMOUT	CONNECTED	Timeout. DTs are incomplete(DT packet missing). NACK transmission. Retransmission counter++. Release processing to CLOSED state.
	RELEASE	Timeout. REL packet missing. Release processing to CLOSED state.
EX-BUSY	CLOSED	BUSY occurred. To a CLS_BSY state. (Busy state form CLOSED)
	CONNECTED	BUSY occurred. BUSY transmission, and proceed to BUSY state.
	RCV_RDY	BUSY occurred. (BUSY released once, but reverted to BUSY again). Busy transmission, and proceed to BUSY state.
EX-RRDY	BUSY	BUSY release occurred. RRDY transmission to RCV_RDY state.
EX-ABORT	All states	ABORT occurred. ABORT transmission to release connection, and proceed to CLOSED state.

3.2 State Transition Table

3.2.1 State Transition Table (Server side)

When various event under the vertical axis have occurred under the varying states indicated on the horizontal axis, protocol processing and state transition take place at the cells at which the rows and columns intersect.

Symbol for state:
(ex) A event occurring originally on server other than packet reception from the terminal.

Notation of cell entries:
(Action) -> (transit destination state)

Abbreviations within the cells:
Packet abbreviation, see chapter on protocol format.
For classification of (1)-(4), (a)-(c), see Event Definition Table.

CONN: Send connection establishment packet
REL : Send connection release request packet
DT : Send DT packet
POLL: Send POLL packet
AB : Send ABORT packet
cct : Confirm and record CC table
mct : Confirm and record MC table
ce : Connection establishment
cr : Connection release
rr : Release server communication resource (tables, buffers)
irr : Release of separate re-transmission resource (buffers)
ap : Report to AP communications results (CC table)
error: Raise error notification to local within the server. No change in state.

Event of blank cells are to be ignored and the state to be continued.

The station transition of protocol processing on the server side is expressed in terms of (1) multicast main transition table and (2) state transition table of separate re-transmission to BUSY terminals. Examples of the transitions between the two types of transition tables are shown in the following.

(1) Transition from main transition to separate re-transmission

Following multicast re-transmission, all terminals other than those returning BUSY notification are to release connections after concluding re-transmission (all ACK except BUSY) and when all RACK indicators have returned ((c) (2)BS of the cell [RACK reception] in [Connection Release Response wait (RW)] of main transition table), proceed to the state transition of separate re-transmission. In the main transition table, the BUSY separate re-transmission phase (BS) is to be continued.

(2) Data transfer at separate re-transmission

When proceeded to separate re-transmission, if there are terminals that have had BUSY canceled, those terminals are on standby for separate re-transmission (BOP) and when a transfer request is issued, separate re-transmission is started. When separate re-transmission for any single terminal has been completed, separate re-transmission directed at the said terminal has been completed (BCL). The order for multiple separate re-transmissions is controlled by the scheduler and separate transmission is conducted in serial, or in a parallel.

(3) From separate re-transmission transition to main transition

When all separate re-transmissions have been completed, all transitions are completed by returning to the main state transition table. (In BUSY separate re- transmission phase (BS) of the main transition table, BUSY separate re-transmission completed.

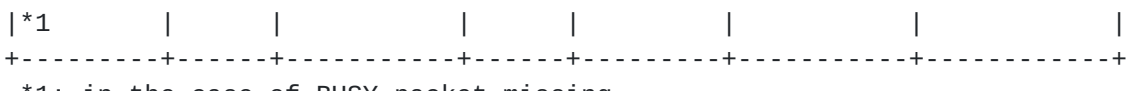
Multicast Main Transition Table (Server side)

State / Event	CLOSED CL	Connection Establishment Response wait	OPEN OP	ACK/NACK wait AW	Connection Release Response wait RW	Separate Retransmission Phase BS
Transmission request (ex)	error	error	DT ->AW	error	error	error
ACK reception				mct, cct, (2)REL-> (1)AW, (2)RW		
NACK reception				mct,cct, (3)REL ->(1)AW, (2)OP, (3)RW		
BUSY reception		CACK completed terminals cct->CW	cct -> (1)OP (2)BS	cct, (3)REL ->(1)AW, (2)OP, (3)RW, (4)BS		
Time out (ex)		(not last) (a)->CONN (last)ccp, (1)ce-> (1)OP, (2)CL,(3)BS		(not last) POLL->AW (last) mct,cct, cr,-> (1)OP, (2)BS, (3)CL	ce, cct ->(a)CL, (b)OP, (c)BS	
BUSY separate retransmission completed (ex)						rr, ap ->CL

continue to the next page

Multicast Main Transition Table (Server side) (contn'd)

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



*1: in the case of BUSY packet missing

State Transition Table of Separate Re-transmissions to BUSY Terminals (Server side)

State / Event	Busy Release wait RRW	Separate Retrans-mission wait BOP	ACK/NACK wait BAW	Connec-tion Release Response wait BRW	Retrans-mission to this terminal completed BCL
RRDY reception	cct,mct confirm ->BOP				
Transmission request (ex)	error	DT ->BAW	error	error	error
ACK reception			mct,REL ->BRW		
NACK reception			mct ->BOP		
BUSY reception		irr ->RRW	irr ->RRW		
Time out (ex)	(not last) POLL->RRW (last) cct,irr ->BCL		(not last) POLL->BAW (last) cct,irr ->BCL	irr,cct ->BCL	
ABORT reception (from terminals conducting separate retransmission)	irr,cct ->BCL	irr,cct ->BCL	irr,cct ->BCL	irr,cct ->BCL	
RACK reception				irr,cct	

3.2.2 State Transition Table (Terminal side)

	CLOSED	CONNECTED	RELEASE	BUSY	RCV_RDY	CLS_BSY
CONN	Send CACK ->CONN'ED	Send CACK ->CONN'ED	Ignore	Ignore	Ignore	CACK(NO) ->CLS_BSY
DT	Ignore	(not BUSY) OK/NG entry Not LAST ->CONN'ED LAST&allOK Send ACK ->RELEASE LAST & partly NG Send NACK Retrans. CNT++ ->CONN'ED	Ignore	Ignore	(BUSY) OK/NG entry Not LAST ->RCV_RDY LAST&allOK Send ACK ->RELEASE LAST & partly NG Send NACK Retrans. CNT++ ->RCV_RDY	Ignore
POLL	Ignore	Send NACK ->CONN'ED	Send ACK ->RELEASE	Send BUSY	Send RRDY ->RCV_RDY	Ignore
REL	Ignore	Ignore	RelProc. Send RACK ->CLOSED	Ignore	Ignore	Ignore
ABORT	Ignore	RelProc. ->CLOSED	RelProc. ->CLOSED	RelProc. ->CLOSED	RelProc. ->CLOSED	Ignore
EX-TMOUT	Never occur	Send NACK Retrans. CNT++ ->CONN'ED	Release process ->CLOSED	Never occur (timer is not set)	Send NACK Retrans. CNT++ ->CONN'ED	Ignore
EX-BUSY	->CLS_BSY	Send BUSY ->BUSY	Error ->RELEASE	Ignore	Send BUSY ->BUSY	Ignore
EX-RRDY	Error ->CLOSED	Error ->CONN'ED	Error ->RELEASE	Send RRDY ->RCV_RDY	Error ->RCV_RDY	->CLOSED
EX-ABORT	Error ->CLOSED	Send ABORT RelProc. ->CLOSED	Send ABORT RelProc. ->CLOSED	SendABORT RelProc. ->CLOSED	Send ABORT RelProc. ->CLOSED	Ignore

"LAST": All DT packets which the terminals are to receive in that pocess

of transmission/retransmission have been received. The DT packet with the LAST flag "on" does not necessarily mean this is the last.

"RelProc": Release all communication resources for that connection.

3.3 Communication Sequence Examples

(This section contains only referential information to assist in understanding definitions in 3.1 and 3.2.)

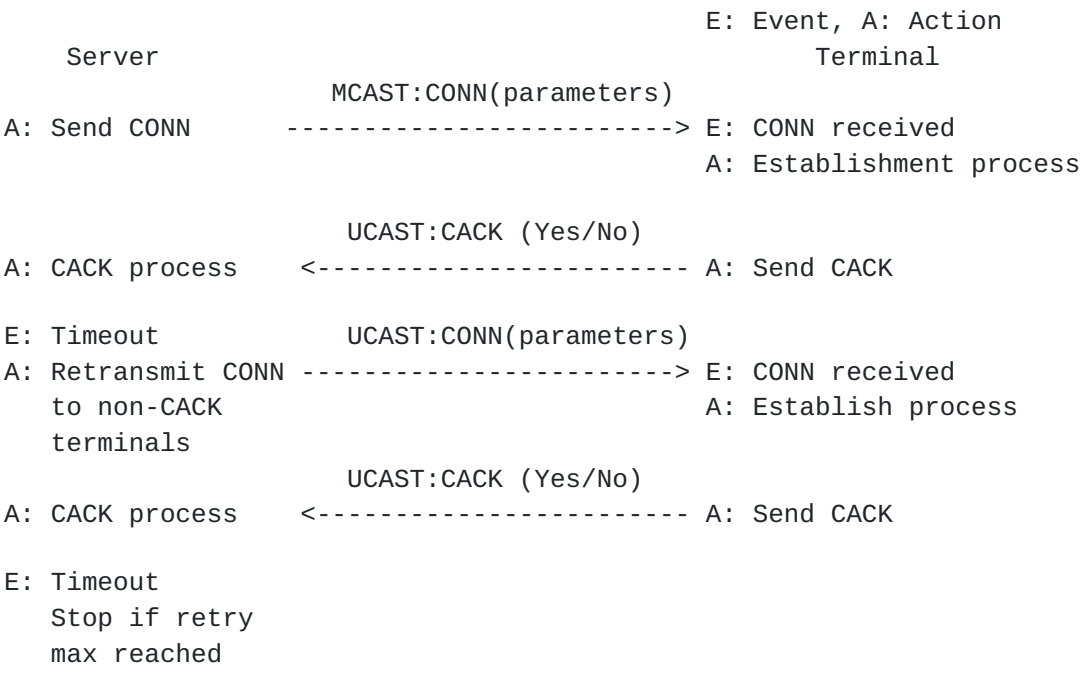
<Connection Establishment Process>

Connection establishment is conducted by transmitting a connection establishment request CONN from the server via multicast. It is confirmed by sending a CACK from the terminal side via unicast.

In case CACKs do not arrive from all the terminals, the server time-outs while waiting and retransmits a CONN via unicast to each non-responding terminal. The retransmission by timeout is repeated up to the maximum retransmission count to establish the connection.

The parameters associated to a CONN connection establish request includes protocol version, message size, number of blocks and block length (number of bytes per block).

A terminal may return YES or NO as a parameter of the CACK connection establishment confirmation. A NO indicates refusal to the connection establishment request, indicating that the terminal cannot receive data.



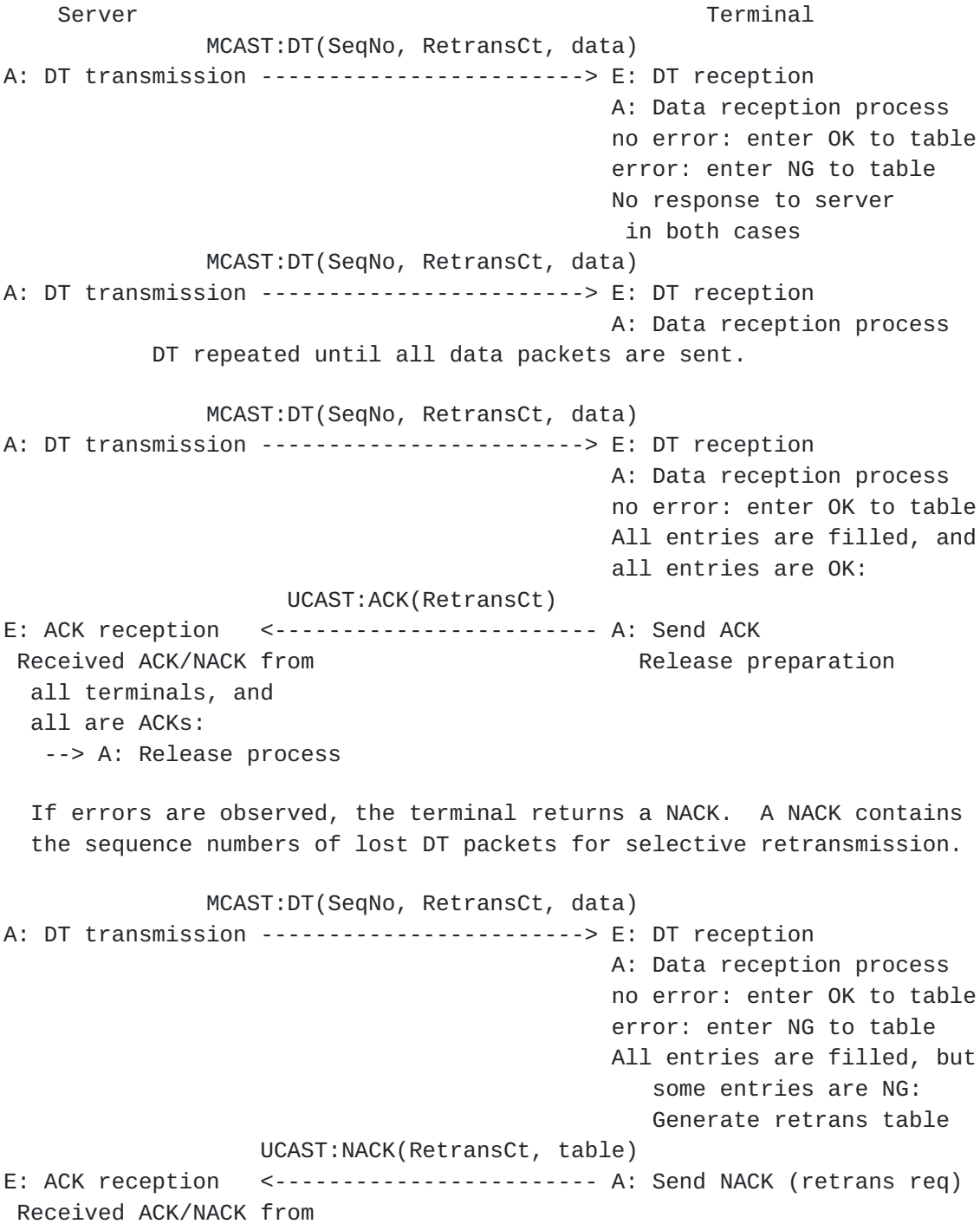
<Data Transmission via Multicast>

Data is transferred via multicast between the server and the terminals which established the connection. The data (message) provided by the

application is divided into DT packets and transmitted. The last DT packet is padded to the fixed DT length.

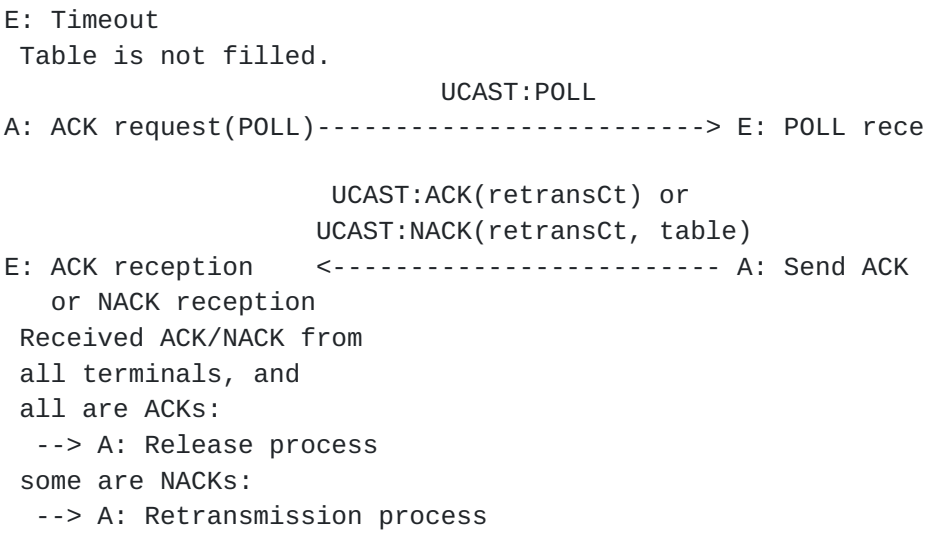
A DT packet contains parameters such as sequence number and retransmission count.

After the last DT packet of the multicast data transmission phase is transmitted, each terminal responds with an ACK or a NACK via unicast according to the DT reception status. It returns an ACK if all DT packets are received without error.



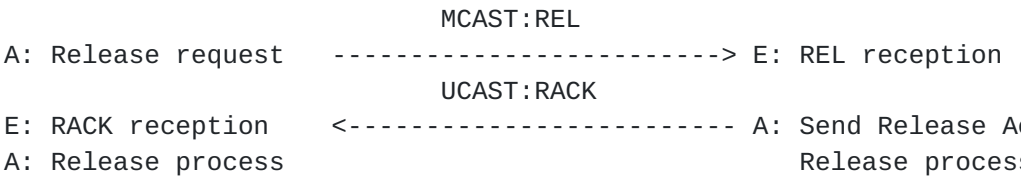
```
  all terminals, and
  some are NACKs:
   --> A: Retransmission process
```

If some DT packets are lost during transmission, the receiver does not observe the condition that all packets have arrived. In this case, a watchdog timer detects DT packet loss after a predefined time, and returns a NACK.



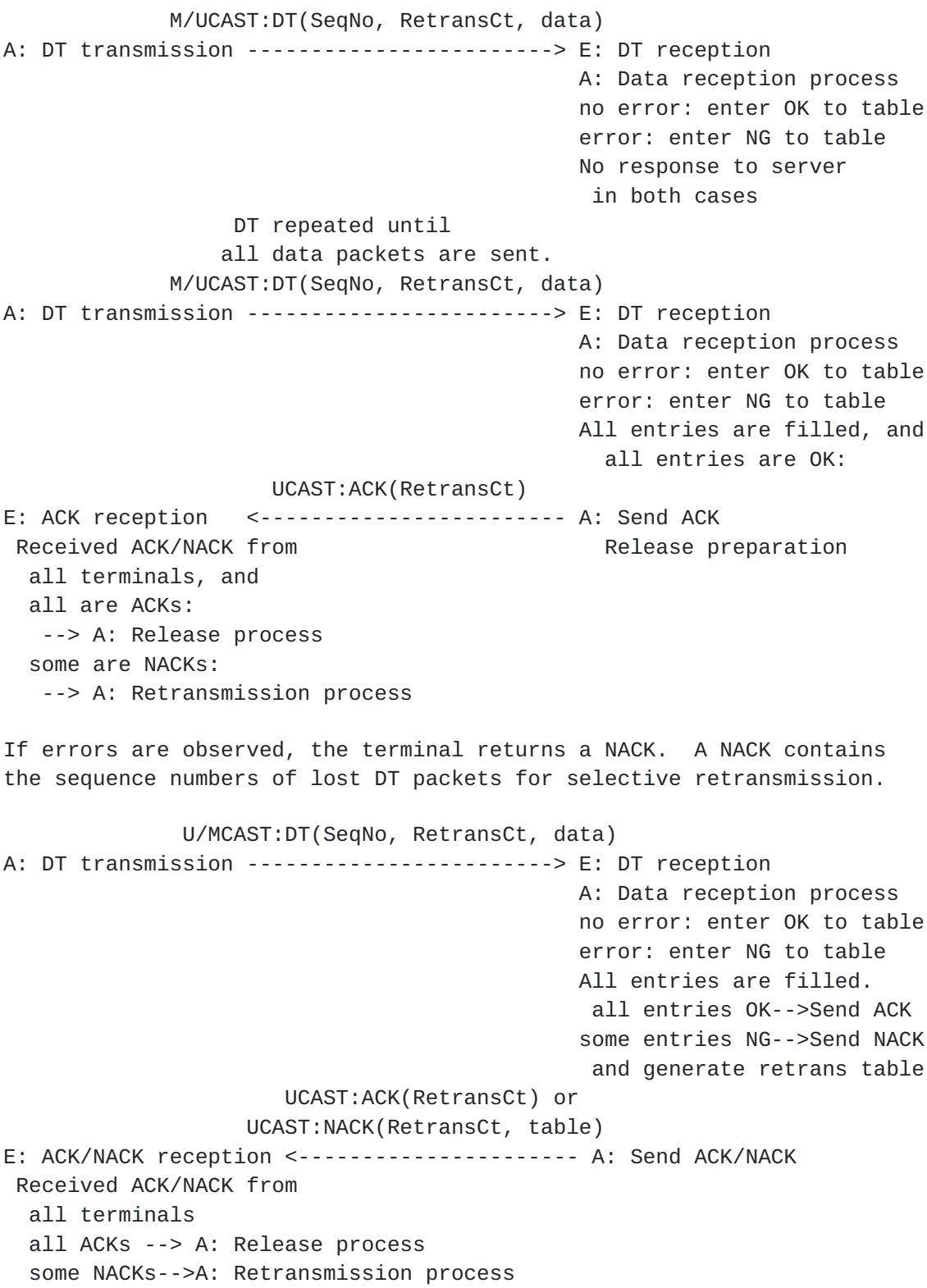
<Release Process>

Each time a multicast data transmission or retransmission phase completes, the terminals which received all the data without error releases the connection and leave from the session. The release process is such that the server sends a REL release request packet to the terminals via multicast, and only those terminals that received all data without error and are ready to release the connection (in the state of "RELEASE") return an RACK packet and release the connection.



<Retransmission Process>

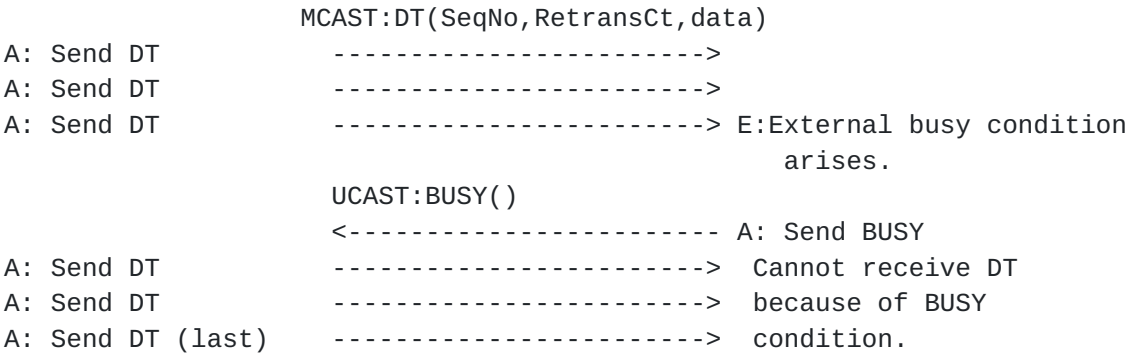
The server retransmits data if one or more terminals request retransmission by NACKs. Whether the server retransmits data via multicast or unicast is decided based on the evaluation function about the necessary time, communication cost and so on. If multicast is used, the server retransmits all the DT packets in the merged list of requests from the terminals. They are sent to all the connected terminals, and the terminals stores them selectively based on their requirements.



If errors are observed, the terminal returns a NACK. A NACK contains the sequence numbers of lost DT packets for selective retransmission.

<BUSY process>

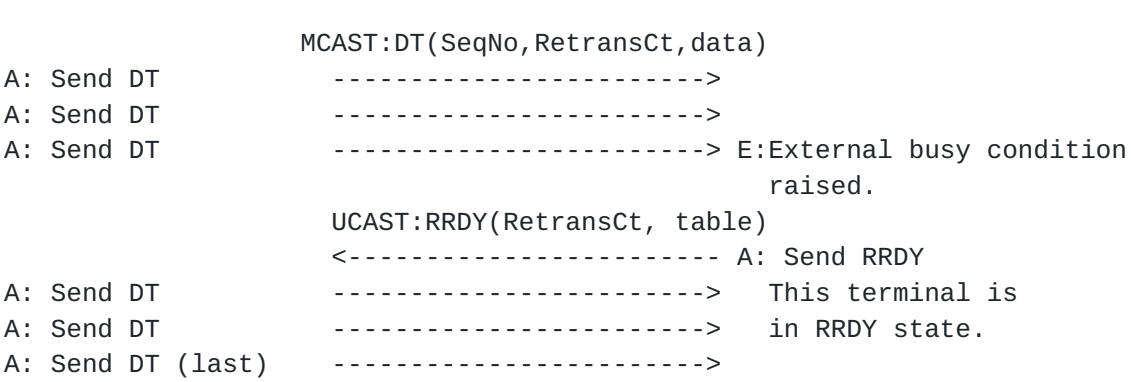
A terminal can indicate that it is in BUSY state because of its own reason. This is one case of flow control, and the terminal is left behind in the multicast data transmission phases. Actually, a terminal that indicates BUSY is excluded from the data transmission and retransmission processes from then. After all the multicast transmission/retransmission processes complete, the server retransmits the missing data to each individual BUSY terminals that lifted the BUSY condition via unicast.



ACK/NACK Wait state

E: ACK/NACK reception <---ACK/NACKs from------
other terminals

MCAST retransmission continues



ACK/NACK wait state

E: ACK/NACK reception <---ACK/NACKs from------
other terminals

```
All non-BUSY terminals completed.
BUSY separate retransmission state starts.

                  UCAST:DT(SeqNo,BusyFlag,data)
A: Send DT          ------------------------> E: BusyDT reception
                                              A: Data reception process
                                                To CONNECTED state
A: Send DT          ------------------------> E: BusyDT reception
A: Send DT (last)   ------------------------> E: BusyDT reception

ACK/NACK Wait state

E: NACK reception   <------------------------ A: Send NACK

                        Retransmission

E: ACK reception    <------------------------ A: Send ACK
A: Release process                            To RELEASE state
   Send REL         ------------------------> E: REL reception

E: RACK reception   <------------------------ A: Send RACK

            Retransmission to the next BUSY terminal
```

4. Packet Format

4.1 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.

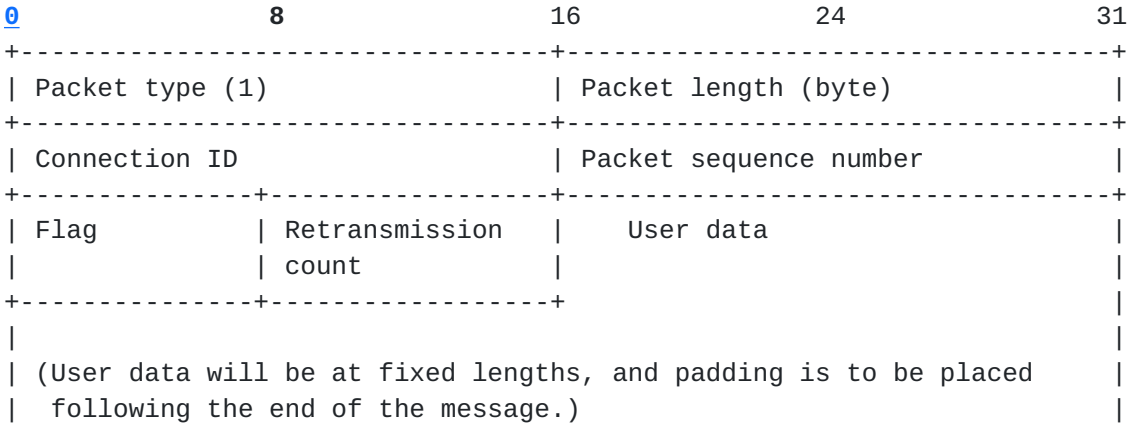
- Packets to be used in this protocol are to be transferred as user data of lower layer protocols. When UDP is used in lower layer, it

is stored in the data octets portion of the UDP User Datagram and transferred.

4.2 Packet Formats

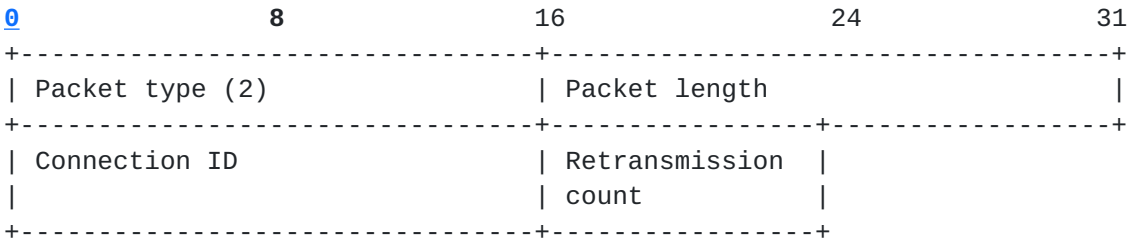
Legend UC: transferred by unicast MC: transferred by multicast

(1) Data packet (DT) :
[MC the first transmission and retransmission],
[UC retransmission and separate retransmission after busy]
A server to receivers.

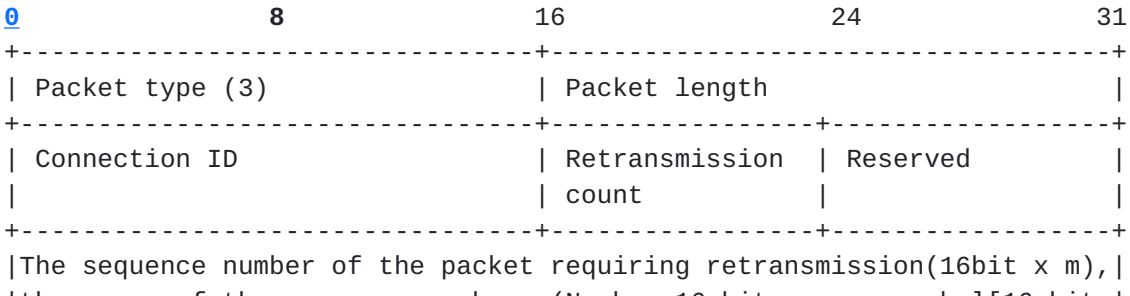


Packet type: integer value 1 to 11
Connection ID: integer (the same value as that used in CONN)
Packet sequence numbers: integer value 1 to 2**16-2
Flag: Separate re-transmission (BUSY re-transmission)(16th bit is ON)
LAST (17th bit is ON)
Retransmission count: integer value 0 (initial transmission),
1 (1st time re-transmission), 2, ---

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



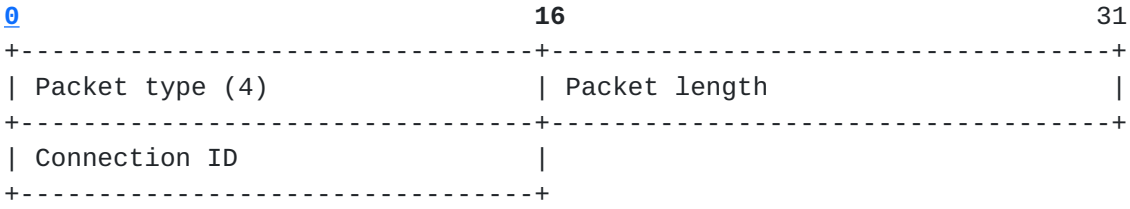
(3) NACK : [UC] receivers to a server



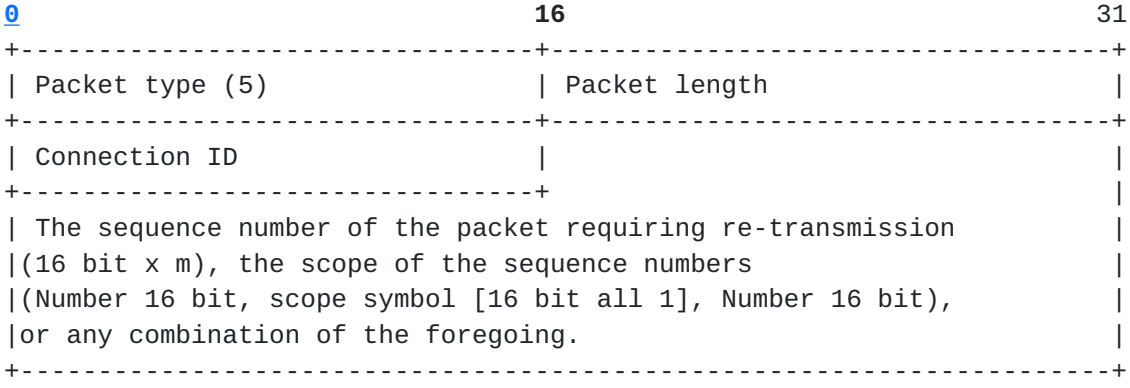
|all 1], Number 16 bit), or any combination of the foregoing. |
+---+

Scope of sequence numbers "10 to 15" is encoded "10 scope symbol 15".

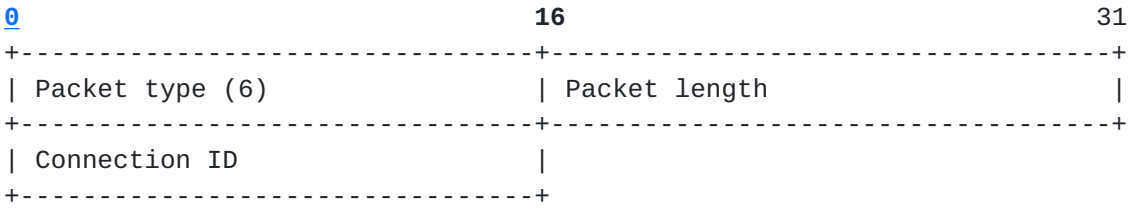
(4) BUSY Notification (BUSY) : [UC] a receiver to a server



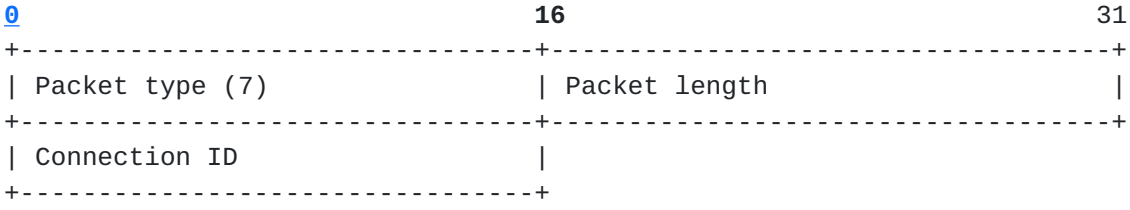
(5) BUSY release (RRDY) : [UC] a receiver to a server



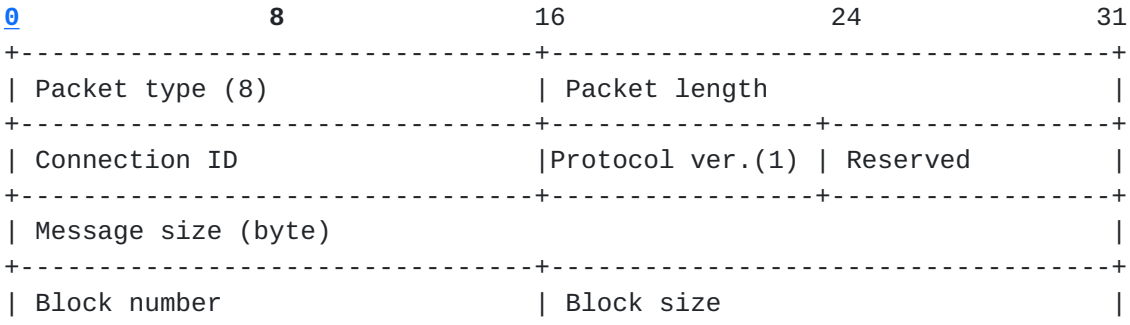
(6) ABORT : [MC] a server to receivers, [UC] a receiver to a server

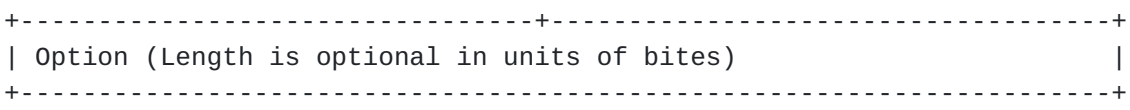


(7) State inquiry (POLL) : [UC] a server to a receiver



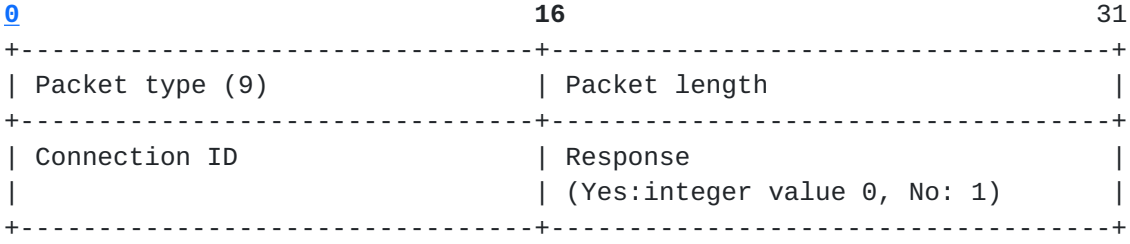
(8) Connection establishment request (CONN) :
[MC the first time], [UC retransmission] a server to receivers



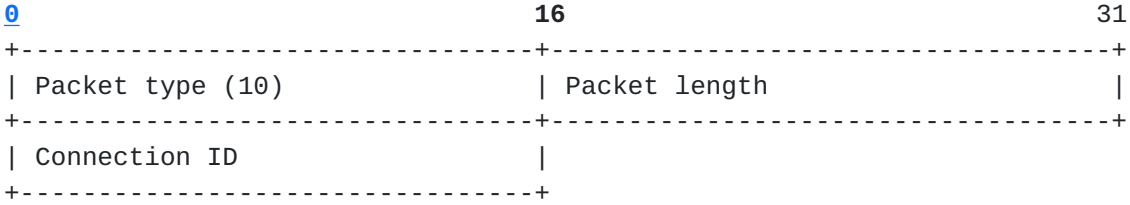

Option (Length is optional in units of bites)

- 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
- Message sizes at 32 bit may be designated up to about 4.2 Gbytes, and block sizes at 16 bit up to about 65 Kbyte
- Option field may be used for transfer of optional parameters (e.g. file names) with agreement between APs.

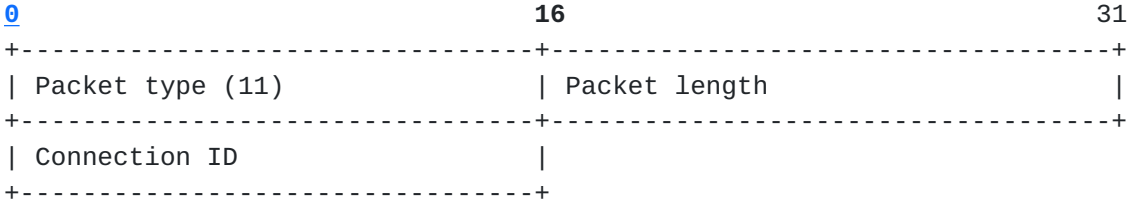
(9) Connection establishment response (CACK):[UC] receivers to a server



(10) Connection release request (REL) :
[MC after multicast transmission and retransmission],
[UC after unicast retransmission] a server to receivers



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



Under the protocol of this version, the foregoing packets are to be used. But with expansion of functions and other factors, there is the possibility of additional packet types being used.

Addendum for RMTP V1 protocol specifications

Extended features which are not defined in RMTP V1 specifications are described in this addendum. Some are realized as an local implementation without being defined as protocols and others are to be defined as protocols in RMTP V2 specification. Some performance reports on RMTP are also included in this addendum.

A1.Scalability

A1.1 Results of analyses and tests

RMTP performance, evaluated through protocol analyses and implementation tests, was reported in [SS96]. The analyses verified that the NACK based selective repeat error recovery procedure adopted in RMTP is Scalable in regard to receiver number and data size, and is applicable to high error rate environments such as 10 % packet loss. The implementation tests included large-scale receiver emulators with artificial errors and delays.
The tests confirmed that RMTP supports thousands of receivers. For example, 2 Mbytes of data can be distributed to five thousand receivers within three minutes using 0.8 Mbps bandwidth even against 1 % data loss.

A1.2 Response implosion

The major issue for response handling is evading response implosion. There are three causes of response implosion. The first is the frequency of ACK/NACK sent from receivers and this can be avoided by restricting ACK/NACK to just one time per re/transmission round.
The second is the size of response packet; smaller sizes are preferred. For example, in the encoding of lost packets in the NACK, range symbols is used to indicate the burst packets lost. The third cause is receiver number; this can be offset by using the backoff time algorithm as follows [DA94]. Each receiver holds its response for some, random, uniformly distributed delay period, called the backoff time, before sending it to the server. The time range of the distribution is passed to the receivers in the connection establishment parameters.
The algorithm is also used for responses in connection management (CACK/RACK).

The applicability of the algorithm to large scale multicasting was also examined through implementation and tests stated in [SS96].
The algorithm worked well and avoided response implosion for thousands of receivers if the backoff time range was appropriate. However, the transfer time linearly increases with the number of receivers for over five thousand receivers in 2 Mbyte data distribution tests. This shows

that there is some Scalability limit to the backoff time algorithm for over five thousand receivers.

The implementation of the backoff time algorithm in receivers is a local matter and is not specified in Vesion 1 specification. The notification of backoff time range from a server to receivers in the connection establishment packet (CONN) is to appear in RMTP Version2 specification.

A2. Flow/congestion control

A congestion control mechanism is indispensable for multicast data transmission since the effect of multicast flow on network traffic is far larger than unicast flows. Very special care is needed to achieve reliable multicast which might cause unnecessary retransmission load on congested networks. Flow control which avoids sending excessive multicast data and keep multicast flow moderate is a key technology against congestion. To this end, two flow control mechanism; separate retransmission and monitor-based rate control are adopted in RMTP[SS96].

A2.1 Separate retransmission

Separate retransmission is used for local congestion limited to some receivers and avoids redundant multicast retransmission to those receivers. When the performance of a receiver declines, the server freezes retransmission control to the receiver until re/transmission to the normal receivers is completed.

If one or more receivers cannot continue multicast data transfer for some reason such as poor receiver performance or congestion, separate unicast retransmission is conducted after the rounds of multicast retransmission finish. Such a receiver issues a BUSY packet to declare its condition to the server in the multicast transmission/retransmission phase. After recovering from its exceptional state, the receiver issues a ReceiveReady packet (RRDY) to show that it is able to receive retransmission packets. The states of received packets are kept in both the server and receiver until separate retransmission starts.

The server itself may detect the congestion affected receivers by the status of NACK or the timeout after POLL. It handles them as separated receivers. Congestion is detected by packet loss in end-to-end communication. This is also effective for the communication suspension caused by receivers moving into low transmission quality areas under mobile wireless environment.

The separated receivers may be re-grouped and unreceived data may be retransmitted by multicast to the group. However, this requires server-specified membership controls which today's routing protocols do not support.

The separate retransmission by BUSY notification is included in RMTP V1 specification. The separate retransmission by the detection of

performance degraded receivers by the server is to appear in RMTP V2 specification.

A2.2 Monitor-based rate control

Rate control based on end-to-end monitoring is used for global congestion, i.e. all receivers suffer performance declines due to network congestion. Rate control adapts to the overall receiver condition and reflects the data receiving condition in the current data transmission parameters. Congestion is detected by packet loss in end to-end communication.

Window flow control has been used in one-to-one transport protocols such as TCP; continuous acknowledgment from the receiver and a round trip timer are used. If window flow control is adopted in multicast, a large round trip time needs to be set to accommodate the many receivers, which will result in very low transmission performance. Window-based flow control also does not suit NACK-based multicast recovery procedures such as AFDP [KC96] and RMTP where ACK is not used to report each data packet reception. Rate control can be based on the NACKs occasionally emitted by receivers [KC96]. However, since RMTP uses a NACK only once for each round of data transfer or retransmission, receiver state monitoring by NACK transfer is not effective. Accordingly, RMTP adopts monitor-based rate control which is independent of error recovery and is sensitive to receiver performance.

The proposed rate control scheme is executed by monitoring receipt packet numbers in receivers as follows.

1. **The server indicates the start of packet number measurement to** receivers by setting the start bit of a data packet on. Each receiver starts counting the receiving packet number from the start packet.

2. **The server indicates the stop of packet number measurement to** receivers by setting the stop bit of a data packet on. When receiving the packet, each receiver stops counting the receiving packet number and sends back a report packet to the server including the number of packets received during the monitoring period.

3. **The server collects the reports, grasps the number of packets** received, and changes the transmission rate accordingly. The rate is defined as the number of packets to be continuously transmitted in an interval and the length of an interval. The decision is based on the trend of receivers conditions by such criterion as the mean value of received packet number.

By continuing the above scheme, the server can adaptively set the transmission rate to the overall transfer performance based on conditions such as network congestion. The effectiveness of the scheme depends on the criterion which decides the change of rate in step 3.

Response implosion can be avoided by requiring only representative

receivers to send back the reports. For example, one representative is chosen for each subnetwork. Rough conditions of the whole receivers can be estimated through monitoring the representatives.

Tests with up to one hundred receivers showed that the transfer time is improved by applying the proposed rate control [SS96], [SS97]. At the same time, variation in transfer time become small with the rate control.

Thus, the monitor-based rate control scheme improves the transfer performance and contributes to the stability of reliable multicast. This is accomplished by maximizing the transmission rate while avoiding unnecessary overflow.

The monitor based rate control is effective when most receivers have similar performance and experience similar conditions, so in combination with the use of receiver separation (A.2.1) the total flow control performance can be enhanced.

In rate flow control, the flow can be controlled by the function which decides a new rate based on the measured values as described in step 3. A moderate rate setting function which is sensitive to network congestion will be needed to avoid aggravating network congestion. A slow rate start mechanism comparable to TCP is a candidate for congestion control in the networks where.multicast traffic co-exists with TCP traffic. Several rate setting functions are under study in the time of writing this note.

A3. Security

A3.1 Layered consideration

Security measures for reliable multicast are proposed to realize the practical applications that demand control of information distribution. As fundamental measures at the middleware level, packet scrambling and multicast authentication are proposed to avoid the illegal use of multicast data and unauthorized participation in multicast groups[ST96]. They compensate the defect of IP multi-cast without incurring excessive processing overheads. These proposals focus on efficient measures to meet the minimum security level needed for most applications instead of pursuing strict security for specific applications.

Here, we consider communication security in three levels: Application level, Middleware level, and Network level (Fig.1).

Strong security is to be provided by Application level functions as required by each application, Network level functions rectify the vulnerability of the line and routers, and Transport level functions provide the common secure functions needed by various applications in end-to-end communications. Most of today's available security tools suit the Application and Network levels.

Layer (resources)	Security issues and measures
Application (applications)	Security depending on each application requirement - Confidentiality and integrity of application data - User authentication
Transport (middleware)	Common security between end systems - Confidentiality of transport data - End-system authentication
Network (OS,router,link)	Security for network - Encryption and authentication of IP packets - Link-by-link encryption and authentication

Fig.1 Layered security architecture

For the Network level, the security technologies being standardized in IPSEC WG/IETF can be used as strong security measures. RMTP relies on the results of IPSEC WG for Network level security such as the authentication and encryption of IP packets.

This note focuses on the efficient measures at the Transport level needed to ensure that a variety of applications meet the minimum security instead of pursuing strict security. We omit discussion of strict end-to-end security implementation for each application.

The following security features have been tested in RMTP implementation and are to appear in RMTP V2 specification.

A3.2 Packet scrambling

Multicast distribution is achieved by using common group addresses such as a class D address for IP multicast; receivers can freely pick up the distributed IP packets by designating their address. Even unregistered receivers can pick up the distributed packets. Because group distribution by common address is a simple and acceptable solution for multicast realization, the IP multicast defect of weak confidentiality must be overcome. Packet scrambling is proposed to eliminate this defect.

Packet scrambling is proposed at the middleware level to enhance the fundamental security of multi-cast applications. Encryption incurs a lot of processing overhead as stated in the recent study of Internet protocols [AR95] and has been implemented as hardware in network devices. Instead of using encryption, a simple packet scrambling

technique that has little overhead but still prevents illegal use is proposed for multicast transport.

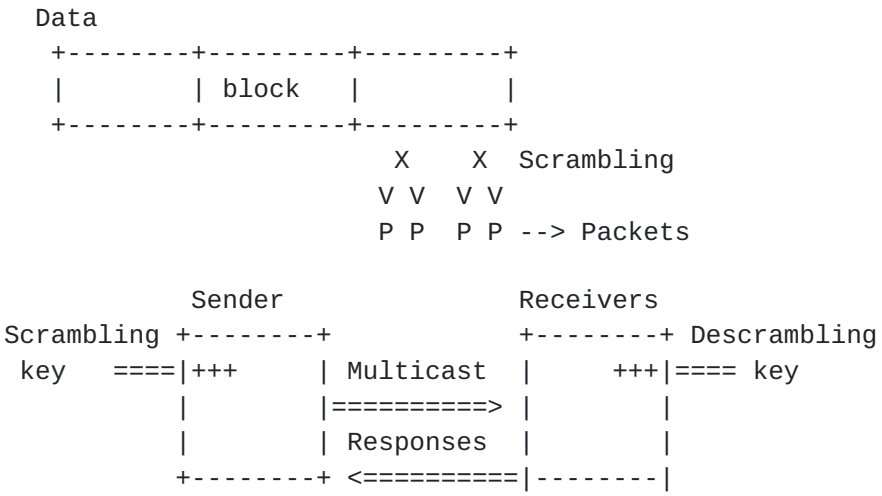


Fig. 2 Packet scrambling procedure

The algorithm proceeds as follows (Fig.2). The server process assigns new scrambled sequence numbers to the data packets by a scrambling key instead of following the order of stored data. The receivers replace the altered sequence numbers of received packets with the true sequence numbers during the packet receiving process by a descrambling key. Data recovery procedures between the server and the receives use the altered sequence numbers. Altered packet sequence numbers are used for both way communications.

Whether scrambling is used or not is indicated in the connection request packet which is sent from the server to the receivers before commencing data distribution. The descrambling key, which is calculated >from the scrambling key, is assumed to be delivered to the registered receivers by a method such as secure one to one protocols or off-line means. Scrambling and descrambling are performed using fixed size of data blocks (n packets), by permutating the sequence number according to the permutating code (n pieces of number?? digits??) and by the descrambling key which is the reverse function of the code used for scrambling.If the message is not a multiple of a fixed size (n packets), padding packets are used.

The experiment of 2 Mbyte data distribution to one hundred receivers showed that such scrambling and descrambling procedures do not visibly increase the computing load as observed by CPU time.
These observations confirm that scrambling does not increase the transfer time and the processing load because number permutation is done when constructing the sending packet in the server.
Descrambling also does not increase the receiving time and the

receiving load because packet renumbering is done when placing the received packets into receiver memory.

A3.3 Receiver authentication

Masquerading is another threat in group communications where many members are expected to join and leave a multicast group. Multicast groups that include mobile receivers suffer this defect because their addresses change as the receivers move to other subnetworks. Authentication for each transport packet might be applied with significant processing cost. A more practical alternative, an authentication architecture suitable for multicast transport protocols is proposed to provide the basic authentication commonly needed by a variety of multicast applications.

Authentication in the connection establishment phase before commencing multicast data distribution is used as shown in Fig. 3. This benefits from connection-oriented reliable multicast protocols.

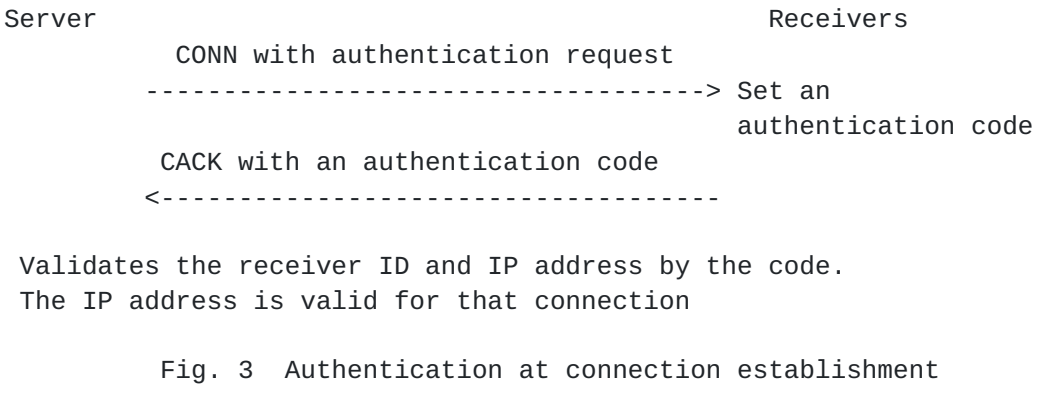


Fig. 3 Authentication at connection establishment

In the connection establishment phase before data distribution, the server requests receiver authentication by including an authentication indication with a random number in the connection request packet multicasted to the receivers. Each receiver sends back a response that includes its identification and the authentication code calculated from the identification, the secret key shared with the server, and the random number notified in the connection request. The key is assumed to be distributed safely by the server or by a delegate of the server. The random number is changed for each connection establishment to avoid the illegal reuse of the code.

The server checks the authentication code by comparing the value calculated by the receiver's identification and the shared key. If the receiver is authenticated, the server registers the receiver identification and its IP address derived from the receiver's IP packet in the retransmission management table which is used for data recovery in reliable muticast.

After authentication for all responding receivers is completed, the server commences multicast distribution based on the retransmission management table. The server relies on the registered IP address of receivers to validate the receiver's identification throughout a connection. Unregistered responses are not accepted by the server and consequently do not establish reliable multicast connection to such receivers. Further protection against unauthorized receivers is provided by changing the packet scrambling key or by aborting all connections when an illegal access is detected.

The response packets are indispensable in verifying authorized receivers. The above procedure uses authentication only once for one multicast connection instead of authenticating each response and control packet from the receivers. The proposed procedure consumes less authentication processing overhead than the authentication of all responses from the receivers. We believe that this procedure satisfies the minimum level of security requirement needed for a variety of applications.

We have implemented RMTP with the above authentication procedures and MD5 for generating an authentication code. MD5 is a standard message digest function which has also been proposed to be used in IP packet authentication (as an option). Tests confirmed that the application of authentication does not affect the transfer performance.

A3.4 Group key distribution

The proposed packet scrambling and multicast authentication architecture assumes key distribution management between server and receivers. Several key management systems are usable including manual key distribution.

Manual key distribution is effective for sharing secret keys among a small number of receivers. A key management system is indispensable in order to make key ad-ministration effortless in large-scale multicast applications. The proposed design allows decoupling of the key management mechanism which can then be provided by another system. The separation of the key management protocol from the proposed procedures allows us to benefit from the results of key management research which has a long history.

Recent research has introduced the key distribution procedure based on the multicast routing protocol in the network layer [BA96]. The approach assumes the experimental multi-cast routing protocol, Core Base Tree, and relies on a multicast routing tree whose nodes are authenticated when spanning the tree. This method is promising

although it might cause extra overhead in maintaining the routers. Key management strength needs to be evaluated while taking performance and management overheads into consideration.

References

[AR95] Atkinson, R., Security Architecture for the Internet Protocol, IETF RFC1825, Aug. 1995.

[BA96] Ballardie, A., Scalable Multicast Key Distribution, IETF RFC1949 Experimental, May 1996.

[CZ85] Cheriton, D. R. and Zwaenepoel, W., Distributed Process Groups in the V-Kernel, ACM Transactions on Computer Systems, Vol. 3, No. 2, pp. 77-107, May 1985.

[DA94] Danzig, P. B, IEEE, Flow Control for Limited Buffer Multicast, IEEE Transactions on Software Engineering, Vol. 20, No. 1, pp. 1-12, Jan. 1994.

[KC96] Kotsopoulos, S.and Cooperstock, J.R., Why Use a Fishing Line When You Have a Net? An Adaptive Multicast Data Distribution Protocol, 96 USENIX Technical Conference, Jan. 1996.

[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.

[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.

[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.

[SS96], [SS97], and [ST96] are available at http://isserv.tas.ntt.co.jp/chisho/rmtp/rmtprefe.htm or http://www.trl.ibm.co.jp/rmtp/rmtprefe.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