11

Performance evaluation of reliable multicast transport protocol for large-scale delivery

T. Shiroshita, T. Sano,
O. Takahashi, and M. Yamashita
NTT Information and Communication Systems Laboratories
1-2356 Take, Yokosuka, 238-03 Japan.
siro@isl.ntt.jp

N. Yamanouchi and T. Kushida
IBM Research,
Tokyo Research Laboratory
1623-14 Shimotsuruma,
Yamato, 242 Japan.
yamanouc@trl.ibm.co.jp

Abstract

This paper analyzes the performance of a reliable multicast transport protocol and discusses experimental test results. The Reliable Multicast Transport Protocol has been proposed to support "reliable" information delivery from a server to thousands of receivers over unreliable networks via IP-multicast. The protocol provides high-performance for most receivers through the advantage of IP multicast while also supporting temporarily unavailable or performance impaired receivers. Its applicability to large scale delivery is examined using an experimental network and the backoff time algorithm which avoids ACK implosion. The two types of flow control with the protocol are also examined. Separate retransmission is used to offset the local performance decline limited to a small number of receivers. Monitor-based rate control is used to offset the global performance declines due to causes such as network congestion.

Keywords

Reliable multicast protocol, large-scale delivery, rate-based flow control, performance analysis

1 INTRODUCTION

The information age requires large-scale reliable information distribution functions. Scalable information delivery has become feasible with the advent of high-speed networks which offer large bandwidth. However, the number of users still affects the reliability and performance of information distribution systems [NE94].

For scalable information distribution, reliable multicast has been studied for efficient and reliable information distribution. A generic solution to all categories is not feasible because of the inhomogeneity of service requirements and different solutions have been proposed for each service category. Representative reliable multicast can be considered as follows based on the service type and needed reliability.

W. Dabbous et al. (eds.), *Protocols for High-Speed Networks V*

(a) Massive information delivery: Digital information prepared in a server is distributed to many subscribers without any data flaw with some service time constraint. Electronic newspaper delivery and data replication for naming or archive services are typical applications [OB94]. The main research issue is scalability and efficient multicast which achieves complete data replication by recovery procedures [KC96], [PTK94]. This type of service is very promising in actual publishing services; it suits for the broadband networks that are now emerging [IT91].
(b) Group communications: Members of a group share the same information which is distributed to all members. Shared whiteboard is a typical application in this category. The shared information is updated by short multicast messages by each member. The main research issue is efficient concurrent multicast which guarantees the message order [BS91], [CS93], [EM95]. Scalability may be also required while relieving order constraint [FJM95].
(c) Distributed simulation: Many simulation sites execute simulations and share the results by multicast. Each site must distribute its state information to other sites within a limited time frame to effect the simulation. The main research issue is scalable multicast while keeping the severe time constraint [HSC95], [SK96].

The Reliable Multicast Transport Protocol (hereafter RMTP) [STS95], [STY96], [TS96] has been proposed for type (a) applications by using IP-multicast [DE90]. RMTP supports a "reliable" information delivery from a server to thousands of subscribed receivers. "Reliable" information delivery means complete data replication in the receivers and receiver confirmation before and after data delivery.

This paper analyzes the performance of RMTP and discusses experimental test results. While the large-scale implementation of reliable information delivery is expected, few reports have addressed this subject for type (a) applications. Only a general case model analysis can be applied for large-scale delivery including thousands of receivers [PTK94].

The proposed protocol RMTP is realized in the transport layer with an unreliable transport protocol UDP over connectionless best-effort network protocol IP. Responses are reported to the server as end-to-end transport communications instead of using gateways (inter-mediate node) which gather some responses and respond to the server (root node) or upper gateways. The reason why gateways are not used is that they entail tree structure maintenance including the structure of lower gateways and leaf nodes. This end-to-end retransmission concept has been argued [SRC84] and is supported by most reliable multicast protocols. Furthermore, imposing gateway functions on the intermediate hosts or routers creates unwarranted costs when constructing and maintaining a large-scale information delivery system.

RMTP is a receiver-initiated protocol in which receivers are responsible for detecting data packet loss and sending NACK to the server. An analysis based on a generalized model reports that receiver-initiated reliable multicast protocols provide substantially higher throughput (packet processing rate) than sender-initiated reliable multicast protocols [PTK94] and most reliable multicast protocols follow the receiver initiated approach [WMK], [KC96]. RMTP follows with this idea and fundamentally supports receiver-initiated reliable multicast. In addition to NACK, RMTP also uses ACK for complete receipt confirmation for the whole data set from each receiver.

The ACK implosion problem; response overflow caused by response concentration to the server, which is inevitable with end-to-end confirmation can be solved by the backoff time algorithm [CZ85], [DA94]. In this paper, the applicability of the backoff time algorithm is examined for large scale delivery in the framework of RMTP.

Another intractable issue in reliable multicast for large scale information delivery is that receiving conditions decline by network congestion or receiver local problems. These receiving performance degradation factors will eventually occur in each receiver or globally in all

receivers. We try to resolve this issue by providing high-performance multicast to most receivers while still supporting temporarily unavailable or performance impaired receivers. To this end, two flow control procedures; separate retransmission and monitor-based rate control are adopted in RMTP.

Separate retransmission is provided to offset any local performance decline experienced by a receiver due to causes such as occurred in wireless environment. When the performance decline of a receiver is detected, the server freezes the retransmission control to the receiver until re/transmission to other normal receivers are completed. Thus, the proposed procedure prioritizes the normal state receivers while still supporting temporarily unavailable receivers.

The monitor-based rate control is used to offset the global performance decline experienced by all receivers; causes include network congestion. Rate control adapts to the whole receivers and reflects the whole data receiving conditions to the current data transmission rate.

After examining related works in Section 2, the design concept of RMTP is described in Section 3. The error control part of the protocol is analyzed and compared with experimental results as to throughput (transfer time) and server processing load (packet processing amount) in Section 4. The adaptive end-to-end rate control is also evaluated through performance tests.

2 RELATED WORKS ON RELIABLE MULTICAST

Reliable multicast protocols

Reliable multicast protocols applicable to massive information delivery are described below and compared against our RMTP. The trend in reliable multicast protocols for massive information delivery and group communication are reported exhaustively in [OB94] and [DD96].

Multicast Transport Protocol (MTP) [AFM92] is a general purpose multicast protocol over UDP and IP-multicast. It bases retransmission on receiver-initiated NACK and fixed-size window control which causes heavy retransmission overhead. MTP also adopts dynamic membership control via a group master which is also responsible for retransmission. Due to these heavy control overheads, MTP has not been implemented. MTP's retransmission idea can be found in Reliable Broadcast Protocols (RBP) [CM84]. RBP was proposed for broadcast LANs and is not scalable beyond LANs.

Reliable Multicast Protocol (RMP) [WMK] is a successor of RBP. RMP reduces the group master overhead by rotating the role of group master among all members as in RBP. However, its scalability is still questionable and reported tests cover less than 10 receivers. RMP seems to be somewhat dedicated to CSCW applications judging from its support of ordering guarantee and multi-RPC.

Adaptive File Distribution protocol (AFDP) [KC96] is a promising, recently proposed multicast protocol for massive information delivery. AFDP is also a receiver-initiated protocol as is RMTP. AFDP uses only NACK for data receive confirmation, and also supports rate control based on NACK reporting. These data recovery and flow control ideas can be found in NETBLT [CLZ87] which was developed for one-to-one bulk data delivery. AFDP sends NACK several times for each round of data transfer, while RMTP uses it only once and sends only one NACK/ACK for each round. Furthermore, no amendment for NACK loss is supported in AFDP as is true in the previous three multicast protocols. Testing of AFDP has been reported for 16 receivers with 7 Mbyte data and 68 receivers with 133 Kbyte.

XTP, which was developed primarily for high speed data transfer, also adopts reliable multicast. However, the XTP revision 4.0 specification [XT95] leaves details of flow and error controls to the user implementation while sufficiently supporting multicast group management.

ACK implosion

Any system that makes the information serving process dependent on responses from the information receiving process suffers from the ACK implosion problem [DA94]. The response concentration to the server results in response loss and causes redundant retransmission. In our protocols, the server requires explicit responses of complete receipt notification (ACK) or incomplete receipt notification (NACK).

Against ACK implosion, a promising solution is the backoff time algorithm which was proposed by [CZ85]. Its behavior with limited socket buffers was analyzed by the queuing model and tested for ten real receivers and rather large size response packets [DA94]. We have examined and tested the proposed protocol using backoff time algorithm in a real Ethernet LAN environment for up to ten thousand receivers by receiver emulation including packet loss and network delay. The results confirm that the backoff time algorithm is effective even in large-scale environments with thousands of receivers. (see Section 4)

Other research [PTK94] analyzed reliable multicast protocols in the form of generic model that uses NACK for each data packet loss. The revised protocol also uses multicast for NACK and reduces the number of NACKs sent to the server in such a way that receivers do not issue NACKs if other receivers have already sent a NACK (by multicast) for the data packet. The analysis showed that, while some overhead is imposed on receivers, server performance is improved by this amendment. However, the protocols have the restrictive assumption that NACKs are never lost in the network. Further, multicasting by receivers in large scale high delay networks will cause a flood of responses all over the network.

A recent research [H96] introduced a different approach for the implosion problem, the Local Group Concept (LGC), in which receivers in a LAN consist a local group and coordinately execute retransmisssion. The data retransmission is achieved first in the LAN by a local group controller, then the data missed by the local group is reported to the server and the server retransmits the data to the local groups. Simulations in a restricted WAN and LANs showed LGC approach achieved better end-to-end transfer delay and wide area link traffic than the retransmission all supervised by a server. If LGC is implemented with RMTP, it will enhance the receiver scale order by the size of local group.

Another recent research [LP96] has also proposed a reliable multicast transport protocol for massive-data delivery which is based on periodical ACKs and window flow controls. The protocol adopts hierarchical gateways to offset ACK implosion. The approach is completely in the opposite direction to the RMTP proposed in this paper. Experiments for 1 Mbyte file distribution for eighteen receivers in a real LAN and WAN environment has been reported.

3 RELIABLE MULTICAST TRANSPORT PROTOCOLS

3.1 Basic multicast retransmission procedure

The proposed multi-round multicast retransmission procedure, which meets the requirement of complete reliability, is shown in the multicast transmission /retransmission phase in **Figure 1.**

The whole data set such as a file, is split into multiple transport packets with sequence numbers. After the first round of multicasting all packets, packets not received due to error or loss are reported to the server from the receivers by unicasting NACK. The server determines the retransmission packets needed from the unreceived packet reports in NACK; packet duplication is prevented by referring to their sequence numbers. The server then multicasts the retransmission packets in the second round data transfer. Thus, retransmission is based on the selective-repeat procedure which requires fewer retransmission packets than go-back N

procedures. The server continues retransmission until no packet loss is reported by receivers. In the receivers, duplicated packets are neglected.

This multi-round procedure assumes retransmission management tables to record the numbers of successfully transmitted packets both in the server and receivers. The server manages all packet conditions for each receiver. The size of the whole data set, the total number of data packets, and data packet size are indicated in the connection establishment phase prior to the data transmission phase from the server to the receivers and the information is used for packet retransmission management at both sides.

The number of retransmissions as a function of the error rate was assessed. **Figure 2** shows the number of retransmissions needed to realize multicast retransmission versus the bit error rate (BER) or packet error rate (PER) based on the analysis in **Appendix A**. The error rate is assumed to include all transmission error and packet loss throughout the network from an sending end to a receiving end for each packet. Delivery amount is 2 Mbytes and the packet error rate is shown for the 1 Kbyte packet size. Several curves are drawn for differing numbers of receivers up to 500,000.

Note on data size: A 32 page daily newspaper amounts to 2 Mbytes (text and compressed images). The Internet version of the NYTimesFax (A4 size 8 pages, pdf format) occupies about 100 Kbytes, which suggests that a 160 page book would compress to 2 Mbytes.

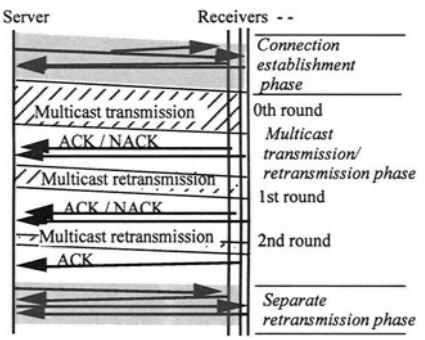

Figure 1 Communication sequence overview.

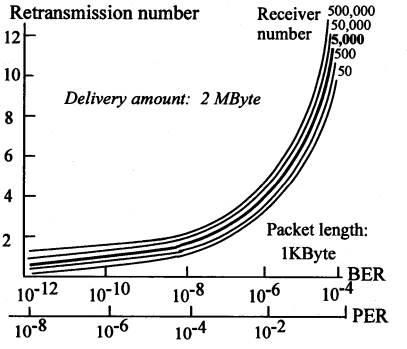

Figure 2 Retransmission number vs error rate.

As an example, for the 5000 receiver case, the graph shows the number of multicast retransmissions needed for 10,000,000 packets (2000 packets x 5000 receivers) to be correctly received. For the PER of 10^{-2} (BER 1.22 x 10^{-6}), three or four retransmissions are needed to complete data transfer. The BER 10^{-6} corresponds to rather low quality networks, since the standard ATM cell loss and error rates for international connections being discussed for Quality of Service in ITU-T SG13 are 3 x 10^{-7} and 4 x 10^{-6} which correspond to BERs of less than 1 x 10^{-8} (a cell is 53 bytes) and the BER for packet switched networks is said to be better than 10^{-9}. We note that in the best-effort type networks such as the Internet, PER may decline to 10%. Even for PER 10% , the graph shows that seven retransmissions achieve complete delivery to 5000 receivers. These estimations have been confirmed in the experimental tests in Section 4.2.

Figure 2 also shows that receiver number does not strongly affect the retransmission number. Data size also has a little impact on retransmission number in the assessment according to the same analysis in Appendix A.

3.2 Reliable Multicast Transport Protocol in detail

This section describes detailed procedures in RMTP and examines flow controls. The major communication phases are the connection establishment phase, the multicast transmission /retransmission phase, and the separate retransmission phase as shown in **Figure 1.**

3.2.1 Connection management

An explicit connection establishment procedure is used in order to pass to the receivers the information needed for retransmission such as total packet number. Furthermore, connection establishment and release procedures are necessary to confirm the registered users prior to data transfer and to charge for successful data delivery.

Membership is decided before the connection establishment phase by some other protocol or manually and is assumed to be constant during the data transfer phase until communication is complete and all connections are released. This assumption is realistic for bulk-data multicasting, while other interactive multicast communications such as whiteboard may require dynamic membership control during the data transfer phase.

Connection establishment and release procedures in RMTP are as follows.
The server sends connection establishment request packet (CONN) to all receivers via multicast based on the membership list given by the application. Each receiver sends an acknowledgment packet (CACK) which includes Yes (ready to communicate) or No (unable to communicate) to the server via "unicast". The server also uses a timer to wait for CACK responses and retransmit CONN by "unicast". Authentication parameters may also be included in the CACK for secure communication establishment.

The server multicasts a connection release packet (REL) after each round of data re/ transmission and the receiver's response is ReleaseACK (RACK). NACK or BUSY receivers just ignore REL packets.

3.2.2 Response handling

Response handling procedures in the recovery operations stated in Section 3.1 are as follows. The sequence numbers of all data packets not received for one round of data transmission are included in a NACK and reported to the server by unicast. When a receiver receives all packets, the receiver sends ACK to the server by unicast. The reason why RMTP uses the explicit ACK instead of implicitly confirming the success of data receipt by timeout (no NACK) in the server, is that the timeout is used for the response loss detection In case of the response timeout, the server sends a POLL packet which solicits ACK/NACK (see **Figure 3**). The usage of POLL saves the cases of ACK/NACK loss or a short time receiver inability and, consequently, stops these problems causing unnecessary multicast retransmission traffic.

The inevitable issue for response handling is evading response implosion. There are two causes of response implosion. One is the frequency of ACK/NACK issued from receivers and this can be avoided by restricting ACK/NACK to just one time per re/transmission round. The other cause is receiver number and this is 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 range of the distribution is informed to the receivers beforehand included 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 is examined in a later section.

The current response packet sizes of ACK, CACK, and RACK are 5, 6, and 4 bytes, respectively. NACK size is variable. NACK header is 8 bytes and the following NACK information size is (the number of lost packets and/or range symbol) x (the number or symbol size; 2 bytes). The range symbol is used to express the continuous burst error packets in a shorter NACK packet. Packet loss of 1% yields NACKs of 48 bytes for 2000 data packets for random error.

In the multicast re/transmission phase, only multicast was used in the analyses and tests, although unicast retransmission can be used for a small number of receivers depending on the priority of reducing traffic or optimizing throughput.

3.2.3 End-to-end flow controls

Two flow control procedures; separate retransmission and monitor-based rate control are applied to RMTP depending on the state of performance degradation.

Separate retransmission
Separate retransmission is used for the local performance decline limited to a few receivers and avoids to continue unnecessary redundant multicast retransmission. When the performance decline of a receiver is detected, the server freezes the retransmission control to the receiver until re/transmission to other normal receivers are completed. If one or more receivers cannot complete multicast data transfer procedures for some reason such as poor receiver performance or human interruption, 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 low performance receivers by the status of NACK or the timeout after POLL and handle them as BUSY/RRDY declared receivers. This is effective for the communication suspension caused by receivers moving into low transmission quality areas under mobile wireless environment.

Monitor-based rate control
The monitor-based rate control is used for the global performance decline such that all receivers suffer performance declines due to network congestion. Rate control adapts to the whole receiver condition and reflects the data receiving condition to on the current data transmission by making monitoring procedures independent from recovery procedures.

Window flow control has been used in one-to-one transport protocols such as TCP based on the continuous acknowledgment from a receiver. Window-based flow control 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 in AFDP is based on the NACKs occasionally emitted by receivers. 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 from error recovery and 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 receipt packets during the monitor period.
3. The server collects the reports, compares them to the number of transmitted packets, and changes the transmission rate accordingly. The rate is defined as the number of packets to be continuously transmitted in a interval. The decision is done based on the trend of most of the receivers conditions by such criterion as a mean value of receipt packets number.

By continuing the above scheme, the server can adaptively set the transmission rate to the whole receiving performance based on overall conditions such as network congestion. Effect of the scheme depends on the criterion which decides the change of rate in the step 3. Elaborated criteria are used in the experiment shown in Section 4.3.

The implosion which may be caused by reports can be avoided by selecting representative receivers which send back the reports among the all receivers when very large number of receivers exist. Rough conditions of the whole receivers can be estimated through monitoring the representatives.

4. PERFORMANCE EVALUATIONS

4.1 Model evaluations

Performance evaluations on connection establishment and multicast re/transmission phases are shown based on model analysis prior to the experimental evaluation in the next section.

The performance of the proposed data distribution procedure was modeled and evaluated by the criteria of transfer time and packet processing amount, both of which are important in service and system design. The analysis mainly addresses the basic multicast retransmission procedure including the use of POLL, while the effect of BUSY is not included as it is receiver environment dependent. The derivation directly shows the mean values of transfer time and packet processing amount.

(1) Distribution time evaluation

Term definitions

N_k: the number of receivers in the kth round transfer

N_0: the whole number of receivers to be transmitted to

M_k: Message length of kth retransmission (Number of packets forming the message)

M_0: the whole data set size given by application, M_1: the first retransmission data size

N_k and M_k are obtained by the calculations given in **Appendix A.**

T_k: total time of kth round transfer

T_0: the initial transfer, T_1: the first retransmission

N'_k: the number of responses (ACK/NACK) that overflow in the server or the number of first POLL packets for the kth round

N''_k: the number POLL packets lost during the first POLL transmission which need retransmission

δ: mean response producing interval at the "receiver set". NACK is assumed to be sent by the receiver in the same way as ACK. $1/\delta$ corresponds to response rate of the receiver set according to a uniform random distribution over $[0, \delta N_k]$, $[0, \delta N'_k]$, and $[0, \delta N''_k]$

v: transfer rate at server (packets / second); constant rate is assumed in the model

e: packet loss rate of a packet transiting the whole network one way

Pol(x): probability that at least one POLL is used as received from x receivers in a round
V(x): mean number of packets overflowing the server buffer as received from x receivers in a round
Since buffer overflow rate in a round is expressed as Buf(x) = V(x) / x , Pol(x) is obtained as

$$Pol(x) = 1 - (1-e)^x + (1-e)^x Buf(x). \quad (1)$$

The first two terms are the network loss rate, and the third one is the overflow rate at the server. V(x) is calculated through direct queuing system calculation as follows [STY96].

$$V(x) = \sum_{n=b+1}^{x} P_{n,b}, \text{ where } P_{n,b} = (1 - 1/\rho)^b \sum_{m=0}^{n-b-1} {}_{m+b-2}C_{b-2}\, 1/\rho^m, \text{ and } \rho = 1/\delta\mu.$$

μ: service rate at server.

The timer value for the server to wait for responses from receivers is TAT + δ Nk + α, where TAT: turn around time and α: a surplus time taken to set the timer.

Optimal performance is given by α = 0, but for implementation some positive value is set.

In the following derivation, trailing terms which express retransmission of POLL more than once are neglected. That is, the rate of POLL loss more than once is very small and POLL transmission is assumed to be at most twice for each silent receiver.

The data transfer time for the kth round can be calculated as based on the communication sequences in **Figure 3**.

$$\begin{aligned} Tk &= Mk / v + TAT + \delta Nk + \alpha \\ &\quad + N'k / v + (TAT + \alpha) Pol(Nk) + N''k / v + (TAT + \alpha) Pol(N'k) + (\text{POLL loss} > \text{once}) \\ &\cong (Mk + eNk + V(Nk) + V(N'k)) / v + (Nk + V(Nk) + V(N'k)) \delta \\ &\quad + (1 + Pol(Nk) + Pol(N'k)) (TAT + \alpha) \end{aligned} \quad (2)$$

where N'k = e Nk + (1- e) V(Nk), N''k = e N'k + (1- e) V(N'k),
and the higher order small terms e^2, eδ, and so on are neglected.

In the last two equations for N'k and N''k, the first and second terms express the network loss rate of response packets and the buffer overflow rate at the server, respectively.

In the equation for Tk, the coefficients of 1/v, δ, TAT, and α express the effects of data packet transmission time, backoff time, turn around time, and timer surplus time, respectively.

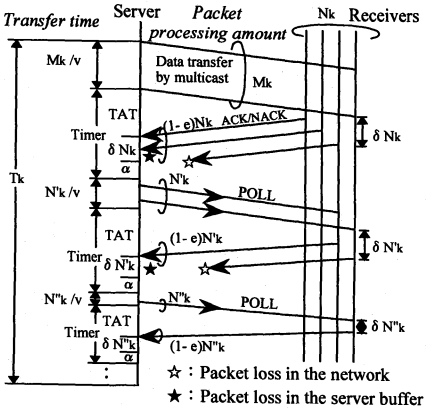

Figure 3 The k th round retransmission sequences.

Note that if message size Mk is small the effect of backoff time dominates the data transfer time as the order of δ is as great as that of 1/v. Hence, the bigger the message size M is, the smaller the relative overhead of the backoff time becomes. This means that the proposed procedure suits the distribution of large amounts of data.

In the same way, the transfer time of connection establishment is obtained as

$$TC \cong (1 + e\,N + V(N) + V(N')) / v + (N + V(Nk) + V(N'k)) \delta + (1 + Pol(N) + Pol(N')) (TAT + \alpha), \quad (3)$$

where $N' = e\,N + (1- e\,) V(N)$, $N'' = e\,N' + (1- e\,) V(N')$.

Finally, the total data transfer time can be obtained as

$$T = TC + \sum_{k=0}^{\infty} Tk. \quad (4)$$

(2) Packet processing load

Packet processing load at a server can be calculated in the same way as the transfer time with the same definitions and assumptions. Packet processing load refers to UDP packet amount needed to send and receive RMTP packet encapsulated in UDP user data. The packet processing load corresponds to CPU time used in the operating system (CPU system time).

The number of packets sent, received, and sent-and-received for the kth retransmission is expressed as follows based on the communication sequences in **Figure 3**.

$$PSk = Mk + e\,Nk + V(Nk) + V(N'k), \qquad PRk = Nk + (1 - e\,) (V(Nk) + V(N'k)),$$
$$Pk = Mk + (1 + e\,)Nk + (2 - e\,) (V(Nk) + V(N'k)), \quad (5)$$

where $N'k = e\,Nk + (1- e\,) V(Nk)$ and Nk is as given in **Appendix A.**

The number of packets sent, received and sent-and-received during connection establishment phase are

$$PCS = 1+ e\,N + V(N) + V(N'), \qquad PCR = N + (1 - e\,) (V(N) + V(N')),$$
$$PC = 1 + (1 + e\,)N + (2 - e\,) (V(N) + V(N')), \quad (6)$$

where $N' = e\,N + (1- e\,) V(N)$.

The number of packets processed for connection release with no retransmission is given by PLS = 1 + K, PLR = N, and PL = 1 + K + N, where K is the retransmission number. Consequently, the number of packets sent, received and sent-and-received by the server is

$$PS = PCS + \sum_{k=0}^{\infty} PSk + PLS, \qquad PR = PCR + \sum_{k=0}^{\infty} PRk + PLR, \qquad P = PC + \sum_{k=0}^{\infty} Pk + PL. \quad (7)$$

As the CPU system time used in an operating system can be considered to be defined only by the numbers of sent and received packets irrespective of the type of packets, processing load in the operating system is approximated by the following expression.

$$\text{CPU system time} = a_1\,PS + a_2\,PR + a_3, \quad (8)$$

where coefficients a_1, a_2, and a_3 are determined based on the values measured when executing the protocol in a real environment. The values are estimated in the next section. On the other hand, CPU user time consumed by the user process is considered to depend on the type of packets and cannot be approximated simply by the number of packets processed.

4.2 Experimental evaluations and comparison to the model analysis

(1) Implementation
We implemented RMTP on Sun workstations SS4/20, SS4/2, etc. with Solaris 2.3 OS in NTT Labs, and IBM RS6000 with AIX 4.1 OS in IBM TRL. Both OSs support IP-multicast and the programs were implemented as application processes on UNIX socket interfaces.

Two types of RMTP program source code were independently developed in the two research organizations based on the RMTP protocol specification collaboratively developed by them. The two teams jointly conducted inter-connectibility tests in a LAN environment and confirmed the interoperability of the protocol in September 1995.

The following performance tests were executed on the implementation in Sun WSs. The network environment was a 10 Mbps Ethernet LAN with three subnetworks connected by a router; IP multicast was supported. The tests did not include the effect of BUSY in the same way as the analysis in Section 4.1.

(2) Performance tests for large-scale information delivery
Two different emulators were used, Mars and Lares, in order to create a large scale communication environment consisting of hundreds and thousands of receivers.
(a) Full-specification medium-scale emulator (Mars)
Mars produces an environment of hundreds of receivers and high delay networks. Mars consists of multiple RMTP receiver processes. Mars can produce specified artificial packet loss and specified artificial delay. As Mars supports many full specification RMTP receiver processes in one workstation, the receiver buffer overflows when a relatively high transmission rate is set at the server.
(b) Limited-specification large-scale emulator (Lares)
Lares produces an environment of thousands of receivers by limiting the protocol procedures. Lares is an RMTP receiver process which emulates thousands of receivers and responds always positively with CACK, ACK, and RACK. ACK arriving process based on the backoff time algorithm can be created by setting an inter-packet gap in the responding Lares process.

[Experimental results and comparison to the model analysis]
By combining Mars and Lares we generated environments of differing receiver scale, packet loss rates, and network delays. The following results are based on the 1 % packet loss of Mars by specifying the loss rate and by setting the transmission rate so as to achieve good transfer performance. The go-and-back delay value (turn around time) is 100 msec (assuming two ATM switches and two routers over 3000 km optical networks) and 600 msec (assuming satellite two links or public packet-switched network in use). UDP socket buffer of 50 Kbytes and UDP packet size of 1 Kbyte are used. The backoff time is used for over 1000 receiver cases with δ values from 2 to 4 ms. Data size was always 2 Mbytes.

Figures 4 and **5** show the transfer time and processing load as functions of receiver numbers. In the case of about ten receivers, only real receivers were used. Seven workstations installed with Mars, which emulates ten to twenty receivers, were used to emulate 100 receivers for 100 to 5000 receiver cases. Mars was also used for the 50 receiver case. Two to ten workstations with Lares, emulating 75 to 1000 receivers, were used for 500 to 10,000 receiver cases. Tests were done several times for the same case and the graphs in the figure are the mean values.

Figure 4 shows that RMTP achieves acceptable transfer time for practical applications. For example, 2 Mbytes (one newspaper) can be delivered to 5000 users within 3 minutes. Furthermore, the curve linearly increases with the number of receivers for over 5000 receivers. This shows the limitation of the backoff time algorithm for over ten thousand receivers.

The transfer time obtained by equation (4) is depicted as theoretical values in **Figure 4** with the timer surplus time α set to 0 to show optimal values. The fact that the surplus time was set large enough to accept delayed responses explains the difference between observed values and theoretical values. However, the delay does not strongly impact the transfer time or CPU processing load and is neglected in the graph.

For comparison, we tested the repeated use of four parallel FTP with no artificial packet loss in the same LAN environment. The parallel FTP processes consumed the full 10 Mbps bandwidth of the Ethernet LAN while RMTP used only some bandwidth depending on the selected transmission rates. Only 12 % of bandwidth was used for data transmission from the server for over 100 receivers using the fixed rate of 0.8 Mbps.

In additional tests for 2 Mbyte data delivery to 100 emulated receivers, 10 % packet error rate case required 68 sec by five retransmissions while 0 % and 1 % cases required 22 sec with no retransmission and 40 sec by two retransmissions.

As for packet processing load at the server, the same characteristics can be observed in **Figure 5**. Here, regarding the CPU system time values obtained by equation (8), coefficients a_1, a_2, and a_3 can be determined from the test results. Consequently, the approximation is obtained as CPU system time = 7.0×10^{-5} PS + 2.3×10^{-4} PR + 0.1. The coefficients show that receiving a packet incurs 3 times CPU load as sending a packet does.

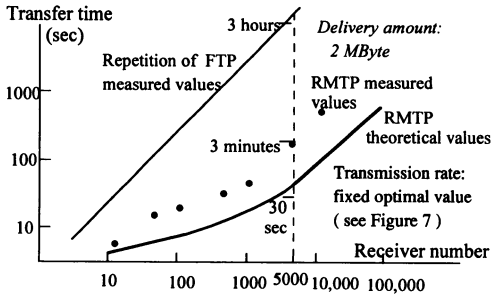

Figure 4 Transfer time in the server vs delivery scale.

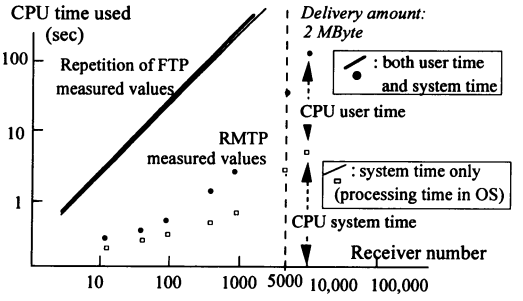

Figure 5 CPU load in the server vs delivery scale.

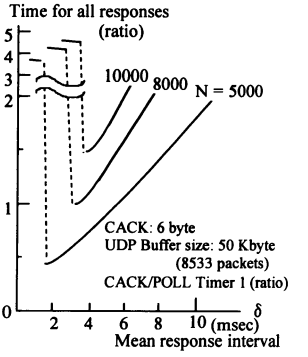

Figure 6 Effect of backoff time.
Observed in connection establishment.

Lastly, the impact of back off time on transfer time is shown in **Figure 6**. The curves show that the backoff time needs to be small for large numbers of receivers to ensure high transfer time efficiency while backoff time that are too short yields drastic overhead by lost packet waiting timeout. UDP buffer overflow in the server is also observed with small backoff times in the 5000 and 8000 receiver cases even if a large enough buffer (8533 packets) is used. This may be caused by excessive buffer management.

4.3 Experiences from rate control

A performance test of the monitor-based rate control scheme was executed in the same network environment. Before examining the adaptive rate control, the effect of transmission rate on communication performance was examined. **Figure 7** shows the test results of transfer time ratio versus transmission rate at the server observed in the test of previous section where the fixed rates were used. The curves show that the transmission rates must be carefully controlled to achieve good performance under unstable network or receiver conditions.

The rate control tests included the case of RMTP with the proposed adaptive rate control and the case of RMTP with optimally selected fixed rates. The two criteria of adaptive rate control are adopted; (1) the transmission rate is increased or decreased depending on the number of receivers in which the number of lost data packets exceeds a specified value and (2) the rate is decreased depending on the number of receivers which do not respond in a specified time. The tests were executed for 10 Mbyte data delivery to real seven receivers and 100 receivers consisting of seven workstations running Mars.

The frequency for 20 times trials versus transfer time ratio observed is shown as a histogram in **Figure 8**. The both of 100 and seven receiver cases shows that the transfer time is improved by applying the proposed rate control. At the same time, variances of transfer time become small when the rate control is used. 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.

Additional tests including 600 msec delay did not show much impact to transfer time in the same way as the tests in Section 4.2.

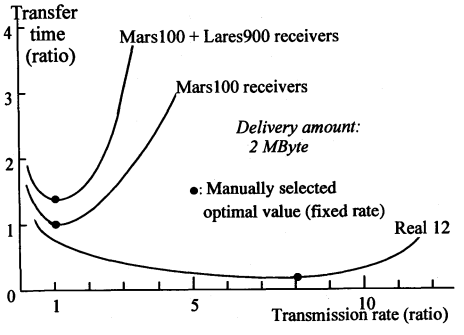

Figure 7 Transfer time vs transmission rate.

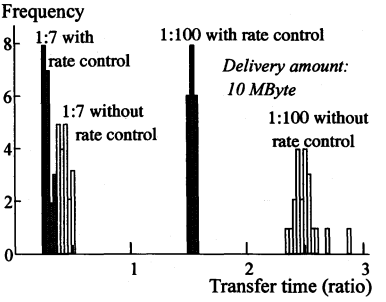

Figure 8 Histogram of transfer time with and without rate control.

5. CONCLUDING REMARKS

Error recovery procedure of Reliable Multicast Transport Protocol (RMTP), which has been proposed for delivering large amounts of data to large numbers of users, was evaluated through analyses and performance tests in LAN environment including large-scale receiver emulators. The tests confirmed the feasibility of proposed retransmission procedure in realizing large scale information delivery for thousands of receivers. Against the ACK implosion caused by end-to-end responses, the backoff time algorithm was applied and test results showed that the algorithm is applicable up to about ten thousand receivers in practical use.

As for flow control for reliable multicast, the monitor-based rate control was addirionally applied to the protocol for global performance degradation and showed enhanced performance and stability regarding transfer time. The separate retransmission, in which some performance degraded receivers are detected and retransmitted after suspension, was also realized in the protocol. However, the criteria of applying the separate retransmission and their effectiveness is still under evaluation.

As the user number increases, security issues such as data confidentiality and user authentication, and network administration issues such as multicast routing will also become important for practical multicast services. RMTP is going to be used in corporate information delivery systems first. Multicast administration issues are to be studied through field tests.

REFERENCES

[AFM92] Armstrong, S., Freier, A., and Marzullo, K., Multicast Transport Protocol, *RFC1301*, IETF, 1992.

[BS91] Birman, K., Schiper,A, and Stephenson, P., Lightweight Causal and Atomic Group Multicast, *ACM Trans. on Computer Systems*, Vol. 9, No. 3, pp. 272-314, Aug. 1991.

[CLZ87] Clark, D.D., Lambert, M.L., and Zhang, L., NETBLT: A High Throughput Transport Protocol, *ACM SIGCOMM'87, CCR* Vol. 18, No. 5, Aug. 1987.

[CM84] Chang, J.M. and Maxemchuk, N.F., Reliable Broadcast Protocol, *ACM Trans. on Computer Systems*, Vol. 2, No. 3, pp. 251-273, 1984.

[CS93] Cheriton, D.R. and Skeen, D., Understanding the Limitations of Causally and Totally Ordered Communication, *14th Synmp. Operating System Principles,* ACM SIGOPS, pp. 44-57, Dec. 1993.

[CZ85] Cherington, 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.

[DD96] Dabbous, W. and Diot, C., Group Communication; a state of the art, Submitted to *IEEE Journal on Selected Area in Communication.* Special Issue on Group Communication, 1996.

[DE90] Deering, S.E., Multicast Routing in Internetworks and Extended LANs, *ACM Trans. on Computer Systems*, No. 8, pp. 85-110, 1990.

[EM95] Mayer, E., An Evaluation Framework for Multicast Ordering Protocols, *ACM SIGCOMM '92*, CCR, Vol. 22, No. 4, Aug. 1992.

[FJM95] Floyd, S., Jacobson, V., McCanne, S., Liu. C. G., and Zhang, L., A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing, *ACM SIGCOMM '95, CCR,* Vol. 25, No. 4, pp. 342-356, Aug. 1995.

[H96] Hofmann, M., A Generic Concept for Large-Scale Multicast, *International Zurich Seminar on Digital Communications*, Feb. 1996.
[HSC95] Holbrook, H.W., Singhal, S.K., and Cheriton, D.R., Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation, *ACM SIGCOMM '95, CCR*. Vol. 25, No. 4, pp. 328-341, Aug. 1995.
[IT91] B-ISDN SERVICE ASPECTS, *Recommendation I.211*, ITU-T, 1991.
[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.
[LP96] Lin,.J. C., Paul, S., RMTP: A Reliable Multicast Transport Protocol, *IEEE Infocom'96,* Mar. 1996.
[NE94] Neuman, B.C., Scale in Distributed Systems, *Readings in Distributed Computing Systems*, IEEE Computer Society Press, 1994.
[OB94] Obraczka, K., Massively Replicating Services in Wide-area Internetworks, PhD dissertation, Computer Science Department, University of Southern California, Dec. 1994.
[PTK94] Pingali, S., Towsley, D. and Kurose, J.F., A Comparison of Sender-Initiated and Receiver Initiated Reliable Multicast Protocols, *ACM SIGMETRIX '94,* Vol. 22, No. 1, pp. 221-230, 1994.
[SK96] Smith, W.G. and Koifman, A, Distributed Interactive Simulation Intranet Using RAMP, a Reliable Adaptive Multicast Protocol, Proceedings from the Fourteenth Workshop on Standards for the Interoperability of Distributed Simulations, Orlando, Florida, Mar. 1996.
[SRC84] Saltzer, J.H., Reed, D.P., Clark, D.D., End-To-End Argumrnts in Systems Design, *ACM Transactions on Computer Systems,* Vol. 2, No. 4, pp. 277-288, Nov. 1984.
[STS95] Shiroshita, T., Takahashi, O., Sano, T., Yamashita, M., Nakamura, Y., Yamanouchi, N., and Kushida, T., Reliable Information Delivery System for Internets, *IPSJ SIG Notes, AVM*, Vol.95, No. 117, Dec. 1995. (in Japanese)
[STY96] Shiroshita, T., Takahashi, O., Yamashita, M., Yamanouchi, N., and Kushida, T., Reliable Multicast Transport Protocol and its Applicability to Emerging Networks, *Technical Report of IEICE*, IN95-140, Mar. 1996. (in Japanese)
[TS96] Takahashi, O., Shiroshita, T, Sano, T., Yamashita, M., Maruyama, M., and Nakamura, Y., A Proposal for Reliable Information Multicast Environment: Implementation and Evaluation, *IFIP 14 th World Computer Congress - Advanced IT Tools*, Sep. 1996.
[XT95] XTP Forum, Xpress Transport Protocol Specification Revision 4.0, XTP 95-20, Mar. 1995.
[WMK] Whetten, B., Montgomery, T., and Kaplan, S., A High Performance Totally Ordered Multicast Protocol,Theory and Practice in Distributed Systems, Springer Verlag, *LCNS 938*.

Appendix A Assessment of basic multicast retransmission procedure

This appendix assesses how many rounds of retransmission are needed until the basic multicast retransmission procedure proposed in section 3.1 completes all data retransmission; i.e., the whole data is correctly received by the receivers. The error rate is assumed to include all transmission error and packet loss throughout the network from an sending end to a receiving end for each packet.

[Definitions and assumptions]

N: Number of receivers

M: Message size (Number of packets forming the whole data set)

e: packet error (loss) rate (PER) for a packet multicasted through the whole network and received by a receiver. PER is considered as error rate in the network by assuming packets are not lost in the receiver if transfer speed is carefully chosen.
Sk: Number of packets to be confirmed by the receivers in the kth round of data transmission, that is, number of packets that has not yet received by the receivers just before the kth round transmission
Mk: Message size of kth data retransmission (number of packets) (M0 = M)
Nk: Number of receivers waiting for data retransmission in the kth round (N0 = N)
Sk is expressed as follows.

$$S_0 = M\,N, \qquad S_k = e^k M\,N \tag{a1}$$

Mk and Nk are obtained as follows.

$$M_1 = \{1 - (1 - e)^N\}\,M, \qquad N_1 = \{1 - (1 - e)^M\}\,N$$

$$M_2 = \{1 - (1 - e)^{S_1/M_1}\}\,M_1, \qquad N_2 = \{1 - (1 - e)^{S_1/N_1}\}\,N_1, \quad \text{----------}$$

$$M_k = \{1 - (1 - e)^{S_{k-1}/M_{k-1}}\}\,M_{k-1}, \qquad N_k = \{1 - (1 - e)^{S_{k-1}/N_{k-1}}\}\,N_{k-1} \tag{a2}$$

where Sn/Mn: mean number of receivers, Sn/Nn: mean number of message in the n th round,
$(1- e)^x$: a probability that all x (= N, Sn/Mn ; number of receivers) packets copied by IP-multicast are received without any packet loss for a packet or a probability that all x packets (M, Sn/Nn ; message size) are received without any packet loss for a receiver, and
$1- (1- e)^x$: a probability that packet loss occur in some of x receivers to which the packet needs to be delivered, that is, the packet needed to be retransmitted in the next round or a probability that packet loss occur in some of x packets which must be received by the receiver, that is, the receiver needs to be delivered in the next round.

Set (Sk, Mk, Nk) is calculated by the above equations as in the following example.

Example:
Packet error rates (PER) $e = 10^{-2}$ to 10^{-6} which correspond to bit error rates (BER) 1.22×10^{-6} to 1.22×10^{-10} when 1 Kbyte size packet is used, since 1 PER = 8,000 BER with assuming error occurs randomly for any bit. M = 2000 packets, N = 5000 receivers.

Table A Retransmission number assessment

k \ e	10^{-6}	10^{-4}	10^{-2}
0	(10^7, 2000, 5000)	(10^7, 2000, 5000)	(10^7, 2000, 5000)
1	(10, 10, 10)	(10^3, 906, 787)	(10^5, 2000, 5000)
2		(0.1, 0.1, 0.1)	(10^3, 906, 787)
3			(10, 10, 10)
4			(0.1, 0.1, 0.1)

The values of blanks are all smaller than 0.01.

The retransmission numbers can be read as follows.
For PER 10^{-6}, multicast is considered to complete with the 1st round of retransmission.
For 10^{-4}, multicast is considered to complete with the 1st or 2nd round of retransmission.
For 10^{-2}, multicast is considered to complete with the 3rd or 4th round of retransmission.
In the same way, we can estimate the retransmission numbers for higher error rate cases.
7 retransmissions for e = 10 %, 13 retransmissions for 30 %, where M = 2000, N = 5000.