Throughput Analysis of ARQ Protocol for Group Communications *

6 T − 7

Takayuki Tachikawa, Hiroaki Higaki, and Makoto Takizawa †
Tokyo Denki University ‡
e-mail{tachi, hig, taki}@takilab.k.dendai.ac.jp

Abstract

Automatic repeat request (ARQ) protocols have been analyzed for quite some time, but group communication where multiple senders send messages to multiple receivers in a group has not discussed so much. In the group communication, since acknowledgments of messages can be piggy-backed by messages, both the forward and feedback channels are considered as same type of channel. In addition, since each process has both the retransmission and resequencing buffer, efficient buffer allocation schemes can be adopted. In this paper, we analyze the throughput of ARQ protocol for such group communications.

1 Introduction

Automatic repeat request (ARQ) protocols are used to reliably deliver packets in the correct sequence. There are three types of ARQ as follows.

- Stop and Wait (SW)
- Go Back N (GBN)
- Selective Retransmission (SR)

These protocols are analyzed and expanded, e.g. multiple copies scheme [4] and postponed retransmission scheme [1]. But most of them are discussed to realize reliable delivery of packets from one transmitter to one receiver by using uni-directional data channel. This means that the data is transmitted only from the transmitter and the receiver process does not transmit data but ACK and NACK through the feedback channel. In order to exchange packets between two processes, i.e. both processes transmit data packet, two independent uni-directional channels are needed. We discuss the bi-directional channel where the forward and feedback channels are not distinguished. When two processes use the bi-directional channel, each process has both the retransmission and resequencing buffers to send and receive packets. In addition, the ACK and NACK are piggy-backed by data packets. While most of previous works assume that the transmitter always has new packets to transmit, we assume that desired transmission rates of two processes are not same and they are dynamically changing. We try to allocate retransmission and resequencing buffers dynamically for their desired transmission rates.

In Group communication [6], multiple processes transmit data packets to multiple receivers in the group. We consider that the group communication protocol is not a combination of multicast protocols [7, 11]. Each process has both the retransmission and resequencing buffers for each process in the group. We consider how to allocate the buffers for each process.

*グループ通信における ARQ プロトコル
†立川 敬行, 桧垣 博章, 滝沢 誠
‡東京電機大学

In this paper, we discuss two points of the ARQ protocol. First, we consider how to share the retransmission and the resequencing buffers in one-to-one communication. Second, we consider how to share the resequencing buffer for each sender in group communication.

2 System Model

A communication system is composed of processes and bi-directional channels between every two processes. Each process can send fixed length data packets with acknowledgment information, i.e. piggy back, and acknowledge packets which do not include data. Each packet contains error detection bits. Hence, we assume that errors are only packet losses.

There are four types of communication in Figure 1. (1) 1-to-1 communication is discussed for normal ARQ protocols. (3) 1-to-n communication is a multicast (multi-receivers) communication. We consider (2) 1-to-1 (bi-directional) and (4) n-to-n communications.

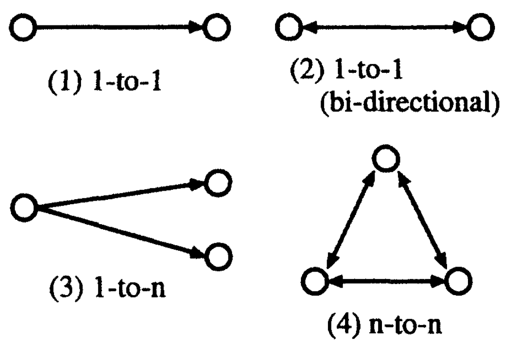

Figure 1: Types of communications.

3 1-to-1 Communication

Each process has two types of buffer, retransmission and resequencing ones. Figure 2(1) shows the 1-to-1 communication using uni-directional channel. Figure 2(2) shows the 1-to-1 communication using bi-directional channel. Here, the size of retransmission buffer of transmitter and the size of resequencing buffer of receiver is same, i.e. $RTB_A = RSB_B$ and $RTB_B = RSB_A$. The total size of buffers is fixed and equal to N, i.e. $RTB_A + RSB_A = RTB_B + RSB_B = N$.

Each process can share both the retransmission and the resequencing buffer space. In the beginning of a transmission, $RTB_A = RSB_A = RTB_B = RSB_B = N/2$. If the desired transmission rate is decreased, the process A can decrease the retransmission buffer size and increase the resequencing buffer size. After that, the process reports it to another process B. B decreases the resequencing buffer and increases the retransmission buffer size. Conversely, if the desired transmission rate of A is increased, A increases the retransmission buffer size, and reports it. If B receives the information which reported the increase of A's retransmission buffer size, B increases the resequencing buffer size immediately.

By using two independent uni-directional channel, the processes cannot share the buffers, and the buffer size has to be fixed. On the other hand, since each process can share the buffers, the size of buffers can be dynamically changed by using bi-directional channel. Hence, the buffer sizes are proportional to the current desired transmission rates of both processes.

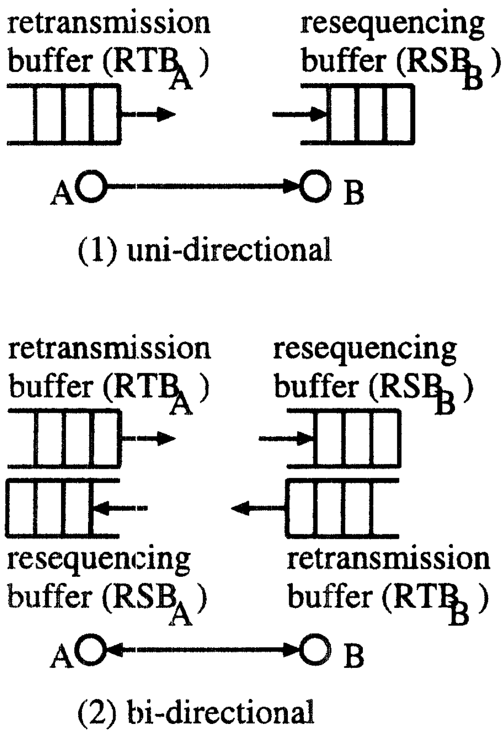

Figure 2: Buffering capacities.

4 n-to-n Communication

In the n-to-n communication, the group is composed of n processes. Each process can transmit data packets to n processes in the group. Hence, each process receives packets from n processes as shown in Figure 3. Each process can share the multiple resequencing buffers for each process. If the desired transmission rates of each process is fracuated, the processes set the buffer size in proportion to the transmission rates.

In the beginning of a transmission, the resequencing buffer size is shared equally for each process, i.e. RSB/N for each process. After that, each process can change the buffer sizes for each process by reporting the desired transmission rates.

5 Concluding Remarks

In this paper, we presented the way to share both the retransmission and resequencing buffers by using bi-directional channel. This method is suitable for two-way communications in the presence of desired transmission rate changes. In addition, we presented

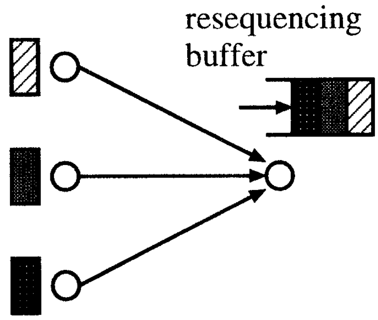

Figure 3: Packets buffering from multiple sender.

the way to share the resequencing buffer for each process in group communications.

References

[1] Cam, R. and Leung, C., "Throughput Analysis of Some ARQ Protocols in the Presence of Feedback Errors," *IEEE Trans. on Communication*, Vol. 45, No. 1, 1997, pp. 35–44.

[2] Chang, J. and Yang, T., "End-toEnd Delay of an Adaptive Selective Repeat ARQ Protocol," *IEEE Trans. on Communication*, Vol. 42, No. 11, 1994, pp. 2926–2933.

[3] Djuknic, G. and Schilling, D., "Performance Analysis of an ARQ Transmission Scheme for Meteor Burst Communications," *IEEE Trans. on Communication*, Vol. 42, No. 2/3/4, 1994, pp. 268–271.

[4] Kallel, S. and Leung, C., "Efficient ARQ Schemes with Multiple Copy Decoding," *IEEE Trans. on Communication*, Vol. 40, No. 3, 1992, pp. 642–650.

[5] Kallel, S., "Efficient Hybrid ARQ Protocols with Adaptive Forward Error Correction," *IEEE Trans. on Communication*, Vol. 42, No. 2/3/4, 1994, pp. 281–289.

[6] Nakamura, A., Tachikawa, T., and Takizawa, M., "Group Communication Protocols: Properties and Evaluation," *Proc. of IEEE Workshop on Services in Distributed and Networked Environments (SDNE)*, 1994, pp.139−146.

[7] Sakakibara, K. and Kasahara, M., "Multicast Hybrid ARQ Scheme Using MDS Codes and GMD Decoding," *IEEE Trans. on Communication*, Vol. 43, No. 12, 1995, pp. 2933–2940.

[8] Tachikawa, T. and Takizawa, M., "Distributed Protocol for Selective Intra-group Communication," *Proc. of the 3rd IEEE International Conf. on Network Protocols (ICNP-95)*, 1995, pp.234−231.

[9] Yao, Y., "An Effective Go-Back-N ARQ Scheme for Variable-Error-Rate Channel," *IEEE Trans. on Communication*, Vol. 43, No. 1, 1995, pp. 20–23.

[10] Yoshimoto, M., Takine, T., Takahashi, Y., and Hasegawa, T. "Waiting Time and Queue Length Distributions for Go-Back-N and Selective-Repeat ARQ Protocols," *IEEE Trans. on Communication*, Vol. 41, No. 11, 1993, pp. 1687–1693.

[11] Wang, J. and Silvester, J., "Optimal Adaptive Multireceiver ARQ Protocol," *IEEE Trans. on Communication*, Vol. 41, No. 12, 1993, pp. 1816–1829.