

Performance Evaluation of Some Dynamic ARQ Schemes for Multichannel Communication System

Chun-Xiang Chen*, Janfei Ai, Yuto Izumi, and Shiho Okusako

Prefectural University of Hiroshima, Hiroshima City, Japan

* Corresponding author's Email: chen@pu-hiroshima.ac.jp

Abstract: In this paper, we propose two dynamic error control schemes and apply them to the multichannel data communication system. The proposed schemes are based on the basic Stop-and-Wait ARQ and Selective Repeat ARQ protocols, The former and latter are used for the error controls in a round-trip time and among the multiple channels in a slot, respectively. The state of each channel is assumed as Markov chain with two states. The operations of the proposed schemes are described and their performances are evaluated through computer simulation. The numerical results show that the performances of the proposed schemes are improved remarkably comparing with the traditional scheme.

Keywords: Dynamic error control; Hybrid ARQ scheme; Markov process; System throughput; Resequencing buffer;

1. Introduction

ARQ (automatic repeat request) [1]-[3] and FEC (forward error correction) [4] and hybrid FEC/ARQ [10, 14] are widely used for error controls in data communication systems. Among these schemes, ARQ and its varied schemes are applied in many communication systems due to their simplicity and high reliability.

There are three classical ARQ schemes[1]-[3]: Stop-and-Wait (SW) ARQ, Go-Back- N (GBN) ARQ and Selective-Repeat (SR) ARQ. The principle of ARQ is that the transmitter sends packets with embedded error-detectable code to the receiver, and retransmits the packets negatively acknowledged (NACK) by the receiver. In SW ARQ, the transmitter sends out a packet, then stops the next transmission until the corresponding acknowledgment (NACK or ACK) arrives. When an NACK is received, the transmitter retransmits the corresponding packet till it is successfully received in the receiver; on the contrary, when a positive acknowledgment (ACK) packet is received, the

transmitter sends the following packet and so forth. In GBN ARQ, the transmitter sends packets continuously and retransmits the NACK-ed packet and all the following packets transmitted in the round-trip time, while in SR ARQ, the transmitter only retransmits the NACK-ed packet. Compared with SW ARQ and SR ARQ schemes, the implementation of GBN ARQ scheme is simpler than that of SR ARQ scheme in which re-ordering control at the receiver is indispensable, and the throughput performance is better than that of SW ARQ scheme in the environment in which the round trip time (RTT) is relatively longer and the error rate is lower (such as in wired data transmission system).

However, in wireless communication system, the wireless link is very vulnerable and easily interfered by environment noise. This means that error rate of the channel is relatively high; error occurs with large burstness and ARQ scheme is favorable in such environment. Furthermore, in CSMA/CA protocol [16], the

transmitter and the receiver share a common channel (or channels) in time division manner, SW ARQ scheme is adopted as the error control protocol in data link layer [11].

To enlarge the bandwidth of channels in a communication system, utilizing multiple parallel channels as a single link has become quite popular. MIMO (Multiple Input Multiple Output) [12], channel bonding adopted by IEEE 802.11n or link aggregation in an Ethernet environment can be considered as such kind of applications. Chen et al. proposed a channel grouping method [9] and packet-to-channel assignment [6] on a GBN ARQ scheme in the multiple-channel system. Ding et al. proposed a channel assignment rule of ARQ protocols for multiple channels and analyzed the throughput [7].

Analysis of the performances (including throughput, packet delay, resequencing delay or buffer occupancy at the receiver etc.) on error control protocols depends largely on operation of the protocol, the number of channels and error pattern (channel states). In general, if the analytical model of the system is precise to the practical system, the performance analysis becomes complicated. In the previous works, performances of the multiple-channel system have been investigated in the random-error environment. Nagaoka et al. [8] analyzed the throughput efficiency of the multiple-channel system with burst errors. However, they did not consider the influence of the state transition of channels on the channel assignment (packet-to-channel) rule. Chen et al. [15] analyzed the throughput efficiency of GBN ARQ scheme under taking the states of the channels into account, however, they only considered the system with two channels.

In this paper, we focus on the communication system with multiple parallel channels, in which SR ARQ scheme is employed as the error control scheme within one slot and SW ARQ is used for the error control in a round trip time (RTT). We propose two dynamic ARQ schemes and evaluate their performances by computer simulation. The effectiveness of the proposed schemes is clarified by numerical results.

The rest of this paper is organized as follows. Section 2. describes the channel models. The proposed dynamic ARQ schemes and the operations are described in Section 2.2. The performance evaluation is illustrated in Section 3., and conclusion remarks are given in Section 4.

2. System Model and Operations

The system model consists of a transmitter and a receiver connected with M parallel forwarding channels (numbered $0, 1, \dots, M-1$) and a feedback channel. The transmitter is able to transmit one packet on each channel in a slot in parallel (as a result, M packets are transmitted in a slot), and then stops the continuous transmission until the ACK (or NAK) arrives. The *slot* is defined as the time interval in which a packet is transmitted over a channel. Before we describe the proposed error control schemes, we give the assumption of the channel state because the operations of the proposed schemes are largely dependent on the states of the channels.

2.1 Transition of channel states

On each channel, packet errors occur according to a Markov process with two states (i.e. states 0 and 1). States 0 and 1 represent error-free state and erroneous state, respectively (Fig. 1). The transition probabilities from state 0 to 1 and from state 1 to 0 are assumed to be p_i and q_i for channel i , respectively, $i = 0, 1, 2, \dots, M-1$. Therefore, the m -step state-transition probability of channel i from state α to β is given by Eq. (1) ([5, 8]).

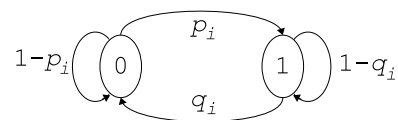


Figure 1 Transition of channel state

$$t(i)_{\alpha,\beta}^{(m)} = \frac{\beta p_i + (1-\beta)q_i}{p_i + q_i} + \frac{(-1)^{\alpha-\beta}((1-\alpha)p_i + \alpha q_i)\delta_i^m}{p_i + q_i}, \quad (1)$$

where $\alpha, \beta = 0$ or 1 , and δ_i is the *decay factor* defined by

$$\delta_i = 1 - p_i - q_i, \quad (-1 < \delta_i < 1). \quad (2)$$

If $\delta_i = 0$ ($p_i + q_i = 1$), it means that the channel states 0 and 1 are independent without burstiness. Also, the states of the channels are assumed to be independent without correlation.

2.2 Operation at the transmitter

In this section, we describe the operations at the transmitter under the static ARQ scheme (traditional

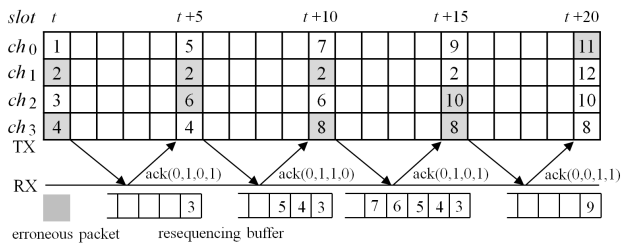


Figure 2 An example of the operations in the static ARQ over multiple parallel channels ($M = 4, r = 4$)

one) and two dynamic ARQ schemes proposed in this paper, called *dynamic ARQ-I* and *dynamic ARQ-II*, respectively.

At the transmitter, packets waiting for transmission are numbered in sequence. We assume that at least M packets are always ready to be transmitted in the transmission buffer and that the round-trip time from the end of the slot of the transmission to the completion of receiving the corresponding acknowledgment packet, is r slots.

1) At the beginning of a slot (referring to slot t , Fig. 2), the transmitter sends one packet on each channel (i.e. channel $0, 1, \dots, M-1$) in parallel according to the packet-to-channel schemes referring to static ARQ scheme (*Static ARQ*), dynamic ARQ with theoretical prediction (*Dynamic ARQ-I*), and dynamic ARQ with former-state prediction (*Dynamic ARQ-II*). After the completion of the transmission at the end of slot t , the transmitter stops the successive transmission of the following packets to wait the acknowledgment being sent back by the receiver.

2) The transmitter receives the acknowledgment packet after r slots (one round-trip time) of a transmission, and collates the acknowledgment packet to determine whether the packets have been successfully received or not. It deletes the packets successfully received from the transmission buffer and composes the M packets to be transmitted in next slot, according to the packet-to-channel scheme. We assume that the collation and the composition complete simultaneously with the end of receiving the acknowledgment packet.

3) The transmitter composes the M packets and sends them on each channel according to the following schemes.

In the *static ARQ* scheme (Fig. 2), when the transmitter receives the acknowledgment packet, it deletes the packets correctly received at the receiver from the transmission buffer, and composes the M packets of the packets received erroneously and of the new packets in sequence which are not transmitted to date. It

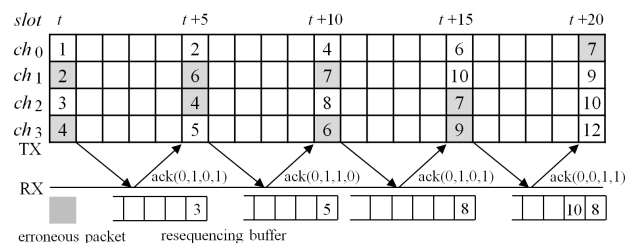


Figure 3 An example of the operations in the dynamic ARQ-I over multiple parallel channels ($M = 4, r = 4$)

sends the M packets on each channel, where the packet being retransmitted is sent on the channel same as it has been sent on at the first time, and the packet being transmitted first time is sent on the other channel. Fig. 2 illustrates an example of the operation in the static ARQ scheme, where $M = 4$ and $r = 4$. $\text{ack}(s_0, s_1, s_2, \dots, s_{M-1})$ is the acknowledgment packet, where $s_i = 0$ (or $s_i = 1$) means that the packet transmitted on channel i has been received correctly (or in error), $i = 0, 1, 2, \dots, M-1$. The occupancy of the resequencing buffer at the receiver shows the number of the packets stored temporarily for resequencing since the receiver must deliver the packets to the application in the original order.

In the *dynamic ARQ-I* (or *ARQ-II*) scheme, the transmitter takes the states of the channels into account when it retransmits the packets received erroneously at the receiver. The principle of the dynamic ARQ scheme is to transmit the packet with the smallest sequencing number over the channel in the best state (i.e. state 0) and the packet with the second smallest number over the channel in the second best state and so forth. It is noted that the states of the channels do not vary with the error control schemes. The throughput efficiency is not affected by the dynamic schemes, but the resequencing delay or the buffer occupancy at the receiver should be.

In the *dynamic ARQ-I* scheme, we assume that the transmitter knows of the state-transition probability of each channel according to the channel model (formula (1)). When it receives the acknowledgment packet which includes the ACKs and/or NAKs about the M packets transmitted before a round-trip time (r slots), the transmitter exactly knows the states of the M channels at that time. Now the transmitter predicts the probability of state 0 for each channel in the following slot by calculating the state-transition probability $t(i)_{\alpha,0}^{(r+1)}$ by formula (1), where $i = 0, 1, 2, \dots, M-1$ and $\alpha = 0$ if the packet transmitted on channel i is received correctly, otherwise $\alpha = 1$. The transmitter

sorts the M channels according to $t(i)_{\alpha,0}^{(r+1)}$ in the descending order and consists of a channel set Q ; which is that the channel with the biggest $t(i)_{\alpha,0}^{(r+1)}$ is put at the first in the Q , and the channel with the second biggest $t(i)_{\alpha,0}^{(r+1)}$ is put at the second in the Q and so forth. The transmitter transmits the M packets in parallel as follows. The packet with the smallest sequence number among the M packets is transmitted on the first channel in the Q and the packet with the second smallest sequence number is transmitted on the second channel in the Q and so forth.

Fig. 3 shows an example of the operations in the dynamic ARQ-I scheme, where $M = 4$ and $r = 4$. At the beginning of slot t , the order of the channels in Q is assumed to be $\{ch_0, ch_1, ch_2, ch_3\}$, packets 1, 2, 3, 4 are transmitted on the channels in the order of the channels in Q . The transmitter receives the acknowledgment packet ($ack(0, 1, 0, 1)$) in slot $t + 4$. $ack(0, 1, 0, 1)$ notices the transmitter that the packets transmitted over channel 0 and channel 2 have received correctly and the packets transmitted over channel 1 and channel 3 have received in error. This means that channel 0 and channel 2 were in state 0 (good state) and channel 1 and channel 3 in state 1 (erroneous state) in that slot. Therefore, in this example, the transmitter simply calculates the probability of state 0 for each channel in the following slot, i.e. $t(0)_{0,0}^{(5)}$, $t(1)_{1,0}^{(5)}$, $t(2)_{0,0}^{(5)}$, $t(3)_{1,0}^{(5)}$ in slot $t + 5$ and makes up Q as $Q = \{ch_0, ch_2, ch_3, ch_1\}$. Therefore, packets 2, 4, 5, 6 are transmitted on channels 0, 2, 3 and 1 in slot $t + 5$, one per channel, respectively. At the beginning of slot $t + 10$, the transmitter calculates the probability of state 0 for each channel in the following slot, i.e. $t(0)_{0,0}^{(5)}$, $t(1)_{1,0}^{(5)}$, $t(2)_{1,0}^{(5)}$, $t(3)_{0,0}^{(5)}$ and makes up Q as $Q = \{ch_0, ch_3, ch_1, ch_2\}$, packets 4, 6, 7, 8 are transmitted on channels 0, 3, 1, 2, one per channel, respectively, and so forth. The occupancy of the resequencing buffer is listed at the bottom of the figure.

In the dynamic ARQ-I scheme, the transmitter is supposed to have some known information about the channel state. It can get the information in some way from the past acknowledgment packets. However, it is not always easy to do so. Taking this into account, the proposed dynamic ARQ-II scheme predicts the states of the channels just based on the latest acknowledgment packet. When the transmitter receives the acknowledgment packet which includes the ACKs and/or NAKs about the M packets transmitted before a round-trip time (r slots), the transmitter exactly knows the states of the M channels at that time. The states of the channels in the following slot are supposed to be

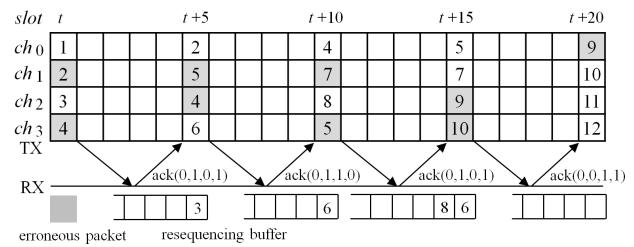


Figure 4 An example of the operations in the dynamic ARQ-II over multiple parallel channels ($M = 4, r = 4$)

the same states in the slot of the preceding transmission. The transmitter sorts the M channels according to the state of the channel in the order of state 0 and state 1, and makes up the channel set Q . The channels in state 0 are put in the head of Q , and the channels in states 1 in the rear of Q while the channels in the same state are selected randomly. Finally, the transmitter sends the M packets on each channel in the order of the channels in Q .

Fig. 4 illustrates an example of the operations in the dynamic ARQ-II scheme, where $M = 4$ and $r = 4$. At the beginning of slot t , the order of the channels is $Q = \{ch_0, ch_1, ch_2, ch_3\}$, packets 1, 2, 3, 4 are transmitted on the channels in the order of the channels in Q . The transmitter receives the acknowledgment packet ($ack(0, 1, 0, 1)$) in slot $t + 4$, it knows that channels 0 and 2 are in state 0 and channels 1 and 3 in state 1. Q is composed as $Q = \{ch_0, ch_2, ch_1, ch_3\}$. Packets 2, 4, 5, 6 are transmitted on each channel in the order of the channels in Q in slot $t + 5$. $Q = \{ch_0, ch_3, ch_1, ch_2\}$ in slot $t + 10$, packets 4, 5, 7, 8 are transmitted on channels 0, 3, 1, 2, one per channel, and so forth.

2.3 Operation at the receiver

When the packet crosses a channel, the packet error occurs with some probability corresponding to the channel model described in Section 2.1. When the packets arrive at the receiver after a half round-trip time from the end of the transmission at the transmitter, the receiver receives these packets, collates the contents in the packet with the error-detection code (i.e. checksum) and composes of the acknowledgment packet as the format $ack(s_0, s_1, s_2, \dots, s_{M-1})$. If the packet error occurs in the packet transmitted over the channel i , $i = 0, 1, 2, \dots, M - 1$, s_i is set to be 1; otherwise $s_i = 0$.

At the receiver, the selective-repeat ARQ scheme is employed for the error controls among the multiple-parallel channels. The receiver deletes all the packets

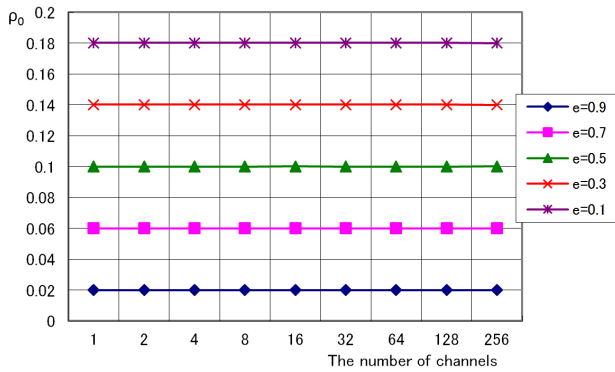


Figure 5 The normalized throughput ρ_0 ($r = 4, \delta = 0$)

received erroneously, and delivers the *ordered packets* from the resequencing buffer to the upper layer. The ordered packet is the packet which is received correctly and all the preceding packets with smaller sequencing number have been received correctly. On the contrary, the *out-of-order* packet is a correctly received packet while at least one packet with smaller sequencing number has not been received correctly. The out-of-order packet is stored in the resequencing buffer for re-ordering. The number of the out-of-order packets in the resequencing buffer is shown at the bottom of Figures 2, 3 and 4.

3. Simulation Results

In this section, we investigate the performances of the ARQ error control schemes proposed in this paper through computer simulation. Here the performances are evaluated by these indexes: throughput efficiency, average (or maximum) number of the packets in resequencing buffer (i.e. the average (or maximum) occupancy of the resequencing buffer) and average sojourn delay of the packet in the resequencing buffer. The *throughput efficiency* (ρ) is defined as the long-term ratio of the number of packets successfully accepted by the receiver to the total number of transmission slots. We also define the normalized throughput efficiency (ρ_0) as $\rho_0 = \frac{\rho}{M}$.

In the following numerical results, we assume that $p_i = p$ and $q_i = q$ for convenience (i.e. $\delta_i = \delta = 1 - p - q$). The average error rate e on each channel is given by $e = \frac{p}{p+q}$. At this assumption, ρ_0 is theoretically obtained as $\rho_0 = \frac{1-e}{(r+1)}$. Fig. 5 shows an example of the normalized throughput ρ_0 . It is clear that ρ_0 does not vary with the number of channels, and ρ_0 has no difference within the three ARQ schemes. It is true because the packets received correctly are accepted without discard at the receiver for any one of

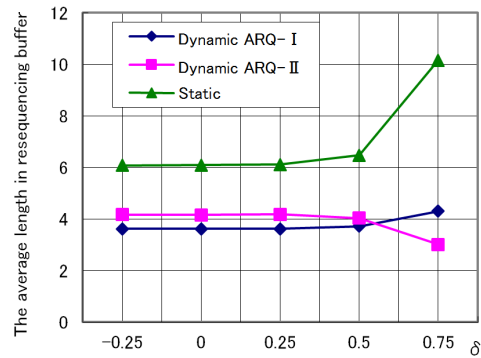


Figure 6 The average number of packets in resequencing buffer ($r = 4, e = 0.2, M = 10$)

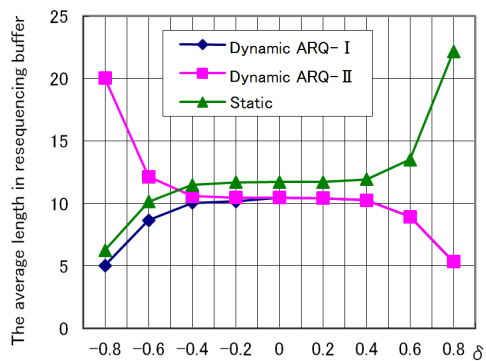


Figure 7 The average number of packets in resequencing buffer ($r = 4, e = 0.5, M = 10$)

the ARQ schemes.

Figures 6 and 7 illustrates the relationship between the average number of packets in the resequencing buffer (shown by q_{ave}) and the decay factor δ , where $r = 4$ and $M = 10$ and e is assumed to be constant. When δ increases (i.e. $p + q$ becomes smaller), the sojourn time that channel state stays in the same state becomes longer [15]. Note that the longer the sojourn time in state 1, the larger the error burstiness of the channel. From these figures, it is clear that q_{ave} significantly depends on the error control scheme although the average error rate e is constant. q_{ave} in the static ARQ scheme increases with increment of δ especially in the interval of $0.5 < \delta < 1$. q_{ave} in the dynamic ARQ-I scheme is almost same as that in the dynamic ARQ-II scheme. However, in the interval of $-1 < \delta < -0.5$, q_{ave} in the dynamic ARQ-II scheme is remarkably larger than it in the other schemes. It can be explained as follows. When δ increases negatively, p and q becomes larger; the state of the channel transits frequently between states 0 and 1. This is contrary to the premise of the dynamic ARQ-II scheme that the state of the channel changes slowly.

Figures 8 and 9 illustrate the relationship between the average number of the packets (q_{ave}) in the resequencing

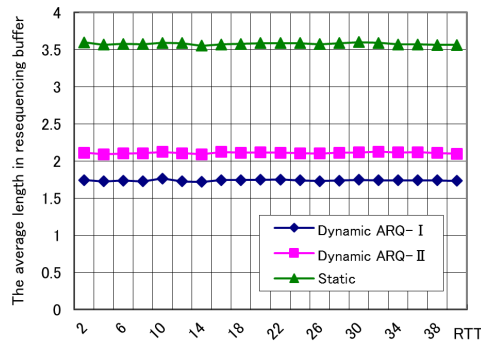


Figure 8 The average number of packet in resequencing buffer ($\delta = 0, e = 0.1$)

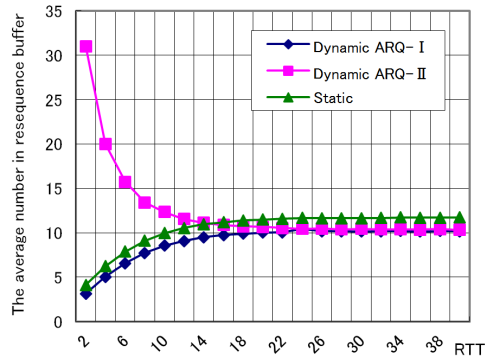


Figure 9 The average number of packets in resequencing buffer ($\delta = -0.8, e = 0.5, M = 10$)

quencing buffer and the round-trip time (rtt). The numerical results shows that q_{ave} is almost a constant in each error control scheme when $\delta = 0$, and q_{ave} of the static ARQ scheme is the longest and q_{ave} of the dynamic ARQ-I is the shortest. On the another hand, at $\delta = -0.8$ and $e = 0.5$ (relatively high, Fig. 9), q_{ave} of the dynamic ARQ-II scheme increases with the decrement of r when r is relatively small. When r becomes larger ($r > 14$), q_{ave} almost does not vary with r ; the difference in q_{ave} among these schemes is insignificant.

4. Concluding Remarks

This paper proposes two dynamic ARQ schemes considering the state transition of the channels for the multiple channel systems. The proposed scheme is a hybrid ARQ combined the SW ARQ scheme with the SR ARQ scheme. The core idea of the proposed schemes is that, the transmitter sends the packet with the smallest sequence number on the channel in the best state.

We investigated the throughput efficiency and the average number of the packets in the resequencing buffer for every scheme. The results can be simply summarized as follows. q_{ave} in the static ARQ scheme

is largest especially when δ is relatively high ($\delta > 0.4$). On the contrary, the dynamic ARQ-II scheme shows the longest q_{ave} when $\delta < -4$. The dynamic ARQ-I is adaptively respond to δ in the all intervals and gives the best performances.

Acknowledgments

This work was partially supported by JSPS KAKENHI, Grant-in-Aid for Scientific Research (22500066).

References

- [1] S. Lin, D. J. Costello, and M. J. Miller, "Automatic-Repeat-request error control scheme," *IEEE Commun. Mag.*, vol. 22, no.12, pp.5-17, Dec. 1984.
- [2] H. Brunnel and M. Moeneclaey, "On the throughput performance of some continuous ARQ strategies with repeated Transmissions," *IEEE Trans. Commun.*, vol.COM-34, no.3, pp.244-249, Mar. 1986.
- [3] G. Benelli, "A selective ARQ scheme protocol with a finite-length buffer," *IEEE Trans. Commun.*, vol.41, no.7, pp.1102-1111, July 1993.
- [4] T. Mizuochi, "Recent Progress in Forward Error Correction for Optical Communication System," *IEICE Trans. Commun.*, vol.E88-B no.5, pp.1934-1946, May 2005.
- [5] M. Komatsu, C.-X. Chen, and K. Kinoshita, "Traffic analysis of the stop-and-wait ARQ over Markov error channel," *IEICE Trans. Commun.*, vol. E77-B, no.4, pp.477-484, April 1994.
- [6] C.-X. Chen, M. Komatsu, and K. Kinoshita, "Performance analysis of go-back-N ARQ scheme on parallel Channels system," *IEICE Trans. Commun.(Japanese Edition)*, vol.J80-B-I, no.4, pp.169-178, April 1997.
- [7] Z. Ding and M. Rice, "Throughput analysis of ARQ protocol for parallel multichannel communications," *Proc. Of IEEE Globecom 2005*, pp.1279-1283, 2005.
- [8] K. Nagaoka, C.-X. Chen, and M. Komatsu, "Throughput efficiency of go-back-N ARQ protocol on Parallel Multi-Channel with burst errors," *IEICE Trans. on Commun.*, vol.E91-B, no.12, pp.3994-3997, Dec. 2008.
- [9] C.-X. Chen, M. Komatsu, and K. Kinoshita, "Channel-grouping methods on go-back-N ARQ scheme in multiple-parallel-channel system,"

IEICE Trans. Commun., vol.E77-B, no.2, pp265-269, Feb. 1994.

- [10] Y.-H. Jung, "MIMO Common Feedback Method for Multicast H-ARQ," *IEICE Trans. Commun.*, vol. E93-B, no.1, pp.222-225, Jan. 2010.
- [11] T. Hu., D. Afshartous, and G. Young, "Parallel Stop and Wait ARQ in UMTS - Performance and Modeling," *Proceedings of the 2004 World Wireless Congress 2004*, San Francisco, CA, May 2004.
- [12] Z. Ding, "ARQ Techniques for MIMO Communication Systems," Ph D., *Dept. of Elec. Computer Eng., Brigham Young University*, U T. 2006
- [13] W. Luo, K. Balachandran, S. Nanda, and K. K. Chang, "Delay analysis of selective-repeat ARQ with applications to link adaptation in wireless packet data systems," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1017-1029, May 2005.
- [14] A. M. Cipriano, P. Gagneur, G. Vivier, and S. Sezginer, "Overview of ARQ and HARQ in Beyond 3G Systems," *2010 IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops*, pp.424-429, Sept. 2010.
- [15] C.-X. Chen, K. Nagaoka, and M. Komatsu, "Adaptive Go-Back- N ARQ Protocol over Two Parallel Channels with Slow State Transition," *IEICE Trans. Fundamentals*, vol.E94-A, no.12, Dec. 2011.
- [16] IEEE, "Part II : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11TM-2007*, June 2007.