PAPER

New Go-Back-*N* ARQ Protocols for Point-to-Multipoint Communications

Hui ZHAO†, Toru SATO† *and* Iwane KIMURA†, *Members*

SUMMARY This paper presents new go-back-N ARQ protocols for point-to-multipoint communications over broadcast channels such as satellite or broadcast radio channels. In the conventional go-back-N ARQ protocols for multidestination communications, usually only error detection codes are used for error detection and m copies of a frame are transmitted at a time. In one of our protocols, a bit-by-bit majority-voting decoder based on all of the m copies of a frame is used to recover the transmitted frame. In another protocol, a hybrid-ARQ protocol, which is an error detection code concatenated with a rate repetition convolutional code with the Viterbi decoding, is used. In these protocols, a dynamic programming technique is used to select the optimal number of copies of a frame to be transmitted at a time. The optimal number is determined by round trip propagation delay of the channel, the error probability, and the number of receivers that have not yet received the message. Analytic expressions are derived for the throughput efficiency of the proposed protocols. The proposed point-to-multipoint protocols provide satisfactory throughput efficiency and perform considerably better than the conventional protocols under high error rate conditions, especially in environments with a large number of receivers and large link round trips. In this paper we analyze the performances of the proposed protocols upon the random error channel conditions.
key words: *communication theory, satellite communication, point-to-multipoint communication, error detection codes, go-back-N ARQ*

1. Introduction

Point-to-multipoint communications have become more and more popular in applications, such as file distributions, video text transmissions, and the teleconferencing. Comparing to the studies of point-to-point protocols, where many efficient hybrid-ARQ protocols have been proposed and analyzed [1]–[8], most of the studies of point-to-multipoint communications are focused on error detection code protocols [9]–[14]. Among these multidestination protocols, go-back-N ARQ protocols are the most commonly used protocols. The main idea in these protocols is to send multiple copies of a message instead of just sending one copy of a message and some strategies have been studied. Gopal and Jaffe [9] studied three different go-back-N ARQ schemes for point-to-multipoint communications. The three schemes differ in the way that a sender uses different size of memory for storing the ACK/NACKs to reduce the transmission copies of a message. Wang and Silvester [11] improved their schemes by that the optimal number of copies is selected which depends not only on the round trip propagation delay and transmission error probability, but on the number of receivers that have not yet successfully received the data frame. Lee [14] proposed a slightly different scheme with Wang and Silvester and analyzed three situations for transmission copies m ($\leqq N$, $> N$ and $= \infty$). Among the go-back-N ARQ protocols for multidestination communications [9]–[14], the Wang and Silvester's protocol [11] has the best performance.

Hybrid select-repeat ARQ protocols have been studied in multidestination transmission systems [15]–[17], but in the conventional go-back-N ARQ multidestination protocols, only error detection code is used. Therefore the throughput efficiency is low at high channel error rates and these protocols can only be used in a low channel-error probability. In communication environments where the channel error rate is high, the link round-trip delay is large, and where there are a large number of receivers, such as satellite broadcast links, the throughput efficiency of the conventional go-back-N ARQ protocols for multidestination transmissions becomes unacceptable. In conventional go-back-N ARQ multidestination protocols, only error detection codes are used. As the channel error rate becomes higher, the errors will be detected more often, and therefore more retransmissions will be required. As the number of receivers increases, the probability of the errors detected by all the receivers will increase. As the link-trip delay increases, the retransmission bits will increase if errors are detected. All these factors decrease the throughput efficiency. Especially in the USAT (Ultra Small Aperture Terminal) system, reduction of the transmitter power and the antenna aperture is essential, and thus the development of the protocols for high error-rate conditions will becomes more and more important.

We assume that the channel is in the random error condition, because burst errors can be converted into random errors by employing a proper interleaving technique under most of practical conditions. Characteristics of burst error channels without interleaving are often quite different from those of random error channels, and are usually treated separately using a two-state Markov model ([18] and [19], [9] and [20], and [21] and [22]).

In this paper, we present and analyze three go-back-

Manuscript received August 20, 1993.
Manuscript revised January 14, 1994.
†The authors are with the Faculty of Engineering, Kyoto University, Kyoto-shi, 606-01 Japan.

N ARQ protocols for point-to-multipoint communications that can be implemented when receivers contain finite size buffers. These protocols are based on the protocols discussed by Wang and Silvester[11]. In Wang and Silvester's protocols, only error detection code is used and m copies of a frame are sent to the receivers at a time. In a receiver end, a receiver detects each copy of a frame and sends back an ACK/NACK according to the results of decoding on each copy. In our protocol 1, m copies of a frame are transmitted at a time. At a receiver end, instead of sending back an ACK/NACK upon detecting each copy of a frame correctly or erroneously as in Wang and Silvester's protocols, m copies of a frame are checked for errors one by one, an ACK is sent upon receiving at least one copy correctly and a NACK is sent upon receiving all the m copies erroneously. In protocol 2, we improve the protocol 1 by that, all m copies of a frame are first checked for errors as in the protocol 1, and the erroneous copies are not discarded but are stored in a buffer. If all of the m copies are detected erroneously, a bit-by-bit majority-voting decoding based on all the m copies is used to recover the transmitted frame and then error detection is made on this recovered message. An ACK/NACK is sent back depending on whether the recovered message is correct or not. In protocol 3, a hybrid ARQ protocol, which is an error detection code concatenated with an error correction code (rate repetition convolutional code with the Viterbi decoding), is used. The m received sequences are first decoded by the Viterbi decoder and the decoded sequence is then checked for errors. An ACK/NACK is sent back depending on whether this sequence is correctly decoded or not. A dynamic programming technique is used to select the optimal number of copies of a frame to be transmitted at a time.

The proposed protocols are described in Sect. 2. The analyses of throughput efficiency of the proposed protocols are given in Sect. 3. Numerical results and comparisons with conventional protocols are given in Sect. 4.

2. The Proposed Protocols

The broadcast communication environment we consider consists of $r+1$ stations, one being the transmitter and the other r being receivers. The communication between the sender and receivers is over a broadcast channel such as satellite or broadcast radio channels. Data messages are sent in fixed length data frames and time is segmented into fixed length slots whose duration is equal to the transmission time of a data frame. Each transmission frame carries a sequence number, information of the number of copies of a frame to be transmitted in the transmitted packet sequence, and a data frame. Each data frame includes a cyclic redundancy code (CRC) at the end to enable the receivers to detect errors. When a message is ready for transmission, it is numbered and is encoded into a frame by a CRC encoder. The feedback channel is assumed to be error free so that a receiver can reliably inform the transmitter whether it has successfully decoded a frame or not. The return frame from a receiver carries a sequence number, an ACK/NACK, and the channel bit error rate (BER) information. We also assume that there is always a data frame waiting to be transmitted at the sender and the nodal processing time is negligible.

Suppose the number of blocks that can be transmitted during the round-trip delay between the transmitter and the ith receiver is N_i $(i=1,\ldots,r)$, and for simplicity, we assume $N_i = N$ $(i=1,\ldots,r)$. We also assume that $N > m$ and this is the real case for satellite communications. The broadcast channels are supposed to be the random error channels.

We now describe the go-back-N ARQ protocols that are suitable for point-to-multipoint communications over broadcast channels.

2.1 Protocol 1

This protocol is similar to Wang and Silvester's scheme[11]. At the transmission end, the transmitter continuously sends messages to receivers (sends m copies of a frame at a time) and starts a timeout clock for the m copies of the frame. In each frame, the sequence number of the frame and the number of copies m are added in the front of the frame. The number of copies m depends on the number of receivers K that has not yet successfully received the message, the channel error probability ε, and the round trip delay N. We hereafter denote m as $m(K)$ in order to explicitly show that m may be changed depending on K. The optimal number of copies is denoted by $m^*(K)$. A dynamic programming technique is used to solve the optimization problem. In the meantime, the transmitter keeps listening ACK/NACKs for previous frames to keep track of those receivers that have not yet received message correctly. A frame-outstanding list for a data frame is updated when an ACK for that frame is received. If ACKs of a data frame from all the receivers are arrived at the frame-outstanding list, the sender stops the timeout clock associated with that data frame. Otherwise, after the timeout occurs and the sender has not received all ACKs, the sender goes back to the unsuccessful data frame and retransmits it and all data frames which follow it in the transmitting sequence.

At the receiver end, upon received a copy of a frame, a receiver first checks the sequence number of the frame. If the sequence number of the frame is bigger or smaller than the one the receiver expected, this frame is discards. Because when the sequence number is smaller than the expected one, the frame is the one which has been correctly received by the receiver and when the sequence number is bigger than the expected one, the

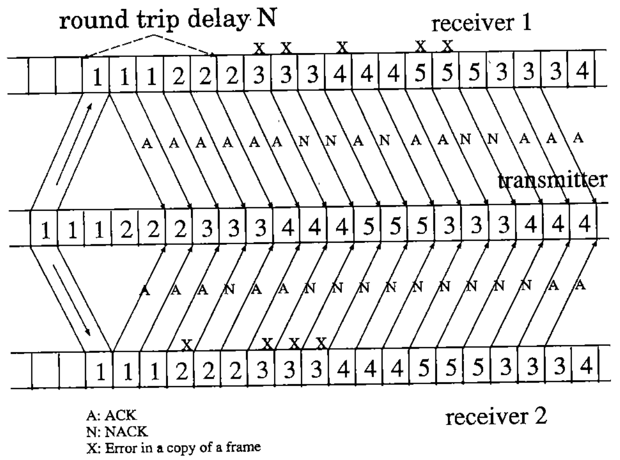


Fig. 1 ACKs/NACKs transmissions in Wang and Silvester's protocol for $N = 5$, $r = 2$, and $m(K) = 3$.

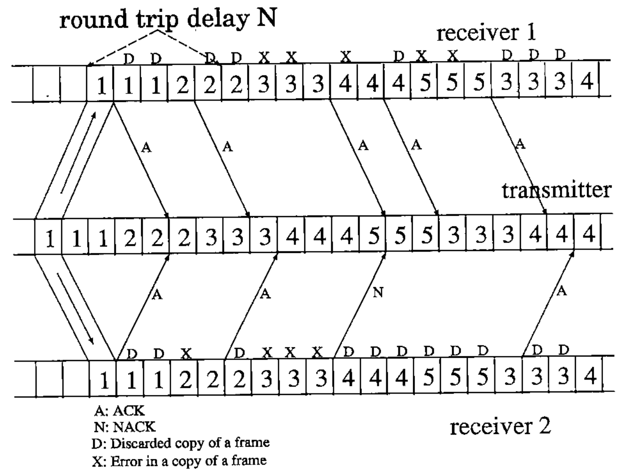


Fig. 2 ACKs/NACKs transmissions in Protocol 1 for $N = 5$, $r = 2$, and $m(K) = 3$.

receiver has detected errors in the expected frame, and therefore discards subsequent frames. If the sequence number is the one the receiver expected, then the receiver checks the number of copies $m(K)$ of a frame and error detecting is applied to the $m(K)$ copies of the frame. If no error is detected in a copy of the frame, the sequence number of this error free frame is stored in a memory, an ACK together with the sequence number and the channel BER information is sent back, and the information data is delivered to the destination. If this copy of the frame is erroneous, the receiver discards the erroneous copy and checks the next received copy. If the next copy of the frame is also erroneous, the receiver discards this erroneous copy also and checks another received copy of the frame. Continue this process till $m(K)$-th copies. If all the $m(K)$ copies of a frame are erroneous, a NACK together with sequence number and channel BER information is sent back and this frame and subsequence frames with a frame number bigger than that of the erroneous frame are discarded. The transmitter sends out another $m(K)$ copies of the message. This process continues until all r receivers acknowledge a message.

The difference between our protocol 1 and the Wang and Silvester's scheme[11] is that, in Wang and Silvester's scheme, a receiver detects each copy of a frame and sends back an ACK/NACK according to the results of decoding each copy, as shown in Fig. 1, while in our protocol 1, $m(K)$ copies of a frame are checked for errors as a whole. An ACK is sent upon receiving at least one copy correctly and a NACK is sent upon receiving all the $m(K)$ copies erroneously, as shown in Fig. 2. Thus for $m(K)$ copies of a frame, only one ACK/NACK is sent back in our protocol, while $m(K)$ ACKs/NACKs are sent back in Wang and Silvester's protocols. Therefore, our protocol can reduce the number of transmission of ACK/NACK's, and can reduce the process time of the frame-outstanding list at the transmission end.

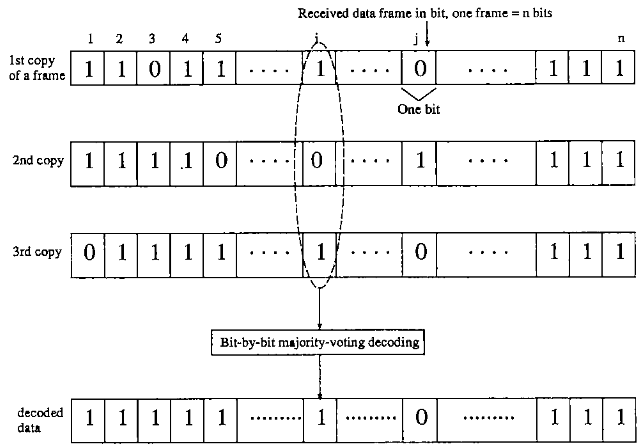


Fig. 3 Principle of bit-by-bit Majority-voting decoding.

2.2 Protocol 2

In this protocol, a bit-by-bit majority-voting decoder is used to recover the message. In Fig. 3, we show the principle of bit-by-bit majority-voting decoding with three copies of a frame. The three copies of the i-th bit are combined and decoded by the bit-by-bit major-voting decoder producing one bit of the decoded data. By doing this from $i = 1$ to n, we can get the decoded sequence. In this protocol, the system can be considered as an cascaded coding system: the bit-by-bit majority-voting decoder can be looked as an inner decoder and the CRC decoder as an outer decoder. The CRC bits are used for error detection only. No matter how the data have been decoded, a CRC decoder can detect errors in the data. If the number of parity check bits is z, the fraction of undetectable error patterns is only $1/2^z$ [23], regardless of the length of the code.

The transmitter operates in the same way as in protocol 1. At the receiver end, a receiver first checks the sequence number of a frame and the number of copies

Table 1 Free distances of two families of repetition codes issued from original code $(2,1,6)$ with generator polynomials $(133, 171)$.

R	Generator Polynomials	d_{free}	Generator Polynomials	d_{free}
1/2	(133, 171)	10	(133, 171)	10
1/3	(133, 171, 133)	13	(133, 171, 171)	12
1/4	(133, 171, 133.171)	20	(133, 171, 171, 133)	20
1/5	(133, 171, 133, 171, 133)	23	(133, 171, 171, 133, 171)	22

$m(K)$ a frame being transmitted as in protocol 1. If a copy of a frame is found erroneous, the receiver stores the erroneous copy in a buffer and checks the next copy of the frame. If the next copy of the frame is also erroneous, the receiver stores this erroneous copy also and checks next received copy of the frame. The error detection continues in this manner until $m(K)$-th copy. If the $m(K)$-th copy of the frame is also erroneous, the receiver combines the $m(K)$ copies of a frame and performs error correction by a bit-by-bit majority-voting decoder. The decoded sequence corresponding to information and parity check bits is then checked for errors. If the receiver still detected errors in the message, then it sends a NACK together with the message sequence number and the channel BER information and discards this frame and subsequent received frames. The other operations at the receiver end are same as with protocol 1.

2.3 Protocol 3

In protocol 3, a hybrid-ARQ protocol, which is a forward error correction code concatenated with the error detection code, is used. A rate repetition convolutional code with the Viterbi decoding is used as the forward error correction code.

We first explains the rate repetition convolutional code as following. Suppose S_1 and S_2 denote the two sequences formed at the output of the two adders of a 1/2 convolutional encoder. Successive repetitions of sequences S_1 and S_2 and operation of code combining yield a family of repetition codes of decreasing rates, $1/(2+i), i = 1, 2, \cdots$. Theses codes correspond to repetition codes of decreasing rates, obtained by alternately duplicating the two generator polynomials of the original rate 1/2 code $(G_1(X), G_2(X))$. Two possible families of repetition codes of decreasing rates can be obtained, depending on which of the two generator polynomials $G_1(X)$ or $G_2(X)$ is duplicated first. For example, with an original rate 1/2 code $(2,1,6)$, having generator polynomials $(133, 171)$ [24], the generator polynomials of the two possible families of repetition codes of decreasing rates $1/(2+i), i = 1, 2,$ and 3 are shown in Table 1.

By using a computer search procedure [2], the free distances of these repetition codes have been found, and are indicated in Table 1. As we can see from this table, repetition codes of rates 1/3 and 1/5, obtained by first duplicating the generator polynomial 133, have greater free distances than the ones obtained by first duplicating the generator polynomial 177. Clearly, in order to optimize the performance of the above hybrid ARQ strategy with code combining, one should select the best of the two families of repetition codes of decreasing rates, or, equivalently, order the two sequences S_1 and S_2 for successive repetitions, according to their contribution in the increase of the free distances of the resulting repetition codes. With the starting code of Table 1, sequence S_1 should be repeated first.

By using a computer search procedure [24], [25], the free distance d_{free} and the distance spectra $\{a_{d_{free}+j}\}$, $\{c_{free}+j\}$, $j = 0, 1, 2, \cdots$ of the best family of repetition codes of decreasing rates $1/(2+i)$, $i = 1, 3, 5, 7$, in the sense of a maximal free distance have been found for original rate 1/2 codes of memory $mem = 6$ [26], and are given in Table 2.

The terms $a_{d_{free}+j}, j = 0, 1, 2, \cdots$ refers to the number of incorrect paths at distance $d_{free}+j$, and $c_{d_{free}+j}$ refers to the number of information bit errors on these paths. As for the free distances and the distance spectra of repetition codes of rates $1/(2+i)$ with i even, they can be obtained from the distance spectra of the original rate 1/2 code by multiplying each distance path by i. A corresponding rate repetition Viterbi decoder is used to decode the encoded data at a receiver end.

At the transmission end, some error detection parity bits are first appended to the information bits. This frame of the information is then encoded with the rate 1/2 convolutional encoder. In protocols 1 and 2, $m(K)$ copies of a frame are sent to the receivers at a time. However, in this protocol repetitions alternate between the two sequences S_1 and S_2. More specifically, if the number $m(K)$ of repeats is equal to one, only the original frame is transmitted. If $m(K) = 2$, both sequences S_1 and S_2 are sent to the receivers. If $m(K) > 2$, and is even, then the two sequences S_1 and S_2 are sent equally $m(K)/2$ times each. On the other hand, if $m(K) > 2$, and is odd, then sequence S_1 is sent $(m(K)+1)/2$ times, and sequence S_2 is sent $(m(K)-1)/2$ times. The other operations are same as in protocol 1.

At the receiver end, a receiver first checks the sequence number of a frame. If the sequence number is not the expected one, discards this frame. Otherwise, the receiver checks the number of copies $m(K)$ in a frame as in

Table 2 Distance spectra of the best family of repetition codes issued from original code $(2,1,6)$ with generator polynomials $(133,171)$.

R	Generator Polynomials	d_{free}	$(a_{d_{free}+j}, j=0,1,2,\cdots,17)$ $[c_{d_{free}+j}, j=0,1,2,\cdots,17]$
1/2	(133, 171)	10	(11, 0, 38, 0, 193, 0, 1331, 0, 7275, 0, 40406, 0, 234969, 0, 1337714, 0, 7594819, 0,) [36, 0, 211, 0, 1404, 0, 11633, 0, 77433, 0, 502690, 0, 3322763, 0, 21292910, 0, 134365911, 0]
1/3	(133, 171, 133)	13	(1, 1, 5, 6, 9, 21, 35, 50, 88, 203, 339, 691, 1222, 2098, 4114, 7239, 12888, 23955) [3, 2, 19, 32, 45, 114, 257, 360, 760, 1784, 3035, 6956, 12828, 23104, 48546, 90460, 167376, 330104]
1/5	$(\cdots, 171, 133)$	23	(1, 1, 4, 3, 2, 4, 8, 15, 10, 6, 24, 41, 45, 54, 64, 153, 252, 321) [3, 2, 12, 10, 10, 28, 42, 72, 62, 42, 190, 292, 307, 404, 580, 1398, 2144, 2722]
1/7	$(\cdots, 171, 133)$	33	(1, 1, 4, 3, 1, 1, 1, 3, 8, 15, 10, 1, 0, 5, 24, 41, 44, 47) [3, 2, 12, 10, 3, 6, 7, 22, 42, 72, 62, 6, 0, 36, 190, 292, 302, 342]
1/9	$(\cdots, 171, 133)$	43	(1, 1, 4, 3, 1, 1, 0, 0, 1, 3, 8, 15, 10, 1, 0, 0, 0, 5) [3, 2, 12, 10, 3, 6, 0, 0, 7, 22, 42, 72, 62, 6, 0, 0, 0, 36]

Protocol 1. If $m(K)=1$, the rate repetition Viterbi decoder is not used and only outer error detection decoder is used to detect the errors in a frame. If $m(K)>1$, the $m(K)$ received sequences of a frame are first forming a sequence as issued from a rate $1/m(K)$ convolutional code [2]. A rate repetition Viterbi decoder is applied to this convolutional sequence. The decoded sequence, which corresponds to the information and parity check bits for error detection, is then examined for errors by the error detection decoder. The other operations at the receiver end are same as in protocol 1.

3. Throughput Efficiency

The throughput efficiency of an ARQ protocol is defined as the ratio of the number of information bits delivered to the total number of bits transmitted. Before transmission, every k bit information is encoded into n bits by an (n,k) CRC encoder. In the analysis, it is assumed that the probability of the undetected error is negligible.

We define the following terms.

ε: bit error rate (BER) of the binary symmetric channel.

p: the probability that a frame is received error-free by a receiver. Assuming that each frame has n bits, for protocol 1, we have

$$p=(1-\varepsilon)^n. \tag{1}$$

For protocol 2, because the bit-by-bit majority-voting decoding on the $m(K)$ copies of a frame is used, the bit error rate after decoding becomes

$$\varepsilon' = \begin{cases} \sum_{j=\frac{m(K)+1}{2}}^{m(K)} \binom{m(K)}{j}\varepsilon^j(1-\varepsilon)^{m(K)-j}, & m(K)\text{: odd} \\ \sum_{j=\frac{m(K)}{2}+1}^{m(K)} \binom{m(K)}{j}\varepsilon^j(1-\varepsilon)^{m(K)-j} \\ \quad +\frac{1}{2}\binom{m(K)}{m(K)/2}[\varepsilon(1-\varepsilon)]^{m(K)/2}, & m(K)\text{: even} \end{cases} \tag{2}$$

The probability that a frame is decoded correctly by a receiver is

$$p'=(1-\varepsilon')^n. \tag{3}$$

For protocol 3, let the free distance and the distance spectra of the rate $1/m(K)$ convolutional code be $d_{free}^{m(K)}$ and $a_d^{m(K)}$, respectively. The probability of the decoding error event $P_{evt}(m(K))$ of the Viterbi decoding using this rate $1/m(K)$ code is bounded as [15]

$$P_{evt}(m(K)) \leqq \sum_{d=d_{free}^{m(K)}}^{\infty} a_d^{m(K)} P_d \tag{4}$$

where P_d is the probability that a wrong path at distance d is selected. For the binary symmetric channel (BSC) with bit error rate ε, P_d is given by [15]

$$P_d = \begin{cases} \sum_{j=\frac{d+1}{2}}^{d} \binom{d}{j}\varepsilon^j(1-\varepsilon)^{d-j}, & d\text{: odd} \\ \sum_{j=\frac{d}{2}+1}^{d} \binom{d}{j}\varepsilon^j(1-\varepsilon)^{d-j} \\ \quad +\frac{1}{2}\binom{d}{d/2}[\varepsilon(1-\varepsilon)]^{d/2}, & d\text{: even} \end{cases} \tag{5}$$

and the probability p'' that a receiver successfully receives the data frame within $m(K)$ copies after the Viterbi decoding satisfies

$$p'' \geqq (1-P_{evt}(m(K)))^l \tag{6}$$

where l is the number of trellis levels. With a code of rate b/V, $l=L/V$. In this protocol, $l=n$.

Equation (6) can be explained as follows. The probability $p''=Pr\{C\}$ that the decoded path is correct is given by

$$Pr\{C\}=Pr\{c^{(1)},c^{(2)},\cdots,c^{(i)},\cdots,c^{(l)}\} \tag{7}$$

where $c^{(i)}$ is the event that the i-th branch of the transmitted path is selected. $Pr\{C\}$ can be expressed as

$$\begin{aligned} Pr\{C\} &= Pr\{c^{(l)}/c^{(l-1)},\cdots,c^{(i)},\cdots,c^{(1)}\} \\ &\quad \times Pr\{c^{(l-1)}/c^{(l-2)},\cdots,c^{(i)},\cdots,c^{(1)}\} \\ &\quad \cdots \times Pr\{c^{(i)}/c^{(i-1)},\cdots,c^{(1)}\} \\ &\quad \cdots \times Pr\{c^{(2)}/c^{(1)}\} \times Pr\{c^{(1)}\}. \end{aligned} \tag{8}$$

Each term $Pr\{c^{(i)}/c^{(i-1)},\cdots,c^{(1)}\}$ in Eq. (8) is the probability that the i-th branch of the transmitted path is selected given that all branches of the transmitted path up to level $i-1$ have been selected. Clearly, we can write

$$Pr\{c^{(i)}/c^{(i-1)},\cdots,c^{(1)}\} \geqq Pr\{c^{(i)}\}. \quad (9)$$

Moreover, the probability $Pr\{C^{(i)}\}$ is bound [27] by

$$Pr\{c^{(i)}\} \geqq 1 - P_{evt}(m(K)) \quad (10)$$

Therefore, from Eqs. (8)–(10), $Pr\{C\}$ is bound by

$$Pr\{C\} \geqq (1 - P_{evt}(m(K)))^l \quad (11)$$

Next we define q_j as the probability that all K receivers successfully receive a data frame within j copies. Clearly, for protocol 1, we have

$$q_j(K) = (1-(1-p)^j)^K. \quad (12)$$

For protocol 2, we have

$$q_j(K)\begin{cases} = & (1-(1-p)^j)^K, \quad 1 \leqq j < m(K), \\ \geqq & (1-(1-p'))^K, \quad j = m(K). \end{cases} \quad (13)$$

For protocol 3, if $m(K)=1$, we have

$$q_j(K) = (1-(1-p)^j)^K \quad (14)$$

and if $m(K) \geqq 1$, we have

$$q_j(K) = \begin{cases} 0, & 1 \leqq j < m(K), \\ (1-(1-p''))^K, & j = m(K). \end{cases} \quad (15)$$

We define $T(K)$ as the average number of timeslots required to successfully transmit a frame to all K receivers. We assume that P_K is the probability that all K receivers receive the data frame in $m(K)$ copies, and P_l is the probability that exactly l receivers receive the data frame in $m(K)$ copies. For this case the time slots required to perform necessary retransmissions is given by $(N+m(K)-1+T(K-l))$, which is the sum of one retransmission cycle plus timeslots for $(K-l)$ receivers which failed to receive the data frame. Therefore, $T(K)$ can be written in a recursive form as follows:

$$\begin{aligned} T(K) &= P_K \cdot m(K) \\ &\quad + \sum_{l=0}^{K-1} P_l \cdot (N + m(K) - 1 + T(K-l)) \\ &= q_{m(K)}(K) \cdot m(K) + (1 - q_{m(K)}(1))^K \\ &\quad \times (N + m(K) - 1 + T(K)) \\ &\quad + \sum_{l=1}^{K-1} \binom{K}{l} q_{m(K)}(l)(1 - q_{m(K)}(1))^{K-l} \\ &\quad \times (N + m(K) - 1 + T(K-l)). \end{aligned} \quad (16)$$

The method we use is a dynamic programming where the stage is set to be the number of receivers that have not yet received the data frame. That is, we first select optimal $m(1)$, which is represented by $m^*(1)$, to get the minimum $T(1)$ denoted by $T^*(1)$, and then select $m^*(2)$ to get $T^*(2)$. This procedure is continued until we get $T^*(r)$, where r is the number of receivers. By substituting and rearranging Eq. (16), we get the recursive functional equation for $T^*(r)$

$$T^*(K) = \min_{m(K)} \frac{A(K)}{B(K)} \quad (17)$$

with $K = 1, 2, \cdots, r$.

For protocol 1, we have

$$\begin{aligned} A(K) &= m^*(K) + \sum_{l=1}^{K-1} \binom{K}{l} [1-(1-p)^{m^*(K)}]^l \\ &\quad \times [(1-p)^{m^*(K)}]^{(K-l)} \\ &\quad \times (N - 1 + T^*(K-l)) \end{aligned} \quad (18)$$

$$B(K) = 1 - [(1-p)^{m^*(K)}]^K. \quad (19)$$

For protocol 2, we have

$$\begin{aligned} A(K) &= m^*(K) + \sum_{l=1}^{K-1} \binom{K}{l} [1-(1-p')]^l \\ &\quad \times (1-p')^{(K-l)} \cdot (N - 1 + T^*(K-l)) \end{aligned} \quad (20)$$

$$B(K) = 1 - (1-p')^K. \quad (21)$$

For protocol 3, if $m(K)=1$, $A(K)$ and $B(K)$ are the same as in protocol 1, and if $m(K)>1$, we have

$$\begin{aligned} A(K) &= m^*(K) + \sum_{l=1}^{K-1} \binom{K}{l} [1-(1-p'')]^l \\ &\quad \times (1-p'')^{(K-l)} \cdot (N - 1 + T^*(K-l)) \end{aligned} \quad (22)$$

$$B(K) = 1 - (1-p'')^K. \quad (23)$$

The throughput efficiency for protocol 1 and 2 is given by

$$\eta = \frac{k}{n} \cdot \frac{1}{T^*(r)} \quad (24)$$

and for protocol 3

$$\eta = \frac{k}{n+h} \cdot \frac{1}{T^*(r)} \quad (25)$$

where h is the constraint length of a convolutional code.

4. Numerical Results and Comparisons

In this section, we examine the results of numerical computations and comparisons of the three protocols for the throughput efficiency. In the go-back-N ARQ protocols for multidestination communications listed in Refs. [9]–[14], Wang and Silvester's protocol [11] has the best performance. Because protocol 1 has almost the same

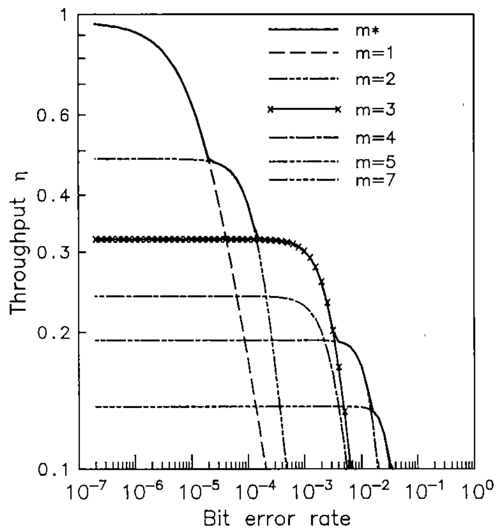


Fig. 4 Throughput versus BER for protocol 2 with optimal copies and with $m(K) = 1, 2, 3, 4, 5$, and 7 copies for $r = 5$ and $N = 40$.

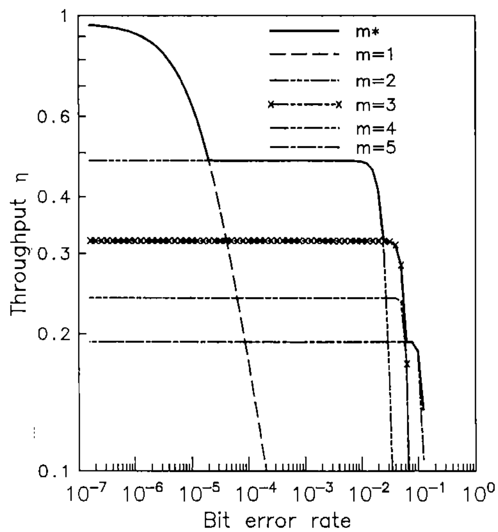


Fig. 5 Throughput versus BER for protocol 3 with optimal copies and with $m(K) = 1, 2, 3, 4$, and 5 copies for $r = 5$ and $N = 40$.

throughput rate as Wang and Silvester's protocol [11] we only need to compare our protocol 2 and 3 with protocol 1. We assume that the error detection code used here is the (1023,1003) BCH code.

Figure 4 shows the throughput efficiency versus the channel bit error rate for protocol 2 with optimal number of copies and with different fixed-copies for $N = 40$ and $r = 5$. For protocol 3, a lower bound on the throughput efficiency can be obtained using Eqs. (6) and (17). The rate 1/2 convolutional code used in protocol 3 has memory $mem = 6$ and generator polynomials (133, 171) [24]. The distance spectra of the repetition codes of rates $1/m(K)$, $m(K) \geq 2$, can be found in Table 2.

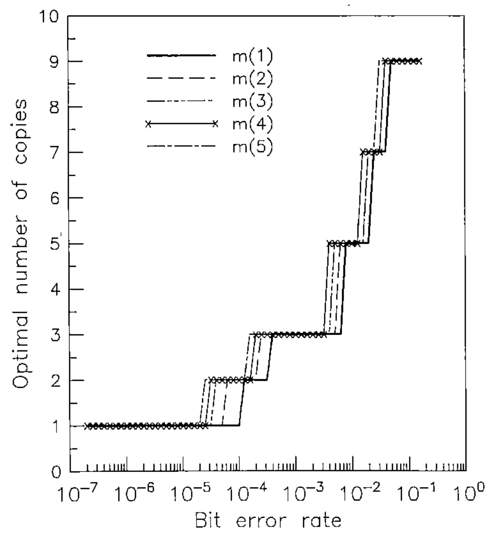


Fig. 6 Optimal number of copies versus BER for protocol 2 with $r = 5$ and $N = 40$.

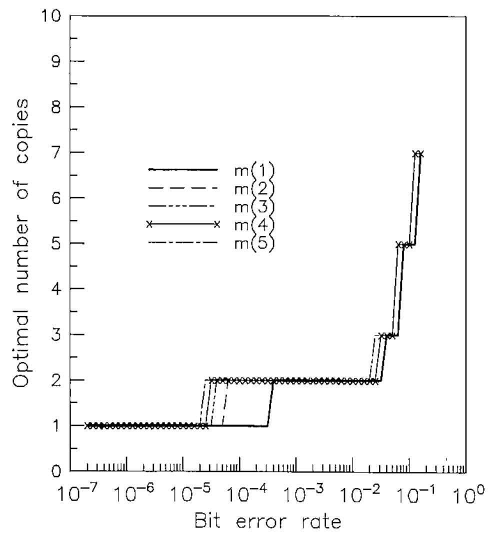


Fig. 7 Optimal number of copies versus BER for protocol 3 with $r = 5$ and $N = 40$.

Figure 5 shows the throughput efficiency versus the channel bit error rate for protocol 3 with optimal copies and with different fixed-copies for $N = 40$ and $r = 5$. In fact, these two figures show the throughput obtained by transmitting 1, 2, 3, 4, 5, 7 and the optimal number of copies of a data frame. The maximum throughput almost follows the envelope of all those fixed-copy curves, and is achieved by transmitting the optimal number of copies of a data frame. Figures 6 and 7 show the optimal number of copies versus the channel bit error rate corresponding to Figs. 4 and 5 when there are K (K = 1, 2, 3, 4, and 5) receivers that have not yet received the message correctly.

Figures 8–10 show the throughput efficiency versus the number of receivers for $N = 40$ and $\varepsilon = 10^{-4}$, 10^{-3} and 10^{-2}, respectively. It is clear that as r increases, the

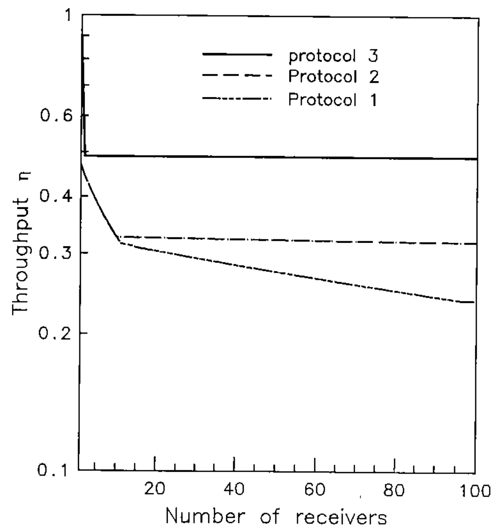


Fig. 8 Throughput versus number of receivers for BER $= 10^{-4}$ and $N = 40$.

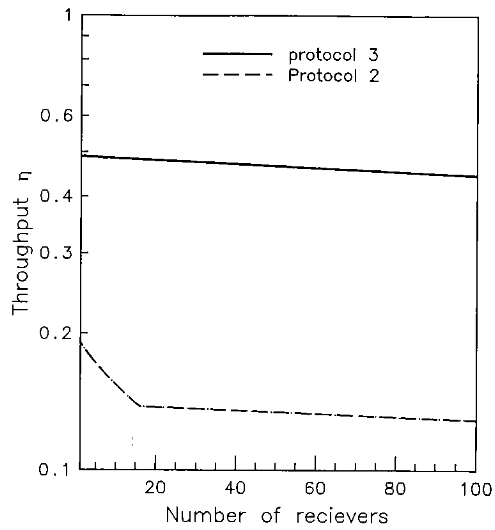


Fig. 10 Throughput versus number of receivers for BER $= 10^{-2}$ and $N = 40$.

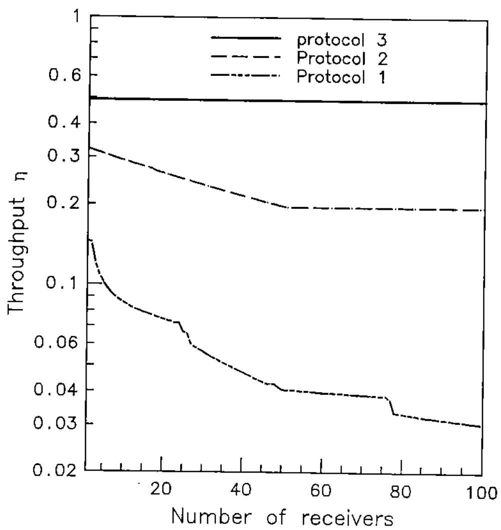


Fig. 9 Throughput versus number of receivers for BER $= 10^{-3}$ and $N = 40$.

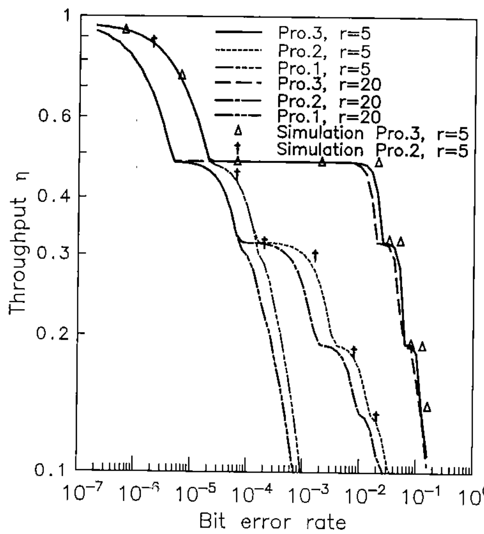


Fig. 11 Throughput versus BER for $r = 5$ and 20, and $N = 40$.

throughput decreases. The throughput of the protocol 1 is not shown in Fig. 10 because it is very low (lower that 10^{-2}). Figure 11 shows the throughput efficiency versus the channel bit error rate for the three protocols with $r = 5$ and 20, and $N = 40$.

Because the majority-voting decoding and the Viterbi decoding are used on $m(K)$ copies of a frame, our protocol 2 and 3 have much higher throughput than protocol 1 which has the same throughput rate as the conventional protocol [11].

By transmitting optimal number of copies of a data frame, we can obtain a much higher throughput than that achieved by only transmitting a fixed number of copies of a frame. Protocol 2 has a medium complexity and provides medium throughput. Protocol 3 has a high throughput under high error rate, but requires higher complexity for the transmitter and receivers, because the rate repetition convolutional code encoder and the Viterbi decoder are used. In our protocol 2 and 3, a finite buffer is needed for each receiver and at most $m(K)$ copies of a frame are stored in a buffer. From Figs. 6 and 7 we find that $m(K) \leq 9$. One frame contains 1024 bits of data, so that maximum data bits of the erroneous copies which need to be stored in a buffer is 9 kbits.

In order to verify the performances of the proposed protocols, computer simulations have been carried out. The simulations are performed on the DSPW (digital signal processing work-system). In order to obtain reliable statistics the number of samples should be sufficiently large. In our simulations, 100 Mbit samples are measured for each destined BER. The results of the computer simulations for protocol 2 and 3 with $N = 40$ and $r = 5$ are also given in Fig. 11. The results of simulations agree with that of numerical calculations under lower channel error conditions, and have a slightly higher throughput under higher channel error conditions. The difference between the simulations and calculations is mainly caused by the fact that Eqs. (13) and (6) used for the numerical calculations for protocol 2 and 3, respectively, gives the lower bound of the throughput, while the simulations provide more realistic values.

5. Conclusions

In this paper new go-back-N ARQ protocols which combined the error detection code with error correction code for point-to-multipoint communications are proposed. In these protocols, a dynamic programming technique is used to select the optimal number of copies of a frame to be transmitted at a time. The optimal number depends on the channel error probability, the round trip propagation delay, and the number of receivers that have not yet successfully received the data frame. Analytic expressions are derived for the error probabilities and throughput efficiency of the proposed protocols. Numerical computations showed that the proposed point-to-multipoint protocols provide satisfactory throughput efficiency under high error rate conditions especially in environments with a large number of receivers and large link round trips. Results of computer simulation are also given. It was confirmed that the proposed protocols have better performance than the conventional GBN-ARQ protocols for multidestination communications, especially for communications under high error rate conditions, such as in the information broadcasting system using USAT receivers.

Acknowledgment

Numerical simulation used in this paper was performed on Digital Signal Processing Work-system provided by Sony-Tektronix Inc..

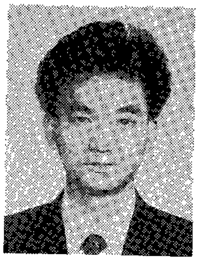
References

[1] Benelli, G., "Two concatenated coding schemes for ARQ protocols," *IEE Proc. I, Commun. Speech and Vis.*, vol.136, pp.213–219, Jun. 1989.
[2] Kallel, S., "Analysis of a type II hybrid ARQ scheme with code combining," *IEEE Trans. Commun.*, vol.COM-38, pp.1133–1137, Aug. 1990.
[3] Lin, M.C. and Guu, M.Y., "The performance analysis of a concatenated ARQ scheme using parity retransmissions," *IEEE Trans. Commun.*, vol.COM-39, pp.1869–1874, Dec. 1991.
[4] Kousa, M.A. and Rahman, M., "An adaptive error control system using hybrid ARQ schemes," *IEEE Trans. Commun.*, vol.COM-39, pp.1049–1057, Jul. 1991.
[5] Wicker, S.B., "Adaptive rate error control through the use of diversity combining and majority-logic decoding in a hybrid-ARQ protocol," *IEEE Trans. Commun.*, vol.COM-39, Mar. 1991.
[6] Lee, T.H., "Performance analysis of efficient concatenated coding scheme for ARQ protocols," *IEE Proc. I, Commun. Speech and Vis.*, vol.138, pp.148–152, Jun. 1991.
[7] Kallel, S. and Leung, C., "Efficient ARQ schemes with multiple copy decoding," *IEEE Trans. Commun.*, vol. COM-40, pp.642–650, Mar. 1992.
[8] Zhao, H., Sato, T. and Kimura, I., "A hybrid-ARQ protocol with adaptive rate error control," *IEICE Trans. Fundamentals*, vol.E76-A, no.12, pp.2095–2101, Dec. 1993.
[9] Gopal, I.S. and Jaffe, J.M., "Point-to-multipoint communication over broadcast links," *IEEE Trans. Commun.*, vol.COM-32, pp.1034–1044, Sep. 1984.
[10] Sabnani, K. and Schwartz, M., "Multidestination protocols for satellite broadcast channels," *IEEE Trans. Commun.*, vol.COM-33, pp.232–240, Mar. 1985.
[11] Wang, J.L. and Silvester, J.A., "Optimal adaptive ARQ protocols for point-to-multipoint communication," in *Proc. IEEE INFOCOM '88*, pp.704–711, New Orleans, USA, Mar. 1988.
[12] Cho, Y.J. and Un, C.K., "Continuous multidestination ARQ schemes for high error-rate channels," *Electron. Lett.*, vol.24, pp.694–695, May 1988.
[13] Lee, T.H., "Some continuous multidestination ARQ schemes for high error-rate conditions," *Electron. Lett.*, vol.26, pp.1686–1687, Sep. 1990.
[14] Lee, T.H., "Improving performance of multidestination ARQ schemes under high error rate conditions," *Electron. Lett.*, vol.27, pp.293–294, Jan. 1991.
[15] Deng, R.H., Zhang, X.Y. and Tham, Y.K., "New point-to-multipoint communication protocols," *IEE Proc. I, Commun. Speech and Vis.*, vol.136, pp.312–316, Aug. 1989.
[16] Chandranm, S.R. and Lin, S., "Selective-repeat-ARQ schemes for broadcast links," *IEEE Trans. Commun.*, vol.COM-40, pp.12–19, Jan. 1992.
[17] Nakamura, M. and Kodama, T., "An efficient hybrid ARQ scheme for broadcast data transmission systems," *IEICE Trans.*, vol.J73-A, no.2, pp.277–283, Feb. 1990.
[18] Kallel, S. and Haccoun, D., "Generalized type II hybrid ARQ scheme using punctured convolutional coding," *IEEE Trans. Commun.*, vol.38, pp.1938–1946, Nov. 1990.
[19] Kallel, S., "Analysis of memory and incremental redundancy ARQ schemes over a nonstationary channel," *IEEE Trans. Commun.*, vol.40, pp.1474–1480, Sep. 1992.
[20] Deng, R.H., "Hybrid ARQ Schemes forlg point-to-multipoint communication over nonstationary broadcast channel," *IEEE Trans. Commun.*, vol.41, pp.1379–1387, Sep. 1993.
[21] Sakakibara, K. and Kasahara, M., "Performance of a multicast error control protocol based on a product code structure – part I: on random error channels," *IEICE Trans. Fundamentals*, vol.E75-A, no.12, pp.1674–1683, Dec. 1992.
[22] Sakakibara, K. and Kasahara, M., "Performance of a multicast error control protocol based on a product code structure – part II: on burst error channels," *IEICE Trans. Fundamentals*, vol.E75-A, no.12, pp.1684–1695, Dec. 1992.
[23] Lin, S. and Costello, Jr., D.J., *Error Control Coding: Fundamentals and Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1983.
[24] Conan, J., "The weight spectra of some short low-rate convolutional codes," *IEEE Trans. Commun.*, vol.COM-32, pp.1050–1053, Sep. 1984.
[25] Haccoun, D. and Montreuil, P., "Algorithms for the weight spectrum determination of convolutional codes," in *Books*

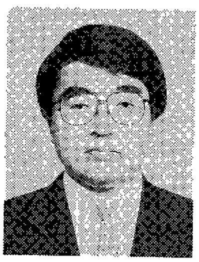
of Abstr., 1988 Int. Symp. Inform. Theory, Kobe, Japan, pp.49–50, Jun. 1988.

[26] Larsen, K.J., "Short convolutional codes with maximal free distance for rate 1/2, 1/3, and 1/4," *IEEE Trans. Inform. Theory*, vol.IT-19, May 1973.

[27] Viterbi, A.J., "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun.*, vol.COM-19, pp.751–772, Oct. 1971.



Hui Zhao was born on April 14, 1961. He received the B.S. degree in electrical engineering from Harbin Institute of Technology, Harbin, P.R. China in 1982 and the M.S. degree in electrical engineering from Shanghai University of Science and Technology, Shanghai, P.R. China in 1988. From 1988 to 1990, he was employed in the Shanghai Research Institute of Radio Equipment, the Ministry of Aeronautic and Astronautic Industry (MAAI), P.R. China, involved in the digital communication system development. He is now studying towards the Ph.D. degree at the Department of Electrical Engineering II, Faculty of Engineering, Kyoto University, Kyoto, Japan. His current research interest includes error control techniques, coding theory and satellite communications.



Toru Sato was born on March 27, 1954. He received the B.E., M.E., and Ph.D. degrees in electrical engineering from Kyoto University, Kyoto, Japan in 1976, 1978, and 1982, respectively. He was a graduate student research associate of Arecibo Observatory, National Astronomy and Ionosphere Center from 1979 to 1980. He joined Radio Atmospheric Science Center of Kyoto University in 1983 as a research associate. He is now an associate professor at Department of Electrical Engineering II, Kyoto University. His major research interests have been system design and signal processing for atmospheric radars, radar remote sensing of the atmosphere, observations of precipitation using radar and satellite signals, radar observation of space debris, signal processing for subsurface radars, and digital satellite communication. He was awarded Tanakadate Prize in 1986. He is a member of the Society of Geomagnetism and Earth, Planetary and Space Sciences, the Japan Society for Aeronautical and Space Sciences, the Institute of Electrical and Electronics Engineers, and American Meteorological Society.

Iwane Kimura was born on December 25, 1932. He received the B.E., M.E., and Ph.D. degrees in electrical engineering from Kyoto University, Kyoto, Japan in 1955, 1957, and 1961 respectively. Since 1960, he has been a staff of Kyoto University; Department of Electronics and Department of Electrical Engineering II where he is now a Professor radio engineering since 1971. He has also been a visiting Professor of Institute of Space and Astronautic Science from 1981 to 1991. From 1964 to 1965, he was a research associate at Radioscience Laboratory, Stanford University on leave from Kyoto University. His research interests have been in radio science; particularly remote-sensing of upper atmospheres by radar techniques, and space plasma physics, such as theoretical study on propagation and generation of radio waves in magnetospheric and ionospheric plasmas, plasma measurements by rocket-Doppler technique, and active wave experiments in space plasmas using scientific satellites. He was awarded Inada Memorial Prize in 1958, and Tanakadate Prize in 1961. He is a member of the Institute of Electrical Engineers of Japan, the Society of Geomagnetism and Earth, Planetary and Space Sciences, and the American Geophysical Union.