

Delay Analysis of the Selective-repeat ARQ Protocol with the Per-Flow Resequencing Scheme

TOSHIHIRO SHIKAMA,[†] SHOICHIRO SENO,[†] TAKASHI WATANABE^{††}
and TADANORI MIZUNO^{††}

An SR (Selective-Repeat) ARQ (Automatic Repeat reQuest) protocol is used to recover from packet errors effectively over a low-quality communication channel. This SR ARQ has a problem of large delay due to resequencing of received packets. To mitigate this problem, the PFRS (Per-Flow ReSequencing) scheme was proposed, where the resequencing is performed independently for each upper-layer flow, while detection of lost packets and associated retransmissions are performed on the basis of the whole flows multiplexed over SR ARQ. This paper models the SR ARQ protocol, where the maximum number of retransmissions is limited, by a collection of simple stop-and-wait protocols, and shows numerical calculation results for the delay distribution of retransmission and resequencing. The validity of the analysis is confirmed by comparing numerical calculations with simulation results. The results prove the effectiveness of the PFRS scheme for the case where the number of flows over SR ARQ is large.

1. Introduction

ARQ (Automatic Repeat reQuest) protocols are important for recovering packets lost as a result of transmission errors over a low-quality communication channel. Go-Back-N ARQ has hitherto been widely used because of its simple retransmission mechanism. However, as the bandwidth of wireless communication increases, it has become increasingly common to employ Selective-Repeat (SR) ARQ, which is more complicated. This SR ARQ is effective, as it retransmits the minimum number of packets that actually encounter transmission errors. The receiving side of SR ARQ has to perform the resequencing function, which retains correctly received packets that arrive after some packets are received in error. This resequencing is needed to preserve the sequence integrity of packet communication. However, it incurs a large packet delay, since all packets correctly received after the lost packet have to be retained until the lost one is retransmitted and correctly received. Another defect of the resequencing is that when the lost packet is retransmitted and received correctly, all packets waiting for the lost one are released and transferred to the upper layer at the same time. A large burst of packets, which might have undesirable effects on other packet flows, is generated by the resequencing.

quencing.

To date, studies of the performance of SR ARQ have focused mainly on queueing analysis of its sending side^{1),2)}. As for the resequencing performed by the receiving side, numerical analysis of the delay and buffer occupancy due to the resequencing has been carried out for the case where a transmission channel is fully loaded³⁾. The RLC (Radio Link Control), which is a layer 2 protocol defined by 3GPP, employs SR ARQ⁵⁾, and there has been a study of the effect of out-of-order packets caused by the limit of the hold time in resequencing by the RLC⁶⁾. In order to mitigate the delay and bursty packet output due to the resequencing, the Per-Flow ReSequencing (PFRS) scheme was proposed⁷⁾. This scheme mitigates the problems by performing the resequencing for each upper-layer flow independently. However, its effectiveness has not been well studied with regard to the relation between the resequencing delay and the number of flows over SR ARQ. The present paper analyzes the resequencing delay of the PFRS scheme, and proves that the delay is significantly reduced by the scheme.

There have been various studies related to the analysis presented in this paper, and the differences between our analysis and these studies need to be clarified. One study analyzes the resequencing delay of SR ARQ for the case where multiple parallel channels exist between a sender and a receiver⁴⁾. In that study, the num-

[†] Mitsubishi Electric Corporation
^{††} Shizuoka University

ber of retransmissions is unlimited and parallel channels are fully loaded, while the boundaries of slots are aligned on all channels. These conditions are different from those on which we are focusing in this paper. Another study presents a heuristic analysis of the resequencing delay of the UMTS RLC protocol, where an upper layer SDU is segmented into an integral number of link-layer PDUs⁸⁾. The receiving side reorders and reassembles the PDUs to reconstruct an SDU. Although the aims of the study are partly similar to those of our analysis, it differs in the following respects:

- It presents an approximate analysis based on heuristics to avoid considering a huge number of states. We present an exact analysis based on all possible states of SR ARQ on the channel. Although the results of the heuristic approach reveal good agreement with simulation results, the limitations of the heuristics are not clear.
- It assumes that a channel is fully loaded (the heavy traffic assumption), while our analysis in this paper introduces a parameter specifying utilization of the channel. Our analysis covers the fully loaded case by setting the parameter to a specific value.

The rest of this paper is organized as follows: Section 2 describes the PFRS scheme, the conditions of the analysis, the analytical model, and the state probability. Section 3 derives the delay distribution due to resequencing and retransmission. Section 4 shows the results of numerical calculations, including a comparison with simulation results. Finally, our conclusions are presented in Section 5.

2. The PFRS Scheme and Its Analytical Model

2.1 Conventional SR ARQ and the PFRS Scheme

An example of a retransmission sequence in the conventional SR ARQ is shown in **Fig. 1**. SR ARQ preserves the order of all packets on a transmission channel. This SR ARQ has problems of unnecessary retention of packets and associated delays. In this figure, two flows, a and b, are multiplexed over SR ARQ. Packets a1 and a2 are lost due to transmission errors and retransmitted. In the case of conventional SR ARQ, packets b1 and b2 of flow b are unnecessarily retained until the lost packet a1 is retransmitted and received. This situation is a kind of HOL (head of line) blocking for flow b.

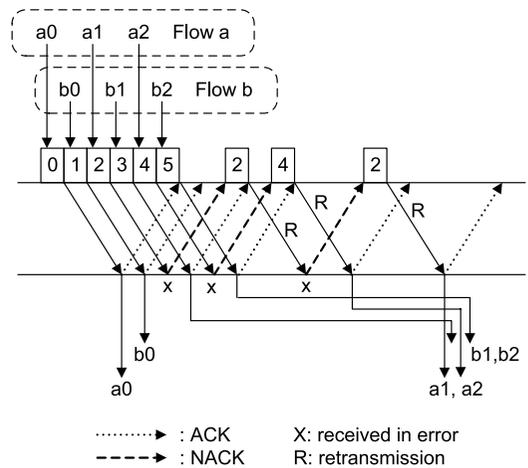


Fig. 1 An example of a sequence in the full resequencing scheme.

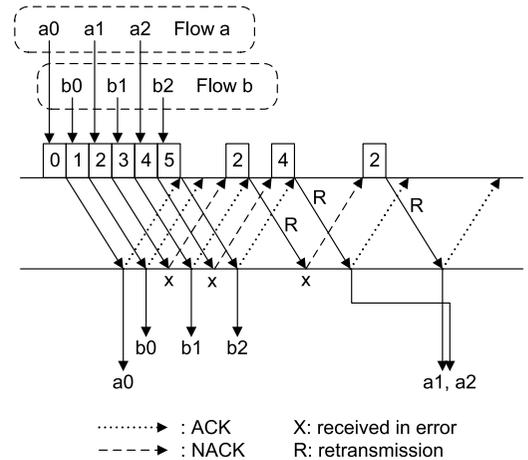


Fig. 2 An example of a sequence in the PFRS scheme.

The basic idea behind the PFRS scheme is that there is no need to maintain packet sequence integrity among different upper-layer flows while the packet order has to be preserved for the same upper-layer flow. The PFRS scheme performs resequencing for each upper-layer flow independently, while it detects and retransmits lost packets based on the whole packet flows.

In the case of the PFRS scheme, as shown in **Fig. 2**, resequencing is performed for each flow independently, while acknowledgment and retransmission of packets are carried out by SR ARQ in the conventional way, and packets b1 and b2 are delivered to the upper layer without being retained. Thus, the PFRS scheme resolves the invalid suspension of packets due

to the HOL blocking.

SR ARQ is generally performed in layer 2, where packets on upper-layer flows are transparently transferred. In order to realize the PFRS scheme, we assume that the layer 2 header includes extra information identifying upper-layer flows and the order of packets in each flow. As an example of such extra information, a pointer is proposed⁹). The pointer utilizes the sequence number of SR ARQ to realize the identification of flows and the order of packets effectively.

The PFRS scheme is thought to have the advantage of reducing the resequencing delay. This advantage will be significant if the number of flows over SR ARQ becomes large. A motivation of this paper is to evaluate the relation between the resequencing delay and the number of flows for the case where either conventional SR ARQ or the PFRS scheme is employed. It should be stressed that no existing analysis has focused on multiple flows over SR ARQ with independent resequencing on each flow. This situation, which the present paper attempts to analyze, is completely new.

2.2 Analytical Model and Associated Assumptions

To evaluate the resequencing delay of the conventional scheme and the PFRS scheme, analysis is performed under the following assumptions:

1. Packet errors occur at random on a transmission channel with the error rate ϵ .
2. If a packet is received without error, a positive acknowledgment (ACK) is returned to the sender; otherwise, a negative acknowledgment (NACK) is returned. An ACK or NACK is sent immediately when a packet is received correctly or in error.
3. No error occurs for either positive or negative acknowledgments.
4. Retransmission has high priority compared with transmission of a new packet, and is performed immediately after a NACK is returned to the sender.
5. The maximum number of retransmissions is limited to N_r . If a packet retransmitted N_r times is still received in error, packets waiting for this packet at the receiver are released and transferred to the upper layer.
6. The length of a packet is fixed and time is divided into slots. The duration of a

slot is the time needed for sending one packet.

7. In each slot, if a channel is available (i.e., if there is no retransmission), a new packet to be transmitted exists with probability α .
8. There are m flows over the transmission channel. A new packet belongs to flow j at random with probability β_j ($j = 1, \dots, m$), where $\sum_{j=1}^m \beta_j = 1$.
9. A collection of consecutive slots, from the beginning of a packet transmission time till a reception of the associated acknowledgment for this packet, is called a frame.
10. The number of slots in a frame is assumed to be M . Each slot in a frame is numbered from 0 to $M - 1$.
11. The number M consists of the number of slots corresponding to the round trip delay and one slot for a packet transmission.
12. A collection of slots in the same position of consecutive frames is called a sub-channel. There are sub-channels from 0 to $M - 1$ in a frame.

An example of slots, frames, and sub-channels is shown in **Fig. 3**. In accordance with the above assumptions, once a new packet is sent on one of the sub-channels, retransmission of the packet is always performed on the same sub-channel until the packet is successfully received or aborted on account of the maximum number of retransmissions having been reached. Accordingly, SR ARQ can be modeled by a collection of independent M sub-channels, on each of which a simple stop-and-wait protocol is performed. **Figure 4** shows an example of modeling the SR ARQ protocol by multiple stop-and-wait protocols, where the sequence of packets corresponds to the cases of Figs.1 and 2. We

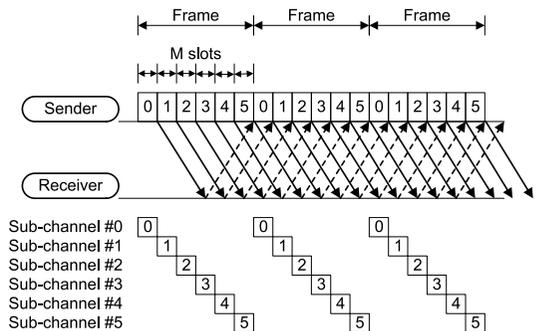


Fig. 3 An example of slots, frames, and sub-channels.

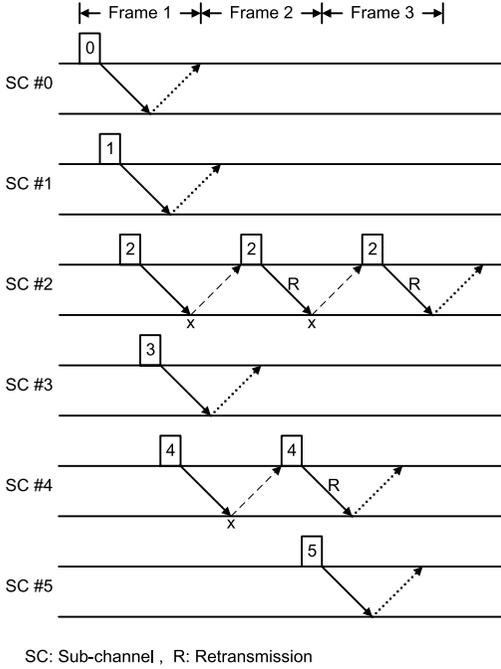


Fig. 4 Modeling of the SR ARQ protocol by a collection of multiple stop-and-wait protocols.

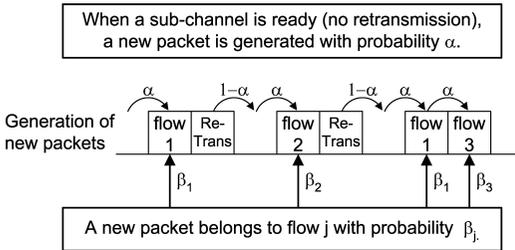


Fig. 5 Packet generation of multiple flows with parameters α and β .

introduced the probability α to change the total load of the transmission channel. Since it is expected that the resequencing delay will decrease when the load of the channel becomes small, we can investigate this effect by changing α . We also introduced the probability β_j ($j = 1, \dots, m$) to study the effect of multiple flows over the transmission channel. Packet generation of multiple flows with these parameters is depicted in **Fig. 5**. When a slot is available (no retransmission), transmission of a new packet is performed with probability α . The new packet belongs to one of the flows, for example flow j , at random with probability β_j .

We can analyze the collection of these sub-channels to obtain the distribution of the resequencing delay. In the following analysis, we

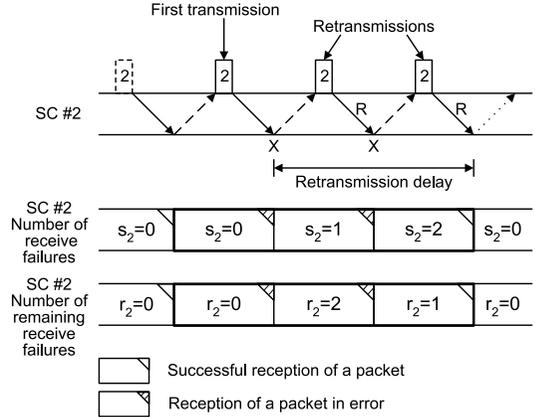


Fig. 6 An example of state transition on sub-channel #2.

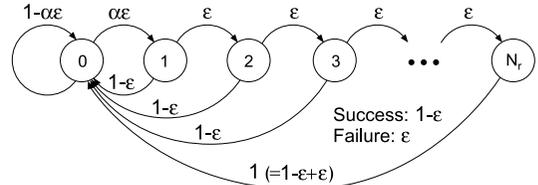


Fig. 7 A state transition diagram based on the number of consecutive receive failures of a packet.

will focus on a single sub-channel. However, this never means any loss of generality in the analysis, as the phase of a frame is arbitrary.

2.3 State Probability of a Sub-channel

We consider the case where the state of a sub-channel is defined by the number of consecutive receive failures of a packet. For example, **Fig. 6** shows changes in the state of sub-channel #2. When a packet is received in error, the state value is increased by 1; otherwise, the state is reset to 0. If a packet is received in error after the maximum number of retransmissions, the state also returns to 0. If new packets are consecutively sent without any transmission error, the value of the state remains at 0 in consecutive slots. If we define the state as the number of receive failures of a packet, the state transition is represented as in **Fig. 7**. The probabilities of state $p_s(k)$ and packet loss rate p_L are given respectively by

$$p_s(0) = \frac{1 - \epsilon}{1 - \epsilon + \alpha\epsilon - \alpha\epsilon^{N_r+1}}, \tag{1}$$

$$p_s(k) = \frac{\alpha(1 - \epsilon)\epsilon^k}{1 - \epsilon + \alpha\epsilon - \alpha\epsilon^{N_r+1}}, \tag{2}$$

$1 \leq k \leq N_r.$

$$p_L = \epsilon^{N_r+1}. \tag{3}$$

In the analysis of the resequencing delay described in the subsequent section, it is necessary to know the number of remaining receive failures till the end of current packet transmission. We call this the number of remaining receive failures. Figure 6 also shows the change of state based on this number. In this case, when a packet is received in error for the first time, the state transits to the total number of receive failures until the end of this packet transmission. After that, the value of the state is decreased by 1 each time a packet is received in error. It should be noted that the end of packet transmission means that the packet is either received successfully or aborted because the maximum number of retransmissions has been reached. If we define the state as the number of remaining receive failures, its state probability $p_r(r)$ can be calculated on the basis of the original state probability $p_s(k)$.

In the case where $r = 0$

$$\begin{aligned}
 p_r(0) &= \{1 - \alpha + \alpha(1 - \varepsilon)\}p_s(0) \\
 &+ \sum_{k=1}^{N_r-1} (1 - \varepsilon)p_s(k) + p_s(N_r) \\
 &= \frac{1 - \varepsilon}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}} \quad (4)
 \end{aligned}$$

In the case where $1 \leq r \leq N_r$

$$\begin{aligned}
 p_r(r) &= \alpha\varepsilon^r(1 - \varepsilon)p_s(0) \\
 &+ \sum_{k=1}^{N_r-r-1} (1 - \varepsilon)\varepsilon^r p_s(k) \\
 &+ \varepsilon^r p_s(N_r - r) \\
 &= \frac{\alpha(1 - \varepsilon)\varepsilon^k}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}} \quad (5)
 \end{aligned}$$

As indicated above, the state probability, based on the number of remaining receive failures, takes the same form as Eqs. (1) and (2). We can intuitively explain this on the basis of a cycle from the end of the previous packet transmission to the end of the current packet transmission. In each cycle, there is one-to-one mapping between the state value of the first definition (the number of receive failures) and the same state value of the second definition (the number of remaining receive failures), as depicted in Fig. 6. Accordingly, the probability of each state is the same for both of these state definitions.

As mentioned before, we will study the case where the load of the transmission channel is changed by the parameter α . The transmission

channel becomes vacant with probability $(1 - \alpha)$ on condition that the state of a sub-channel is 0. Then the utilization factor ρ of the transmission channel is given by

$$\begin{aligned}
 \rho &= 1 - (1 - \alpha)p_s(0) \\
 &= \frac{\alpha(1 - \varepsilon^{N_r+1})}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}}. \quad (6)
 \end{aligned}$$

3. Delay Distribution due to Resequencing and Retransmission

In this section we will calculate two types of delay. One is pure resequencing delay and the other is the delay including both retransmission and resequencing.

3.1 Delay Distribution of Pure Resequencing

3.1.1 In the Case Where Resequencing Occurs ($i \neq 0$)

The delay due to pure resequencing is from the time a packet is correctly received to the time it is delivered to the upper layer. We assume that a packet of flow j is correctly received on sub-channel #0 after it has been retransmitted u times ($u = 0, 1, 2, \dots, N_r$).

Figure 8 shows the case where a packet on sub-channel #0 is delayed because of the need to wait for reception of a packet that belongs to the same flow j on sub-channel #2. In Fig. 8, a packet on sub-channel #0 is received correctly after u ($u = 2$) receive failures. The first receive failure occurs at time t_0 . When we observe the states of other sub-channels at this moment, a resequencing delay occurs if there is at least one sub-channel whose state (the number of remaining receive failures) is greater than or equal to $u + 1$. In the figure, the states of sub-channels #2 and #4 are 4 and 3, respectively, at time t_0 .

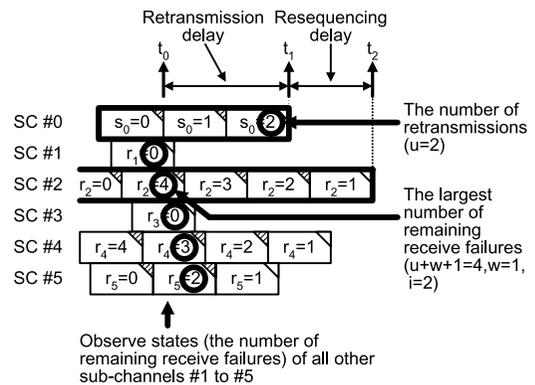


Fig. 8 An example of a resequencing delay.

If the state (the number of remaining receive failures) of sub-channel # i is the largest among the sub-channels from #1 to # $(M - 1)$ and its value is $u + w + 1$ ($w = 0, 1, 2, \dots, N_r - 1$), the resequencing delay of the packet on sub-channel #0 becomes $wM + i$ slots. In Fig. 8, the state of sub-channel #2 is the largest, and therefore $i = 2$, $u + w + 1 = 4$, and $w = 1$. Since M is assumed to be 6 slots in this figure, the resequencing delay becomes $wM + i = 8$ slots. Let us define the probability of this event as $P_{reseq}(w, i|\beta_j)$, which can be calculated from the following equation:

$$P_{reseq}(w, i|\beta_j) = \sum_{u=0}^{N_r-w-1} p_t(u)R(w + u + 1, i|\beta_j), \quad (7)$$

where $p_t(u)$ corresponds to the case where a packet on sub-channel #0 is received correctly after u receive failures.

$$p_t(u) = \frac{(1 - \varepsilon)\varepsilon^u}{1 - \varepsilon^{N_r+1}} \quad (8)$$

$R(w + u + 1, i|\beta_j)$, which will be defined and calculated below, corresponds to the case where a packet on sub-channel # i belongs to flow j and the state of sub-channel # i is the largest among those of sub-channels from #1 to # $(M - 1)$ with value $w + u + 1$.

More precisely, $R(n, i|\beta_j)$ is defined as the probability for the case where all the following conditions are satisfied:

- A packet on sub-channel # i belongs to flow j , which is the same flow as the packet on sub-channel #0.
- The number of remaining receive failures of a packet on sub-channel # i is n .
- Packets on sub-channels from #1 to # $(i - 1)$ do not belong to flow j , or if there is a packet of flow j , the number of its remaining receive failures is less than or equal to that of the packet on sub-channel # i .
- Packets on sub-channels from # $(i + 1)$ to # $(M - 1)$ do not belong to flow j , or if there is a packet of flow j , the number of its remaining receive failures is less than that of the packet on sub-channel # i .

We define the probability $U(n|\beta_j)$ for the event that a packet on a sub-channel does not belong to flow j or, if it belongs to flow j , the number of its remaining receive failures is less than or equal to n .

$$U(n|\beta_j) = 1 - \beta_j + \beta_j \sum_{r=0}^n p_r(r) = 1 - \beta_j + \beta_j \frac{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{n+1}}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}} \quad (9)$$

As each sub-channel is independent, the probability that the numbers of remaining receive failures of sub-channels from #1 to # $(M - 1)$ are r_1, \dots, r_{M-1} , respectively, is given by the products of each state's probability:

$$P_{rob}(r_1, \dots, r_i, \dots, r_{M-1}) = \prod_{k=1}^{M-1} p_r(r_k) \quad (10)$$

From this equation, $R(n, i|\beta_j)$ can be calculated so that the above-mentioned conditions are satisfied.

$$R(n, i|\beta_j) = \left(\prod_{k=1}^{i-1} U(n|\beta_j) \right) \cdot \beta_j p_r(n) \cdot \prod_{k=i+1}^{M-1} U(n - 1|\beta_j) = \beta_j p_r(n) U(n|\beta_j)^{i-1} \cdot U(n - 1|\beta_j)^{M-1-i} \quad (11)$$

By using Eqs. (7), (9), and (11), we can calculate the distribution of the resequencing delay for the case where $i \neq 0$.

3.1.2 In the Case of No Resequencing ($i = 0$)

A resequencing delay occurs if a packet on sub-channel #0 that has been received correctly waits for a packet sent on one of the other sub-channels. Since $i = 0$ corresponds to the case where a packet on sub-channel #0 waits for another packet on the same channel, there is no possibility of such an event, and therefore the probability is 0 in the case where $w \neq 0$. However, there is a possibility that the delay is 0 in the case where $w = 0$, and therefore the probability $P_{reseq}(0, 0|\beta_j)$ exists. **Figure 9** shows an example in which packets on all other sub-channels belong to flow j and a resequencing delay does not occur.

In this figure, a packet on sub-channel #0 is received correctly after receive failures have occurred u ($u = 3$) times. We observe the states (the number of remaining receive failures) of other sub-channels at time t_0 , which is the time at which the first transmission of the packet is received. No resequencing delay ever occurs if a

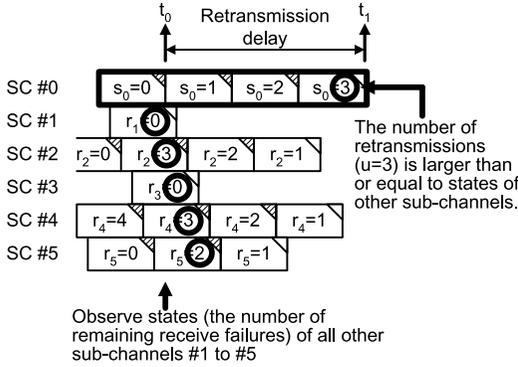


Fig. 9 An example of no resequencing delay.

packet on each sub-channel does not belong to flow j or, if it belongs to flow j , the number of remaining receive failures is less than or equal to u .

$$P_{reseq}(w, 0|\beta_j) = \begin{cases} \sum_{u=0}^{N_r} p_t(u)U(u|\beta_j)^{M-1} & w = 0 \\ 0 & w \neq 0 \end{cases} \quad (12)$$

3.1.3 Resequencing Delay Distribution for the Case of Multiple Flows

We can calculate the resequencing delay distribution for all of the packets sent over m flows as follows:

$$P_{reseq}(w, i) = \sum_{j=1}^m \beta_j P_{reseq}(w, i|\beta_j). \quad (13)$$

If the packet generation probability is the same for all m flows, then the distribution is calculated as follows, where $\beta = 1/m$:

$$P_{reseq}(w, i) = P_{reseq}(w, i|\beta) \quad (14)$$

3.2 Delay Distribution of Both Retransmission and Resequencing

Let us define probability $P_{delay}(u, i|\beta_j)$ as the sum of the retransmission delay and the resequencing delay is $(uM + i)$ slots in the case where a packet of sub-channel #0 is successfully received after receive failures have occurred u times.

3.2.1 In the Case Where Resequencing Occurs ($i \neq 0$)

The sum of the retransmission and resequencing delays becomes $(uM + i)$ slots for the following case. A packet of sub-channel #0 is received correctly after u or fewer receive failures. We observe the states (the number of remaining receive failures) of other sub-channels at time t_0 ,

which is the time at which the first transmission of the packet is received. The state of sub-channel # i is the largest among those of packets belonging to flow j on other sub-channels, except for sub-channel #0, and its value is $u + 1$.

$$\begin{aligned} P_{delay}(u, i|\beta_j) &= \left(\sum_{k=0}^u p_t(k) \right) R(u + 1, i|\beta_j) \\ &= \beta_j q_t(u) p_r(u + 1) U(u + 1|\beta_j)^{i-1} \\ &\quad \cdot U(u|\beta_j)^{M-1-i} \\ &= \beta_j q_t(u) p_r(u + 1) \frac{U(u|\beta_j)^{M-1}}{U(u + 1|\beta_j)} \\ &\quad \cdot \left(\frac{U(u + 1|\beta_j)}{U(u|\beta_j)} \right)^i, \end{aligned} \quad (15)$$

where $q_t(u)$ is the probability that a packet is received correctly after u or fewer receive failures:

$$q_t(u) \triangleq \sum_{k=0}^u p_t(k) = \frac{1 - \varepsilon^{u+1}}{1 - \varepsilon^{N_r+1}}. \quad (16)$$

3.2.2 In the Case of No Resequencing ($i = 0$)

A packet on sub-channel #0 is received correctly after receive failures have occurred u times. We observe the states (the number of remaining receive failures) of other sub-channels at time t_0 , which is the time at which the first transmission of the packet is received. The resequencing delay is 0 if a packet on each sub-channel does not belong to flow j or, if it belongs to flow j , the number of remaining receive failures is less than or equal to u .

$$P_{delay}(u, 0|\beta_j) = p_t(u)U(u|\beta_j)^{M-1} \quad (17)$$

3.2.3 In the Case of Multiple Flows

In the same manner as in the case of the resequencing delay, the delay distribution of both retransmission and resequencing can be calculated as follows:

$$P_{delay}(u, i) = \sum_{j=1}^m \beta_j P_{delay}(u, i|\beta_j) \quad (18)$$

If the packet generation probability of each flow is the same, the delay distribution of both retransmission and resequencing can be calculated as follows:

$$P_{delay}(u, i) = P_{delay}(u, i|\beta). \quad (19)$$

4. Results of Numerical Calculations Based on the Analysis

Numerical calculations based on the analysis were performed for the case in which SR ARQ

is applied to a satellite channel whose transmission rate is 1.5 Mb/s and whose packet length is 1,500 bytes. The propagation delay of the satellite channel is assumed to be 0.25 sec. Under these conditions, the calculated value of M becomes 64. Simulations were also performed under the same conditions, to confirm the validity of the analysis.

4.1 In the Case Where α is Changed

Table 1 shows the utilization factor P , the average retransmission delay, the average resequencing delay, and the average delay due to both retransmission and resequencing for various values of α . In this table, we select $\beta = 1$, which corresponds to the case in which conventional full resequencing is performed or the number of upper-layer flows is 1. The parameter α is introduced to change the utilization factor of the satellite channel. As α decreases, the utilization factor also becomes small and takes almost the same value as α . However, if the packet error rate becomes large, the utilization factor becomes larger than α because of additional traffic due to retransmissions. **Figure 10** shows the average retransmission delay and the average resequencing delay for various values of α , where the delay is normalized by the number of slots M in a frame. In this figure, the left column indicates the average retransmission delay, while the other columns show the average resequencing delay. From **Table 1** and **Fig. 10**, it is clear that the average resequencing delay is dominant in relation to the average retransmission delay. The reason for this is that only a packet which is lost and retransmitted undergoes a retransmission delay, while a large number of subsequent packets received correctly undergo a resequencing delay. As shown in the table, if the packet error rate is fixed, the resequencing delay decreases as the utilization factor of the channel becomes small, since the number of packets involved in the resequencing decreases as the utilization factor becomes small. **Figure 11** shows the delay distribution due to the resequencing alone, where ε is 0.1 and α is 0.9, while **Fig. 12** shows the delay distribution due to both retransmission and resequencing under the same conditions. In these figures, the simulation results are also plotted. Since the results of the simulations and numerical calculations overlap considerably, this proves the validity of the analysis described above. The probability of the delay decreases in steps as the delay increases,

Table 1 Utilization factor, average retransmission delay (slot), average resequencing delay (slot), and average delay due to both retransmission and resequencing (slot).

$N_r = 3, \beta = 1.0$			retrans.	reseq.	total
α	ε	ρ	delay	delay	delay
1.0	10^{-4}	1.0000	0.0064	0.2012	0.2076
	10^{-3}	1.0000	0.0641	1.9770	2.0410
	10^{-2}	1.0000	0.6465	16.7627	17.4092
	10^{-1}	1.0000	7.0855	71.8210	78.9065
0.9	10^{-4}	0.9000	0.0064	0.1811	0.1875
	10^{-3}	0.9001	0.0641	1.7831	1.8472
	10^{-2}	0.9009	0.6465	15.3821	16.0286
	10^{-1}	0.9091	7.0855	69.4644	76.5499
0.8	10^{-4}	0.8000	0.0064	0.1610	0.1674
	10^{-3}	0.8002	0.0641	1.5884	1.6524
	10^{-2}	0.8016	0.6465	13.9441	14.5906
	10^{-1}	0.8163	7.0855	66.7985	73.8840
0.7	10^{-4}	0.7000	0.0064	0.1409	0.1473
	10^{-3}	0.7002	0.0641	1.3928	1.4569
	10^{-2}	0.7021	0.6465	12.4459	13.0923
	10^{-1}	0.7216	7.0855	63.7298	70.8153
0.6	10^{-4}	0.6000	0.0064	0.1208	0.1272
	10^{-3}	0.6002	0.0641	1.1964	1.2605
	10^{-2}	0.6024	0.6465	10.8845	11.5309
	10^{-1}	0.6250	7.0855	60.1194	67.2049

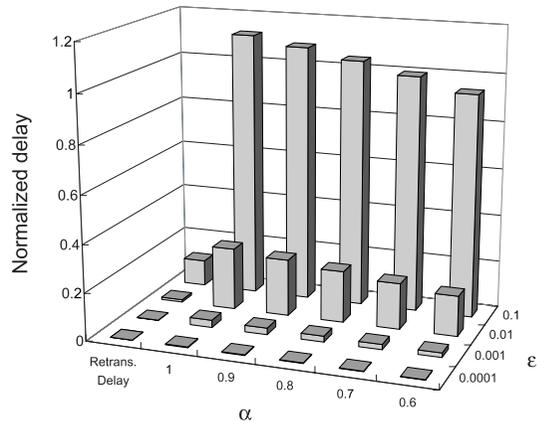


Fig. 10 Average retransmission delay and resequencing delay vs. α ($M = 64, N_r = 3, \beta = 1.0$).

because of the different number of retransmissions. In the case where the number of retransmissions is the same, for example from 0 to 63 slots, the probability increases as the delay becomes large. This is because a packet sent after a retransmitted packet in a short time interval, is likely to be involved in the resequencing and its resequencing delay will also be large.

4.2 In the Case Where N_r is Changed

Table 2 shows the delay and packet loss rate

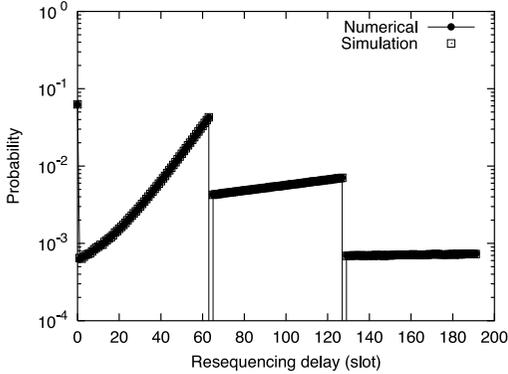


Fig. 11 Distribution of the resequencing delay ($\varepsilon = 0.1, M = 64, N_r = 3, \alpha = 0.9, \beta = 1.0$).

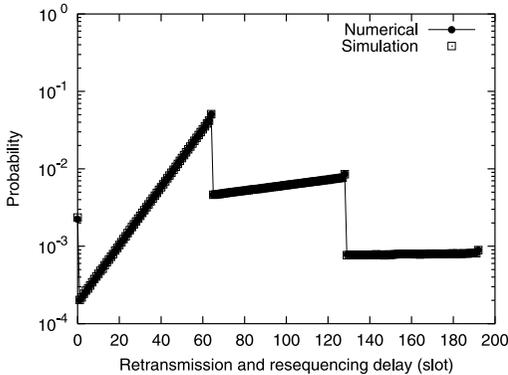


Fig. 12 Distribution of the delay, including both retransmission and resequencing ($\varepsilon = 0.1, M = 64, N_r = 3, \alpha = 0.9, \beta = 1.0$).

p_L for the cases where $\alpha = 0.9$ and the maximum number of retransmissions N_r is changed from 1 to 5. Other conditions are the same as in the case shown in Table 1. When N_r is small ($= 1$), though the delay becomes small, the packet loss rate is non-negligible for large packet error rates. When N_r is larger than 3, the packet loss rate becomes small, and the delay is almost the same for cases where N_r is 4 and 5.

4.3 In the Case Where β is Changed

Figure 13 shows the average retransmission delay and the average resequencing delay for various values of β , where $\alpha = 0.9$. As mentioned in the case of Fig. 10, the left column indicates the average retransmission delay, while the other columns show the average resequencing delay for various values of β . The delay is also normalized by the number of slots M in a frame. $\beta = 1.0, 0.5, 0.2,$ and 0.1 correspond to the cases where the number of homogeneous upper-layer flows are 1, 2, 5, and 10, respec-

Table 2 Packet loss rate, average retransmission delay (slot), average resequencing delay (slot), and average delay due to both retransmission and resequencing (slot).

$\alpha = 0.9, \beta = 1.0$			retrans.	reseq.	total
N_r	ε	p_L	delay	delay	delay
1	10^{-4}	10^{-8}	0.0064	0.1811	0.1875
	10^{-3}	10^{-6}	0.0639	1.7778	1.8417
	10^{-2}	10^{-4}	0.6337	14.9246	15.5583
	10^{-1}	10^{-2}	5.8182	47.2160	53.0342
2	10^{-4}	10^{-12}	0.0064	0.1811	0.1875
	10^{-3}	10^{-9}	0.0641	1.7831	1.8472
	10^{-2}	10^{-6}	0.6463	15.3740	16.0203
	10^{-1}	10^{-3}	6.9189	65.0906	72.0095
3	10^{-4}	10^{-16}	0.0064	0.1811	0.1875
	10^{-3}	10^{-12}	0.0641	1.7831	1.8472
	10^{-2}	10^{-8}	0.6465	15.3821	16.0286
	10^{-1}	10^{-4}	7.0855	69.4644	76.5499
4	10^{-4}	10^{-20}	0.0064	0.1811	0.1875
	10^{-3}	10^{-15}	0.0641	1.7831	1.8472
	10^{-2}	10^{-10}	0.6465	15.3822	16.0287
	10^{-1}	10^{-5}	7.1079	70.2163	77.3242
5	10^{-4}	10^{-24}	0.0064	0.1811	0.1875
	10^{-3}	10^{-18}	0.0641	1.7831	1.8472
	10^{-2}	10^{-12}	0.6465	15.3822	16.0287
	10^{-1}	10^{-6}	7.1107	70.3243	77.4350

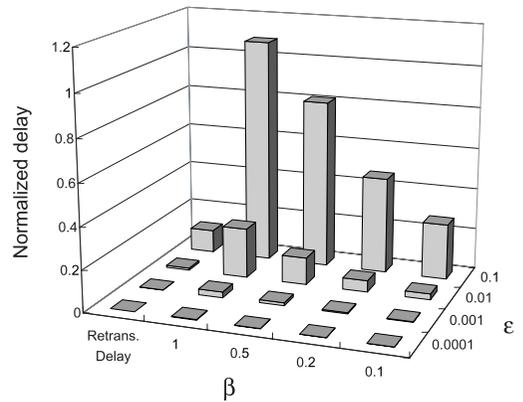


Fig. 13 Average retransmission delay and resequencing delay vs. β ($M = 64, N_r = 3, \alpha = 0.9$).

tively. The conventional full resequencing corresponds to the case in which $\beta = 1.0$, where the order of all packets is preserved irrespective of the number of upper-layer flows. It is clear that the resequencing delay decreases drastically as the number of upper-layer flows increases. The significant advantage of the PFRS scheme can be clearly observed by comparing the delay in the case where β is small to the delay in the case where $\beta = 1.0$. For example,

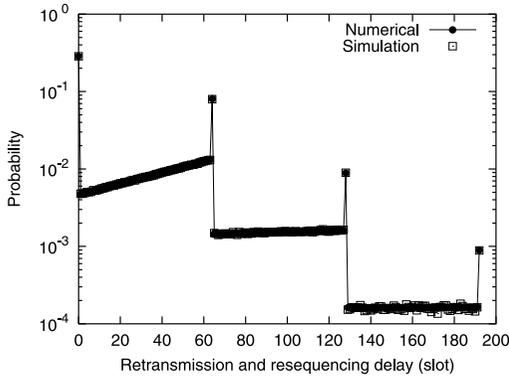


Fig. 14 Distribution of the delay, including both retransmission and resequencing ($\varepsilon = 0.1, M = 64, N_r = 3, \alpha = 0.9, \beta = 0.2$).

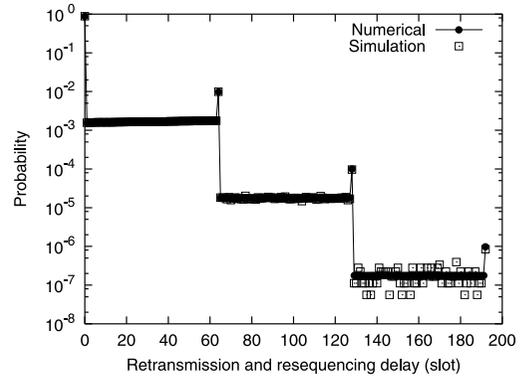


Fig. 15 Distribution of the delay, including both retransmission and resequencing ($\varepsilon = 0.01, M = 64, N_r = 3, \alpha = 0.9, \beta = 0.2$).

when the packet error rate ε is 0.01, the average resequencing delay in the case where $\beta = 1.0$ is 15.4 slots, while the average delay in the case where $\beta = 0.1$ is 1.8 slots. The resequencing delay of the PFRS scheme is reduced to 11.7% of the delay of the conventional scheme. The reason for this is that the number of packets in one flow becomes small, and therefore the number of packets involved in the resequencing also decreases.

The number of flows that are simultaneously active over a transmission channel depends on the number of users and applications. It is reported that a certain web browser opens up to 8 TCP connections per web server using HTTP/1.0 and 2 TCP connections using HTTP/1.1¹⁰). This implies that the PFRS scheme can be normally expected to be advantageous even in a single-user case. **Figure 14** shows the delay probability distribution due to the retransmission and resequencing, where the number of upper-layer flows is 5 ($\beta = 0.2$). Other conditions are the same as in the case of Fig. 12. **Figure 15** shows the delay probability distribution due to the retransmission and resequencing for the case where the conditions are the same as in Fig. 14, except that the packet error rate ε is 0.01. The results of the simulations and numerical calculations also overlap considerably, which proves the validity of the analysis. In these figures, peaks due to retransmission delays (multiples of M slots) become outstanding as the probability of resequencing delay decreases because of the per-flow resequencing.

Figure 16 shows the delay probability distribution where the rates of flows are not the same. In this figure, the number of upper-layer

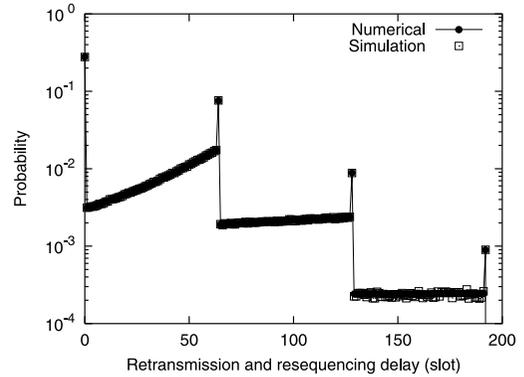


Fig. 16 Distribution of the delay, including both retransmission and resequencing, for multiple flow rates ($\varepsilon = 0.01, M = 64, N_r = 3, \alpha = 0.9, \beta_1 = 0.5, \beta_i = 0.1, i = 2, 3, 4, 5, 6$).

flows is 6. The probability of one flow (β_1) is 0.5 and the probability of the other five flows ($\beta_i, i = 2, 3, 4, 5, 6$) is 0.1. Other conditions are the same as in the case of Fig. 15. Since the results of simulations are consistent with numerical calculations, it is clear that the analysis in this paper can also be applied to a mixture of different flow rates.

5. Conclusion

In this paper we have presented an analysis of the delay probability distribution of the resequencing delay due to the use of SR ARQ. In this analysis, the maximum number of retransmissions is limited, and the utilization of a transmission channel as well as multiple upper-layer flows is considered. The calculation results are closely consistent with the simulation results. The analysis in this paper can be ap-

plied to both the conventional full resequencing and the newly proposed PFRS schemes. On the basis of the calculation results, it is shown clearly that the resequencing delay is dominant in relation to the retransmission delay, and that this delay can be significantly reduced by the PFRS scheme as the number of flows over SR ARQ increases. This proves the effectiveness of the PFRS scheme. Although we presented numerical calculations and simulations for a satellite channel, the PFRS scheme and the analysis are also useful for high-speed terrestrial wireless communications, where efficient SR ARQ is essential for recovering from transmission errors due to low channel quality.

References

- 1) Konheim, A.G.: A Queueing Analysis of Two ARQ Protocols, *IEEE Trans. Commun.*, Vol.COM-28, No.7, pp.1004–1014 (1980).
- 2) Anagnostou, M.E. and Protonotarios, E.N.: Performance Analysis of the Selective Repeat ARQ Protocol, *IEEE Trans. Commun.*, Vol.COM-34, No.2, pp.127–135 (1986).
- 3) Rosberg, Z. and Shacham, N.: Resequencing Delay and Buffer Occupancy Under the Selective-repeat ARQ, *IEEE Trans. Inf. Theory*, Vol.35, No.1, pp.166–173 (1989).
- 4) Shacham, N. and Shin, B. C.: A Selective-Repeat-ARQ Protocol for Parallel Channels and Its Resequencing Analysis, *IEEE Trans. Commun.*, Vol.40, No.4, pp.773–782, (Apr. 1992).
- 5) 3GPP: TS 25.322, Radio Link Control (RLC) protocol specification (2004).
- 6) Koga, H., Ikegami, T., Hori, Y. and Oie, Y.: Out-of-sequence in Packet Arrivals due to Layer 2 ARQ and Its Impact on TCP Performance in W-CDMA Networks, *Proc. 2003 Symposium on Applications and the Internet (SAINT 2003)*, pp.398–401 (2003).
- 7) Shikama, T. and Mizuno, T.: A Proposal of the Re-sequencing Scheme for the Selective-Repeat ARQ and Its Performance Evaluation, *Trans. IEICE*, Vol.J88-B, No.4, pp.718–727 (2005) (in Japanese).
- 8) Rossi, M. and Zorzi, M.: Analysis and Heuristics for the Characterization of Selective Repeat ARQ Delay Statistics Over Wireless Channels, *IEEE Trans. Vehic. Techn.*, Vol.52, No.5, pp.1365–1377, (Sep. 2003).
- 9) Shikama, T. and Mizuno, T.: A Per Flow Resequencing Scheme Using the Sequence Number of Selective-Repeat ARQ, *Trans. IEICE*, Vol.J88-B, No.6, pp.1017–1028 (2005) (in Japanese).
- 10) Chakravorty, R., Banerjee, S., Rodriguez, P., Chesterfield, J. and Pratt, I.: Performance Optimizations for Wireless Wide-Area Networks: Comparative Study and Experimental Evaluation, *Proc. ACM MobiCom 2004*, pp.159–173 (2004).

Appendix

A.1 Derivation of the State Probability of a Sub-channel

We will derive the state probability of a sub-channel given by Eqs. (1) and (2). On the basis of the state transition diagram shown in Fig. 7, since we consider the equilibrium case, the flow rate into each state is equal to the flow rate out of each state.

$$\alpha \varepsilon p_s(0) = p_s(1)$$

$$\varepsilon p_s(1) = p_s(2)$$

$$\varepsilon p_s(2) = p_s(3)$$

$$\vdots$$

$$\varepsilon p_s(N_r - 1) = p_s(N_r)$$

Then $p_s(k)$, where $1 \leq k \leq N_r$, can be represented by $p_s(0)$.

$$p_s(1) = \alpha \varepsilon p_s(0)$$

$$p_s(2) = \varepsilon p_s(1) = \alpha \varepsilon^2 p_s(0)$$

$$\vdots$$

$$p_s(N_r) = \varepsilon p_s(N_r - 1) = \alpha \varepsilon^{N_r} p_s(0)$$

Since $\sum_{k=0}^{N_r} p_s(k) = 1$, we can obtain $p_s(0)$ as follows:

$$p_s(0)(1 + \alpha \varepsilon + \alpha \varepsilon^2 + \dots + \alpha \varepsilon^{N_r}) = 1$$

$$p_s(0) = \frac{1 - \varepsilon}{1 - \varepsilon + \alpha \varepsilon - \alpha \varepsilon^{N_r+1}}$$

Accordingly, we can obtain the probability of state $p_s(k)$, where $1 \leq k \leq N_r$:

$$p_s(k) = \frac{\alpha(1 - \varepsilon)\varepsilon^k}{1 - \varepsilon + \alpha \varepsilon - \alpha \varepsilon^{N_r+1}}$$

A.2 Derivation of the State Probability of the Number of Remaining Receive Failures

We will derive the state probability of the number of remaining receive failures given by Eqs. (4) and (5). We have to consider two cases: one where the number of remaining receive failures is 0, and the other where it is not 0.

The number of remaining receive failures becomes 0 in one of the following cases:

- If the state is 0, there is no packet to be transmitted, or a packet sent with the probability α is successfully received.

- If the state is k , where $1 \leq k \leq N_r$, the current packet is successfully received at the next retransmission.
- If the state is N_r , the state returns to 0 with probability 1 at the next retransmission.

Thus the probability $p_r(0)$ can be calculated as follows:

$$\begin{aligned}
 p_r(0) &= \{1 - \alpha + \alpha(1 - \varepsilon)\}p_s(0) \\
 &+ \sum_{k=1}^{N_r-1} (1 - \varepsilon)p_s(k) + p_s(N_r) \\
 &= \frac{(1 - \alpha\varepsilon)(1 - \varepsilon)}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}} \\
 &+ \frac{(1 - \varepsilon)^2\alpha \sum_{k=1}^{N_r-1} \varepsilon^k}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}} \\
 &+ \frac{\alpha(1 - \varepsilon)\varepsilon^{N_r}}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}} \\
 &= \frac{1 - \varepsilon}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}}.
 \end{aligned}$$

When $1 \leq r \leq N_r$, the number of remaining receive failures becomes r in one of the following cases:

- If the state is 0, a packet sent with the probability α is received successfully after receive failures have occurred r times.
- If the state is k , where $1 \leq k \leq N_r - r - 1$, the current packet is successfully received after receive failures have occurred r times.
- If the state is $N_r - r$, the state returns to 0 with probability 1 after receive failures have occurred r times.

Thus the probability $p_r(r)$ can be calculated as follows:

$$\begin{aligned}
 p_r(r) &= \alpha\varepsilon^r(1 - \varepsilon)p_s(0) \\
 &+ \sum_{k=1}^{N_r-r-1} (1 - \varepsilon)\varepsilon^r p_s(k) \\
 &+ \varepsilon^r p_s(N_r - r) \\
 &= \frac{\alpha(1 - \varepsilon)\varepsilon^r}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}}.
 \end{aligned}$$

(Received May 17, 2005)

(Accepted November 1, 2005)

(Online version of this article can be found in the IPSJ Digital Courier, Vol.2, pp.81–93.)



Toshihiro Shikama received his B.E and M.E. degrees from Tokyo Institute of Technology in 1974 and 1976, respectively. From 1984 to 1985, he was at University of Waterloo, where he received the MASc degree. Since joining Mitsubishi Electric Corp. in 1976, he has engaged in developing a computer network using a satellite channel, high speed ring type LANs, time division multiplexers, ATM equipment, a high speed IP switch, and network security systems. He is a member of IPSJ, IEICE, and IEEE Communications Society.



Shoichiro Seno received his B.S. and M.S. degrees in applied physics from Tokyo Institute of Technology, Tokyo, Japan, in 1981 and 1983, respectively. Since joining Mitsubishi Electric Corporation in 1983, he has been engaged in R&D of LAN internet-networking, high-speed protocol processing, multi-protocol router, network security, mobile networking, and control of optical networks. He is currently a head researcher at Information Technology R&D Center, Mitsubishi Electric Corporation, Japan. Mr. Seno is a member of IPSJ and IEICE.



Takashi Watanabe received the M.E. and D.E. degrees from Osaka University, Japan in 1984, and 1987, respectively. In 1987, he joined the Faculty of Engineering, Tokushima University as an assistant professor. In 1990, he moved to the Faculty of Information, Shizuoka University. He was a visiting researcher at University of California, Irvine in 1995. He is currently a professor of Faculty of Informatics, Shizuoka University. His interests include computer networks and distributed system, especially MAC/Routing protocols for ad hoc and sensor networks. He is a member of JPSJ, IEICE, IEEE Communications/Computer Society, and ACM SIGMOBILE. He is currently Chair of the special interest groups of mobile computing and ubiquitous communications (MBL) of IPSJ.



Tadanori Mizuno received the B.E. degree in industrial engineering from the Nagoya Institute of Technology in 1968 and received the Ph.D. degree in computer science from Kyushu University, Japan, in 1987. In 1968, he joined Mitsubishi Electric Corp. Since 1993, he is a Professor of Faculty of Engineering, Shizuoka University, Japan. He moved to the Faculty of Information, Shizuoka University in 1995. His research interests include mobile computing, distributed computing, computer networks, broadcast communication and computing, and protocol engineering. He is a member of IPSJ, IEICE, IEEE Computer Society, and ACM.
