Regular Paper

A Proposal for Service Differentiation by a Link Layer Protocol Based on SR ARQ and Its Evaluation

Toshihiro Shikama,† Tetsushi Matsuda,††
Yoshiaki Terashima,†† Takashi Watanabe†††
and Tadanori Mizuno†††

This paper studies the situation where multiple IP flows are aggregated over a single wireless channel and an error recovery by retransmissions is performed by Selective-Repeat (SR) ARQ. We propose MQ-PFRS (Multi-QoS Per-Flow Resequencing) ARQ that provides a differentiated service for an IP flow depending on its QoS class, such as real-time or non-real-time. MQ-PFRS ARQ eliminates interferences among IP flows by resequencing received packets independently for each IP flow. It also controls the maximum packet delay by limiting the persistency of retransmissions and the maximum resequencing time for each packet. This paper also presents an analysis of the probability distribution of real-time packet delays. Simulation results show that the delay variation of real-time traffic is effectively controlled by proposed MQ-PFRS ARQ and the packet delay distribution is quite consistent with the results of the analysis. This means that MQ-PFRS is effective for handling multiple QoS classes of IP flows and the quality of real-time traffic transferred by the scheme can be predicted by the analysis.

1. Introduction

Wireless communications supporting mobile terminals require high performance error recovery, where retransmissions by ARQ (Automatic Repeat reQuest) are generally required in addition to FEC (Forward Error Correction). In the case of wired networks, lost segments are generally recovered using end-to-end retransmissions by TCP, assuming that loss of segments is caused by congestion of a network. Since TCP reduces its congestion window when it retransmits a lost segment, the performance of TCP is reduced unnecessarily when segments are lost due to transmission errors. In order to solve this problem, local retransmissions which hide transmission errors over a wireless channel and provide improved transmission quality to upper layer protocols are employed. These local retransmissions are performed by the ARQ function at the data link layer, where a connection is generally established to perform the layer functions including ARQ. In the following description, we introduce the term “an ARQ connection” to refer to a data link layer connection. We also call a protocol data unit at the data link layer “a frame” and a protocol data unit at the network layer “a packet” or “an IP packet”.

In this paper, we study the situation where a number of IP flows are transferred over an ARQ connection on a channel. Each of these IP flows belongs to one of the QoS classes, which are defined by requirements with respect to the average packet delay, the variation of the delay, the packet loss rate, and so on. It is well understood that retransmissions by ARQ incur large delays and their variations; ARQ is conventionally employed only for non-real-time traffic, where a negligibly small packet loss rate is required while packet delays are not bounded. However, when the quality of transmission is seriously degraded due to a poor channel condition, limited retransmissions can be effective to improve the packet loss rate of other QoS classes, where packet delays are required to be bounded.

To date, service differentiation has not been considered among IP flows passing through a single ARQ connection; homogeneous non-real-time traffic has been assumed. In this paper, we propose MQ-PFRS (Multi-QoS Per-Flow Resequencing) ARQ which is able to recover packet losses effectively and to provide multiple QoS classes over a single ARQ connection flexibly.

In general, multiplexing a number of IP flows over a single ARQ connection causes undesirable interferences among IP flows and QoS

† Fukui University of Technology
†† Mitsubishi Electric Corporation
††† Shizuoka University

classes due to, for example, queueing and resequencing delays. To solve this problem, one approach is to employ an ARQ connection for each IP flow. However, this approach is not realistic and incurs large processing overheads, since it needs the detection of a start and an end of each flow, and an establishment and a release of an ARQ connection associated with each flow have to be performed. There is also an overhead of independent acknowledgments on each ARQ connection.

To achieve a simple control and suppress overheads, it is desirable that a fixed ARQ connection is applied to aggregated IP flows. Another approach is to employ an ARQ connection for each QoS class, where IP flows of the same class are multiplexed on an associated ARQ connection. This approach still has a problem of aggregating multiple ARQ connections over a physical bandwidth. MQ-PFRS ARQ is able to achieve multiple QoS classes over a single ARQ connection flexibly by eliminating interferences among IP flows over the connection as much as possible.

The rest of this paper is organized as follows: Section 2 explains the basic architecture of MQ-PFRS ARQ and Section 3 presents the per-flow resequencing with limited packet suspension time. Section 4 shows an analysis of the delay distribution of real-time traffic carried by the proposed scheme. Section 5 describes the simulation model as well as its conditions and presents simulation results. Section 6 describes related work. Finally our conclusions are presented in Section 7.

2. The Architecture of MQ-PFRS ARQ

MQ-PFRS ARQ is based on Selective-Repeat (SR) ARQ with PFRS (Per-Flow Re-Sequencing) [1], which enables independent resequencing for each IP flow at the receiving side of SR ARQ. Although SR ARQ is able to achieve high throughput at a relatively high packet loss rate, it requires reordering of packets at the receiving side. When a number of IP flows are multiplexed over a single ARQ connection, the following delays, beside packet retransmission delays, increase due to the interferences among IP flows.

- Queueing delay at a transmission queue
- Queueing delay at a retransmission queue
- Resequencing delay at the receiving side

The resequencing, which is the last item, causes a large delay when multiple IP flows are aggregated over a channel.

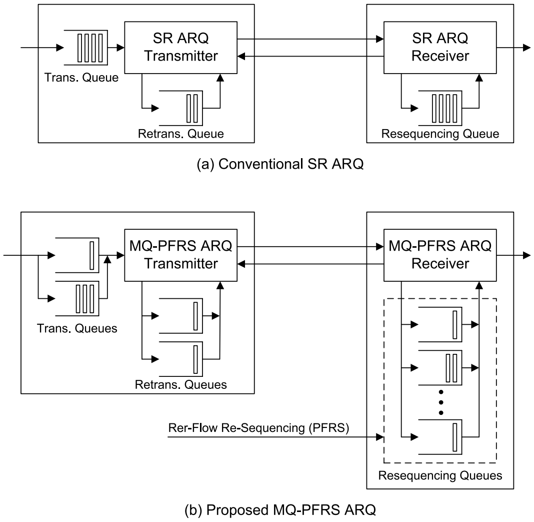

Fig. 1 Construction of queues in conventional ARQ and in MQ-PFRS ARQ.

In the subsequent explanation, we assume two kinds of QoS classes: non-real-time and real-time. But it is possible to define other multiple QoS classes. As explained before, the non-real-time class requires a negligibly small packet loss rate and packet delays that are not bounded. The real-time class requires bounded packet delays but allows the packet loss rate, which is below a specific value.

Figure 1 shows an equivalent configuration of queues in conventional SR ARQ and in the proposed MQ-PFRS ARQ. In MQ-PFRS ARQ, the following mechanisms are introduced into conventional SR ARQ to eliminate the interferences among IP flows and QoS classes.

(1) The sending side of MQ-PFRS ARQ provides two transmission queues corresponding to the QoS classes. New packets in the real-time queue are sent with a high priority.

(2) A retransmission queue is also provided for each QoS class. Retransmissions of real-time packets are performed with the highest priority.

(3) Resequencing needed at the receiving side of SR ARQ is performed for each IP flow independently.

(4) Persistency of retransmissions is limited by the maximum number of retransmissions. The maximum number of retransmissions is differentiated between QoS classes.

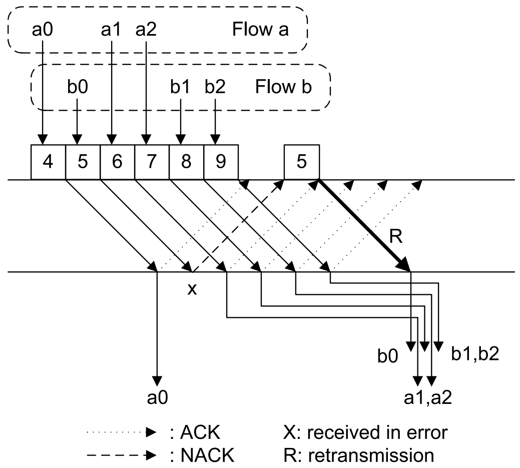

Fig. 2 An example sequence of the conventional SR ARQ.

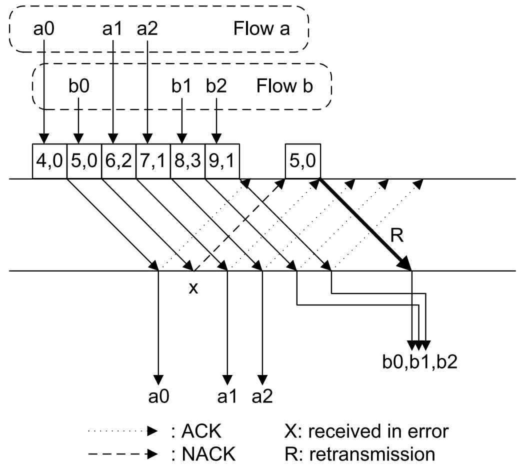

Fig. 3 An example sequence of PFRS ARQ.

(5) Resequencing delay (holding time of a packet by resequencing) is limited by a timer. If a lost packet is not received during the timer period, packets retained by the resequencing are transferred to the upper layer when a timeout occurs. The timer value is associated with the maximum number of retransmissions and is differentiated between the QoS classes.

Although the functions of (1) and (2) are straightforward, (3) was introduced in the PFRS scheme [1]. The functions of (4) and (5) are new to the service differentiation.

We will explain the necessity for mechanisms (3), (4), and (5). Conventional SR ARQ, which preserves packet order for the whole packet flows over SR ARQ, has problems of unnecessary suspension of packets and associated delays. **Figure 2** illustrates this problem; two flows, a and b, are multiplexed over an SR ARQ connection. Packet b0 is lost due to transmission errors and retransmitted. In the case of conventional SR ARQ, packets a1 and a2 of flow a are unnecessarily retrained until the lost packet b0 is retransmitted and received correctly. This situation is a kind of head of line (HOL) blocking for flow a. In the case of MQ-PFRS ARQ, as illustrated in **Fig. 3**, the resequencing is performed for each flow independently, while acknowledgments and retransmissions of packets are done by SR ARQ as the conventional way, and packets a1 and a2 are delivered to the upper layer without being retained. Thus, MQ-PFRS ARQ resolves the invalid suspension of packets due to the HOL blocking. In Fig. 3, an additional number is assigned to each SR ARQ sequence number. We call this number a pointer and will explain it in the next section.

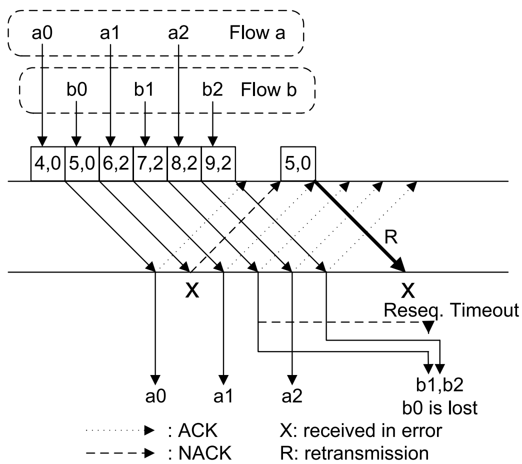

Fig. 4 An example sequence of MQ-PFRS ARQ.

The conventional SR ARQ has a problem if persistency of retransmissions is limited to suppress retransmission delays. SR ARQ enters a deadlock condition waiting for a lost packet which will not be retransmitted any further. Then, if the persistency of retransmissions is limited, waiting time by resequencing also has to be bounded by a timer. **Figure 4** shows an example sequence of a resequencing timeout in proposed MQ-PFRS ARQ. In this figure, we assume that the maximum number of retransmissions is limited to 1. As retransmitted packet b0 is lost again, this packet will never be recovered. Packets b1 and b2 waiting for packet b0 are transferred to the upper layer when a resequencing timeout for packet b0 occurs. The resequencing timer value T_m has to be determined according to the following equation: where N_r is the maximum number of retransmissions of the received packet, R is the number of retransmissions of the received packet, and T_r is a round-trip delay from the start of successful packet transmission to the reception of its acknowledgment. We also define by $\triangle t$ a margin for the variation of the round-trip delay.

$$T_m = T_r \cdot (N_r - R) + \triangle t. \quad (1)$$

The value of N_r is different depending on the QoS classes. The reason for this dynamic adjustment of the resequencing timer value is that if a packet is correctly received after retransmissions of R times, its maximum resequencing delay is reduced by $T_r \cdot R$.

3. Per-flow Resequencing with Limited Packet Suspension Time

In order to realize MQ-PFRS ARQ, identification of upper layer flows is required. This is done by referring to the IP and TCP or UDP headers of a packet. The receiving side also needs information concerning the order of packets carried by frames. It is desirable that a header of each frame includes both the flow identification and the order information of a packet carried by the frame.

We employ a pointer, proposed by the paper [4)], in a frame header to identify an IP flow and the order of a packet carried by the frame. The pointer can be used even where loss of frames occurs as a result of limited resequencing time. Each frame carrying a packet includes both the conventional SR ARQ sequence number and the pointer, which is the number introduced in Fig. 3. The pointer is the difference between the MQ-PFRS ARQ sequence number of the current frame (SN_{cur}) and the MQ-PFRS ARQ sequence number of the previous frame (SN_{prev}) that carries a packet of the same flow to which the current packet belongs.

$$Pointer = (SN_{cur} - SN_{prev}) \bmod 2^N, \quad (2)$$

where N is the number of bits of the pointer. If there is no need of resequencing, we use 0 as the value of the pointer. The pointer is added to the conventional SR ARQ header; its format is depicted in **Fig. 5**. In this figure, C identifies a class of traffic, for example real-time or non-real-time, while R identifies the number of retransmissions of the current frame. The actual value of the sequencing timer is determined

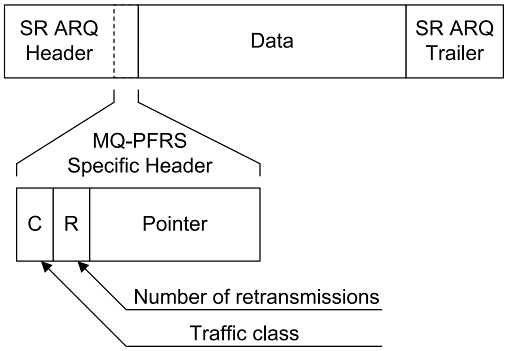

Fig. 5 MQ-PFRS ARQ specific frame header information.

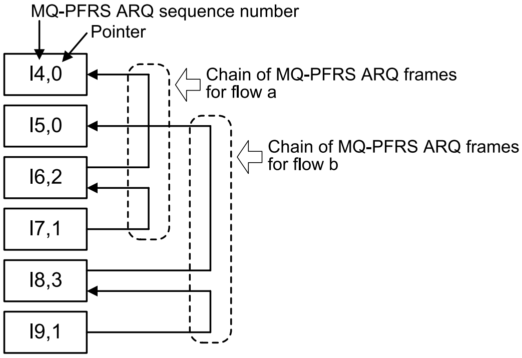

Fig. 6 Identification of frames belonging to the same flow by the pointer.

based on this number as explained in Eq. (1).

Figure 6 illustrates the identification of frames carrying packets of the same flow for the sequence illustrated in Fig. 3. We denote a frame as Ix,y, where x is the sequence number of SR ARQ, while y is the pointer value. The pointer, which is introduced in addition to the conventional sequence number, identifies an IP flow and the order of packets belonging to the same flow as presented. **Figure 7** shows an example of resequencing by using the pointer for the sequence illustrated in Fig. 4. In this example we assume a table that stores out-of-sequence frames and associated next-frame-pointers. Although the pointer in a frame is in reverse direction (decreasing order of the sequence number), a pointer of in-sequence direction (increasing order of the sequence number), which is called the next-frame-pointer in Fig. 7, is needed. The next-frame-pointer is acquired by storing the sequence number of an out-of-sequence frame in the row identified by the pointer of the out-of-sequence frame. When an in-sequence packet of an IP flow is received,

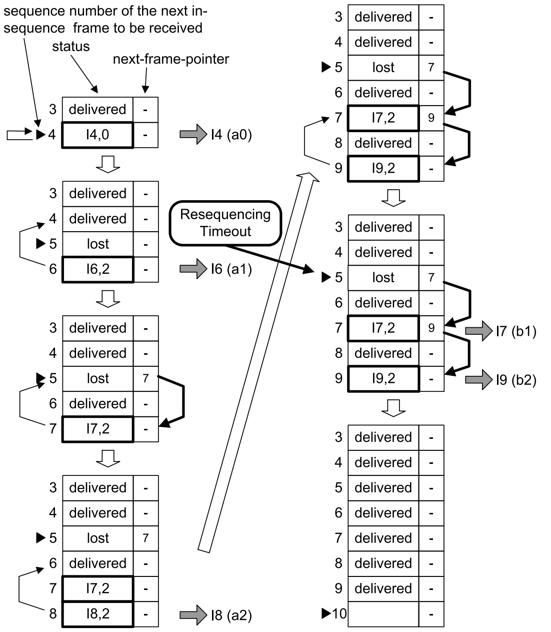

Fig. 7 An example of resequencing procedures including a timeout.

or a resequencing timeout occurs, the release of retained packets is performed effectively by using the next-frame-pointer as illustrated in Fig. 7.

When frame I4,0 is received, since its pointer value is 0, a packet carried by this frame is forwarded to the upper layer without being retained. As its sequence number is also the value which is expected, the status of this sequence number is marked as delivered and the sequence number of the next one expected to be received is updated by 1. After frame I5,0 is lost, frame I6,2 is received. Since the pointer value of this frame is 2 and its ARQ sequence number is 6, the receiver checks whether a packet of frame I4 (4=6−2) has been delivered to the upper layer. The packet of frame I4 has already been delivered, and the packet of frame I6 is also delivered to the upper layer without being retained. Next, when frame I7,2 is received, the receiver checks, whether frame I5 (5=7−2) has been delivered. As frame I5 has not been received yet, frame I7,2 is retained due to the resequencing and a resequencing timer is assigned to the lost frame I5 and is invoked. Since it becomes clear that a packet of frame I7 follows a packet of frame I5, the receiver writes value 7 in the row of sequence number 5 as the next-frame-pointer. When frame I8,2 is received, since a packet of frame I6 (6=8−2) has been delivered, a packet carried by frame I8,2 is also delivered. When frame I9,2 is received, since frame I7 has been retained, this frame is also retained for the resequencing. Since the receiver can find that frame I9 is the subsequent frame of I7, it writes value 9 in the row of sequence number 9 as the next-frame-pointer. When the resequencing timer expires, the receiver delivers the packets of frames I7 and I9 according to the next-frame-pointers. Then the sequence number of the next in-sequence frame is updated from 5 to 10, since frames of sequence number 6 to 9 have all been delivered. If lost I5,0 is received before the timer expiration, though this instance is not depicted in the figure, the receiver delivers packets of frames I5, I7, and I9 according to the next-frame-pointers, and stops the resequencing timer. As explained above the per-flow resequencing by the pointer scheme can be applied where the resequencing delay is limited.

As pointed out in the paper [4], the pointer scheme has the following advantages.

- There is no limitation on the number of IP flows multiplexed over MQ-PFRS ARQ, provided that the number of bits for the pointer is equal to that of MQ-PFRS ARQ sequence number.
- There is no need of synchronization between a sender and a receiver.
- The receiver just performs the resequencing based on the pointer value. This scheme is also effective for handling a large number of short life time flows.

Although MQ-PFRS ARQ has to refer to the upper layer header to identify flows, it never modifies the header; it preserves end-to-end semantics of upper layer protocols.

4. Delay Analysis of the Real-time Packets

In the case of real-time traffic, we assume an admission control so as to not overflow the bandwidth of a channel. Therefore, queueing delays at the transmission queue and the retransmission queue are relatively small for this traffic. Delays due to retransmission and resequencing are considered to be dominant. The numerical analysis of the delay and buffer occupancy due to the resequencing has been carried out where a channel is fully loaded and the number of retransmissions is not limited [2]. There is also an analysis of the resequencing delay distribution, where the maximum number of retransmissions is limited and a channel is par-

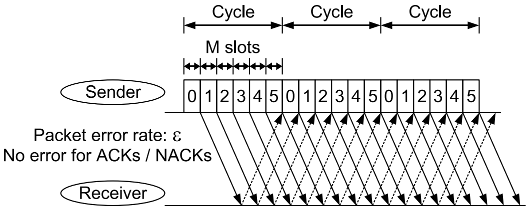

Fig. 8 An SR ARQ model employed in the analysis.

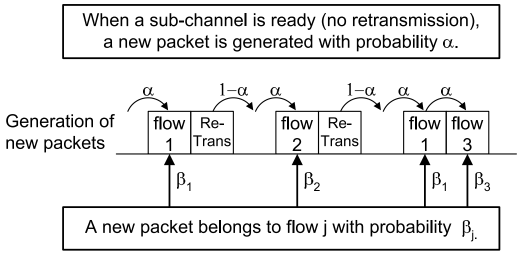

Fig. 9 A packet generation model of multiple real-time flows.

tially loaded[3)]. If the effect of non-real time traffic can be ignored and the length of real-time packets is fixed, the delay distribution of the real-time packets can be calculated by using the results[3)] as follows:

Figure 8 shows an SR ARQ model employed in the analysis. It is assumed that time is segmented into fixed size slots, in each of which a real-time packet is sent. We define M as the number of slots in a cycle of which duration is equal to the sum of a round-trip delay in the unit of slots and one slot for sending a packet. It is also assumed that a packet error occurs at random with the rate ε.

Figure 9 shows a packet generation model of multiple real-time flows. We assume that a packet of whole real-time flows is generated at random with probability α in each slot, if it is ready for a new packet (no retransmission), and a packet belonging to real-time flow j also occurs at random with probability β_j among packets of all real-time flows. We assume that the rate of each real-time flow is the same, then β_j is the same for all j ($1 \le j \le N_{RT}$) with the value $1/N_{RT}$, where N_{RT} is the number of real-time flows. The packet loss rate P_L for real-time flows is given by,

$$P_L = \varepsilon^{N_r+1}. \tag{3}$$

We define $P_{delay}(u, i|\beta)$ as the probability that the sum of the retransmission delay and resequencing delay is $uM + i$ slots, where u is the number of retransmissions ($0 \le u < N_r$) and $0 \le i < M$. From the results of the paper[3)], $P_{delay}(u, i|\beta)$ can be obtained as follows.

Where resequencing occurs ($i \neq 0$)

$$P_{delay}(u, i|\beta) = \beta q_t(u) p_r(u+1) \cdot \frac{U(u|\beta)^{M-1}}{U(u+1|\beta)} \cdot \left(\frac{U(u+1|\beta)}{U(u|\beta)}\right)^i, \tag{4}$$

where $q_t(u)$, $p_r(u+1)$ and $U(n|\beta)$ are given by as follows:

$$q_t(u) = \frac{1-\varepsilon^{u+1}}{1-\varepsilon^{N_r+1}}, \tag{5}$$

$$p_r(u+1) = \frac{\alpha(1-\varepsilon)\varepsilon^{u+1}}{1-\varepsilon+\alpha\varepsilon-\alpha\varepsilon^{N_r+1}}, \tag{6}$$

$$U(n|\beta) = 1 - \frac{\alpha\beta(\varepsilon^{n+1}-\varepsilon^{N_r+1})}{1-\varepsilon+\alpha\varepsilon-\alpha\varepsilon^{N_r+1}}. \tag{7}$$

Where there is no resequencing ($i = 0$)

$$P_{delay}(u, 0|\beta) = p_t(u) U(u|\beta)^{M-1}. \tag{8}$$

Let R_{CH}, R_{RT}, and L_{RT} be the rate of a wireless channel, the bit rate of UDP traffic, and the byte length of a UDP packet, respectively. We also define L_{FOH} as the overhead of a frame in bytes. If the rates of all real-time flows are the same with the value R_{RT}, the parameters α and β can be calculated as follows:

$$\alpha = \frac{N_{RT} \cdot R_{RT} \cdot (L_{FOH} + L_{RT})}{R_{CH} \cdot L_{RT}}, \tag{9}$$

$$\beta = 1/N_{RT}. \tag{10}$$

Let B_{ERR} be the bit error rate of the wireless channel. The packet error rate ε is given by

$$\varepsilon \approx B_{ERR} \cdot 8 \cdot (L_{FOH} + L_{RT}). \tag{11}$$

A slot duration T_s is calculated as,

$$T_s = \frac{L_{FOH} + L_{RT}}{R_{CH}}. \tag{12}$$

We define the round trip delay of the wireless channel as T_r. The parameter M can be calculated as follows:

$$M = \left\lceil \frac{T_r}{T_s} \right\rceil + 1. \tag{13}$$

Using the parameters α, β, ε, and M, we can obtain the delay distribution by Eqs. (4) and (8). We will compare the numerical values derived from these equations with simulation results in the next section.

5. Simulation Model, Its Results, and Comparison with the Analysis

In order to evaluate the advantages of MQ-PFRS ARQ, simulations were performed. We employed the VINT network simulator ns-2 [5] and added MQ-PFRS ARQ specific functions to SSCOP (Service Specific Connection Oriented Protocol) [6], which is used as a base SR ARQ protocol. We assume a network configuration depicted in **Fig. 10**, where MQ-PFRS ARQ is performed at the data link layer on a satellite channel, over which a number of real-time and non-real-time flows are aggregated. We also suppose that UDP is employed for real-time packets while TCP is employed for non-real-time packets. **Table 1** summarizes simulation conditions. We assume that the overhead due to the MQ-PFRS ARQ header L_{FOH} is 10 bytes, where the new overhead of 4 bytes is added to the original SR ARQ header. These 4 bytes include 3 bytes for the pointer and 1 byte for both QoS class and the number of retransmissions. Segmentation of IP packets at the data link layer is not considered; the MQ-PFRS ARQ header is added to an IP packet, of which size is normally up to 1,500 bytes. In this case, the overhead by the MQ-PFRS ARQ header is not outstanding. If the size of IP packets is small (for example VoIP), the overhead due to the MQ-PFRS ARQ header might be significant. However, the size of the IP header (at least 20 bytes) is also a problem here. One approach to mitigate this is to apply the IP header compression [13]; details of the compression with MQ-PFRS ARQ are left for further study. Suppression of the pointer size for the PFRS scheme is studied in the paper [4]. This can also be applied to mitigate the problem.

The margin for the variation of the round-trip delay Δt in the case of real-time traffic is set to 0.1 sec, while the margin Δt for non-real-time traffic is set to 1.5 sec. We selected the large margin for non-real-time traffic to suppress spurious timeouts of the resequencing timers possibly caused by the large delay variation of non-real-time traffic. Based on the simulation conditions presented in Table 1, the parameters of the analysis can be calculated as shown in **Table 2**.

5.1 Comparison of Analysis and Simulation Results in the Case of MQ-PFRS ARQ

Figures 11 and **12** show complementary cumulative distribution functions (CCDFs) of packet delays by proposed MQ-PFRS ARQ, where the bit error rate B_{ERR} is 10^{-6} and 10^{-5}, respectively. The CCDF shows the probability that packet delays are larger than the value of the x-axis. Results of the analysis are also plotted in these figures. It is clear that the CCDFs by the analysis agree with the characteristics of the simulation results of real-time packets (UDP).

Table 3 shows the comparison of the analysis and simulation results of real-time (UDP) packets in the MQ-PFRS scheme. The comparison includes the packet loss rate, the average delay, and the 99th percentile, which indicates that delays of 99% of packets are less than or equal to this value. The average delay of the

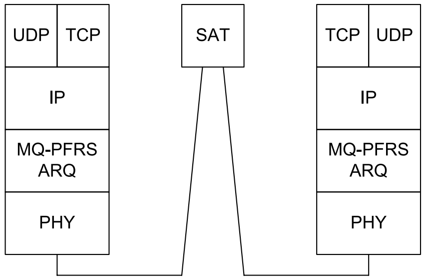

Fig. 10 A network configuration of the simulations.

Table 1 Simulation conditions.

Rate of a sat. channel (R_{CH})	2 M bit/s
Round-trip delay (T_r)	0.25 sec
Bit error rate (B_{ERR})	$10^{-6}, 10^{-5}$
Characteristic of bit errors	Random
SR ARQ protocol	SSCOP with MQ-PFRS
Overhead of a frame (L_{FOH})	10 bytes
UDP packet length (L_{RT})	500 bytes
Bit rate of UDP traffic (R_{RT})	300 kbit/s
Max. No. of retrans. for UDP	1
Δt for real-time traffic (UDP)	0.1 sec
No. of UDP connections (N_{RT})	3
TCP type	NewReno
TCP segment length	1,500 bytes
TCP window size	32 kbytes
Max. No. of retrans. for TCP	3
Δt for non-real-time traffic (TCP)	1.5 sec
Number of TCP connections	3

Table 2 Values of the parameters in the analysis.

M	247	N_r	1
ε	0.00408 (B_{ERR}=10^{-6})	α	0.459
	0.0408 (B_{ERR}=10^{-5})	β	0.333333

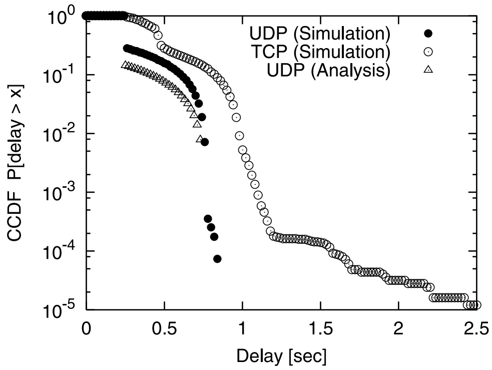

Fig. 11 Complementary Cumulative Distribution Function (CCDF) of real-time (UDP) and non-real-time (TCP) packet delays by MQ-PFRS ARQ, where the bit error rate B_{ERR} is 10^{-6}.

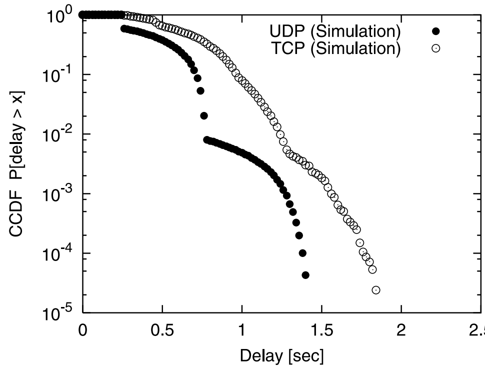

Fig. 13 Complementary Cumulative Distribution Function (CCDF) of real-time (UDP) and non-real-time (TCP) packet delays by conventional SR ARQ with priority queues, where the bit error rate B_{ERR} is 10^{-6}.

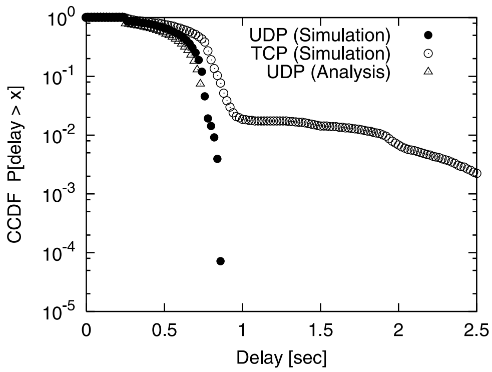

Fig. 12 Complementary Cumulative Distribution Function (CCDF) of real-time (UDP) and non-real-time (TCP) packet delays by MQ-PFRS ARQ, where the bit error rate B_{ERR} is 10^{-5}.

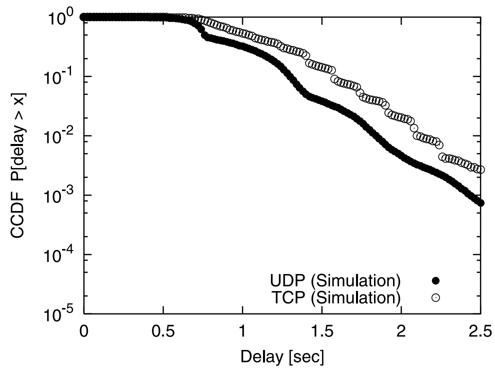

Fig. 14 Complementary Cumulative Distribution Function (CCDF) of real-time (UDP) and non-real-time (TCP) packet delays by conventional SR ARQ with priority queues, where the bit error rate B_{ERR} is 10^{-5}.

Table 3 Analysis and simulation results of real-time (UDP) packets by MQ-PFRS ARQ.

Bit error rate $B_{ERR}= 10^{-6}$	MQ-PFRS ARQ	
	Analysis	Simulation
Packet loss rate	1.65×10^{-5}	1.33×10^{-5}
Average delay (sec)	0.29	0.35
99th percentile (sec)	0.72	0.77

Bit error rate $B_{ERR} = 10^{-5}$	MQ-PFRS ARQ	
	Analysis	Simulation
Packet loss rate	1.65×10^{-3}	1.78×10^{-3}
Average delay (sec)	0.50	0.58
99th percentile (sec)	0.75	0.80

analysis tends to be relatively smaller than the results of simulations. The reason is that the analysis does not consider the variation of the queueing and round-trip (acknowledgment) delays. However, the packet loss rate and the 99th percentile is shown by the analysis to take values close to the simulation results; in particular the difference of the 99th percentile is less than 7% in both bit error rate conditions. This means that the delay distribution of real-time packets in proposed MQ-PFRS ARQ can be predicted by simple analysis, and the quality of service can be prescribed.

5.2 Comparison of MQ-PFRS ARQ with Other ARQ Schemes

Figures 13 and **14** show the CCDFs of packets by conventional SR ARQ with priority queues for the same conditions as Figs. 11 and 12. Neither the per-flow resequencing nor the control of retransmission persistency is performed. Although the delays of real-time packets are much improved when the bit error rate

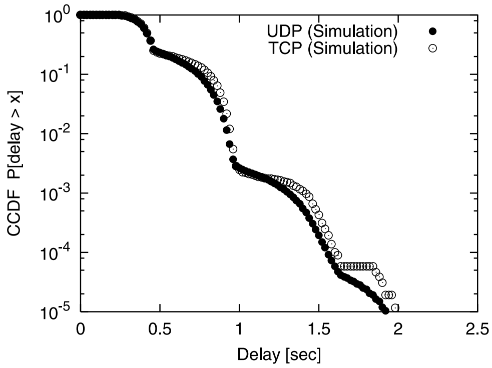

Fig. 15 Complementary Cumulative Distribution Function (CCDF) of real-time (UDP) and non-real-time (TCP) packet delays by PFRS ARQ, where the bit error rate B_{ERR} is 10^{-6}.

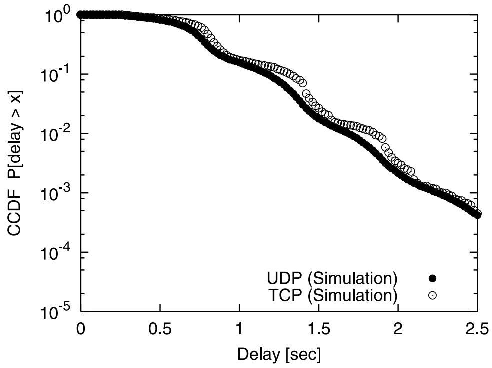

Fig. 16 Complementary Cumulative Distribution Function (CCDF) of real-time (UDP) and non-real-time (TCP) packet delays by PFRS ARQ, where the bit error rate B_{ERR} is 10^{-5}.

is 10^{-6} due to the effect of priority queueing, the range of real-time (UDP) packet delays becomes large when the bit error rate is 10^{-5}.

Figures 15 and **16** show the CCDFs of packets by PFRS ARQ for the same conditions as Figs. 11 and 12. In these figures, the per-flow resequencing is performed; neither the service differentiation by priority queueing nor the control of retransmission persistency is performed. Although the range of the delay distribution is relatively limited when the bit error rate is 10^{-6}, its range is extended when the bit error rate becomes large (10^{-5}).

Table 4 shows the summary of simulation results for real-time (UDP) packets by four SR ARQ schemes consisting of MQ-PFRS ARQ, conventional SR ARQ, conventional SR ARQ with priority queues, and PFRS ARQ. The summary presents the packet loss rate, the average delay, and the 99th percentile. We can

Table 4 Simulation results of real-time (UDP) packets.

ARQ Scheme	MQ-PFRS		Conven.	
Bit error rate	10^{-6}	10^{-5}	10^{-6}	10^{-5}
Packet loss rate	1.33×10^{-5}	1.78×10^{-3}	0.00	0.00
Av. delay (sec)	0.35	0.58	0.59	0.99
99th percentile (sec)	0.77	0.80	1.05	1.90

ARQ Scheme	Conven. with priority queues		PFRS	
Bit error rate	10^{-6}	10^{-5}	10^{-6}	10^{-5}
Packet loss rate	0.00	0.00	0.00	0.00
Av. delay (sec)	0.45	0.92	0.48	0.77
99th percentile (sec)	0.78	1.84	0.93	1.69

Table 5 Simulation results of non-real-time (TCP) packets.

ARQ Scheme	MQ-PFRS		Conven.	
Bit error rate	10^{-6}	10^{-5}	10^{-6}	10^{-5}
Av. delay (sec)	0.52	1.02	0.61	1.07
99th percentile (sec)	0.98	1.91	1.05	2.00

ARQ Scheme	Conven. with priority queues		PFRS	
Bit error rate	10^{-6}	10^{-5}	10^{-6}	10^{-5}
Av. delay (sec)	0.68	1.14	0.50	0.86
99th percentile (sec)	1.24	2.10	0.95	1.85

observe that the average delay and the 99th percentile of real-time (UDP) packets are much improved by MQ-PFRS ARQ as compared with other SR ARQ schemes, but the residual packet loss rate exists instead. The significant feature of MQ-PFRS ARQ is that the delay variation of the real-time packets is effectively controlled against the change of the bit error rate at the cost of the residual packet loss rate. It is clear that either the priority queueing alone or the per-flow resequencing alone is not enough to control the delay variation of real-time packets (UDP).

Table 5 shows the same summary for non-real-time (TCP) packets. Although some advantage of PFRS ARQ is observed, there is no significant difference between the four SR ARQ schemes.

Form the results of these simulation results, it can be said that proposed MQ-PFRS ARQ can achieve service differentiation between real-time traffic and non-real-time traffic by a single

ARQ connection.

6. Related Work

The problem of HOL blocking by the transport protocol is identified where a large number of signaling messages by SIP (Session Initiation Protocol) are transferred over a TCP connection. SCTP (Stream Control Transmission Protocol), which is a new Transport protocol standardized by IETF, accommodates multiple streams over a single SCTP connection and avoids HOL blocking by reassembling data from each stream independently [7]. It is reported that SCTP makes it possible to reduce the volume of receive buffers by avoiding HOL blocking [8]. The basic idea behind the scheme proposed in this paper is the same as the solution adopted by SCTP. However, since SCTP aims at providing no packet loss service, it does not limit the maximum number of retransmissions and associated resequencing delays. There are no differentiated services for streams over an SCTP connection.

There are ARQ protocols that control persistency of retransmissions [11]. For example, Radio Link Control (RLC) is a link layer protocol defined by 3GPP that limits the maximum number of retransmissions [9],[10]. Retransmissions by RLC is performed based on SR ARQ, but it performs the resequencing for whole SDUs (Service Data Units) from the upper layer. Independent persistency of retransmissions for each QoS class is a new approach proposed by this paper.

There is a study on priority scheduling introduced in Layer 2 SR ARQ protocol [12], where real-time (UDP) packets and non-real-time (TCP) packets are identified and stored in separate buffers, and UDP packets are always sent with a high priority. Resequencing of real-time (UDP) packets is also performed separately from non-real-time (TCP) packets, however any realization architecture is not presented. Although this scheme can eliminate interferences between real-time and non-real-time packets, interferences among packets of the same QoS class still remain. The scheme also limits the maximum number of retransmissions, but this number is the same for both real-time and non-real-time packets; it is difficult to specify the optimum value for this number.

As explained before, the proposed scheme can optimize the maximum number of retransmissions and associated maximum resequencing delay depending on the QoS class of each packet. It can also eliminate interferences among IP flows of the same QoS class. These advantages of the proposed scheme have not been presented in previously published work.

7. Concluding Remarks

This paper studied the case where multiple IP flows were aggregated over a single channel and an error recovery by retransmissions was performed by Selective-Repeat ARQ. We have proposed MQ-PFRS ARQ which provides a differentiated service for each IP flow depending on its QoS class. We have explained problems derived from the resequencing function of SR ARQ. MQ-PFRS ARQ can eliminate interferences among IP flows by resequencing received packets independently for each IP flow. It also controls the maximum packet delay by limiting the persistency of retransmissions and the maximum resequencing time for each packet. In order to identify an IP flow and the order of packets, we extended the pointer scheme so that it can cope with a limited suspension time of retained packets. The analysis of the probability distribution of real-time packet delays was also presented. Simulation results show that the delay variation of real-time traffic is effectively controlled by the proposed MQ-PFRS ARQ and the average packet delay and the 99th percentile of packet delays strongly agrees with the results of the analysis. We can conclude that MQ-PFRS ARQ is effective for accommodating multiple kinds of IP flows and the quality of real-time traffic can be predicted by the analysis.

Although the performance of proposed MQ-PFRS ARQ was evaluated for satellite communications, the results of the paper can be generally applied to high-speed wireless networks, where a product of the bandwidth and delay is much larger than the time needed to send a packet.

References

1) Shikama, T. and Mizuno, T.: A proposal of the Re-Sequencing Scheme for the Selective-Repeat ARQ and Its Performance Evaluation, *IEICE Trans. Comm.*, Vol.J88-B, No.4, pp.718–727 (2005) (Japanese Edition).
2) 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).

3) Shikama, T., Seno, S., Watanabe, T. and Mizuno, T.: Delay Analysis of the Selective-Repeat ARQ Protocol with the Per Flow Resequencing Scheme, Online version: *IPSJ Digital Courier*, Vol.2, pp.81–93, http://ipsj.or.jp (2006). Paper version: *IPSJ Journal*, Vol.47, No.2, pp.369–381 (2006).
4) Shikama, T. and Mizuno, T.: A Per Flow Resequencing Scheme Using the Sequence Number of Selective-Repeat ARQ, *IEICE Trans. Commun.*, Vol.J88-B, No.6, pp.1017–1028 (2005) (Japanese Edition).
5) VINT Project, Network Simulator ns-2, http://www.isi.edu/nsnam/ns.
6) ITU-T Recommendation Q.2110: B-ISDN signalling ATM adaptation layer — service specific connection oriented protocol (SSCOP) (1994).
7) Stewart, R., Xie Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and Paxson, V.: Stream Control Transmission Protocol, *IETF RFC 2960* (2000).
8) Fu, S. and Ivancic, W.: SCTP over Satellite Networks, *IEEE CCW 2003*, pp.112–116 (2003).
9) 3GPP: 3G TS 25.322 V.3.5.0: RLC Protocol Specification (2000).
10) Inamura, H., Ishikawa, T., Atsumi, Y. and Takahashi, O.: Evaluation of Link ARQ and TCP over W-CDMA Network, *IPSJ Journal*, Vol.43, No.12, pp.3859–3868 (2002) (Japanese Edition).
11) Fairhurst, G. and Wood, L.: Advice to Link Designers on Link Automatic Repeat Request (ARQ), *RFC 3366* (2002).
12) Koga, H., Morita G. and Oie, Y.: On Quality of Real-time Communication in W-CDMA Networks and Its Improvement, Technical Report of IEICE, IN2002-232, pp.17–22 (2003) (Japanese Edition).
13) Degermark, M., Nordgren, B. and Pink, S.: IP Header Compression, *IETF 2507* (1999).

(Received July 18, 2006)
(Accepted April 6, 2007)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.3, pp.456–467.)

Toshihiro Shikama received his B.E. and M.E. degrees from Tokyo Institute of Technology in 1974 and 1976, respectively. He received his Ph.D. degree from Shizuoka University in 2006. Since joining the 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. Since April of 2007, he has been a professor of the Department of Electrical and Electronic Engineering, Fukui University of Technology. He is a member of IPSJ, IEICE, and IEEE Communications Society.

Tetsushi Matsuda received his B.E. and M.E. from the University of Tokyo in 1985 and 1987, respectively. He works for the Mitsubishi Electric Corp. His research interests include communication systems and distributed computing systems. He is a member of IPSJ and IEICE.

Yoshiaki Terashima is a head researcher in the Information Technology R & D Center, Mitsubishi Electric Corporation in Japan. He received his Ph.D. degree from Shizuoka University. He is currently leading the Disaster Rescue Network Project that realizes network-sensor data fusion based on an ad-hoc disaster rescue network. His research interests include distributed processing architecture and a testing methodology for an ad-hoc network.

Takashi Watanabe received his B.E., M.E. and Ph.D. degrees in communications engineering from Osaka University in 1982, 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 Engineering, Shizuoka University. He was a visiting researcher at the University of California, Irvine from 1995 through 1996. He is currently a professor of the Graduate School of Science and Technology, Shizuoka University. He is a translator of the Japanese version of 802.11 Wireless Networks: The Definitive Guide, 2nd edition (published by O'REILLY, 2005). His current research interests include computer networks, mobile networking, ubiquitous networks, ad hoc networks and sensor networks. He is a member of IEEE, IEEE Communications Society, IEEE Computer Society, ACM SIGMOBILE, IEICE and 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 the Mitsubishi Electric Corp. Since 1993, he has been a Professor of Shizuoka University, Japan. Currently, he is a Dean of the graduate school of Science and technology of Shizuoka University. 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.