A Dynamic Packet Recovery Mechanism for Realtime Service in Mobile Computing Environments

Kwangroh Park, Yeunjoo Oh, Kyungshik Lim, and Kyoung-Rok Cho

This paper analyzes the characteristics of packet losses in mobile computing environments based on the Gilbert model and then describes a mechanism that can recover the lost audio packets using redundant data. Using information periodically reported by a receiver, the sender dynamically adjusts the amount and offset values of redundant data with the constraint of minimizing the bandwidth consumption of wireless links. Since mobile computing environments can be often characterized by frequent and consecutive packet losses, loss recovery mechanisms need to deal efficiently with both random and consecutive packet losses. To achieve this, the suggested mechanism uses relatively large, discontinuous exponential offset values. That gives the same effect as using both the sequential and interleaving redundant information. To verify the effectiveness of the mechanism, we extended and implemented RTP/RTCP and applications. The experimental results show that our mechanism, with an exponential offset, achieves a remarkably low complete packet loss rate and adapts dynamically to the fluctuation of the packet loss pattern in mobile computing environments.

Keywords: RTP/RTCP, Gilbert model, adaptive packet recovery, exponential offset, redundant data.

Manuscript received July 24, 2002; revised July 2, 2003.

Kwangroh Park (phone: +82 42 860 5290, email: krpark@etri.re.kr) is with Computer&Software Research Laboratory, ETRI, Daejeon, Korea.

Yeunjoo Oh (email: yjoh@etri.re.kr) is with Network Technology Laboratory, ETRI, Daejeon, Korea.

Kyungshik Lim (email: kslim@knu.ac.kr) is with Department of Computer Science, Kyungpook National University, Daegu, Korea.

Kyoung-Rok Cho (email: krcho@chungbuk.ac.kr) is with Department of Computer and Communication Engineering, Chungbuk National University, Cheongju, Korea.

I. Introduction

In these days, there is an increasing demand for efficient realtime multimedia service in both the wired and wireless Internet [1]-[6]. Traditionally, realtime transport protocol/realtime transport control protocol (RTP/RTCP) [6] has been widely used as a transportation mechanism of the realtime multimedia data on the wired Internet. By assigning a sequence number and timestamp to each packet, RTP/RTCP allows the receiver to efficiently deal with packet loss, out-of-order arrival, and jitter. However, we cannot avoid the loss of realtime data due to network congestion and errors. A number of packet loss recovery schemes have been widely studied in the wired Internet [1], [4], [7], [8]. Notice that in RTP/RTCP, the receiver only needs to replay the transmitted data with the same time interval used in the data generation process [7].

The use of RTP/RTCP in the wireless mobile Internet makes the packet loss recovery process more difficult and complex to solve. Wireless links have a high packet loss rate and limited bandwidth, and even the allocated bandwidth in a cell varies over time. Many mobile customers often experience frequent disconnections caused by physical barriers or handoffs. These are also one of the major causes of packet losses, especially in the realtime data transmission. The mobile computing environment also has varying traffic characteristics according to the terminal mobility or changes in the wireless links. Because of these characteristics, the mobile computing environment has a higher packet loss rate and an even higher consecutive packet loss rate compared to wired environments. Therefore, an effective packet loss recovery scheme with adaptive behavior to rapid changes in packet traffic and a relatively high packet loss rate is essential, especially for realtime data transmission in mobile computing environments.

There are many packet recovery mechanisms, such as automatic repeat request (ARQ), forward error correction (FEC), error-concealment techniques, and interleaving methods [4], [5], [9], [10].

ARQ is a closed-loop mechanism where the sender retransmits the lost packets. Although it can recover the packet losses completely, it is not suitable for realtime data because of the retransmission latency [4], [5], [10]-[13]. FEC-based mechanisms send redundant information, along with the original data, so that lost data can be recovered, at least partially, from this redundant information. The FEC-based mechanisms are divided into media-independent FEC and media-specific methods [10]-[13]. Originally, there was much interest in the provision of media-independent FEC using block codes (e.g., based on Reed-Solomon [14] or on parity codes [15], [16]) to provide redundant information. Unfortunately, these techniques have the disadvantage of introducing additional delays, since a source must wait for the entire block of packets before computing and transmitting the redundancy packet. These schemes can be used for one-way, near realtime audio transmission, but are not suitable for interactive audio communications.

A way of avoiding this delay is to use error-concealment algorithms at the receiver to correct the effect of the missing packets [9] instead of sending redundant information. Simple error-concealment techniques replace the missing audio unit with silence, white noise, or a repeated segment. Thus, these techniques work well for relatively small loss rates (below 10%) and for small packets (4-40 ms of audio) but they break down when the loss length approaches the length of a phoneme (5-100 ms). More sophisticated concealment techniques, such as adaptive packetization and concealment (AP/C) [17], exploit together the network loss characteristics and the property of long-term correlation within a speech signal to mitigate the impact of packet losses. This is accomplished by an adaptive choice of the packetization interval of the voice stream at the sender. Modern speech codecs, such as G.729 [18], use concealment algorithms that are defined as part of their specification. These algorithms interpolate the missing codec parameters based on surrounding and previous values. These techniques lead to a good quality of the reconstructed speech if the number of consecutive lost frames is small and if the loss does not occur at an unvoiced/voiced transition [19]. Hence, error-concealment schemes should be regarded not as substitutes for FEC, but rather a combination of them.

In interleaving methods, the packets are first rearranged into a matrix form and then transmitted in a column-major order [4], [10]. Notice that pure interleaving methods cannot recover any packet losses, although they reduce the consecutive packet loss ratio. Hence, some studies have considered attaching redundant data to interleaved data packets for recovering lost packets. Though this approach may improve the packet loss recovery ratio, it cannot avoid the overall time delay. It would also make little sense to add much redundant information when the loss rate is very low [10], [11], [13].

These approaches have been demonstrated to work well in fixed networks [9], [20]-[23]. However, in mobile computing environments, they cannot always offer any guarantee due to mobility [24], [25], changing conditions, and unpredictable radio link consumption. Therefore, we propose an adaptive packet recovery mechanism that is based on the packet loss characteristics periodically obtained from the destination. Here, we selected the audio stream as a specific example of applications. Additionally, we restricted the latency time to 160 ms since we considered realtime interactive services. It corresponds to at most 8 instances of redundant data for each packet when the packet is generated for every 20 ms. However, more redundant packets introduce more overheads for the packet size and bandwidth. The maximum number of redundant packets was selected considering all these factors and some previous results, such as those in [4] and [11]. We used the RTCP protocol as the feedback scheme. In fact, the feedback scheme using the receiver report RTCP (RR) packet is already used for similar purposes [11], [26]-[29]. However, the RTCP RR packet does not support the appropriate fields for application-specific information. Therefore, we newly defined and used the format of an extended application-defined RTCP (APP) packet.

In this paper, we first analyze the characteristics of packet losses in wired and wireless communications based on the Gilbert model. Using the current packet loss characteristic parameters reported by the receiver periodically, the sender dynamically adjusts the amount and the way of adding redundant information to minimize the required bandwidth for wireless links and to recover frequent and consecutive losses in mobile computing environments. Note that in order to deal with both random and consecutive packet losses efficiently, our scheme uses relatively large, discontinuous exponential offset values. That achieves a remarkably low packet loss rate after the recovery process at the receiver and less bandwidth consumption even in frequent burst errors of mobile computing environments.

In section II, the packet loss process is analyzed with the Gilbert model from the probabilistic point of view. Based on this analysis, we present an adaptive packet recovery scheme and its detailed algorithms, which can be used for various packet loss characteristics and a relatively high packet loss rate. An implementation and its experimental analysis are shown in section III. Finally, we present conclusions and future work in section IV.

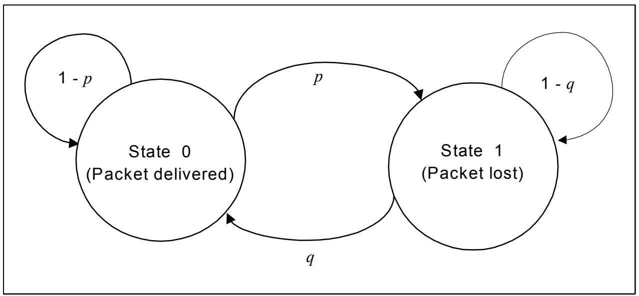

Fig. 1. The Gilbert model.

II. Design of an Adaptive Packet Loss Recovery Algorithm

1. The Effect of Exponential Offset Values

Recently, a lot of effort has gone into modeling data loss in packet networks based on the probability theory. Among the results, the Gilbert model has experimentally proved itself as having the capability of well approximating both random and burst errors [8], [11], [12], [30], [31]. We take advantage of the Gilbert model to analyze and recover from packet losses in mobile computing environments.

The Gilbert model explains packet loss using a 2-state Markov chain (Fig. 1). A packet arriving at the destination without any disruption is state-0, and the packet loss is state-1. In this case, p and q are used to show the characteristics of packet loss. In other words, when the *n*-th packet has arrived, the loss of the next $(n+1)$-th packet represents a transition from state-0 to state-1. The probability of this situation is represented by p. Therefore, the probability of a pack loss occurring for $(n+1)$ is $(1-p)$. Similarly, when the *n*-th packet is lost, the probabilities of $(n+1)$-th packet to arrive and to be lost are represented by q and $(1-q)$, respectively.

Let X_n denote a random variable that is set to 1 if packet *n* is lost, and 0 otherwise. Using X_n, the transition probabilities p and q are expressed in the following way.

$$p = \Pr[X_{n+1} = 1 \mid X_n = 0], \tag{1}$$

$$q = \Pr[X_{n+1} = 0 \mid X_n = 1]. \tag{2}$$

The probability that consecutive packets will preserve the same state expresses geometric distribution [30], [32]. Thus, according to the geometric distribution, the probability of a packet being in state-0 is calculated as follows:

$$\Pr[X_n = 0] = \frac{q}{p+q}, \tag{3}$$

$$\Pr[X_n = 1] = \frac{p}{p+q}. \tag{4}$$

The Gilbert model includes an existing procedure for computing the packet loss rate. For example, in the RTP/RTCP protocol, the fraction lost value is used to represent the ratio of lost packets out of a number of transmitted packets. This value expresses the probability of a packet being in state-1, and thus equal to (4). Using transit probability, the Gilbert model also has the ability of the computing consecutive loss rate as well as the occasional loss rate. In other words, the probability that *n*-th, (*n*+1)-th, and (*n*+2)-th packets are lost consecutively is calculated as follows:

$$\begin{aligned}
&\Pr[X_n = 1, X_{n+1} = 1, X_{n+2} = 1] \\
&= \Pr[X_{n+2} = 1 \mid X_{n+1} = 1] \cdot \Pr[X_{n+1} = 1 \mid X_n = 1] \cdot \Pr[X_n = 1] \\
&= (1-q) \cdot (1-q) \cdot \frac{p}{p+q} \\
&= (1-q)^2 \frac{p}{p+q}.
\end{aligned} \tag{5}$$

In this paper, we calculate the packet loss rate based on the Gilbert model to consider both occasional and consecutive packet losses. Note that when $p+q=1$, the model turns into the Bernoulli model (or a 1-state Markov chain model). In the Bernoulli model, arrivals or losses of each packet are treated as independent events, so the probabilities of packet loss and arrival occurring are represented by p and $(1-p)$, respectively. Using X_i, it can be represented as follows:

$$p = \Pr[X_i = 1] \tag{6}$$

$$(1-p) = \Pr[X_i = 0]. \tag{7}$$

Until now, many studies on packet recovery schemes have considered small and successive offset values: –1, –2, and –3. The notation *–i* means that redundant information of the (*n-i*)-th packet is sent in the primary packet, *n*. The use of these successive offset values can achieve a high recovery rate in occasional random errors. However, in the case of burst errors, the recovery rate will be very low.

In our scheme, the receiver analyzes received packets based on the Gilbert model and reports the analyzed packet loss and delivery probabilities *p* and *q* to the sender periodically. The sender then makes use of the probabilities to determine both the minimum amount and the way of adding redundant information to the primary packet to achieve the highest recovery rate and the intended service quality.

The more redundant information is added at the source, the more lost packets can be reconstructed. However, sending more redundant copies implies increasing the bandwidth requirement at the source (and henceforth the packet loss rate) and increasing the end-to-end delay (since the receiver has to wait longer for the redundant information). Therefore, a robust FEC scheme can be adaptive and choose the FEC mechanism according to the network characteristics (such as packet loss process, available bandwidth, etc.) at any given time and depending upon its impact on the end-to-end delay.

In mobile computing environments, both the total packet loss rate and the successive packet loss rate are relatively high. Although it seems natural to use more redundant data for relatively high loss rate conditions, we should keep the amount of redundant data at a minimum, considering the limited bandwidth of the wireless network. We try to minimize the overheads according to the sending of redundant data. This means that the amount of redundant data used at any given time should also depend on the characteristics of the loss process at that time.

The offset values –1 and –2 can be used to recover occasional random packet losses, while the offset values –4 and –8 can be used to recover consecutive packet losses. Thus, in the case of burst errors, we try to increase the offset value of the redundant information to get the same effect of the interleaving scheme.

For example, the offset values of –1, –2, –3 and –4 may fail when seven successive packets are lost (Fig. 2). In contrast, the offset values of –1, –2, –4 and –8 can recover much more data (Fig. 3).

Dealing with successive packet losses, relatively big offset values are more profitable. However, we cannot use an offset value that is too large since we need to preserve the realtime characteristics. For example, it is common to send packets at every 30 ms for audio data sampled at 8000 Hz. For an offset value of –8, it can make a maximum latency of 240 ms. Since realtime audio data shows very low quality with latency higher than 150-250 ms [9], it is difficult to use any higher offset values. That is why we adopt 4 as the maximum number of redundant data packets.

2. Derivation of Complete Packet Loss Boundary Graphs

As presented above, when we use a redundant data scheme, the rate of nonrecovered data, even though we used redundant data, is more important than the total packet loss rate. In this paper, we introduce the "complete packet loss rate," which represents the final loss rate after the recovery process with redundant data. Our goal is to find a sequence of offset values that increase exponentially so that the complete packet loss rate meets a certain user requirement and the bandwidth consumption by redundant information is as low as possible.

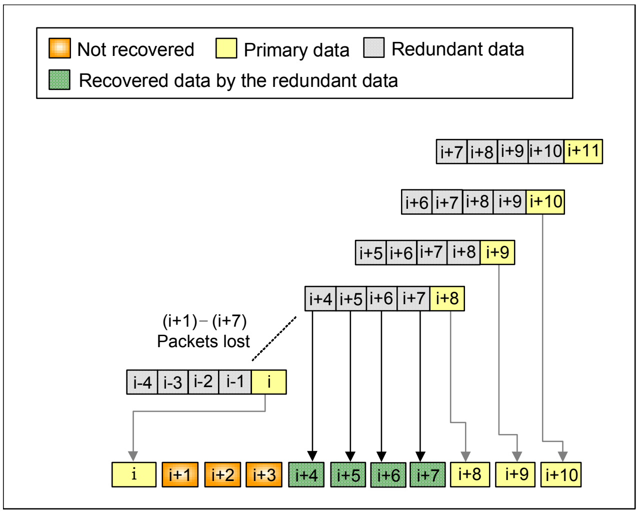

Fig. 2. The recovery scheme using redundant data with four successive offset values.

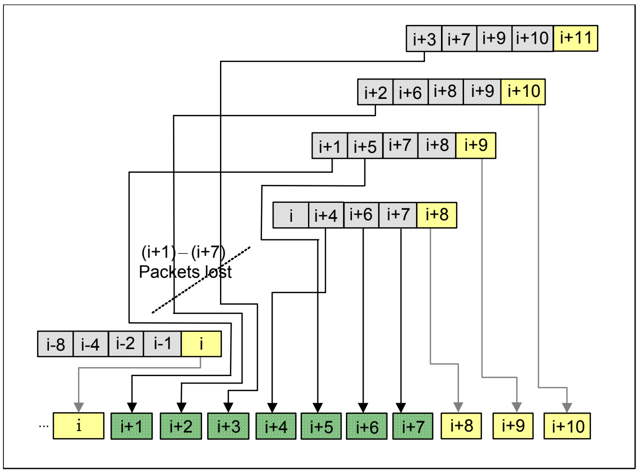

Fig. 3. The recovery scheme using redundant data with exponential offset values.

As pointed out earlier, we use the redundant data with offset values of –1, –2, –4, and –8. Including the case of no redundant data, we use a total of five redundant data methods and named them the R0-method, R1-method, R2-method, R3-method, and R4-method. The R0-method corresponds to the case of no redundant data. The R1-, R2-, R3-, and R4-methods use the offset values of (–1), (–1, –2), (–1, –2, –4) and (–1, –2, –4, –8), respectively. For example, for the n-th primary packet to be sent, the R3-method will generate three instances of redundant data for the (n–1)th, (n–2)th, and (n–4)th packets.

Since the R3-method and R4-method use relatively big offset values, they can achieve the effect of lowering the successive packet loss rate, which is similar to the interleaving methods. Since the characteristics of successive packet loss are closely related to the total packet loss rate, we should include the offset values of –1 and –2, which contribute to the recovery of isolated packet losses. For these reasons, we chose the combinations of (–1, –2, –4, –8) offset values.

Let $LossR0$, $LossR1$, $LossR2$, $LossR3$, and $LossR4$ be the complete packet loss rates of the R0-, R1-, R2-, R3- and R4-methods, respectively. Then, from the packet loss and delivery probabilities, p and q, respectively, reported from the receiver, the sender would predict the complete packet loss rate for each method as follows:

$$LossR0 = \Pr[X_n = 1] = \frac{p}{(p+q)}, \tag{8}$$

$$\begin{aligned} &LossR1 \\ &= \Pr[X_n = 1, X_{n+1} = 1] \\ &= \Pr[X_{n+1} = 1 \mid X_n = 1] \cdot \Pr[X_n = 1] \\ &= (1-q)\frac{p}{(p+q)}, \end{aligned} \tag{9}$$

$$\begin{aligned} &LossR2 \\ &= \Pr[X_n = 1, X_{n+1} = 1, X_{n+2} = 1] \\ &= \Pr[X_{n+2} = 1 \mid X_{n+1} = 1] \\ &\quad \times \Pr[X_{n+1} = 1 \mid X_n = 1] \cdot \Pr[X_n = 1] \\ &= (1-q)(1-q)\frac{p}{(p+q)} \\ &= (1-q)^2 \frac{p}{(p+q)}, \end{aligned} \tag{10}$$

$$\begin{aligned} &LossR3 \\ &= \Pr[X_n = 1, X_{n+1} = 1, X_{n+2} = 1, X_{n+4} = 1] \\ &= \Pr[X_{n+4} = 1 \mid X_{n+2} = 1] \cdot \Pr[X_{n+2} = 1 \mid X_{n+1} = 1] \\ &\quad \cdot \Pr[X_{n+1} = 1 \mid X_n = 1] \cdot \Pr[X_n = 1] \\ &= \{(1-q)^2 + pq\} \cdot (1-q) \cdot (1-q) \cdot \frac{p}{(p+q)} \\ &= \{(1-q)^2 + pq\} \cdot (1-q)^2 \cdot \frac{p}{(p+q)}, \end{aligned} \tag{11}$$

$$\begin{aligned} &LossR4 \\ &= \Pr[X_n = 1, X_{n+1} = 1, X_{n+2} = 1, X_{n+4} = 1, X_{n+8} = 1] \\ &= \Pr[X_{n+8} = 1 \mid X_{n+4} = 1] \cdot \Pr[X_{n+4} = 1 \mid X_{n+2} = 1] \\ &\quad \cdot \Pr[X_{n+2} = 1 \mid X_{n+1} = 1] \\ &\quad \cdot \Pr[X_{n+1} = 1 \mid X_n = 1] \cdot \Pr[X_n = 1] \\ &= \{(1-q)^4 + pq(6 - 4p + p^2 - 8q + 3pq + 3q^2)\} \\ &\quad \cdot \{(1-q)^2 + pq\}(1-q)^2 \cdot \frac{p}{(p+q)}. \end{aligned} \tag{12}$$

The complete packet loss rate monotonically decreases with respect to the increasing number of redundant data. Hence, for the transition probabilities of p and q, where their values are between 0 and 1, the following inequality conditions are always satisfied.

$$LossR0 \geq LossR1 \geq LossR2 \geq LossR3 \geq LossR4. \tag{13}$$

This monotonic decreasing property makes it easy to calculate the complete packet loss rate and select an appropriate one for the methods. Figure 4 represents the complete loss rates of each method using (8) through (12), where the absolute loss rate is increased as the p value becomes higher and abruptly decreased as the q value becomes lower.

Now we introduce an arbitrary threshold value to the complete packet loss rate. In environments with comparatively high loss rate and low bandwidth, such as mobile computing, it might be more reasonable to allow a little loss than to try to reliably receive all of the transmitted data. For example, in realtime multimedia applications, a little loss of data might not affect the quality of service seriously [10] and retransmission for full recovery of lost packets might decrease the quality of realtime service due to longer end-to-end latency.

In this case, the attractive alternative is to use the threshold of the loss rate that can maintain the tolerated quality in realtime

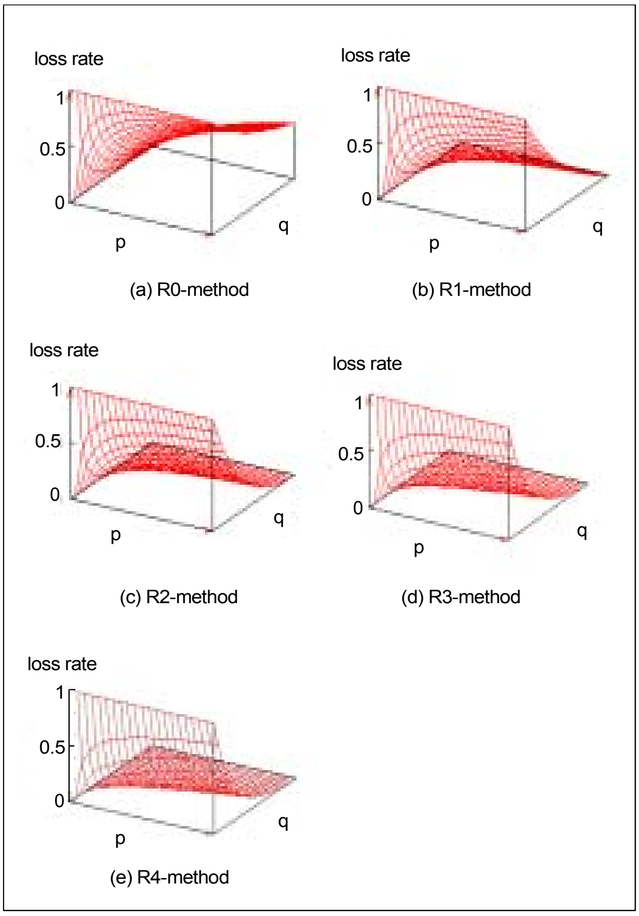

Fig. 4. The complete packet loss rates for each method.

applications. Thus, by allowing a loss rate that is lower or the same as the threshold, we can achieve the intended quality and reduce the overhead of the bandwidth.

Figure 5, obtained from Fig. 4, shows the boundary graphs of the redundant methods according to the threshold α, where α is 5%, 10%, 30%, and 40%, respectively. Using Fig. 5, we can obtain the domains of the p and q values such that the complete packet loss rate is lower than the threshold value of the loss rate. This domain can be expressed in a two-dimensional graph where each area is shown in Fig. 4. Each redundant method has a lower complete loss rate than the threshold loss rate where all p and q values exist on the upper left area. Therefore, given p' and q' values, the threshold of the loss rate is satisfied when we use a redundant method whose boundary line is located to the lower right area relative to the (p', q') position in the graph.

In Fig. 5, we can see that when the threshold loss rate increases, the boundary of each redundant method moves into the lower right corner of the graph. From the viewpoint of a specific redundant method, the maximum value of p becomes larger and the minimum value of q becomes smaller as α increases. The decreasing value of minimum q represents the consecutive loss rate being higher because $(1-q)$ reflects the characteristics of the consecutive packet losses.

On the other hand, considering a specific (p, q), the number of usable methods increases as α increases. For example, given $p = 0.2$, $q = 0.6$, and α= 5%, R2-method, R3-method, and R4-method are only satisfied with a complete loss rate that is less than the threshold loss rate value. However, when α is 30%, R0-method and R1-method are also satisfied.

3. Implementation through Extension of RTP/RTCP

The receiver plays realtime multimedia data after recovering the lost data with redundant data. After a regular time interval, it gathers the loss characteristics of transmitted packets and feedbacks information in the form of transit probabilities of the Gilbert model, p and q, to the sender. Using this information, the sender computes and predicts the loss rate of each redundant method (section II.2) and selects the appropriate

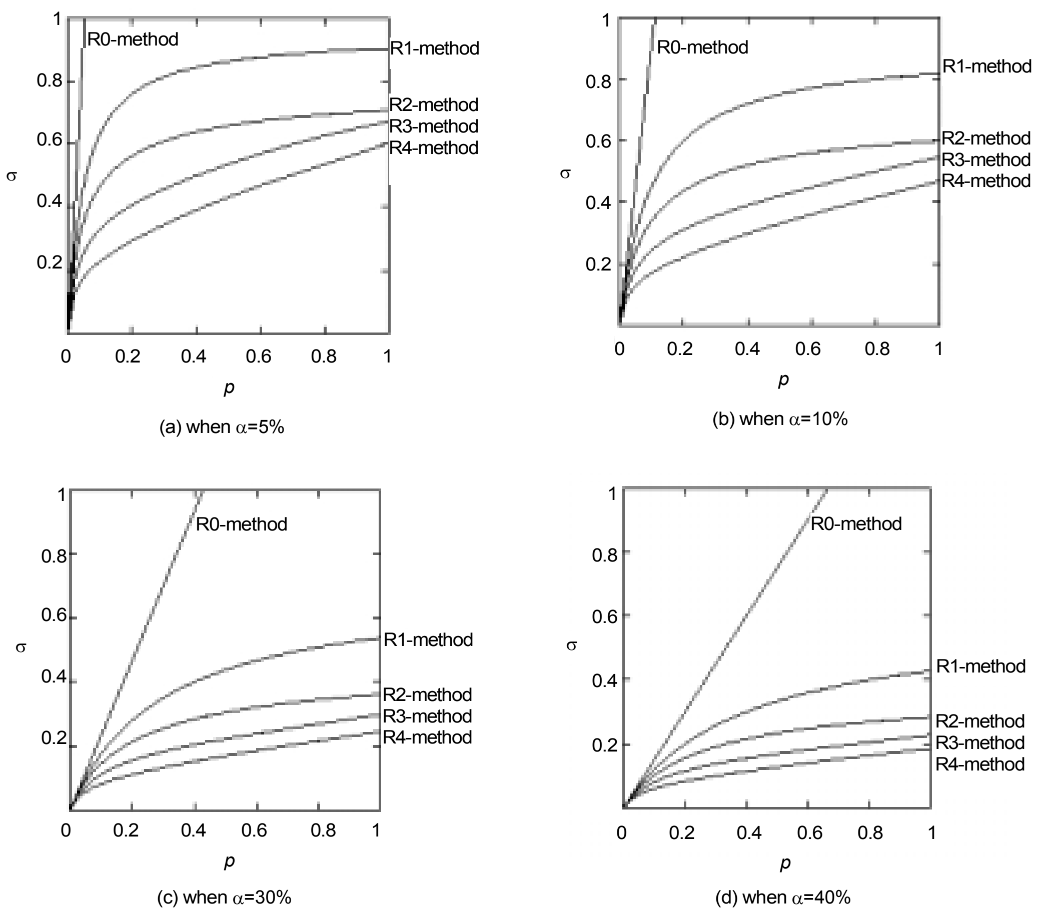

Fig. 5. The complete packet loss boundary graph according to the threshold α.

redundant method. Because the sender and receiver exchange the information to select an appropriate method periodically, this proposed scheme can dynamically adapt to changes of the characteristics of the packet loss patterns over networks.

A. Extension of RTP/RTCP Packets

Based on RFC1889 and RFC1890 specifications, we use the RTP data packet and RTCP APP packet to implement our scheme. As Fig. 6 shows, when X field is set to 1 in an RTP packet, it indicates that there is a header extension part, which includes redundant information, between the RTP header and the data part. The size of redundant data can be set in the length field of the header extension part. In section III, we experiment with the redundant data of the G.723.1 speech codec, whose bitrate is 6.3 Kbps, and the codec processes speech in 30 ms frames; thus, an RTP data packet has no redundant data or has redundant data of 24 to 96 bytes. The RTP header extension part contains the following information:

- *extension type*: type of redundant method (R1- 4)
- *redundant data length*: total size of redundant data block
- *redundant data*: redundant payload field

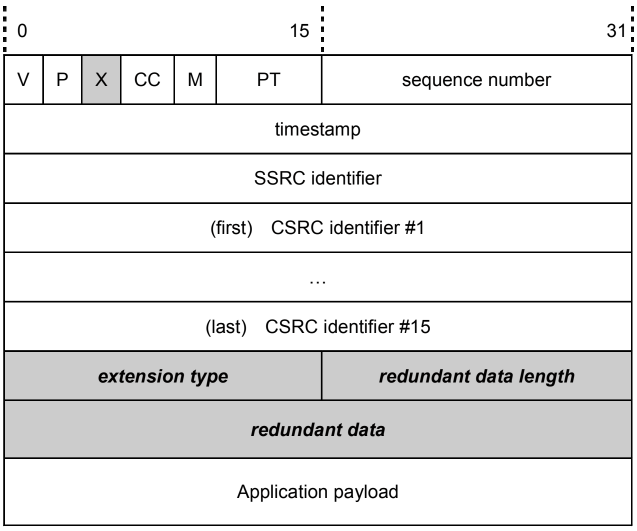

Fig. 6. An RTP packet including redundant data block.

Table 1 shows the overheads for each method used in our scheme. Clearly, adding redundant information increases the bandwidth requirements of the codec. However, the use of G.723.1 codec as the redundant information adds only a small amount of overhead to an RTP packet. For example, using one speech frame (30 ms) per packet, the use of G.711 pulse code modulation (PCM) codec as the primary and G.723.1 as the redundant data results in about a 10-40% increase in the size of a packet used in each method. For example, in case of R4-method, an RTP packet has 352 bytes including the RTP header length (12 bytes), the extension type field (2 bytes), and the redundant data length field (2 bytes).

Table 1. Overhead for each method using redundant data.

Used methods	The amount of redundant (bytes per frame)	An RTP packet size including extension type and redundant length (bytes)	Increasing ratio for RTP packet
No redundant	0	252 (12+240)	1
R1-method	24	280 (12+240+4+24)	1.11
R2-method	48	304 (12+240+4+48)	1.20
R3-method	72	328 (12+240+4+72)	1.30
R4-method	96	352 (12+240+4+96)	1.40

In our scheme, the receiver sends the feedback information to the sender via an RTCP APP packet to deliver the transit probabilities p and q, since the current RTCP RR packet does not support the appropriate fields for the application-specific information. Figure 7 shows the format of an extended RTCP APP packet. The RTCP APP header contains the following information:

- *name*: indicates the probability information of receiving status (= "PVAL")
- *p_value*: a value of transit probability p
- *q_value*: a value of transit probability q

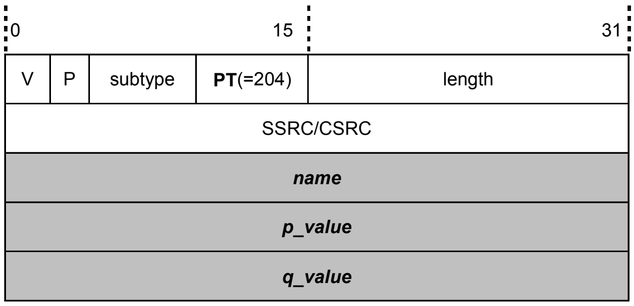

Fig. 7. The format of extended RTCP APP packet.

B. Receiver Action Algorithm

The main role of the receiver in this scheme is to keep track of the transit probabilities, p and q, in order to periodically report to the sender. Although the transit probabilities are the theoretical values in the Markov chain process, we can calculate the probabilities by tracking the state transition of the received packets.

As Fig. 8 shows, we can analyze the number of the state transition by the sequence number of the received packets. We denote the number of state transitions corresponding to the transit probabilities of p, q, $(1-p)$, and $(1-q)$ as P, Q, P', and Q', respectively.

Table 2 shows the calculated transit probabilities p and q for the case shown in Fig. 8.

$$p = \frac{P}{(P+P')} = \frac{4}{13} \cong 0.307. \tag{14}$$

$$q = \frac{Q}{(Q+Q')} = \frac{4}{6} \cong 0.667. \tag{15}$$

Thus, the fraction lost value in an RTCP RR packet is calculated as follows:

$$\Pr[X_n = 1] = \frac{p}{p+q} = \frac{12}{38} \cong 0.316. \tag{16}$$

Whenever the receiver receives an RTP data packet, it updates the number of cases corresponding to the probabilities, p and q. It also calculates the probabilities p and q when it needs to transmit an RTCP APP packet to the sender. Therefore, the default time interval is set to around 5 seconds, which is recommended by the RTP/RTCP protocol. For voice data of 8000 Hz, it indicates a transmission of about 165 packets during the period.

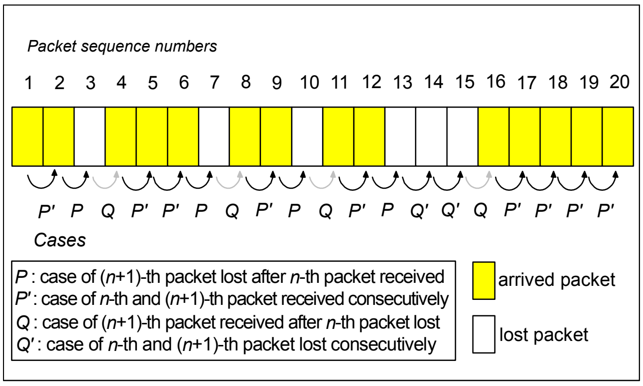

Fig. 8. The analysis of the state transitions over received packets.

Table 2. The number of state transitions for each case from Fig. 8.

The transit of states	P (S0→S1)	P' (S0→S0)	Q (S1→S0)	Q' (S1→S1)
The number of states	4	9	4	2

C. Sender Action Algorithm

Whenever a sender receives an RTCP APP packet, it calculates the complete loss rates of each redundant method using the p and q values in the packet. The complete loss rate of a method is defined as an expected value of the future loss rate that is supposed to be obtained when using the corresponding method. For example, when the expected value of the loss rate is 0.06, the probability that a specific packet cannot be recovered at all, even if redundant data in other packets is used, is 6%.

The sender selects a redundant method that has a complete loss rate less than the threshold. The threshold α represents the maximum value of allowable loss rate that can be determined by users. When two or more methods satisfy the threshold value, the sender selects a method that has the minimum number of redundant data among them. For example, when both the R2-method and the R3-method are satisfied with the threshold value, the receiver selects the R2-method. Therefore, it maintains a complete loss rate that is less than the threshold, and it also minimizes the overhead of bandwidth requirements by reducing the size of packets compared to the R3-method. The selected method is used until the sender receives the next RTCP APP packet.

III. Experimental Analysis

1. Experimentation

To evaluate the scheme proposed in section II, we implemented the adaptive realtime audio transmission application based on the RTP protocol modules, rtptools [33] and NeVoT [34]. The two end systems, Pentium III 450 MHz notebooks with 128 MB memory running Linux, were interconnected via an Internet simulator over a 10 Mbps Ethernet. The Internet simulator was used to simulate the characteristics of packet losses in mobile computing environments and generate random and burst losses based on the transition probabilities, p and q, of the Gilbert model. For one session between a sender and receiver, we considered a bandwidth of about 1 Mbps.

For experimental data, we used PCM (64Kbps) data for the primary data whose sampling rate was 8000 Hz. For redundant data, we used G.723.1 codec to reduce the bandwidth requirement for transmission. The redundant data were included in the RTP header extension field. The information for packet losses was included in the RTCP APP packet, and the receiver combined the RTCP APP packet and RTCP RR packet, and then sent the compound packet to the sender.

To reflect the characteristics of packet losses and delay in mobile computing environments, we determined the domain of p and q, and experimented with about thirty (p, q) sampling values within the domain. Figure 9 shows the packet loss rates for p, when q was given in the range of 0.1 to 0.9 for our experimentation. Since mean loss rates and consecutive loss rates are higher on mobile networks than on wired networks, we could determine the maximum and minimum values of p and q by assuming the packet loss rates to be approximately from 20% to 50% and the number of consecutive lost packets to be approximately from 2 to 7 and by referring to Fig. 9.

The value of q was derived using the mean residence time (MRT) of the Gilbert model. The MRT represents the average residence time in state-1, which represents the packet loss. Thus, the MRT derived by a stochastic process was $1/q$ [11], [12]. The average number of consecutively lost packets ranged from 2 to 7. Therefore, the value of $1/q$ varied from 2 to 7. Consequently, the value of q will be between 0.143 and 0.500. Due to the fact that the speech quality is not generally satisfying with packet loss rates any higher than 10% [2], [9], we set the threshold value of the target loss rate to 5% in this experiment.

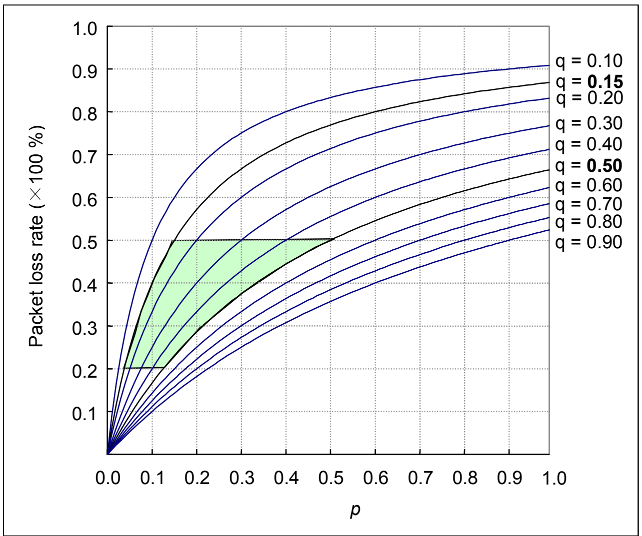

Fig. 9. The determination of experimental domain, given $0.2 \le$ loss rate ≤ 0.5.

2. Measurement and Analysis

We carried out our experiment over a LAN environment. We observed and compared the complete loss rates of each redundant method when the mean packet loss rate was 25.5%, and $(p, q) = (0.12, 0.35)$. We set the measuring period to 5 seconds, which is equivalent to the period of the feedback information in RTCP.

Figure 10 shows the evolution of the loss rate measured over time for the connection between the sender and receiver, while each redundant method was statically used under the same experimental network condition with the target loss rate, i.e., the threshold value of a tolerated loss rate at 5%. Table 3 shows the comparison of the result values measured in Fig. 10 with the results of the theoretical probability as computed in section II.2. The experimental results are similar to those of the theoretical.

In the theoretical results, the R0-method had a mean loss rate of 24.0%. The R1-method, R2-method, R3-method, and R4-method had mean loss rates of 15.4%, 9.8%, 4.4%, and 1.2%, respectively.

We observed that the R3-method and R4-method had lower loss rates than the desired threshold value α (= 5%). Therefore, the adaptive redundant transmission scheme described in section II.2 used the R3-method and R4-method in order to maintain the loss rate close to the threshold value of α, and selected the R2-method when the loss rate was temporarily low to minimize the bandwidth requirements.

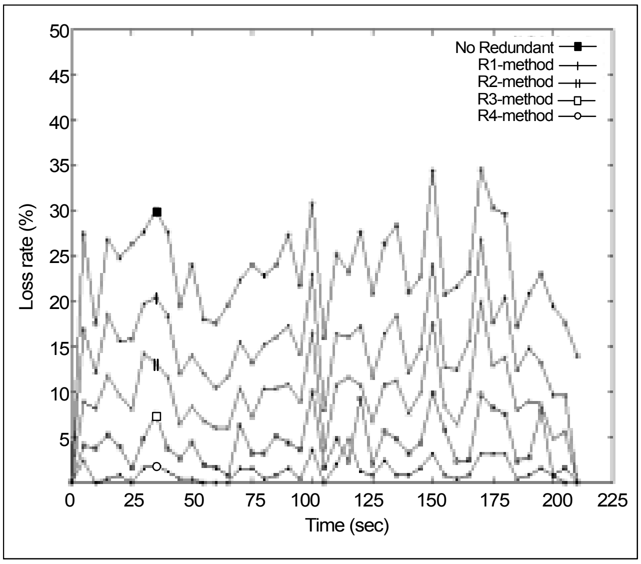

Fig. 10. The complete packet loss rates for static redundant information, where $(p, q) = (0.12, 0.35)$ and α= 5%.

Table 3. The complete loss rates, where $(p, q) = (0.12, 0.35)$.

Methods (being used statically)	Measured results (%)	Theoretical results (%)
R0-method (no redun.)	24.0	25.5
R1-method	15.4	16.6
R2-method	9.8	10.8
R3-method	4.4	5.0
R4-method	1.2	1.6

Figure 11 shows the complete loss rates from two cases where the nonredundant method and adaptive redundant transmission mechanism were used over the same connection. As we can see in Fig. 11, the complete loss rates fluctuated around the threshold value α when the adaptive redundant transmission mechanism was used, even though the loss rate in the experimental network varied between 15% and 40%.

We observed that the R2-method, R3-method and, R4-method utilized 9.5%, 57.2%, and 33.3% respectively, in the adaptive redundant transmission scheme. To analyze these results, recall the complete packet loss boundary graph in Fig. 5. When the threshold value α was 5%, it corresponded to Fig. 5 (a), and for convenience, the same boundary graph is shown in Fig. 12.

Point (1) in Fig. 12 indicates that the p value is equal to 0.12 and the q value is 0.35. As the point is located around the

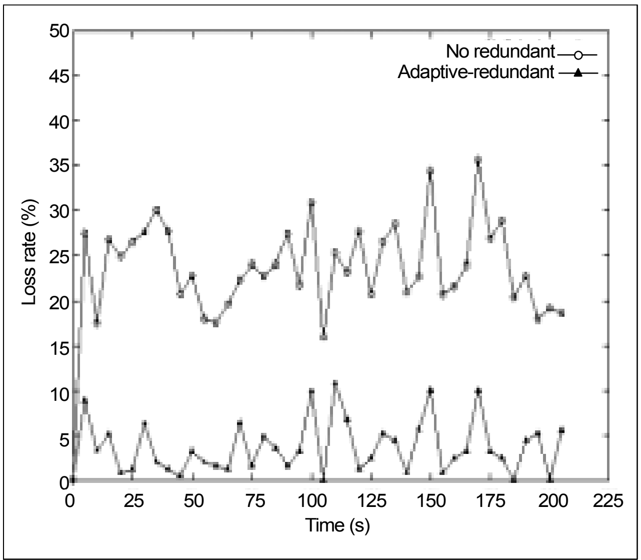

Fig. 11. The measured loss rate of the nonredundant and of the adaptive redundant mechanism, where $(p, q) = (0.12, 0.35)$, α=5%.

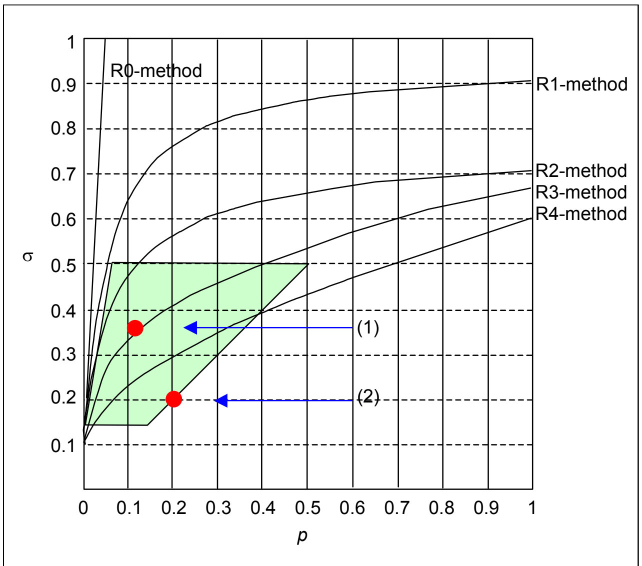

Fig. 12. The boundary of domain of each redundant method, where α=5%.

Table 4. The comparison for the values of (p, q), where the mean packet loss rate is 33%.

(p,q) Redundant Methods	(0.1, 0.2)	(0.15,0.3)	(0.2, 0.4)	(0.3, 0.6)
Non-Redundant	31.45 %	32.03 %	32.31 %	33.29 %
R1-method	25.23 %	22.51 %	19.27 %	13.28 %
R2-method	20.16 %	15.60 %	11.29 %	5.17 %
R3-method	13.60 %	8.41 %	4.75 %	1.85 %
R4-method	6.78 %	3.21 %	1.68 %	0.62 %
Adaptive -method	7.33 %	4.16 %	3.42 %	3.49 %

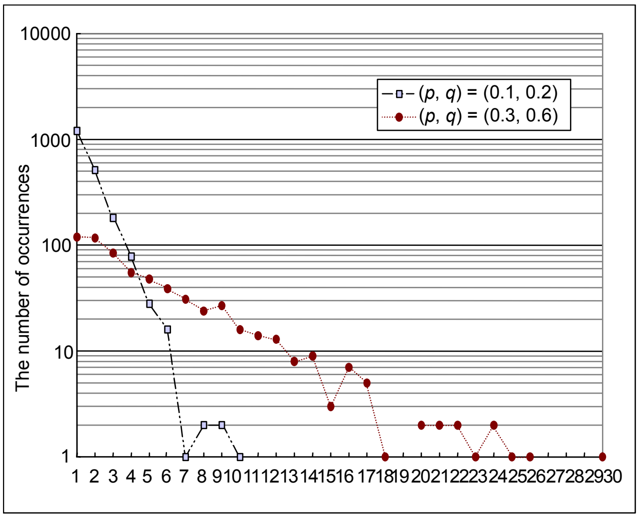

Fig. 13. Frequency distribution of the number of consecutively lost packets.

boundary of the R3-method, we used the R3-method or R4-method to remain close to the threshold value α. If the R1-method or R2-method was used, we could not satisfy the threshold. If we used only the R4-method, we would obtain lower loss rates than the threshold. Consequently, the bandwidth requirements would be increased due to more usage of redundant data. Because point (2), given that $(p, q) = (0.2, 0.2)$, is located below the boundary of the R4-method, none of the redundant methods satisfy the threshold value α.

Table 4 shows the complete loss rates for the value of (p, q) where the mean loss rate is 33%. We observed that the complete loss rates measured of each (p, q) differed from that of others. This result indicates that the packet loss rates as well as the characteristics of consecutive losses are important factors in the loss process.

Mainly, $(1–q)$ reflects the characteristics of consecutive losses, which also implies that the lower the q value, the higher the consecutive losses. Thus, when the q value is less than the limit of consecutive loss that each redundant method accepts, the complete loss rate is higher because the lost packets are not successfully recovered.

Figure 13 shows the distribution of the number of occurrences of n consecutive losses for $(p, q) = (0.1, 0.2)$ and $(p, q) = (0.3, 0.6)$. Both satisfy the geometric distribution similarly, however they differ in the slope of the distributions. The result indicates that although we have the same packet loss rate, the characteristics of the lost packets can be very different. This fact shows that the above analysis is important for the selection of an appropriate redundant method in the recovery process of lost packets.

IV. Conclusion and Future Work

In this paper, we proposed an adaptive packet loss recovery scheme that efficiently recovers occasional and consecutive packet losses in mobile computing environments. The proposed scheme analyzes, based on the Gilbert model, the characteristics of packet losses that may occur due to a low bandwidth, higher bit error rate, user mobility, etc. The sender adaptively adjusts both the amount and the way of adding redundant data based on the transition probabilities periodically reported by the receiver.

For consecutive packet losses, our scheme particularly selects the redundant data with the offset values, such as –4 and –8, to get the effect of an interleaving scheme. From the use of the exponential offset values in our scheme, we obtained a result that minimized the mean packet loss rate as well as the consecutiveness of packet losses. This adaptive scheme also reduces the bandwidth requirements by dynamically selecting the redundant method according to the current network conditions.

The proposed mechanism used the RTP protocol to transmit realtime data and redundant information and extended the RTCP APP packet to deliver feedback information about packet losses. To verify the effectiveness of our scheme, we implemented the adaptive realtime audio transmission application and then simulated and analyzed it with two end systems. These analyses demonstrated that the proposed scheme greatly improved the recovery rates of lost packets in mobile computing environments.

Traditionally, realtime multimedia services can be classified into streaming services, such as audio on demand or video on demand, and interactive services, such as internet telephony. In the case of streaming services, we were able to minimize the quality degradation due to packet loss or delay using prebuffering and/or large offset values. However, we also considered interactive services as the application area of our mechanism and thus we chose the maximum latency bound as 240 ms, which is the latency limit for interactive applications. Therefore, we have the maximum offset value of –8, which is the maximum value for 240 ms latency.

In this paper, we focused on the algorithm itself based on the unicast model between a sender and a receiver. We think that an extension to the multi-case situation and testing the scheme on real customers will be valuable for future work. For example, as a brute-force approach, we can select the worst-case values among the multiple feedback packets. However, we hope to perform various experiments on this multicast situation [6] to make it a meaningful result. Moreover, additional experiments should be conducted with a wider range of realtime video applications, such as internet multimedia conferencing, or video on demand.

References

[1] Kevin Brown and Suresh Singh, "Quality of Service Guarantees in Mobile Computing," *J. Computer Commun.*, vol. 19, 1996, pp. 359-371.

[2] Kevin Brown and Suresh Singh, "M-UDP: UDP for mobile cellular networks," *ACM SIGCOMM Computer Commun. Review*, vol. 26, no. 5, Oct. 1996, pp. 60-78.

[3] Kevin Brown and Suresh Singh, "Extensions to RTP to Support Mobile Networking," *3rd Int'l. Workshop on Mobile Multimedia Commun.*, Sept. 25-27, 1996.

[4] J. Park and D. Ko, "Implementation and Evaluation of Lost Packet Recovery Using Low Bit-Rate Redundant Audio Data," *IEEK J.*, vol. 35. no.7, 1998, pp. 1-5.

[5] M. Kang, S. Kung, and D. Kim, "Implementation and Performance Evaluation of the Dynamic Redundant Transmission Mechanism for Loss Recovery of Audio Packets on Teleconferencing System Using RTP/RTCP," *KIPS J.*, vol. 5, no. 10, Oct. 1998, pp. 2641-2653.

[6] Myung-Ki Shin, Yong-Jin Kim, Ki-Shik Park, and Sang-Ha Kim, "Explicit Multicast Extension (Xcast+) for Efficient Multicast Packet Delivery," *ETRI J.*, vol .23, no. 4, Dec. 2001, pp. 202.

[7] Internet Engineering Task Force, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, Jan. 1996.

[8] J. Bae, E. Lee, J. Lee, M. Shin, and J. Ham, "Implementation of a WWW System for Real-Time Multimedia Data Services," *KISS J. (C)*, vol.5, no.2, 1999, pp. 157-166.

[9] V. Hardman, A. Sasse, M. Handley, A. Handley, and A. Watson, "Reliable Audio for Use over the Internet," *Proc. INET'95*, Honolulu, HI, June 1995, pp. 171-178.

[10] J. Park and D. Ko, "Implementation of an audio broadcasting service over the Internet," *KIPS J.*, vol. 23, no. 6, 1998, pp. 1496-1502.

[11] J-C. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-Based Error Control for Internet Telephony," *Proc. Infocom'99*, New York, March 1999, pp. 1453-1460.

[12] Jean-Chrysostome Bolot, Hugues Crepin, and Andres Vega

Garcia, "Analysis of Audio Packet Loss in the Internet," *Proc. NOSSDAV 95* (*Network and Operating System Support for Digital Audio and Video*), Durham, NH, Apr. 1995, pp. 154-165.
[13] C. Perkins and O. Hodson, "A Survey of Packet Loss Recovery Techniques for Streaming Media," *IEEE Network Magazine*, vol. 12, no. 5, Sept.-Oct. 1998, pp. 40-48.
[14] R. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, 1993.
[15] N. Shacham and P. Mc Kenney, "Packet Recovery in High-Speed Networks Using Coding and Buffer Management," *Proc. IEEE Infocom'1990*, vol. 1, June 1990, pp. 124–131.
[16] D.R. Figuereido and E. de Souza e Silva, "Efficient Mechanisms for Recovering Voice Packets in the Internet," *Proc. IEEE Globecom'99*, 1999, pp. 1830–1836.
[17] H. Sanneck, "Concealment of Lost Speech Packets Using Adaptive Packetization," *Proc. IEEE Multimedia Systems 1998*, Austin, TX, June 1998, pp. 140-149.
[18] International Telecommunications Union, "Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (csacelp)," *Tech. Rep*., ITU-T Recommendation G.729, March 1996.
[19] H. Sanneck and N. Le, "Speech Property-Based FEC for Internet Telephony Applications," *Proc. of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference (MMCN)*, San Jose, CA, Jan. 2000, pp. 38–51.
[20] J-C. Bolot and A. Vega Garcia, "The Case for FEC-Based Error Control for Packet Audio in the Internet," *ACM Multimedia Systems*, 1997.
[21] M. Podolsky, C. Romer, and S. McCanne, "Simulation of Fec-Based Error Control for Packet Audio in the Internet," *IEEE Infocom'98*, Apr. 1998, pp. 505-515
[22] C. Perkins et al., "RTP Payload for Redundant Audio Data," IETF *RFC2198*, Sept. 1997.
[23] C. Padhye, K.J. Christensen, and W. Moreno, "A New Adaptive FEC Loss Control FEC Loss Control Algorithm for Voice over IP Applications," *IEEE IPCCC'00*, 2000, pp.307-313.
[24] Hewlett Packard Mobile IP, http://www.hpl.hp.com/personal /Jean_Tourrilhes/Mobile IP/
[25] C. Perkins, "IP Mobility Support," *IETF RFC 2002*, Oct. 1996.
[26] C. Boutremans and J.Y. Le Boudec, "Adaptive Delay Aware Error Control for Internet Telephony," *Proc. of 2nd IP Telephony workshop*, Columbia University, New York, Apr. 2001, pp. 81-92.
[27] Pedro M. Ruiz and Emilio García. "Improving User-Perceived QoS in Mobile and Wireless IP Networks Using Real-Time Adaptive Multimedia Applications". *Proc. of the 13th Int'l Symposium on Personal, Indoor and Mobile Radio Commun. (PIMRC'2002)*, Lisbon, vol. 3, Sept. 2002, pp.1467-1471.
[28] M.K. Ji, S.H. Lee, and T.U. Choi et al. "Selecting an Audio Redundancy Codec Combination for Error Control in Internet Telephony," *Proc. of IEEE ICPADS 2001*, Kyoung-Ju, Korea, June 2001, pp. 645-652.
[29] M.K. Ji, S.H. Lee, T.U. Choi, S.H. Park, and K.D. Chung, "An Adaptive FEC Error Control Algorithm Using Loss and Delay for Internet Telephony," *Proc. of IEEE ICME2001*, Tokyo, Japan, Aug. 2001.
[30] M. Miyoshi et el. "Improving TCP Performance for Wireless Cellular Networks by Adaptive FEC Combined with Explicit Loss Notification," *IEICE Trans.*, vol.E85-B, no.10, Oct. 2002, pp. 2208.
[31] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and Modeling of the Temporal Dependence in Packet Loss," *Proc. IEEE Infocom'99 NY*, Mar. 1999, pp. 345-352.
[32] H. Sanneck and N. Le, "Speech Property-Based FEC for Internet Telephony Applications," *Proc. of the SPIE/ACM SIGMM Multimedia Computing and Networking Conf. 2000 (MMCN 2000)*, San Jose, CA, Jan. 2000, pp. 38-51.
[33] rtptools, http://www.cs.columbia.edu/~hgs/rtptools/
[34] NeVot, http://www.cs.columbia.edu/~hgs/nevot

Kwangroh Park received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, Korea in 1982 and 1985 respectively, and his Ph.D. degree in the Computer and Communication Engineering from Chungbuk National University, Cheongju, Korea in 2002. He joined Electronics and Telecommunications Research Institute, in 1984 and has been engaged in the project of digital switching and CDMA mobile systems, etc. He also developed the residential gateway system as a project leader of home network team. Now, he is a project leader of control software research team of embedded software technology center. His research interests are home network middleware and home server technology, and home digital multimedia services.

Yeunjoo Oh received her B.S. in Computer Science from Geungsang National University, Korea, in 1998, and her M.S. degree Kyungpook National University in 2001. Since 2001, she has been working as a Researcher in the Network Technology Laboratory, Electronics and Telecommunication Research Institute, Daejeon, Korea. Her research interests include the transmission for realtime multimedia application, Mobile IP, Voice Over IP, Home Network, and End-Quality of Service.

Kyungshik Lim received his B.S degree in Electronics Engineering from the Kyungpook National University, Daegu, Korea, in 1982, his M.S. degree in Computer Science from the Korea Advanced Institute of Science and Technology, Seoul, Korea, in 1985 and his Ph.D. degree in Computer Science from the University of Florida, Gainesville, FL, in 1994. From 1985 to 1998, he had been a principal member of the research staff of the Electronics and Telecommunications Research Institute, Daejeon, Korea. Since March 1998, he has been a faculty member of the Computer Science Department in the Kyungpook

National University, Daegu, Korea. His research interests include mobile computing, wireless Internet, home networking, and network games.

Kyoung-Rok Cho received the B.S. degree in Electronic Engineering from Kyoungpook National University, Taegu, Korea in 1977, and M.S. and Ph.D. degree in Electrical Engineering from the University of Tokyo, Japan, in 1989 and 1992, respectively. From 1979 to 1986, he was with TV research center of Gold Star Company in Korea. He is currently a Professor in Department of Computer and Communication Engineering of Chungbuk National University, Korea, since August 1992. His research interests are in the field of the high-speed and low-power circuit design, SoC and Platform design for the communication system. From 1999 to 2000, he was a visiting scholar at Oregon State University, OR. He is a member of Institute of Electrical and Electronics Engineer (IEEE), and Korea Institute Telecommunication Electronics (KITE).