ECLIPSE: Error Correction based on Loss probability and data Importance for Plural Server Environment

Akihiro FUJIMOTO*, Yusuke HIROTA*, Hideki TODE† and Koso MURAKAMI*
*Department of Information Networking, Osaka University
Email: {fujimoto.akihiro, hirota.yusuke, murakami}@ist.osaka-u.ac.jp
†Department of Computer Science and Intelligent Systems, Osaka Prefecture University
Email: tode@cs.osakafu-u.ac.jp

***Abstract*—Video streaming technique is widely used in the Internet with the development of access networks. Although some techniques such as Forward Error Correction or Content Distribution Network are effective for video streaming services, they are not sufficient in the case that streams are suffered from heavy packet loss. For this problem, we propose a novel error correction method called "ECLIPSE: Error Correction based on Loss probability and data Importance for Plural Server Environment". The ECLIPSE recovers lost packets with a parity stream which exploits multiple streams. In order to improve video quality, ECLIPSE gives priority to each data of multiple streams in terms of recovery from packet losses. It is designed for recovering important data with higher probability. The priority of each data is set based on two criteria. The one is the inherent importance of the data within one stream. The other is the loss probability of one stream which is compared with the loss probabilities of other streams. We also clarify the performance of the proposed method through theoretical analysis. Moreover, we focus on the creation method of redundant data. They are transferred from another server for multiple worse conditioned streams, which greatly affects the recovery capability. We propose the adaptive creation method for various network conditions.**

Our simulation results show that the ECLIPSE can recover more important data with higher probability. Furthermore, it can also achieve almost the same restoration ratio among streams of recovery targets which have different loss probabilities.

***Index Terms*—Video streaming, Recovery priority, Parity stream, Forward Error Correction, Content Delivery Network**

I. Introduction

In recent years, video streaming services have become popular and major applications in high speed networks. However, video streaming over networks has several problems due to their characteristics such as high bit rate or delay restriction. Content Distribution Network (CDN) is effective for such video streaming services [1]. This system has multiple cache servers in various locations in the network. Therefore, clients can receive their requested contents from appropriate server in terms of transmission delay and/or server's load. Moreover, parallel video distribution methods exploiting these multiple servers have been proposed [1], [2]. In these methods, one content is separated or encoded into multiple pieces and distributed through multiple routes. Using path diversity, parallel video distribution can decrease the harmful effect from a path under bad condition.

Packet loss is also an important problem in video streaming services through best-effort network such as the Internet. A video content consists of many pieces of data which have various importance. The video content can be played back with not high quality but acceptable quality if data of relatively minor importance is lost. On the other hand, the quality degradation due to important data loss keeps during certain term. Therefore, recovery of lost data is necessary to alleviate the quality degradation. Various types of methods have been studied to recover the lost data, such as Automatic Repeat reQuest (ARQ) [3], Forward Error Correction (FEC) [4] and hybrid of the two approaches [5]. However, these recovery methods do not consider the difference of data importance.

Considering the importance of data, video quality can be further improved. Unequal Error Protection (UEP) method has been proposed to recover important data with higher probability [6], [7]. The basic idea of this method is to protect more important data with more redundant data. By recovering more important data instead of less important data, UEP method can improve the video quality. In references [6], [7], maximum distance separable codes are used to provide UEP property. These methods typically classify data by the importance and separately protect them. These methods can recover all lost data with equal importance when the amount of lost data within each class is less than corresponding redundant data. However, it is hard to dynamically adjust redundancy in these methods because of their complex calculation. On the other hand, the rateless code methods based on UEP are proposed in [8], [9], [10], [11], [12]. Rateless code generates arbitrary number of encoded symbols from source symbols (original data). Clients can reconstruct source symbols if they can receive encoded packets slightly more than the number of source symbols.

Manuscript received February 15, 2011; revised August 14, 2011; accepted November 8, 2011.

doi:10.4304/jcm.7.2.165-176

These UEP methods realize gradual quality degradation even if packet loss rate increases. However, they can not provide sufficient video quality when the stream is heavily damaged.

In order to manage heavily damaged video streams, we propose a data recovery method exploiting multiple video streams which are transferred from plural servers. To deal with multiple streams as a virtual stream can reduce the influence of heavy packet losses in specific streams. The proposed method efficiently improves video quality by estimating utility of redundant data. Considering video characteristics, this method is designed for recovering important data with higher probability. In addition, this method deals with the data of streams whose packet loss rates are high as important data for fairness of quality. In order to satisfy these requirements, the proposed method gives priority for recovery to each packet of multiple streams based on both inherent importance in each stream and loss probabilities of these streams. For these features, we call the proposed method "ECLIPSE: Error Correction based on Loss probability and data Importance for Plural Server Environment". We also analyze the performance of the proposed method. Moreover, we focus on the redundant data creation method, which corresponds to the core part of the recovery capability enhancement, and propose the adaptive creation method in order to reduce the influence from the network condition.

The rest of this paper is organized as follows. Section II introduces related works. Section III describes ECLIPSE in detail. Section IV investigates the performance of the proposed method through theoretical analysis, and improved method is described in Section V. Section VI evaluates the performance of the proposed method through computer simulations. Finally, Section VII concludes this paper.

II. Related Works

In order to improve video quality, some types of UEP method have been proposed. As one of UEP methods, Wu et al. have proposed the FEC scheme which offers unequal importance protection for H.264/SVC in [6]. H.264/SVC divides a content into some layers ($L_0, L_1 \cdots, L_n$), where $L_i (1 \leq i \leq n)$ is decoded using from L_0 to L_{i-1}. A receiver can play back in higher quality when higher layer is decoded. This FEC scheme assigns more FEC redundancy to lower layers so that more layers can be decoded. Unequal Growth Code is designed as the short block-length rateless code for video streaming [9]. This method protects a data packet with parity packets which are generated in rateless code based UEP manner. In this method, more important data is encoded into parity packets with higher probability. In [13], [14], randomized linear network coding methods with UEP property have been proposed. In these methods, nodes construct overlay networks. Each node encodes incoming packets into outgoing packets using linear equation, where coefficients are randomly selected. A receiver can decode original packets by solving corresponding linear equation with received encoded packets. In these methods, packets are also classified into some priority classes according to their importance. In [13], linear equation is created such that more important packets are included into more encoded packets. Thomos et al. optimize the number of packets of each priority class included into one encoded packet in [14]. These UEP methods are suitable for video characteristic that a video content consists of various important pieces of data. However, since these UEP methods protect per only one stream, the lost data can not be sufficiently recovered when packet loss rate of a specific stream becomes high.

Path diversity is effective to alleviate the influence from local congestion. Some methods improve robustness against packet loss by combination of path diversity and other techniques [15], [16], [17]. Zhang et al. propose a FEC method using path diversity which divides a content to some parts and transmits them through multiple paths in [15]. This method maximizes the overall quality by assigning appropriate traffic volume per path. In addition, FEC redundancy is optimized based on these paths' bandwidths and packet loss rates. Multiple Description Coding (MDC) also uses path diversity [16]. This method encodes a content to some descriptions and transfers these descriptions through multi-paths. The receiver can play back the content in quality corresponding to the number of correctly received descriptions, even if some descriptions are not correct. We have proposed Inter-Stream FEC [17]. While the conventional FEC makes parity packets per stream, Inter-Stream FEC makes parity packets from data of multiple streams. Even if one stream is suffered from many lost packets, Inter-Stream FEC can recover the lost data by decoding parity packets with their component data of all the other correctly received streams. However, it has not considered the various importance of video data yet.

In this paper, our proposed method makes redundancy exploiting multiple video streams with considering importance of video data for these problems. The redundancy is decoded by an intermediate node using incoming packets from multiple sources, similar to UEP network coding [13], [14]. The feature of our proposed method is to target multiple streams of different contents. In addition, the proposed method optimizes loss protection for whole of them. On the other hand, the approach in [14] optimizes loss protection per stream. The next section explains ECLIPSE, which is the recovery method that leverages multiple streams' data considering data importance and packet loss rate.

III. Design of ECLIPSE

A. Overview

Our proposed method, ECLIPSE, mainly has two features: restoration exploiting multiple streams from plural servers and differential protection based on both of loss probabilities and data importance. The proposed method recovers lost packets using parity packets constructed by XOR operations of multiple original streams' data. This policy enables to recover a heavily damaged stream

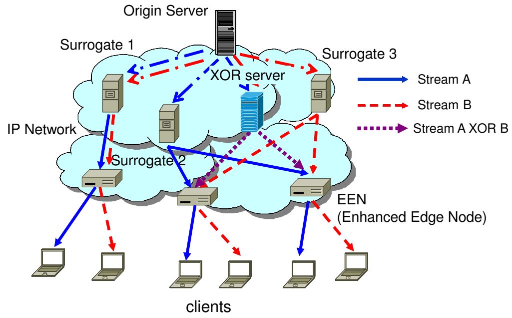


Fig. 1. Target CDN model of the proposed method.

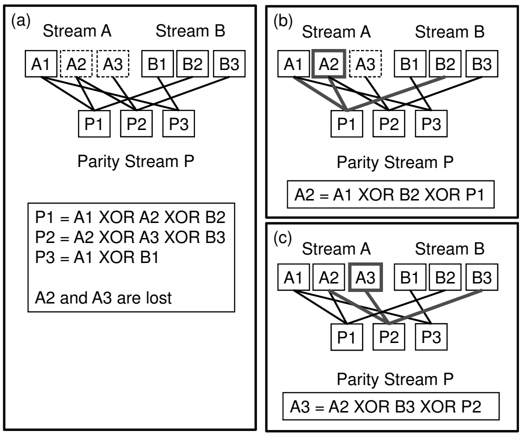


Fig. 2. Example of decoding of parity packets. Solid squares are received packets and dotted squares are lost packets. (a) The EEN receives original packets (A1, B1, B2, B3) and parity packets generated by XOR operation of linked packets. Stream A's packets A2 and A3 are lost. (b) A2 is recovered by XOR operation of corresponding packets. (c) A3 is recovered using A2.

with well-conditioned streams. In addition, the proposed method protects more valuable packets with more parity packets for providing fair and high quality video streaming. As the value for restoration, the proposed method assigns "recovery priority" per packet based on its importance of data and loss probability of the corresponding stream. The proposed method realizes differential protection by making parity packet based on the recovery priority. Moreover, cooperation with conventional FEC method (i.e. Reed-Solomon Erasure Code) can improve recovery capability. This provides more efficient restoration than adjusting redundancy per stream does.

B. System Model

Figure 1 shows the target video streaming system. This system is basically composed of an origin server, surrogates which are arranged in various locations by a content provider, and many clients. Note that surrogates receive limited number of contents (e.g. popular contents) from an origin server in advance for load balancing. In addition, the proposed method deploys the Enhanced Edge Nodes (EENs) and treats some surrogates as "XOR servers."

An EEN is the edge node having packet recovery capability. We assume that an EEN is deployed nearby users (e.g., hotel or condominium apartment) by a content provider for improving video quality. When a client requests a video content from an EEN, the request is delivered to a surrogate having the requested content with round-robin policy for load balancing. If no surrogate has requested content, the request message is delivered to a suitable surrogate in terms of load or distance. In this case, the surrogate newly receives the requested content from the origin server and serves for the request. The surrogate streams the requested content and the client receives it through the EEN. The intermediate EEN manages forwarding streams and recovers packets which are lost between surrogates and itself. In order to recover the lost packets, the EEN requests parity streams from "XOR servers."

XOR servers are nodes which generate parity streams and transmit them to EENs. Although XOR servers can be dynamically selected from surrogates, we assume that a small percent of surrogates are pre-selected as XOR servers for simplicity. We also assume that an XOR server generates parity streams only from contents it has when the XOR server does not have a part of the requested stream to be recovered. In this case, the EEN sends a request message for the not-protected streams to another XOR server. An XOR server selects the streams as targets of restoration according to the information included into request message for parity stream. This message includes three information; stream ID, the sequence number which indicates the start position of encoding and reception ratio per stream. The reception ratio is computed by an EEN as the ratio of the number of received packets to the number of packets the EEN should receive. The EEN estimates the number of packets which should be received based on sequence numbers of packets or the information from the surrogates. The computation is executed when the request message for parity streams is created. We assume the interval of the reception ratio calculation for the message creation is 1 sec. The XOR server generates parity packets with XOR operation among the packets of recovery target streams. These parity packets are sent as parity streams. The parity stream is decoded by the EEN for restoration. Figure 2 shows the example of decoding process when the EEN receives the two original streams and the corresponding parity stream.

C. Assignment Policy of Recovery Priority

The proposed method intensively protects the packets having important data and/or those of less received streams. For this differentiated protection, the proposed method assigns "recovery priority" to each packet based on two criteria; data importance and loss probability. First, the XOR server checks the importance of data (i.e. picture type of MPEG video) which each packet has. The basic priority is decided based on this data importance. This paper defines $r_i (i = 1, \cdots, I)$ as the basic priority, where r_i is a positive number and $r_{i+1} > r_i$. More important

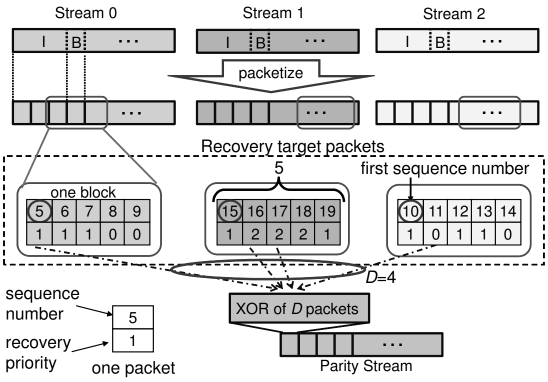


Fig. 3. Creation of parity packet: A XOR server randomly selects D packets among the recovery target packets. The XOR server executes XOR operation to selected D packets. In this example, block size is 5 and D is set to 4.

packets are assigned the basic priority with larger value. In addition to the basic priority, the XOR server dynamically assigns additional priority $s_j(j = 1, \cdots, J)$ per stream based on notified reception ratio by an EEN. The XOR server assigns s_j to the stream whose reception ratio is between Th_j and Th_{j+1}. Here, $Th_j(j = 1, \cdots, J)$ is the threshold for determining additional priority and $Th_j > Th_{j+1}$. Finally, we define recovery priority as $u_{(i,j)}$. The packets with basic priority r_i which belong to a stream of priority s_j are assigned recovery priority $u_{(i,j)} = r_i + s_j$.

The proposed method classifies packets into H classes based on the data importance and loss probability. When the recovery priority $u_h(h = 1, \cdots, H)$ is defined as priority per class, all pairs of (i, j) which fulfill $u_h = u_{(i,j)}$ belong to the class h.

The proposed method creates parity packets using u_h for differential protection. This creation method greatly affects on the performance. Next section describes the creation method of parity packet considering recovery priority.

D. Creation of Parity Packet Considering Recovery Priority

Figure 3 shows the example of creation process on the proposed method. The XOR server first obtains stream IDs and sequence numbers from the request message. Then, the XOR server selects a constant number of packets from each indicated stream. The set of one stream's packets is called "block," and the set of blocks are called "recovery target packets" in this paper. The first packet of each block is the packet with indicated sequence number by request message. Parity packets are created from packets within blocks of multiple streams. Note that these blocks are determined so that packets within these blocks arrive at the EEN within relatively short time. Therefore creation process of parity packet from packets within these blocks shortens the time for the EEN to wait required packets for decoding a parity packet. In addition, since each parity packet is independently

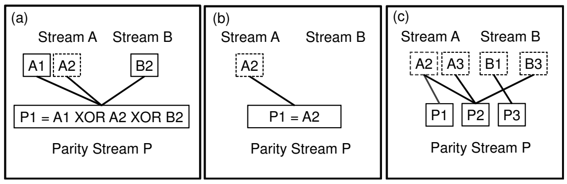


Fig. 4. (a) In this case, EEN receives packets A1, B2 and P1, while packet A2 is lost. P1 was created by XOR operation of connected packets, namely A1, A2 and B2 in the XOR server. (b) The EEN executes XOR operation of received packets. As a result, P1 has information about only A2. (c) By repeating this process about all received parity packets, remained parity packets have various degrees.

decoded from the others, a parity packet can instantly recover a lost packet when it can be decoded. Moreover, the EEN absorbs the delay caused by decoding process by buffering packets. These feature can smoothly recover lost packets with small latency, and as a result, the proposed method can transmit recovered packets before exhaustion of clients' reception buffer, which causes the video quality degradation. Therefore, it is expected that those parity packets can keep real-time characteristics of video streaming. Next, the XOR server gives recovery priority to each packet with assignment policy of recovery priority. Then, the proposed method creates parity packets. A parity packet is created from randomly selected packets within recovery target packets. The proposed method selects more prioritized packets with higher priority. Therefore, packets with more priority are encoded into more parity packets and their opportunities for restoration increase.

In the proposed method, the selection probability $P_h^{(sel)}$ of a packet with u_h is calculated by Eq. (1). In Eq. (1), w_h is the weight of recovery priority u_h for differential protection. The value N_h is the number of packets with priority class h.

$$P_h^{(sel)} = \frac{w_h}{\sum_{\rho=1}^{H} w_\rho N_\rho}. \quad (1)$$

Equation (1) shows that larger weight can increase the selection probability.

When the XOR server makes a parity packet, it randomly selects a packet according to Eq. (1) and encodes it into the parity packet through XOR operation. This selection and encoding is repeated D times. As a result, parity packet is created by the XOR operation of these D packets. Note that this paper assumes that all packets are the same size to simplify encoding.

IV. Analysis of the Proposed Method

In order to design more effective coding, we clarify the characteristics of the proposed method through theoretical analysis. For analyzing the performance of rateless coding, references [11], [18], [19] used technique called "And-Or tree analysis". We analyze our proposed method with similar way to their analyses.

TABLE I
SYMBOLS USED IN ANALYSIS

symbol	representation
x	Step counter of recovery process
D	Maximum degree of one parity packet
d	Degree of one parity packet
$\Omega(d)$	Degree distribution
$\Omega'(d)$	Normalized degree distribution
β_d	The probability that a lost packet is encoded into a parity packet with d degree
l	Average loss rate
M	The number of remained parity packets
m	The number of parity packet into which a lost packet is encoded
F_x	The probability that one packet is still lost at xth step
E_x	The probability that one parity packet recovers a lost packet at xth step
$P^{(enc)}$	Probability that a lost packet is encoded per parity packet.
λ_m	Probability that a parity packet is created from m parity packet
δ_m	Probability that a parity packet includes a lost packet encoded into m parity packet
H	The number of priority class

A. Analysis of Basic Model

As an analysis for simple case, we first consider the case that packets are not classified according to their recovery priority. This paper assumes that the recovery process repeats recovery step until no more packets can be recovered or all packets are recovered. Moreover, the recovery of xth step can use the packets which are recovered before xth step. Since one parity packet can recover one packet, this recovery step is repeated as many times as the number of received parity packets.

Our analysis focuses on parity packets and lost original packets. We suppose that parity packets are processed using received original packets before recovery process. Let degree be the number of original packets which are encoded into one parity packet. The encoded original packets are "connected" to corresponding parity packets in Fig. 4 (a). An edge represents this connection. In this paper, all parity packets originally have D degree since they are created from D original packets. When an original packet is received, it is executed XOR operation with corresponding parity packets. This operation removes the information of the received original packet from the parity packets. This is represented by being "unconnected" between corresponding packets. As a result, the degree of the parity packet decreases (Fig. 4 (b)). When XOR of each received packet is operated to corresponding parity packets, the parity packets have connection with only lost packets (Fig. 4 (c)). We call these parity packets are remained parity packets in this paper. The number of remained parity packets is represented by M.

The symbols used in this analysis are shown in Table I.

Let $\Omega(d)$ be the degree distribution after decoding with received original packets, where $d(d = 0, 1, \cdots, D)$ is degree of one parity packet. This distribution depends on the loss probability. Since the degree reduces when corresponding packets are received, the degree after decoding equals the number of lost packets among its component packets. When d packets are lost among D packets, the remained degree of a parity packet is d. Therefore, $\Omega(d)$ is shown in Eq. (2), where average packet loss rate is l.

$$\Omega(d) \quad = \quad {}_D C_d l^d (1-l)^{(D-d)}. \tag{2}$$

In this paper, we focus on remained parity packets. Therefore, we define the normalized distribution $\Omega'(d)$. The distribution $\Omega'(d)$ is available by normalizing $\Omega(d)$ with the ratio of the d which is larger than zero (Eq. (3)).

$$\Omega'(d) \quad = \quad \frac{{}_D C_d l^d (1-l)^{(D-d)}}{1-(1-l)^D}. \tag{3}$$

Next, we describe the probability that one lost packet is recovered by one parity packet. We define E_x as the probability that one lost packet is recovered by one parity packet at xth step. A parity packet with degree d can recover one lost packet when the other $d-1$ packets have been received or recovered. Therefore, one lost packet which is encoded into a parity packet with degree d is recovered at xth step with probability $(1 - F_x)^{(d-1)}$, where F_x is the probability that one packet is still lost at xth step. Note that $F_1 = 1$ because no restoration has been done in first step. Let β_d be the probability that one lost packet is connected with parity packet having d edges. When there are M remained parity packets, the total number of edges are $M \times \sum_{i=1}^{D} d\Omega'(i)$. The $M \times d\Omega'(d)$ edges are connected with parity packets having d edges among them. The probability β_d equals the probability that one lost packet is connected with one of the $M \times d\Omega'(d)$ edges. Therefore, β_d is calculated by $\frac{d\Omega'(d)}{\sum_{i=1}^{D} d\Omega'(i)}$. The probability E_x is given by Eq. (4).

$$E_x \quad = \quad \sum_{d=1}^{D} \left\{ \beta_d (1-F_x)^{d-1} \right\}. \tag{4}$$

A lost packet can not be recovered when no parity packet recovers it. Therefore, a packet is still lost at $x+1$th step with probability $F_{(x+1)}$, when all parity packets fail to recover it at xth step. Let $F_{(x+1)}^{(m)}$ be the probability that one packet which is encoded into m parity packets is still be lost at $(x+1)$ step. The probability $F_{(x+1)}^{(m)}$ is given by Eq. (5) using the probability E_x with which one parity packet can recover the lost packet.

$$F_{(x+1)}^{(m)} \quad = \quad (1-E_x)^m. \tag{5}$$

The probability $F_{(x+1)}$ is calculated by considering the all cases about m. When δ_m is the probability that a parity packet is connected with lost packets which is encoded into m parity packets, $F_{(x+1)}$ is shown as follows.

$$F_{x+1} = \sum_{m=0}^{M} \delta_m F_{(x+1),m} = \sum_{m=0}^{M} \delta_m (1-E_x)^m. \quad (6)$$

The probability δ_m is calculated as follows. Let $P^{(enc)}$ be the probability that one lost packet is encoded into one parity packet. The value $P^{(enc)}$ equals the probability that one lost packet is selected through d selection as component of parity packet from all lost packets. Let n be the number of original packets and n' be the number of lost packets, where $n' = n \times (1-l)$. This probability is calculated by Eq. (7).

$$P^{(enc)} = \sum_{d=1}^{D} \left(\Omega'(d) \times \frac{{}_{n'-1}C_{d-1}}{{}_{n'}C_d} \right) = \sum_{d=1}^{D} \left(\Omega'(d) \times \frac{d}{n'} \right). \quad (7)$$

The value λ_m denotes the probability that a packet is encoded in m parity packets. Then λ_m is described as follows, where M is the number of remained parity packets.

$$\lambda_m = {}_MC_m (P^{(enc)})^m (1-P^{(enc)})^{M-m} \sim \frac{e^{-MP^{(enc)}} (MP^{(enc)})^m}{m!}. \quad (8)$$

One parity packet is connected with a lost packet which is encoded into m parity packet with probability δ_m. The probability δ_m is calculated with the similar way as β_d, and it is shown by $\frac{m\lambda_m}{\sum_{m=0}^{M} m\lambda_m}$.

The performance of the proposed method is denoted by Eq. (6) when packets are not classified. Next, we extend the above discussion into the case that packets are classified to several classes by their loss probabilities.

B. Analysis of Extended Model

In this section, we analyze the performance considering priority classes. Our analysis considers the possible combination of degrees of priority classes, unlike the previous work [11] that uses the average of degree in order to simplify the calculation. Although the analysis in [11] considers the priority class, our analysis can investigate the impact of the degree distribution in more detail due to its strict calculation. When packets are classified based on their recovery priority, they are encoded into parity packets with different probability per class (Eq. (1)). Therefore, the number of encoded original packets into a parity packet also is different. Let D_h be the average number of encoded original packets with recovery priority u_h. The value D_h is represented by Eq. (9).

$$D_h = D \times P_h^{(enc)}. \quad (9)$$

Considering recovery priority, the degree distribution is also replaced. Let d_h be the remained parity packet's degree corresponding the packets with recovery priority u_h. Let l_h be the loss probability of packet with recovery priority u_h. The degree distribution considering recovery priority $\Omega'(d_1, \cdots, d_H)$ is shown as follows.

$$\Omega'(d_1, \cdots, d_H) = \frac{\prod_{h=1}^{H} {}_{D_h}C_{d_h} l_h^{d_h} (1-l_h)^{(D_h - d_h)}}{1 - \prod_{h=1}^{H} (1-l_h)^{D_h}} \quad (10)$$

Let $\beta_{h,(d_1,\cdots,d_H)}$ be the probability that a lost packet with recovery priority u_h is encoded into a parity packet which is created from $\sum_{h=1}^{H} d_h$ packets. Note that d_h is the number of encoded packets with recovery priority u_h. The probability $\beta_{h,(d_1,\cdots,d_H)}$ is also calculated with a similar way as β_d. Let $\Omega'_h(d)$ be the probability that one parity packet is connected with d lost packets which is assigned recovery priority u_h. The probability $\Omega'_h(d)$ is calculated by sum of all $\Omega'(d_1, \cdots, d_H)$ whose d_h is d. The probability $\beta_{h,(d_1,\cdots,d_H)}$ is calculated using $\Omega'_h(d)$ as follows.

$$\beta_{h,(d_1,\cdots,d_H)} = \frac{d_h \Omega'(d_1, \cdots, d_H)}{\sum_{d=1}^{D_h} d\Omega'_h(d)}. \quad (11)$$

The probability that one parity packet recovers one lost packet depends on its degree. Let $E_{h,x}$ be the recovery probability on step x, and $F_{h,x}$ be the probability that one packet with recovery priority u_h is still lost at xth step. Let $E_{h,x,(d_1,\cdots,d_H)}$ be the probability that one parity packet whose degrees of recovery priorities are $d_1, \cdots, d_H$ can recover one lost packet with recovery priority u_h at xth step. As mentioned above, one lost packet with degrees $d_1, \cdots, d_H$ can recover lost packet with recovery priority u_h when all the other lost packets have been recovered except the lost packet. Therefore, $E_{h,x,(d_1,\cdots,d_h)}$ is calculated as follows.

$$E_{h,x,(d_1,\cdots,d_H)} = (1-F_{1,x})^{d_1} \times \cdots \times (1-F_{h-1,x})^{d_{h-1}} \times (1-F_{h,x})^{(d_h - 1)} \times (1-F_{h+1,x})^{d_{h+1}} \times \cdots \times (1-F_{H,x})^{d_H}. \quad (12)$$

Specifically, the value $d_h - 1$ is the number of packet with recovery priority u_h encoded into the parity packet except the lost packet. The probability $E_{h,x}$ is calculated considering all cases about degree distribution as follows.

$$E_{h,x} = \sum_{d_1=0}^{D_1} \cdots \sum_{d_h=1}^{D_h} \cdots \sum_{d_H=0}^{D_H} \beta_{h,(d_1,\cdots,d_H)} E_{h,x,(d_1,\cdots,d_H)}. \quad (13)$$

Note that $d_h \geq 1$ because the parity packets have the connection with the lost packet.

The probability that one packet is encoded into one parity packet is calculated separately per class. Let n'_h be the number of lost packets with recovery priority u_h. The

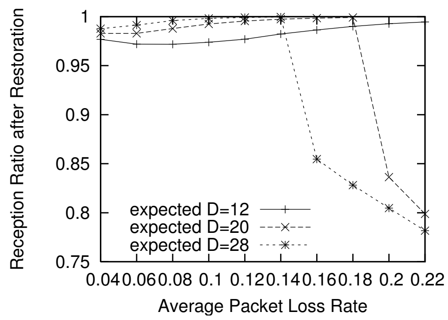


Fig. 5. Expected reception ratio after restoration with parity stream. The expected reception ratio is calculated by $\lim_{x\to\infty} 1 - n' \times F_x$.

probability $P_h^{(enc)}$ that a packet with recovery priority u_h is encoded into one parity packet is given as follows using n'_h and $\Omega'_h(d)$.

$$P_h^{(enc)} = \sum_{d=1}^{D_h} \left(\Omega'_h(d) \times \frac{d}{n'_h} \right). \quad (14)$$

The probability that one packet is encoded in m parity packet is calculated separately per class. Let $\lambda_{h,m}$ be the encoded probability of one packet with recovery priority u_h. The probability is given by Eq. (15).

$$\lambda_{h,m} \sim \frac{e^{-MP_h^{(enc)}}(MP_h^{(enc)})^m}{m!}. \quad (15)$$

The probability δ_m is also extended for prioritized packets and we define it as $\delta_{h,m}$, where $\delta_{h,m} = \frac{m\lambda_{h,m}}{\sum_{m=0}^{M} m\lambda_{h,m}}$. After all, one packet with recovery priority u_h is still lost at $(x+1)$th step with the probability $F_{h,x+1}$ through xth recovery step. The probability $F_{h,x+1}$ is shown as follows.

$$F_{h,x+1} = \sum_{m=0}^{M} \delta_{h,m}(1 - E_{h,x})^m. \quad (16)$$

Figure 5 shows the expected reception ratio after restoration through the above analysis. Note that restoration in this section is processed using only parity stream. In Fig. 5, the reception ratio after restoration is plotted when average loss rate changes 0.04 to 0.22. Each line corresponds to the result with different D. This figure shows that the appropriate D changes due to the average packet loss rate. This result indicates that setting D based on the loss probability can improve the recovery capability. Next section proposes the adaptive control method of degree D.

V. Control Method of the Number of Encoded Packets

The number of the encoded packets into one parity packet, D, is crucial for the recovery capability. One parity packet can recover only one lost original packet when the rest $D-1$ original packets are received. Generally, large D can increase the probability that a lost packet is included in the D packets. However, if two or more packets are lost within D packets, the parity packet can not recover any lost packets. This means that D should be decided carefully.

As mentioned in previous section, Fig. 5 shows that the reception ratio after restoration has individual peak point according to different D. This result verifies the above mention. Therefore, the control method of D is required to correspond with various network conditions.

The proposed method checks whether D is appropriate based on the estimated efficiency of parity packet. This paper considers the probability that one parity packet recovers a lost packet, R_D (Eq. (17)). The probability R_D means that $D-1$ original packets are received and just one original packet is lost. Let q_i be a packet loss rate of the stream which ith selected packet belongs to.

$$\begin{aligned} R_D &= q_1 \times (1-q_2) \times \cdots \times (1-q_D) \\ &+ (1-q_1) \times q_2 \times \cdots \times (1-q_D) + \cdots \\ &+ (1-q_1) \times \cdots \times (1-q_{D-1}) \times q_D \\ &= \left\{ \prod_{i=1}^{D} (1-q_i) \right\} \times \left\{ \sum_{i=1}^{D} \frac{q_i}{(1-q_i)} \right\}. \end{aligned} \quad (17)$$

When R_D is close to 1.0, the parity packet can quickly recover a lost packet with high probability. However, D tends to be small in this case. This means that some of original packets are not sufficiently protected. Therefore, this paper selects large D for sufficient restoration at the expense of longer decoding time. It is desirable that all lost packets are protected with parity packet. The probability that one lost packet is not protected by any parity packet is given by λ_0 (Eq. (8)). When τ is the acceptable ratio of non-protected packets, λ_0 should fulfill the following equation.

$$\lambda_0 = (1 - P^{(enc)})^M < \tau. \quad (18)$$

Equation (18) is transformed as follows.

$$\begin{aligned} P^{(enc)} &= \sum_{d=1}^{D} \left(\Omega'(d) \times \frac{d}{n'} \right) \\ &= \sum_{d=1}^{D} \left(\frac{{}_D C_d l^d (1-l)^{(D-d)}}{1-(1-l)^D} \times \frac{d}{n'} \right) \\ &\sim \frac{1}{n'\{1-(1-l)^D\}} \sum_{d=1}^{D} \left\{ d \times \frac{e^{-Dl}(Dl)^d}{d!} \right\} \\ &\sim \frac{e^{-Dl}Dl}{n'(Dl - \frac{D(D-1)}{2}l^2)} \times \sum_{d=1}^{D} \left\{ \frac{(Dl)^{d-1}}{(d-1)!} \right\} \\ &\sim \frac{1}{n'(1-\frac{D-1}{2}l)} > 1 - \tau^{1/M}. \end{aligned} \quad (19)$$

$$D > \frac{2}{l} \left\{ 1 - \frac{1}{n'(1-\tau^{1/M})} \right\} + 1. \quad (20)$$

Specifically, when τ is enough small, Eq. (20) approaches to Eq. (21).

$$D > \frac{2(n'-1)}{n'l} + 1. \quad (21)$$

For simplicity, we employ Eq. (21).

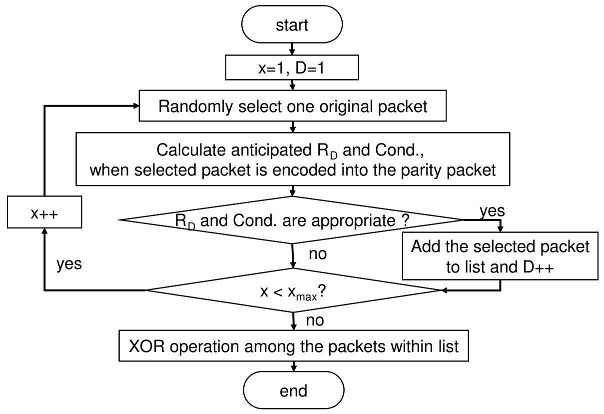


Fig. 6. Flowchart of the parity packet creation. The condition Eq.(21) is represented by "Cond." in this figure. Considering the encoding time, the number of selection is limited to x_{max}, whose default value is 30.

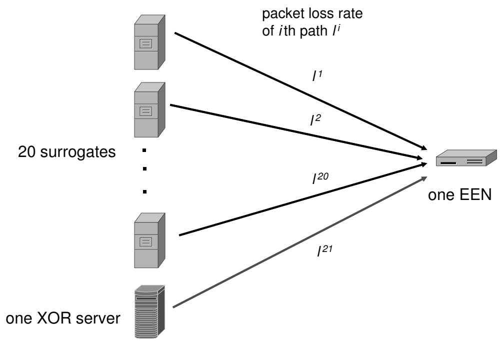


Fig. 7. Simulation model.

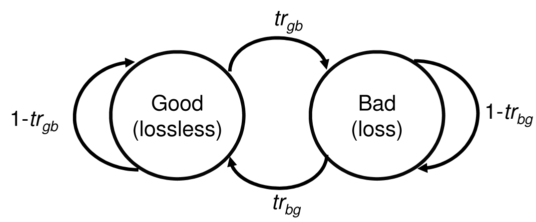


Fig. 8. Packet loss occurs according to Gilbert model. In this model, a packet is lost when path is under bad condition. The path condition changes to next condition per each packet's passage. When a path is under good condition, its condition transits to bad condition with probability tr_{gb}. The probability of transition from bad condition to good condition is represented by tr_{bg}.

The ECLIPSE creates parity packets with above two methods. Figure 6 shows the flowchart of the parity packet creation.

VI. Performance evaluation

A. Simulation Model

We evaluate the proposed method through computer simulations. The simulation model is illustrated in Fig. 7. We assume no packet is lost between the EEN and clients, and therefore, this simulation is focused on the surrogates, XOR server and EEN. Between each server and EEN, packet loss occurs with different rate. These loss rates follow normalized distribution whose mean value is plr and standard deviation is sd. The value sd is set to 0.05 in the simulations, while the value plr is changed as a parameter. The loss characteristics are modeled via Gilbert model in Fig. 8 [20]. In this loss model, the average loss rate plr is calculated as follows.

$$plr = \frac{tr_{gb}}{tr_{gb} + tr_{bg}}. \tag{22}$$

The burst loss length "len" is given as follows.

$$len = \frac{1}{tr_{bg}}. \tag{23}$$

The values tr_{gb} and tr_{bg} are given such that plr and len are required value.

The common conditions are shown in Table II. In addition, this paper assumes that a packet does not have the data composed of multiple frames. When a residual data size is less than the packet size, the rest part is filled with padding signal. Table III shows the default values of parameters in the proposed method. The ratio of important packets to whole packets is calculated based on a sample video data.

B. Numerical Result

1) Impact of parameters: Under the above environment, we first evaluate the performance changing some parameters as basic evaluations. In order to verify the impact only of parameters, we consider a scenario that no packet is lost between the XOR server and the EEN in this section. Figure 9 shows the recovery capability when the threshold R_D for control of D is various values. Final reception ratio is the ratio of received or recovered packets after restoration with both of conventional FEC and the proposed method. This figure shows that the final reception ratio is drastically improved with increasing the value of the threshold in the range where R_D is less than 0.25. On the other hand, when R_D is more than 0.25, the final reception ratio decreases slightly with the increase of R_D. This means that too small threshold decreases the restoration performance due to frequent creation of parity packet using too many original packets. From this result, this paper uses 0.25 for R_D in the following evaluation. Figure 10 shows the comparison about the performance of control method of D (prop with ctrl) with the no dynamic control case when D is 12 (prop $D = 12$) and D is 20 (prop $D = 20$). These constant values are the best D when packet loss rates are 0.20 and 0.12, respectively. We underscore that each stream is protected by FEC redundancy and that a parity packet can recover lost packets when only one packet is lost among the encoded D packets. In other words, lost packets can not be recovered if two or more packets among the encoded D packets are lost. If D is large value, this unrecovered case often occurs. However, if D is small value, only few packets are encoded, and therefore, most of lost packets can not be recovered. The final reception ratio depends

TABLE II
COMMON CONDITIONS

name	value
Transmission protocol	UDP
Packet size	1500 [Bytes]
Transmission bit rate	1.2 [Mbps]
Request interval for parity stream	1.0 [s]
The number of recovery target streams	20
Block size	100 [packets]
FEC redundancy (n, k)	(100, 95)
Burst loss length	5

TABLE III
DEFAULT VALUE OF PARAMETERS IN THE PROPOSED METHOD

name	value
Basic priority r_i	(1, 2)
Priority of more important data	2
Priority of less important data	1
Additional priority s_j	(0, 1, 2)
Threshold for reception ratio Th_j	(1.0, 0.9, 0.8)
Recovery priority u_h	(1, 2, 3, 4)
Weight of priority w_h	(1, 2, 4, 8)
Ratio of important packets to whole packets	0.73
Ratio of parity packets to expected lost packets	1.5

on this characteristics. In addition, simple FEC method also influences the final reception ratio. As prominently shown in Fig. 10 ($D = 12$), change of the final reception ratio is divided into three patterns when D is a fixed value. 1) The first pattern is shown when plr is very low (e.g. $0.04 < plr < 0.08$). In this case, the probability that no packet is lost within the encoded D packets in a parity packet is high. These mean that a parity packet tends not to be used for restoration, especially, when D is small value. However, simple FEC method can recover lost packets even if parity packets can not recover them. Therefore, when D is small (e.g. $D = 12$), the most recovered packets are due to simple FEC method, and as a result, the final reception ratio slightly decreases with increasing plr. 2) The second pattern is shown when plr is low (e.g. $0.08 < plr < 0.16$). When plr increases, the probability that one packet is lost within the encoded D packets also increases. This means a parity packet can also effectively recover the lost packet in addition to the simple FEC. Therefore, the final reception ratio increases with increasing plr. 3) The third pattern is shown when plr is high (e.g. $plr > 0.16$). In this case, the probability that a parity packet fails to recover the lost packet is high when plr is high, because two or more packets are lost among the encoded D packets. Therefore the final reception ratio decreases with increasing plr. On the other hand, the proposed method with control D can maintain its recovery capability even if plr changes. From these

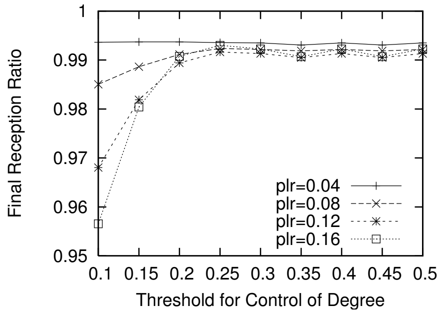


Fig. 9. Final reception ratio with changing threshold for control of D.

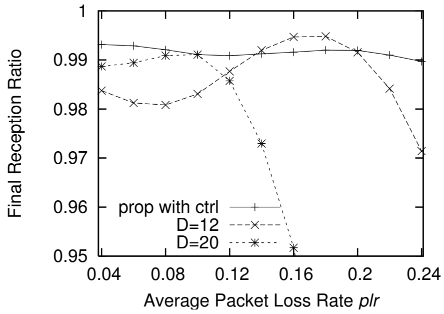


Fig. 10. Final reception ratio with changing packet loss rate.

results, the proposed method with control of D is verified having self-adjusting capability for network condition. Figure 11 shows the final reception ratio when weight of priority is various values. The legends represent weight of each priority in ascending order. The final reception ratio of important packets is represented with “high” and that of less important packet is represented with “low”. Note that the redundancy is 1.2 in this result. In the proposed method, the proportion of weight of each priority affects the recovery capability. Figure 11 shows that the gap of recovery capabilities between important packets and other packets is wide, when higher priority has larger portion of weight. We confirmed that when redundancy is larger, the gap between different weights is less. Figure 12 shows the performance about the recovery capability with changing redundancy from 0.8 times to 1.7 times as many as the number of expected lost packets. The value plr is set to 0.04 to 0.18. This figure shows that more parity packets can more improve recovery capability. Specifically, when the EEN receives parity packets 1.3 times as many as the lost packets, the EEN can recover most of lost packets at the same degree in any packet loss rate. We also show the relationship between the value of plr and the number of parity packets from another perspective in Fig. 13. Figure 13 shows the required number of parity packets for the final reception ratio to exceed the given value. The legends represent the required final reception ratio. The required number of parity packets linearly increases with increasing plr. This result shows that it is valid to decide the number of parity packets as the proportional number of the packet loss rate.

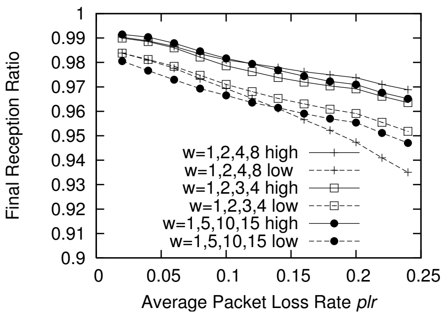


Fig. 11. Final reception ratio with changing weight of priority.

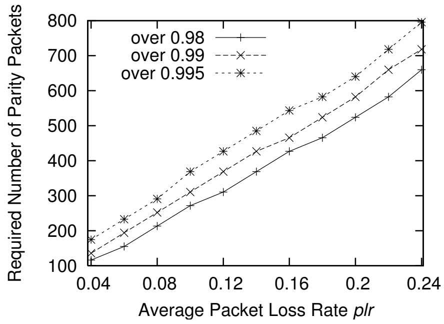


Fig. 13. Required number of parity packets for final reception ratio to exceed given value with changing packet loss rate.

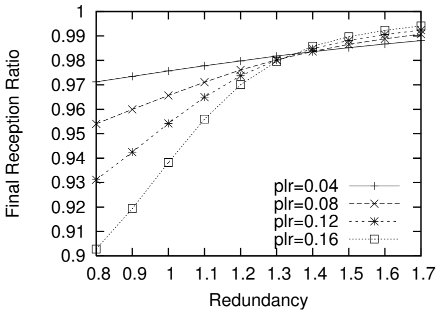


Fig. 12. Final reception ratio with changing redundancy.

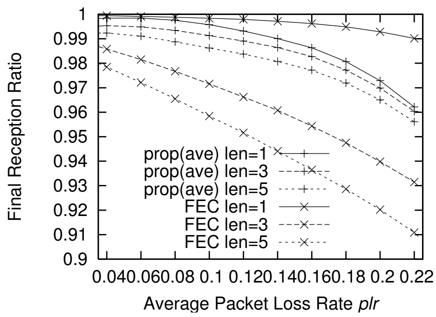


Fig. 14. Comparison about final Reception ratio with conventional FEC about burst loss length.

2) Evaluation of restoration characteristics: Next, we evaluate the proposed method in comparison with conventional FEC method. In this section, a parity stream is also suffered from packet loss unlike previous section for evaluations on more practical environment. In each of the proposed method and FEC method, the redundancy is set to 1.5 times as many as the expected loss packets. Note that the redundancy of FEC method is calculated per stream and the sum of redundancy is almost same as the proposed method. Figure 14 shows the final reception ratio when burst loss length is set to be various values. In the legends, "prop" represents our proposed method and len is the burst loss length which is given as the parameter of Gilbert model. While the FEC method recovers most lost data when burst loss length is 1, its performance drastically degrades when burst loss length is lager value. This result represents the simple FEC method can not provide sufficient protection in the environment that packet loss explosively occurs. On the other hand, the proposed method can maintain its recovery capability even when burst loss length is large. We also evaluate about the fairness of the proposed method. Figure 15 shows distribution of final reception ratio against the reception ratio before restoration when average packet loss rate is 0.10 and burst loss length is 5. In each graph, the final reception ratios of 600 streams are plotted. Figure 15 shows that FEC can not sufficiently recover the streams whose packet loss rates are high. On the other hand, the proposed method recovers all streams at similar level regardless of their reception ratios.

Finally, we evaluate the results of Peak Signal to Nose Ratio (PSNR) in Figs. 16 and 17. Each figure shows the PSNRs before and after recovery process to 600 streams. In these evaluations, plr is 0.10 and len is 5. The FEC method can not recover lost packets of a heavily damaged stream. Therefore, PSNRs of the FEC method is scattered in Fig. 16. Figure 17 shows that PSNRs are in high level in the proposed method. In particular, PSNRs of the proposed method are more similar level than those of the FEC method. We also show the average and variance of PSNR and the ratio of the number of streams whose PSNRs are larger than 35 dB in Table IV. The value 35 dB corresponds to the PSNR where user could achieve good video quality in [21]. We confirm the above mentioned performance of the proposed method from this results. These results indicate that the proposed method can provide fair and high quality video streaming services.

VII. Conclusion

This paper proposed the recovery method suitable for video streaming whose data have various importance. The goal of the ECLIPSE is to improve QoS and fairness among streams. In order to realize the goal, the ECLIPSE assigns "recovery priority" to each packet based on the importance of data and the reception ratio of each stream. Firstly, it assigns more priority to packets with more important data. Secondly, additional priority is assigned to packets of streams whose loss rates are high. The proposed method recovers important data with higher probability by the restoration based on the recovery

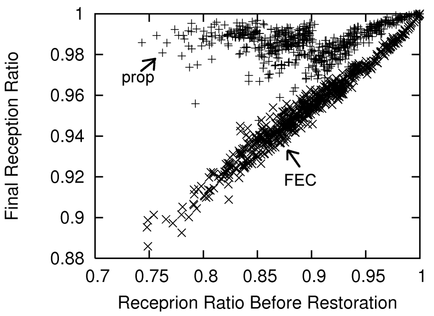


Fig. 15. Distribution of final reception ratio.

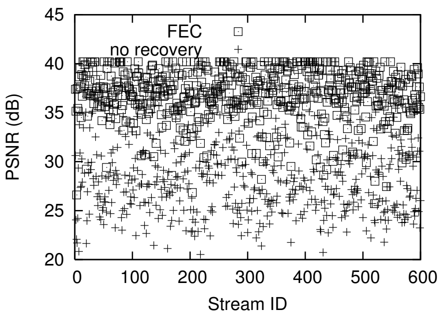


Fig. 16. Distribution of PSNR of FEC method.

priority. In addition, we analyzed the performance of the ECLIPSE for designing more effective coding. The control method for parity packet creation was also proposed in order to improve recovery capability. Through computer simulations, we confirmed that the proposed method achieves the required characteristics mentioned above. As future works, we evaluate the performance of the proposed method under the environment with delay constraint. The cooperation of the multiple EENs or XOR servers should be designed in order to provide high quality video streaming services for future networks.

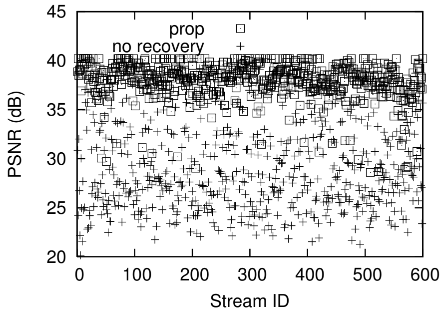


Fig. 17. Distribution of PSNR of the proposed method.

TABLE IV
EVALUATION ABOUT PSNR

	average	variance	ratio of streams whose PSNR > 35 dB
FEC	36.131	2.7715	0.73177
prop	37.879	1.9879	0.94000

REFERENCES

[1] G. Peng, "CDN: Content distribution network," *Arxiv preprint cs/0411069*, pp. 1–26, 2004.

[2] J. Apostolopoulos, T. Wong, and S. Wee, "On multiple description streaming with content delivery networks," *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, no. June, pp. 1736–1745, 2002.

[3] T. Lam, L. Karam, R. Bazzi, and G. Abousleman, "Reduced-delay selective ARQ for low bit-rate image and multimedia data transmission," in *Proc. IEEE ICASSP*, 2005, pp. 309–312.

[4] X. Yu, J. Modestino, and Y. Chan, "A model-based evaluation methodology for assessing the efficacy of packet-level fec for delay-constrained video network transport," *IEEE ICIP 2006*, pp. 1289–1292, 2006.

[5] R. Cohen, G. Grebla, and L. Katzir, "Cross-Layer Hybrid FEC/ARQ Reliable Multicast with Adaptive Modulation and Coding in Broadband Wireless Networks," *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*, pp. 1917–1925, Apr. 2009.

[6] P.-J. Wu, J.-N. Hwang, C.-N. Lee, and Y.-C. Teng, "Receiver driven overlap FEC for scalable video coding extension of the H.264/AVC," *2009 IEEE International Symposium on Circuits and Systems*, pp. 637–640, May 2009.

[7] E. Maani and A. K. Katsaggelos, "Unequal Error Protection for Robust Streaming of Scalable Video Over Packet Lossy Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 3, pp. 407–416, Mar. 2010.

[8] A. Talari and N. Rahnavard, "Unequal error protection rateless coding for efficient MPEG video transmission," *MILCOM 2009 - 2009 IEEE Military Communications Conference*, pp. 1–7, Oct. 2009.

[9] A. G. Dimakis, J. Wang, and K. Ramchandran, "Unequal Growth Codes: Intermediate Performance and Unequal Error Protection for Video Streaming," *2007 IEEE 9th Workshop on Multimedia Signal Processing*, pp. 107–110, 2007.

[10] S.-K. Chang, K.-C. Yang, and J.-S. Wang, "Unequal-Protected LT Code for Layered Video Streaming," *2008 IEEE International Conference on Communications*, pp. 500–504, May 2008.

[11] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Rateless Codes With Unequal Error Protection Property," *IEEE Transactions on Information Theory*, vol. 53, no. 4, pp. 1521–1532, Apr. 2007.

[12] J. Chakareski and P. Frossard, "Distributed streaming via packet partitioning," in *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 1529–1532.

[13] Y. Lin, B. Li, and B. Liang, "Differentiated Data Persistence with Priority Random Linear Codes," *27th International Conference on Distributed Computing Systems (ICDCS '07)*, June 2007.

[14] N. Thomos, J. Chakareski, and P. Frossard, "Prioritized Distributed Video Delivery With Randomized Network Coding," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 776–787, Aug. 2011.

[15] H. Zhang, J. Zhou, and J. Li, "M2FEC: An effective FEC based multi-path transmission scheme for interactive multimedia communication," *Journal of Visual Communication and Image Representation*, vol. 21, no. 2, pp. 120–128, Feb. 2010.

[16] M. Biswas, M. R. Frater, J. F. Arnold, and M. R. Pickering, "Improved Resilience for Video Over Packet Loss Networks With MDC and Optimized Packetization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 10, pp. 1556–1560, Oct. 2009.

[17] H. Tode, T. Fujie, and K. Murakami, "Inter-Stream FEC Method for Multiserver Content Distribution," *IEEE GLOBECOM 2007-2007 IEEE Global Telecommunications Conference*, pp. 2043–2047, Nov. 2007.

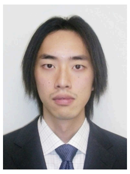
[18] M. Luby and M. Mitzenmacher, "Analysis of Random Processes via And-or Tree Evaluation," *Proceeding of the ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 364–373, 1998.

[19] D. Sejdinovi, R. J. Piechocki, and A. Doufexi, "AND-OR Tree Analysis of Distributed LT Codes," *IEEE Information Theory Workshop on Networking and Information Theory, 2009. ITW 2009*, pp. 261–265, June 2009.

[20] E. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J*, vol. 39, no. 9, pp. 1253–1265, 1960.

[21] Z. Guan, D. Yuan and H. Zhang, “Optimal and Fair Resource Allocation for Multiuser Wireless Multimedia Transmissions”, *EURASIP J. Wirel. Commun. Netw.*, vol. 2009, no. 17, pp. 17:1–17:10, Jan. 2009.

Akihiro Fujimoto received the B. E. and M. E. degrees from Osaka University, Osaka, Japan, in 2008 and 2010 respectively. At present, he is a doctor course student at the Department of Information Networking, Graduate School of Information Science and Technology, Osaka University. His research interests include multimedia streaming system.



Yusuke Hirota received the B. E., M. E. and Ph. D. degrees from Osaka University, Osaka, Japan, in 2004, 2006 and 2008, respectively. He was adopted as assistant professor at the Graduate School of Information Science and Technology, Osaka University on Oct. 2008. His research interest includes routing and wavelength assignment, switching architecture in optical networks, multicasting, multimedia streaming system, peer to peer communication, and autonomous networks. Dr. Hirota is a member of IEEE and OSA.

Hideki Tode received the B. E., M. E., and Ph. D degrees in communications engineering from Osaka University in 1988, 1990, and 1997, respectively. He was adopted as assistant professor at the Department of Communications Engineering, Osaka University from a doctor’s course student on Dec. 1, 1991. In 1998 and 1999, he shifted and promoted to a lecturer and associate professor in the Department of Information Systems Engineering. In 2002, He belonged to the Department of Information Networking, Graduate School of Information Science and Technology, Osaka University. Since 2008, he became a professor in the Department of Computer Science and Intelligent Systems, Graduate School of Engineering, Osaka Prefecture University. His current research interests include high speed node architecture, routing problem, traffic control, optical network architecture and contents delivery applications on various networks. Dr. Tode is a member of IEEE.

Koso Murakami received the B. E., M. E. and Ph. D degrees from Osaka University, Osaka, Japan in 1971, 1973 and 1991 respectively. From 1973 to 1995, he was with Fujitsu Laboratories Ltd., Kawasaki, Japan and engaged in research and development of digital switching systems, Asynchronous Transfer Mode switching systems and photonic switching technologies. From 1990 to 1995, he was a Manager of Communications Network Systems Laboratory of the same company and was responsible for research and development of broadband ISDN, Intelligent network, Telecommunications Management Network and personal communication networks. In April 1995, he joined Osaka University and was a Professor of Computation Center of the same University from 1995 to 1997. From 1998 to 2001, he was a Professor in the department of Information System Engineering of Osaka University. Since 2002, he has been a Professor of the Department of Information Networking in Graduate School of Information Science and Technology, newly established in Osaka University. From 2001 to 2004, he was concurrently Director of Collaborative Research Center of Advanced Science and Technology of the same University. From April 2004, he has served concurrently as Manager of Intellectual Property Office of Osaka University. His research interests extend to multimedia information networking architecture and photonic network. Prof. Murakami is a Fellow of IEICE and an IEEE Fellow.