

High Quality Streaming System with Hierarchical Cache Servers Based on Inter-Stream FEC Function

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 using hierarchical cache servers is effective to provide high quality services to many users. However, less-accessed videos tend to be sparsely cached on cache servers far from users. Streaming qualities are degraded due to relatively high packet loss rate and long delay. In order to address the issues on heavy and bursty packet loss over such long distance transmission, we have focused on recovery characteristic of a multi-server Forward Error Correction (FEC) function named “Inter-Stream FEC.” In this paper, we propose a high quality streaming system exploiting hierarchical cache servers. The proposed system improves users’ experience by adaptively applying Inter-Stream FEC. We also propose a parity data request method combines with parity server selection considering cache status of individual server and disjointness between paths. Simulation results show the proposed system can achieve higher restoration performance compared with a streaming system using traditional FEC, particularly when bursty loss occurs.

Index Terms—Forward Error Correction, Inter-Stream FEC, Hierarchical cache servers, Path diversity

I. INTRODUCTION

In recent years, video streaming services have become popular and major applications in the Internet. The video streaming has some characteristics such as high bit rate and delay restriction. In order to improve Quality of Service (QoS) and Quality of Experience (QoE), video streaming distribution systems need to consider its characteristics. One of effective approaches for such video streaming services is to cache content on multiple cache servers, such as Content Delivery Network (CDN) [1] and IPTV [2]. In general, cache servers are organized into hierarchical structure [3], [4], [5]. In such structure, more popular content is inherently cached on cache servers closer to users. Therefore, the cache servers close to users can deal with many requests for popular content. On the other hand, less-popular content tends to be only sparsely cached on distant servers. The cache servers far from users must deal with such users’ requests.

Even if cache servers are applied, packet loss could cause significant degradation of video qualities, especially, on streams delivered from distant servers. Therefore, recovery of lost data is necessary to alleviate the quality degradation. Various methods have been studied to recover the lost data, such as Automatic Repeat reQuest (ARQ) [6], Forward Error Correction (FEC) [7] and hybrid of them [8]. However, since these recovery methods generally send data for recovery through the same path with the original data, the data for

recovery often suffers from packet loss, especially when many packets are bursty lost.

For this problem, we have proposed a FEC approach leveraging multiple servers, named “Inter-Stream FEC” [9], [10]. While the conventional FEC, called “intra-stream FEC” in this paper, makes parity packets from individual stream and appends them to the stream, Inter-Stream FEC makes a parity stream using data of N streams transmitted originally from multiple servers. This method can recover bursty lost packets of one stream as long as all the other N streams, including the parity stream, are correctly received. The recovery characteristic of Inter-Stream FEC is considered to be suitable for heavy and bursty packet loss over long distance transmission. However, the previous work focused on the effective FEC algorithm and did not consider the characteristics of distribution system, such as cache server hierarchy and path disjointness. In order to maximize the performance, Inter-stream FEC should be adaptively applied based on the condition of the distribution systems.

In this paper, we propose an adaptive streaming system exploiting hierarchical cache servers which enables to provide high quality services based on Inter-Stream FEC function. Our goal is to provide high quality services to more users without significant increase of redundancy. For this goal, firstly, the proposed system adaptively applies Inter-Stream FEC to video streams which should be strongly protected. Secondly, in order to further improve the recovery capability and redundancy, we propose a parity data request method combined with parity server selection considering cache status of individual server and disjointness between transmission paths. This method requests the minimum number of parity streams which sufficiently recover lost packets, from appropriate parity stream servers. The main contribution of this paper is to analyze the range of application of Inter-Stream FEC in a realistic environment, and to present a system design for high quality streaming services by establishing cache-and-topology aware parity data request method.

The rest of this paper is organized as follows. Section II introduces related works and the overview of the Inter-Stream FEC. Section III describes the proposed distribution system in detail. Section IV evaluates the performance of the proposed distribution system through computer simulations. Finally, Section V concludes this paper.

II. EXISTING RESTORATION APPROACHES FOR QoE IMPROVEMENT

A. Restoration exploiting Multi-path Transmission

Many researchers have proposed reliable multimedia streaming schemes exploiting multi-path transmission in single source content distribution. The studies in [11], [12], [13] utilize FEC methods in order to recover lost packets. In [11], video packets are transmitted through multiple paths with FEC redundancy, which is assigned per path based on its characteristics. Tsai et al. utilize path diversity for dispersing bursty packet loss over different FEC blocks in order to improve FEC efficiency in [12]. This method decentralizes bursty packet loss by sending video packets in an interleaving manner across multiple paths. Zhang et al. propose a FEC method which is designed to maximize the estimated video quality in [13]. This method calculates FEC redundancy, video bitrate and a transmission schedule which can maximize video quality by solving an optimization problem.

In contrast to multi-path transmission from single source in above studies, references [14] and [15] propose path diversity using multiple source nodes. Chakareski and Girod propose a rate-distortion optimization framework in the receiver-driven multi-server streaming system in [14]. In [15], Multiple Description Coding (MDC) is combined with path diversity in CDN. In MDC, video data are encoded into multiple descriptions. A receiver can play back at baseline quality if it receives one of descriptions. In the case that the receiver receives more descriptions, it can play back at higher quality. Path diversity is feasible in principle because descriptions through different paths show different loss patterns. Random network coding is the technique which inherently leverages both source diversity and path diversity [16]. In the random network coding, intermediate nodes on the multiple paths independently process the relayed packets. Each node encodes randomly selected incoming packets into an outgoing packet and sends the coded packets to the next nodes. Receivers can reconstruct the original packets if they can receive enough number of coded packets. This property is effective for improving loss resilience since the receivers have only to receive enough coded packets. In this method, the receivers do not have to concern packet loss rate per path.

B. Overview of Inter-Stream FEC

The above mentioned studies provide resilience against packet loss by properly processing each unicast/multicast stream. On the other hand, those methods require complex calculation for individual stream to deal with fluctuations of network condition rapidly. Our previous proposed method, “Inter-Stream FEC,” has a different concept of recovery [9]. Inter-Stream FEC improves the robustness against the change of local network condition by forming a protection group of streams, called “XOR group,” and protecting packets inside the group from packet loss. The XOR group is strategically organized so that the recovery probability exceeds a certain threshold. This method makes parity packets by XOR operations of streams’ data within the XOR group, instead of

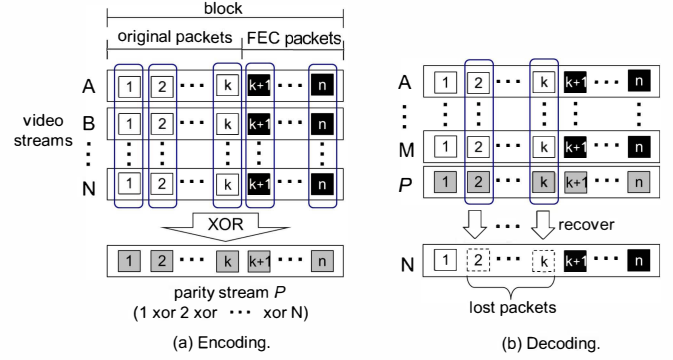


Fig. 1. Illustration of encoding/decoding of Inter-Stream FEC.

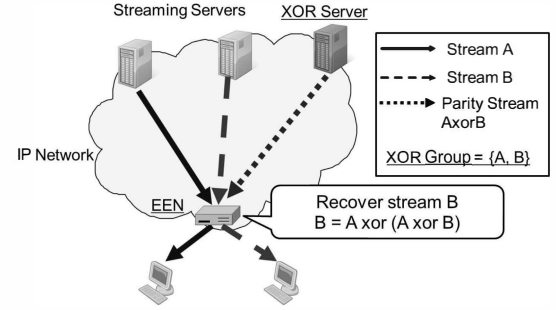


Fig. 2. Illustration of the Inter-Stream FEC. The EEN recovers stream B with Inter-Stream FEC. Parity stream is encoded by an XOR server.

adjusting redundancy per stream (Fig. 1 (a)). This policy enables the parity packets to recover a heavily damaged stream with other well-conditioned streams (Fig. 1 (b)). In particular, this policy effectively works when these streams are sent through fully disjoint paths. Moreover, the method can be implemented independently of other FEC methods which are processed per stream.

Figure 2 illustrates the overview of Inter-Stream FEC. The system deploys the “Enhanced Edge Nodes (EENs)” and “XOR servers” in addition to streaming servers. An EEN is the edge node having user management and packet recovery functions. We assume that EENs are located nearby users (e.g., hotel or condominium apartment) by a content provider for improving video quality. A streaming server sends the requested content by UDP transmission. The corresponding client receives its data through the EEN. In addition, when some forwarded streams suffer from packet loss, the EEN requests an XOR server to send it a parity stream in order to recover the lost packets (see Fig. 2). Thus, this method leverages the path diversity derived from the multiple servers in a group. The EEN can recover lost packets as long as multiple streams do not simultaneously suffer from heavy packet loss. The details are described in [9].

III. HIGH QUALITY STREAMING SYSTEM WITH HIERARCHICAL CACHE SERVERS BASED ON INTER-STREAM FEC FUNCTION

A. Motivation

For providing high quality video streaming service to many users, it is effective to organize cache servers in a hierarchical

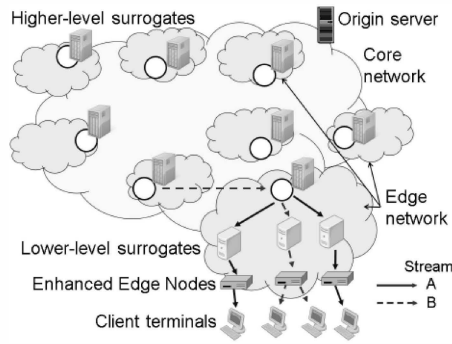


Fig. 3. The proposed hierarchical CDN system.

manner. This structure enables a content provider to efficiently use server resources, localize video traffic and improve servers' response time. For the above advantage, this paper designs a streaming system on the basis of hierarchical cache servers.

In streaming systems with hierarchical cache servers, more popular content is cached at servers nearer clients. This means that the popular content is delivered to clients with no or very few lost packets. Therefore, traditional intra-stream FEC can completely recover the packet loss in most cases for the popular content. On the other hand, less-popular content suffers from higher packet loss rate since such content tends to be delivered to clients from distant servers via many hops. Whereas the packet loss rate could exceed the recovery capability of intra-stream FEC, Inter-Stream FEC can effectively recover these lost packets leveraging path diversity. Thus, in this paper, we propose to protect less-popular content with Inter-Stream FEC.

A proper parity stream construction considering cache status of individual cache server is also necessary in order to improve the recovery capability. This is because a cache server does not cache all of less-popular contents due to limited cache capacity. That is, it is hard for EENs to find the cache server which has all the required contents to construct a parity stream (i.e., copies of member streams' videos within an XOR group). In other words, the simple XOR grouping (e.g., the previous approach in Ref. [9]) is insufficient to construct appropriate parity streams in the focused environment.

B. Streaming System with Inter-Stream Function

Figure 3 shows an overview of the proposed hierarchical CDN system. The proposed system has two types of cache servers. First type of cache servers are named "higher-level surrogates." These servers aggregate the content requests which are located at the core network. Second type of cache servers are named "lower-level surrogates." These servers are located at edge network to enhance distribution performance for local users. Popular content (e.g., stream A) is inherently cached in lower-level surrogates. On the other hand, less-popular content (e.g., stream B) tends to be sparsely cached in higher-level surrogates.

When a client requests a stream, the request message is transmitted to an EEN. First, the EEN asks nearby lower-level

surrogates whether they have the requested content. In the case that some lower-level surrogates have the content, the EEN receives the content from the nearest (or least-loaded) surrogate among them and forwards it to the requesting client. Otherwise, the EEN asks the nearest lower-level surrogate to request the content from higher-level surrogates. When the lower-level surrogate receives video packets from a higher-level surrogate having the requested content, the lower-level surrogate caches video data within the packets and forwards the data to the client via the EEN. When no higher-level surrogate has the requested content, the higher-level surrogate that is the nearest to the lower-level surrogate caches the streamed content from origin server and simultaneously forwards it.

This paper mainly focuses on protection of less-popular content with Inter-Stream FEC as mentioned in Section III-A. In most cases, less-popular contents are cached at only higher-level surrogates, whereas popular contents are cached at the lower-level ones, to maximize total users' benefit (e.g. quick response) to their requests. Therefore, higher-level surrogates need the additional function of XOR servers for less-popular content distribution. This is also feasible from the perspective of transmission and/or CPU load of surrogates. Higher-level surrogates can afford making parity stream, since most of the requests from clients are accepted by lower-level surrogates.

C. Issue of XOR Grouping

In order to improve the recovery capability and the redundancy of parity stream, XOR group should be properly organized. We have proposed an optimum XOR grouping method in [9]. However, this optimum method cannot be directly used for rarely cached contents such as less-popular ones. This is why at least one surrogate must have all the copies of member streams' videos within an XOR group.

Therefore, in XOR grouping for less-popular contents, an XOR group should be organized based on cache status of individual higher-level surrogate. In addition, in order to avoid degradation of recovery capability caused by loss of parity streams, an EEN has to appropriately select higher-level surrogates as XOR servers. Considering the close relationship between XOR server selection and XOR grouping, we propose a parity stream request method combined with XOR server selection.

D. Parity Stream Request Method

When an EEN observes that packet loss rates of some forwarding streams increase and these rates are likely to exceed the recovery capability of intra-stream FEC, the EEN decides to organize XOR groups based on the packet loss rates. In this paper, we suppose that the EEN decides to request parity streams when packet loss rates of some streams become higher than threshold Th_{plr} .

The proposed parity stream request method is composed of three phases; selection of XOR server candidates, selection of XOR group candidates, and parity stream request based on XOR server/XOR group candidates. In the first phase, the EEN makes a candidate list of higher-level surrogates which

TABLE I
NOTATIONS IN THE PARITY STREAM REQUEST METHOD.

$\mathcal{V} = \{v_p \mid 0 \leq p \leq P\}$	Set of all video streams forwarded to underlying clients by EEN
v_p	Video stream (id p) of \mathcal{V}
plr_p	Packet loss rate of stream v_p
$\mathcal{RT} = \{v_p \in \mathcal{V} \mid plr_p > Th_{plr}\}$	Recovery target streams
$\mathcal{S} = \{hs-i \mid 0 \leq i \leq I\}$	Set of all higher-level surrogates
$hs-i$	Higher-level surrogate (id i) of \mathcal{S}
$\mathcal{X}(r) \subseteq \mathcal{S}$	Candidates for XOR servers with Rank r selected in Phase 1
$\mathcal{G}(i) = \{\mathcal{G}(i, q) \mid 0 \leq q \leq Q\}$	XOR group candidates of $hs-i$ organized in Phase 2
$\mathcal{G}(i, q) \subseteq \mathcal{V}$	XOR group candidate (id q) of $\mathcal{G}(i)$
\mathcal{A}	Set of XOR groups selected as the solution in Phase 3

should operate as XOR servers. These candidates are ranked into some classes based on expected possibility of packet loss. Ranking the candidates into some classes intends to make the following process fast and easy by localizing search space. In the second phase, the EEN selects candidates for XOR groups. As mentioned in Section III-C, it is difficult to organize the optimum XOR groups from less-popular contents. Hence, the proposed method takes the approach that organizes the semi-optimum XOR groups based on cache status of each higher-level surrogate. Finally, in the third phase, the EEN selects a minimum set of XOR groups that can include all recovery target streams based on the candidates selected in the previous phases. Then, the EEN sends request message for parity streams to the corresponding higher-level surrogates according to the selected XOR groups.

In the following part, the process of each phase is described in detail. Table I shows the notations used in the following part. To help readers understand, we also show the relations between the elements of the proposed parity stream request method in Fig. 4. The rounded rectangle at the top of the figure represents XOR server candidates. Each XOR server candidate, represented by a rectangle, is classified based on its Rank. Each diamond at the center of the figure represents XOR group candidate. Each XOR group candidate is organized based on the cache status of the XOR server candidate connected by the solid arrow. The member streams of an XOR group candidate are represented by circles at the bottom of the figure connected with it by dotted arrows. The solution set of XOR groups, which is a set of XOR groups covering all recovery targets, is highlighted in the figure.

Phase 1: Selection of XOR Server Candidates

In order to raise recovery capability of Inter-Stream FEC, it is desirable to select XOR servers which can send parity streams with low packet loss rates. In general, when transmission paths of two senders have no shared link, packet loss patterns of these paths are different from each other. Therefore, when video streams sent from a higher-level surrogate to an EEN suffer from packet loss, the EEN should select XOR servers whose transmission paths include no shared link with the degraded path. The distance from a sender to a receiver also affects packet loss rate. Thus, as the index of the possibility of packet loss, we assign “Rank” to each surrogate

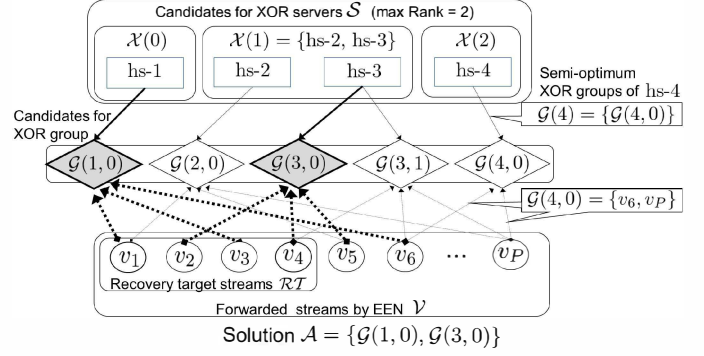


Fig. 4. Relations between elements of the proposed request method.

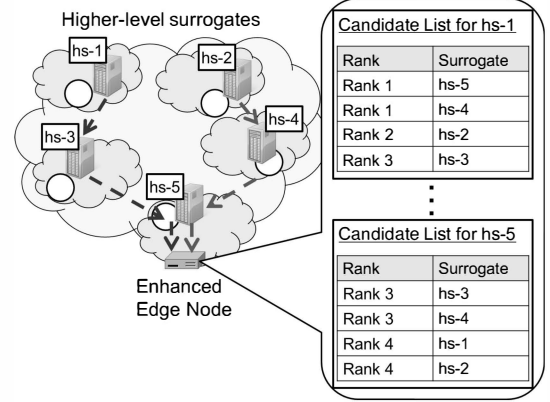


Fig. 5. Illustration of candidate list. An EEN has a candidate list for each higher-level surrogate $hs-x$.

pair based on path disjointness and Round Trip Time (RTT).

In the following, we focus on one EEN (EEN_k) and higher-level surrogates \mathcal{S} . Let $Path(i, k)$ be the path from $hs-i$ to EEN_k , and $RTT(i, k)$ be the RTT of $Path(i, k)$. We also define path disjointness of $Path(j, k)$ against $Path(i, k)$, $D_{(i,j)}^{(k)}$, $i \neq j$, as Eq. (1).

$$D_{(i,j)}^{(k)} = 1 - \frac{\text{\# of shared links b/w } Path(i, k) \text{ and } Path(j, k)}{\text{\# of links included in } Path(i, k)} \quad (1)$$

This index means that $Path(j, k)$ with larger $D_{(i,j)}^{(k)}$ is affected with less probability from quality degradation of $Path(i, k)$.

Path disjointness $D_{(i,j)}^{(k)}$ is calculated by the EEN_k . Shared links are estimated by each EEN using *traceroute* in the same way as [17]. The estimation is done in a cycle of one minute. At this time, RTT is updated using the *traceroute* messages.

Under the above definition, higher-level surrogates are ranked into four classes based on the following conditions, when packet loss rate on $Path(i, k)$ exceeds Th_{plr} .

Rank 1 $D_{(i,j)}^{(k)} > Th_{disjoint}$ and $RTT(j, k) < Th_{RTT}$

Rank 2 $D_{(i,j)}^{(k)} > Th_{disjoint}$ and $RTT(j, k) \geq Th_{RTT}$

Rank 3 $D_{(i,j)}^{(k)} \leq Th_{disjoint}$ and $RTT(j, k) < Th_{RTT}$

Rank 4 $D_{(i,j)}^{(k)} \leq Th_{disjoint}$ and $RTT(j, k) \geq Th_{RTT}$

Higher-level surrogates within Rank 1 are expected to be able to send parity streams without packet loss. Rank 4 has

the opposite characteristics. Therefore, the proposed method deals with higher-level surrogates within Rank 1 as primary candidates for XOR servers. After selecting Rank 1, Rank 2, 3, 4 are selected in order.

The values $D_{(i,j)}^{(k)}$ and $RTT(j, k)$ can be calculated before the parity stream request process. In this paper, the proposed system makes a candidate list for individual higher-level surrogate in advance (See Fig. 5). Therefore, in Phase 1, the EEN has only to refer the candidate list for $hs-i$ to select XOR server candidates.

Phase 2: Selection of XOR Group Candidates

In this phase, the EEN organizes the semi-optimum XOR groups based on cache status of each higher-level surrogate. Its policy is simple. This per-surrogate XOR grouping follows the XOR grouping method in [9]. This method organizes an XOR group so that the expected recovery probability of the recovery targets satisfies target quality.

When recovery target stream $v_p \in RT$ is a member of XOR group $\mathcal{G}(j, q)$, the parity stream generated from $\mathcal{G}(j, q)$ is expected to recover v_p with probability $P^{rec}(v_p, \mathcal{G}(j, q))$, which is calculated by Eq. (2).

$$P^{rec}(v_p, \mathcal{G}(j, q)) = (1 - plr_{parity}) \times \prod_{v_u \in \{v_p\} \setminus \mathcal{G}(j, q)} (1 - plr_u). \quad (2)$$

In Eq. (2), plr_{parity} represents the expected packet loss rate of the parity stream generated from $\mathcal{G}(j, q)$. Note that we assume $plr_{parity} = 0.0$, however, plr_{parity} could be estimated from current and/or past records of packet loss rate on $Path(j, k)$. Equation (2) expresses the probability that both the parity stream and all the other streams except v_p are correctly received. Using Eq. (2), the probability that v_p is correctly received or recovered by the parity stream, $P^{cor}(v_p, \mathcal{G}(j, q))$, is shown in Eq. (3).

$$P^{cor}(v_p, \mathcal{G}(j, q)) = (1 - plr_p) + plr_p \times P^{rec}(v_p, \mathcal{G}(j, q)). \quad (3)$$

In order to improve efficiency, this method organizes XOR group $\mathcal{G}(j, q)$ from the maximal number of video streams under the condition $P^{cor}(v_p, \mathcal{G}(j, q)) > (1 - Th_{plr})$ for all $v_p \in \mathcal{G}(j, q)$.

Phase 3: Request of Parity Streams

In this phase, the EEN selects XOR groups to properly request parity streams. The basic policy is to preferentially select XOR group candidates from among XOR server candidates within higher Rank. In addition to this policy, the EEN tries to select a minimum set of XOR groups.

Figure 6 shows the pseudo code of the parity stream request method. The proposed method solves the minimum set of XOR groups in a greedy manner as shown in Algorithm 1. First, the EEN checks XOR group candidates among XOR server candidates with Rank 1, i.e., $\mathcal{H} = \{\mathcal{G}(i) \mid hs-i \in \mathcal{X}(1)\}$. Next, the EEN selects the XOR group that includes the largest number of recovery targets from \mathcal{H} . Then, the EEN adds the selected XOR group into the solution \mathcal{A} and removes the included recovery targets from \mathcal{RT} . When no candidate within

Algorithm 1 Selection of minimum set of XOR groups

Input: \mathcal{V}, \mathcal{S}
Output: \mathcal{A}

- 1: Find $hs-i \in \mathcal{S}$ from which packet loss rate of streams $> Th_{PLR}$
- 2: $\mathcal{RT} = \{v_p \in \mathcal{V} \mid \text{packet loss rate of } v_p > Th_{PLR}\}$
- 3: Calculate $\mathcal{X}(r), r = 0, 1, \dots, \text{max Rank from } \mathcal{S}$ //Phase 1
- 4: Calculate $\mathcal{G}(j), j \neq i$ from \mathcal{V} //Phase 2
- 5: //Procedure in Phase 3
- 6: $\mathcal{A} = \phi$ // Solution of parity stream request
- 7: for $r = 0$ to max Rank do
- 8: //Pick up XOR group candidates based on surrogates' Rank
- 9: $\mathcal{H} = \phi$
- 10: for all $hs-j \in \mathcal{X}(r)$ do
- 11: $\mathcal{H} = \mathcal{H} \cup \mathcal{G}(j)$
- 12: end for
- 13: while $\mathcal{H} \neq \phi$ do
- 14: //Add the XOR group candidate which can cover the most recovery targets into solution
- 15: Select $\mathcal{G}(j, q)$ which has max $card(\mathcal{C})$, where $\mathcal{C} = \mathcal{G}(j, q) \cap \mathcal{RT}$
- 16: if $card(\mathcal{C}) == 0$ then
- 17: break //Check the next Rank
- 18: end if
- 19: $\mathcal{A} = \mathcal{A} \cup \mathcal{G}(j, q)$
- 20: $\mathcal{H} = \mathcal{H} \cap \neg \mathcal{G}(j, q)$
- 21: $\mathcal{RT} = \mathcal{RT} \cap \neg \mathcal{C}$
- 22: if $\mathcal{RT} == \phi$ then
- 23: return \mathcal{A}
- 24: end if
- 25: end while
- 26: end for
- 27: if $\mathcal{RT} \neq \phi$ then
- 28: //When some recovery targets are not covered, the EEN organizes them into XOR groups for origin server.
- 29: Organize \mathcal{RT} into $\mathcal{G}(o, q)$, where o represents origin server
- 30: $\mathcal{A} = \mathcal{A} \cup \mathcal{G}(o, q)$, where $\mathcal{G}(o, q) = \{\mathcal{G}(o, q)\}$
- 31: end if
- 32: return \mathcal{A}

Algorithm 2 Request of parity streams

Input: \mathcal{A}

- 1: for all $\mathcal{G}(i, q)$ (or $\mathcal{G}(o, q)$) $\in \mathcal{A}$ do
- 2: Request $hs-i$ (origin server) to send the parity stream encoded from $\mathcal{G}(i, q)$ ($\mathcal{G}(o, q)$)
- 3: end for

Fig. 6. Procedure of parity stream request method.

\mathcal{H} includes any recovery target within $\mathcal{RT} \neq \phi$, the EEN checks XOR group candidates among XOR server candidates with next Rank, i.e., $\mathcal{H} = \{\mathcal{G}(i) \mid hs-i \in \mathcal{X}(2)\}$. Until all the recovery targets are included into XOR groups within the solution \mathcal{A} , the EEN repeats the above process for each Rank in ascending order. When no XOR group includes some recovery targets, the EEN organizes the remained streams into new XOR groups of origin server, and adds them into the solution \mathcal{A} . After the selection of XOR groups, the EEN requests the corresponding higher-level surrogates (or origin server) to send the parity streams based on the solution \mathcal{A} as shown in Algorithm 2.

IV. PERFORMANCE EVALUATION

A. Simulation Model

We evaluate the proposed system through computer simulations. In the simulations, we use USA topology with 28 nodes

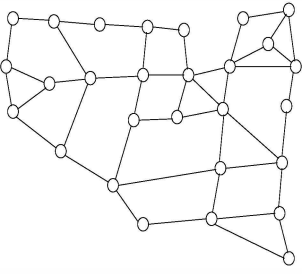


Fig. 7. USA Topology.

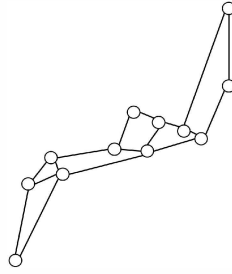


Fig. 8. Japan Topology.

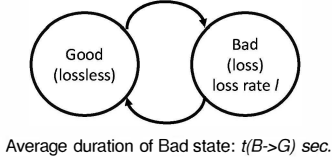
Average duration of Good state: $t_{G \rightarrow B}$ sec.Average duration of Bad state: $t_{B \rightarrow G}$ sec.

Fig. 9. Packet loss model.

and 45 links (Fig. 7), and Japan topology with 12 nodes and 17 links (Fig. 8). In USA topology, an EEN can relatively easily compute XOR servers from which the EEN should request parity streams by the proposed request method. This is because many nodes are densely connected each other in USA topology. On the other hand, Japan topology consists of a few nodes which are sparsely connected. In such case, content is rarely cached in surrogates and a lot of transmission paths tend to share the same link. Hence, we use Japan topology as an environment where it is difficult for the EEN to find proper XOR servers. We focus on the protection for less-popular contents in the evaluations. Therefore, for simplicity, the simulations consider one EEN, higher-level surrogates and one origin server. The EEN and origin server are connected with randomly selected nodes. Higher-level surrogates are connected with each node respectively. Since an EEN is located nearby clients, packet loss rarely occurs between the EEN and its managing clients. Therefore, this model supposes the EEN as virtual client receiving multiple streams and does not take care of client terminals. Note that each stream is protected by intra-stream FEC with fixed redundancy.

In each link of the topologies, packets are lost with different pattern. In this model, we generate bursty packet loss on each link according to the following two-state transition model (See Fig. 9). When a packet is transmitted in *Good* state, no packet is lost. On the contrary, in *Bad* state, packets are lost with a probability of l ($0 \leq l \leq 1$). The average duration of each state follows exponential distribution with mean $t_{G \rightarrow B}$ and $t_{B \rightarrow G}$ seconds, respectively. After this duration, this model transits to the opposite state.

The common conditions are shown in Table II. Table III shows the default values of parameters in the evaluations.

B. Numerical Results

Under the above environment, we evaluate the proposed system in comparison with the system using conventional intra-stream FEC (Compared system). We as-

TABLE II
COMMON CONDITIONS.

name	value
Transmission protocol	UDP
Packet size	1500 [Bytes]
Transmission bit rate	2.4 [Mbps]
Link delay	100 [ms]
Request interval for parity stream	1.0 [s]
The number of less-popular contents	100
The probability that content is cached in a surrogate	0.2
The number of recovery target streams	60
FEC redundancy (n, k)	(165, 150)
Threshold of packet loss rate for parity stream request Th_{plr}	0.05

TABLE III
DEFAULT VALUE OF PARAMETERS IN THE EVALUATIONS.

name	value
Threshold of path disjointness	0.5
for surrogate ranking $Th_{disjoint}$	
Threshold of RTT for surrogate ranking Th_{RTT}	300 [ms]
Packet loss rate in <i>Bad</i> state l	0.2
Average duration in <i>Good</i> state $t_{G \rightarrow B}$	500 [s]
Average duration in <i>Bad</i> state $t_{B \rightarrow G}$	10 [s]

signed the additional FEC packets per block in the compared system, so that the redundancy is basically set to as the similar quantity as the proposed method. Specifically, the number of additional FEC packets is calculated by $\lceil Red_{prop} / \text{original packets per block } (k) \rceil$, where Red_{prop} represents whole parity packets generated by the Proposed system. Note that the Inter-Stream FEC system in [9] assumes that surrogates have data of all contents. Since the assumption is unsuitable in the target environment, we exclude the system in this paper.

Figure 10 shows the final reception ratio, which means the reception ratio after recovery process of Inter-/intra-stream FEC, in USA topology. The final reception ratio is represented by the ratio of received or recovered packets to whole original packets. The “No recovery” in Fig. 10 shows the reception ratio in the case that no recovery method is applied. In Fig. 10, final reception ratios of all methods are degraded with the increase of packet loss rate in *Bad* state l . However, the proposed system greatly improves its final reception ratio compared with the others even when packet loss rate l is high. This is because the proposed system can recover bursty lost packets of one stream using other correctly received streams. On the other hand, the final reception ratio of the compared system drastically decreases when the packet loss rate l is over 0.12, where lost packets exceed the recovery capability of intra-stream FEC. This means that recovery capability of intra-stream FEC becomes increasingly insufficient with the increase of packet loss rate l , when the assigned redundancy is similar level with the proposed system. Figure 11 shows the final reception ratio in Japan topology. In Japan topology, final reception ratio of each method shows similar characteristic as that in USA topology. From these results, we can conclude the proposed system can perform high recovery capability even in small and sparse topology.

Figure 12 shows the redundancy of each system in USA

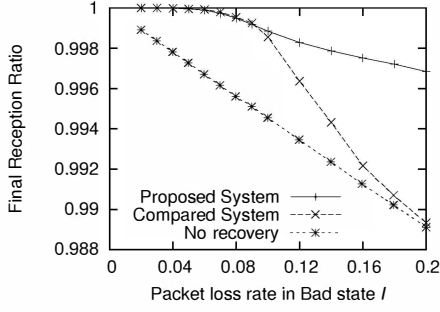


Fig. 10. Reception ratio after recovery process in USA topology.

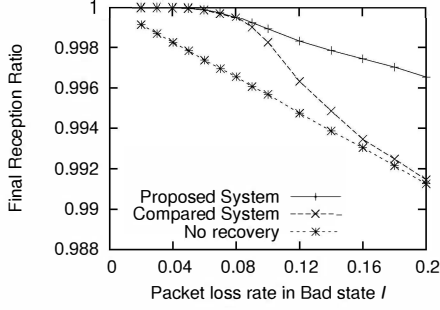


Fig. 11. Reception ratio after recovery process in Japan topology.

topology. The redundancy is represented by the ratio of parity packets generated by Inter-Stream FEC (or sum of additional FEC packets in the compared system) to whole original packets. Note that the parity data of the compared system is set to be fixed and slightly larger value than that of the proposed system. Therefore, the redundancy of the compared system is always larger than that of the proposed system, and it shows step-wise behavior. Obviously, the redundancy of the proposed system increases with the increase of packet loss rate in *Bad state l*. However, the redundancy is significantly small, because the proposed system appropriately adjusts the amount of parity streams according to the network condition. On the other hand, the compared system cannot sufficiently recover lost packets with the similar redundancy, especially, when packet loss rate is high. From Fig. 13, redundancy characteristics in Japan topology is also similar as that in USA topology. Note that, in Japan topology, network-wide probability that bursty packet loss occurs is less than USA topology, because bursty packet loss independently occurs per link in the simulations. Therefore, the proposed system generates fewer number of parity streams in Japan topology, which has less links than USA topology. These results show that the proposed system can achieve high recovery capability with little redundancy, especially, when bursty packet loss occurs such that the compared system cannot recover them.

We also evaluate about the impact of two thresholds; $Th_{disjoint}$ and Th_{RTT} , in both topologies. In these evaluations, packet loss rate in *Bad state l* is set to 0.2.

Tables IV and V show the results under various thresholds about path disjointness in USA and Japan topology, respectively. Threshold Th_{RTT} is set to 600 ms. Each column represents average values of final reception ratio, redundancy,

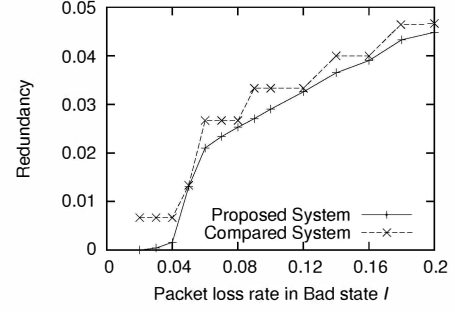


Fig. 12. Redundancy of each recovery method in USA topology.

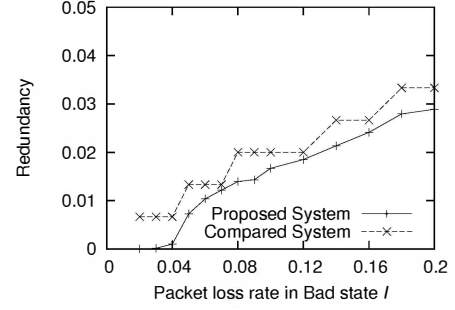


Fig. 13. Redundancy of each recovery method in Japan topology.

path disjointness of selected XOR servers $D_{(i,j)}^{(k)}$, RTT of selected XOR servers $RTT(j, k)$ and Packet loss rate of parity streams, respectively. Table IV shows that path disjointness of XOR servers increases as $Th_{disjoint}$ increases, except when $Th_{disjoint}$ is 1.0. When $Th_{disjoint}$ is 1.0, the EEN selects XOR servers which can send the minimum number of parity streams without considering path disjointness. Hence, path disjointness becomes larger and redundancy becomes fewer in this case. In the other cases, surrogates with higher path disjointness than $Th_{disjoint}$ are preferentially selected as XOR servers. As a result, path disjointness of XOR servers increases with the increase of $Th_{disjoint}$. From this table, we also see that the EEN can improve packet loss rate of parity streams by proper selection of $Th_{disjoint}$. On the other hand, Table V shows that the performance of the proposed system keeps at the same level regardless of $Th_{disjoint}$ in Japan topology. This is because that the EEN is forced to select XOR servers with smaller path disjointness by the limitation of available surrogates and transmission paths of Japan topology, even when $Th_{disjoint}$ is set to large value. Moreover, when compared with Table IV, Table V shows that parity streams tend to suffer from packet loss more frequently in Japan topology due to its smaller path disjointness.

Tables VI and VII show the results under various thresholds about RTT in USA and Japan topology, respectively. Threshold $Th_{disjoint}$ is set to 0.5. Table VI shows that the EEN can select nearer surrogates as XOR servers using proper threshold Th_{RTT} in USA topology. In the simulation, the performance of the proposed system becomes highest when Th_{RTT} is 600 ms. On the other hand, Table VII shows that the performance of the proposed system also keeps at the same level regardless of Th_{RTT} , similarly as the case of Table V. This is caused

TABLE IV

IMPACT OF THRESHOLD OF PATH DISJOINTNESS IN USA TOPOLOGY.

$Th_{disjoint}$	final reception ratio	redundancy	path disjointness	RTT (ms)	plr of parity streams
0.0	0.997088	0.0395	0.769	484	0.0187
0.2	0.997085	0.0395	0.769	483	0.0191
0.4	0.997121	0.0401	0.786	485	0.0174
0.6	0.997114	0.0404	0.803	487	0.0173
0.8	0.997119	0.0413	0.812	489	0.0185
1.0	0.996961	0.0393	0.726	480	0.0244

TABLE V

IMPACT OF THRESHOLD OF PATH DISJOINTNESS IN JAPAN TOPOLOGY.

$Th_{disjoint}$	final reception ratio	redundancy	path disjointness	RTT (ms)	plr of parity streams
0.0	0.996545	0.0292	0.579	496	0.0436
0.2	0.996545	0.0292	0.579	496	0.0436
0.4	0.996536	0.0289	0.578	497	0.0441
0.6	0.996538	0.0290	0.581	498	0.0444
0.8	0.996566	0.0295	0.587	499	0.0436
1.0	0.996534	0.0286	0.551	497	0.0471

TABLE VI

IMPACT OF THRESHOLD OF RTT IN USA TOPOLOGY.

Th_{RTT} (ms)	final reception ratio	redundancy	path disjointness	RTT (ms)	plr of parity streams
0	0.996952	0.0382	0.788	619	0.0188
200	0.997011	0.0389	0.797	573	0.0183
400	0.997092	0.0406	0.794	509	0.0175
600	0.997107	0.0405	0.803	486	0.0171
800	0.997071	0.0403	0.791	535	0.0173
1000	0.997034	0.0388	0.786	582	0.0186

TABLE VII

IMPACT OF THRESHOLD OF RTT IN JAPAN TOPOLOGY.

Th_{RTT} (ms)	final reception ratio	redundancy	path disjointness	RTT (ms)	plr of parity streams
0	0.996532	0.0285	0.576	514	0.0450
200	0.996532	0.0287	0.579	508	0.0443
400	0.996568	0.0290	0.583	497	0.0438
600	0.996538	0.0290	0.581	498	0.0444
800	0.996552	0.0291	0.579	511	0.0444
1000	0.996532	0.0285	0.576	514	0.0450

by the same reason as the evaluation about $Th_{disjoint}$.

V. CONCLUSION

In this paper, in order to establish seamless and highly robust contents distribution technology, we proposed a hierarchical CDN system with practical Inter-Stream FEC function. The proposed system combines parity stream request method with XOR server selection considering two metrics: disjointness between path to the EEN from one surrogate and that from the XOR servers, and Round Trip Times (RTTs) between XOR servers and the EEN. Considering these metrics improved recovery capability of Inter-Stream FEC. In addition, our proposed method properly organizes XOR groups according to the cache status of surrogates, in order to reduce overall traffic. Therefore, the proposed system can provide high recovery capability and redundancy efficiency even in the environment that optimal parity streams cannot be constructed. We demonstrated that the proposed system can achieve higher restoration performance compared with traditional FEC method through computer simulation. As future works, the effectiveness of the proposed method to video quality will be evaluated by some indexes such as Peak Signal to Noise Ratio (PSNR). Since the proposed method has certain level of dependency on cache status of surrogates, the proposed method should cooperate with the contents cache method which is suitable for Inter-Stream FEC.

REFERENCES

- [1] G. Peng, "CDN: Content distribution network," *Arxiv preprint cs/0411069*, pp. 1–26, 2004.
- [2] R. Doverspike, G. Li, K. N. Oikonomou, K. Ramakrishnan, R. K. Sinha, D. Wang, and C. Chase, "Designing a reliable IPTV network," *IEEE Internet Computing*, vol. 13, no. 3, pp. 15–22, May 2009.
- [3] H. Che, Z. Wang, and Y. Tung, "Analysis and design of hierarchical Web caching systems," in *Proceedings of IEEE INFOCOM 2001*, vol. 3, pp. 1416–1424, Apr. 2001.
- [4] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li, "Design and deployment of a hybrid CDN-P2P system for live video streaming: experiences with LiveSky," in *Proceedings of the 17th ACM International Conference on Multimedia*, pp. 25–34, Oct. 2009.
- [5] L. Chen, M. Meo, and A. Scicchitano, "Caching video contents in IPTV systems with hierarchical architecture," in *Proceedings of IEEE ICC 2009*, 6 pages, June 2009.
- [6] D. Kliazovich, N. Ben Halima, and F. Granelli, "Context-aware receiver-driven retransmission control in wireless local area networks," in *Proceedings of IEEE ICC 2009*, 5 pages, June 2009.
- [7] Z. Li, J. Chakareski, L. Shen, and L. Wang, "Video quality in transmission over burst-loss channels : a forward error correction perspective," *IEEE Communications Letters*, vol. 15, no. 2, pp. 238–240, Feb. 2011.
- [8] M. Wu, S. Makyharia, H. Liu, D. Li, and S. Mathur, "IPTV multicast over wireless lan using merged hybrid ARQ with staggered adaptive FEC," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 363–374, June 2009.
- [9] H. Tode, T. Fujie, and K. Murakami, "Inter-Stream FEC method for multiserver content distribution," in *Proceedings of IEEE GLOBECOM 2007*, pp. 2043–2047, Nov. 2007.
- [10] A. Fujimoto, Y. Hirota, H. Tode and K. Murakami, "Parity stream construction method based on estimated utility for multi-server video streaming system," in *Proceedings of IEEE ICC 2011*, 5 pages, June 2011.
- [11] D. Jurca, P. Frossard, and A. Jovanovic, "Forward error correction for multipath media streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 9, pp. 1315–1326, Sep. 2009.
- [12] M.-F. Tsai, N. K. Chilamkurti, S. Zeadally, and A. Vinel, "Concurrent multipath transmission combining forward error correction and path interleaving for video streaming," *Elsevier Computer Communications*, vol. 34, no. 9, pp. 1125–1136, June 2011.
- [13] H. Zhang, J. Zhou, and J. Li, "M2FEC: An effective FEC based multipath transmission scheme for interactive multimedia communication," *Journal of Visual Communication and Image Representation*, vol. 21, no. 2, pp. 120–128, Feb. 2010.
- [14] J. Chakareski and B. Girod, "Server diversity in rate-distortion optimized media streaming," in *Proceedings of ICIP 2003*, vol. 3, pp. 645–648, Sep. 2003.
- [15] J. Apostolopoulos, T. Wong, and S. Wee, "On multiple description streaming with content delivery networks," in *Proceedings of IEEE INFOCOM 2002*, vol. 3, pp. 1736–1745, June 2002.
- [16] H. Wang and C.-C. J. Kuo, "Robust video multicast with joint network coding and video interleaving," *Journal of Visual Communication and Image Representation*, vol. 21, no. 2, pp. 77–88, Feb. 2010.
- [17] T. Nguyen and A. Zakhori, "Path diversity with forward error correction (pdf) system for packet switched networks," in *Proceedings of IEEE INFOCOM 2003*, vol. 1, pp. 663–672, Apr. 2003.