A Robust Luby Transform Encoding Pattern-Aware Symbol Packetization Algorithm for Video Streaming Over Wireless Network

Dongju Lee and Hwangjun Song

***Abstract*—In this paper, we propose a Luby transform encoding pattern-aware symbol packetization algorithm to minimize the quality degradation of video streaming service caused by packet losses over wireless network. To achieve this goal, the relationship among Luby transform encoded symbols is analyzed based on Luby transform encoding pattern, and the proposed packetization algorithm is designed to minimize packet loss effects by reducing the dependency among packets conveying Luby transform encoded symbols. Finally, experimental results are provided to show the performance of the proposed algorithm**

***Index Terms*—LT encoding pattern, Luby transform codes, packetization algorithm, video streaming, wireless network.**

I. Introduction

VIDEO streaming has been regarded as one of the key applications of multimedia technologies among both network and video coding experts [1]. It is, however, a difficult problem to transmit video streaming traffic over the network because video contains a large amount of data and comes with strict quality-of-service (QoS) requirements. To reduce the amount of data, it is indispensable to employ effective video compression algorithms. So far, digital video coding techniques have advanced rapidly. International standards such as MPEG-1, 2 [2] and 4 [3], H.261, H.263/+/++ [4], H.264 [5], and H.265 [6] have been established or are under development to accommodate different needs by ISO/IEC and ITU-T, respectively. The characteristic of compressed video data can be controlled by rate control algorithm although its generic characteristic is variable-bit-rate due to entropy coder and scene change/inconstant motion change of the underlying video. Furthermore, it is very vulnerable to transmission errors and losses because of the extremely high compression ratio.

It is a more challenging problem to efficiently provide a stable video streaming service of high quality over wireless network. Large-scale path loss, small-scale fading, multi-path, and interference cause random variations in the received SNR in wireless links. Such variations increase the bit error rate (BER) because it is more difficult for the modulation scheme to decode the received signal in the case of lower SNR. To mitigate transmission errors and losses over wireless network, Automatic Repeat reQuest (ARQ) and forward error correction (FEC) are widely used. ARQ increases delay because it has to retransmit lost data after receiving feedback information. Thus, ARQ may not be suitable for delay sensitive video streaming. On the other hand, FEC requires some redundant data to compensate for errors and losses without any feedback information over the network. This feature is generally appropriate for delay sensitive video streaming.

Fountain codes [7] such as Luby transform (LT) [8], Raptor [9], and Online [10] codes are emerging erasure codes and block-based FEC schemes. The characteristics such as high coding efficiency, low encoding/decoding processing time, and flexibility are very useful for transmitting delay-sensitive data over error-prone wireless network. Actually, Raptor codes are standardized by 3GPP [11] and DVB-H [12] as an application layer forward error correction (AL-FEC) scheme. So far, many fountain code-related algorithms have been proposed in the literature to efficiently deliver multimedia data over network. In [13], Sanghavi proposed an effective algorithm to determine the degree distribution for partial recovery. In [14], Rahnavard *et al.* proposed a fountain code-based unequal error protection scheme by assigning more links to important source symbols. In [15], Woo *et al.* proposed an LT code-based unequal error protection scheme using encoded symbols with a low degree. In [16], Bogino *et al.* proposed a new fountain code scheme by means of a sliding window approach for multimedia streaming. This scheme provides a good performance in terms of efficiency, reliability, and memory utilization. In [17], Nguyen *et al.* proposed a systematic version of LT codes. In [18], Vukobratovic *et al.* proposed a solution for scalable video multicast based on unequal error protection expanding window fountain codes. In [19], Ahmad *et al.* proposed a flexible transmission framework for video streaming based on fountain codes and feedback information, and presented an efficient rate scheduling algorithm that aims at optimizing the transmission strategy. In [20], Gasiba *et al.* proposed efficient fountain decoding algorithms to improve the performance of MBMS video streaming. The proposed algorithms increase

Manuscript received September 03, 2010; revised December 17, 2010 and February 16, 2011; accepted February 19, 2011. Date of publication March 10, 2011; date of current version July 20, 2011. This work was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-(C1090-1111-0004)) and WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-2010-000-10100-0). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhu Liu.

The authors are with the Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Gyungbuk 790-784, Korea (e-mail: ldjblaze@postech.ac.kr; hwangjun@postech.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2011.2124448

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on July 19,2024 at 08:21:00 UTC from IEEE Xplore. Restrictions apply.

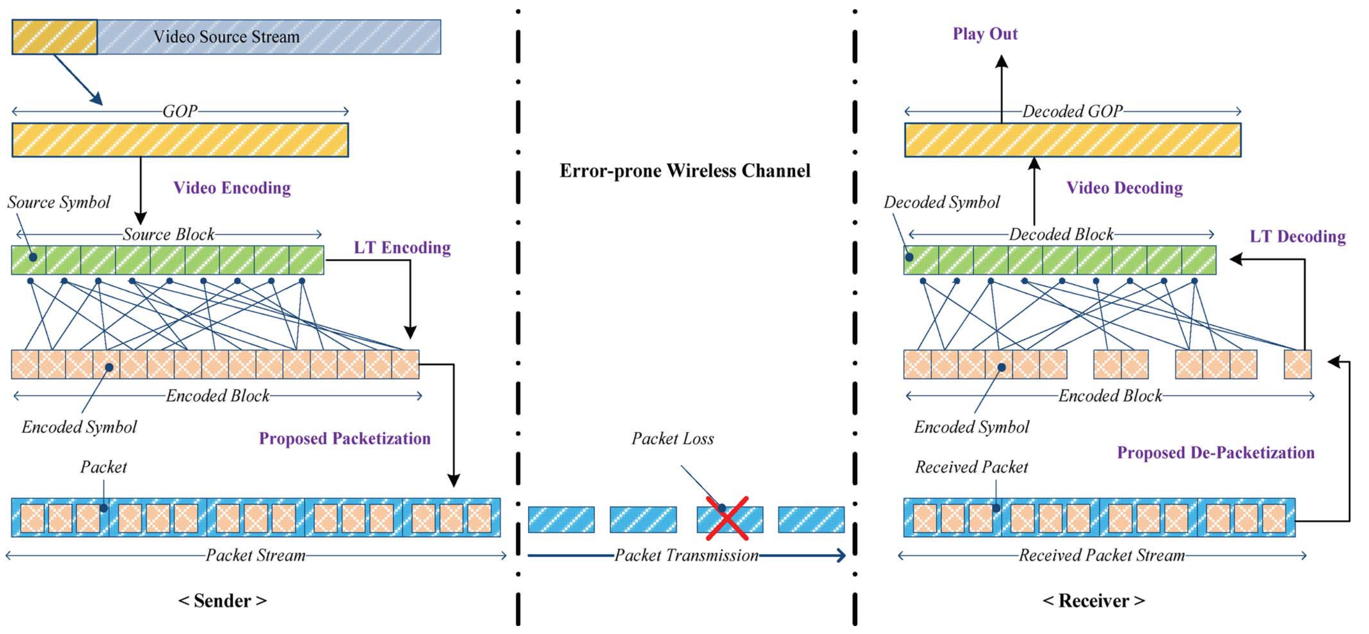


Fig. 1. Overall video streaming system architecture under our consideration.

the symbol recovery probability by using a significantly large number of partially corrupted UDP packets. In [21], Thomos *et al.* proposed network coding algorithms for video streaming in error-prone overlay networks. Basically fountain codes are used for robust transmission.

In this paper, we propose a robust LT encoding pattern-aware packetization algorithm of LT encoded symbols to support video streaming service over wireless network. LT codes are the first realization of a class of erasure codes that we call universal erasure codes [7], which are adopted in this paper due to their simple structure. The goal of the proposed algorithm is to effectively transmit LT encoded symbols over error-prone wireless network. First of all, the relationship among LT encoded symbols is analyzed based on LT encoding pattern. Based on the analysis, LT encoded symbols are packetized to reduce the correlation among packets so that the effect of a lost packet is locally limited. The paper is organized as follows. The details of the proposed packetization algorithm and its performance analysis are described in Section II, and experimental results are given in Section III. Finally, concluding remarks are given in Section IV.

II. Proposed LT Encoding Pattern-Aware Symbol Packetization Algorithm

Fig. 1 shows the overall video streaming system architecture under our consideration in error-prone wireless network. Compressed video data are generated by video codec to decrease the amount of data. The video data are LT encoded for robust transmission through the wireless network. During the LT encoding process, the compressed video data stream should be divided into source blocks. A source block is partitioned into source symbols of a predefined size. At this moment, an LT encoded symbol is mapped to a portion of the source symbols as shown in the Fig. 1, i.e., it is generated by performing bitwise XOR operations on the selected source symbols. The process is repeated until the last LT encoded symbol is generated. The mapping process is controlled by the generator polynomial, which is one of the most important design factors. Now, the LT encoded symbols should be packetized before the transmission because most of the state-of-the-art wireless networks support packet switching. Sometimes, packet losses during transmission are inevitable over error-prone wireless network. Thus, the number of the received LT encoded symbols may not always be enough for the successful decoding in time. If a large number of packets are continuously transmitted until the receiver obtains enough LT encoded symbols for successful LT decoding, error-free video streaming is guaranteed over the error-prone wireless network. Actually, it is very challenging to always guarantee successful LT decoding in time since video streaming is strictly delay-constrained and a wireless network has only very limited resources. To efficiently handle this problem, it is important to estimate how many packets (or additional encoded symbols) should be transmitted to deliver a sufficient number of encoded symbols to the receiver through the error-prone wireless network (it is called overhead in the following). Another essential problem is how to reduce the LT decoding failure rate with the same number of packets. In fact, LT decoding performance is related to how to packetize the encoded symbols because their dependency exists due to the iterative decoding process. Hence, the decoding performance may be seriously deteriorated when the dependency is not considered. For example, when the LT encoded symbols are randomly grouped into packets, a lost packet may hinder many source symbols from being successfully decoded. In general, video streaming data are very vulnerable to data losses since they are highly compressed to reduce the amount of data. Hence, a small amount of lost data can incur serious video quality degradation. To avoid disastrous loss, the relationship among LT encoded symbols is analyzed based on

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on July 19,2024 at 08:21:00 UTC from IEEE Xplore. Restrictions apply.

LT encoding pattern. Based on the analysis, the proposed packetization algorithm is designed to reduce the correlations among packets while increasing those among encoded symbols in each packet. As a result, the LT decoding success rate can be improved at the receiver by minimizing the effect of a lost packet. In other words, we can achieve the same LT decoding success rate with a smaller amount of overhead.

A. LT Encoding Pattern-Aware Symbol Relationship Analysis

We examine the relationship among LT encoded symbols to improve successful decoding probability. In recent years, some research effort has been devoted to analyzing the performance of fountain codes. In this paper, an And-Or tree [22] is adopted to investigate the LT encoding pattern. The symbol descriptions are summarized in Table I, and a simple example is given in Fig. 2. In the And-Or tree with a depth of $2l$, the depth of the root is zero and the depth of children directly connected to the root is 1. The depth of each node increases further down the tree. The nodes at even depths such as $0, 2, 4, \dots, 2l-2$ are called Or-nodes and those at odd depth such as $1, 3, 5, \dots, 2l-1$ are called And-nodes. Every node has a value of 0 or 1. The Or-nodes have the Or-operation value of their children (0 in the case of no children). The And-nodes have the And-operation value of their children (1 in the case of no children) [22]. The Or-nodes are source symbols while the And-nodes are encoded symbols. They are mapped to each other as parents and children. Since the root is an Or-node and T_i is an And-node, all source symbols in the $\mathbf{S}_{T_i}$ should be effectively decoded to obtain the root by using T_i regardless of the other target encoded symbols. Thus, the successful decoding of the root by using T_i is affected by encoded symbols at depth 3. That is, the elements of $\mathbf{S}_{T_i}$ should be decoded in advance to make T_i contribute to the successful decoding of the root. As shown in Fig. 2(b), the receiver can successfully decode more source symbols in $\mathbf{S}_{T_i}$ when T_i and more elements of $\mathbf{E}_{T_i}$ are available simultaneously. Accordingly, the decoding success probability of the root is improved. This means that T_i and the elements of $\mathbf{E}_{T_i}$ have a strong dependency on each other. Therefore, when T_i is lost during transmission, its children encoded symbols in $\mathbf{E}_{T_i}$ are disconnected from the root over the tree. Thus, they cannot contribute to the decoding of the root. To enhance the LT decoding success rate, this relationship must be considered during the packetization process. Based on the above observations, we can obtain the following useful properties.

1) Property 1: The decoding success rate of the root can be improved by assigning $\{T_i, 1 \le i \le N_T\}$ to different packets [23].

2) Property 2: The decoding success rate of the source symbols in $\mathbf{S}_{T_i}$ can be increased if more symbols of $\mathbf{E}_{T_i}$ are available [14].

3) Property 3: The decoding success rate of the root can be enhanced by packetizing T_i and the elements of its $\mathbf{E}_{T_i}$ into the same packet (a detailed analysis is given in Section II-C).

Incidentally, $\mathbf{E}_{T_i}(root_p) \cap \mathbf{E}_{T_j}(root_q)$ for $i \ne j$ or $p \ne q$ may not be a null set, where $\mathbf{E}_{T_i}(root_p)$ is a set of encoded symbols associated with T_i when a source symbol p is selected as a root and $\mathbf{E}_{T_j}(root_q)$ is a set of encoded symbols associated with T_i when a source symbol q is selected as a root. This means that some LT encoded symbols may be cross-listed in multiple sets. In this paper, the dependency among And-Or trees is implicitly considered by regulating the number of $\mathbf{E}_{T_i}$ put into the same packet with a T_i, and only the tree of depth 4 is considered to reduce the required computational complexity. To completely handle this dependency among trees during the packetization, more complicated tree analysis is required at the cost of higher computational complexity.

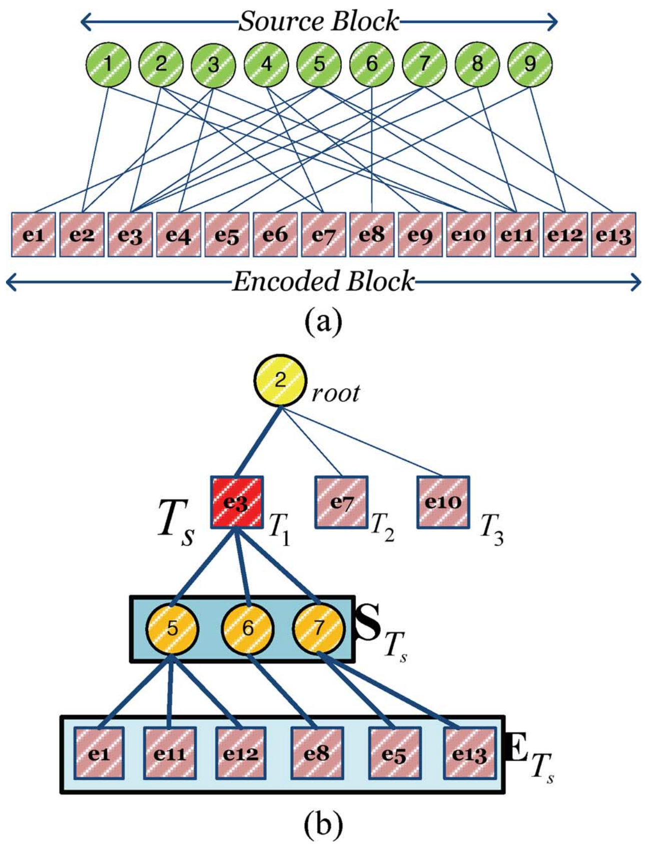


Fig. 2. Example of relationship analysis among LT encoded symbols by using the And-Or tree. (a) Mapping pattern among source symbols and encoded symbols. (b) Corresponding And-Or tree when source symbol #2 is selected as a root.

B. Proposed Packetization Algorithm of LT Encoded Symbols

T_s is a selected element of $\{T_i, 1 \le i \le N_T\}$ to make a contribution towards decoding the root. Based on the previous properties, the proposed packetization algorithm is designed to insert more elements of $\mathbf{E}_{T_s}$ and T_s into the same packet. $\mathbf{E}_{T_s}^{pkt}$ is defined as a subset of $\mathbf{E}_{T_s}$ whose elements are put into the same packet with T_s. The packet is called the target packet in the following. The number of elements of $\mathbf{E}_{T_s}$ that are packetized with T_s is determined by a control variable c_{thr} $(0 \le c_{thr} \le \min\{(n_{ps}-1), |\mathbf{E}_{T_s}|\})$ of the proposed packetization algorithm. When c_{thr} is set to 0, the proposed packetization algorithm is equivalent to the conventional packetization algorithm. Before the detailed description, it is assumed that n_{ps} is not less than 2. If n_{ps} is equal to 1, it means that only an LT encoded symbol is inserted into a packet. In this case, we do not need to consider the packetization effect. However, it is recommended in 3GPP MBMS [11] that a

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on July 19,2024 at 08:21:00 UTC from IEEE Xplore. Restrictions apply.

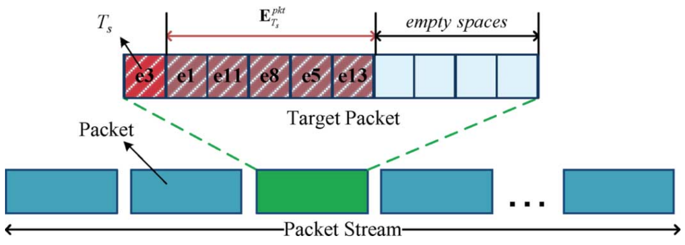


Fig. 3. Example of packetizing process when source symbol #2 and T_1 in Fig. 2 is selected as the root and T_s, respectively, with $n_{ps} = 10$ and $c_{thr} = 5$.

packet includes more than two encoded symbols for effective video streaming. In [20] and [24], it is shown that the smaller symbol size can generally increase decoding performance due to the increased number of source symbols. Thus, the above assumption ($n_{ps} \geq 2$) does not lose any generality. Now, the proposed packetization algorithm is summarized as follows.

Step 0: Select the first source symbol as a root.

Step 1: Select a T_s with $d_{T_s} \geq 2$ in $\{T_i, 1 \leq i \leq N_T\}$ of the root that has not been assigned to any packet until now (if all $T_i, 1 \leq i \leq N_T$ are already packetized, go directly to Step 4).

Step 2: Put the T_s into the target packet chosen by the following rules (*Rule* i has a higher priority than *Rule* (i + 1)). If any packet is not chosen by the rules below, go to Step 4.

Rule 1: If an empty packet exists, choose the packet as the target packet.

Rule 2: If the available space at a packet is not less than $c_{thr} + 1$ (e.g., if T_s and c_{thr} elements of $\mathbf{E}_{T_s}$ can be inserted into the same packet), then choose the packet as the target packet.

Rule 3: When any packet does not satisfy *Rule* 2 (i.e., when T_s and c_{thr} elements of $\mathbf{E}_{T_s}$ cannot be inserted into the same packet), if $d_{T_s} - 1$ is less than c_{thr} and the available space at a packet is not less than d_{T_s}, then choose the packet as the target packet with $c_{thr} = d_{T_s} - 1$ in order to insert T_s and $(d_{T_s} - 1)$ elements in $\mathbf{E}_{T_s}$ into the same packet as the best alternative policy of *Rule* 2.

Step 3: Determine a $\mathbf{E}_{T_s}^{pkt}$ by randomly choosing c_{thr} elements in $\mathbf{E}_{T_s}$, where $|A|$ means the cardinality of the set A (i.e., number of elements in the set A). Put $\mathbf{E}_{T_s}^{pkt}$ into the target packet as shown in Fig. 3.

Step 4: If no more roots exist, go to Step 5. Otherwise, the next source symbol is sequentially selected as a root. Repeat Step 1–3 with the new root and the initial c_{thr}.

Step 5: Randomly distribute the remaining encoded symbols into the residual spaces of the packets.

C. *Performance Analysis of the Proposed Packetization Algorithm*

The number of the successfully received elements of $\mathbf{E}_{T_s}$ with T_s is adopted as the performance measure. $\mathbf{E}_{T_s}^{prop}$ and $\mathbf{E}_{T_s}^{conv}$ are defined as subsets of $\mathbf{E}_{T_s}$ whose elements arrive at the receiver without errors in the cases of the proposed packetization algorithm and the conventional algorithm, respectively. Now, we can claim that the proposed packetization algorithm works better than the conventional algorithm if $|\mathbf{E}_{T_s}^{prop}| > |\mathbf{E}_{T_s}^{conv}|$. We consider two scenarios, i.e., when T_s is lost during transmission and when T_s is available at the receiver. Actually, we do not need to consider the case that T_s is lost because $\mathbf{E}_{T_s}$ does not affect the decoding of the root as mentioned previously. Thus, it is assumed that T_s successfully arrives at the receiver. Under the assumption that packet losses occur randomly, we can calculate $|\mathbf{E}_{T_s}^{prop}|$ and $|\mathbf{E}_{T_s}^{conv}|$ as follows:

$$\left|\mathbf{E}_{T_s}^{conv}\right| = \frac{|\mathbf{E}_{T_s}|}{n_p} + \left\{|\mathbf{E}_{T_s}| - \frac{|\mathbf{E}_{T_s}|}{n_p}\right\} \cdot (1 - PLR) \tag{1}$$

$$\left|\mathbf{E}_{T_s}^{prop}\right| = (c_{thr} + \alpha) + (|\mathbf{E}_{T_s}| - (c_{thr} + \alpha)) \cdot (1 - PLR) \tag{2}$$

$$\alpha = \frac{|\mathbf{E}_{T_s}| - c_{thr}}{n_{sym} - (c_{thr} + 1)} \cdot (n_{ps} - (c_{thr} + 1)) \tag{3}$$

where n_p is the number of transmitted packets (i.e., it is the smallest integer greater than n_{sym}/n_{ps}, where n_{sym} is the number of encoded symbols), α denotes the average number of $\mathbf{E}_{T_s}$ symbols in the target packet when the remaining symbols are randomly allocated to packets, and PLR is packet loss rate. To guarantee $|\mathbf{E}_{T_s}^{prop}| > |\mathbf{E}_{T_s}^{conv}|$, our control variable c_{thr} should be set to satisfy the following equation:

$$c_{thr} > \frac{|\mathbf{E}_{T_s}|}{n_p} - \alpha. \tag{4}$$

By combining (3) and (4), we can obtain the following equation:

$$c_{thr} > \frac{|\mathbf{E}_{T_s}| \cdot \{n_{sym} - 1 - n_p \cdot (n_{ps} - 1)\}}{n_p \cdot \{n_{sym} - |\mathbf{E}_{T_s}| - n_{ps}\} + |\mathbf{E}_{T_s}|}. \tag{5}$$

In general, the probability distribution of the degree of an LT encoded symbol is represented by $\{\Omega_1 \ldots\ldots \Omega_k\}$, where k is the number of source symbols and Ω_i means the probability that an encoded symbol has the degree i. Then the distribution is represented by the generator polynomial $\Omega(x) = \sum_{i=1}^{k} \Omega_i x^i$, and the average degree of an encoded symbol is $\Omega'(1)$ [14]. Thus, $\lambda_j^{T_s}$ and $|\mathbf{E}_{T_s}|$ are represented by

$$\lambda_j^{T_s} = \frac{\Omega'(1) \cdot n_{sym}}{k} = \Omega'(1) \cdot \gamma, \text{ for } 1 \leq j \leq (d_{T_s} - 1) \tag{6}$$

$$|\mathbf{E}_{T_s}| = \sum_{j=1}^{d_{T_s}-1} \left(\lambda_j^{T_s} - 1\right) = (\Omega'(1) - 1) \cdot (\Omega'(1) \cdot \gamma - 1) \tag{7}$$

where γ is the LT encoding overhead ratio (i.e., $\gamma = n_{sym}/k$), Finally, we can derive the sufficient condition for c_{thr} so that the proposed packetization algorithm works better than the conventional packetization algorithm. See (8) at the bottom of the next page. It is observed during the experiment that the right-hand side of the above equation is always less than 1 in the wide range of γ. If c_{thr} is set to an integer number between 1 and $\min\{(n_{ps} - 1), |\mathbf{E}_{T_s}|\}$, it is guaranteed that the proposed packetization algorithm can provide better performance than the

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on July 19,2024 at 08:21:00 UTC from IEEE Xplore. Restrictions apply.

conventional packetization algorithm. Detailed experimental results are provided in Section III-A.

III. Experimental Results

During the simulation, the proposed packetization algorithm is implemented by using Java and C/C++. H.264/AVC JM Software [25] is used as a video codec. *Harbour* and *Soccer* of CIF size and 15 frames per second (fps) are employed as test video sequences. The GOP structure is set to include one I-frame and 29 P-frames. The Frame Copy mode of H.264/AVC JM Software (which conceals damaged blocks of the frame using blocks at the same location in previous decoded frames) is adopted for error concealment when some regions in a frame are lost. This section consists of two parts; in the first part, the experiment is executed to verify that the proposed packetization algorithm works reasonably well, and in the second part, the proposed packetization algorithm is applied to deliver video streaming data over an error-prone wireless network. Every simulation is repeated 1000 times and the numbers in the following are the average values of the repeated simulations. The probability distribution of the degree of LT codes $\{\Omega_1 \ldots\ldots \Omega_k\}$ is determined by the robust soliton distribution and two parameters of this degree probability distribution (constant value and allowable decoder failure probability) are set to 0.03 and 0.5, respectively [7], [8].

A. Performance Verification of the Proposed Packetization Algorithm

We first test the condition for the control variable c_{thr} of the proposed packetization algorithm to guarantee better performance than the conventional packetization algorithm, and then compare the proposed packetization algorithm and the conventional packetization algorithm in terms of the LT decoding failure rate according to c_{thr}. Fig. 4 represents the right-hand side values of (8) as γ increases when k and n_{ps} are fixed. As shown in the figures, the lower limit of c_{thr} is always much less than 1 in the wide range of γ. By the way, c_{thr} is an integer between 1 and $\min\{(n_{ps}-1), |\mathbf{E}_{T_s}|\}$ as mentioned in Section II-B. This means the control variable c_{thr} of the proposed packetization algorithm always satisfies (8) to provide better LT decoding performance than the conventional packetization algorithm in the wide range of γ.

Now, we compare the number of elements of $\mathbf{E}_{T_s}$ in the target packets. Experimental results are provided in Table II and Fig. 5. It is apparently observed in Table II that the proposed packetization algorithm presents a higher average value of $\left|\mathbf{E}_{T_s}^{pkt}\right|$ than the conventional packetization algorithm regardless of c_{thr}. Furthermore, the average value of $\left|\mathbf{E}_{T_s}^{pkt}\right|$ becomes large as c_{thr} increases as expected, i.e., more elements of $\mathbf{E}_{T_s}$ are packetized with T_s compared to the conventional packetization algorithm. Incidentally, the standard deviation values of $\left|\mathbf{E}_{T_s}^{pkt}\right|$ show the same phenomena as well. The increased standard deviation means that the decoding success rate may be different according to the root. This difference is caused by the dependency among trees as mentioned in Section II-A. Actually, it is desirable that the standard deviation is relatively small to guarantee almost the same decoding success rate for every symbol. To keep the standard deviation in the tolerable ranges and maximize the average value of $\left|\mathbf{E}_{T_s}^{pkt}\right|$ simultaneously, c_{thr} should be regulated. An example is shown in Fig. 5. Fig. 5 represents the distribution of the number of elements of $\mathbf{E}_{T_s}$ packetized with T_s when n_{ps} and c_{thr} is fixed to 10 and 5, respectively. The (a) and (c) of the figure represent the proposed packetization algorithm while the (b) and (d) show the conventional packetization algorithm. As shown in (b) and (d), the number of elements of $\mathbf{E}_{T_s}$ packetized with T_s look uniformly distributed. By the way, the number of elements of $\mathbf{E}_{T_s}$ packetized with T_s decreases as the proposed packetization algorithm progresses since the dependency among And-Or trees is not considered. However, the proposed packetization algorithm still shows better performance than the conventional packetization algorithm. In the following, c_{thr} is empirically fixed to 5 since $\left|\mathbf{E}_{T_s}^{pkt}\right|$ is almost saturated and the standard deviation value increases rapidly when c_{thr} is greater than 5 as shown in Table II.

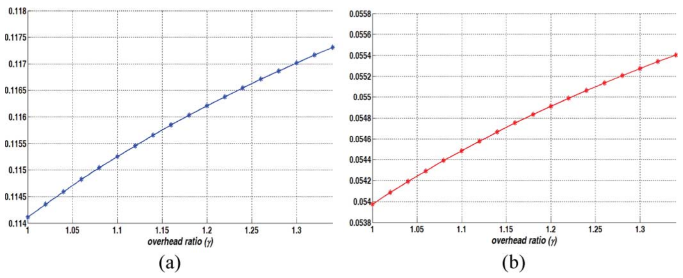


Fig. 4. Lower limit of c_{thr} in (8) according to γ: (a) when $k = 1000$ and $n_{ps} = 10$, and (b) when $k = 2000$ and $n_{ps} = 10$.

Fig. 6 shows the resulting LT decoding failure rate when n_{ps} is 10. Compared to the conventional packetization algorithm, the proposed packetization algorithm always provides a lower average LT decoding failure rate regardless of c_{thr} as expected from Fig. 4. It is also observed that the average LT decoding failure rate of the proposed packetization algorithm decreases as c_{thr} increases because more elements in $\mathbf{E}_{T_s}$ are available with T_s at the receiver. Furthermore, the performance improvement is more noticeable when the packet loss rate is large as shown in the figures. However, the LT decoding failure rate is almost saturated when γ is larger than about 1.3. Even in

$$c_{thr} > \frac{(\Omega'(1)-1)\cdot(\Omega'(1)\cdot\gamma-1)\cdot\{n_{sym}-1-n_p\cdot(n_{ps}-1)\}}{n_p\cdot\{n_{sym}-(\Omega'(1)-1)\cdot(\Omega'(1)\cdot\gamma-1)-n_{ps}\}+(\Omega'(1)-1)\cdot(\Omega'(1)\cdot\gamma-1)} \tag{8}$$

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on July 19,2024 at 08:21:00 UTC from IEEE Xplore. Restrictions apply.

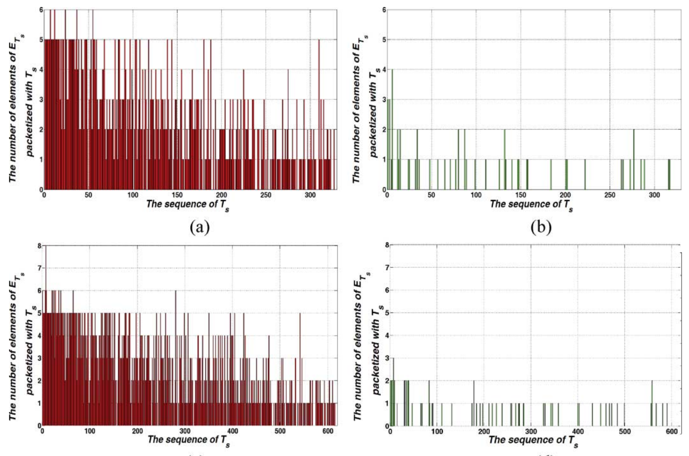


Fig. 5. Comparison of the number of elements of $\mathbf{E}_{T_s}$ packetized with T_s between the proposed and the conventional packetization algorithms with $n_{ps} = 10$ and $c_{thr} = 5$. (a) Proposed algorithm when $k = 1000$ and $\gamma = 1.26$. (b) Conventional algorithm when $k = 1000$ and $\gamma = 1.26$. (c) Proposed algorithm when $k = 2000$ and $\gamma = 1.20$. (d) Conventional algorithm when $k = 2000$ and $\gamma = 1.20$.

TABLE I
SYMBOL DESCRIPTIONS

Symbol	Description
root	Target source symbol
T_i	The i_{th} target encoded symbol of the root
N_T	The number of children of root
d_{T_i}	Degree of T_i
$\mathbf{S}_{T_i}$	Set of source symbols associated with T_i (T_i's children)
$\lambda_j^{T_i}$	The number of links on the j_{th} source symbol in $\mathbf{S}_{T_i}$ $1 \le j \le (d_{T_i} - 1)$
$\mathbf{E}_{T_i}$	Set of encoded symbols associated with T_i
n_{ps}	The number of encoded symbols per packet

these ranges, the proposed packetization algorithm still provides slightly better performance than the conventional packetization algorithm. Consequently the proposed packetization algorithm is able to enhance the LT decoding success rate.

B. Video Quality Comparison When Packet Loss Pattern Is Random

The proposed packetization algorithm is applied to a video streaming service when the packet loss pattern is random. The peak signal-to-noise ratio (PSNR) is used as a performance measure. The symbol size, packet payload size, and c_{thr} are set to 50 bytes, 512 bytes and 5, respectively. Fig. 7 shows the experimental results when packet losses happen randomly and the PLR is fixed to 0.1. In the figures, left figures are the plots of the average decoding failure rate, and the right figures are the corresponding PSNR plots. For all test video sequences, the proposed packetization algorithm provides a lower average LT decoding failure rate and higher average PSNR value than the conventional packetization algorithm. It is observed that the maximum PSNR improvement is about 4 dB when γ is around 1.18 and the improvement is generally more significant when γ is relatively small. When γ is less than 1.18, the decoding failure rate is so high that the receiver cannot display video. The PSNR curves of the proposed packetization algorithm and the conventional packetization algorithm are provided in Fig. 8. For the subjective video quality comparison, pictures are captured at the 30th frame and the 56th frame of *Harbour* and *Soccer* sequences since the PSNR differences are very obvious. The results are presented in Fig. 9 (*Harbour* and *Soccer* are encoded with γ of 1.21 and 1.22, respectively). Although the video decoder at the receiver uses the frame copy technique for error concealment,

TABLE II
RELATION BETWEEN c_{thr} AND $\left|\mathbf{E}_{T_s}^{pkt}\right|$ WHEN $k = 1000$ AND $\gamma = 1.26$

c_{thr}	$\left\|\mathbf{E}_{T_s}^{pkt}\right\|$	
	Proposed packetization algorithm (avg./standard dev.)	Conventional packetization algorithm (avg./standard dev.)
1	1.105 / 0.662	0.171 / 0.508
3	1.983 / 1.301	
5	2.369 / 1.760	
7	2.614 / 2.320	
9	2.804 / 2.692	

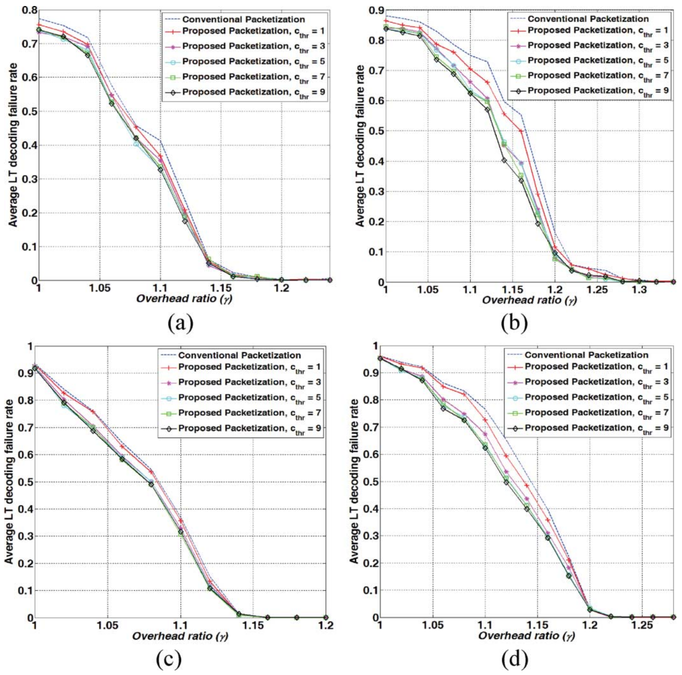


Fig. 6. Average LT decoding failure rate comparison between the proposed and conventional packetization algorithms (n_{ps} is set to 10). (a) when $k = 1000$ and $PLR = 0.05$, (b) when $k = 1000$ and $PLR = 0.1$, (c) when $k = 2000$ and $PLR = 0.05$, and (d) when $k = 2000$ and $PLR = 0.1$.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on July 19,2024 at 08:21:00 UTC from IEEE Xplore. Restrictions apply.

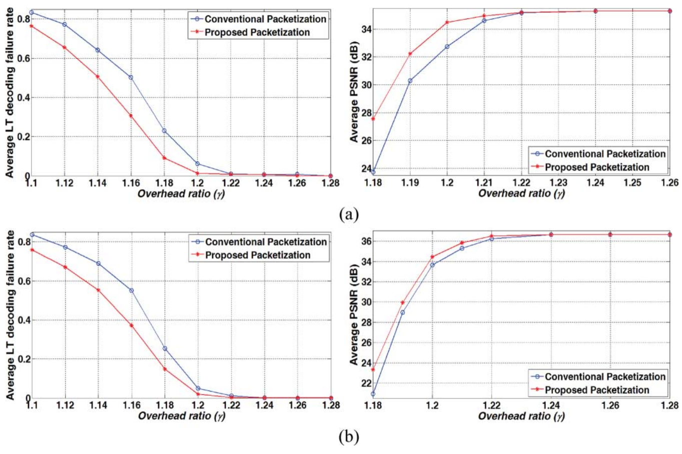


Fig. 7. Comparison of average LT decoding failure rate and PSNR of the test video sequence between the proposed and the conventional packetization algorithms when $n_{ps} = 10, c_{thr} = 5$, and $PLR = 0.1$. (a) *Harbour*. (b) *Soccer*.

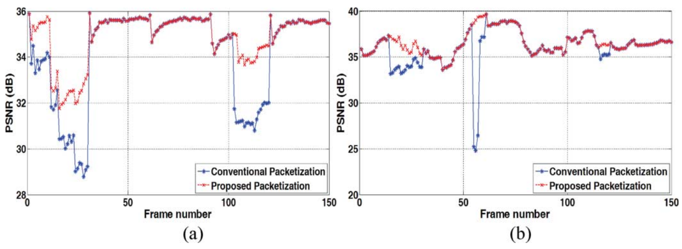


Fig. 8. PSNR of frame sequences comparison between the proposed and the conventional packetization algorithms. (a) *Harbour* when γ is set to 1.21. (b) *Soccer* when γ is set to 1.22.

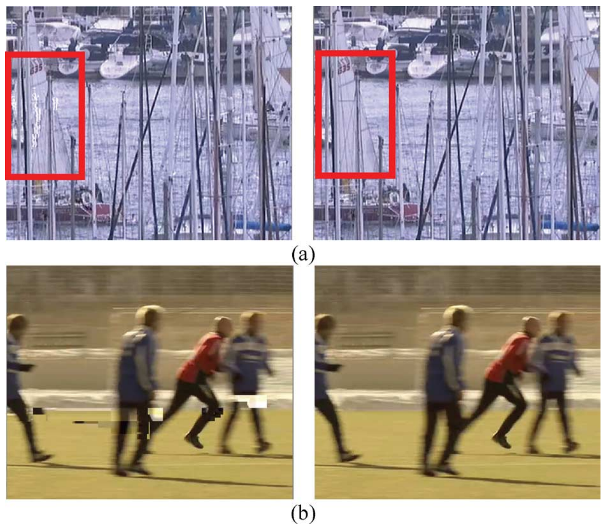


Fig. 9. Subjective video quality comparison between the proposed packetization algorithm (right figure) and the conventional packetization algorithm (left figure). (a) *Harbour* (30th frame) when γ is set to 1.21. (b) *Soccer* (56th frame) when γ is set to 1.22.

it cannot avoid video quality degradation completely. As shown in the figures, the proposed packetization algorithm can provide much better human visual perceptual quality.

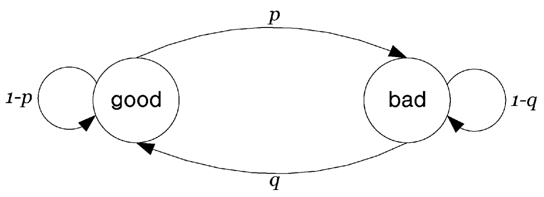


Fig. 10. Two-state Markov channel model (good: no packet loss, bad: packet loss).

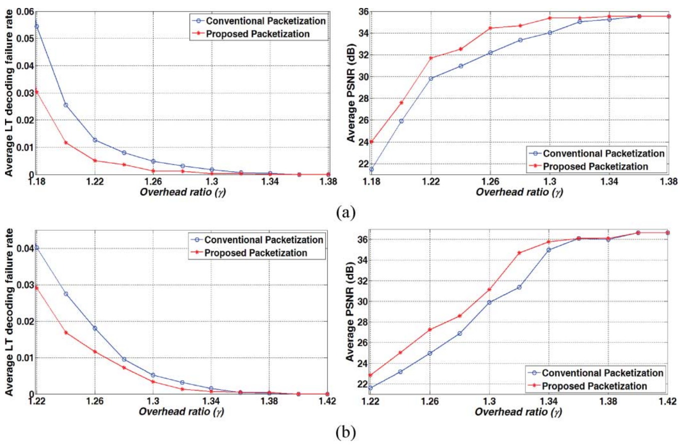


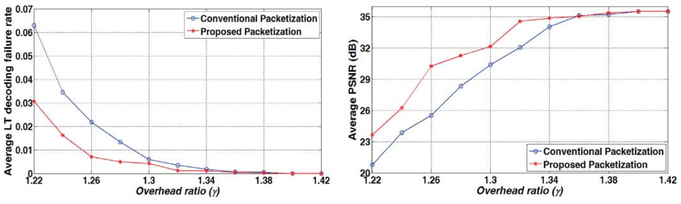
Fig. 11. Performance comparison of average LT decoding failure rate and PSNR of the test video sequence between the proposed and the conventional packetization algorithms when $n_{ps} = 10$, $c_{thr} = 5$, and the transition probability distribution is Case 1. (a) *Harbour*. (b) *Soccer*.

C. *Video Quality Comparison When Packet Loss Pattern Is Burst*

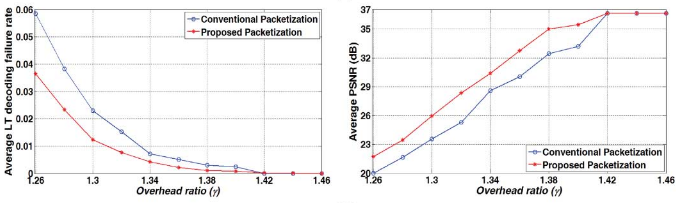
It is well known that the packet loss pattern in wireless network looks burst rather than random. To examine the performance of the proposed packetization in a more realistic wireless environment, a two-state Markov channel model is employed as shown in Fig. 10. In the figure, the state transition probability consists of p (transition probability from good state to bad state) and q(transition probability from bad state to good state). To consider various wireless environments, we test three transition probability distribution cases as shown in Table III [26], [27]. The simulation results are given in Figs. 11–13. Since a high packet loss rate and a burst packet loss length occur very frequently in Cases 1 and 2, γ is increased for successful LT decoding compared to Fig. 7 in Section III-B. Also it is obviously observed in Figs. 11 and 12 that the proposed packetization algorithm can provide a lower average LT decoding failure rate and a higher average PSNR value. When a packet loss pattern is generated with the state transition probability distribution of Case 3, it looks very random, relatively similar to that of Section III-B. The required γ is smaller than those of Cases 1 and 2. Exactly the same phenomenon is observed as Cases 1 and 2 in terms of average LT decoding failure rate and average PSNR value. In summary, the proposed packetization algorithm works better than the conventional packetization algorithm over diverse wireless environments.

Finally, we would like to give some remarks on the computational complexity and the end-to-end delay of the proposed packetization algorithm. First, the encoding and packetization

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on July 19,2024 at 08:21:00 UTC from IEEE Xplore. Restrictions apply.

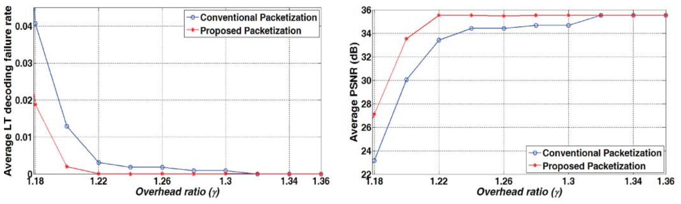


(a)

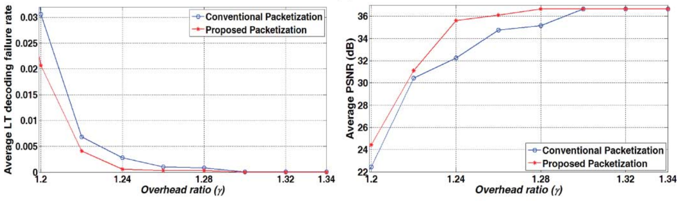


(b)

Fig. 12. Performance comparison of average LT decoding failure rate and PSNR of the test video sequence between the proposed and the conventional packetization algorithms when $n_{ps} = 10$, $c_{thr} = 5$, and the transition probability distribution is Case 2. (a) *Harbour*. (b) *Soccer*.



(a)



(b)

Fig. 13. Performance comparison of average LT decoding failure rate and PSNR of the test video sequence between the proposed and the conventional packetization algorithms when $n_{ps} = 10$, $c_{thr} = 5$, and the transition probability distribution is Case 3. (a) *Harbour*. (b) *Soccer*.

TABLE III
STATE TRANSITION PROBABILITY DISTRIBUTIONS

	p	q
Case 1	0.0087	0.1491
Case 2	0.0120	0.1367
Case 2	0.0418	0.5306

processing time is adopted as a performance measure. Under the assumption that the LT encoding symbol mapping matrix is already known, the processing time is measured at the laptop computer with Windows 7 OS, 2.66-GHz Intel Core 2 Quad CPU, and 4 GB of RAM. Table IV shows the average processing time (LT encoding and packetization) for a source block (a GOP) of video sequences. As shown in the table, the proposed packetization algorithm requires a slightly higher processing time than the conventional packetization algorithm. The reason is that the processing time to search for T_s, target packet, c_{thr} elements in $\mathbf{E}_{T_s}$ can be reduced when the mapping matrix is already known. We can prevent an additional waiting delay for the symbols included

TABLE IV
COMPARISON OF AVERAGE PROCESSING TIME OF A SOURCE BLOCK (A GOP) BETWEEN THE PROPOSED AND THE CONVENTIONAL PACKETIZATION ALGORITHMS WHEN $n_{ps} = 10$ AND $c_{thr} = 5$, AND AVERAGE k OF *HARBOUR* AND *SOCCER* ARE 2684 AND 2642, RESPECTIVELY

γ	**Average processing time of a source block (*ms*)**			
	Conv. packetization algorithm		Prop. packetization algorithm	
	Harbour	*Soccer*	*Harbour*	*Soccer*
1.20	7.86	6.54	9.80	8.74
1.24	8.11	6.72	10.01	9.07
1.28	8.56	6.87	10.33	9.29
1.32	8.74	7.08	10.68	9.45
1.36	9.09	7.27	10.99	9.80

in the packet since the encoded symbol generating sequence is controllable by arbitrarily accessing the row of the mapping matrix corresponding to the current encoded symbol. Second, we consider end-to-end delay of the proposed algorithm. The initial playback delay of the proposed packetization algorithm may be slightly increased compared to the conventional packetization algorithm although the proposed packetization algorithm performs all these processes in parallel after encoding the first source block. However, the effect is minor since video decoding is processed in a GOP unit, and video playback generally begins after buffering a number of GOPs for smooth video streaming. As shown in Table IV, the additional processing time for a source block is less than 10 *ms*, which is insignificant compared to the buffering delay and video decoding delay.

IV. CONCLUSION

In this paper, we have presented an efficient LT encoding pattern-aware packetization algorithm for LT encoded symbols to provide a robust video streaming service over error-prone wireless network. The relation among LT encoded symbols has been analyzed based on LT encoding pattern, and some useful properties are derived. Based on the properties, a robust packetization algorithm has been implemented to reduce dependency among packets. We have verified the performance of the proposed packetization algorithm mathematically and empirically. Finally, we have shown by the experimental results that the proposed algorithm provides much better video streaming service over error-prone wireless network. By combining other advanced fountain codes using feedback information from the receiver [19] or supporting UEP (unequal error protection) [21], it is expected that the proposed packetization algorithm can provide more robust video streaming service with a smaller overhead. We also think that the basic concept of the proposed packetization algorithm may be extended to other fountain codes with some modifications.

REFERENCES

[1] J. G. Apostolopoulos, W. Tan, and S. J. Wee, Video Streaming: Concepts, Algorithms, and Systems, Mobile and Media Systems Laboratory, HP Laboratories, Palo Alto, CA, 2002, HPL-2002-260.
[2] *Generic Coding of Moving Pictures and Associated Audio Information*, ISO/IEC 13818 (MPEG-2), Nov. 1994.
[3] *Overview of the MPEG-4 Standard*, ISO/IEC JTC 1/SC 29/WG 11 N4030, Mar. 2001.
[4] *Video Coding for Low Bit-Rate Communication*, ITU-T Recommendation H.263 version 2, Feb. 1998.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on July 19,2024 at 08:21:00 UTC from IEEE Xplore. Restrictions apply.

[5] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Recommendation H.264 version 3, Nov. 2007.
[6] [Online]. Available: http://www.h265.net/.
[7] J. C. MacKay, “Fountain codes,” *Proc. Inst. Elect. Eng.—Commun.*, vol. 152, no. 6, pp. 1062–1068, 2005.
[8] M. Luby, “LT codes,” in *Proc. Annu. Symp. Foundations of Computer Science*, 2002, pp. 271–280.
[9] A. Shokrollahi, “Raptor codes,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.
[10] P. Maymounkov, Online Codes, New York University, 2002, Tech. Rep. TR2002-833.
[11] Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; Protocols and Codecs, 3GPP, 2008, 3GPP TS 26.346 V8.1.0.
[12] Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H), 2004, ETSI EN 302304 v1.1.1.
[13] S. Sanghavi, “Intermediate performance of rateless codes,” in *Proc. IEEE Information Theory Workshop*, 2007, pp. 478–482.
[14] N. Rahnavard, B. N. Vellambi, and F. Fekri, “Rateless codes with unequal error protection property,” *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1521–532, Apr. 2007.
[15] S. S. Woo and M. K. Cheng, “Prioritized LT codes,” in *Proc. 42nd Annu. Conf. Information Sciences and Systems*, pp. 568–573.
[16] M. C. O. Bogino, P. Cataldi, M. Grangetto, E. Magli, and G. Olmo, “Sliding-window digital fountain codes for streaming of multimedia contents,” in *Proc. IEEE Int. Symp. Circuits and Systems*, May 2007, pp. 3467–3470.
[17] T. Nguyen, L. Yang, and L. Hanzo, “Systematic Luby transform codes and their soft decoding,” in *Proc. IEEE Workshop Signal Processing Systems*, Oct. 2007, pp. 67–72.
[18] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, and Z. Xiong, “Scalable video multicast using expanding window fountain codes,” *IEEE Trans. Multimedia,* Special Issue on Quality-Driven Cross-Layer Design for Multimedia Communications, vol. 11, no. 6, pp. 1094–1104, Aug. 2009.
[19] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, “Adaptive unicast video streaming with rateless codes and feedback,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 2, pp. 275–285, Feb. 2010.
[20] T. Gasiba, T. Stockhammer, J. Afzal, and W. Xu, “System design and advanced receiver techniques for MBMS broadcast services,” in *Proc. IEEE Int. Conf. Communications 2006*, Jun. 2006, pp. 5444–5450.
[21] N. Thomos and P. Frossard, Network Coding of Rateless Video in Streaming Overlays, EPFL, 2009, Tech. rep. LTS-REPORT-2009-011.
[22] M. Luby, M. Mitzenmacher, and A. Shokrollahi, “Analysis of random processes via And-Or tree evaluation,” in *Proc. 9th Annu. ACM-SLAM Symp. Discrete Algorithms*, Jan. 1998, pp. 364–373.
[23] S. Han and H. Song, “Efficient deployment of fountain code for video transmission over wireless packet-switching network,” in *Proc. IEEE Consumer Communications & Networking Conf.*, Jan. 2009.
[24] J. Afzal, T. Stockhammer, T. Gasiba, and W. Xu, “Video streaming over MBMS: A system design approach,” *J. Multimedia*, vol. 1, no. 5, pp. 25–35, Aug. 2006.
[25] *H.264/AVC Reference Software, JM15.0.*, [Online]. Available: http://iphome.hhi.de/suehring/tml/.
[26] A. Konrad, B. Zhao, A. Joseph, and R. Ludwig, “A Markov-based channel model algorithm for wireless networks,” *Wireless Netw.*, vol. 9, no. 3, pp. 189–199, May 2003.
[27] H. Sanneck and G. Carle, “A framework model for packet loss metrics based on loss runlengths,” in *Proc. SPIE/ACM SIGMM Multimedia Computing and Networking Conf.*, Jan. 2000.

Dongju Lee received the B.S. degree from the Department of Computer Science and Engineering, Pusan National University, Pusan, Korea, in 2009. Currently, he is pursuing the M.S. degree in the Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Gyungbuk, Korea.

His research interests are in the areas of channel coding and video streaming system over wireless networks.

Hwangjun Song received the B.S. and M.S. degrees from the Department of Control and Instrumentation (EE), Seoul National University, Seoul, Korea, in 1990 and 1992, respectively, and the Ph.D. degree in electrical engineering systems from the University of Southern California, Los Angeles, in 1999.

From 1995 to 1999, he was a research assistant in Signal and Image Processing Institute (SIPI) and Integrated Media Systems Center (IMSC), University of Southern California. From 2000 to 2005, he was an assistant Professor/Vice Dean of Admission Affairs at Hongik University, Seoul, Korea. Since February 2005, he has been with the Department of Computer Science and Engineering and Division of IT Convergence Engineering, Pohang University of Science and Technology (POSTECH), Gyungbuk, Korea. His research interests include multimedia signal processing and communication, image/video compression, digital signal processing, network protocols necessary to implement functional image/video applications, control systems, and fuzzy-neural systems.

Dr. Song received the Haedong Best Paper Award from the Korean Institute of Communication Science in 2005.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on July 19,2024 at 08:21:00 UTC from IEEE Xplore. Restrictions apply.