Layer-Aligned Multipriority Rateless Codes for Layered Video Streaming

Hsu-Feng Hsiao, *Member, IEEE*, and Yong-Jhih Ciou

***Abstract*— There exists a multitude of techniques, including automatic repeat request and error correction codes, to minimize data corruption when transmitting over error-prone networks. Streaming of multimedia data can usually withstand a certain level of data loss, yet have strict limitations on the latency tolerance. To enable acceptable reliability of transmission and low transmission latency, the channel coding approach is usually more appealing at the cost of additional bandwidth. In this paper, an *N*-cycle layer-aligned overlapping structure, which is good for layered data, is proposed. Accordingly, layer-aligned multipriority rateless codes were developed with favorable probabilities to control the protection strength for each layer of the streaming data. The major contribution of this paper is the analytical model developed to predict the failure decoding probabilities for each video layer and it is shown to achieve accurate estimation. A prediction model to estimate the expected decompressible video frames was developed for use with the developed codes for streaming scalable videos. By maximizing the number of expected decompressible video frames, the protection strength of the developed codes can then be determined. Simulation results show that the developed codes are good for streaming layered videos, which are difficult to deal with using traditional rateless codes, with or without unequal error protection.**

***Index Terms*— Rateless codes, scalable video coding, unequal error protection, video streaming.**

I. Introduction

FOR ONE-to-many multimedia streaming over error-prone networks within heterogeneous environments, the connection quality from the server to each client may vary. As well as fluctuating network conditions, the causes of heterogeneous environments also include the computational and display ability of the clients. A traditional approach, known as simulcast, copes with heterogeneous environments through source coding, encoding videos into several bit streams. An appropriate bit stream is then chosen to transmit to each client. The disadvantages of simulcast, when compared with layered multicast, such as the receiver-driven layered multicast [1], include larger bandwidth consumption and storage requirements, especially when the number of users is large and the available bandwidth for each user differs significantly. Simulcast usually requires more bandwidth when there are multiple heterogeneous users simultaneously downloading data from a single server, since several streams encoded at different bit rates need to be transmitted over the internet. Alternatively, a video can be encoded using scalable video-coding techniques, such as scalable extension in MPEG-4 AVC [2]. The functionalities provided by scalable extension in MPEG-4 AVC allow a video encoded into a bit stream to be further partitioned into substreams with the desired bit rates and video resolutions. Therefore, when compared with simulcast, streaming videos encoded using scalable video coding can better fit the heterogeneous environments that a streaming service may deal with.

When transmitting over error-prone channels, packets may be discarded or damaged. Common techniques to combat transmission loss include the automatic repeat request (ARQ) [3] mechanism and error control coding. Strong ARQ is usually not preferable in multimedia streaming applications due to increased delay and exacerbated jitter at the application layer. On the other hand, error control techniques may improve delay-sensitive streaming applications better. For example, Reed–Solomon codes [4], which belong to the class of maximum distance separable codes, provide a popular channel coding technique. However, the computational complexity of Reed–Solomon codes increases exponentially to the number of symbols. In DVB-H, multiprotocol encapsulated forward error correction is implemented by interleaving the information and protection packets resulting from the Reed–Solomon error erasure code to deal with the burst error. The code complexity of fountain codes [5]–[7] (also known as rateless codes) is linear time. For a fairly large number of message blocks, fountain codes can generate a virtually infinite number of coded blocks on demand and on the fly. Other efficient coding methods include raptor codes [6], low-density parity check codes [9], [10], Tornado codes [11], and online codes [12]. A modified version of raptor codes has been adopted in the 3GPP MBMS standard [8], as well as in the DVB-H and DVB-IPTV standards for transmission over DVB networks and IP networks, respectively.

Before streaming multimedia data, the content is compressed and divided into sections with multiple symbols (or blocks). Channel coding, such as rateless erasure codes, is then applied to each section. The purpose of the sections is to avoid undesired delay. For traditional rateless codes, data are first split into message blocks of equal length. The encoding of rateless codes produces coded blocks as needed. To generate a coded block, a degree d is first chosen according to a predefined degree distribution. d message blocks are

Manuscript received April 12, 2013; revised July 31, 2013 and October 1, 2013; accepted January 21, 2014. Date of publication January 28, 2014; date of current version August 1, 2014. This work was supported in part by the Ministry of Economic Affairs of Taiwan under Grant 102-EC-17-A-03-S1-214 and in part by the National Science Council of Taiwan under Grant 101-2221-E-009-086-MY3. This paper was recommended by Associate Editor J. F. Arnold.

The authors are with the Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: hillhsiao@cs.nctu.edu.tw; cyj.cs98g@nctu.edu.tw).

Digital Object Identifier 10.1109/TCSVT.2014.2302533

then selected uniformly from the same section to form a coded block by performing an exclusive-OR (XOR) operation on those message blocks. If the number of message blocks is large enough, the number of possible coded blocks can be considered as more than enough for practical usage. The decoding process of rateless codes is achieved using the belief propagation algorithm. Since the encoding and decoding stages rely on XOR operations, the computational complexity is minimal.

The design complexity of rateless codes lies in the generation of the degree distribution. As reported in [5], one possibility for the choice of the degree distribution is the ideal soliton distribution, where the probability of degree 1 is inversely proportional to the number of message blocks. However, in practice, this distribution suffers from a lower coding efficiency. A robust soliton distribution used in LT codes [5], which adjusts and normalizes the probabilities of the ideal soliton distribution, has been further developed. In raptor codes [6], an approach to determine the output degree distribution was developed so that a large fraction of the input symbols can be recovered, and the average degree is small.

To improve the coding efficiency, a sliding window scheme [14], [15] is proposed by connecting two sections into a window such that part of the window overlaps with a previous window. Message blocks in the same window are selected to produce coded blocks. Allowing the windows to overlap can enlarge the effective window size and improve the coding efficiency. Traditional rateless codes can only provide equal protection for each message block. For data with unequal importance, a better approach is to intelligently protect the data according to its importance. The cascading error protection scheme in [16] and [17] suggests that different portions of media streams can be encoded at various times to achieve unequal error protection through the Reed–Solomon codes. The optimized distribution of parity packets is determined by maximizing the video quality. In [18], an expanding window scheme for rateless codes provides unequal error protection by increasing the number of times data with greater importance.

In [19], the unequal error protection property is achieved by selecting message blocks with various probabilities. An analytical model is also available to predict the probability of unsuccessful decoding. For message blocks that are chosen with a higher probability, the chance to successfully decode them is supposed to be greater. The degree distribution of that method follows the same table used in Raptor codes [6]. However, the prediction error of the analytical model is between 15% and 45%. A conceptually similar approach was proposed in [20] to achieve unequal error protection by creating multiple duplicates. Essentially, more duplicates of a message block results in a greater probability for that message block to be chosen. However, there is no prediction model provided to estimate the performance of the duplications. Therefore, the number of duplications required for each message block relies on experimentation.

The contributions of this paper are as follows. An N-cycle layer-aligned overlapping structure is designed for unequal error protection of data with multiple priorities, where N is the number of priorities and each part of a unique priority in a section is protected by N windows. Layer-aligned multipriority rateless codes (LMRCs) are proposed based on this structure, where the protection strength can be controlled unequally for different portions of the data. An analytical model to predict the probabilities of the unsuccessful decoding of each part of the data is derived, which is the major contribution of this paper. To evaluate the importance of the proposed codes, we designed a resource distribution scheme for video streaming, which is compressed with a scalable video encoder. The resource distribution scheme is based on another proposed model, which calculates the number of decompressible video frames at given probabilities of unsuccessful decoding. Using the analytical model of LMRCs and the model of expected decompressible video frames, the resource distribution scheme can decide on the protection strength required to achieve the best video quality on the receiver side.

This paper is organized as follows. Section II presents the proposed N-cycle layer-aligned overlapping structure, the corresponding LMRCs, and the development of the analytical model. The prediction of expected decompressible frames for the resource distribution scheme is presented in Section III. In Section IV, we will compare the experimental results with standard methods. Section V concludes this paper.

II. Proposed LMRCs

For one-to-many multimedia streaming, scalable video coding is regarded as a promising compression approach to deal with time-dependent bandwidth fluctuation among heterogeneous receivers. The scalable extension to H.264/MPEG-4 AVC is considered as the state of the art for scalable video coding. A video compressed using such technology is composed of one or several video layers. The base layer of an AVC scalable video stream can be configured to be MPEG-4 AVC compatible, and the enhancement layers of the stream are encoded by fidelity, spatial, and temporal scalability coding tools. To decode a video frame of a higher layer at the decoder side, all the lower layers of the coded stream are required. The more accumulated video layers that are received, the better the decoded video quality will be. Therefore, the importance of each layer in a layered video is different. LMRCs are proposed in this section for the unequal error protection of data with multiple priorities, where the protection strength can be varied for different portions of the layered data. An analytical model to predict the protection strength will be proposed thereafter.

To analyze the potential performance of rateless codes, the AND–OR tree analysis technique has been adopted in [12] and [13]. Since we modified the AND–OR tree structure to analyze the proposed LMRCs, a brief introduction of conventional AND–OR trees is as follows. An AND–OR tree, T_L, is a generated tree of depth $2L$. The root of this tree is at depth 0. Its children are at depth 1, and their children are at depth 2, and so on. An OR node with no children is assumed to have a value of zero, and an AND node with no children is assumed to have a value of one. An example of such AND–OR trees is shown in Fig. 1. An OR node in the AND–OR tree represents a message block in rateless codes, and an AND node stands for a coded block.

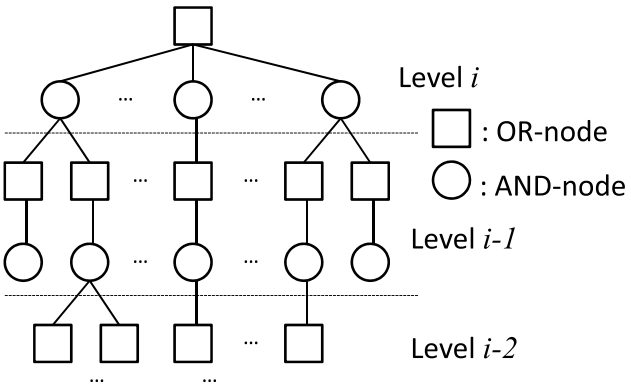

Fig. 1. Example of conventional AND–OR trees.

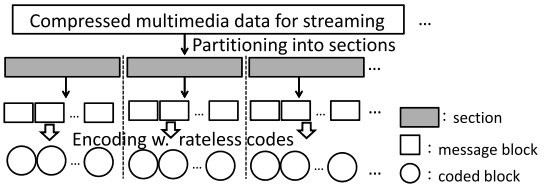

Fig. 2. Streaming protection with rateless codes.

All nodes at depth $0, 2, 4, \ldots, 2L$–2 are labeled as OR nodes, and are evaluated by performing a logical OR operation on their children. Similarly, all nodes at depth $1, 3, 5, \ldots, 2L$–1 are labeled as AND nodes, and are evaluated by performing a logical AND operation on their children. The analysis is used to model the probability that T_L's root is evaluated as zero by applying a deductive approach. More details can be found in [13], where an AND–OR tree lemma was suggested to be used in a deductive manner iteratively. The coding efficiency of rateless codes improves when there are more message blocks in a section.

A. N-Cycle Layer-Aligned Overlapping Structure and LMRCs

Multimedia streaming has recently become one of the primary sources of bandwidth consumption. For video streaming protected using generic rateless codes, compressed videos require division into sections. The content in a section is then split into message blocks, where a number of coded blocks are produced for each section. The section size is usually determined according to the delay restrictions of the streaming application. However, when the length of a section is larger, the coding efficiency of generic rateless codes is improved at the cost of a longer system delay. In general, a larger section size leads to a smaller coding overhead, while a smaller section size results in a shorter coding delay. A typical coding flow for rateless codes on streaming data is shown in Fig. 2.

In the traditional sliding window scheme [14], a window slides from the position of the previous window by a constant distance s. As a result, part of the window overlaps with the previous window. If the length of each layer is different, the overlapped portion cannot synchronize with the data despite the choice of s. Therefore, in each window, the contained video layers can be various and not suitable for analysis. As shown in Fig. 3, the combination of message blocks in each window is different from the others, even if there are only two video

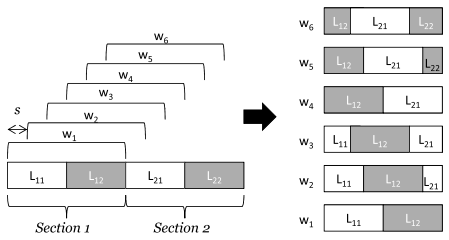

Fig. 3. Content of each window in a sliding window scheme.

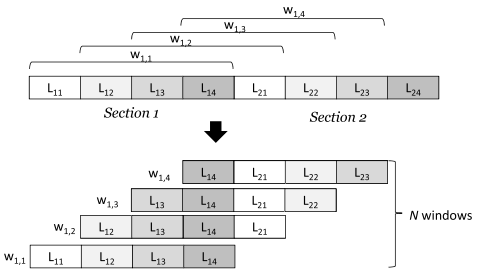

Fig. 4. Example of a four-cycle layer-aligned overlapping structure.

layers. In addition, since there is no analytical model presented in [14], the probabilities of choosing message blocks in each portion of a window are determined through experimentation.

In this paper, an N-cycle layer-aligned overlapping structure is proposed for layered data with N layers, and therefore N priorities. An example of the four-cycle layer-aligned overlapping structure is shown in Fig. 4. The data of the jth layer in the ith section are labeled as L_{ij}. The windows for channel coding are formed as follows. The first window covers the first section entirely; the second window slides by a distance equal to the length of the first layer in a section; the third window slides by a distance equal to the length of the second layer, and so on. Each window consists of N successive layers, which may be from either one or two sections. A window $w_{i,1}$ will contain data from layers $(L_{i1}, L_{i2}, \ldots, L_{iN})$, and a window $w_{i,j}$ will contain data from layers $(L_{ij}, L_{i(j+1)}, \ldots, L_{iN}, L_{(i+1)1}, L_{(i+1)2}, \ldots, L_{(i+1)(j-1)})$, where $1 < j \leq N$.

The data in each layer are partitioned into message blocks in each section. Note that the number of message blocks in different layers can vary. For each section, there are exactly N windows covering either part of or all of that section. Any two successive windows differ only in one layer. In addition, any layer L_{ij} in each section except for the first and the last sections will be covered by exactly N windows.

The proposed LMRCs are built on the N-cycle layer-aligned overlapping structure mentioned above. The message blocks for each window are formed from N different layers. Each of the desired coded blocks for each window is produced using the following procedure.

1) Determine a degree d according to a predefined degree distribution.
2) Choose d message blocks from N different layers. The probability of choosing a certain message block from video layer j is p_j, where $1 \leq j \leq N$. Suppose that there are n_j message blocks from layer j in each window, $1 = \sum_{j=1}^{N} p_j n_j$.

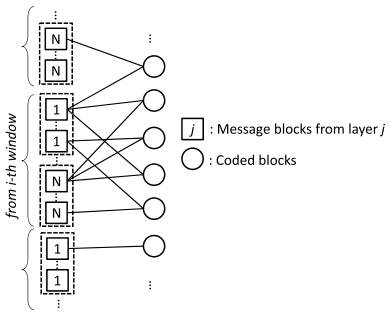

Fig. 5. Bipartite graph G of an N-cycle layer-aligned overlapping structure.

3) A coded block is formed from the result of an XOR operation on the chosen message blocks.

The number of desired coded blocks depends on the desired unsuccessful/successful decoding probability of message blocks as well as the available bandwidth for video streaming.

The protection strength provided by the LMRCs on different layers can be controlled through the vector of favorable probabilities $(p_1, p_2, \ldots, p_N)$. Alternatively, a weighing vector $(\omega_1, \omega_2, \ldots, \omega_N)$ of layers is defined, where ω_j is the probability to choose a random message block from video layer j. The relationship for the conversion between ω_j and p_j is

$$p_j = \frac{\omega_j}{\sum_{i=1}^{N} n_i \cdot \omega_i}. \tag{1}$$

The decoding of message blocks can be done using the belief propagation algorithm. A message block is recovered by finding the corresponding coded block without any unrecovered message blocks except itself. A recovered message block can then help reduce the number of unrecovered message blocks connected to other coded blocks.

B. Analytical Model for LMRCs

In this section, an analytical model is developed to determine the probabilities of unsuccessfully decoding each part of the data. The probability of an unsuccessful decoding is defined as the number of unrecovered message blocks divided by the number of message blocks in the same partition. The analytical model can be used to find a suitable vector of favorable probabilities for transmitting layered data. We can treat the coding paradigm of the LMRCs as a bipartite graph G, (an example of which is shown in Fig. 5), and form an AND–OR tree from it. A table of many symbols used in this section is shown in Table I.

For each window, there are k message blocks that are treated as OR nodes in an AND–OR tree. There are n_j message blocks from layer j in each window, therefore, $1 \leq j \leq N$, and $k = \sum_{j=1}^{N} n_j$. $\gamma_w k$ coded blocks will be produced as AND nodes for each window.

Let $\Omega(x) = \sum_{d=1}^{k} \Omega_d x^d$ be the polynomial generator corresponding to the probability distribution of the degrees of coded blocks. Ω_d is the probability of a coded block having degree d.

TABLE I
SYMBOL DEFINITION

Term	Definition
k	Number of message blocks in a window
n_j	Number of message blocks of layer j in a window
$GT_{i,j}$	AND-OR tree with height $2i$ where the root node is an OR-node at level i from layer j
$p_{i,j}$	Probability for the root node of an AND-OR tree $GT_{i,j}$ to be evaluated as 0
q_i	Probability for an AND-node at level i to be evaluated as 0
Ψ_d	Probability for an AND-node with a fixed vector d to be evaluated as 0
A_d	Probability for an edge to connect to an AND-node of degree d
N	Number of video layers
p_j	Probability of choosing a certain message block from video layer j
$\lambda_{m,j}$	Probability for an OR-node with degree m from layer j
$R_{m,j}$	Probability of an edge connecting to an OR-node with degree m from layer j
μ	Average number of degrees of a coded block

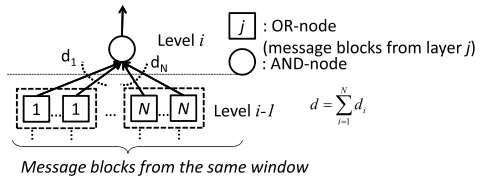

Fig. 6. Connection from the OR nodes in one window to the parent node, an AND node.

If an OR node is evaluated as 1 (true), the corresponding message block is recovered. If an AND node is evaluated as 1, it can help to recover its neighbors. The objective of the analytical model for the LMRCs is to derive the probability $p_{i,j}$ for the root node of an AND–OR tree $GT_{i,j}$ to be evaluated as 0 (false). The notation $GT_{i,j}$ is used to symbolize that the height of the AND–OR tree is $2i$ and that the root node is an OR node at level i from layer j, where $1 \leq j \leq N$. An example of the structure of the AND–OR tree is shown in Fig. 1.

To derive the probability $p_{i,j}$, we first determine the probability q_i for an AND node at level i to be evaluated as 0. Suppose that the number of OR nodes from layer j at level i-1 connected to an AND node with degree $(d + 1)$ at level i is d_j, where $j = 1, \ldots, N$, as shown in Fig. 6. Vector $(d_1, d_2, \ldots, d_N)$ is noted as the layer vector $\boldsymbol{d}$ of the AND-node connection. The probability Ψ_d for an AND node with a fixed vector $\boldsymbol{d}$ to be evaluated as 0 is given by

$$\psi_{\boldsymbol{d}} = 1 - \prod_{j=1}^{N} (1 - p_{i-1,j})^{d_j}. \tag{2}$$

The probability q_i can be obtained by considering all combinations of possible degree distributions

$$q_i = \sum_{d=0}^{k-1} \left(A_{d+1} \cdot \left(\sum_{\forall \boldsymbol{d}: d_1 + d_2 + \cdots + d_N = d} wdp(\boldsymbol{d}) \cdot \psi_{\boldsymbol{d}} \right) \right) \tag{3}$$

where wdp $(\boldsymbol{d})$ is the probability for a coded block (an AND node) with layer vector $\boldsymbol{d}$. In other words, wdp $(\boldsymbol{d})$ is the

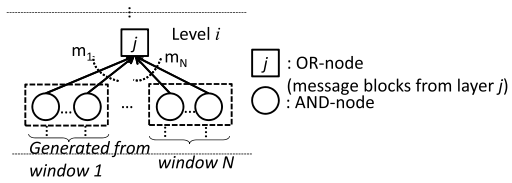

Fig. 7. Connection from the AND nodes to the parent node, an OR node of layer j.

probability for a coded block to select d_i message blocks from layer i, for all $1 \leq i \leq N$. The value of wdp $(\boldsymbol{d})$ can be calculated using the multivariate version of Wallenus' noncentral hypergeometric distribution [21] for uneven selection. A_d is the probability for an edge to connect to an AND node of degree d. Since there are μ degrees of a coded block on average, the probability A_d can be calculated as

$$A_d = \frac{d \cdot \Omega_d}{\mu} = \frac{d \cdot \Omega_d}{\Omega'(1)}. \tag{4}$$

The probability $p_{i,j}$ for the OR node from layer j at level i to be 0 is calculated by considering the scenario where all its child nodes are evaluated as 0. According to the proposed N-cycle layer-aligned overlapping structure, each message block is used in N windows for producing coded blocks. Suppose that the number of AND nodes that are produced in window j and connected to a certain OR node at the same level, level i, is m_j, where $j = 1, \ldots, N$, as shown in Fig. 7. Vector $(m_1, m_2, \ldots, m_N)$ is noted as the window vector $\boldsymbol{m}$ of the OR node connection.

The probability $\lambda_{m,j}$ for an OR node with degree m from layer j can then be expressed as

$$\lambda_{m,j} = \sum_{\forall \boldsymbol{m}: m_1+m_2+\cdots+m_N=m} \left(\prod_{x=1}^{N} \binom{\mu\gamma_w k}{m_x} \cdot p_j^{m_x} \cdot (1-p_j)^{\mu\gamma_w k - m_x} \right) \tag{5}$$

where p_j is the probability of choosing a certain message block from layer j and $\mu\gamma_w k$ is the total number of degrees produced in a window.

In addition, the probability $R_{m,j}$ of an edge connecting to an OR node with degree m from layer j is calculated as

$$R_{m,j} = \frac{m \cdot \lambda_{m,j} \cdot n_j}{N \cdot (\mu \cdot \gamma_w \cdot k \cdot p_j \cdot n_j)}. \tag{6}$$

The numerator in (6) is the number of edges connected to message blocks with degree m from layer j, and the denominator is the sum of edges that connect to an OR node from layer j in N windows. The probability $p_{i,j}$ of all child nodes of an OR node from layer j at level i being 0 can then be expressed as

$$p_{i,j} = \sum_{m=0}^{N \cdot \mu \cdot \gamma_w \cdot k \cdot p_j - 1} (R_{m+1,j} \cdot q_i^m). \tag{7}$$

The probability of unsuccessfully decoding every layer of the data can then be calculated interactively using (3) and (7).

C. Fast Calculation of the Analytical Model for the LMRCs

The computation complexity of the proposed LMRCs is the same as the complexity of other rateless codes using the same degree distribution. As for the analytical model, the computation complexity of the calculation of $p_{i,j}$ can be overwhelming, especially when the value of k is very large. In particular, the calculation of $\lambda_{m,j}$ in (5), which is needed to calculate $p_{i,j}$, is the worst one. The possible combinations in (5) are obviously too enormous to calculate in real time.

When k is large and p_j is small, the Poisson distribution can be used to simplify the calculation of $\lambda_{m,j}$

$$\begin{aligned}
&\lambda_{m,j} \\
&= \sum_{\forall \boldsymbol{m}: m_1+m_2+\cdots+m_N=m} \left(\prod_{x=1}^{N} \binom{\mu\gamma_w k}{m_x} \cdot p_j^{m_x} \cdot (1-p_j)^{\mu\gamma_w k - m_x} \right) \\
&\approx \sum_{\forall \boldsymbol{m}: m_1+m_2+\cdots+m_N=m} \left(\prod_{x=1}^{N} \frac{e^{-\mu\gamma_w k p_j} \cdot (\mu\gamma_w k p_j)^{m_x}}{m_x!} \right) \\
&= e^{-N\mu\gamma_w k p_j} \cdot (\mu\gamma_w k p_j)^m \cdot \sum_{\forall \boldsymbol{m}: m_1+m_2+\cdots+m_N=m} \left(\frac{1}{\prod_{x=1}^{N} m_x!} \right).
\end{aligned} \tag{8}$$

By applying the multinomial theorem, (8) becomes

$$\begin{aligned}
\lambda_{m,j} &\approx e^{-N\mu\gamma_w k p_j} \cdot (\mu\gamma_w k p_j)^m \\
&\quad \cdot \sum_{\forall \boldsymbol{m}: m_1+m_2+\ldots+m_N=m} \left(\frac{1}{\prod_{x=1}^{N} m_x!} \right) \\
&= e^{-N\mu\gamma_w k p_j} \cdot (\mu\gamma_w k p_j)^m \cdot \frac{N^m}{m!} \\
&= e^{-N\mu\gamma_w k p_j} \cdot \frac{(\mu\gamma_w k p_j N)^m}{m!}.
\end{aligned} \tag{9}$$

Therefore, the calculation of $R_{m,j}$ is reduced to

$$\begin{aligned}
R_{m,j} &= \frac{m \cdot \lambda_{m,j} \cdot n_j}{N \cdot (\mu \cdot \gamma_w \cdot k \cdot p_j \cdot n_j)} \\
&\approx e^{-N\mu\gamma_w k p_j} \cdot \frac{(\mu\gamma_w k p_j N)^{m-1}}{(m-1)!}.
\end{aligned} \tag{10}$$

Using the approximation of $R_{m,j}$ shown in (10), (7) results in

$$\begin{aligned}
p_{i,j} &= \sum_{m=0}^{N \cdot \mu \cdot \gamma_w \cdot k \cdot p_j - 1} (R_{m+1,j} \cdot q_i^m) \\
&\approx e^{-N\mu\gamma_w k p_j} \cdot \sum_{m=0}^{N \cdot \mu \cdot \gamma_w \cdot k \cdot p_j - 1} \frac{(\mu\gamma_w k p_j N q_i)^m}{m!}.
\end{aligned} \tag{11}$$

Since the summation in (11) is the Taylor series expansion of an exponential term, the approximation of $p_{i,j}$ can be written as

$$\begin{aligned}
p_{i,j} &\approx e^{-N\mu\gamma_w k p_j} \cdot e^{\mu\gamma_w k p_j N q_i} \\
&= e^{\mu\gamma_w k p_j N(-1+q_i)}.
\end{aligned} \tag{12}$$

A fast calculation method of $p_{i,j}$ was developed and the final result in (12) shows that it is now much more computationally efficient and can be finished in real time.

III. Expected Decompressible Video Frames of the Proposed LMRCs

The vector of favorable probabilities $(p_1, p_2, \ldots, p_N)$ can control the probability of unsuccessfully decoding message blocks from video layer j, as shown in either (7) or (12). The decoding status of different video layers will affect the achievable quality in the received video stream.

The base layer of an MPEG-4 AVC scalable video stream can be configured to be MPEG-4 AVC compatible, and the enhancement layers of the stream are encoded by the scalability tools, such as SNR, spatial, and temporal scalabilities. The spatial scalability is offered by encoding the video at different resolutions for different layers. For the inter layer prediction, interlayer motion/residual prediction and intraprediction are utilized to facilitate the compression. With the SNR scalability, the substream provides the same spatio–temporal resolution as the complete bit stream, but with a lower video quality.

To decompress a video frame of a certain layer of a video encoded with scalable video coding tools, such as the scalable extension of MPEG-4 AVC, all the lower layers of the same unit of the coded stream are required. The decoded video quality improves as the number of accumulated video layers increases. In this section, a prediction model to link the relationship between the protection strength of the proposed LMRCs and the decompressible video frames is shown. This prediction model is used to determine the favorable probabilities $(p_1, p_2, \ldots, p_n)$ by maximizing the expected decompressible frame number. The more video frames that can be decompressed, the better the video quality will become.

Although the reference pictures can be arbitrarily assigned in the MPEG-4 AVC, temporal scalability using hierarchical-B motion compensation was shown to be the more efficient [2]. Therefore, temporal scalability using hierarchical-B motion compensation, as shown in Fig. 8, is used for the following derivation as well as our simulations. The content of a number of groups of pictures (GOPs) is compressed and treated as one section. It should also be noted that the key picture in a GOP is encoded as an instantaneous decoder refresh (IDR) picture.

Let $pl(j)$ be the probability that all message blocks from layer j in a window can be recovered correctly. $pl(j)$ can be expressed as

$$pl(j) = (1 - p_{i,j})^{n_j}. \tag{13}$$

Due to the accumulating nature of scalable video coding, i.e., that a frame of layer j can be decompressed when all the layers i are received correctly, where $1 \le i \le N$, the number of decompressible video frames f_n can be calculated using

$$f_n = \sum_{i=1}^{N} n(i) \cdot \left(\prod_{j=1}^{i} pl(j) \right) \tag{14}$$

where $n(j)$ represents the number of frames of layer j in a section. For the example in Fig. 8, $n(1) = 1$, $n(2) = 1$, $n(3) = 2$, and so on.

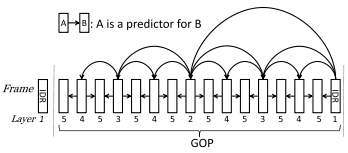

Fig. 8. Hierarchical-B motion compensation in MPEG-4 AVC.

The resource distribution scheme will determine the vector of favorable probabilities $(p_1, p_2, \ldots, p_N)$, which maximizes the number of expected decompressible frames, as shown in (15), where f_n is obtained using (3), (12), and (14), such that the number of coded blocks is affordable with a given available bandwidth. Note that (3) and (12) are from the analytical model of the proposed LMRCs

$$(p_1, p_2, \ldots, p_N)^* = \underset{\forall (p_1, p_2, \ldots, p_N)}{\arg\min} f_n. \tag{15}$$

IV. Simulation Results

In this section, we will first examine the prediction performance of the analytical model for the proposed LMRCs. The prediction model for the expected decompressible frames described in Section III will also be verified. The performance of the proposed codes will then be compared against other methods in the literature.

A. Accuracy Analysis of the Analytical Model for LMRCs

To determine the accuracy of the analytical model for the LMRCs, the following experiments were performed. The first simulations were for data with two layers ($N = 2$), where $n_1 : n_2 = 3 : 7$, and the degree distribution used in Raptor codes for coded blocks [6] was applied.

The weighing vector $(\omega_1, \omega_2) = (\omega_1, 1-\omega_1)$, with different ω_1 (0.05–0.95, with a step size = 0.05) was simulated. The prediction error of each layer between the predicted results using the analytical model and the experimental results by actually coding and decoding is defined as

$$\begin{aligned} &\text{Prediction error } (\%) \\ &\quad = \frac{1}{\text{Number of simulations}} \\ &\qquad \cdot \sum_{\text{for each } \omega_1} \frac{|\text{Prediction} - \text{Simulation}|}{\text{Simulation}} \cdot 100\%. \end{aligned} \tag{16}$$

The performance of the analytical model is shown in Fig. 9 with respect to increasing coding overhead and the number of sections in the simulations. In Fig. 10, the performance is shown with respect to different window sizes. The x-axis is the coding overhead ε, and is defined as $\varepsilon = N\gamma_w - 1$.

It can be observed from Figs. 9 and 10 that the prediction error is less than 10% for all the simulation results. In general, the accuracy of the analytical model improves with a higher number of sections and a bigger window size. The reason for this is that there is a higher chance to transform the bipartite graph to an AND–OR tree structure when using bigger windows. Any layer L_{ij} in each section will be covered by exactly N windows during the development of the analytical model. However, this is not the case for the first and the last sections.

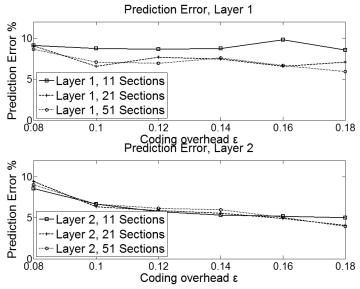

Fig. 9. Prediction error of the analytical model at $k = 2000$.

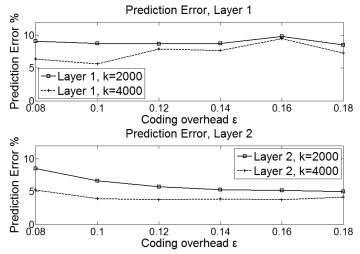

Fig. 10. Prediction error of the analytical model at 11 sections.

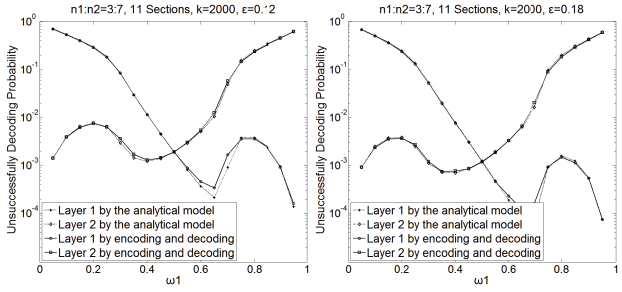

Fig. 11. Performance of the analytical model at 11 sections.

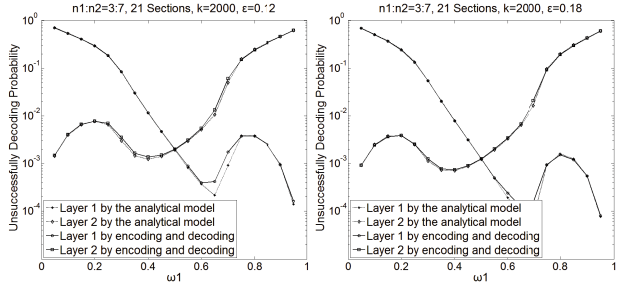

Fig. 12. Performance of the analytical model at 21 sections.

Therefore, when there are more sections, the influence of the first and last sections are smaller.

The probabilities of unsuccessfully decoding the scenarios with two layers at various weighing vectors, coding overhead, and the number of sections are shown in Figs. 11–13. As also found similarly in the expanding window fountain codes [18] and rateless codes with unequal error protection [19], local optimal values exist in our proposed codes where the unsuccessfully decoding probability of one layer reaches its local minimal while the probability of the other layer is not significantly deteriorated. The two local optimal values in

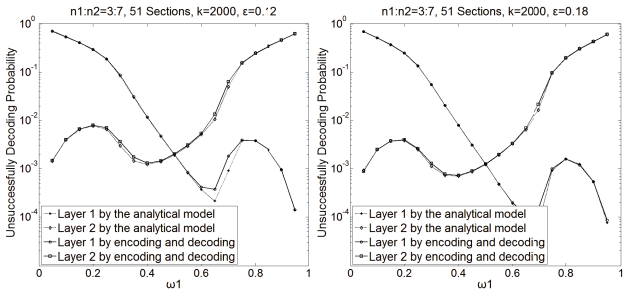

Fig. 13. Performance of the analytical model at 51 sections.

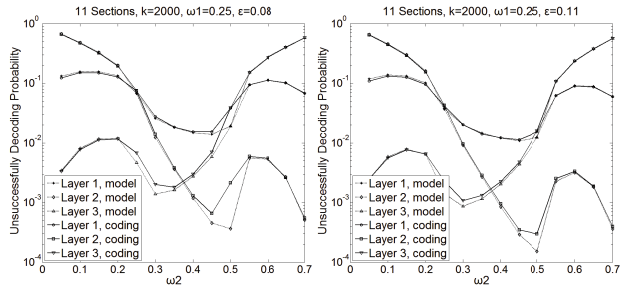

Fig. 14. Performance of the analytical model at three layers, lower overhead.

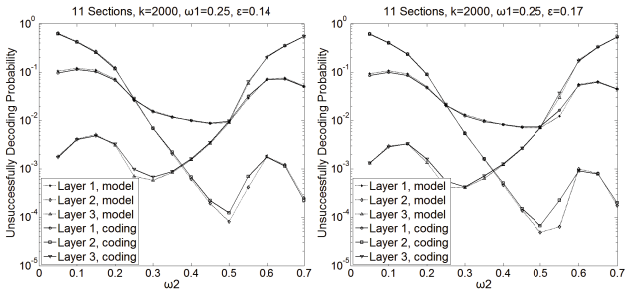

Fig. 15. Performance of the analytical model at three layers, higher overhead.

Figs. 11–13 are not symmetric, due to the uneven distribution of n_1 and n_2 in the simulations.

The second simulations were for data with three layers ($N = 3$), where $n_1 : n_2 : n_3 = 1 : 1 : 2$, and the degree distribution used in Raptor codes for coded blocks was also applied. The weighing vector $(\omega_1,\ \omega_2,\ \omega_3) = (0.25, \omega_2, 0.75 - \omega_2)$ with different ω_2 (0.05–0.7, with a step size $= 0.05$) was simulated. The probabilities of unsuccessfully decoding the scenarios at various weighing vectors and coding overhead are shown in Figs. 14 and 15.

In scalable video coding, the bit rates of different video layers can be affected by the rate distortion performance as well as by the distribution of available bandwidth from a server to its clients. The third simulations were for data with two layers ($N = 2$) at different $n_1 : n_2$ ratios. In addition to $n_1 : n_2 = 3 : 7$, the experiments at 5:5 and 7:3 were conducted as well. The performance of expanding window fountain codes [18] is also included for performance comparison. The expanding window fountain codes were used on the scalable video streams in [22] due to its support of unequal error protection.

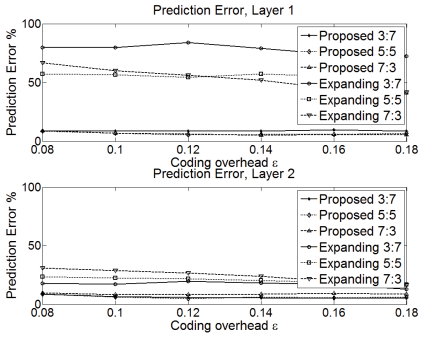

Fig. 16. Prediction performance of the analytical models for the proposed method and the expanding window fountain codes. $k = 2000$, 11 sections.

The prediction performance of the analytical models for the proposed LMRCs and the analytical model for the expanding window fountain codes presented in [18] and [22] is shown in Fig. 16. The prediction model for expanding window scheme is described in [22, Eq. (1)]. For the expanding window fountain codes, various probabilities Γ of selecting layer 1 window are used and the mean prediction error is calculated for a given coding overhead.

The performance of the proposed model is consistently good with respect to different $n_1 : n_2$ ratios, while the performance of the analytical model for the expanding window fountain codes is not. In general, the prediction ability for the expanding window fountain codes suffers obviously for the inner window, layer 1. The reason could include: the analytical model for the expanding window fountain codes cannot handle smaller size of windows well, and the prediction is not accurate at lower Γ when the coded symbols produced for the inner window are not enough. On the other hand, the ratio is not a sensitive factor and has a little effect on the prediction errors caused by the proposed model. The prediction errors caused by the expanding window scheme are higher and the errors fluctuate from one ratio to another. The analytical models for these two unequal error protection schemes can be used to determine the parameters, such as the weighing vectors in the proposed method and the Γ in the expanding window scheme. Therefore, the accuracy of an analytical model is rather important.

For expanding window fountain codes, the prediction accuracy at $n_1 : n_2 = 5 : 5$ is better than at the other two ratios. At $n_1 : n_2 = 5 : 5$, the performance comparison of unsuccessfully decoding probability is shown in Fig. 17. At lower Γ, the prediction performance of the expanding window fountain codes is significantly worse, especially when $\Gamma \leq 0.3$. It is interesting to observe that at larger Γ, the probability for layer 1 can be quite low, while the unsuccessfully decoding probability is extremely high for layer 2. Even if the extremely worse recovery of layer 2 is tolerable, the advantage of having lower failure probability of layer 1 will become less important if an outer code can be applied to the proposed method and the expanding window fountain codes.

B. Accuracy Analysis of the Model for the Expected Decompressible Video Frames

The performance of the model for the expected decompressible video frames is examined in this section. The model of

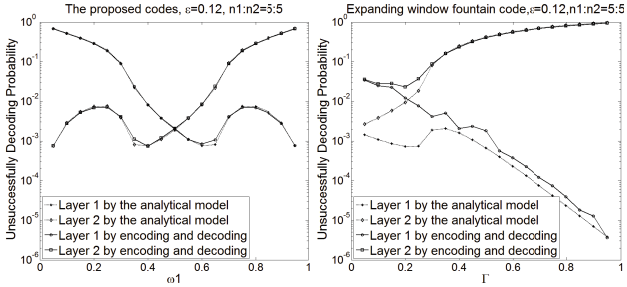

Fig. 17. Performance of the proposed codes and the expanding window fountain codes. $k = 2000$, 11 sections.

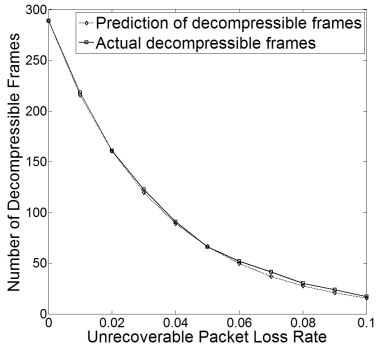

Fig. 18. Predicted decompressible frames in comparison with the actual decompressed frames.

the expected decompressible video frames is obtained using the dependency characteristic of scalable video coding. This prediction model can be used to determine the protection strength of the proposed LMRCs by maximizing the expected decompressible frame number, as described in Section III. Since the proposed LMRCs are based on rateless codes, the randomness of the generation of coded blocks will alleviate the difference due to different channel models, as long as the average packet loss is at the same level.

A video of soccer at 325×288 was compressed using the scalable extension of MPEG-4 AVC into five temporal layers without the protection of channel coding. The GOP size was 16 frames and each key picture was encoded as an IDR picture. The packet size was 1500 bytes and the influence of different unrecoverable packet loss rates generated using uniform distribution was studied. The parameter information (sequence parameter set and picture parameter set) required by MPEG-4 AVC and the first frame were assumed to be received correctly.

The prediction results of the model for the expected decompressible video frames are shown in Fig. 18. The prediction results of the model are observed to be very accurate.

C. Performance Comparison of the Proposed LMRCs

In this section, we will use the proposed LMRCs to protect video compressed with the scalable extension of MPEG-4 AVC. The proposed method determines the favorable probabilities $(p_1, p_2, \ldots, p_n)$ by maximizing the expected decompressible frame number at a given bandwidth, as shown in (15).

Three other methods were also included in the experiments. First, the rateless codes with an unequal error protection

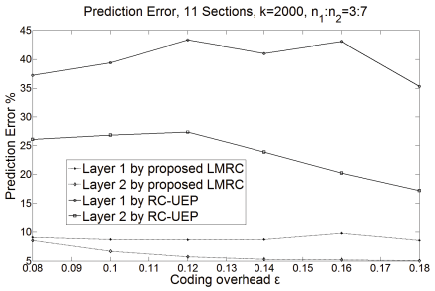

Fig. 19. Prediction performances of the analytical models: LMRC and RC-UEP.

property (RC-UEP) [19], where the optimal parameter k_M for RC-UEP was determined at a given bandwidth by performing an exhaustive search on k_M such that the predicted decoding probability obtained from the prediction model described in [19] will result in the maximal number of expected decompressible frames according to the model described in Section III. The other two methods were the generic rateless codes: RC Orig, which does not differentiate between two layers, and RC layer, which treats the two layers in each section as two different independent virtual sections. In RC Orig, a fixed number of coded blocks for each section were generated by choosing a number of message blocks to perform XOR operations with. In RC layer, the number of coded blocks for each virtual section is proportional to the number of message blocks within the virtual section, while the number of total coded blocks is identical to those produced by other methods.

The predictor errors of the analytical models, RC-UEP, and the proposed LMRC, are shown in Fig. 19 with respect to increasing coding overhead. The simulations were for two layers ($N = 2$) and $n_1 : n_2 = 3 : 7$. The prediction error of each layer between the predicted results using the corresponding analytical model and the experimental results by actually coding and decoding is calculated by (16). From the results, the accuracy of the proposed analytical model is much better.

All four methods described above follow the degree distribution used in Raptor codes for coded blocks [6]. In this simulation, the video soccer at 325 × 288 was compressed into two video layers where the compressed size of the first layer was about one-third of the total bit rate. The number of message blocks k in a section was 2033. There are 1825 frames encoded using MPEG-4 AVC scalable extension with GOP = 16. For each GOP, two frames are used for layer 1 and 14 frames are for layer 2.

The experimental results of all four methods with respect to increasing coding overhead (which is translated from given bandwidth) are shown in Figs. 20 and 21. The performance of the proposed LMRCs is shown to greatly improve on the other methods, especially in limited bandwidth scenarios. Rateless coding can achieve better coding efficiency with a larger number of message blocks (in a section). Therefore, the performance of RC Orig is better than RC Layer. The proposed method with an accurate analytical model connects two sections together such that N windows can help to decode message blocks in a section, and is observed to result in

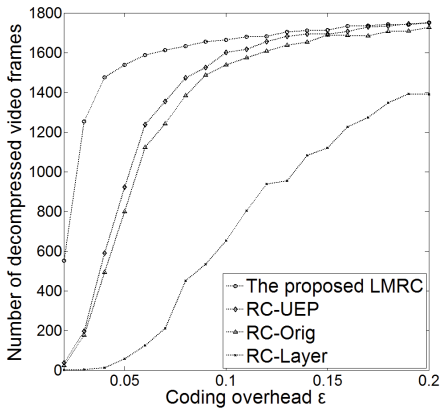

Fig. 20. Number of decompressed video frames versus the coding overhead using different codes.

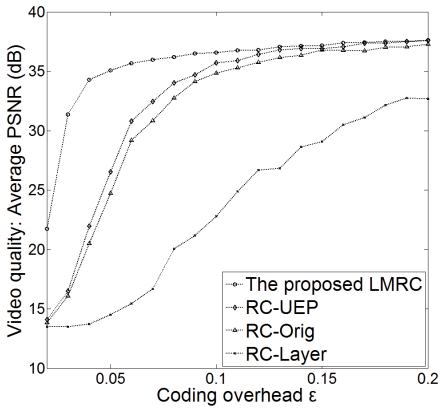

Fig. 21. Received video quality in PSNR versus the coding overhead using different codes.

the best decoded video quality. The average PSNR values reported in Fig. 21 are computed by calculating the per-frame PSNR first and averaging the PSNR values per frame in the same reconstructed video. If a video frame cannot be decompressed, simple temporal repetition technique is used for error concealment.

V. Conclusion

In this paper, we proposed a novel unequal error protection scheme, called LMRCs, for data transmission with multiple priorities. In the proposed N-cycle layer-aligned overlapping structure, each layer in a section is protected by N windows, and the protection strength can be controlled through the vector of favorable probabilities. The length in each of the N different priorities can vary. An analytical model for the LMRCs was also developed. The analytical model was shown to achieve high prediction accuracy, compared with the much lower accuracy of the prediction model for rateless codes with unequal error protection in the literature. In addition, our analytical model can be executed efficiently. Together with the model to predict the expected decompressible video frames at a given packet loss rate, the vector of favorable probabilities for the proposed LMRCs can be determined for given channel conditions. We demonstrated, using simulated results, that the analytical model is accurate and that the

LMRCs are suitable for layered streaming, especially when the section size is limited.

The proposed LMRCs in this paper do not use outer codes as in raptor codes. However, applying outer codes can help to alleviate the issue of higher error floor at a small price of additional computation and coding overhead. Even though streaming of multimedia can usually withstand a certain level of data loss, as a future work, it should be quite intriguing to extend the proposed LMRCs to the layer-aligned multipriority raptor codes and develop an algorithm to determine proper code rates of outer codes for different video layers with different priorities. In addition, if the ratio can be decided freely and a utility function can be formulated to quantitate layer priorities, similar to the expected decompressible video frames introduced in Section III, an optimal ratio can be studied to further increase the performance of streaming scalable videos that are protected using the proposed method.

REFERENCES

[1] S. McCanne, V. Jacpbson, and M. Vetterli, "Receiver-driven layered multicast," *ACM SIGGCOM Comput. Commun. Rev.*, vol. 26, no. 4, pp. 117–130, Oct. 1996.

[2] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[3] G. Fairhurst and L. Wood, "Advice to link designers on link Automatic Repeat reQuest (ARQ)," Int. Eng. Task Force, Geneva, Switzerland, Tech. Rep. RFC 3366/BCP 0062, Aug. 2002.

[4] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, Jun. 1960.

[5] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci.*, Nov. 2002, pp. 271–280.

[6] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[7] M. Mitzenmacher, "Digital fountains: A survey and look forward," in *Proc. IEEE Inf. Theory Workshop*, Oct. 2004, pp. 271–276.

[8] *Specification Text for Systematic Raptor Forward Error Correction*, 3GPP, Sophia Antipolis Cedex, France, Apr. 2005.

[9] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[10] J. S. Plank and M. G. Thomason, "On the practical use of LDPC erasure codes for distributed storage applications," Dept. Comput. Sci., Univ. Tennessee, Knoxville, TN, USA, Tech. Rep. UT-CS-03-510, Sep. 2003.

[11] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[12] P. Maymounkov, "Online codes," NYU, New York, NY, USA, Tech. Rep. TR2003-883, 2002.

[13] M. G. Luby, M. Mitzenmacher, and M. A. Shokrollahi, "Analysis of random processes via and-or tree evaluation," in *Proc. 9th Annu. ACM-SIAM Symp. Discrete Algorithms*, Oct. 1998, pp. 364–373.

[14] M. C. Bogino, P. Cataldi, M. Grangetto, E. Magli, and G. Olmo, "Sliding-window digital fountain codes for streaming of multimedia contents," in *Proc. IEEE ISCAS*, May 2007, pp. 3467–3470.

[15] P. Cataldi, M. Grangetto, T. Tillo, E. Magli, and G. Olmo, "Sliding-window raptor codes for efficient scalable wireless video broadcasting with unequal loss protection," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1491–1503, Jun. 2010.

[16] W.-C. Wen, H.-F. Hsiao, and J.-Y. Yu, "Dynamic FEC-distortion optimization for H.264 scalable video streaming," in *Proc. IEEE Workshop Multimedia Signal Process.*, Oct. 2007, pp. 147–150.

[17] W.-C. Wen and H.-F. Hsiao, "Hierarchical optimization of cascading error protection scheme for H.264 scalable video streaming," *J. Signal Process. Syst.*, vol. 62, no. 3, pp. 359–371, Mar. 2011.

[18] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, and R. Piechocki, "Expanding window fountain codes for unequal error protection," *IEEE Trans. Commun.*, vol. 57, no. 9, pp. 2510–2516, Sep. 2009.

[19] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Rateless codes with unequal error protection property," *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1521–1532, Apr. 2007.

[20] S. Ahmad, R. Hamzaoui, and M. M. Al-Akaidi, "Unequal error protection using fountain codes with applications to video communication," *IEEE Trans. Multimedia*, vol. 13, no. 1, pp. 92–101, Feb. 2011.

[21] K. T. Wallenius, "Biased sampling: The non-central hypergeometric probability distribution," Ph.D. dissertation, Dept. Statistics, Stanford Univ., Stanford, CA, USA, 1963.

[22] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, and Z. Xiong, "Scalable video multicast using expanding window fountain codes," *IEEE Trans. Multimedia*, vol. 11, no. 6, pp. 1094–1104, Oct. 2009.

Hsu-Feng Hsiao (M'05) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan; the M.S. degree in electrical engineering from National Chiao-Tung University, Hsinchu, Taiwan; and the Ph.D. degree in electrical engineering from University of Washington, Seattle, WA, USA, in 1995, 1997, and 2005, respectively.

He was an Engineering Officer with the Communication Research Laboratory, Ministry of National Defense, Taipei, from 1997 to 1999. From 2000 to 2001 he was a Software Engineer with HomeMeeting, Redmond, WA, USA. He was a Research Assistant with the Department of Electrical Engineering, University of Washington, in 2005. He has been an Assistant Professor with the Department of Computer Science, National Chiao-Tung University, since 2005. His research interests include multimedia signal processing and wired/wireless communications.

Yong-Jhih Ciou received the B.S. degree in computer science from Chung Yuan Christian University, Chung Li, Taiwan, and the M.S. degree in computer science from National Chiao-Tung University, Hsinchu, Taiwan, in 2009 and 2011, respectively.

He has been with Trend Micro, Taipei, Taiwan, since 2012.