Intermediate Performance Analysis of Growth Codes

Nikolaos Thomos, *Member, IEEE*, Rethnakaran Pulikkoonattu, *Member, IEEE*, and Pascal Frossard, *Senior Member, IEEE*.

Abstract—Growth codes are a subclass of Rateless codes that have found interesting applications in data dissemination problems. Compared to other Rateless and conventional channel codes, Growth codes show improved intermediate performance which is particularly useful in applications where performance increases with the number of decoded data units. In this paper, we provide a generic analytical framework for studying the asymptotic performance of Growth codes in different settings. Our analysis based on Wormald method applies to any class of Rateless codes that does not include a precoding step. We evaluate the decoding probability model for short codeblocks and validate our findings by experiments. We then exploit the decoding probability model in an illustrative application of Growth codes to error resilient video transmission. The video transmission problem is cast as a joint source and channel rate allocation problem that is shown to be convex with respect to the channel rate. This application permits to highlight the main advantage of Growth codes that is improved performance (hence distortion in video) in the intermediate loss region.

Index Terms—Rateless Codes, Growth Codes, data dissemination, error resilient, video streaming.

I. INTRODUCTION

Rateless codes [1] have been proposed as an efficient method to design decentralized data dissemination systems without employing expensive routing protocols as they do not require any source coordination. This becomes feasible due to the rateless property that allows on-the-fly generation of arbitrary numbers of packets. The most successful Rateless codes are certainly the Raptor codes [2] that are based on LT codes [3]. Raptor codes perform close to maximum distance separable (MDS) codes and only require slightly more symbols than the number of source symbols for successful recovery. These appealing properties, combined with linear encoding and decoding times, have conducted the Raptor codes to be adopted by recent communication protocols like 3GPP [4] and DVB-H [5].

In practice, Rateless codes need a feedback channel for transmitting a termination message from sinks to sources in order to notify that the data decoding is successful. Unfortunately, in many scenarios a feedback channel is not available. Thus, the transmission rate should be predetermined by taking into account the network conditions. However, the channel conditions vary dynamically, may deteriorate fast and become rapidly different from the predicted ones. This leads to inaccurate estimation of the transmitted rate, and thus to inefficient exploitation of the network resources with possibly decoding failures. If the decoder receives an insufficient number of packets for forming a full rank decoding matrix, common Rateless codes have unfortunately poor recovery properties; they are characterized by on-off performance, which means that the transmitted data can be either fully recovered or not at all. Partial decoding is possible, but in general it is limited. Growth codes [6] have been proposed to enhance the intermediate performance of Rateless codes. They permit recovery of more data than conventional codes when the available data is insufficient for complete message recovery. The improved intermediate performance of Growth codes has attracted the attention of the data communication community, especially for the transmission of unequal importance data streams as proposed in [7].

In this paper, we propose a new and complete decoding performance analysis for Growth codes based on the Wormald method [8] that is typically used for the analysis of graph stochastic processes. Contrarily to previous studies, the proposed method for characterizing the decoding performance of Growth codes is generic and valid for various codeblocks[1] size and accurately predicts the decoding probability for both intermediate and high reception rates (when more symbols than the number of source symbols have been received). It therefore provides a more generic performance analysis than asymptotic performance bounds with arbitrary long blocks such as those derived in [9]. An accurate performance analysis of small and medium blocksize is very important in the design of data communication systems and our novel performance analysis shows that the decoding performance of Growth codes can be approximated by an exponential model. We apply our analysis in an illustrative example in video communications. We combine our model with the video distortion model proposed in [10] and cast a joint source and channel rate allocation problem for optimized error resilient transmission. This problem is shown to be convex and the optimal source and channel rate can be determined by methods

N. Thomos is with the Signal Processing Laboratory 4 (LTS4), Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, and the Communication and Distributed Systems laboratory (CDS), University of Bern, Bern, Switzerland (e-mail: nikolaos.thomos@epfl.ch).

R. Pulikkoonattu is with Broadcom Corp, USA (e-mail: rethna@broadcom.com).

P. Frossard is with the Signal Processing Laboratory 4 (LTS4), Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland (e-mail: pascal.frossard@epfl.ch).

[1]In this paper, the term codeblock is used to denote a group of packets that are encoded together by a channel code.

such as dynamic programming and Lagrangian optimization. Our design comes close to the concepts of [11], [12] where unequal error protection Rateless codes have been designed specifically for streaming. We however here concentrate on the intermediate of the channel codes, which is one of the main advantages of Growth codes. We show that these codes provide an interesting solution for error resilient video streaming with graceful degradation of quality when channel conditions deteriorate.

The paper is organized as follows. In Section II, we present the general principles of Rateless codes focusing especially on Growth codes. We analyze Growth codes recovery performance using the Wormald method, discuss the expected Growth codes behavior and provide a model for the symbol recovery probability in Section III. In Section IV, we cast an optimization problem for determining the optimal source and channel rate allocation in an illustrative example of video transmission. Finally, conclusions are drawn in Section V.

II. RATELESS CODES

A. General structure

Data dissemination systems usually employ Reed Solomon (RS) codes [13] to cope with packet erasures on communication channels. RS codes can recover the transmitted data only when the received set of packets is equal to the number of source symbols k. However, the application of RS codes is limited by their decoding complexity that grows quadratically with the block size. Rateless codes solve this problem in applications with complexity or delay constraints as they offer linear encoding times. Linear decoding times can be further achieved when the degree distribution is carefully designed and decoding is performed by means of belief propagation. These codes however incur a small performance penalty ϵ and require slightly more packets than the number of source packets k for decoding, (i.e., the minimal number of packets for decoding is $(1+\epsilon)\cdot k$). The decoding probability increases with the number of received packets since the probability to form a full rank system of packets at decoder increases with the number of packets. Rateless codes generally do not have a pre-determined code rate and they generate as many symbols as needed by each client for successful decoding. This permits better exploitation of the available network resources as systems employing Rateless codes can easily cope with network dynamics; inaccurate estimation of the channel conditions do not directly lead to over-protecting or under-protecting the source data as in rate-optimized codes.

In more details, the Rateless encoded symbols are generated by combining (XOR-ing) source packets selected uniformly at random. The number of combined packets (the degree of the corresponding codeword) is determined by the degree distribution function $\Omega(x) = \sum_{i=1}^{d_{\max}} \Omega_i \cdot x^i$, where Ω_i denotes the probability of generating a symbol with degree i (a symbol that is the result of XOR-ing i data symbols). The parameter $d_{\max}$ is the maximum allowable symbol's degree which cannot exceed the number of source symbols. If x_i stands for the ith source packet and y_j is the jth transmitted symbol respectively,

we can write

$$y_j = \sum_{l=1}^{d_j} \bigoplus x_l,$$

where d_j is the degree of the jth symbol and $\sum \bigoplus$ is the bitwise XOR operation. Rateless codes have an implicit structure as the decision about the number of packets to be combined for the generation of the y_j is made by sampling $\Omega(x)$; the identity of the combined packets is selected uniformly at random. Due to the implicit structure of Rateless codes a small header called "ESI" is appended to each packet. It conveys information to the decoder about the packets that have been combined for the generation of encoded packets. The ESI is usually the seed of a pseudorandom generator used for generating the encoded packet. Please note that the terms "packet" and "symbol" are used interchangeably throughout the analysis.

Next, we focus on Growth codes that are a particular class of Rateless codes with advanced intermediate decoding performance and sustained performance at high rates. Such characteristics are particularly interesting for transmission of time-constrained data that can benefit from partial decoding.

B. Growth codes

Popular Rateless codes such as LT codes and Raptor codes have small overheads and permit decoding from a set of symbols than is slightly larger than the set of source packets. These codes have however poor intermediate performance when the number of symbols is not sufficient for perfect decoding. On the other hand, Growth codes [6] offer better intermediate performance as partial recovery is possible, but are characterized by a larger overhead ϵ than LT or Raptor codes.

We give now more details about the design of Growth codes. The encoding procedure of Growth codes is similar to that of LT codes. Symbols are generated according to a degree distribution function $\Omega(x)$. The design of $\Omega(x)$ is quite intuitive, i.e., when few packets have arrived to a client, it is better to receive packets of degree one that permit immediate decoding. Instantaneous decodability of packets of degree one however decreases with the number of received packets like in the coupons collector problem [14]. Indeed, the probability that a packet is a duplicate increases with the number of received packets with degree one [6].

Based on this intuition, the degree of the encoded symbols progressively increases until sinks are able to recover the transmitted data content. The Growth codes degree distribution is thus given as

$$\Omega(k) : \Omega_i^* = \max\left(0, \min\left(\frac{K_i - K_{i-1}}{k}, \frac{k - K_{i-1}}{k}\right)\right) \quad (1)$$

where the parameter K_i is computed by the following recursive relation [6] for the kth transmitted symbol:

$$K_j = K_{j-1} + \sum_{i=R_{j-1}}^{R_j - 1} \frac{\binom{k}{j}}{\binom{i}{j-1}(k-i)} \quad (2)$$

with $R_j = \frac{jk-1}{j+1}$ and $K_1 = \sum_{i=0}^{R_1-1} \frac{k}{k-i}$. In order to recover R_j symbols, the receiver has to get K_j symbols in expectation. The Growth codes decoding probability as a function of the number of received packets k is determined by

$$P_{r,d}(k) = \begin{cases} \frac{(k-r+1)}{k}, & d = 1 \\ \frac{(k-r+1)\binom{r-1}{d-1}}{\binom{k}{d}}, & d = 2,\ldots,r \\ 0, & d > r \end{cases} \quad (3)$$

It corresponds to the probability of decoding a symbol of degree d when r symbols have been already decoded. The Growth codes, encoding procedure, can be summarized as follows

---

Procedure 1 Growth Codes encoding

1: Choose randomly the degree d of the LT encoded symbol by sampling $\Omega(d)$.
2: Choose uniformly d distinct symbols.
3: Combine the symbols by XOR operations

On the client side, two algorithms can be used for Growth codes decoding namely the Decoder-S and the Decoder-D [6]. The Decoder-D ignores the packets that are not immediately useful, *i.e.*, the packets whose distance to the recovered packets is larger than one, while Decoder-S stores in a list all received packets even if they are not immediately useful and checks the list of packets as the decoding process progresses. Although Decoder-D has slightly inferior decoding performance compared to that of Decoder-S, it is preferable for low-cost receivers performing on-the-fly decoding. In our analysis, we restrict our attention only to the Decoder-D that is summarized in Procedure 2.

Procedure 2 Growth Codes decoding: Decoder-D algorithm

1: **if** a symbol y with $d = 1$ arrives at client **then**
2:    Insert the symbol y in $\mathcal{D}$.
3: **end if**
4: **while** $\mathcal{D} \neq \emptyset$ **do**
5:    Select a symbol x from $\mathcal{D}$.
6:    **if** $x \notin \mathcal{L}$ and $\text{dist}(x, \mathcal{L}) = 1$ **then**
7:       XOR x with the symbols $\in \mathcal{L}$ that used to generate the symbol x.
8:       Insert the recovered symbol in $\mathcal{L}$.
9:       **if** in the meanwhile a new instantaneous decodable symbol y arrives at client **then**
10:          Insert the symbol y in $\mathcal{D}$.
11:       **end if**
12:    **end if**
13: **end while**

In the decoding algorithm, $\mathcal{D}$ is the ripple of the Growth codes, $\mathcal{L}$ is a list that contains the decoded symbols and $\text{dist}(x, \mathcal{L}) = 1$ means that the symbol is instantaneously decodable, *i.e.*, an original source symbol can be recovered. Upon receiving a packet, the client examines whether the packet is instantaneously decodable and is not already in $\mathcal{L}$.

When it is true, the packet is decoded and then inserted in $\mathcal{L}$, otherwise the packet is deleted. This procedure continues as long as the source is decoded or $\mathcal{D}$ is empty.

III. ANALYSIS OF GROWTH CODES PERFORMANCE

A. Wormald method

The Wormald method [15] is an analytical tool that is used to study the expected behavior of stochastic processes. It is based on the idea that a system stays close to the expected behavior [8] after a series of random steps, with very high probability. Such a behavior can further be determined by a set of differential equations.

Let us consider a graph random process $G(t)$. The process starts with graph $G(0)$ from which edges are repeatedly removed according to a probabilistic rule that is known a priori. This removal procedure results in a probabilistic set of sequences $G(0), G(1), \ldots, G(t)$, where $G(t)$ denotes the t-th graph in that process. When no edge can be removed from $G(t)$, the process becomes stationary. Thus, we have $G(t+1) = G(t)$ and this is the final graph of this stochastic process. Usually this happens for large t ($t \to \infty$). For many systems the above procedure however terminates after a finite number of steps, *i.e.*, at time $t = T_d$; this forms a family of graphs $G(0), G(1), \ldots, G(T_d)$. The Wormald method studies the behavior of such a sequence of processes.

Since Decoder-D algorithm for Growth codes decoding is described by a stochastic process (*i.e.*, Markov process), the Wormald method can be used for the analysis of its performance. In particular, the residual Tanner graph $G(t)$ during decoding is characterized by a set of pairs $(V_i(t), C_i(t))$, where $V_i(t)$ and $C_i(t)$ denote the total number of edges connected with variable nodes and respectively check nodes of degree i at time t. In the Growth Codes analysis, the graph actually contains k variable nodes where k is equal to the number of source symbols.

Next, we discuss the appropriateness of using Wormald theorem to characterize the decoding performance of Growth codes. We start by giving the conditions that Growth codes decoding should fulfill in order to use the Wormald method. Then, we use this method to describe the evolution of the decoding process.

B. Wormald theorem

In this section, we describe the Wormald theorem and present the conditions that have to be satisfied to apply the Wormald method [16]. It should be noted that the Wormald theorem can also be used for analyzing other Rateless codes, as long as their encoding does not include any precoding step.

Let us consider that $G(t)$ has a state space $\{0, \ldots, \theta\}^d$, $d \in \mathbb{N}$ and a probability space $\mathcal{S}$. Consider a sequence $\{G^m(t)\}_{m>1}$ of a Markov random process where $G_i^m(t)$ is the i-th component of $G^m(t)$. Denote a subset $\Gamma \subset \mathbb{R}^{d+1}$ containing the vectors $[0, g_1, \ldots, g_d]$ such that,

$$\mathbb{P}\left(\frac{G_i^{(m)}(t=0)}{m} = g_i\right) > 0, \forall i \in [1, d], m > 1 \quad (4)$$

Let f_i be functions from $\mathbb{R}^{d+1}$ to $\mathbb{R}^d$, satisfying the following conditions:

1) For $t < m$, there exists a constant c_i^m such that,

$$\left| G_i^{(m)}(t+1) - G_i^{(m)}(t) \right| \leq c_i^m, 1 \leq i \leq d \quad (5)$$

2) For $t < m$ and $\forall i \in [1, d[$,

$$\mathbb{E}\left[G_i^{(m)}(t+1) - G_i^{(m)}(t) | G^{(m)}(t) \right]$$
$$\triangleq f_i\left(\frac{t}{m}, \frac{G_1^{(m)}(t)}{m} \ldots, \frac{G_d^{(m)}(t)}{m} \right) \quad (6)$$

3) $f_i, \forall i \leq d$ is Lipschitz continuous function on the intersection of Γ with the half space $\{(t, g_1, \ldots, g_d) : t \geq 0\}$, *i.e.*, if $x, y \in \mathbb{R}^{d+1}$ belong to this intersection, then there exists a Lipschitz constant ζ such that,

$$|f_i(x) - f_i(y)| \leq \zeta \sum_{j=1}^{d+1} |x_j - y_j| \quad (7)$$

The above conditions are respectively the boundedness, the trend and the Lipschitz conditions. The boundedness condition implies that the function is bounded. The trend condition imposes that we can describe the process with a time series without any knowledge of the serial correlation between $G^m(t)$. Finally, the Lipschitz condition means that the functions f_i are not steep and bounded by ζ. Under these conditions, the following holds true:

1) For the vector $[t, g_1, g_2, \ldots, g_d] \in \Gamma$, the system of differential equations

$$\frac{\partial g_i}{\partial \tau} = f_i(\tau, g_1, \ldots, g_d), 0 \leq i \leq d \quad (8)$$

has a unique solution for $g_i(\tau) : \mathbb{R} \to \mathbb{R}$ in Γ with the initial condition $g_i(0) = x_i, 1 \leq i \leq d$.

2) There exists a strictly positive constant δ such that

$$\mathbb{P}\left(\left| \frac{G_i^{(m)}(t)}{m} - g_i\left(\frac{t}{m}\right) \right| \geq \delta m^{-\frac{1}{6}} \right) < \frac{dm^{\frac{2}{3}}}{e^{\frac{m^{\frac{1}{3}}}{2}}} \quad (9)$$

for $0 \leq t \leq m\tau_{\max}$ and for each i. The term g_i is the unique solution obtained by solving Eq. (8) with the initial conditions $g_i(0) = \mathbb{E}\left[\frac{G_i^{(m)}(t=0)}{m}\right]$ and $\tau_{\max} = \tau_m$ is the supremum of those τ to which the solution can be extended, under some boundedness criteria [8], [15].

In other words, Eq. (9) states that, when m is big enough, each realization of the process $G_i^{(m)}(t)$ is close to the (unique) solution of Eq. (8), with high probability. This permits us to use a set of differential equations to describe the decoding process. A formal proof of the applicability of the Wormald theorem on the Decoder-D algorithm can be further found in [15].

C. Expected Growth Codes behavior

We have seen above that a set of differential equations can describe the expected decoding performance as the decoding procedure evolves. In particular, the Decoder-D algorithm proceeds as long as it can find check nodes of degree one

to process, otherwise the decoding halts. At each decoding step a check node with degree one is randomly chosen to be eliminated. The outgoing edge of the selected node is connected with one of the $\sum_j V_j(t)$ variable nodes, while the total number of edges $E(t)$ in the (residual) graph G_t is equal to $\sum_j jV_j(t)$. Thus, the probability that an edge is connected to a variable node of degree i is $\frac{iV_i(t)}{\sum_j jV_j(t)}$.

The expected decrease in the number of variable nodes with degree i at time t is then expressed as,

$$\mathbb{E}[V_i(t+1) - V_i(t)|V(t), E(t)] = -\frac{iV_i(t)}{\sum_j jV_j(t)} \quad (10)$$

Similarly to the variable nodes, we can determine the expected decrement in check nodes degree. When an edge of a check node of degree one is removed, the number of check nodes of degree one is reduced by one. If the removed edge is connected to a check node of degree $i+1$, the residual degree changes from $i+1$ to i with probability $\frac{(i+1)C_{i+1}(t)}{\sum_j jC_j(t)}$. Therefore, in expectation, $-1 + \frac{\sum_j j^2 V_j(t)}{\sum_j jV_j(t)}$ edges are removed.

The expected decrease of check nodes with degree i is written as,

$$\mathbb{E}[C_i(t+1) - C_i(t)|V(t), E(t)] \approx$$
$$\frac{i[C_{i+1}(t) - C_i(t)]}{\sum_j jV_j(t)} \frac{\sum_j j(j-1)V_j(t)}{\sum_j jV_j(t)}, i \geq 2 \quad (11)$$

The examined node is removed from the set of check nodes of degree one, and the expected decrease in degree is given as

$$\mathbb{E}[C_1(t+1) - C_1(t)|V(t), E(t)] \approx$$
$$-1 + \frac{[C_2(t) - C_1(t)]}{\sum_j jV_j(t)} \frac{\sum_j j(j-1)V_j(t)}{\sum_j jV_j(t)} \quad (12)$$

Since we assume that the process stays close to its expected behavior, as the decoding algorithm corresponds to a stochastic process, we can drop the expectation and write $v_i(\tau) \approx \frac{V_i(t)}{nV'(1)}$ and $c_i(\tau) \approx \frac{C_i(t)}{nV'(1)}$, where $\tau \triangleq \frac{t}{nV'(1)}$ denotes the normalized time. $V(t)$ is the degree distribution function of the variable nodes and $V(1)'$ is the average degree of $V(t)$. Formally, c_i (or v_i) describes the fraction of check nodes (or variable nodes) connected to i edges. Then, we make the following approximations

$$\frac{V_i(t+\delta t) - V_i(t)}{\delta t} \approx \frac{\partial v_i(\tau)}{\partial t} \quad (13)$$

and

$$\frac{C_i(t+\delta t) - C_i(t)}{\delta t} \approx \frac{\partial c_i(\tau)}{\partial t} \quad (14)$$

We can now express the differential equations involving variable and check nodes degree distributions, which govern the behavior of the decoder:
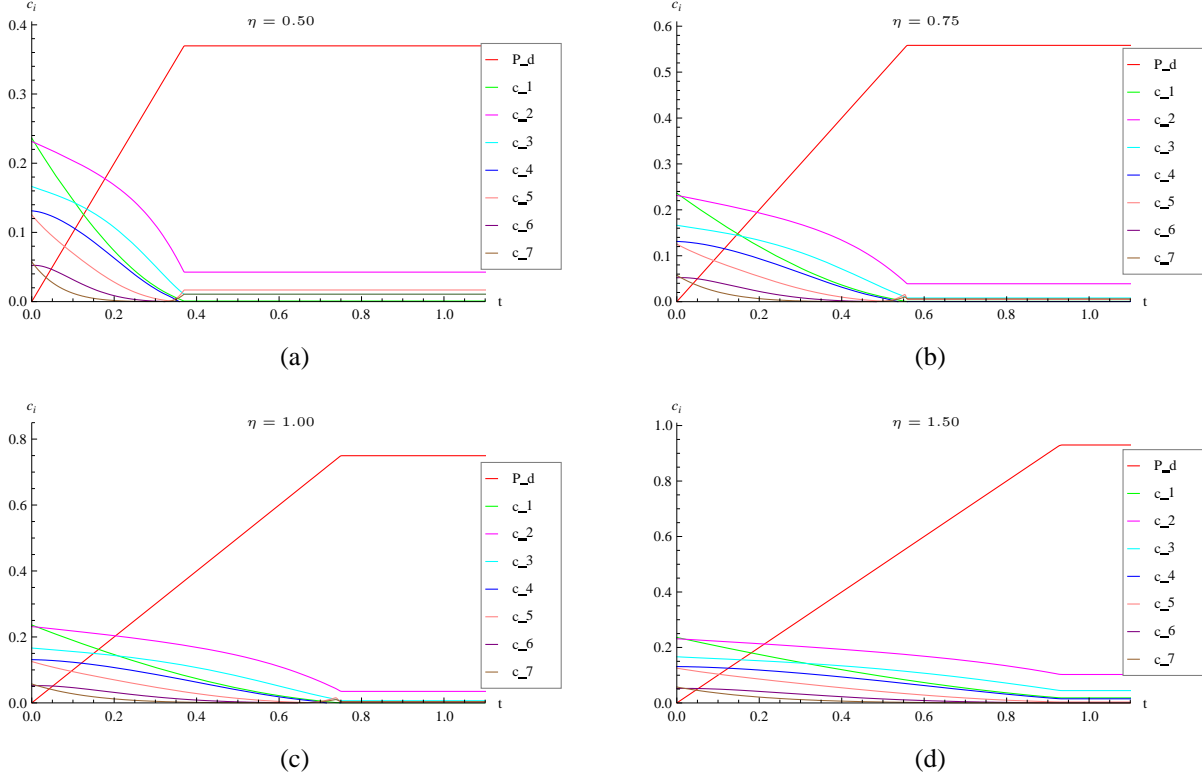
Fig. 1. Temporal evolution of the fraction $c_i(t)$ of check nodes of degree i for various reception rates η: (a) $\eta = 0.5$, (b) $\eta = 0.75$, (c) $\eta = 1.0$, and (d) $\eta = 1.5$.

$$\frac{\partial v_i(t)}{\partial t} = -i\frac{v_i(t)}{\sum_{j=1}^{d_v^{\max}} jv_j(t)}, \quad (15)$$

$$\frac{\partial c_i(t)}{\partial t} = [c_{i+1}(t) - c_i(t)]\frac{i\sum_j^{d_c^{\max}} j(j-1)c_j(t)}{\sum_{j=1}^{d_c^{\max}} jc_j(t)}, \quad (16)$$

$$\frac{\partial c_1(t)}{\partial t} = -1 + [c_2(t) - c_1(t)]\frac{\sum_j^{d_c^{\max}} j(j-1)c_j(t)}{\sum_{j=1}^{d_c^{\max}} jc_j(t)} \quad (17)$$

The parameters $d_c^{\max}$ and $d_v^{\max}$ are the maximum degrees of the check and variable nodes respectively; we typically set them to a very large value (theoretically infinity). Eq. (15) shows the expected decrease in the number of variable nodes of degree i. Eq. (16), which is valid $\forall i, i \geq 2$, and Eq. (17) describe the residual degree changes from $i+1$ to i when an edge connected with a check node of degree $i+1$ is removed. The first term (-1) in Eq. (17) is due to the removal of an edge from the check node of degree one.

The solution of these differential equations with appropriate initial conditions describe the evolution of the degree distribution in the check and variable nodes. For Growth codes, we have that the initial check nodes distribution of the check nodes is $\Omega(d)$ in Eq. (1). Then, in the limit of large number of source packets k, the variable nodes follow a Poisson distribution [17], since the source symbols are selected uniformly at random during the encoding process.

A closed form solution to these equations is non-trivial to determine. However, we can solve them numerically. For verifying our analysis we examine the degree evolution of the nodes during decoding with a Decoder-D algorithm. We choose the receiving symbol rate η (normalized to the source rate) in the range $[0, 1.5]$ (i.e., for $\eta < 1$ we have intermediate performance). The evolution of various degree edges at different η values is shown in Fig. 1 where the number k of source packets is set to 1000. The fraction of variable nodes of degree 0 corresponds to the recovery probability P_d, i.e., $P_d = v_0$. The evolution is in accordance with the results presented in [6]. From Fig. 1, it becomes obvious that the number of nodes with degree zero increases with the decoding time, but it reaches a stationary point that determines the performance limit of the codes. We can also observe that the performance improves with the symbol rate. Furthermore, our model based on Wormald method closely matches the simulation results in [6], which are derived specifically for moderate codeblocks such as $k = 1000$. Our method is however more generic and applies to different settings. For example, the proposed analysis method is appropriate for large codeblocks and medium size codeblocks, i.e., $k < 1000$ while other analysis works [9] can be used exclusively for large codeblocks $k > 10000$. Interestingly, the presented analysis method is valid for any η value. In contrary to our method, the work in [9] can be applied only in the intermediate range.

D. Recovery probability

The above analysis can be used to characterize the decoding procedure and derive the symbol decoding probability for the Decoder-D. We thus define the symbol recovery probability as $\mathbb{P}_d = v_0$ (v_0 stands for the probability of recovering a symbol, its degree is zero). The probability $\mathbb{P}_d$ is depicted in Fig. 2 for various code rates η. The dotted line represents the symbol recovery performance for Growth codes derived by the Wormald method, the dashed-doted line represents the results of the experimental evaluation for $k = 1000$ and the dashed line describes the recovery performance of an "ideal" code that is able to recover a new source symbol with every received Growth encoded symbol. The "ideal" code is presented for the sake of the completeness, since such code does not exist in practice. From the Fig. 2, we can infer that, for large receiving symbol rates ($\eta > 1.5$), Growth codes can decode the full message as $\mathbb{P}_d$ reaches one.

To the best of our knowledge, no closed formula exists for the symbol decoding probability in the Decoder-D algorithm, which is valid for any arbitrary check nodes degree distribution; it is quite difficult to derive such an analytical model from the differential equations (15)-(17). By observing Fig. 2, we however note that an exponential model is a good fit. Thus, the symbol decoding probability can be approximated as

$$\mathbb{P}_d(\eta) \approx 1 - \lambda e^{-\mu\eta^2} \tag{18}$$

where $\eta = r/k$ is the coderate and r denotes the number of received symbols. Similarly, we define the probability of symbol decoding failure as

$$\mathbb{P}_l(\eta) = \lambda e^{-\mu\eta^2}. \tag{19}$$

The values of μ and λ depend on the underlying degree distribution function and not on the code length. This is expected as our model based on Wormald method also depends only on the degree distribution of the employed codes. We found through experimentation that for Growth codes $\lambda = 0.926854$ and $\mu = 1.39361$. The evaluation of the model of Eq. (18) is depicted with the continuous line in Fig. 2 for $k = 1000$. Since our model is a function of the η value and not of the code length k, we can use it to characterize the performance of Growth codes of various codeblocks. We see from Fig. 2 that our model agrees with the experimental evaluation.

IV. ERROR-RESILIENT VIDEO STREAMING

The analysis of the Growth Codes performance is important for the optimization of error resilient video communications where the good intermediate performance of the Growth codes can be very beneficial. Video is indeed a signal that may tolerate some losses and still offer a decent rendering quality. In particular, the above permits to select appropriate source and channel rates for optimized communications on rate-constrained lossy channels. The efficiency of video transmission systems is measured with respect to the video quality at the clients. This quality typically depends on the encoded video at sources and on the effect of losses during transmission that are respectively driven by the source and channel rates. In
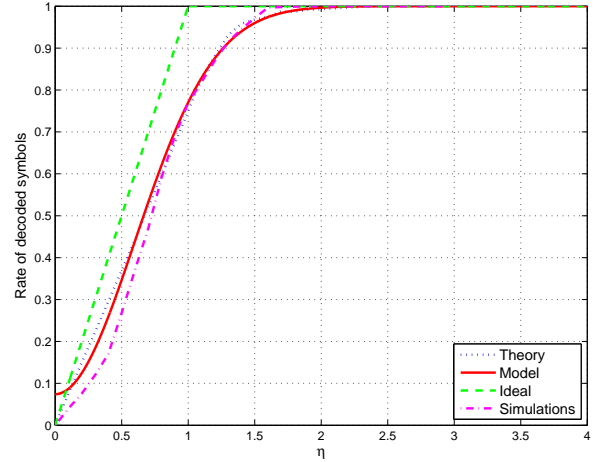

Fig. 2. Symbol recovery probability of Growth codes. The continuous curve is $\mathbb{P}_d(\eta) = 1 - \lambda e^{-\mu\eta^2}$ (η denotes the reception rate), the dotted line corresponds to the growth code performance derived by Wormald method, the dashed-doted line represents the results of the experimental evaluation, and the dashed line the ideal intermediate performance (it corresponds to the performance of an "ideal" code that is able to recover with every received Growth symbol a new source symbol). The codeblock size k is equal to 1000.

the following, we quantify the end-to-end performance of an illustrative video application using Growth codes in order to highlight the benefits of good intermediate performance. Note that a similar analysis can be done for LT codes, which also fit with the representation used in the Wormald analysis above.

A. Illustrative video model

Video compression is first applied to an image sequence in order to reduce the transmission rate. However, compression renders video data sensitive to packet losses. Therefore, the compressed video data should be protected to avoid rapid quality degradation in case of packet losses. Typically, this is achieved by forward error correction coding (FEC) schemes.

The overall distortion for communication scenarios where channel codes are used for FEC depends on the efficiency of the employed source compression scheme and the channel codes. Whereas the source coding distortion D_s decreases with increasing source rate R, the channel coding distortion D_c increases with lesser redundancy. Several distortion models are proposed in the literature [10], [18]. Considering hierarchically structured compression schemes such as MPEG, an analytical expression has been found in [10] for the end-to-end distortion D, where the distortion depends on the MPEG coding parameters and the average recovery probability of the deployed FEC codes.

The source distortion can be written as

$$D_s(R) = \alpha R^{-\beta} \tag{20}$$

where R is the source bit rate, $\alpha > 0$ and $\beta > 0$ are parameters that depend on the source encoder and on the video content.

The distortion, in case of loss, takes into account error propagation, error patterns and error concealment. Therefore, it reads

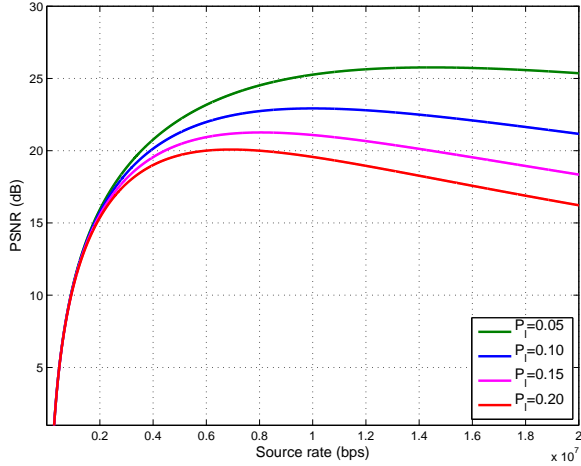$$D_L(R) = b\left(1 + \frac{R}{2aN_sL_p}\right)\mathbb{P}_l(\eta) \tag{21}$$

Fig. 3. Illustration of the PSNR (in dB) for the Foreman CIF sequence encoded at different rates (R_s) for various decoding failure probabilities $\mathbb{P}_l(n)$ using the model of Eq. (27).

where N_s is the average number of video slices[2] per second and L_p is the packet size (in bytes) of the transmitted video data. The parameter a indicates the type of video loss/distortion pattern that depends on the packetization scheme; b is a constant related to the spatio-temporal complexity of the sequence and error concealment scheme. Finally, $\mathbb{P}_l(\eta)$ is the average residual packet loss probability after FEC decoding and $\eta = r/k$ is a redundancy factor with r being the number of channel symbols and k the number of source symbols. We propose to use this end-to-end distortion model in our illustrative example of optimized Growth Codes video transmission scheme, where we replace the packet loss probability $\mathbb{P}_l(\eta)$ by the one obtained in the performance analysis of Section III.

B. Optimal joint source and channel coding

Since the video quality depends on the source and channel distortions, it depends on the source bit rate R. Thus, the optimal value of R that minimizes the end-to-end distortion function $D(R)$ can be determined by

$$R^\star = \arg\min_R \ D(R) \qquad (22)$$
$$s.t. \ (1+\epsilon)R \le B$$

where

$$
\begin{aligned}
D(R) &= (1 - \mathbb{P}_l(\eta))\, D_s(R) + \mathbb{P}_l(\eta) D_c(R) \quad (23)\\
&= D_s(R) + \mathbb{P}_l(\eta)\,(D_c(R) - D_s(R))\\
&= D_s(R) + D_L(R)
\end{aligned}
$$

and $D_c(R)$ denotes the concealment distortion (*i.e.*, the average distortion between original macroblocks and the concealed marcoblocks at the receiver in case of loss) [10] and $D_L(R) =$

---

[2]Each slice consists of several macroblocks, which are the basic encoding unit of H.264/AVC and other block-based video compression schemes.

$\mathbb{P}_l(\eta)\,(D_c(R) - D_s(R))$ the average distortion between a losslessly and lossily transmitted versions, as defined in [10]. Trivially, for lossless transmission, $D_c(R)$ is equal to zero. Here, the parameter $\mathbb{P}_l(\eta)$ is a constant where the affordable loss rate is determined by the video application constraints. The parameter B in Eq. (23) stands for the channel capacity, whose constraint takes into account *i.e.*, the amount of added channel redundancy (i.e., ϵ) for achieving a given symbol decoding failure rate $\mathbb{P}_l$. The value of ϵ can be computed from Eq. (19) by setting $\eta = 1+\epsilon$. We further assume that there is no estimate for π, the actual channel packet loss rate. Obviously, when an estimate for π is available, other Rateless codes such as LT codes would be a better solution, as they characterized by smaller overhead value ϵ than Growth codes. Growth codes are however interesting in a setup where the actual channel loss rate is unknown, and where the application can benefit from improved intermediate performance.

Now, taking into account Eqs. (20) and (21), we can write Eq. (23) as

$$D(R) = \alpha R^{-\beta} + b\left(1 + \frac{R}{2aN_sL_p}\right)\mathbb{P}_l(\eta) \qquad (24)$$

Then, the source and channel rate optimization problem of Eq. (23) is rewritten as

$$R^\star = \arg\min_R \ \left[\alpha R^{-\beta} + b\left(1 + \frac{R}{2aN_sL_p}\right)\mathbb{P}_l(\eta)\right]. \quad (25)$$

Before determining the optimal source coding rate $R^\star$, we first prove that $D(R)$ is convex.

Lemma 1: The end-to-end distortion function $D(R)$ is convex.

Proof: In order to prove the convexity of $D(R)$ is convex, we compute the first derivative of $D(R)$ with respect to the rate R.

$$
\begin{aligned}
\frac{\partial}{\partial R}D(R) &= \frac{\partial}{\partial R}\left[\alpha R^{-\beta} + b\left(1 + \frac{R}{2aN_sL_p}\right)\mathbb{P}_l(\eta)\right]\\
&= \frac{b\mathbb{P}_l(\eta)}{2aN_sL_p} - \alpha\beta R^{-(1+\beta)}
\end{aligned}
$$

Therefore, the stationary point $R^\star$ that satisfies $\frac{\partial}{\partial R}D(R) = 0$ is,

$$R^\star = \left(\frac{2a\alpha\beta N_sL_p}{b\mathbb{P}_l(\eta)}\right)^{\frac{1}{1+\beta}} \qquad (26)$$

We now have to prove that the second derivative $\frac{\partial^2}{\partial R^2}D(R)$ at $R^\star = \left(\frac{2a\alpha\beta N_sL_p}{b\mathbb{P}_l(\eta)}\right)^{\frac{1}{1+\beta}}$ is non-negative.

Thus,

$$
\begin{aligned}
\frac{\partial^2}{\partial R^2}D(R) &= \alpha\beta\,(1+\beta)\,(R^\star)^{-(2+\beta)}\\
&= \alpha\beta\,(1+\beta)\left(\frac{b\mathbb{P}_l(\eta)}{2a\alpha\beta N_sL_p}\right)^{\frac{2+\beta}{1+\beta}},
\end{aligned}
$$

which is clearly positive since $\alpha, \beta > 0$. Hence, $D(R)$ is convex and a unique minimum exists for a fixed $\mathbb{P}_l(\eta)$. ∎

Therefore, we can conclude the following. When $(1 + \epsilon)R^\star \ge B$, the optimal rate $R^\star$ is computed by Eq. (26);
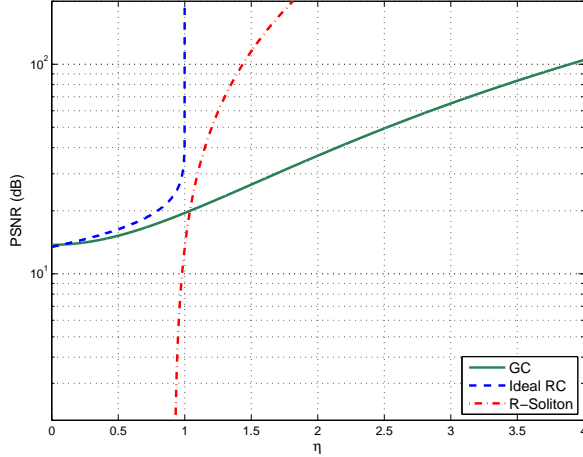
Fig. 4. Comparison of the proposed optimized Growth codes (GC) with ideal Rateless codes (ideal RC) and LT codes "R-Soliton" for transmission of Foreman CIF sequence with respect to various η values. The distortion is given in terms of PSNR.
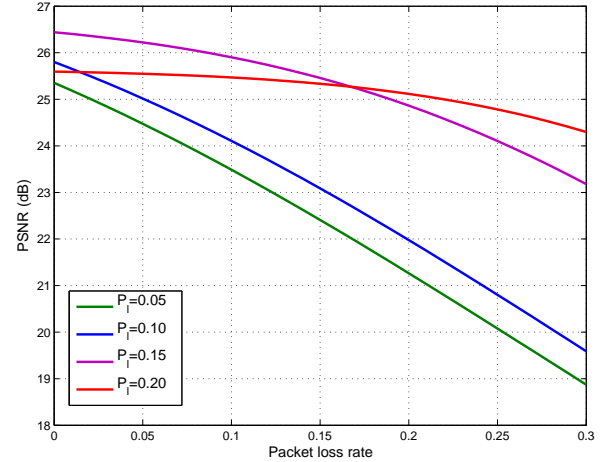

Fig. 5. Evaluation of the proposed scheme for transmission of the Foreman CIF sequence over different values of loss rate and different acceptable packet error rates $\mathbb{P}_l$ when channel capacity B is 15 $Mbps$.

otherwise, $R^\star = \frac{B}{1+\epsilon}$. This is due to the fact that the plane $(1+\epsilon)R - B$ intersects $D(R)$ at $R = \frac{B}{1+\epsilon}$.

Hence, we have that the minimum distortion $D^\star$ is given by,

$$D^\star = \alpha(R^\star)^\beta + b\left(1 + \frac{R^\star}{2aN_sL_p}\right)\lambda e^{-\mu\eta^2} \qquad (27)$$

where the optimal rate is equal to

$$R^\star = \left(\frac{2a\alpha\beta N_sL_p}{b\lambda e^{-\mu\eta^2}}\right)^{\frac{1}{1+\beta}}. \qquad (28)$$

The distortion for video transmission under different packet recovery probabilities, when Growth codes are used for error protection and the model parameters of [10] are adopted is shown in Fig. 3. We investigate the distortion in both intermediate region $0 \leq \eta < 1$ and full recovery region $\eta \geq 1$. From Fig. 3, we can see that $D(R)$ is strictly convex.

C. Evaluation of rateless video distortion

We are interested in the average distortion performance when a certain number of coded packets reach the receiver. No synchronization between the servers and receivers is assumed, which means that some packets may be received multiple times.

Distortion comparisons in terms of PSNR are illustrated in Fig. 4 when the $D(R)$ convex hull does not intersects with the plane $(1+\epsilon)R - B$, i.e., there is enough bandwidth to accommodate the transmission of the video encoded at rate computed by Eq. (26). The source coding parameters are chosen based on the model in [10]. The proposed rate optimal Growth codes are denoted as GC in Fig. 4. The optimal source rate value is calculated by Eq. (26). The proposed scheme is compared with an ideal Rateless code, which can recover as many source symbols as the received coded symbols, i.e., it recovers ηN source symbols when ηN coded packets are received in the intermediate region (i.e.,

$0 \leq \eta < 1$) and fully recovers the source when $\eta \geq 1$. Clearly, the performance of the ideal code serves as an upper limit of the performance of any channel code. From Fig. 4, we note that when the proposed rate optimal Growth codes are used, the overall distortion follows a smooth waterfall curve. In the intermediate region, the rate optimal Growth codes offer comparable distortion performance to ideal Rateless codes. We also compare with LT codes (denoted as "R-Solition"). The optimal source rate is computed by Eq. (26) using our model (Eq. (19)) with parameter values $\lambda = 5 \cdot 10^8$ and $\mu = -20$ for calculating the decoding failure probability. The large distortion gap between Growth codes and LT codes (denoted as "R-Soliton") in intermediate region is also obvious from Fig. 4. This is expected as the Robust soliton distribution has a small percentage of degree one symbols and it is designed for full recovery. In the full recovery area, the performance of LT codes improves rapidly with η value and approaches the performance of ideal Rateless codes. Hence, we can conclude the full advantage of Growth codes is expressed with a source coder such as a video coder whose utility increases with the decoded data.

For the sake of completeness, we provide comparisons for channel mismatch conditions, i.e., when the rate of GC is optimized for a set of acceptable decoding failure rates $(5, 10, 15, 20)\%$ (these values are considered when the optimal source rates are calculated), but in practice the channel conditions are different than those considered during optimization. Specifically, the source rate is optimized assuming that the channel is free of packet losses, however in practice the actual loss rate varies in the range of $[0, 30]\%$. The results are presented in Fig. 5 for transmission of the Foreman CIF sequence and channel capacity $B = 15$ $Mbps$. From the results, we can observe that the acceptable decoding failure rate plays important role on the robustness of the method. When the acceptable loss rate is 15% or 20%, the optimal source rate is significantly lower than the capacity B. Hence,
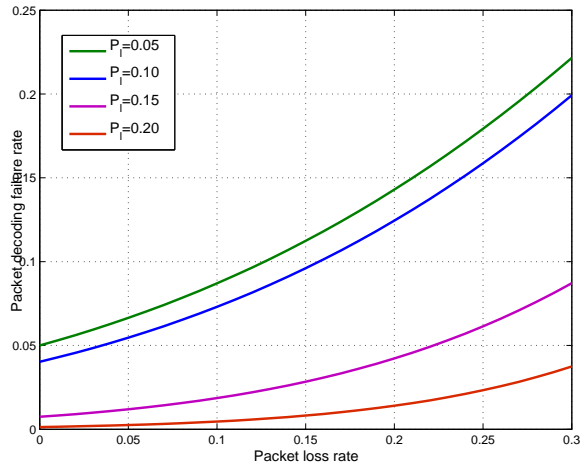
Fig. 6. Decoding packet failure probability for different acceptable packet error rates $\mathbb{P}_l$ when channel capacity B is 15 $Mbps$. Various loss rates are examined.

our scheme is characterized by higher η values and our method is more resilient to variations of the loss rate, since the packet decoding failure rate is not affected significantly. This is verified by Fig. 6 where we can see that the packet failure probability changes smoothly. However, when the acceptable loss rate is low, the optimal source rate is rather high and the scheme is optimized for lower η values. In such case the symbol decoding rate is affected significantly by the increased loss rate. This is due to the fact the actual η value is smaller than the theoretical optimal because of the channel losses. Therefore, the video transmission becomes more fragile to error changes. It is worth noting that when the acceptable failure rate is 5%, the packet decoding failure rate changes from 5% to 22%.

V. CONCLUSIONS

In this paper, we have analyzed the performance of Growth codes using the Wormald method and derived generic bounds that fully characterize the decoding process. We find that an exponential model can nicely describe the decoding performance. In contrary to other schemes in the literature, the proposed analysis framework is appropriate for medium sized codeblocks and not only to large sized codeblocks. It is general and appropriate for analyzing any other Rateless code that does not include a precoding step. We proposed an illustrative video streaming application to demonstrate the benefits of the good intermediate performance of Growth codes. We have casted a joint source and channel rate allocation problem whose convex objective function permits to select of the good intermediate performance of Growth codes. Our illustrative experiments show that Growth codes offer good intermediate performance that are useful for video applications whose quality monotonically grows with the number of packets. Finally, Growth codes are pretty robust to inaccurate estimations of the channel status, which is very interesting in the design of practical streaming applications.

REFERENCES

[1] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," in *Proc. of ACMSIG-COMM'98*, Vancouver, BC, Canada, Jan. 1998, pp. 56–67.
[2] A. Shokrollahi, "Raptor codes," *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.
[3] M. Luby, "LT codes," in *Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '02)*, Vancouver, Canada, Nov. 2002, pp. 271–280.
[4] "3GPP TS 26.346 V7.1.0, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; Protocols and Codecs," June 2005.
[5] "ETSI EN 302 304 Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H) European Telecommunication Standard, Nov. 2004," [Online], Available on web site http://www.dvb-h-online.org/technology.
[6] A. Kamra, J. Feldman, V. Misra, and D. Rubenstein, "Growth Codes: Maximizing Sensor Network Data Persistence," in *Proc of ACM Sigcomm'06*, Pisa, Italy, Sep. 2006.
[7] A. G. Dimakis, J. Wang, and K. Ramchandran, "Unequal Growth Codes: Intermediate Performance and Unequal Error Protection for Video Streaming," in *Proc. of Multimedia Signal Processing, MMSP'07*, Chania, Greece, Oct. 2007, pp. 107–110.
[8] N. C. Wormald, "Differential Equations for Random Processes and Random Graphs," *Annals of Applied Probability*, no. 5, pp. 1217–1235, Jan. 1995.
[9] S. Sanghavi, "Intermediate Performance of Rateless Codes," in *in Proc. of IEEE Information Theory and Applications, ITA'07,*, Lake Tahoe, CA, USA, Sep. 2007, pp. 478–482.
[10] P. Frossard and O. Verscheure, "Joint Source/FEC Rate Selection for Quality-Optimal MPEG-2 Video Delivery," *IEEE Trans. Image Processing*, vol. 10, no. 12, pp. 1815–1825, Dec. 2001.
[11] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Rateless Codes With Unequal Error Protection Property," *IEEE Trans. Information Theory*, vol. 53, no. 4, pp. 1521–1532, Apr. 2007.
[12] 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.
[13] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.
[14] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.
[15] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2007.
[16] H. Jeffreys and B. S. Jeffreys, "The Lipschitz Condition," 1.15 in Methods of Mathematical Physics, 3rd ed. Cambridge, England: Cambridge University Press, 1988.
[17] B. Bollobas, *Random Graphs*. Cambridge University Press, 2001.
[18] S. Sakazawa, Y. Takishima, M. Wada, and Y. Hatori, "Coding control scheme for a multi-encoder system," in *Proc. of the 7th Int. Workshop Packet Video*, Brisbane, Australia, Mar. 1996.