Reduction of Padding Overhead for RLNC Media Distribution With Variable Size Packets

Maroua Taghouti, Daniel E. Lucani, *Senior Member, IEEE*, Juan A. Cabrera, Martin Reisslein, *Fellow, IEEE*, Morten Videbæk Pedersen, and Frank H. P. Fitzek

***Abstract*—Random linear network coding (RLNC) can enhance the reliability of multimedia transmissions over lossy communication channels. However, RLNC has been designed for equal size packets, while many efficient multimedia compression schemes, such as variable bitrate (VBR) video compression, produce unequal packet sizes. Padding the unequal packet sizes with zeros to the maximum packet size creates an overhead on the order of 20%–50% or more for typical VBR videos. Previous padding overhead reduction approaches have focused on packing the unequal packet sizes into fixed size packets, e.g., through packet bundling or chaining and fragmentation. We introduce an alternative padding reduction approach based on coding macro-symbols (MSs), whereby an MS is a fixed-sized part of a packet. In particular, we introduce a new class of RLNC, namely MS RLNC which conducts RLNC across columns of MSs, instead of the conventional RLNC across columns of complete packets of equal size. Judiciously arranging the source packets into columns of MSs, e.g., through shifting the source packets horizontally relative to each other, supports favorable MS RLNC coding properties. We specify the MS RLNC encoding and decoding mechanisms and analyze their complexity for a range of specific MS arrangement strategies within the class of MS RLNC. We conduct a comprehensive padding overhead evaluation encompassing both previous approaches of packing the unequal size packets into fixed size packets as well as the novel MS RLNC approaches with long VBR video frame size traces. We find that for small RLNC generation sizes that support low network transport delays, MS RLNC achieves the lowest padding overheads; while for large generation sizes, both the previous packing approaches and the novel MS RLNC approaches effectively reduce the padding overhead.**

***Index Terms*—Random linear network coding (RLNC), variable-sized packets, variable bitrate (VBR) video, video streaming, zero-padding.**

I. Introduction

A. Motivation: Padding Overhead in RLNC

MODERN media distribution scenarios involve a wide range of heterogeneous underlying communication channels, including lossy channels, e.g., wireless channels [4]–[8]. The effects of lossy channels on media distribution can be mitigated through error correction coding [9]–[14]. Recently, random linear network coding (RLNC) has emerged as a popular error correction coding scheme for a wide range of complex lossy networks. RLNC has been employed in a wide range of systems involved in media distribution, including data storage systems [15]–[20], multicast and broadcast distribution networks [21]–[25], peer-to-peer distribution networks [26]–[29], as well as general content distribution and streaming systems [30]–[34].

RLNC has been designed for equal size packets, see Section II-B. However, popular efficient multimedia compression schemes, in particular video compression schemes, are typically most efficient when operating in a variable bitrate (VBR) mode [35], [36]. Therefore, VBR video has received significant interest for media distribution [37]–[41]. Padding the unequal VBR video packets with zeros [42], [43], i.e., the so-called zero-padding to obtain equal size packets for RLNC encoding, creates a high overhead, as quantified in Section II.

B. Existing Padding Reduction Approaches

The padding overhead in RLNC for distribution of VBR encoded media has received very limited attention to date. Compta *et al.* [44] have proposed padding reduction approaches that seek to pack the unequal size packets into fixed size packets that are amenable to conventional RLNC. In particular, the simple bundling approach [44] is similar to the classical bin packing problem, i.e., strives to pack variable size objects (packets) into the least number of bins (packets) of the same capacity (fixed packet size). Bin packing is an NP-hard problem, i.e., has very high computational complexity. The chaining and fragmentation approach [44] strings the packets together back-to-back into one long string and then fragments

Manuscript received July 11, 2018; revised December 1, 2018; accepted December 22, 2018. Date of publication February 4, 2019; date of current version September 4, 2019. This work was supported in part by the German Research Foundation (DFG) through the CoSIP Project under Grant FI 1671/1-1, in part by the Collaborative Research Center 912 Highly Adaptive Energy-Efficient Computing, as well as by the Danish Council for Independent Research through the TuneSCode Project under Grant DFF - 1335-00125, and in part by the Aarhus Universitets Forskningsfond under Project AUFF-2017-FLS-7-1. Preliminary versions of Sections II and III appeared in [1]–[3]. *(Corresponding author: Martin Reisslein.)*

M. Taghouti is with the Tunisia Polytechnic School, University of Carthage, Carthage 1054, Tunisia, and also with the Deutsche Telekom Chair of Communication Networks, Technische Universität Dresden, 01062 Dresden, Germany (e-mail: maroua.taghouti@tu-dresden.de).

D. E. Lucani is with DIGIT Centre, Department of Engineering, Aarhus University, 8200 Aarhus, Denmark (e-mail: daniel.lucani@eng.au.dk).

J. A. Cabrera and F. H. P. Fitzek are with 5G Lab Germany, Deutsche Telekom Chair of Communication Networks, Technische Universität Dresden, 01062 Dresden, Germany (e-mail: juan.cabrera@tu-dresden.de; frank.fitzek@tu-dresden.de).

M. Reisslein is with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: reisslein@asu.edu).

M. V. Pedersen is with Steinwurf ApS, 9220 Aalborg, Denmark (e-mail: morten@steinwurf.com).

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

Digital Object Identifier 10.1109/TBC.2019.2892594

0018-9316 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

the long string into fixed size packets. In contrast, to these existing approaches, which are limited to packing the VBR video packets into fixed size packets, we introduce a novel fine-granular RLNC coding approach that can flexibly encode fixed size fragments of packets. The performance evaluations in [44] have been limited to 320 frames of a mostly motion-free video. In contrast, we conduct extensive evaluations with frame size traces of long (several thousand frames) full motion VBR video. We comprehensively evaluate the packet packing approaches as well as the MS RLNC approaches with the long VBR video traces.

The recent study [45] has addressed padding overhead for an XOR based coding scheme. XOR based coding is a special case of network coding on the binary Galois field GF(2), see Section II-B, that suffers from suboptimal throughput and significant linear dependencies of the coding coefficient vectors [46]–[49]. In contrast, this paper addresses the padding overhead for general RLNC for arbitrary Galois fields GF(2^q), $q \geq 1$, which is throughput-optimal and has negligible coding coefficient dependencies for sufficiently large q [22], [50].

C. Contributions: MS RLNC and Padding Evaluation

This study makes two main contributions:

- We develop an alternative to the existing padding reduction approaches that pack the unequal size packets into fixed size packets for conventional packet granularity RLNC. In particular, we introduce a novel class of RLNC based on the encoding of columns of macro-symbols (MSs). The given variable size video packets are partitioned into fixed size MSs. With MS RLNC, the coding properties depend on the length of the longest column of source MSs. More specifically, the number of coded packets required at the receiver is greater than or equal to the longest MS column at the encoder. We introduce a deterministic horizontal shifting scheme that shifts the variable size source packets relative to each other to minimize the length of the longest column. This shifting minimizes the number of coded packets required for decoding at the receiver and thus the latency. We analytically characterize the MS RLNC decoding probability as well as the encoding and decoding complexities.
- We also conduct extensive quantitative evaluations of both the existing packing based approaches and the novel MS RLNC based approaches with long frame sizes of full motion VBR video. We consider different video encoders as well as a range of RLNC generation sizes in the evaluation. We find that for low-delay network transport with small RLNC generations, MS RLNC with deterministic shifting gives favorable performance. For large RLNC generations, both the bundling and the chaining and fragmentation approaches for packing the unequal size packets into fixed size packets and MS RLNC with deterministic shifting effectively reduce the padding overhead.

D. Organization

This paper is organized as follows: Section II gives an overview of the VBR video traffic characteristics and quantifies the padding overhead for forming equal size packets for conventional RLNC. We note that parts of Section II have been presented in the conference paper [1]. Section III introduces macro-symbol (MS) RLNC and quantifies its padding reduction for VBR video. The basic concept of MS RLNC has first been presented in the conference paper [2] and is now presented in a more comprehensive manner in Section III. Section IV introduces the shifting concept for MS RLNC. The shifting of source packets in the context of RLNC with general finite field coding coefficients is entirely novel and for the first time presented in this article. Section V analytically characterizes the MS RLNC decoding probability as well as the computational complexities of MS RLNC encoding and decoding. The computational complexities for the basic MS RLNC concept were first analyzed in the conference paper [3]; the complexity analysis is elaborated in this article and the MS RLNC decoding probability is for the first time examined in detail in this article. Section VI presents the comprehensive evaluation of padding overhead reduction approaches with long VBR video traces. Section VII summarizes the main conclusions of this study and outlines future work directions.

II. Padding in RLNC Based Video Distribution

This section first introduces the VBR video traces that are used for the quantitative evaluations throughout this study, and gives brief tutorial background on RLNC. Then, this section quantifies the padding overhead for the VBR video traces with conventional equal size packet RLNC.

A. Variable Video Packet Sizes

In general, there are two broad types of video, namely conversational video, e.g., from video conferencing applications, and full motion video, e.g., from movies, TV shows, and sporting events. Full motion video typically exhibits a wide variety of scenes with varying levels of motion activity and texture complexity. The varying scene content results in variable bitrate (VBR) video when the video is encoded with the modern efficient video coding standards [35], [36], [52], [53]. The varying sizes (in bytes) of the encoded video frames in VBR video in turn result in varying sizes (in bytes) of the packets carrying the encoded video data over multimedia distribution networks.

This study focuses on full motion video, which will likely account for large portions of the traffic on future wireless networks and the future Internet [54]. We selected five representative frame size traces of long full motion videos that were encoded with a variety of video coding approaches from a publicly available video trace library [51]. The selected videos cover a wide range of video resolution formats ranging from the low-resolution Common Interframe Format (CIF) to High Definition (HD) formats and the high-resolution 4k format, as detailed in Table I. The sizes of the encoded video frames (in bytes) can grow significantly larger than the common Ethernet maximum transfer unit (MTU) of 1500 bytes [55],

TABLE I
CHARACTERISTICS OF REPRESENTATIVE VIDEO TRACES [51] WITH 30 FRAMES PER SECOND (fps) FOR COMMON INTERFRAME FORMAT (CIF) RESOLUTION VIDEO AND 24 fps FOR THE REMAINING VIDEOS

Video Label: Enc. Appr. (Resol. Format)	Resolution [pixels × pixels]	Number of Video Frames	Number of Packets
SVC (CIF)	352 × 288	289040	336336
VP9 (HD)	1920 × 816	17592	45388
SVC (HD)	1920 × 1080	86268	336336
H.264 (HD)	1920 × 816	17592	91536
H.265 (4k)	4096 × 1744	17592	163920

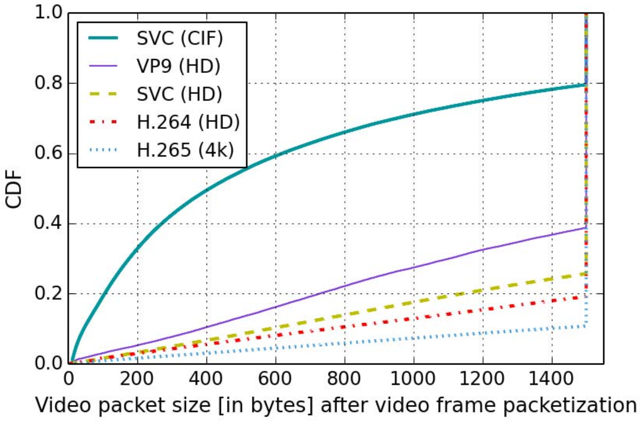


Fig. 1. Cumulative distribution function (CDF) of video packet sizes in bytes per packet.

[56]. Thus, a given large encoded video frame needs to be fragmented into multiple packets for video distribution over a network, resulting in packet numbers that are significantly larger than the number of encoded video frames, see Table I.

Fig. 1 illustrates the cumulative distribution functions (CDFs) of the video packet sizes corresponding to the video traces of Table I. We observe from Fig. 1 a common strong mode of the sizes at 1500 bytes, which corresponds to the prescribed MTU. We also observe that videos with higher resolutions tend to have a higher probability of being fragmented into packets of the MTU size. For instance, only about 20% of the SVC (CIF) video packets have 1500 bytes, while approximately 90% of the H.265 (4k) video packets have 1500 bytes.

B. Background: Generation Based RLNC

Generation based RLNC [57], [58] made network coding practical by performing the coding on successive sets (blocks) of packets, which are referred to as *generations*. The generation size N, i.e., the maximum number of source packets that can be combined to form the coded packets, should be set according to the type of application [59], [60]. We consider the unicast transmission of video data packets using generation based RLNC with full vector RLNC encoding [22], [58], [61]. Full vector RLNC, which is also referred to as non-systematic RLNC, transmits only encoded packets, whereby each encoded packet is obtained by linearly combining all the source packets in the considered generation according to uniformly distributed random coding coefficients over a prescribed Galois field GF(2^q) [22], [62]–[64].

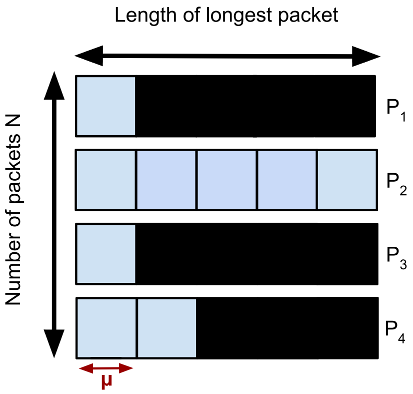


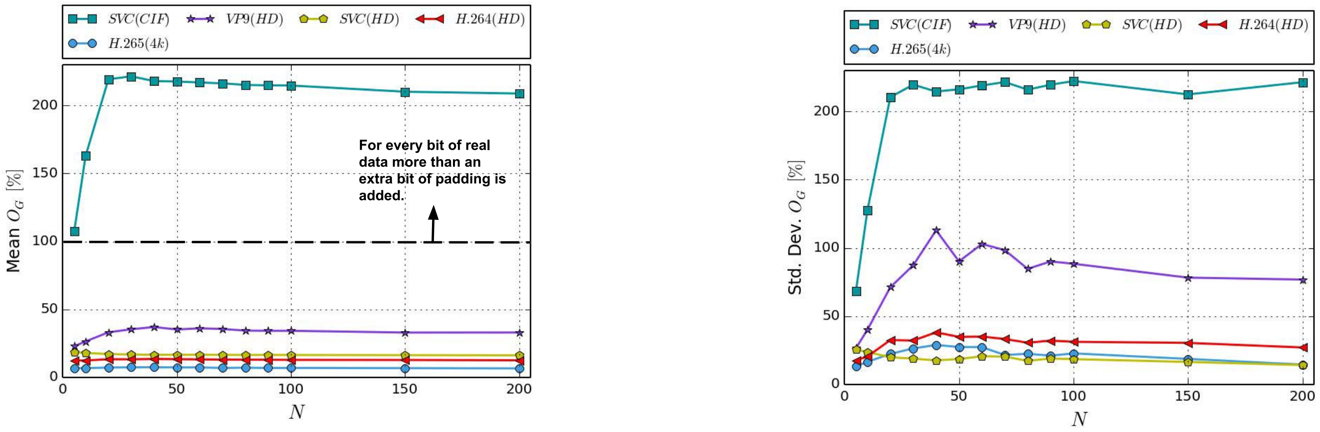
Fig. 2. An example of a generation of $N = 4$ source packets with initial sizes μ, 5μ, μ, and 2μ, whereby μ denotes a prescribed packet size unit, e.g., $\mu = 10$ bytes. The packets with sizes μ and 2μ are padded out to the maximum packet size of 5μ before RLNC coding. The resulting padding overhead (represented by the black rectangles) is $O_G = 11/9 = 122\%$, i.e., the amount of padding overhead is larger than the amount of source data.

More specifically, consider a portion of N successive packets out of a given source packet sequence. Denote the sizes of the source packets by $L_1, L_2, \ldots, L_N$ [in symbols of the underlying GF(2^q), i.e., with q bits per symbol]. If the source packet sequence consists of more than N packets, then multiple generations will be formed. We proceed to consider the coding of a given generation consisting of N successive source packets. In preparation for the encoding, the sender pads the packets that are smaller than the largest packet size $L_{\max} = \max_{1\le n\le N} L_n$ in the generation with trailing zero symbols; these trailing zero symbols are referred to as zero-padding. The sender then performs RLNC on N packets, each of size $L_{\max}$.

C. Padding Overhead for Generation Based RLNC

Random linear coding for forward erasure (or error) correction (FEC) generally creates three main types of overhead, namely the overhead for communicating (signalling) the coding coefficients to the receiver, the overhead due to transmitting extra coded packets, and the padding overhead. The coding coefficient overhead corresponds to the size of the encoding vector, i.e., the number of coding coefficients required for a given coded packet. Several studies have extensively covered the signalling overhead for the coding coefficients, see [65]–[67]. The FEC overhead due to transmitting extra coded packets depends on the coding ratio R, i.e., the number n_c of extra coded packets that are added to a generation of N source packets. Moreover, extra coded packets may need to be transmitted if coding coefficient vectors are linearly dependent.

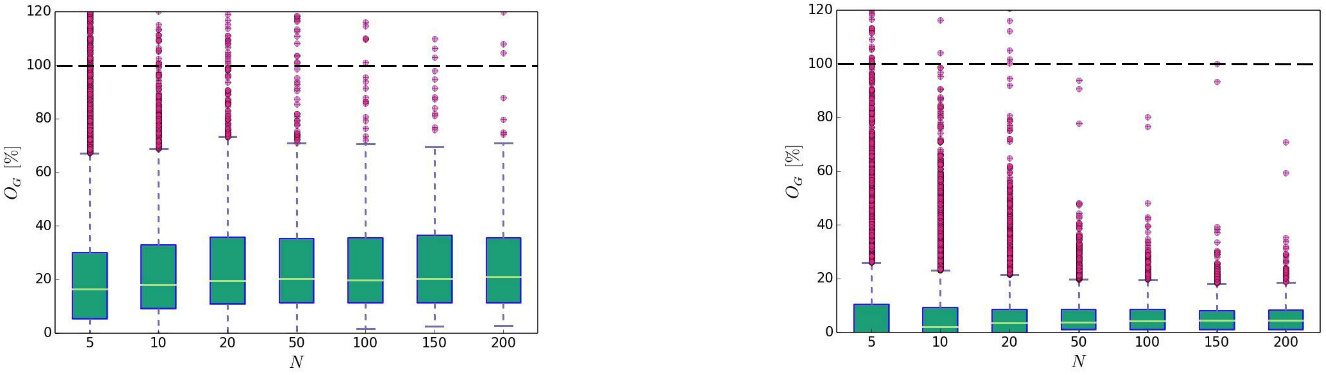
The padding overhead arises since coded packets, i.e., the linear combinations of the source packets, can only be created from source packets of the same size. This study focuses on the padding overhead, which depends on the sizes L_n [in units of symbols, i.e., bytes for the common GF(2^8)] of the packets with indices n, $n = 1, 2, \ldots, N$, in a given generation. In particular, the largest packet size $L_{\max} = \max_{1\le n\le N} L_n$ governs



(a) Mean of Padding Overhead O_G

(b) Standard Deviation of Padding Overhead O_G

Fig. 3. Means and standard deviations of padding overhead O_G [in percent] as a function of generation size N.



(a) VP9 (HD)

(b) H.265 (4k)

Fig. 4. Box plots of padding overhead O_G [in %] as a function of generation size N for H.265 (4k) and VP9 (HD) videos. The boxes give the median as well as lower and upper quartiles, while the whiskers give the 10% and 90% quantiles.

the padding overhead

$$O_G = \sum_{n=1}^{N}(L_{\max} - L_n) \Big/ \sum_{n=1}^{N} L_n, \qquad (1)$$

since each coded packet in the generation is padded out to the size $L_{\max}$ of the largest packet of the generation. This padding overhead for a generation O_G is illustrated in Fig. 2 for an example generation with four variable size source packets.

D. Numerical Results for RLNC Padding Overhead

Fig. 3 illustrates the means and the standard deviations of the padding overhead O_G as a function of the generation size N. We observe that the CIF format video incurs a very high padding overhead O_G that more than doubles the video data; for generations sizes N of 20 and larger, the padding overhead is over 200%, implying that the video data is tripled by the padding. This very high padding overhead for the CIF video is due to the highly variable frame sizes with many small size frames, as shown in Fig. 1. We observe from Fig. 3 that higher resolution videos tend to have lower padding overheads. For generation size $N = 50$, for instance, the VP9 (HD) video has a mean padding overhead of 35% (and a median padding overhead of 20%), while the H.265 (4k) video has a mean padding overhead of 7% (median of 3.8%).

For closer insights, Fig. 4 depicts box plots of the padding overhead O_G for the VP9 (HD) video and the H.265 (4k) video. We initially observe from Fig. 4 that the VP9 (HD) generations comprise more overhead with medians around 20% than the H.265 (4k) generations, which have median overheads around 3–4% (the corresponding means are around 30–35% and 6–7%, respectively). The higher overhead O_G of the VP9 (HD) video is due to its higher proportion of small packets, see Fig. 1.

We also observe from Fig. 4 that small generation sizes N, e.g., $N = 5$, have a much higher occurrence of outliers, i.e., generations with exceedingly high overheads, than large generation sizes, e.g., $N = 200$. This is mainly because small generation sizes N “sample” only a very short portion of the sequence of video frames. When considering only few successive encoded video frames there is a potential for vastly different frame sizes, e.g., when the few frames happen to include a large Intracoded (I) frame and a few small bidirectionally predicted (B) frames of the group of pictures structure of the encoding [51].

Fig. 5 shows box plots of the padding overhead O_G of the considered set of videos (see Table I) for the generation

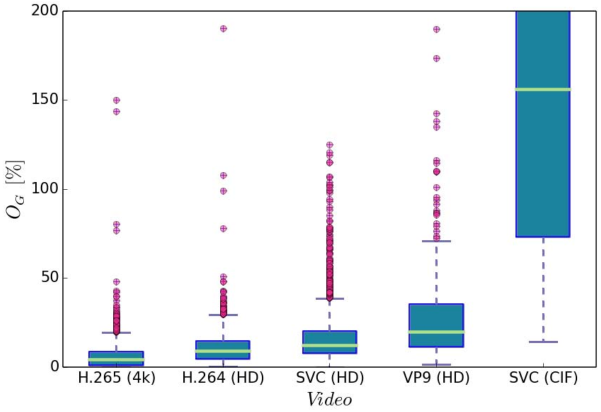


Fig. 5. Box plots of padding overhead O_G in percent for different videos for generation size $N = 100$.

size $N = 100$. We observe from Fig. 5 that higher resolution videos tend to have lower overheads. Nevertheless, we observe from Fig. 5 that even high resolution videos have many outlier generations with padding overheads between 20 and 50%. For high resolution videos, the video frames are quite large and require a large proportion of full (MTU) size packets. However, even for high resolution video with large encoded frame sizes, network distribution with generation based RLNC comes at the expense of a significant zero-padding overhead.

III. Mitigating Padding With MS RLNC

A. MS RLNC Encoding

1) Overview: From Packet to Macro-Symbol (MS) RLNC Coding Granularity: As examined in the preceding section, the padding overhead with generation based RLNC arises due to the packet-level granularity of the conventional RLNC. More specifically, the conventional RLNC creates coded packets by linearly combining columns of full packets of the same size $L_{\max}$ in a given generation. We propose to reduce the padding overhead through RLNC encoding at the finer granularity of columns of macro-symbols (MSs). We define an MS to have a prescribed size μ in terms of symbols of the source packets, whereby a symbol has q bits, i.e., is a symbol of the underlying Galois field GF(2^q). The MS size μ can range from a single symbol, i.e., q bits, to the maximum packet size in the generation, i.e., $L_{\max}$. For $\mu = L_{\max}$, MS RLNC degenerates to the conventional packet granularity RLNC.

2) Setting (Generation Based RLNC): We consider the unicast transmission of one generation consisting of N source packets $P_1, P_2, \dots, P_n, \dots, P_N$. Let $L_1, L_2, \dots, L_n, \dots, L_N$ denote the sizes of the packets [in units of symbols] and denote $L_{\max} = \max_{1 \le n \le N} L_n$ for the maximum packet size in the generation. The main notations are summarized in Table II.

3) Partitioning Source Packets Into Macro-Symbols (MSs): We define the packet lengths in units of MSs as

$$\Lambda_n = \lceil L_n/\mu \rceil, \tag{2}$$

where $\lceil a \rceil$ is the smallest integer larger than or equal to a. We let $\Lambda_{\max}$ denote the size of the largest source packet in units of MSs. Let $P_n = \{s_{n1}, s_{n2}, \dots, s_{n\lambda}, \dots, s_{n\Lambda_n}\}$ denote the sequence of MSs that form source packet P_n. Furthermore, let $\Pi_\lambda,\ \lambda = 1, 2, \dots, \Lambda_{\max}$, denote the distribution of the source packet lengths, i.e., Π_λ denotes the number of source packets of length λ MSs. Note that by the basic properties of a distribution, $\sum_{\lambda=1}^{\Lambda_{\max}} \Pi_\lambda = N$. The source packet partitioning into MSs of size μ symbols is illustrated for an example generation with $N = 4$ source packets of sizes $\Lambda_1 = 1$, $\Lambda_2 = 5$, $\Lambda_3 = 1$, and $\Lambda_4 = 2$ in Fig. 2.

TABLE II
Summary of Main Notations

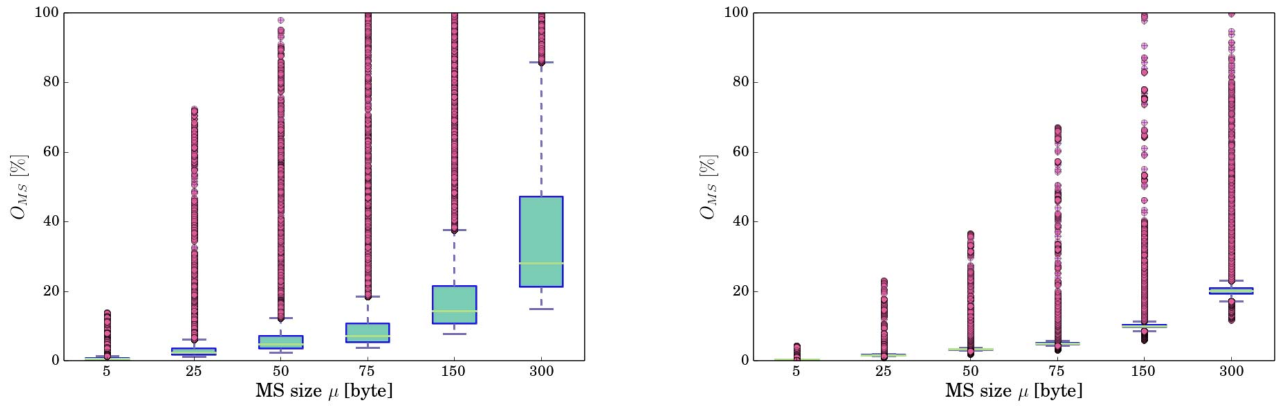
Generation of Source Packets	
q	Specification of finite field size GF(2^q); with q bits per symbol in GF(2^q)
N	Generation size [in number of source packets] with packet index $n = 1, 2, \dots, N$
L_n	Size of source packet n [in # of symbols in GF(2^q)]
Source Packet Partitioning into Macro-symbols (MSs)	
μ	Macro-symbol (MS) size [in # of symbols in GF(2^q)]
Λ_n	Size of source packet n [in MSs], $\Lambda_n = \lceil L_n/\mu \rceil$
$\Lambda_{\max}$	Size of largest source packet in a generation [in MSs]
λ	MS position, i.e., column position, in a given packet, $\lambda = 1, 2, \dots, \Lambda_{\max}$
Π_λ	Source packet size distribution, Π_λ = # of source packets with size λ MSs; $\sum_{1 \le \lambda \le \Lambda_{\max}} \Pi_\lambda = N$
Δ_λ	Source MS degree of column λ, i.e., # of MSs in column λ position = $N - \sum_{\ell=1}^{\lambda-1} \Pi_\ell$
Macro-symbol (MS) RLNC Coding Parameters	
$s_{n\lambda}$	Source MS in column pos. λ, $\lambda = 1, 2, \dots, \Lambda_n$ in source pkt. n
$\alpha_{\eta n}$	Coding coeff. for source pkt. n when generating coded pkt. η
$c_{\eta\lambda}$	Coded MS in coded packet η in MS position λ
MS RLNC Coded Packets	
K	Number of coded packets for a generation
Φ_η	Size of coded pkt. η, $\eta = 1, 2, \dots, K$, in MSs

We define the source MS degree $\Delta_\lambda,\ \lambda = 1, 2, \dots, \Lambda_{\max}$, of the given generation of source packets $P_n,\ n = 1, 2, \dots, N$, as the number of MSs in the column position λ, $\lambda = 1, 2, \dots, \Lambda_{\max}$. In the example generation in Fig. 2, there are four MSs in the left-most ($\lambda = 1$) column, hence, $\Delta_1 = 4$. There are two MSs in the second column from the left ($\lambda = 2$ column), thus, $\Delta_2 = 2$. Furthermore, $\Delta_3 = \Delta_4 = \Delta_5 = 1$. In general, based on the source packet length distribution:

$$\begin{aligned} \Delta_1 &= N \\ \Delta_\lambda &= N - \sum_{\ell=1}^{\lambda-1} \Pi_\ell \ , \ \ \lambda \ \in \{2, \dots, \Lambda_{\max}\}. \end{aligned} \tag{3}$$

MS RLNC does not require that all packets in a generation are padded out to the largest packet in the generation. Instead, MS RLNC only requires that all packets are padded out to a size that is an integer multiple of the MS size μ. For a generation size of N packets, this MS RLNC padding overhead O_{MS} is upper bounded by $N(\mu - 1)$ symbols. This upper bound of the overhead is attained when each packet has a size of some integer multiple of μ plus one symbol, requiring each packet to be padded with $\mu - 1$ zero symbols to a size of an integer multiple of μ.

4) MS Size μ Evaluation: Fig. 6 presents box plots of the MS RLNC padding overhead O_{MS} in percent (relative to the amount of data in the generation $\sum_{n=1}^{N} L_n$) for a generation of size $N = 64$ for different MS sizes μ. For the VP9 (HD) video, we observe from Fig. 6(b) that for small MS sizes μ, the overhead is mainly due to outliers. In particular, for small



(a) SVC (CIF)

(b) VP9 (HD)

Fig. 6. Percentage of zero-padding overhead for MS RLNC with different macro-symbol sizes μ for generations of $N = 64$ packets.

MS sizes $\mu = 5$ to $\mu = 25$ byes, the median overhead is close to zero and the few individual outliers reach only about 20%. The padding overhead with MS RLNC increases with increasing MS size μ. For example, we observe from Fig. 6(a) that for the SVC (CIF) video, the median padding overhead approaches 20% for an MS size of $\mu = 150$ bytes. However, these median MS RLNC padding overheads are still dramatically lower compared to the median padding overheads with conventional RLNC, which was on the order of 100% for SVC (CIF), see Fig. 5. For VP9 (HD), the median conventional RLNC padding overheads were around 20%, see Fig. 4. Note that setting the MS size μ to the MTU size 1500 byte would result in overheads that are as large as or larger than the overheads in Figs. 4 and 5; larger overheads with $\mu = 1500$ bytes arise for generations with $L_{\max} < 1500$ bytes which require only padding to $L_{\max}$ in conventional RLNC. Overall, we conclude from Fig. 6 that reducing the MS size μ significantly reduces the MS RLNC padding overhead. The overhead reductions are particularly dramatic for MS sizes μ below 100 bytes.

5) MS RLNC Encoding: The MS RLNC encoding follows the principles of the conventional generation based RLNC, with the exception that MS RLNC encodes at the granularity of individual MSs. Recall that $s_{n\lambda},\ n = 1, 2, \ldots, N, \lambda = 1, 2, \ldots, \Lambda_n$ denotes the MS in source packet (row position) n and MS position (column position) λ. Let $\alpha_{\eta n}$ denote the coding coefficient for source packet n when forming coded packet $\eta,\ \eta \geq 1$. Let $c_{\eta\lambda}$ denote the coded MS in coded packet (row) number η and MS position (column position) λ. Following the principles of generation based RLNC, the coded MS $c_{\eta\lambda}$ for a given coded packet η is obtained through the linear combination of all the N source MSs $s_{n\lambda},\ n = 1, 2, \ldots, N$, in the MS position (column position) λ with the "weights" given by the coding coefficients $\alpha_{\eta n}$. That is, each source MS in the column position λ is multiplied with its corresponding (for its row n) coding coefficient $\alpha_{\eta n}$, and these products are summed over the entire column, as illustrated in Fig. 7. Formally,

$$c_{\eta\lambda} = \sum_{n=1}^{N} \alpha_{\eta n} s_{n\lambda}, \quad \eta \geq 1,\ \ 1 \leq \lambda \leq \Lambda_n. \tag{4}$$

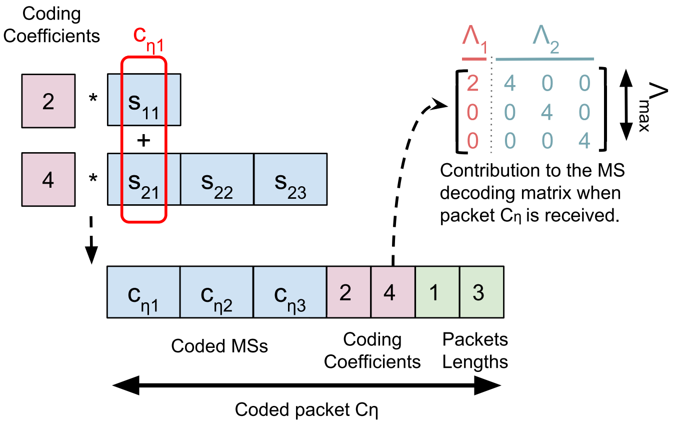


Fig. 7. Example illustration of macro-symbol (MS) RLNC coding for a generation consisting of $N = 2$ source packets with maximum packet size $\Lambda_{\max} = 3$ MSs. The coded MS $c_{\eta\lambda}$ is formed by linearly combining the source MSs $s_{n\lambda}$ in column position λ with the coding coefficients $\alpha_{\eta n}$. The coded packet η consists of the coded MSs $c_{\eta\lambda}$ from all the column positions $\lambda,\ \lambda = 1, 2, \ldots, \Lambda_{\max}$, as well as the coding coefficients $\alpha_{\eta n},\ n = 1, 2, \ldots, N$; also, one coded packet of the generation needs to carry the individual packet lengths Λ_n. The illustration on the right shows the eventual matrix expansion of the encoding vector at the receiver.

Note that the degree Δ_λ (see Eqn. (3)) gives the number of source MSs that are linearly combined to generate the coded MS $c_{\eta\lambda}$.

6) MS RLNC Packet Format: As illustrated in Fig. 7, coded (output) packet C_η contains the coded MSs $c_{\eta1}, c_{\eta2}, \ldots, c_{\eta\Lambda_{\max}}$ along with the coding coefficients $\alpha_{\eta1}, \alpha_{\eta2}, \ldots, \alpha_{\eta N}$. The coding coefficients require Nq bits. In addition, one of the coded packets in the generation (or a few, for reliability) needs to carry the individual packet lengths $\Lambda_1, \Lambda_2, \ldots, \Lambda_N$ of the packets in the considered generation to the receiver to ensure correct decoding. Each packet is at most $\Lambda_{\max}$ MSs long; thus, the overhead for sending the N individual packet lengths is upper bounded by $N \log_2 \Lambda_{\max}$ bits. Packet sequence numbers are not required for the purpose of MS RLNC decoding, only the packet lengths $\Lambda_1, \Lambda_2, \ldots, \Lambda_N$. The coded packets need to carry

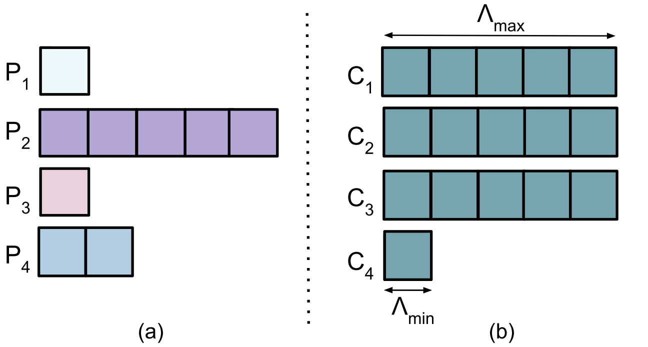


Fig. 8. Illustration of min-sized last coded packet policy for example generation with $N = 4$ packets: $N - 1 = 3$ coded packets have the full-length of the longest source packet $\Lambda_{\max} = 5$, while the last coded packet has the length $\Lambda_{\min} = 1$ of the minimum length source packet.

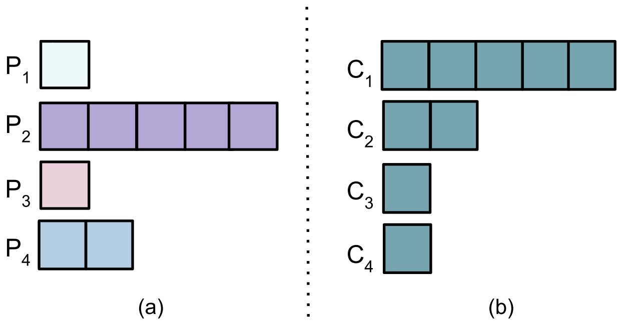


Fig. 9. Illustration of progressive shortening coded packet policy: The lengths of the coded packets in part (b) are matched to the lengths of the source packets in part (a).

a generation index if a data stream consists of multiple generations.

7) MS RLNC Coded Packet Size Distribution: The source should send as many coded packets C_η, $\eta = 1, 2, \ldots, K$, with $K \geq N$, as needed to ensure the total recovery of data at the receiver. For ease of explanation, we initially consider a code rate $R = N/K = 1$. The MS RLNC encoding introduced in Section III-A5 enables a wide range of policies for setting the sizes Φ_η of the encoded packets in units of MSs, whereby one MS corresponds to one encoded MS $c_{\eta\lambda}$.

a) Full-length coded packets: A full-length coded packet policy produces N coded packets $\eta = 1, 2, \ldots, N$, each of size $\Phi_\eta = \Lambda_{\max}$ coded MSs. This full-length coded packet encoding corresponds to the conventional packet granularity RLNC encoding and incurs the padding overhead examined in Section II. Effectively, when computing the coded MSs following Eqn. (4), the non-existent source symbols, e.g., the four rightmost symbols $s_{1,2}$, $s_{1,3}$, $s_{1,4}$, and $s_{1,5}$ in source packet P_1 in the example in Fig. 2 are considered to be zero-symbols. That is, all source packets are implicitly zero-padded to the full-length of the largest packet size $\Lambda_{\max}$ of the generation.

b) Min-sized last coded packet: The min-sized last coded packet policy produces $N - 1$ coded packet with the full-length of the maximum source packet size $\Lambda_{\max}$. The length of the last coded packet is set to the length $\Phi_N = \Lambda_{\min} = \min_{1 \leq n \leq N} \Lambda_n$ of the shortest source packet, as illustrated in Fig. 8. The min-sized last coded packet policy reduces the padding overhead by $\Lambda_{\max} - \Lambda_{\min}$ MSs compared to the full-length coded packet policy.

c) Progressive shortening: The progressive shortening policy sends coded packets with a length distribution that matches the length distribution Π_λ of the source packets, as illustrated in Fig. 9. Note that for progressive shortening, the number of source MSs that are combined to form the coded MS $c_{\eta\lambda}$ in column position λ is equal to the degree Δ_λ (see Eqn. (3)), i.e., the number of occurrences of source MSs in column position λ. That is, the first coded MS $c_{\eta 1}$ (in column position $\lambda = 1$) in a coded packet C_η contains N combined source MSs. The subsequent coded MSs to the right (i.e., in higher column positions λ, $\lambda \geq 2$) contain fewer combined source MSs as governed by the source packet length distribution Δ_λ, see Eqn. (3).

Absent of any loss or corruption of coded packets during the network transport or any finite field dependencies [59], [64], the receiver can decode the coded packets with length distribution Π_λ to recover the source packets with length distribution Π_λ. However, in order to guard against packet loss or corruption and finite field dependencies, additional coded MSs should be sent. MS RLNC opens a wide design space for coded MS transmission policies. One elementary policy is to send a prescribed number Ψ of full-length coded packets beyond the number of full-length coded packets $\Pi_{\Lambda_{\max}}$ in the source packet length distribution, before shortening the coded packets according to the source packet distribution. For instance, for $\Psi = 1$ in the example in Fig. 9, the coded packet lengths are 5, 5, 1, and 1 MS.

An alternative progressive shortening-and-hold policy may hold the coded packet length at prescribed lengths, e.g., full-length packet and half-length packet, for a prescribed number of coded packets Ψ before resuming the progressive shortening. For instance with $\Psi = 1$ and holding at the full- and half-packet length, the source packets of lengths 8, 7, 5, 5, 4, 2, 1, and 1 result in coded packet length 8, 8, 5, 5, 4, 4, 1, and 1. Another approach for controlling the number of coded MS is to exchange feedback between receiver and sender. In the evaluations in this study, we employ the elementary policy with Ψ additional full-length coded packets. The detailed examination of other policies for controlling the numbers of coded MSs is an important direction for future research.

B. MS RLNC Decoding

1) General Full-Length Packet Decoding: This section explains our proposed MS RLNC decoding schemes. We assume initially full-length coded packets, see Section III-A7a and then proceed to explain the modifications for the min-sized last coded packet and progressive shortening policies. MS RLNC decoding is generally based on on-the-fly Gaussian Elimination [68]. In MS RLNC on-the-fly Gaussian Elimination, the triangulation step starts as soon as a new set of Φ_η coded MS is received in a coded packet η, instead of waiting for a batch of coded packets to arrive. The main MS RLNC decoding strategy is to form a triangular matrix of dimension $\sum_{n=1}^{N} \Lambda_n \times \sum_{n=1}^{N} \Lambda_n$ by incorporating every individual received coded MS, starting from the very first received coded MS. Similarly to conventional RLNC decoding, the receiver recovers the original source packets by exploiting the coding coefficients $\alpha_{\eta n}$ carried in the transmitted packets.

Importantly, for MS RLNC the individual packet sizes $\Lambda_n,\ n = 1, 2, \ldots, N$, are needed for the construction of the decoding vectors that correspond to each coded MS received within a coded packet. These decoding vectors can be viewed as an extension to a decoding matrix of size $\Lambda_{\max} \times \sum_{n=1}^{N} \Lambda_n$ as illustrated on the right side of Fig. 7. Each column set of source symbols $s_{1\lambda}, s_{2\lambda}, \ldots, s_{N\lambda}$ can be entirely recovered after receiving the coded MS in column position λ of the coded packet number Δ_λ, i.e., after decoding exactly $\lambda + \Lambda_{\max}(\Delta_\lambda - 1)$ coded MSs, analogous to conventional RLNC decoding [62], [69]–[71].

2) Shortened Packet Decoding: With the min-sized last packet policy, the destination receives coded packets of size $\Lambda_{\max}$ coded MSs, except for one coded packet, which contains $\Lambda_{\min}$ coded macro-symbols. With progressive shortening, the destination receives (absent any packet losses) coded packets with a length distribution matching the length distribution of the source packets. For any such policy with shortened coded packets, after expanding the encoding vector into a decoding matrix illustrated on the right side of Fig. 7, the decoding process starts with the first received coded MS, following the conventional RLNC decoding [62], [69]–[71].

Generally, in RLNC decoding, it is important to check whether a received coded packet is innovative or not, which determines whether the packet is kept for decoding or ignored [62], [69]–[71]. An attractive feature of MS RLNC is that it does not require checking the dependency for every single received coded MS. Based on the checking of the first coded MS $c_{\eta 1}$ in a given coded packet C_η, the receiver can determine for the entire set of coded MSs in the coded packet C_η whether they are innovative or not. In particular, the same set of random coding coefficients $\alpha_{\eta 1}, \alpha_{\eta 2}, \ldots, \alpha_{\eta N}$ is used in the encoding of all coded MSs in the considered coded packet C_η. That is, the set of random coding coefficients $\alpha_{\eta 1}, \alpha_{\eta 2}, \ldots, \alpha_{\eta N}$ for obtaining the coded MS $c_{\eta\lambda}$ is exactly the same as for obtaining the coded MS $c_{\eta(\lambda-1)}$ if $\Delta_\lambda = \Delta_{\lambda-1}$, or a reduced set when $\Delta_\lambda < \Delta_{\lambda-1}$. Thus, the rank remains unchanged and the entire set of output macro-symbols can be discarded on the receiver side when the first coded MS $c_{\eta 1}$ in a given coded packet C_η is determined not to be innovative. This reduces the computational effort invested in linearly dependent packets.

IV. Shifting Schemes for MS RLNC

A. Motivation

A critical aspect of MS RLNC decoding of a generation of N source packets is that with progressive shortening, the leftmost column of source symbols has a source MS degree of $\Delta_1 = N$, see Fig. 9. That is, there are N MSs in column position $\lambda = 1$ (and possibly there are more columns with degree N). Hence, the decoder requires at least N coded packets to recover the set(s) of these N original source MSs. Reducing the maximum degree $\max_{1 \le \lambda \le \Lambda_{\max}} \Delta_\lambda$, i.e., the longest vertical column of source MSs, will reduce the number of unknowns per linear equation in the RLNC decoding and hence reduce the number of coded packets required at the receiver for decoding. The key idea of the shifting schemes is to rearrange the source packets so that the maximum degree is minimized.

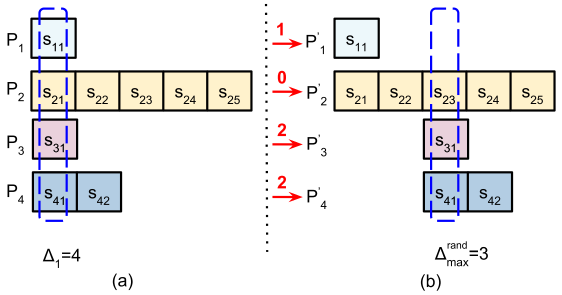


Fig. 10. Example of random shifting: (a) Generation of $N = 4$ original non-shifted packets with unequal sizes; the maximum degree is $\max_{1 \le \lambda \le \Lambda_{\max}} \Delta_\lambda = \Delta_1 = 4$. (b) Randomly shifted packets before vertical encoding; the shifting reduced the maximum degree to $\max_{1 \le \lambda \le \Lambda_{\max}} \Delta_\lambda^{\text{rand}} = \Delta_{\max}^{\text{rand}} = 3$.

We note that the shifting of the source packets is a pre-coding step that occurs only at the encoder. Recoding in the network is oblivious of the pre-coding shifting and will operate as in standard RLNC.

B. Random Shifting

Random shifting moves the source MS $s_{n\lambda}$ in column position λ of source packet n by a random offset to the right, i.e., into the column position corresponding to a higher column index, as illustrated in Fig. 10. In particular, the random offset ϕ_n is drawn randomly from a uniform distribution over $[0,\ \Lambda_{\max} - 1]$. The moving to a higher indexed column position wraps around at column position Λ, i.e., the source MS $s_{n,\lambda}$ is moved to the column position $\lambda + \phi_n \mod \Lambda_{\max}$. The MS RLNC encoding for the specific example of random shifting in Fig. 10 is further illustrated in Fig. 11. Note that the shifts are added to the coded packet header (and need to be sent only once per generation, similar to the packet sizes).

A drawback of random shifting is that in the worst case, the random shifts could leave the maximum source MS degree, i.e., the maximum column length of source MSs, unchanged at N, the number of source packets in the generation. An advantage of random shifting is that it could be amenable to supporting privacy and security mechanisms. We proceed to design a deterministic shifting scheme that minimizes the maximum source MS degree.

C. Deterministic Shifting

Deterministic shifting is inspired by the chain and fragmentation approach [44], where the input packets are first chained together in their original order to form a lumped data set, and are then fragmented into new equal size packets. In the context of MS RLNC, we deterministically right shift the source MSs

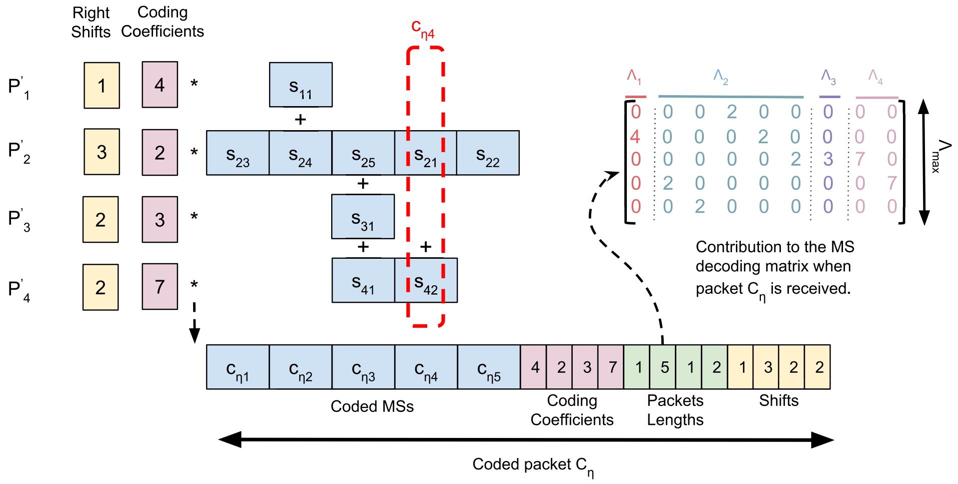


Fig. 11. Example of MS RLNC encoding after random shifting for generation size $N = 4$ and maximum source packet size $\Lambda_{\max} = 5$ MSs with the eventual matrix expansion of the encoding vector.

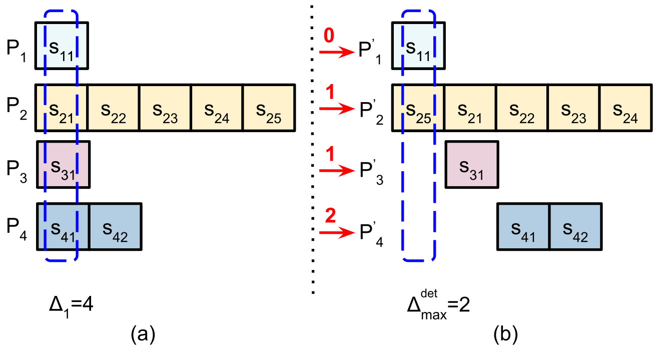


Fig. 12. Deterministic shifting: (a) Generation of $N = 4$ source packets with unequal sizes. (b) The deterministic shifting has reduced the maximum MS source degree to $\Delta^{\text{det}}_{\max} = 2$.

so that the successive source packets essentially form one long chain that continuously cycles through the column positions $1, 2, \ldots, \Lambda_{\max}, 1, 2, \ldots \Lambda_{\max}, 1, 2, \ldots$ More specifically, we right shift the first MS $s_{n,\lambda}$ of source packet n into the column position following the position of the last MS $s_{n-1,\Lambda_{n-1}}$ of the preceding source packet $n-1$ *after* the right shifting of source packet $n-1$. In particular, the first source packet ($n = 1$) is not shifted, i.e., the last MS s_{1,Λ_1} of the first source packet stays in column position Λ_1 corresponding to the length (in MSs) of packet $n = 1$. The first MS of the second packet, i.e., MS $s_{2,1}$ is right shifted into the position immediately to the right of column position Λ_1, i.e., into position $\Lambda_1 + 1 \mod \Lambda_{\max}$ (where the $\mod \Lambda_{\max}$ accounts for the wrap-around). Then, the other MSs of source packet $n = 2$ are placed successively to the right (with wrap-around, if needed) of its first MS, as illustrated in the second row of Fig. 12. This right shifting continues for the subsequent source packets. In the example in Fig. 12, the last source MS of packet $n = 2$ is shifted into column position 1; thus, the first MS $s_{3,1}$ of packet $n = 3$ is right shifted into column position 2, and so on.

The chaining of the source packet through the right shifting effectively creates one long chain with $\sum_{n=1}^{N} \Lambda_n$ MSs. This long chain fills the column positions $1, 2, \ldots, \Lambda_{\max}$ completely for a total of $\lfloor \sum_{n=1}^{N} \Lambda_n / \Lambda_{\max} \rfloor$ times and "spills" over $(\sum_{n=1}^{N} \Lambda_n / \Lambda_{\max}) \mod \Lambda_{\max}$ MSs into the next row. Thus, the maximum source MS degree is

$$\Delta^{\text{det}}_{\max} = \left\lceil \sum_{n=1}^{N} \Lambda_n \Big/ \Lambda_{\max} \right\rceil. \tag{5}$$

This maximum source MS degree $\Delta^{\text{det}}_{\max}$ is a lower bound for the expected number K of coded packets required for decoding the generation of N source packets at the receiver, i.e., $K \geqslant \Delta^{\text{det}}_{\max}$.

Note that with deterministic shifting, the shifts do not need to be signalled to the receiver, as they follow deterministically from the packet sizes that are carried in the packet header to the receiver, as illustrated in Fig. 11. Both shifting schemes can employ different coded packet transmission policies. A full-length coded packet policy transmits K, $K \geq \Delta^{\text{rand/det}}_{\max}$, full-length coded packets η, $\eta = 1, 2, \ldots, K$, each containing $\Phi_\eta = \Lambda_{\max}$ coded MSs. The probability of decoding a generation of N source packets from K received coded packets is derived in the following section and can be used to dimension the number of coded packets K according to quality of service requirements. Alternatively, a min-sized last coded packet policy (analogous to Section III-A7b) transmits $\Delta^{\text{rand/det}}_{\max} - 1$ full-length coded packets followed by a "spill-over" packet consisting of $(\sum_{n=1}^{N} \Lambda_n / \Lambda_{\max}) \mod \Lambda_{\max}$ MSs.

V. Analytical Characterization of MS RLNC

This section analyzes the decoding probability as well as the computational complexities of MS RLNC encoding and decoding.

A. Decoding Probability

MS RLNC multiplies the coding coefficients $\alpha_{\eta n}$, $n = 1, 2, \ldots, N$, with the source MSs in the column position λ (or the source MSs that have been shifted into the column position λ) in (packet) rows n to compute coded MS $c_{\eta\lambda}$. We define

$\mathcal{A}_\lambda$ as the set of coding coefficients that are actually involved in the computation of coded MS $c_{\eta\lambda}$, i.e., the coefficients that have "encountered" an actual source MS and we define these coding coefficients as *actual coding coefficients*. We denote $a_\lambda = |\mathcal{A}_\lambda|$ for the number of actual coding coefficients involved in computing coded symbol $c_{\eta\lambda}$. Furthermore, we define $a_{\ell\lambda} = |\mathcal{A}_\lambda \cap (\bigcup_{1\leq\ell<\lambda} \mathcal{A}_\ell)|$ as the number of actual coding coefficients that were involved in computing coded symbol $c_{\eta\lambda}$ and in computing any of the preceding coded symbols $c_{\eta\ell},\ 1 \leq \ell < \lambda$, in the considered coded packet η. For instance, in the example in Fig. 12, $a_1 = 2$, $a_2 = 2$, and $a_{\ell 2} = 1$.

We define $E_\lambda^{(K)},\ \lambda = 1, 2, \ldots, \Lambda_{\max}$, as the event of decoding the Δ_λ coded MSs in column position λ after the receipt of the Kth coded packet. The event $E_1^{(K)}$ of decoding the coded MSs in column position $\lambda = 1$, and the event $E_2^{(K)}$ for column position $\lambda = 2$, and so on, up to and including the event $E_{\Lambda_{\max}}^{(K)}$ for column position $\lambda = \Lambda_{\max}$ all joined together correspond to the event of decoding the entire generation, i.e., the complete set of N (shifted) source symbols after the receipt of coded packet K. Thus, the probability of decoding the generation with K received coded packets is

$$P_{\text{dec}}^{(K)} = \mathbb{P}\left(\bigcap_{1\leq\lambda\leq\Lambda_{\max}} E_\lambda^{(K)}\right). \tag{6}$$

For conventional packet based RLNC with GF(2^q), N source packets are decoded after the receipt of $K,\ K \geq N$, coded packets with probability [72], [73]:

$$\prod_{n=0}^{N-1}\left(1 - \frac{1}{2^{q(K-n)}}\right). \tag{7}$$

In MS RLNC, there are Δ_1 MSs in column position $\lambda = 1$, which were encoded with $a_1 = \Delta_1$ actual coding coefficients. Replacing N in Eqn. (7) with a_1 gives the probability for decoding these $a_1 = \Delta_1$ MSs from $K,\ K \geq \Delta_{\max}$, received coded packets, i.e.,

$$\mathbb{P}\left(E_1^{(K)}\right) = \prod_{n=0}^{a_1-1}\left(1 - \frac{1}{2^{q(K-n)}}\right). \tag{8}$$

We proceed to evaluate the probability of decoding column 1 (event $E_1^{(K)}$) and column 2 (event $E_2^{(K)}$) with the conditional probability of decoding column 2 given that column 1 has already been decoded, i.e., we evaluate $\mathbb{P}(E_1^{(K)} \cap E_2^{(K)}) = \mathbb{P}(E_1^{(K)}) \cdot \mathbb{P}(E_2^{(K)}|E_1^{(K)})$. The Δ_2 MSs in column position 2 have been encoded with a_2 actual coding coefficients. However, $a_{\ell 2}$ of these actual coding coefficients have been involved in the coding of the MSs in column 1, which have already been decoded. Thus, there are only $a_2 - a_{\ell 2}$ new coding coefficients that have not been involved in decoding the preceding columns. Thus, only $a_2 - a_{\ell 2}$ new coding coefficients need to effectively be considered in the decoding of the present column. However, there are also effectively only $K - (a_2 - a_{\ell 2})$ new received coded packets available that have not been involved in the decoding of the preceding columns. Hence, the conditional decoding probability is

$$\mathbb{P}\left(E_2^{(K)}|E_1^{(K)}\right) = \prod_{n=0}^{a_2-a_{\ell 2}-1}\left(1 - \frac{1}{2^{q(K-(a_2-a_{\ell 2})-n)}}\right). \tag{9}$$

Continuing this reasoning for the subsequent columns up to and including column $\Lambda_{\max}$ gives

$$P_{\text{dec}}^{(K)} = \mathbb{P}\left(E_1^{(K)}\right) \times \cdots \times \mathbb{P}\left(E_{\Lambda_{\max}}^{(K)} \middle| \bigcap_{\lambda<\Lambda_{\max}} E_\lambda^{(K)}\right). \tag{10}$$

Thus, with product notation,

$$P_{\text{dec}}^{(K)} = \prod_{\lambda=1}^{\Lambda_{\max}} \mathbb{P}\left(E_\lambda^{(K)} \middle| \bigcap_{\nu<\lambda} E_\nu^{(K)}\right), \tag{11}$$

where by, analogous to Eqn. (9),

$$\mathbb{P}\left(E_\lambda^{(K)} \middle| \bigcap_{\nu<\lambda} E_\nu^{(K)}\right) = \prod_{n=0}^{a_\lambda-a_{\ell\lambda}-1}\left(1 - \frac{1}{2^{q(K-(a_\lambda-a_{\ell\lambda})-n)}}\right). \tag{12}$$

Therefore, combining Eqns. (11) and (12),

$$P_{\text{dec}}^{(K)} = \prod_{\lambda=1}^{\Lambda_{\max}} \prod_{n=0}^{a_\lambda-a_{\ell\lambda}-1}\left(1 - \frac{1}{2^{q(K-(a_\lambda-a_{\ell\lambda})-n)}}\right). \tag{13}$$

We have verified this decoding probability analysis with simulations that are not included due to space constraints.

B. Computational Complexity

This section analyzes the computational complexity of MS RLNC encoding and decoding and compares with benchmarks. The comparisons are summarized in Table III. We note that conventional RLNC operates on N packets of size $L_{\max}$ and has an encoding complexity of $\mathcal{O}(L_{\max}N^2)$ and a decoding complexity of $\mathcal{O}(N^3)$ [59], [62], [74]–[76]. Also, for benchmarking, we note that the bin packing involved in simple bundling [44] can be solved with complexity $\mathcal{O}(N^2)$ with the First Fit Decreasing heuristic [77], resulting in $N_{\text{SB}},\ N_{\text{SB}} \leq N$, packets of size $L_{\max}$ for the RLNC encoder. Chaining and fragmentation results in N_{CF} packets of size F_{CF} for the RLNC.

1) Computational Complexity of MS RLNC Encoding: The MS RLNC encoding complexity is proportional to the number of source MSs considered in the computation of the coded MSs. In particular, computing a coded packet η of size Φ_η coded MSs requires the computation of Φ_η coded MSs. Computing one coded MS $c_{\eta\lambda}$ from source packets with source MS degree Δ_λ, requires that the Δ_λ source MSs, each of size μ symbols of GF(2^q), in column λ are multiplied with their corresponding coding coefficients and summed. We note that the encoding and decoding operations in the widely used Kodo RLNC library [71] are conducted on a symbol by symbol basis. Thus, irrespective of whether RLNC coding is performed on packets or on MSs (which may be viewed as small packets), the encoder still has to process all symbols (e.g., bytes in GF(2^8)) one by one. The Δ_λ multiplications and

TABLE III
COMPARISON OF ENCODING AND DECODING COMPLEXITY FOR A GENERATION CONSISTING OF N PACKETS. THE NUMBER OF UNKNOWNS INDICATES THE NUMBER OF UNKNOWN ELEMENTS DURING RLNC DECODING. THE MINIMUM NUMBER OF COEFFICIENTS INDICATES THE MINIMUM NUMBER OF CODING COEFFICIENTS REQUIRED FOR RLNC DECODING (IN ABSENCE OF LINEAR DEPENDENCIES OF CODING COEFFICIENTS AND LOSSES)

Scheme	Element size	# Unknowns	Min. # coef	Encoding	Decoding
Conv. RLNC	$L_{\max}$	N	N^2	$\mathcal{O}(L_{\max}N^2)$	$\mathcal{O}(N^3)$
Simple bundl. [44]	$L_{\max}$	N_{SB}	N_{SB}^2	$\mathcal{O}(N^2 + L_{\max}N_{\mathrm{SB}}^2)$	$\mathcal{O}(N_{\mathrm{SB}}^3)$
Chain and Fragm. [44]	$L_{\max}$	N_{CF}	N_{CF}^2	$\mathcal{O}(L_{\max}N_{\mathrm{CF}}^2)$	$\mathcal{O}(N_{\mathrm{CF}}^3)$
Progr. short., Sec. III-A7c	μ	$\sum_{\lambda=1}^{\Lambda_{\max}} \Lambda_\lambda$	N^2	$\mathcal{O}(\mu\Lambda_{\max}\Delta_{\max})$	$\mathcal{O}(\Delta_{\max}^3) - \mathcal{O}(\Lambda_{\max}\Delta_{\max}^3)$
Random shift., Sec. IV-B	μ	$\sum_{\lambda=1}^{\Lambda_{\max}} \Lambda_\lambda$	${\Delta_{\max}^{\mathrm{rand}}}^2$	$\mathcal{O}(\mu\Lambda_{\max}\Delta_{\max}^{\mathrm{rand}})$	$\mathcal{O}({\Delta_{\max}^{\mathrm{rand}}}^3) - \mathcal{O}(\Lambda_{\max}{\Delta_{\max}^{\mathrm{rand}}}^3)$
Det. shift., Sec. IV-C	μ	$\sum_{\lambda=1}^{\Lambda_{\max}} \Lambda_\lambda$	${\Delta_{\max}^{\mathrm{det}}}^2$	$\mathcal{O}(\mu\Lambda_{\max}\Delta_{\max}^{\mathrm{det}})$	$\mathcal{O}({\Delta_{\max}^{\mathrm{det}}}^3) - \mathcal{O}(\Lambda_{\max}{\Delta_{\max}^{\mathrm{det}}}^3)$

sums for obtaining one coded MS incur a computational complexity $\mathcal{O}(\mu\Delta_\lambda)$ for column λ. Summing over all considered columns $\lambda,\ \lambda = 1, 2, \ldots, \Phi_\eta \leq \Lambda_{\max}$, to obtain one coded packet consisting of Φ_η coded MSs incurs the computational complexity

$$\mathcal{O}\left(\mu \sum_{\lambda=1}^{\Phi_\eta} \Delta_\lambda\right). \qquad (14)$$

The computational complexity for encoding an entire generation of N source packets is $\mathcal{O}(\mu \sum_{\eta=1}^{N} \sum_{\lambda=1}^{\Phi_\eta} \Delta_\lambda)$. In comparison, conventional packet based RLNC encoding of N source packets that are padded to the full length of $L_{\max}$ bytes incurs computational complexity $\mathcal{O}(L_{\max}N^2)$ (each column has source MS degree $\Delta_\lambda = N$ in a generation of N full-length source packets). Considering a symbol size of one byte (different symbol sizes would require $L_{\max}$ to be re-defined in terms of number of symbols), we note that $\mu \sum_{\eta=1}^{N} \sum_{\lambda=1}^{\Phi_\eta} \Delta_\lambda \leq L_{\max}N^2$, i.e., MS RLNC with packet shortening ($\Phi_\eta \leq \Lambda_{\max}$) has the same or lower encoding complexity than conventional packet based RLNC. The preceding analysis considered a code rate of one (i.e., no extra coded packets for redundancy); each extra full-length coded packet would incur the computational complexity in Eqn. (14) with $\Phi_\eta = \Lambda_{\max}$ for MS RLNC and the complexity $\mathcal{O}(L_{\max}N)$ for conventional RLNC.

2) MS RLNC Decoding: The decoding needs to recover $\sum_{\eta=1}^{N} \Lambda_n$ source MSs, which is equivalent to decoding an $\sum_{n=1}^{N} \Lambda_n \times \sum_{n=1}^{N} \Lambda_n$ matrix. For analyzing the decoding complexity, consider using the coding coefficients $\alpha_{\eta\lambda},\ \lambda = 1, 2, \ldots, \Lambda$, and source packet lengths $\Lambda_n,\ n = 1, 2, \ldots, N$, in the received coded packet C_η to create a decoding matrix with $\Lambda_{\max}$ rows and $\sum_{n=1}^{N} \Lambda_n$ columns, see right sides of Figs. 7 and 11. Although the coding coefficients are drawn at random, their significance and location in the decoding matrix are deterministic. That is, row λ of any decoding matrix contains the coefficients related to the MSs in column position λ of the source packets. Therefore, we can divide the decoding into sub-decoding steps, similar to decoding non-overlapping generations [60]. As the example in Fig. 7 illustrates, we effectively need to decode $\Lambda_{\max} = 3$ matrices corresponding to the $\Lambda_{\max} = 3$ rows of the decoding matrix. These individual matrices have orders $\Delta_1 \times \Delta_1 = 2 \times 2$, $\Delta_2 \times \Delta_2 = 1 \times 1$, and $\Delta_{\Lambda_{\max}} \times \Delta_{\Lambda_{\max}} = \Delta_3 \times \Delta_3 = 1 \times 1$, as there are 2, 1, and 1 elements in the rows of the decoding matrix Following the Gaussian elimination approach, inverting these matrices has a computational complexity of:

$$\mathcal{O}\left(\sum_{\lambda=1}^{\Lambda_{\max}} \Delta_\lambda^3\right) = \mathcal{O}\Big(\Lambda_{\max}N^3\Big), \qquad (15)$$

which is higher than the $\mathcal{O}(N^3)$ computational complexity of conventional packet based RLNC.

This preceding analysis assumes that the columns $\lambda,\ \lambda = 1, 2, \ldots, \Lambda_{\max}$ are decoded entirely independently, which is not practical. Instead, a practical decoder decodes the columns jointly, i.e., is aware of the coding coefficients whose corresponding source MSs have been decoded in a preceding column. Thus, as noted in Section V-A for decoding of the first column $\lambda = 1$, all $a_1 = \Delta_1$ corresponding actual coding coefficients need to be considered, resulting in a complexity contribution of $\mathcal{O}(a_1^3) = \mathcal{O}(\Delta_1^3) = \mathcal{O}(N^3)$.

For each subsequent column $\lambda,\ \lambda = 2, 3, \ldots, \Lambda_{\max}$, the decoding is simplified if the set $\mathcal{A}_\lambda$ of actual coding coefficients for column λ is identical to one of the sets of the actual coding coefficients for any of the preceding columns $\mathcal{A}_\ell,\ 1 \leq \ell < \lambda$. If $\mathcal{A}_\lambda = \mathcal{A}_\ell$ for some prior column $\ell,\ 1 \leq \ell < \lambda$, then only the final Gaussian elimination step with complexity contribution $\mathcal{O}(\Delta_\lambda^2) = \mathcal{O}(N^2)$ needs to be completed. Thus, in the best case when $\mathcal{A}_\lambda = \mathcal{A}_1$ for all $\lambda,\ \lambda = 2, 3, \ldots, \Lambda_{\max}$, the computational complexity of MS RLNC decoding is

$$\mathcal{O}\Big(N^3 + (\Lambda_{\max} - 1)N^2\Big) = \mathcal{O}\Big(N^3\Big). \qquad (16)$$

For instance, in the example in Fig. 7, the complexity contribution is $\mathcal{O}(N^3)$ for column $\lambda = 1$ of the source symbols (on the left), or equivalently for the first row of the decoding matrix (on the right). Similarly, column $\lambda = 2$ of the source symbols (or equivalently the second row of the decoding matrix) incurs a decoding complexity contribution $\mathcal{O}(N^3)$. However, column $\lambda = 3$ of the source symbols (third row of the decoding matrix) has exactly the same coding coefficient (the 4) as the preceding source symbol column (row of the decoding matrix); thus incurring only a decoding complexity of $\mathcal{O}(N^2)$. The complexities of the MS RLNC schemes are summarized in Table III and compared with the existing approaches.

VI. Numerical MS RLNC Evaluation

A. Evaluation Setup

1) Network Setting: In order to focus on the padding overhead, we consider initially a lossless unicast network transmission with RLNC encoding at the sender and RLNC decoding at the receiver. Subsequently, in Section VI-B3 we consider a lossy network.

2) MS RLNC Implementation: We implemented the introduced MS RLNC coding schemes using the Kodo library [71]. Building on the functionalities of the Kodo library, we designed the MS RLNC encoding and decoding modules as wrappers of the corresponding Kodo modules. Our encoder wrapper maps the source data packets to source MSs. To encode the data, we generate random coefficients, form the corresponding coding vectors, and execute the Kodo encoder to produce the coded packets. Finally, we append the metadata to the coded packet, i.e., the coding coefficients, the packet lengths (and the packet shifts for random shifting). For decoding, we map the coding vector from each received coded packet to a coding matrix. Then we feed the coding matrix along with the coded data to the Kodo decoder.

3) Video Traces and Partitioning Into MSs: We consider the transmission of the full SVC (CIF), VP9 (HD), and H.265 (4k) video traces, see Section II-A. That is, we consider the transmission of 289040 successive SVC (CIF) video frames as well as 17592 VP9 (HD) video frames and 17592 H.265 (4k) video frames. These video traces correspond to 336336, 45388, and 163920 packets, respectively, which are grouped into generations consisting of N packets each. Based on the evaluations in Section III-A4, we set the MSs size to $\mu = 60$ bytes; thus, one MS corresponds to 480 symbols in GF(2) or 60 symbols in GF(2^8).

4) Compared Padding Reduction Approaches: We evaluate the performance of the following MS RLNC approaches: progressive shortening (Section III-A7c) as well as random shifting (Section IV-B), and deterministic shifting (Section IV-C) with the min-sized last packet policy. We compare these MS RLNC approaches against conventional packet based RLNC. Moreover, we compare with simple bundling as well as chaining and fragmentation [44]. Following the recommendations in [44], we set the largest packet size of simple bundling as well as the fragmentation size of chaining and fragmentation to the largest packet size $L_{\max}$ of a given generation.

5) Metrics: We evaluate the overhead percentage, i.e., the total amount of padding overhead [bytes] to the total amount of data [bytes] that needs to be transmitted over the communication network between encoder and decoder for each generation consisting of N packets. We report the overhead for all the generations constituting the considered video traces in box plots.

We also evaluate the number K of coded packets required to complete the RLNC decoding at the receiver of each of the generations constituting the considered video traces. We conducted 100 independent replications of the evaluation of K for each generation to obtain an average K value for each generation, which we refer to as generation K value. The 95% confidence intervals of the generation K values were less than 6% of the corresponding sample means for GF(2^8) and less than 12% of the corresponding sample means for GF(2). These results conform to corresponding general analyses for RLNC decoding [78].

We present box plots that were obtained from these generation K values for a given video trace. The box plots give the median as well as lower and upper quartiles of the individual generation K values across all the generations in a given video trace. The box plot whiskers give the 10% and 90% quantiles; outliers below 10% and above 90% are represented through dots. The K values represent a delay measure, i.e., the total number of coded packet transmissions required to transport a generation of video packets to the receiver.

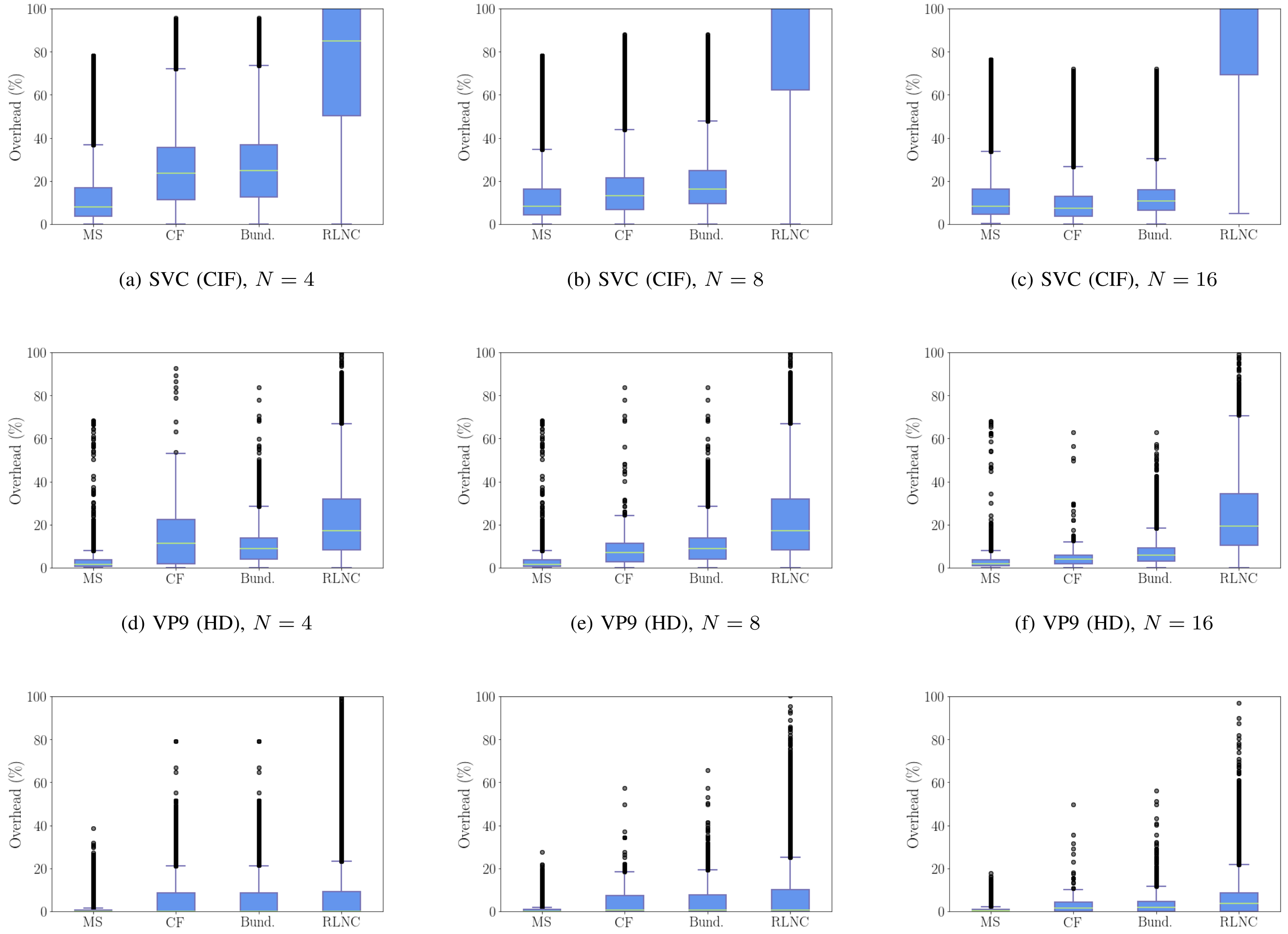
B. Results

1) Padding Overhead: Fig. 13 presents box plots of the overhead for the individual generations, each consisting of N successive video packets. The box plots marked with "MS" represent both MS RLNC with progressive shortening and MS RLNC with deterministic shifting. (MS RLNC with random shifting achieves the same MS results when combined with a progressive shortening policy.) We observe from Fig. 13 that the MS RLNC approaches achieve significantly lower overheads than chaining and fragmentation (CF) and simple bundling (Bund) for the small generation sizes $N = 4$ and 8. On the other hand, the MS RLNC approaches have similar overheads as chaining and fragmentation as well as bundling for generation sizes of $N = 16$ and larger (as we have verified in additional evaluations that are not included to avoid clutter).

Chaining and fragmentation as well as bundling encode complete packets of size $L_{\max}$. Padding overhead arises in chaining and fragmentation due to the long string of chained packets typically having a length $\sum_{n=1}^{N} L_n$ that is not an integer multiple of $L_{\max}$. In particular, there are $\lfloor \sum_{n=1}^{N} L_n / L_{\max} \rfloor$ fragments of size $L_{\max}$ and one (last) fragment of size $\sum_{n=1}^{N} L_n \mod L_{\max}$. This last fragment needs to be padded with $L_{\max} - (\sum_{n=1}^{N} L_n \mod L_{\max})$ zero bits before RLNC encoding. Assuming a uniform distribution of the size of the last fragment size, the padding overhead is on average $L_{\max}/2$. Thus, following Eqn. (1), the corresponding average overhead percentage is $L_{\max}/(2\sum_{n=1}^{N} L_n)$, which is inversely proportional to the generation size N. (Reducing the fragmentation size significantly below $L_{\max}$ would significantly increase the number of fragments and thus the number of required coding coefficients, and is therefore not a viable strategy for reducing the overhead.) The percentage overhead of simple bundling is similarly roughly inversely proportional to the generation size N.

In contrast, the MS RLNC approaches pad each individual packet to an integer multiple of the MS size μ. Assuming that the packet sizes are integer multiples of μ plus a "spillover" part that is uniformly distributed over μ, the average padding overhead is $\mu/2$ per packet. The corresponding percentage overhead is $N\mu/(2\sum_{n=1}^{N} L_n)$, which is independent of the generation size N.

We note that MS RLNC with deterministic shifting as introduced in Section IV-C first pads the packets to an integer



(a) SVC (CIF), $N = 4$

(b) SVC (CIF), $N = 8$

(c) SVC (CIF), $N = 16$

(d) VP9 (HD), $N = 4$

(e) VP9 (HD), $N = 8$

(f) VP9 (HD), $N = 16$

(g) H.265 (4k), $N = 4$

(h) H.265 (4k), $N = 8$

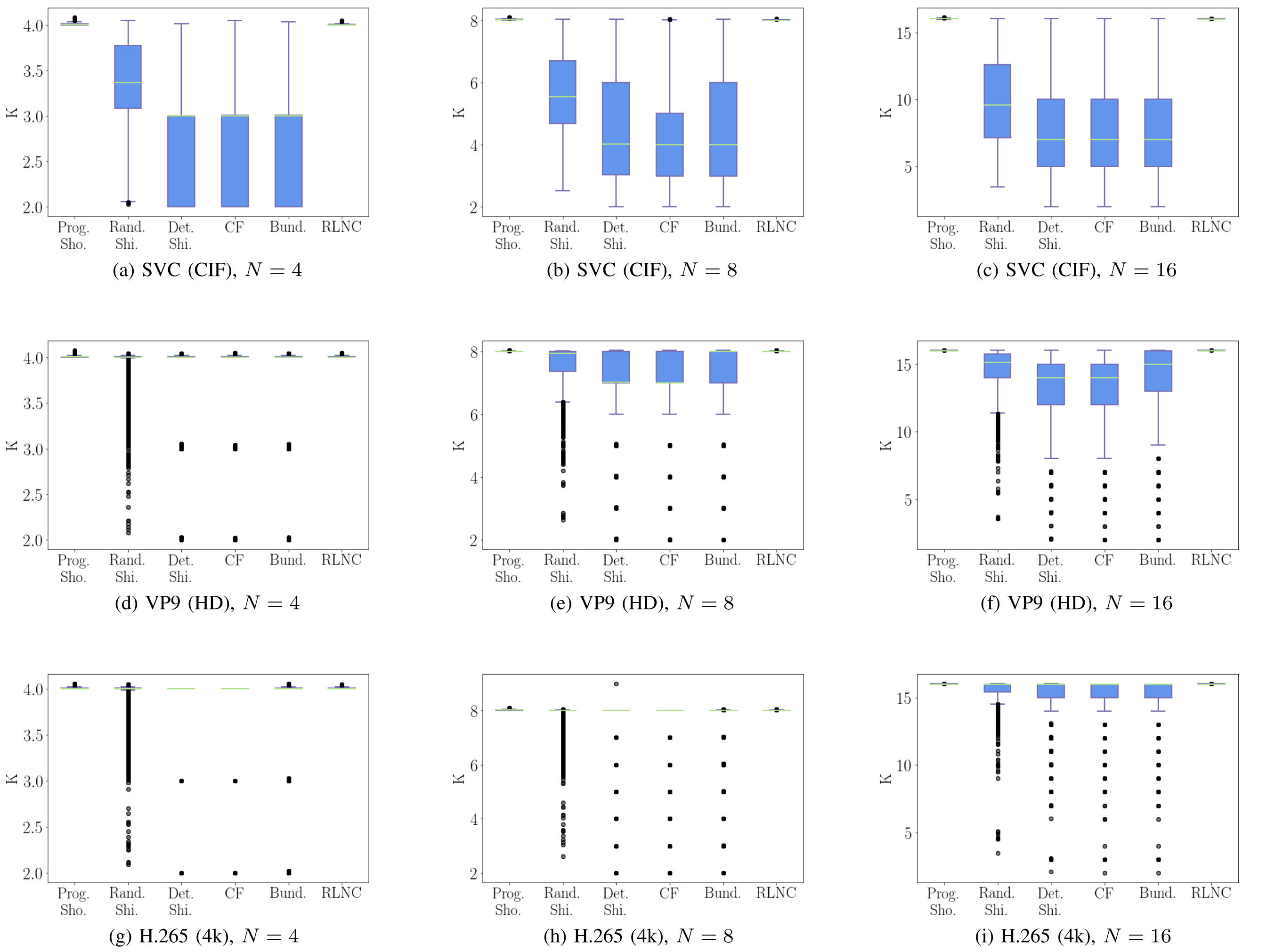
(i) H.265 (4k), $N = 16$

Fig. 13. Box plot comparison of padding overhead in percent for different generation sizes N. Fixed parameters: GF(2^8), $\mu = 60$ bytes, i.e., 25 MSs per packet.

number of MSs and then shifts the packets at the granularity of MSs. An alternative MS RLNC with deterministic shifting approach could first shift the packets at the granularity of individual symbols [bytes in GF(2^8)], analogous to chaining and fragmentation. Then, the shifted packets could be fragmented into MSs (whereby a given MS may contain parts of multiple packets) and encoded with MC RLNC. With this alternative approach, only the "spill over" into the last MS in the last row needs to be padded out to a full MS, reducing the average padding overhead to $\mu/2$ for the entire generation. This alternative approach needs to signal the packet sizes at the granularity of symbols, requiring up to $N \log_2 L_{\max}$ bits, while for the deterministic shifting approach in Section IV-C it is sufficient to signal the packet lengths at the granularity of MSs, requiring up to $N \log_2 \Lambda_{\max}$ bits; the difference of $N \log_2 \mu$ bits is negligible for most scenarios.

Overall we observe from Fig. 13 that the SVC (CIF) video generally requires higher padding overhead than the VP9 (HD) video, even with the various padding reduction techniques. The VP9 (HD) video in turn requires higher padding overhead than the H.265 (4k) video. The SVC (CIF) video has a higher probability of packet sizes below the MTU compared to the VP9 (HD) and H.265 (4k) videos (see Section II). The padding overhead generally increases with the tendency for the video to have small packets, as quantified in Section II. We also observed in Section II and we observe from the RLNC results in Fig. 13 that the padding overhead generally increases with increasing generation size N. In contrast, for all padding reduction approaches, the overhead decreases for increasing generation size N. Thus, one overarching conclusion from this padding overhead study is that padding overhead reduction is particularly critical for large generation sizes N. More specifically, we conclude that for small generation sizes N, the MS RLNC approaches are more effective than chaining and fragmentation or simple bundling. On the other hand, for moderate to large generation sizes, all the padding reduction approaches are approximately equally effective.

2) Number of Coded Packets K (Delay) Without Packet Losses: Figs. 14 and 15 compare the box plots for the number of coded packets K needed for the complete recovery of each generation of N packets from the video traces. Fig. 14 shows the box plots for GF(2^8) for a range of generation sizes N, while Fig. 15 shows the box plots for GF(2) for $N = 8$. We observe from Figs. 14 and 15 that conventional full-length



(a) SVC (CIF), $N = 4$

(b) SVC (CIF), $N = 8$

(c) SVC (CIF), $N = 16$

(d) VP9 (HD), $N = 4$

(e) VP9 (HD), $N = 8$

(f) VP9 (HD), $N = 16$

(g) H.265 (4k), $N = 4$

(h) H.265 (4k), $N = 8$

(i) H.265 (4k), $N = 16$

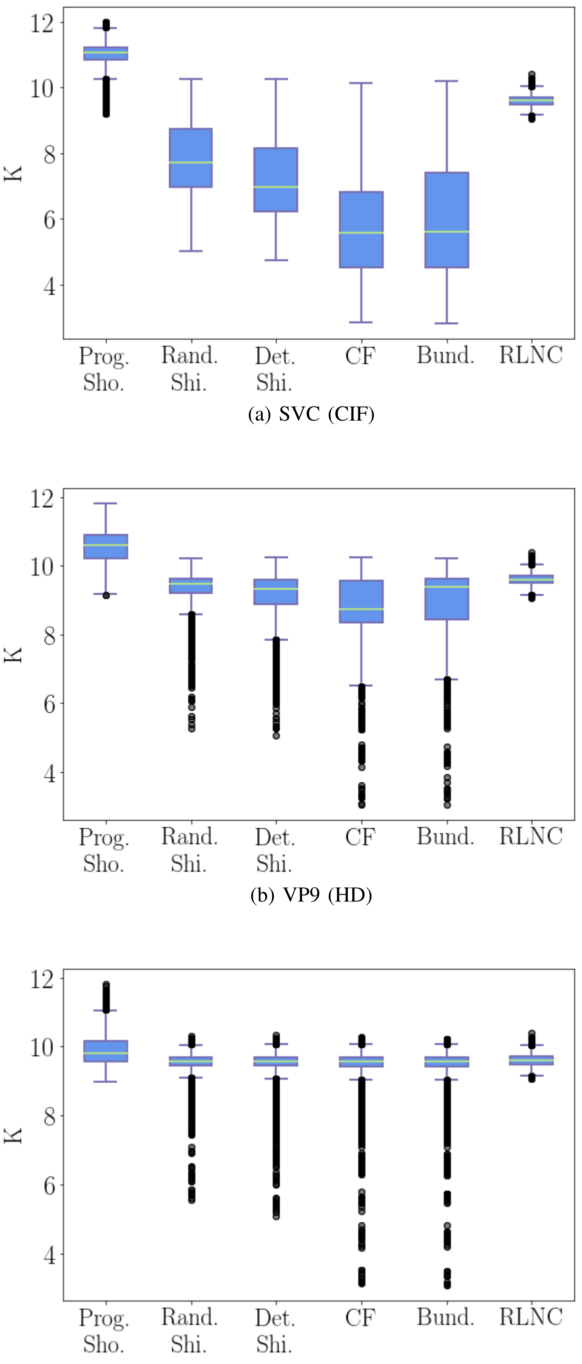
Fig. 14. Box plot comparison of the number K of coded packets needed to decode the N packets in a generation. Fixed parameters: GF(2^8), $\mu = 60$ bytes, i.e., 25 MSs per packet, lossfree network link.

packet RLNC and progressive shortening MS RLNC require almost always $K = N$ coded packets for GF(2^8) and slightly more coded packets for GF(2). The additional packets beyond N are required to overcome occurrences of linear dependent coding coefficients. Generally, it is important to keep in mind that the coded packet count K does not reflect that MS RLNC with progressive shortening produces shortened packets that require less transmission resources than full-length coded packets.

We also observe from Figs. 14 and 15 that MS RLNC with deterministic shifting as well as chaining and fragmentation and bundling tend to have the lowest K values; MS RLNC with random shifting tends to generally have slightly higher K values, but still lower values than conventional RLNC. Importantly, we observe from Fig. 14 that the third quartiles of K for deterministic shifting MS RLNC as well as for chaining and fragmentation are below the minimum K values for conventional RLNC for VP9 (HD) for $N = 16$ and for all SVC (CIF) scenarios. This means that in all these scenarios, deterministic shifting MS RLNC as well as chaining and fragmentation can decode more than 75% of the generations prior to receiving N coded packets at the decoder; whereby N is the minimum number of coded packets required for decoding conventional RLNC. The reduced numbers K of coded packets required with the padding reduction approaches thus indicate reduced latencies for the network transport of the generations of media packets.

These observed reductions of the number of required coded packets K are achieved through the "packing" of the N unequal size packets into fewer equivalent full-size packets with shifting of MSs or chaining and fragmentation or bundling of the packets. In particular, the padding reduction approaches judiciously exploit the varying packet sizes to "pack" the generation data into fewer rows for RLNC encoding and thus reduce the number of coded packets required for transporting a generation. Random shifting MS RLNC is somewhat less effective at this "packing" than the other approaches, mainly because random shifting may not minimize the length of the longest column, see Section IV-B.

Large generation sizes N and videos with a significant number of small packets provide extensive opportunities for the "packing" of unequal size packets into fixed size packets.



(a) SVC (CIF)

(b) VP9 (HD)

(c) H.265 (4k)

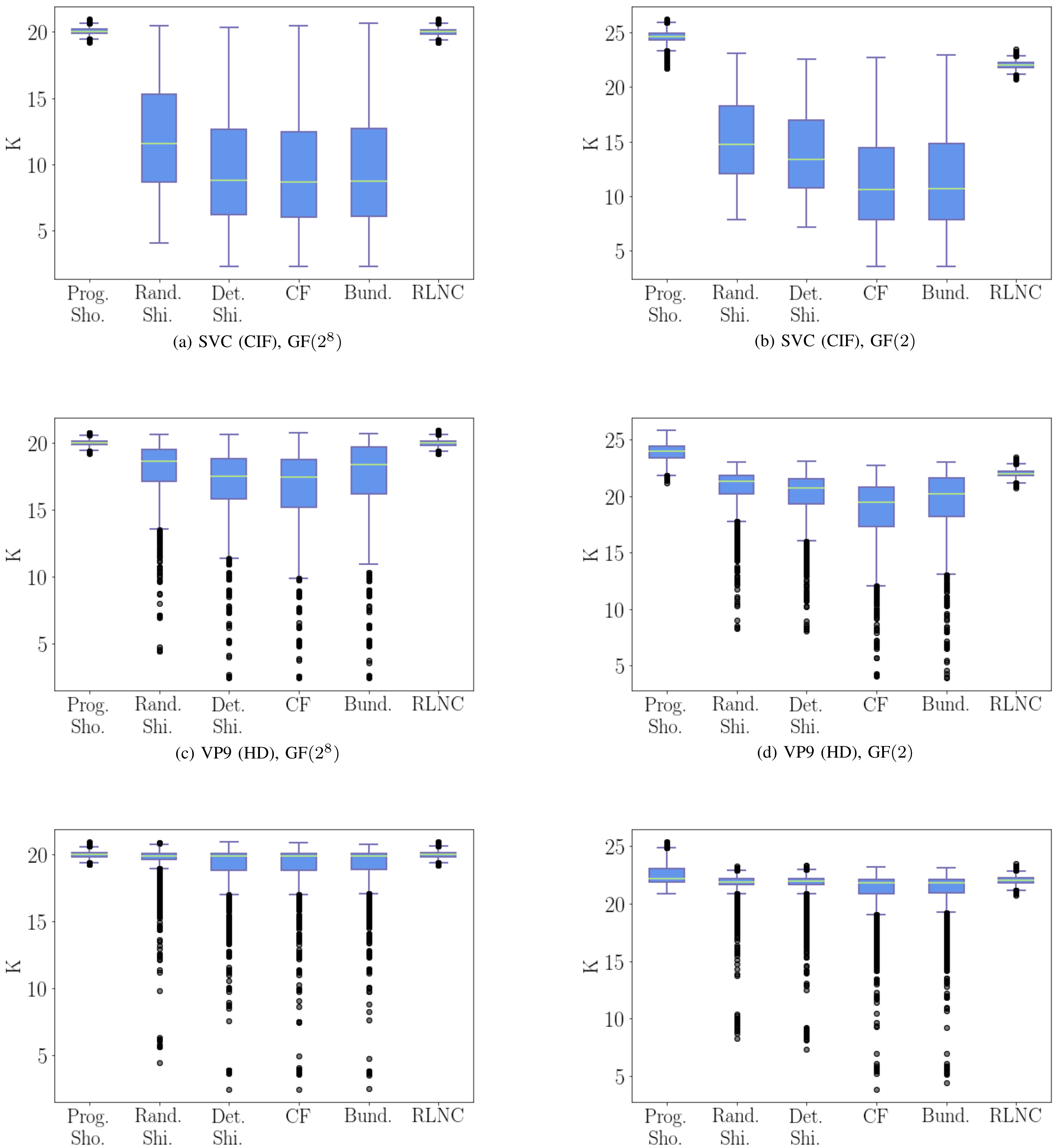
Fig. 15. Box plot comparison of the number K of coded packets for GF(2) needed to decode the $N = 8$ packets in a generation. Fixed parameters: $N = 8$ generation size, GF(2), $\mu = 60$ bytes, i.e., 480 MSs per packet, lossfree network link.

Therefore, we observe more pronounced reductions of the number K of required coded packets for the SVC (CIF) video (with a substantial portion of small frames, see Section II-A) than for the VP9 (HD) and H.265 (4k) videos. Also, the large considered generation size of $N = 16$ exhibits more pronounced reductions of K than the smaller $N = 4$ and 8 generation sizes. We verified in additional evaluations that are not included to avoid clutter that generation sizes above 16 further slightly increase the packing opportunities. For instance, for VP9 (HD) for $N = 64$ and GF(2^8) the upper (90%) whisker of K for deterministic shifting MS RLNC is at 62.

3) Number of Transmitted Coded Packets With Packet Losses: This section presents results for the number of required coded packet transmissions over a lossy link in order to achieve the complete recovery of each generation of $N = 16$ packets of the considered video traces. The considered link independently randomly drops 20% of the transmitted packets. We observe from Fig. 16 for the lossy network link similar performance trends as from Figs. 14 and 15 for the lossfree network link. The padding overhead reduction approaches generally continue to outperform conventional RLNC by requiring fewer packet transmissions. This result is expected since the padding overhead reduction schemes are designed to preserve the RLNC error correction (packet recovery) mechanisms. In particular, simple bundling as well as chaining and fragmentation are pre-processing steps that take place before the RLNC encoding at the sender and are undone after the RLNC decoding at the receiver. The MS RLNC with progressive shortening may make the last MSs (in the highest indexed column positions) in long packets more vulnerable to losses if small fields with a relatively high probability of correlated coding coefficient vectors are used. This is because these last MSs are included in fewer coded packets compared to MS RLNC schemes with mainly full-length coded packets (see Section III-A7c). The evaluation results in Fig. 16 indicate that this vulnerability effect is relatively mild, notice the very slightly increased median numbers of transmitted packets with progressive shortening compared to conventional RLNC for GF(2). For GF(2^8) with negligible coding coefficient vector correlations, we observe the same median numbers of transmitted packets for progressive shortening and conventional RLNC. Also, recall from the MS vs. RLNC comparison in Fig. 13 that the MS RLNC progressive shortening packets have significantly less overhead than the conventional RLNC packets.

MS RLNC with deterministic and random shifting employs mainly full-length packets (except for the last packet, which is sized according to the min-sized last coded packet policy, see Section III-A7b). Accordingly, MS RLNC with shifting has nearly equivalent packet recovery capabilities as conventional RLNC (even with correlated coding coefficient vectors). In particular, the evaluation results in Fig. 16 confirm that MS RLNC with deterministic shifting requires essentially the same low numbers of packet transmissions over the lossy link as chaining and fragmentation (while incurring the same or slightly lower padding overhead, see MS vs. CF comparison in Fig. 13).

Closely comparing Figs. 16(a), (c), and (e) for the transmission over the lossy network link with the corresponding Fig. 14(c), (f), and (i) for the transmission over the lossfree network link, we observe that the network link losses slightly amplify the differences between conventional RLNC and the padding reduction approaches. In particular, for the VP9 (HD) video, deterministic shifting as well as chaining and fragmentation have median K values of 14 compared to a median K value of 16 for RLNC in Fig. 14(f); whereas, deterministic



(a) SVC (CIF), GF(2^8)

(b) SVC (CIF), GF(2)

(c) VP9 (HD), GF(2^8)

(d) VP9 (HD), GF(2)

(e) H.265 (4k), GF(2^8)

(f) H.265 (4k), GF(2)

Fig. 16. Box plot comparison of the number of required coded packet transmissions over a link with 20% packet loss to decode the $N = 16$ packets in a generation. Fixed parameters: $N = 16$ generation size, $\mu = 60$ bytes, i.e., 25 MSs per packet for GF(2^8) and 480 MSs per packet for GF(2).

shifting as well as chaining and fragmentation have median K values of 17.49 as well as 17.45, respectively, compared to 20 for RLNC in Fig. 16(c). The padding overhead reduction approaches "pack" the unequal sized packets in a generation into fewer packets that need to be transmitted over the lossy network link. Fewer packet transmissions imply fewer lost packets and, hence, fewer required additional transmissions to enable RLNC decoding at the receiver. Overall, the results in Fig. 16 demonstrate that the padding reduction approaches are beneficial in reducing the number of required packet transmissions over lossy network links compared to conventional RLNC, especially for videos containing small frames.

VII. Conclusion

We have examined the overhead arising from the padding of variable size media packets, e.g., packets of VBR video, for reliable network transport with random linear network coding (RLNC). Conventional RLNC encodes and decodes at the granularity of complete equal size packets. Thus, existing padding reduction approaches have sought to package the variable size packets into fixed size packets with bundling or chaining and fragmentation approaches. We have introduced a new class of RLNC that conducts the encoding and decoding at the finer granularity of macro-symbols (MSs), whereby multiple MSs constitute a packet.

We have conducted an extensive performance evaluation of padding reduction approaches with long traces of full motion VBR video. We found that for small RLNC generation sizes that are required for low-latency video transmission, the MS RLNC approach with deterministic shifting reduces the padding overhead the most. For moderate to large RLNC generation sizes that incur moderate to large network transport delays, the chaining and fragmentation approach and the bundling approach of forming fixed size packets from variable size packets and MS RLNC with deterministic shifting effectively reduce the padding overhead.

There are several interesting directions for future work on the padding overhead involved in reliable RLNC media transport over lossy networks. This study considered generation based RLNC (also referred to as block based RLNC) which codes N packets at a time and thus effectively advances the encoding window in steps of N packets. Future padding overhead research should investigate online sliding window RLNC approaches that advance the RLNC encoding window at a granularity of individual packets [79]–[82]. This finer-grained advance of the encoding window generally helps to reduce network latencies. Another future research direction is to examine the padding reduction in the context of the RLNC inter-coding of multiple media streams. The present study has focused on the RLNC intra-coding of a single media stream. The RLNC inter-coding of multiple media streams could improve the overall multimedia experience over lossy networks.

Acknowledgment

The authors are grateful to Máté Tömösközi of TU Dresden for assistance with the plots.

References

[1] M. Taghouti, D. E. Lucani, M. V. Pedersen, and A. Bouallegue, "On the impact of zero-padding in network coding efficiency with Internet traffic and video traces," in *Proc. IEEE Eur. Wireless Conf. (EW)*, Oulu, Finland, May 2016, pp. 72–77.

[2] M. Taghouti, D. E. Lucani, M. V. Pedersen, and A. Bouallegue, "Random linear network coding for streams with unequally sized packets: Overhead reduction without zero-padded schemes," in *Proc. Int. Conf. Telecommun. (ICT)*, Thessaloniki, Greece, May 2016, pp. 1–6.

[3] M. Taghouti, D. E. Lucani, F. H. P. Fitzek, and A. Bouallegue, "Random linear network coding schemes for reduced zero-padding overhead: Complexity and overhead analysis," in *Proc. VDE Eur. Wireless Conf.*, 2017, pp. 1–7.

[4] G. Araniti, P. Scopelliti, G.-M. Muntean, and A. Iera, "A hybrid unicast-multicast network selection for video deliveries in dense heterogeneous network environments," *IEEE Trans. Broadcast.*, to be published.

[5] L. Christodoulou, O. Abdul-Hameed, and A. M. Kondoz, "Toward an LTE hybrid unicast broadcast content delivery framework," *IEEE Trans. Broadcast.*, vol. 63, no. 4, pp. 656–672, Dec. 2017.

[6] A. Martin *et al.*, "Network resource allocation system for QoE-aware delivery of media services in 5G networks," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 561–574, Jun. 2018.

[7] J. Nightingale, P. Salva-Garcia, J. M. A. Calero, and Q. Wang, "5G-QoE: QoE modelling for ultra-HD video streaming in 5G networks," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 621–634, Jun. 2018.

[8] L. Zou, R. Trestian, and G. M. Muntean, "E3DOAS: Balancing QoE and energy-saving for multi-device adaptation in future mobile wireless video delivery," *IEEE Trans. Broadcast.*, vol. 64, no. 1, pp. 26–40, Mar. 2018.

[9] C. Diaz, J. Cabrera, F. Jaureguizar, and N. García, "Application-layer FEC scheme configuration optimization via hybrid simulated annealing," *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 479–493, Sep. 2017.

[10] M. A. Khan, A. A. Moinuddin, E. Khan, and M. Ghanbari, "Optimized cross-layered unequal error protection for SPIHT coded wireless video transmission," *IEEE Trans. Broadcast.*, vol. 62, no. 4, pp. 876–889, Dec. 2016.

[11] Y.-C. Liu, C.-F. Chang, S.-K. Lee, and M.-C. Lin, "Deliberate bit flipping with error-correction for PAPR reduction," *IEEE Trans. Broadcast.*, vol. 63, no. 1, pp. 123–133, Mar. 2017.

[12] N. Thomos, E. Kurdoglu, P. Frossard, and M. Van der Schaar, "Adaptive prioritized random linear coding and scheduling for layered data delivery from multiple servers," *IEEE Trans. Multimedia*, vol. 17, no. 6, pp. 893–906, Jun. 2015.

[13] P. U. Tournoux, E. Lochin, J. Lacan, A. Bouabdallah, and V. Roca, "On-the-fly erasure coding for real-time video applications," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 797–812, Aug. 2011.

[14] L. Zhang, Y. Li, P. Cheng, and M. Li, "A layer-mixed FEC scheme for scalable media transmission over mobile TV services," *IEEE Trans. Broadcast.*, vol. 63, no. 2, pp. 309–320, Jun. 2017.

[15] G. Joshi, Y. Liu, and E. Soljanin, "On the delay-storage trade-off in content download from coded distributed storage systems," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 989–997, May 2014.

[16] N. Kumar, S. Zeadally, and J. J. P. C. Rodrigues, "QoS-aware hierarchical Web caching scheme for online video streaming applications in Internet-based vehicular ad hoc networks," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7892–7900, Dec. 2015.

[17] T. H. Luan, L. X. Cai, J. Chen, X. S. Shen, and F. Bai, "Engineering a distributed infrastructure for large-scale cost-effective content dissemination over urban vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 3, pp. 1419–1435, Mar. 2014.

[18] J. Qiao, Y. He, and X. S. Shen, "Proactive caching for mobile video streaming in millimeter wave 5G networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 7187–7198, Oct. 2016.

[19] M. Sipos, J. Gahm, N. Venkat, and D. Oran, "Network-aware feasible repairs for erasure-coded storage," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1404–1417, Jun. 2018.

[20] L. Wang, H. Wu, and Z. Han, "Wireless distributed storage in socially enabled D2D communications," *IEEE Access*, vol. 4, pp. 1971–1984, 2016.

[21] M. Esmaeilzadeh, P. Sadeghi, and N. Aboutorab, "Random linear network coding for wireless layered video broadcast: General design methods for adaptive feedback-free transmission," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 790–805, Feb. 2017.

[22] T. Ho *et al.*, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[23] S.-C. Lin and K.-C. Chen, "Statistical QoS control of network coded multipath routing in large cognitive machine-to-machine networks," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 619–627, Aug. 2016.

[24] C. Xu, P. Wang, C. Xiong, X. Wei, and G.-M. Muntean, "Pipeline network coding-based multipath data transfer in heterogeneous wireless networks," *IEEE Trans. Broadcast.*, vol. 63, no. 2, pp. 376–390, Jun. 2017.

[25] X. Xu, Y. Zeng, Y. L. Guan, and L. Yuan, "Expanding-window BATS code for scalable video multicasting over erasure networks," *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 271–281, Feb. 2018.

[26] U. Lee *et al.*, "Efficient peer-to-peer file sharing using network coding in MANET," *J. Commun. Netw.*, vol. 10, no. 4, pp. 422–429, 2008.

[27] B. Li and D. Niu, "Random network coding in peer-to-peer networks: From theory to practice," *Proc. IEEE*, vol. 99, no. 3, pp. 513–523, Mar. 2011.

[28] E. Magli, M. Wang, P. Frossard, and A. Markopoulou, "Network coding meets multimedia: A review," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1195–1212, Aug. 2013.

[29] D. Niu and B. Li, "Analyzing the resilience-complexity tradeoff of network coding in dynamic P2P networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 11, pp. 1842–1850, Nov. 2011.

[30] B.-W. Chen, W. Ji, F. Jiang, and S. Rho, "QoE-enabled big video streaming for large-scale heterogeneous clients and networks in smart cities," *IEEE Access*, vol. 4, pp. 97–107, 2016.

[31] M. S. Karim, M. Esmaeilzadeh, and P. Sadeghi, "On reducing intercept probability for unsubscribed video layers using network coding," *IEEE Commun. Lett.*, vol. 21, no. 6, pp. 1385–1388, Jun. 2017.

[32] D. E. Lucani *et al.*, "Fulcrum: Flexible network coding for heterogeneous devices," *IEEE Access*, vol. 6, pp. 77890–77910, 2018.

[33] K. Matsuzono, H. Asaeda, and T. Turletti, "Low latency low loss streaming using in-network coding and caching," in *Proc. IEEE Infocom*, Atlanta, GA, USA, 2017, pp. 1–9.

[34] E. Skevakis and I. Lambadaris, "Optimal control for network coding broadcast," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[35] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR video: Tradeoffs and potentials," *Proc. IEEE*, vol. 86, no. 5, pp. 952–973, May 1998.

[36] S. Tanwir and H. Perros, "A survey of VBR video traffic models," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1778–1802, 4th Quart., 2013.

[37] A. Bentaleb, A. C. Begen, and R. Zimmermann, "QoE-aware bandwidth broker for HTTP adaptive streaming flows in an SDN-enabled HFC network," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 575–598, Jun. 2018.

[38] F. H. P. Fitzek and M. Reisslein, "A prefetching protocol for continuous media streaming in wireless environments," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 10, pp. 2015–2028, Oct. 2001.

[39] H. Lee and S. Sull, "A VBR video encoding for locally consistent picture quality with small buffering delay under limited bandwidth," *IEEE Trans. Broadcast.*, vol. 58, no. 1, pp. 47–56, Mar. 2012.

[40] M. Reisslein and K. W. Ross, "High-performance prefetching protocols for VBR prerecorded video," *IEEE Netw.*, vol. 12, no. 6, pp. 46–55, Nov./Dec. 1998.

[41] L. Yu, T. Tillo, and J. Xiao, "QoE-driven dynamic adaptive video streaming strategy with future information," *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 523–534, Sep. 2017.

[42] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: An instant primer," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 63–68, Jan. 2006.

[43] J. Hansen, J. Krigslund, D. E. Lucani, and F. H. P. Fitzek, "Sub-transport layer coding: A simple network coding shim for IP traffic," in *Proc. IEEE Veh. Techn. Conf. (VTC Fall)*, 2014, pp. 1–5.

[44] P. T. Compta, F. H. P. Fitzek, and D. E. Lucani, "Network coding is the 5G key enabling technology: Effects and strategies to manage heterogeneous packet lengths," *Trans. Emerg. Telecommun. Technol.*, vol. 26, no. 1, pp. 46–55, Jan. 2015.

[45] M. Taghouti, D. E. Lucani, and F. H. P. Fitzek, "Random shift and XOR of unequal-sized packets (RaSOR) to shave off transmission overhead," in *Proc. IEEE Conf. Inform. Sci. Syst. (CISS)*, 2017, pp. 1–6.

[46] N. Aboutorab, P. Sadeghi, and S. Sorour, "Enabling a tradeoff between completion time and decoding delay in instantly decodable network coded systems," *IEEE Trans. Commun.*, vol. 62, no. 4, pp. 1296–1309, Apr. 2014.

[47] S. Sorour and S. Valaee, "Completion delay minimization for instantly decodable network codes," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1553–1567, Oct. 2015.

[48] J. Qureshi, C. H. Foh, and J. Cai, "Online XOR packet coding: Efficient single-hop wireless multicasting with low decoding delay," *Comput. Commun.*, vol. 39, pp. 65–77, Feb. 2014.

[49] M. Yu, N. Aboutorab, and P. Sadeghi, "From instantly decodable to random linear network coded broadcast," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 3943–3955, Nov. 2014.

[50] A. Douik, S. Sorour, T. Y. Al-Naffouri, and M.-S. Alouini, "Instantly decodable network coding: From centralized to device-to-device communications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1201–1224, 2nd Quart., 2017.

[51] P. Seeling and M. Reisslein, "Video transport evaluation with H.264 video traces," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1142–1165, 4th Quart., 2012.

[52] M. Rerabek and T. Ebrahimi, "Comparison of compression efficiency between HEVC/H.265 and VP9 based on subjective assessments," in *SPIE Appl. Dig. Image Process.*, vol. 9217, 2014, pp. 1–13.

[53] P. Seeling and M. Reisslein, "Video traffic characteristics of modern encoding standards: H.264/AVC with SVC and MVC extensions and H.265/HEVC," *Sci. World J.*, vol. 2014, pp. 1–16, Feb. 2014.

[54] L. C. Gonçalves, P. Sebastião, N. Souto, and A. Correia, "On the impact of user segmentation and behaviour analysis over traffic generation in beyond 4G networks," *Trans. Emerg. Telecommun. Techn.*, vol. 28, no. 1, pp. 1–16, Jan. 2017.

[55] G. V. der Auwera, P. T. David, and M. Reisslein, "Traffic and quality characterization of single-layer video streams encoded with the H.264/MPEG-4 advanced video coding standard and scalable video coding extension," *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 698–718, Sep. 2008.

[56] G. V. der Auwera and M. Reisslein, "Implications of smoothing on statistical multiplexing of H.264/AVC and SVC video streams," *IEEE Trans. Broadcast.*, vol. 55, no. 3, pp. 541–558, Sep. 2009.

[57] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton Conf. Commun. Control Comput.*, vol. 41, 2003, pp. 40–49.

[58] P. A. Chou and Y. Wu, "Network coding for the Internet and wireless networks," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 77–85, Sep. 2007.

[59] J. Heide, M. V. Pedersen, F. H. P. Fitzek, and M. Médard, "On code parameters and coding vector representation for practical RLNC," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–5.

[60] Y. Li, E. Soljanin, and P. Spasojevic, "Effects of the generation size and overlap on throughput and complexity in randomized linear network coding," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1111–1123, Feb. 2011.

[61] S. Pandi *et al.*, "PACE: Redundancy engineering in RLNC for low-latency communication," *IEEE Access*, vol. 5, pp. 20477–20493, 2017.

[62] I. Chatzigeorgiou and A. Tassi, "Decoding delay performance of random linear network coding for broadcast," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7050–7060, Aug. 2017.

[63] A. Garcia-Saavedra, M. Karzand, and D. J. Leith, "Low delay random linear coding and scheduling over multiple interfaces," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3100–3114, Nov. 2017.

[64] M. Karzand, D. J. Leith, J. Cloud, and M. Médard, "Design of FEC for low delay in 5G," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 8, pp. 1783–1793, Aug. 2017.

[65] J. Cloud and M. Médard, "Network coding over SATCOM: Lessons learned," in *Proc. Int. Conf. Wireless Satellite Syst.*, 2015, pp. 272–285.

[66] D. Gligoroski, K. Kralevska, and H. Øverby, "Minimal header overhead for random linear network coding," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, 2015, pp. 680–685.

[67] N. Thomos and P. Frossard, "Toward one symbol network coding vectors," *IEEE Commun. Lett.*, vol. 16, no. 11, pp. 1860–1863, Nov. 2012.

[68] V. Bioglio, M. Grangetto, R. Gaeta, and M. Sereno, "On the fly Gaussian elimination for LT codes," *IEEE Commun. Lett.*, vol. 13, no. 12, pp. 953–955, Dec. 2009.

[69] P. Garrido, D. E. Lucani, and R. Agüero, "Markov chain model for the decoding probability of sparse network coding," *IEEE Trans. Commun.*, vol. 65, no. 4, pp. 1675–1685, Apr. 2017.

[70] J. Heide, M. V. Pedersen, and F. H. P. Fitzek, "Decoding algorithms for random linear network codes," in *Proc. Int. Conf. Res. Netw.*, 2011, pp. 129–136.

[71] M. V. Pedersen, J. Heide, and F. H. P. Fitzek, "Kodo: An open and research oriented network coding library," in *Proc. Netw. Workshops*, vol. 6827, 2011, pp. 145–152.

[72] O. Trullols-Cruces, J. M. Barcelo-Ordinas, and M. Fiore, "Exact decoding probability under random linear network coding," *IEEE Commun. Lett.*, vol. 15, no. 1, pp. 67–69, Jan. 2011.

[73] X. Zhao, "Notes on 'exact decoding probability under random linear network coding,"' *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 720–721, May 2012.

[74] J. Acevedo *et al.*, "Hardware acceleration for RLNC: A case study based on the xtensa processor with the tensilica instruction-set extension," *Electronics*, vol. 7, no. 9, pp. 180.1–180.22, 2018.

[75] H. Shin and J.-S. Park, "Optimizing random network coding for multimedia content distribution over smartphones," *Multimedia Tools Appl.*, vol. 76, pp. 19379–19395, Oct. 2017.

[76] S. Wunderlich, J. Cabrera, F. H. P. Fitzek, and M. Reisslein, "Network coding in heterogeneous multicore IoT nodes with DAG scheduling of parallel matrix block operations," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 917–933, Aug. 2017.

[77] D. S. Johnson, "Near-optimal bin packing algorithms," Ph.D. dissertation, Dept. Math., Massachusetts Inst. Technol., Cambridge, MA, USA, 1973.

[78] D. E. Lucani, M. Medard, and M. Stojanovic, "Random linear network coding for time-division duplexing: Field size considerations," in *Proc. IEEE Glob. Telecomm. Conf. (GLOBECOM)*, 2009, pp. 1–6.

[79] F. Gabriel, S. Wunderlich, S. Pandi, F. H. P. Fitzek, and M. Reisslein, "Caterpillar RLNC with feedback (CRLNC-FB): Reducing delay in selective repeat ARQ through coding," *IEEE Access*, vol. 6, pp. 44787–44802, 2018.

[80] P. Garrido, D. Leith, and R. Aguero, "Joint scheduling and coding over lossy paths with delayed feedback," *Preprint arXiv:1804.04921*, 2018.

[81] J. K. Sundararajan, D. Shah, M. Medard, and P. Sadeghi, "Feedback-based online network coding," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6628–6649, Oct. 2017.

[82] S. Wunderlich, F. Gabriel, S. Pandi, F. H. P. Fitzek, and M. Reisslein, "Caterpillar RLNC (CRLNC): A practical finite sliding window RLNC approach," *IEEE Access*, vol. 5, pp. 20183–20197, 2017.

Maroua Taghouti received the engineering degree in telecommunications and the master's degree in communications systems from the National Engineering School of Tunis, Tunisia, in 2012. She is currently pursuing the Ph.D. degree with Tunisia Polytechnic School, Carthage, Tunisia, and working simultaneously as a Research Fellow with the Deutsche Telekom Chair of Communications Networks, Technische Universität Dresden, Dresden, Germany. Prior to the master's degree, she studied Mathematics and Physics for two years at the Institute of Preparatory Studies for Engineering of Tunis. Her research interests focus on coding theory, mainly network coding, and compressed sensing for communications. She has served as a Reviewer for different conferences and journals, including IEEE COMMUNICATIONS LETTERS and IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.

Daniel E. Lucani (SM'16) received the B.S. degree (*summa cum laude*) and the M.S. degree (Hons.) in electronics engineering from Universidad Simon Bolvar, Caracas, Venezuela, in 2005 and 2006, respectively, and the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology in 2010. He has been an Associate Professor with the Department of Engineering, Aarhus University since 2017 as well as the CEO and the Lead Scientist of the start-up company Chocolate Cloud ApS since 2014. He was an Associate Professor with Aalborg University from 2012 to 2017 and the University of Porto from 2010 to 2012. He has published over 140 scientific papers in international journals and top-ranked international conferences as well as eight patents and patent applications. His research interests include communications and networks, network coding, information theory, coding theory, distributed storage and computation, and their applications to cloud computing technologies necessary to enable IoT, BigData, and 5G applications and services. He was a recipient of the IEEE ComSoc Outstanding Young Researcher Award for the EMEA region in 2015 and the Danish Free Research Foundation's Sapere Aude Starting Grant. He was the General Co-Chair of the 2014 International Symposium on Network Coding. He is an Associate Editor of the *EURASIP Journal on Wireless Communications and Networking*.

Juan A. Cabrera received the B.Sc. degree in electronics engineering from Simon Bolivar University, Venezuela, in 2013 and the M.Sc. degree in wireless communication systems from Aalborg University, Denmark, in 2015. He is currently pursuing the Ph.D. degree with the Deutsche Telekom Chair of Communication Networks, Technical University Dresden, Germany. He is especially interested in the research areas of network coding, fog computing, distributed storage systems, and mobile edge cloud solutions.

Martin Reisslein (S'96–M'98–SM'03–F'14) received the Ph.D. degree in systems engineering from the University of Pennsylvania in 1998. He is a Professor with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON EDUCATION, IEEE ACCESS, as well as *Computer Networks*. He is the Associate Editor-in-Chief of the IEEE COMMUNICATIONS SURVEYS & TUTORIALS, a Co-Editor-in-Chief of *Optical Switching and Networking*, and chairs the Steering Committee of the IEEE TRANSACTIONS ON MULTIMEDIA.

Morten Videbæk Pedersen was born in Sall, Denmark, in 1982. He received the M.Sc. degree (*cum laude*) and the Ph.D. degree from the Elite Program in wireless communication from Aalborg University, Denmark, in 2009 and 2012, respectively. He is currently a CTO of Steinwurf APS, where his main focus is on low complexity network coding algorithms and cooperative networking protocols. His main research interests are low-complexity network coding algorithms, cooperative networking protocols, and low-level programming. He co-founded the start-up company Steinwurf ApS in 2011, creating industrial grade high performance content distribution systems. Since 2006, he has been a member of the Mobile Devices Research Group, Aalborg University, where he has been a member of the Network Coding Focus Group since 2013. He has created and contributed to several scientific software libraries, latest by creating Kodo a flexible high-performance network coding library, which is currently used in industry and by researchers at several international universities. He was a recipient of the Forum Nokia Developer Champion Award in 2010.

Frank H. P. Fitzek received the Diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology – Rheinisch-Westfälische Technische Hochschule, Aachen, Germany, in 1997, the Ph.D. (Dr.-Ing.) degree in electrical engineering from Technical University Berlin, Germany, in 2002, and the Honorary degree (Doctor Honoris Causa) from the Budapest University of Technology and Economy in 2015. He is a Professor and the Head of the Deutsche Telekom Chair of Communication Networks, Technical University Dresden, Germany, coordinating the 5G Lab Germany. He became an Adjunct Professor with the University of Ferrara, Italy, in 2002. In 2003, he joined Aalborg University as an Associate Professor and later became a Professor. He co-founded several start-up companies starting with Acticom GmbH, Berlin, in 1999. His current research interests are in the areas of wireless and mobile 5G communication networks, mobile phone programming, network coding, cross layer as well as energy efficient protocol design and cooperative networking. He was a recipient of the NOKIA Champion Award several times in a row from 2007 to 2011, the Nokia Achievement Award for his work on cooperative networks in 2008, the SAPERE AUDE research grant from the Danish Government in 2011, and the Vodafone Innovation Prize in 2012.