PACE: Redundancy Engineering in RLNC for Low-Latency Communication

Sreekrishna Pandi, Frank Gabriel, Juan A. Cabrera, Simon Wunderlich, Martin Reisslein, and Frank H. P. Fitzek

***Abstract*—Random linear network coding (RLNC) is attractive for data transfer as well as data storage and retrieval in complex and unreliable settings. The existing systematic RLNC approach first sends all source symbols in a generation without encoding followed by the coded redundant packets at the tail of the generation. This systematic tail RLNC achieves low delay when packet drops are rare; however, recovery of any dropped source symbol requires to wait for the coded packets at the end of the generation. We propose and evaluate a novel PACE RLNC approach that paces the transmissions of coded redundant packets throughout the generation of source symbols. The paced coded packets enable the recovery of dropped source symbols without waiting for the tail end of the generation. More specifically, we propose PACE-Uniform, which uniformly intersperses individual coded packets throughout the generation, and PACE-Burst, which intersperses bursts of code packets. Our extensive simulation evaluations indicate that PACE-Uniform significantly reduces the mean source symbol delay compared to tail RLNC, while achieving nearly the same loss probability. We also demonstrate the PACE-Burst generalizes the concept of pacing the redundant packet transmissions and can be flexibly tuned between PACE-Uniform and the conventional tail RLNC by controlling the number of coded packets in a burst.**

***Index Terms*—Delay; Generation based network coding; Loss probability; Random Linear Network Coding (RLNC); Scheduling.**

I. Introduction

Random Linear Network Coding (RLNC) facilitates the reliable data transfer as well as data storage and retrieval in a wide range of unreliable systems, including wireless networks [1]–[9], vehicular ad hoc networks [10], multicast distribution [11]–[13], peer-to-peer distribution [14], and storage servers [15]–[18]. The practical generation based RLNC [19]–[21] groups $g,\ g > 1$, source symbols of the original data into a so-called generation. The g source symbols in a generation are then jointly encoded and transmitted. In particular, the full vector approach of generation based RLNC [20], [22] transmits all source symbols in a generation in the form of coded packets that are linear combinations of all source symbols in the generation. This full vector approach introduces long delays since all coded packets of a generation need to be received before decoding can commence. In contrast, the systematic approach [23] of generation based RLNC [24]–[27] transmits all g source symbols in uncoded form, followed by ϵ coded forward error correction (FEC) packets [28] at the tail end of the generation. If there are no packet drops or erasures on the transmission channel, then the received uncoded source symbols can be immediately delivered to the receiving application. However, if the transmission channel drops an uncoded source symbol, then the receiver needs to wait for the coded packets at the end of the generation to attempt recovery. Thus, for an application requiring in-order delivery, the dropped source symbol, and the subsequent symbols incur long delays by waiting for the coded packets at the tail end of the generation.

We propose a novel approach to generation based RLNC that intersperses the ϵ coded FEC packets among the g uncoded source symbols, i.e., our approach paces the transmission of the coded packets and is therefore referred to as PACE. More specifically, conventional systematic tail RLNC linearly combines *all* g source symbols in a generation to form the coded FEC packets and sends the ϵ coded packets after the g source symbols. Our PACE-Uniform approach uniformly partitions the g uncoded source symbols in a generation into ϵ sub-generations. Each sub-generation consists of g/ϵ (approximately, rounding issues are addressed in the exact specification in Section IV-B) uncoded source symbols, followed by one coded FEC packet. The coded packet is formed by linearly combining the source symbols in the present sub-generation as well as the preceding sub-generations in the considered generation. We also specify a PACE-Burst approach that generalizes the PACE concept by grouping $B,\ 1 \leq B \leq \epsilon$, coded packets. Our extensive simulation evaluations demonstrate that our PACE approach substantially reduces the mean delay for the source symbol delivery through an RLNC system while achieving approximately the same loss probability (after recovery utilizing the coded packets) as conventional systematic tail RLNC. The proposed PACE RLNC appears therefore well-suited for RLNC systems supporting delay-sensitive applications.

This article is organized as follows. Section II reviews related work on RLNC for delay-sensitive applications. Section III gives a brief tutorial overview of RLNC and introduces the performance evaluation set-up for the considered RLNC system. Section IV introduces PACE-Uniform and presents the detailed evaluation of the delay and loss probability performance of PACE-Uniform in comparison to the conventional systematic tail RLNC. Section VI introduces PACE-Burst and presents the performance evaluation of PACE-Burst for the range of number B of coded packets in a burst. Section VII summarizes this study.

Supported by 5G Lab Germany activities on several projects, including Bosch, Ericsson, and Deutsche Telekom projects.

Please direct correspondence to M. Reisslein.

S. Pandi, F. Gabriel, J.A. Cabrera, S. Wunderlich, and F.H.P. Fitzek are with TU Dresden, Deutsche Telekom Chair, 5G Lab Germany, Dresden, Germany (e-mail: {firstname.lastname}@tu-dresden.de)

M. Reisslein is with Elect. Comp., Energy Eng., Arizona State Univ., Tempe, AZ 85287-5706, USA (e-mail: reisslein@asu.edu), phone 480-965-8593.

II. Related Work

Low-delay communication in network coding systems has been examined from several angles. Several studies have demonstrated the general benefits of network coding for reliable low-delay communication [29]–[33]. One research direction has developed special types of network codes for low-delay communication [34]–[37], such as instantly decodable network codes [38]–[42], online codes [43]–[47], and streaming codes [48]–[50], as well as a variety of adaptive coding schemes [51]–[57]. In contrast, we consider the conventional generation based form of RLNC.

The intercoding of two unicast sessions has been examined in [58], [59], while the joint coding of uplink streams from multiple user equipment nodes to the central enhanced Node B in an LTE system has been examined in [60]. The impact of feedback on the throughput-delay performance has been examined in [61]–[63]. RLNC delay characteristics for multicast and broadcast have been examined in [64]–[66]. We consider a single unicast session without feedback. A few studies have examined network coding in specific networking contexts, e.g., in the context of delay-tolerant networks in [67], for video delivery in [68], [69], for industrial networks in [70], [71], and for sensor networks in [72]. We consider a general point-to-point network model that abstracts the end-to-end path as a single link that drops packets with a prescribed probability E.

The throughput-delay characteristics of generation based full vector RLNC have been examined in detail in [73]–[79]. Dynamic adaption of the generation size of full vector RLNC has been examined in [21], [80]–[82]. Multi-generation mixing [83]–[87] jointly encodes the source symbols from multiple successive generations within a so-called mixing set in a full vector coding RLNC manner. Evaluations have demonstrated that full vector multi-generation mixing increases the throughput and decreases the loss probability; however, the impact on the delay has not been studied in detail.

Generation based systematic RLNC has also been examined from a variety of angles that are complementary to our study. Systematic RLNC in a relay setting [88] with re-coding in intermediate nodes has been studied in [89]. The impact of feedback for systematic RLNC transmissions has been examined in [90]. Systematic RLNC in the specific context of wireless transmissions over the WiMAX link layer with a range of physical and medium access control layer setting has been evaluated in [91], while the LTE link layer has been examined in [92]–[94], and a specific two-hop setting has been studied in [95]. Multimedia broadcast with systematic RLNC has been examined in [96]. The prior studies on generation-based systematic RLNC transmitted the coded packets at the tail end of the generation, incurring long wait times for the recovery of dropped source symbols. In contrast, we pace the coded packet transmissions among the uncoded source symbols to facilitate quick recovery of dropped source symbols.

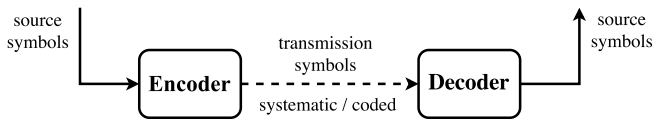


Fig. 1: System under study: Network layer datagrams enter the link layer as source symbols. The encoder applies different RLNC coding approaches and sends systematic (uncoded) source symbols and coded transmission symbols (coded packets) over the channel, which has drop probability E. The link layer delay and loss after the decoder are evaluated.

TABLE I: Summary of main notations

Notation	Definition
G	Generation size = # of source symbols in a generation
C	Coding ratio = Total # of transm. symbols for a generation / G
ϵ	# of FEC packets for a generation, $\epsilon = \lceil (C-1)G \rceil$
τ	Total # of transm. packets for a generation, $\tau = G + \epsilon$

III. Background: Random Linear Network Coding (RLNC)

A. RLNC Overview

RLNC is based on linear finite field arithmetic in the Galois Field GF(2^p) [22]. At the sender, the original data is split into successive *source symbols* (that are also referred to as *source packets*). The goal is to reliably transmit the source symbols over a lossy channel with the aid of RLNC encoding at the source and decoding at the receiver, as illustrated in Fig. 1. Let ψ denote the source symbol length (source symbol size) in units of words, whereby for the typically considered GF(2^8), 1 word = 8 bit = 1 Byte (we set $\psi = 1500$ Byte in our evaluations). A group of G consecutive source symbols forms a generation, i.e., G is the generation size (in units of source symbols). Formally, the generation size G is defined as the maximum number of source symbols that can be linearly combined to form one *transmission symbol*. The original source data of one generation can be represented by a matrix $\mathbf{S}$ with G rows and ψ columns, whereby each row of matrix $\mathbf{S}$ represents one original source symbol.

For encoding, we let C, $C \geq 1$, denote the coding ratio defined as the ratio of the total number of transmission symbols emitted (sent) by the encoder to the total number of source symbols that were fed into the encoder. Note that this coding ratio definition is different from the common definition of the FEC code rate [28], which is the proportion of the data stream that is useful, i.e., non-redundant. Also, note that due to the restriction to integer numbers of sent packets, for a prescribed coding ratio C, we use an actual number of

$$\epsilon = \lceil (C-1)G \rceil \tag{1}$$

redundant FEC packets. We define, for brevity (see Table I for summary of main notations),

$$\tau = G + \epsilon \tag{2}$$

for the actual number of packets sent for a generation, the actual coding ratio is τ/G. A coefficient matrix $\mathbf{C}$ with τ rows and G columns with random coefficients is created for

the encoding. Encoding is then performed by multiplying the coefficient matrix $\mathbf{C}$ with the source symbol matrix $\mathbf{S}$ to obtain the transmission symbol matrix $\mathbf{T}$ with τ rows and ψ columns:

$$\mathbf{T} = \mathbf{CS}. \tag{3}$$

A coded transmission symbol, which is a row of matrix $\mathbf{T}$, jointly with the corresponding coefficient (row) vector of matrix $\mathbf{C}$ forms a *coded packet*.

A receiver that has acquired at least G linearly independent coded packets forms a received symbol matrix $\mathbf{R}$ and a new coefficient matrix $\bar{\mathbf{C}}$. The receiver matrices $\mathbf{R}$ and $\bar{\mathbf{C}}$ differ from the corresponding sender matrices $\mathbf{T}$ and $\mathbf{C}$ in row order and number of rows. To decode and reconstruct the original symbol matrix $\mathbf{S}$, the receiver evaluates $\mathbf{S} = \bar{\mathbf{C}}^{-1}\mathbf{R}$ [97], [98].

B. Full Vector (Non-Systematic) RLNC Encoding

In the full vector type of RLNC encoding [20], [22], which is also referred to as non-systematic RLNC encoding, each transmission symbol is obtained by coding, i.e., linearly combining, *all* G source symbols, as illustrated for an example coefficient matrix $\mathbf{C}$ for $G = 9$ generations and coding ratio $C = 1.3$ in Fig. 2a. Full vector encoding does not permit on-the-fly encoding since the encoder needs all G source symbols of a generation before it can begin encoding. With full vector encoding, any G transmission symbols can be decoded to obtain the original G source symbols. In particular, either all G symbols of a generation are decoded or none of them are decoded. Note that with full vector encoding, all source symbols of a given generation have the same delay since all of them are decoded at once, namely when the last transmission symbol of the generation is received by the decoder.

C. Systematic Tail RLNC Encoding

With systematic tail RLNC encoding [24], [26], [27], the source symbols are first sent without coding, i.e., systematically. By placing a $G \times G$ identity matrix at the top of the coefficient matrix $\mathbf{C}$, as illustrated in Figure 2b for $G = 9$, the top G rows of the transmission matrix $\mathbf{T}$ calculated with Eqn. (3) correspond to the G source symbols. After this systematic phase, i.e., after transmitting these G uncoded source symbols, the $\epsilon = \lceil (C-1)G \rceil$ redundant symbols that are meant for FEC are coded and sent. The ϵ coded transmission symbols are generated by linearly combining all G source symbols in a given generation. That is, all G source symbols are included in the coding coefficients as illustrated for the example coding coefficients in the bottom three rows of matrix $\mathbf{C}$ in Figure 2b. All ϵ coded transmission symbols are sent at the tail end of the generation; therefore, we refer to this conventional systematic RLNC as *tail RLNC*.

With systematic encoding, the systematic (uncoded) packets do not incur any delays due to encoding, i.e., every source symbol can be sent out immediately upon arrival and does not need to wait for other source symbols to fill up a generation. Systematic encoding thus allows for on-the-fly encoding.

D. Performance Evaluation Setup

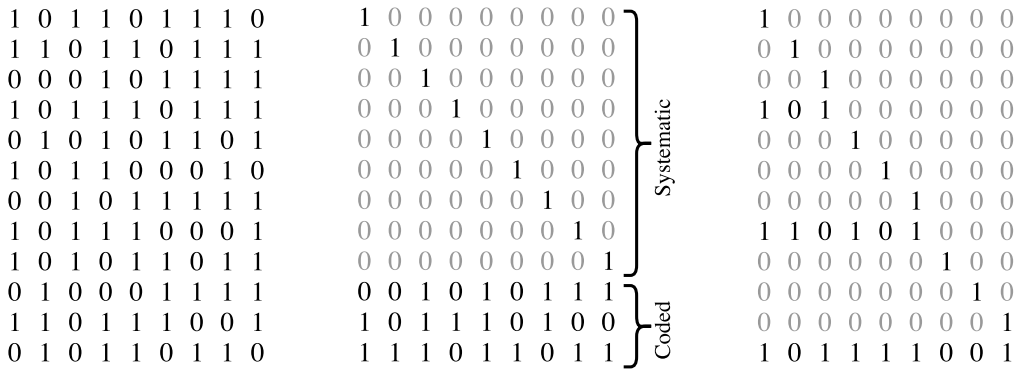
1) Overall Evaluation Methodology: Although RLNC can be employed in a wide variety of contexts, in order to have a concrete context for our evaluation, we consider an RLNC link layer transmission scenario. Other RLNC application contexts can be mapped to the considered link layer context. We conduct an evaluation of the link layer (L2) performance with the different types of RLNC. We evaluate the performance of the different RLNC communication types with discrete event simulations. In the simulations, the network coding functions are executed through the Kodo library [99] in GF(2^8). For each combination of considered RLNC type and parameter settings, we conduct multiple independent simulation replications of the encoding, transmission, and decoding of one generation, i.e., a set of G consecutive source symbols. We average the performance metric samples of the independent replications to obtain the sample means of the performance metrics, which are shown in the result plots in this paper. We also evaluated the 95 % confidence intervals and verified that they are smaller than 5 % of the corresponding sample means. The confidence intervals are not shown in the result plots to avoid visual clutter.

We consider a time-slotted link and physical layer communication system that transmits one coded packet or one uncoded source symbol in one time slot from the output of the encoder to the input of the decoder. The encoding and decoding computation times are assumed to be negligible relative to the packet transmission time slot; the impact of the proposed PACE RLNC on the computation times is examined in Section V-D. Note that with a coding ratio C, the transmission of G source symbols requires τ time slots, since ϵ redundant (FEC) transmission symbols are generated for G source symbols by the encoder. We note that for a given coding ratio C (and ignoring channel losses), all compared types of RLNC coding achieve the same long-run link layer throughput of G source symbols per τ time slots. Note that τ increases with the coding ratio C, see Eqn. (2); thus the throughput decreases with increasing C (or equivalently, the physical layer transmission bitrate required for achieving a prescribed throughput increases with increasing C).

We consider a lossy physical layer channel between encoder and decoder that drops (loses, erases) a given transmitted packet independently with a prescribed probability E, $0 \leq E < 1$. We do not consider channel propagation delays or packet reordering in the physical layer channel.

2) Delay D: We define the link layer source symbol delay (latency) D as the time duration between the starting time instant and ending time instant of the link layer source symbol transmission that are defined as follows. At the starting time instant, the source symbol enters the link layer, i.e., the source symbol transitions from the network layer down into the link layer according to a traffic model, as defined in Section III-D3.

At the ending time instant, the decoded source symbol is emitted by the decoder on the receiver side in the correct in-order sequence up to the network layer. Note that due to the in-order sequence requirement, a given source symbol can only be delivered if *all* preceding source symbols have been



(a) Full Vector Encoder (b) Systematic Tail Encoder (c) PACE-Uniform Encoder

Fig. 2: Illustrative coding coefficient matrices $\mathbf{C}$ in GF(2) for different types of RLNC encoding with $G = 9$ source symbols in a generation and coding ratio $C = 1.3$. The $\tau = 12$ transmission symbols are obtained by multiplying $\mathbf{C}$ with the source symbol matrix $\mathbf{S}$, see Eqn. (3). With full vector encoding, all G source symbols are combined to form the transmission symbols. The systematic tail encoder transmits $G = 9$ uncoded (systematic) source symbols, followed by $\epsilon = 3$ coded symbols. The PACE-Uniform encoder distributes the coded symbols uniformly among the G uncoded source symbols, so that a coded packet is transmitted after every three uncoded source symbols.

decoded and delivered, or the decoder has deterministically determined that the preceding symbols that have been dropped by the channel can never be recovered, i.e., are lost. The delay is only defined for source symbols that are delivered to the network layer at the receiver; lost (unrecovered) source symbols are not considered in the delay measurements.

3) Traffic Models:

a) Datagram Pool: We define the *datagram pool* traffic model as follows. We suppose that the next higher layer, e.g., network layer (L3), has a very large (unlimited) number of packets (e.g., network layer datagrams) to send. The encoder at the link layer (instantaneously) pulls a source symbol from the unlimited pool of network layer datagrams (i) when the source symbol is needed for encoding (if the source symbol is encoded into a transmission symbol) followed instantaneously by transmission over the channel, or (ii) when the source symbol transmission commences (if the symbol is transmitted as an uncoded source symbol). Note that the starting time instants and ending time instants for all G source symbols of a given generation fall within the τ time slots during which the considered RLNC link layer is working on transmitting the considered generation. The delay differences between different RLNC approaches arise due to different starting and ending time dynamics within the span of the τ generation transmission time slots. For instance, a full vector RLNC encoder pulls all G source symbols instantaneously at the starting instant of the generation transmission time slots, instantaneously encodes the G source symbols into τ transmission symbols and then transmits the coded packets at a constant rate of one coded packet per time slot into the channel. The decoder accumulates all received τ coded packets and then instantaneously decodes them and delivers them to the network layer. In contrast, systematic RLNC encoding pulls the G source symbols at the constant rate of one source symbol per slot during the first G slots of the τ generation transmission slots, and then stops pulling source symbols until the start of the next generation. Unless otherwise noted, the datagram pool model is the default traffic model in this study.

b) Constant Packet Rate: The constant packet rate traffic model feeds source symbols at a constant rate of G symbols per τ slots into the link layer, i.e., one source symbol per τ/G slots. Note that such a constant rate source traffic model requires earlier source symbol starting time instants in order to achieve the same ending time instants as with the datagram pool model. Specifically, for full vector encoding, the source symbols need to arrive over the τ slots preceding the actual τ generation transmission slots, so that all G source symbols are available for encoding at the start of the generation transmission slots. For systematic encoding, the last source symbol needs to arrive by the beginning of the Gth slot within the actual τ generation transmission slots. Thus, the constant packet rate traffic model exacerbates the delay performance differences between the different RLNC approaches. That is, systematic RLNC encoding requires earlier starting time instants (in the constant packet rate traffic model compared to the datagram pool model), but full vector RLNC encoding requires yet earlier starting time instants, while both achieve the same ending time instants as for the datagram pool traffic model.

c) Constant Systematic Packet Rate: A variation of the constant packet rate model injects the source symbols at rate of one symbol per slot during the first G slots of a generation and then stays silent for ϵ slots. We refer to this model as the constant systematic packet rate model as the packets arrive one by one just in time for the uncoded source packet transmissions by the systematic tail RLNC, and then there are no packet arrivals while the systematic RLNC sends the ϵ coded packets.

d) Generation Burst: Another alternative traffic model is a burst traffic model that feeds G source symbols into the link layer (encoder) at the starting instant of every period of τ generation transmission slots. With the burst traffic model, the starting time instants of full vector and systematic encoding

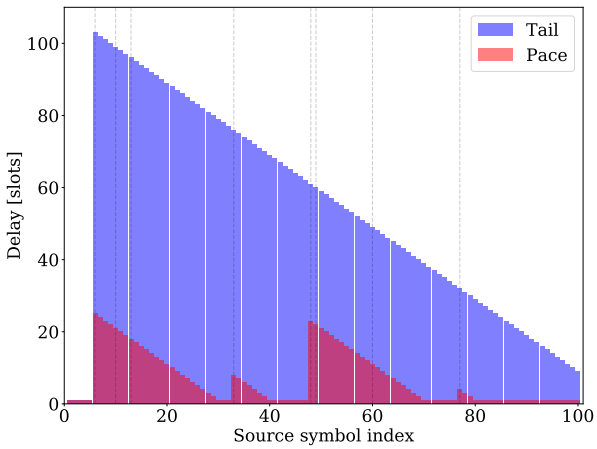


Fig. 3: In-order link layer delay of each symbol in a generation with $G = 100$ symbols, coding ratio $C = 1.1$, and channel error (packet drop) probability $E = 0.1$ for one simulation run: Conventional systematic tail RLNC coding gives high symbol delays after a loss (indicated by a dashed vertical line) since the coded packets are at the end of the generation. The proposed PACE-Uniform RLNC intersperses coded packets uniformly among the uncoded source symbols to permit quick recovery from losses.

are the same (for all source symbols). The ending time instants with the burst traffic model are the same as with the datagram pool traffic model.

4) Packet Loss Probability L: For a given generation of G source symbols that are transmitted by the considered RLNC link layer, we define the packet loss probability L as the ratio of the number of source symbols that cannot be delivered to the network layer at the receiver side to the number G of source symbols in the generation. That is, L corresponds to the proportion of source symbols that are lost by the considered link layer as they cannot be recovered (decoded) by the decoder at the receiver.

E. Shortcomings of Full Vector and Systematic RLNC

Systematic RLNC encoding achieves short delays when there are no or only very rare packet drops on the channel. However, with significant channel packet drops, the delay of systematic RLNC converges to the delay of full vector RLNC. This is because the dropping of a systematic (uncoded) source symbol in systematic RLNC forces the decoder to wait until the dropped symbol can be repaired with a coded transmission symbol (which arrives only after all the systematic source symbols of the generation). Hence, all symbols that follow a dropped symbol are delayed until the coded symbols arrive at the end of the generation transmission slots, causing the delay of systematic RLNC to approach the full vector RLNC delay.

Figure 3 shows the delays of the individual symbols $i = 1, 2, \ldots, G$, for systematic RLNC for an example with $G = 100$ source symbols in a generation and $E = 0.10$ channel packet drop probability (ignore the PACE-Uniform part of the plot for now). In the example in Figure 3, all systematic source symbols up to and including symbol 5 are successfully delivered by the channel and thus can be immediately delivered by the decoder to the network layer, resulting in a delay of $D = 1$ slot. However, source symbol 6 (and seven more source symbols later in the generation marked with vertical dashed lines) are dropped by the channel. The decoder has to buffer the subsequent source symbols 7, 8, …, 100, until eight coded packets arrive in slots 101, 102, …, 108 to enable the recovery of all eight lost packets. Thus, source symbol 100, which was pulled from the network layer at the beginning of slot 100 incurred a link layer delay of nine slots. On the other hand, source symbol 6, which was pulled from the network layer at the beginning of slot 6, incurred a link layer delay of 103 slots, as it had to wait for recovery after the receipt of eight coded packets by the receiver.

IV. Uniform Pacing of Redundancies: PACE-Uniform Specification

A. Basic PACE Concept

In order to prevent the waiting for the coded transmission symbols at the end of the generation in systematic tail RLNC, the coded transmission symbols can be interspersed among the uncoded source symbols. That is, the coded transmission symbols can be transmitted at a prescribed pace interspersed with the uncoded source symbols during the systematic phase, i.e., as *paced redundancies*, instead of sending all the coded transmission symbols at the end of the generation as *tail redundancies*. The paced redundancies allow for the recovery of lost uncoded source symbols as soon as sufficiently many coded transmission symbols have been received. With paced redundancies this recovery can occur during the course of the generation. Thus, the waiting until the end of the generation for the recovery of lost symbols can be avoided. The delay reduction with paced redundancies compared to conventional systematic tail RLNC can be particularly pronounced for large generation sizes G. For large generation sizes G, conventional systematic tail RLNC requires long wait times until coded symbols arrive at end of the generation to recover the losses. Similarly, the delay reduction with paced redundancies is pronounced when losses occur early in a generation.

The paced redundancies can be interspersed (distributed) according to different strategies among the systematic (uncoded) source symbols. We introduce and evaluate a uniform pacing strategy in this section and study a burst strategy in Section VI.

B. PACE-Uniform Specification

PACE-Uniform strives to reduce the delay caused by losses in systematic RLNC by distributing the pacing redundancies uniformly among the systematic (uncoded) source symbols. For instance, with $G = 100$ source symbols in a generation and $C = 1.1$ coding ratio, one coded symbol (pacing redundancy) is sent after every ten systematic symbols.

Formally, PACE-Uniform schedules the coded symbols as follows: Recall that $\epsilon = \lceil (C-1)G \rceil$ denotes the number of redundant FEC packets that are sent for one generation

consisting of G source symbols. We partition the generation into ϵ sub-generations: Sub-generation $s,\ s = 1, 2, \ldots, \epsilon$, consists of σ_s source symbols followed by one coded packet. In particular, we partition a given generation into ϵ sub-generations by dividing the G source symbols equally among the ϵ sub-generations. If G is not an integer multiple of ϵ, then we assign one source symbol of the division remainder $G \bmod \epsilon$ to each of the first $G \bmod \epsilon$ sub-generations. Thus, the number σ_s of source symbols in sub-generation s is

$$\sigma_s = \begin{cases} \lfloor \frac{G}{\epsilon} \rfloor + 1 & s = 1, 2, \ldots, G \bmod \epsilon \\ \lfloor \frac{G}{\epsilon} \rfloor & s = (G \bmod \epsilon) + 1, \ldots, \epsilon. \end{cases} \quad (4)$$

The coded packet for a sub-generation s is formed by coding over, i.e., linearly combining, the source symbols in sub-generation s and its preceding sub-generations $1, 2, \ldots, s-1$ in the current generation. The PACE-Uniform RLNC is illustrated for an example with $G = 9$ source symbols in a generation for the $C = 1.3$ coding ratio, i.e., with $\epsilon = 3$ redundant packets and equivalently with partitioning of the generation into $\epsilon = 3$ sub-generations, in Fig. 2c. Notice that the first coded packet which is obtained through the fourth line in the illustrated coding coefficient matrix $\mathbf{C}$ combines only the first three source symbols (corresponding to the first three lines in the coefficient matrix). The sub-generation structure ensures that there is one coded packet at the very end of the generation that is based on all source symbols in the generation, as illustrated by the bottom line in the coding coefficient matrix $\mathbf{C}$ in Fig. 2c.

Note that pacing the redundancies among the systematic source symbols introduces asymmetries in the protection of the source symbols within the generation. For instance, in Figure 2c, the first three source symbols are protected by three coded packets, whereas the three source symbols between the first and second coded packet are protected by only two coded packets. If one of the first three source symbols is lost, it may be recovered by any of the three following coded packets in the generation, whereas only two coded packets are available for attempting the recovery of a loss among the next three source symbols. This asymmetric protection increases the chances for recovery for the symbols at the beginning of the generation compared to symbols appearing towards the end, as will be studied in detail in Section V-B.

C. PACE-Uniform with Extra Tail Coded Packets

In order to provide additional protection for the source symbols near the end of a generation, we generalize the PACE-Uniform specification of Section IV-B to allocate $T,\ 0 \leq T \leq \epsilon - 1$, coded packets for the tail end of the generation. Only the remaining $\epsilon - T$ coded packets are then considered in the PACE-Uniform scheduling of the coded packets, i.e., ϵ is replaced by $\epsilon - T$ in Eqn. (4). PACE-Uniform always positions *one* of the $\epsilon - T$ coded packets considered for PACE-Uniform scheduling at the tail of the generation; therefore, we refer to the T coded packets, as *extra* tail coded packets, that increase the total number of coded packets at the tail of the generation to $T + 1$. We emphasize that the T extra tail coded packets do *not* increase the coding ratio; rather, the prescribed coding ratio is kept fixed at C. Out of the $\epsilon = \lceil (C-1)G \rceil$ coded packets resulting from the coding ratio C, T coded packets are allocated to the tail and $\epsilon - T$ coded packets are allocated to PACE-Uniform scheduling.

Note that $T = 0$ corresponds to the conventional PACE-Uniform as defined in Section IV-B without any extra coded packets at the tail of the generation, i.e., only one coded packet at the tail of the generation. On the other hand, $T = \epsilon - 1$ corresponds to conventional tail RLNC, since PACE-Uniform is executed with $\epsilon - T = 1$ coded packet, i.e., with one sub-generation containing all source symbols of the generation ($\sigma_1 = G$) followed by one coded packet, to which the $T = \epsilon - 1$ extra coded packets are added.

D. Illustrative Comparison of PACE-Uniform vs. Tail RLNC

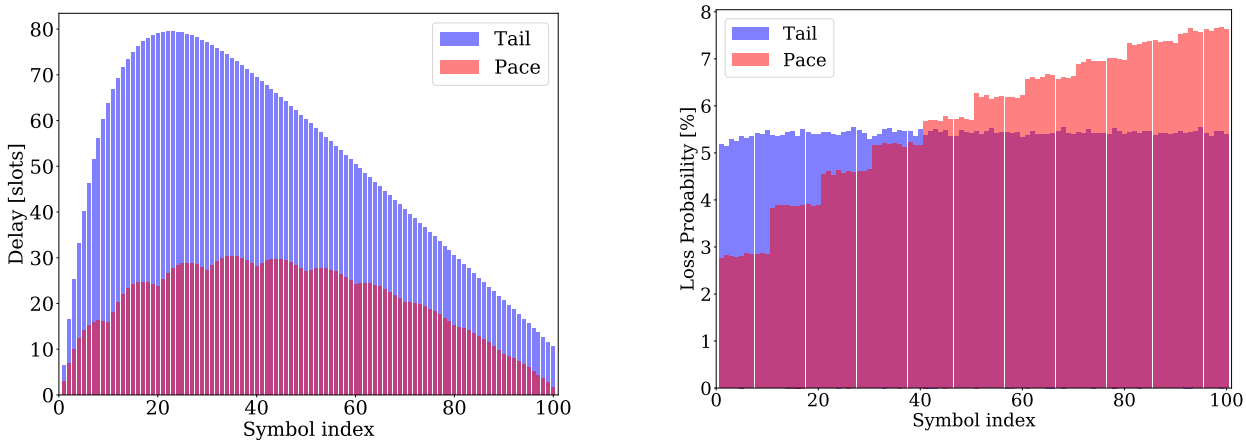
Figure 3 illustrates the in-order link layer delay for the individual source symbols for PACE-Uniform (without extra tail coded packets, i.e., $T = 0$) in comparison with systematic tail RLNC (see Section III-C). With PACE-Uniform, a coded packet is inserted after every ten source symbols; the coded packets are not shown in Figure 3. We observe from Figure 3 that the coded packets that are interspersed with the uncoded source symbols aid in the early recovery of lost symbols. Thus, instead of the one long triangular delay slope for systematic RLNC, we observe multiple short triangular delay spikes in Figure 3. The interspersed (paced) redundancies bring the delay down to one, if enough coded packets have been received to recover from the preceding losses in the generation.

V. PACE-Uniform Evaluation

A. Delay and Loss Probability for Individual Source Symbols

Figure 4a shows the average delay of each symbol obtained from 450,000 simulation replications. For systematic tail RLNC we observe that the delay initially quickly increases with the source symbol index $i,\ i = 1, 2, \ldots, G$, reaching a maximum delay around 80 slots for the source symbols indices i between 20 and 25. Then, the delays drop off towards the end of the generation. This behavior is due to the following two complementary events: First, if source symbol $i,\ i = 1, 2, \ldots, G$, and all preceding symbols $1, 2, \ldots, i-1$ traversed the channel successfully, which occurs with probability $(1-E)^i$, then source symbol i experiences a delay of one slot. Second, if source symbol i or some preceding symbol was dropped on the channel, which occurs with probability $1-(1-E)^i$, then symbol i needs to wait for the coded packets at the end of the generation. In particular, source symbol i has a delay of $G - i + 1$ slots from the beginning of slot i when it was pulled from the network layer to the end of the generation. Moreover, a mean number of EG source symbols is dropped by the channel over the course of the generation and need to be recovered with the coded packets (or declared as deterministically lost). These two complementary events combine to give the expected delay observed for tail RLNC in Figure 4.

For PACE-Uniform, we observe from Figure 4a a ripple shaped delay pattern. The number of ripples corresponds to the number of equidistantly paced coded packets (which are



(a) Delay

(b) Loss Probability

Fig. 4: Averages (over 450,000 simulation replications) of in-order delay [slots] and loss probability [%] of each of the $G = 100$ source symbols in a generation for coding ratio $C = 1.1$ and uniform channel drop probability $E = 0.1$: PACE-Uniform (plotted with $T = 0$ extra tail coded packets) avoids the large delays of systematic tail RLNC.

not plotted) within the generation. The trough of each ripple coincides with the position of a coded packet, indicating that each of the paced coded packets pulls down the average delay by enabling the recovery of some previously lost packets.

For the loss probability we observe from Fig. 4b that tail RLNC achieves uniformly the same loss probability for all source symbols i, $i = 1, 2, \ldots, G$. This is because tail RLNC protects each source symbol i, $i = 1, 2, \ldots, G$, by the same number of ϵ coded packets. On the other hand, we observe for PACE-Uniform from Fig. 4b increasing loss probabilities in form of a "staircase" for increasing source symbol index i. In particular, the $\sigma_1 = 10$ source symbols in the first sub-generation (see Eqn. (4)), which are protected by all ϵ coded packets, have the lowest loss probability. Each successive sub-generation is protected by one less coded packet and accordingly has higher loss probability. The source symbols in the last sub-generation $s = \epsilon$ are protected by only one coded packet and hence have the highest loss probability. This uneven loss probability is a limitation of PACE-Uniform without extra tail coded packets ($T = 0$).

The uneven loss probability of PACE-Uniform can be mitigated with extra tail coded packets, as illustrated for $T = 3$ extra tail coded packets in Fig. 5. Importantly, we observe from Fig. 5 that the $T = 3$ extra tail coded packets, which leave $\epsilon - T = 7$ coded packets to be paced according to PACE-Uniform, substantially reduce the loss probabilities of the source symbols near the end of the generation. With the $T = 3$ extra tail coded packets, the source symbols in the second half of the generation experience loss probabilities that are only slightly above the loss probabilities of the corresponding tail RLNC in Fig. 4b. The maximum source symbol delays with $T = 3$, which are not plotted due to space constraints, reach 47.5 slots, i.e., are still substantially shorter than the tail RLNC delays in Fig. 4a. In additional evaluations we found that $T = 1$ and $T = 5$ extra tail coded packets reduce the loss probabilities of the source symbols in the last sub-generation to 6.9 % and 5.7 %, respectively, while increasing the maximum source symbol delay to 36 and 60 slots, respectively. The uneven loss probability of PACE-Uniform is further examined in Section V-C.

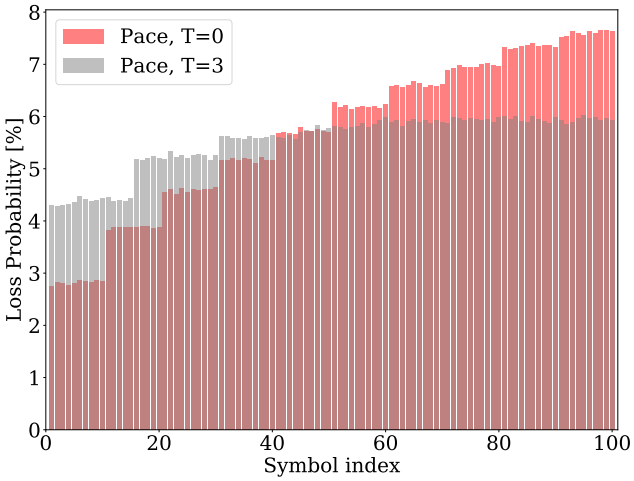
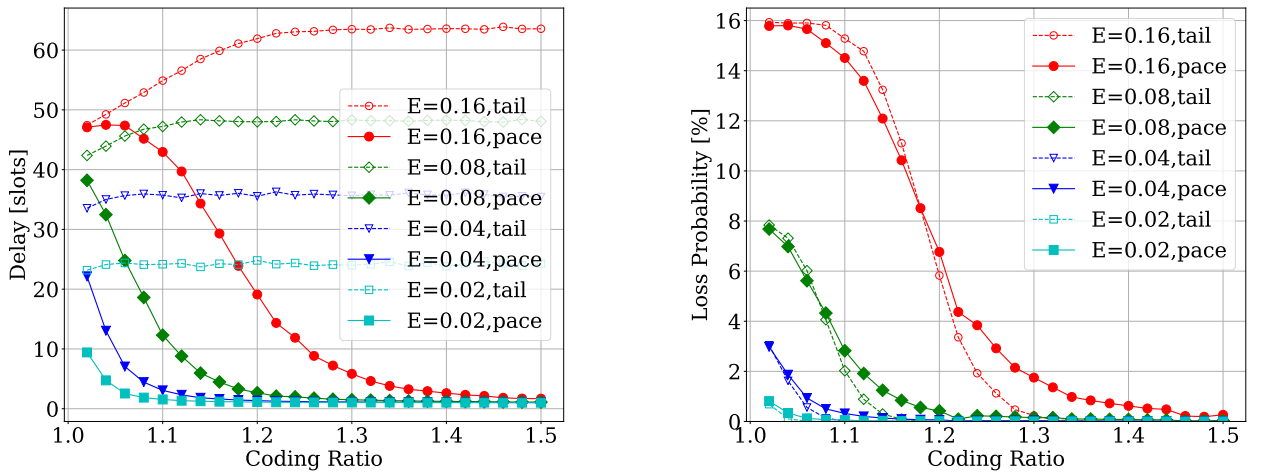


Fig. 5: Average loss probability [%] of each of the $G = 100$ source symbols in a generation for coding ratio $C = 1.1$ and uniform channel drop probability $E = 0.1$: PACE-Uniform with $T = 3$ extra tail coded packets mitigates the loss probability increase for the last symbols with PACE-Uniform (with $T = 0$).

B. Mean Delay and Loss Probability Across a Generation

1) Baseline: Effects of Coding Ratio C and Channel Drop Probability E on Tail RLNC: Figure 6 shows the mean source symbol delay and loss probability for a range of coding ratios C and channel drop probabilities E. The plotted mean delays and loss probabilities were obtained by averaging across all $G = 100$ source symbols in a generation and were obtained for 450,000 independent simulation replications. We observe from Fig. 6a that overall, PACE-Uniform achieves significantly lower delays than systematic (tail) RLNC, while Fig. 6b indicates that both RLNC approaches achieve approximately



(a) Delay (mean across symbols in generation)

(b) Loss Probability (mean across simulated generations)

Fig. 6: Mean delay and loss probability of source symbols with PACE-Uniform compared with systematic (tail) RLNC as a function of coding ratio C for range of channel drop probabilities E. Fixed parameters: Generation size $G = 100$ source symbols, $T = 0$ extra tail coded packets, datagram pool source model.

the same loss probabilities. Examining the delays more closely, we observe that for tail RLNC: (i) the delay curves for higher channel drop probabilities E are overall at a higher delay level, and (ii) a given delay curve (for a given E) initially increases (near the left edge of Fig. 6a) and then levels out as the coding ratio C increases (toward the right of Fig. 6a). With higher channel drop probability E, it is more likely that a source symbol early in a generation is dropped on the channel and thus has to wait for the transmission of all subsequent source symbols and some coded packets until it can be recovered.

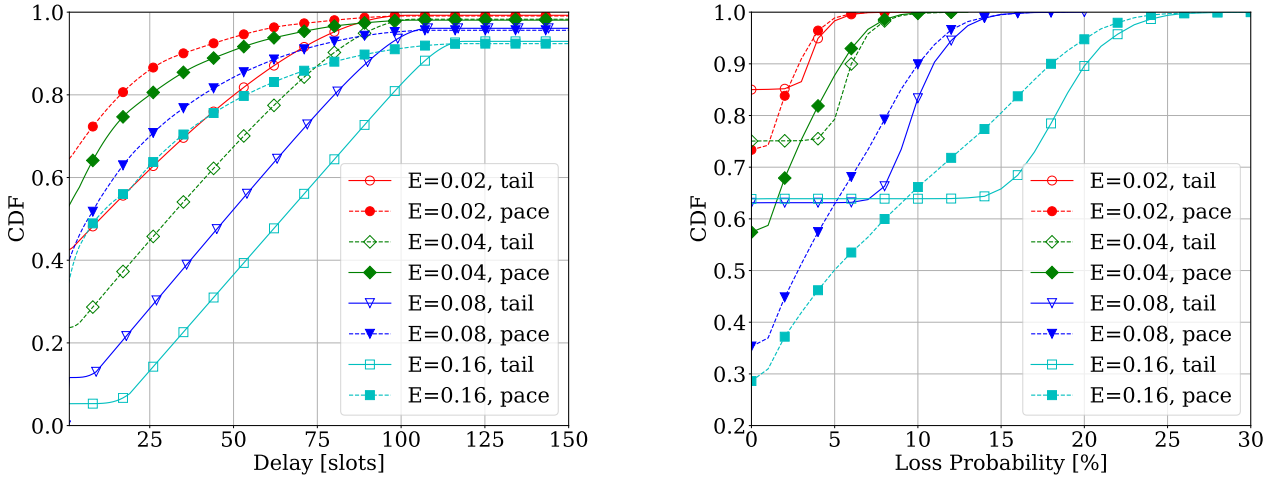
The initial delay increase is small for the low $E = 0.02$ and $E = 0.04$ channel drop probabilities, but becomes more pronounced for the higher $E = 0.08$ and $E = 0.16$ channel drop probabilities. For a small coding ratio C, very few ($\epsilon = \lceil (C-1)G \rceil$) coded packets follow the G systematic (uncoded) source symbols. Thus, the transmission of all systematic source symbols and coded packets can be completed relatively quickly. Thus, the receiver can relatively quickly (after the G transmission slots for the systematic symbols plus the relatively few ϵ coded packets) recover lost source symbols or deterministically determine that lost source symbols cannot be recovered. The few coded packets allow for the recovery of only few lost packets, and thus, there is a high source symbol loss probability for low coding ratios C, especially for high channel drop probabilities E, see Fig. 6b.

When the channel drop probability E is low, only relatively few coded packets are required to recover essentially all lost source symbols. Additional coded packets (beyond the number of lost packets) which are sent with a higher coding ratio C are not useful, as the source symbol loss probability is already reduced to essentially zero for relatively small coding ratios C, see Fig. 6b, $E = 0.02$ and $E = 0.04$ curves. Such additional coded packets do not increase the delay as they follow at the tail of the generation. These additional coded packets do however, unnecessarily increase the transmission bitrate required on the channel. With a higher channel drop probability E, we observe that more coded packets (i.e., a higher coding ratio C) are useful. For instance, we observe from Fig. 6b for the $E = 0.16$ channel drop probability that increasing the coding ratio up to $C = 1.3$ helps in reducing the source symbol loss probability with tail RLNC down to essentially zero. Correspondingly, we observe in Fig. 6a a delay increase up to around $C = 1.3$ for the $E = 0.16$ channel.

2) Effects of Coding Ratio C and Channel Drop Probability E on PACE-Uniform Relative to Tail RLNC Baseline:

a) Delay: Turning to PACE-Uniform RLNC, we observe the opposite delay behavior compared to tail RLNC: the PACE-Uniform delays initially decrease with increasing coding ratio C and then level out near a delay of one. With increasing coding ratio C, more coded packets are interspersed among the systematic systems, i.e., there is a shorter distance between any systematic symbol and the next coded packet, facilitating fast recovery of lost source symbols. On the downside, more coded packets require a higher channel transmission bitrate. Note that for the considered datagram pool traffic model, a higher coding ratio C does not increase the source symbol delay as the source symbols are pulled later from the network layer into the link layer. That is, the source symbols do not need to wait in the link layer for the transmission of additional coded packets; rather the source symbols are pulled into the link layer when it is their turn for transmission.

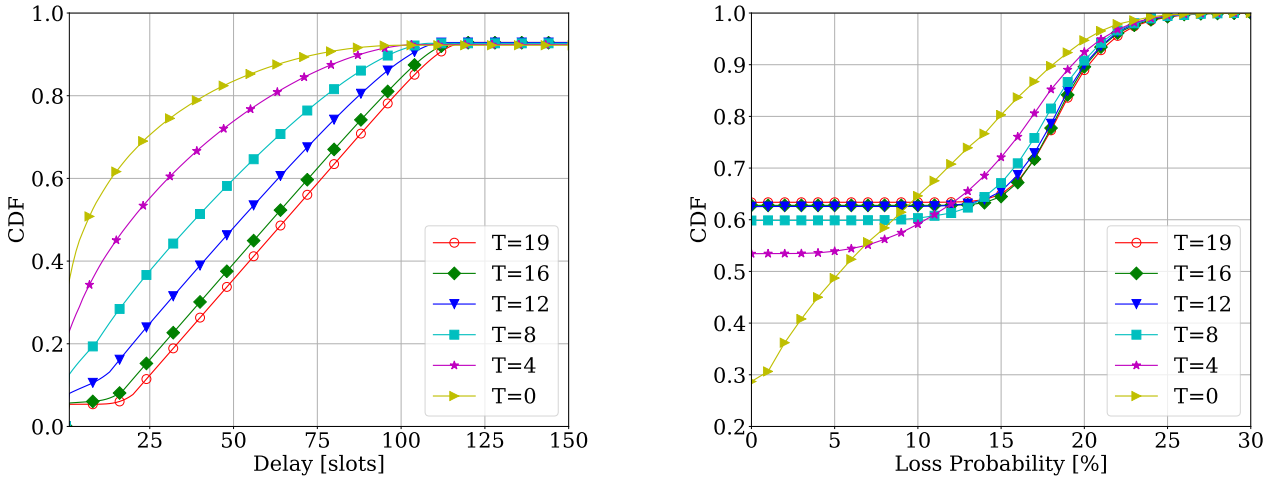
b) Loss: We observe from Fig. 6b that there are two distinct regions of the source symbol loss probability curves of PACE-Uniform in comparison to the curves for tail RLNC: PACE-Uniform achieves somewhat lower loss probabilities than tail RLNC for coding ratios below (i.e., to the left of) a cross-over point. For higher coding ratio (to the right of the cross-over point of the curves), PACE-Uniform gives somewhat higher loss probabilities than tail RLNC. With tail



(a) Delay of individual source symbols

(b) Loss Probability (Number of lost src. symb. / G src. symb.)

Fig. 7: Cumulative distribution functions (CDFs) of source symbol delay and loss probability of PACE-Uniform compared with systematic (tail) RLNC for range of channel drop probabilities E. Fixed parameters: Generation size $G = 100$ source symbols, coding ratio $C = 1/(1-E)$, $T = 0$ extra tail coded packets, datagram pool traffic model.



(a) Delay of individual source symbols

(b) Loss Probability (Number of lost src. symb. / G src. symb.)

Fig. 8: Impact of T extra tail coded packets (with $\epsilon - T$ coded packets remaining for PACE-Uniform scheduling): CDFs of source symbol delay and loss probability. Fixed parameters: Channel drop probability $E = 0.16$, generation size $G = 100$ source symbols, coding ratio $C = 1/(1-E)$, total of $\epsilon = 20$ coded packets, datagram pool traffic model.

RLNC, none of the source symbols that have been dropped on the channel can be recovered if the total number of dropped source symbols is higher than the number of coded packets. In contrast, with PACE-Uniform the source symbols that have been dropped up to (and including) source symbol i can be recovered if the number of coded packets after source symbol i is higher than the number of dropped source symbols up to (and including) symbol i. Thus, with PACE-Uniform, the early source symbols (with low i) are relatively more strongly protected (as they have more subsequent coded packets in the generation) than the symbols later (with high i) in the generation. Thus, for low coding ratios C and high channel drop probabilities E, PACE-Uniform may still be able to recover some of the early source symbols in the generation, whereas tail RLNC cannot recover any lost symbols.

For high coding ratios C, e.g., for $C = 1.3$ for $E = 0.16$, tail RLNC has enough coded packets overall to recover all lost source symbols. In contrast, PACE-Uniform may have some losses late (with high i) in the generation that are not protected by enough subsequent coded packets to be recovered. Overall, the transition of the curves in Fig. 6b from high to low loss probabilities tends to be more abrupt with tail RLNC, whereas PACE-Uniform exhibits a more gradual transition. This transition behavior reflects that tail RLNC either recovers all or none of the dropped symbols, i.e., has either zero losses or all of the dropped symbols become link layer losses. On the other hand, PACE-Uniform degrades more gracefully by potentially still recovering some dropped symbols, even if not

all of the dropped symbols can be recovered.

C. Distribution Function of Individual Source Symbol Delays and Losses

a) Delay of PACE-Uniform with $T = 0$ Extra Tail Coded Packets: We plot the cumulative distribution function (CDF) of the source symbol delay in Fig. 7a. Since we do not consider lost source symbols in the delay measurements (see Section III-D2), the delay CDF curves level out at one minus the loss probability. We observe that PACE-Uniform exhibits concave CDF curves that indicate that low delays are achieved with relatively high probability, whereas tail RLNC exhibits linearly increasing probabilities for higher delays. For instance, for $E = 0.16$, 60 % of the source symbols experience delays of less than 75 slots with tail RLNC; whereas with PACE-Uniform, approximately 60 % of the source symbols experience delays of less than 25 slots. For $E = 0.02$, about 80 % of the tail RLNC symbols have less than 50 slots delay; whereas with PACE-Uniform close to 94 % of the symbols have less than 50 slots delay. We also observe from the higher starting points of the PACE-Uniform CDF curves at the left edge of Fig. 7a compared to the tail RLNC starting points that far more source symbols experience the minimal delay of one slot. For $E = 0.04$, for instance, only about 25 % of the tail RLNC packets have a delay of one slot, compared to approximately 52 % of the PACE-Uniform symbols.

b) Loss of PACE-Uniform with $T = 0$ Extra Tail Coded Packets: Fig. 7b shows the CDFs of the loss probabilities, which are equivalent to the number of lost source symbols per generation for the considered $G = 100$ generation size. We observe that for the high drop probability $E = 0.16$ channel, tail RLNC entirely avoids losses for about 63 % of the generations, but loses between approximately 15 and 25 symbols of the remaining 37 % of the replications. In contrast, PACE-Uniform keeps only about 28 % of the generations loss-free, and then loses a gradually (near linearly) increasing number of symbols from the remaining 72 % of the generations.

c) Impact of T Extra Tail Coded Packets: Fig. 8 shows the source symbol delay and loss CDFs for a range of the number T of extra coded packets at the tail from $T = 0$, i.e., PACE-Uniform without any extra tail coded packets, to $T = \epsilon - 1$, i.e., tail RLNC. We observe that allocating a relatively small number of $T = 4$ out of the total of $\epsilon = 20$ coded packets to the tail of the generation (in addition to the one coded packet positioned at the tail by PACE-Uniform) substantially reduces the chance of a source symbol being lost while only moderately increasing the delay. In particular, Fig. 8b indicates that the probability of lossless link layer transfer is increased from approximately 0.28 for PACE-Uniform ($T = 0$) to about 0.53 with $T = 4$ extra tail coded packets. These $T = 4$ extra tail coded packets incur only moderate delay increases and preserve the concave shape of the delay CDF in Fig. 8a. $T = 8$ extra tail coded packets increase the probability of lossless delivery to 0.60, which is very close to the 0.63 probability of conventional tail RLNC. Thus, more than $T = 8$ extra tail coded packet do not substantially contribute to the increase in the reliability of the RLNC transport in this considered scenario, but they increase the delays.

TABLE II: Evaluation of RLNC decoding computation times: Average number of symbol vector operations per generation as a function of number of extra tail coded packets T; $T = 0$ corresponds to PACE-Uniform; $T = \epsilon - 1$ corresponds to tail RLNC. Fixed parameters as for Fig. 8.

	T			
	0	4	8	$\epsilon - 1 = 19$
Symb. Vector Ops.	896	1095	1257	1654

D. Impact on RLNC Computation Times

In this section we briefly examine the impact of the PACE concept on the RLNC computation times (computation costs). Generally, the computation time for RLNC encoding scales as $O(G)$ with the generation size G, while the computation time for RLNC decoding scales as $O(G^3)$ [98], [100] due to the matrix inversion and multiplication involved in the decoding (see Section III-A). Due to the substantially higher RLNC decoding complexity compared to the encoding complexity, RLNC computation studies often focus on the decoding [97]. In order to evaluate the decoding computation times, we counted, similar to the evaluation approach in [100], the symbol vector operations for the RLNC decoding in Kodo [24], [99], [101]. The setting considered in Fig. 8 required the average numbers of symbol vector operations reported in Table II for decoding a given generation. We observe from Table II that PACE-Uniform, which corresponds to $T = 0$, requires only about half the operations compared to tail RLNC, which corresponds to $T = \epsilon - 1$.

This reduction of the computational complexity is due to the increased sparsity, i.e., the increased number of zero-valued coding coefficients in the PACE-Uniform coding coefficient matrix $\mathbf{C}$ compared to the $\mathbf{C}$ for tail RLNC. More specifically, in PACE-Uniform, the coded packet for a sub-generation s is based only on a linear combination of the source symbols in the sub-generations $1, 2, \ldots s$, whereas the source symbols of the subsequent sub-generations $s+1, \ldots, \epsilon$ are not considered, see Section IV-B. Accordingly, considering a scenario with an integer G/ϵ, for simplicity, only the proportion $(1/\epsilon + 2/\epsilon + \cdots + 1)/\epsilon$ of the coding coefficients are actual random coefficients, as illustrated for a scenario with $G = 9$ and $\epsilon = 3$ for $GF(2)$ in Fig. 2c. The complementary proportion $1 - (1/\epsilon + 2/\epsilon + \cdots + 1)/\epsilon$ of the coding coefficients is set to zero, as they correspond to the not considered subsequent sub-generations. Thus, for moderately large numbers ϵ of coded packets in a generation, up to nearly half of the coding coefficients are set to zero in PACE-Uniform. In the $GF(2)$ scenario illustrated in Fig. 2c, a random coding coefficient is zero-valued with probability 1/2; however, in the commonly considered $GF(2^8)$ scenario, which we consider in our evaluations, a random coefficient is zero-valued with probability $1/2^8$, which is negligible. Hence, for $GF(2^8)$, the proportion of the number of zero-valued coding coefficients is negligible for conventional tail RLNC; while close to half of the PACE-Uniform coding coefficients are zero-valued. Thus,

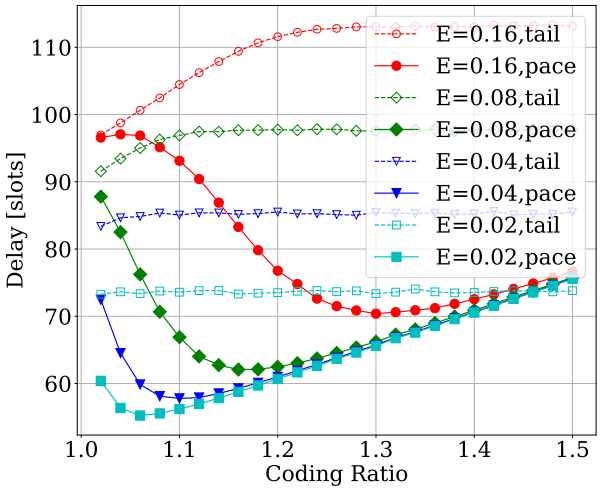


Fig. 9: Burst traffic model: Mean delay and loss probability of source symbols with PACE-Uniform compared with systematic (tail) RLNC as a function of coding ratio C for range of channel drop probabilities E. Fixed parameters: Generation size $G = 100$, $T = 0$ extra tail coded packets.

the asymptotic computational complexity of PACE decoding scales still as $O(G^3)$. However, the increased proportion of zero-valued coding coefficients, which is also referred to as increased *sparsity* of the coding coefficient matrix $\mathbf{C}$, simplifies the RLNC decoding computations [100], [102], resulting in the observed reduction of the decoding computation costs.

E. Effects of Traffic Model

The evaluations so far considered the datagram pool traffic model. In this section, we examine the impact of the traffic model on the PACE-Uniform source symbol delay and loss relative to tail RLNC. Fig. 9 plots the average source symbol delay for the burst traffic model which injects all G source symbols of a generation at the beginning instant of the τ generation transmission slots into the link layer. The source symbol loss plot is not included as it is essentially identical to the loss plot for the datagram pool traffic model in Fig. 6b. Comparing the delays in Fig. 9 with Fig. 6a, we observe that the tail RLNC delay curves have the same shape, but are shifted up by roughly $G/2 = 50$ time slots. With the burst traffic model, source symbol i, $i = 1, 2, \ldots, G$, has to wait for i slots in the link layer for its transmission slot; whereas, in the datagram pool traffic model, it was pulled from the network layer at the beginning of its transmission slot. Thus, source symbol i incurs an additional delay of i slots in the burst model, which averages to $G/2$ slots across the G source symbols in a generation.

PACE-Uniform experiences a similar general delay increase by roughly $G/2$ slots. In addition, the delay curves for PACE-Uniform converge towards a linearly increasing slope for increasing coding ratio C. This linear delay increase is due to the increased wait of source symbols for their transmission slots as more and more coded packets are interspersed among the source symbol transmission slots. Nevertheless, PACE-Uniform still achieves substantial delay reductions compared to tail RLNC. Moreover, in practical operation, it is reasonable to restrict the coding ratio C to values that ensure nearly zero losses, as observed from Fig. 6b, e.g., $C \leq 1.2$ for $E = 0.08$. Higher coding ratios C would increase the required transmission bitrate and the delay with the burst traffic model without significantly reducing the loss probability.

In additional evaluations, we found that the constant systematic packet rate traffic model gives similar shapes of the delay curves as the burst traffic model in Fig. 9, however, at the low delay level of the datagram pool traffic model (i.e., without the $G/2$ delay level increase of the burst traffic model in Fig. 9 relative to Fig. 6a). In other words, the delay plot of the constant systematic packet rate traffic model is essentially the delay plot in Fig. 9 shifted down by $G/2 = 50$ slots.

The constant packet rate traffic model gives similar delays to the constant systematic packet rate traffic model, with very slightly more pronounced delay reductions with PACE-Uniform compared to tail RLNC. PACE-Uniform requires the source symbols a little later than tail RLNC as the source symbol transmission slots are more evenly spread out.

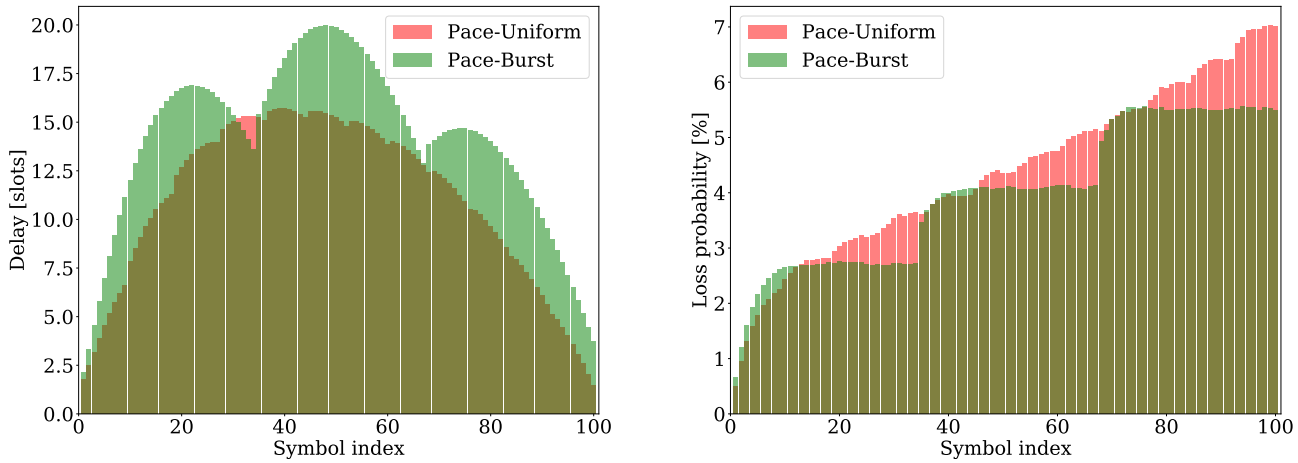
F. Summary of PACE-Uniform Evaluation

Overall, our extensive evaluations indicate that both PACE-Uniform and tail RLNC exhibit similar source symbol loss probability performance. PACE-Uniform exhibits uneven loss probabilities within a generation, which can be mitigated by allocating a small portion of the coded packets as extra tail coded packets. We have found that PACE-Uniform substantially reduces the source symbol delays compared to tail RLNC, even when allocating a small portion of the coded packets as extra tail coded packets. Overall, we conclude that PACE-Uniform significantly reduces the link layer delay while providing similar link layer reliability as tail RLNC and consuming the same transmission bitrate.

VI. PACE-Burst

A. Motivation

We have considered uniformly distributed channel drops so far, i.e., each packet transmitted on the channel was dropped independently with probability E. Many wireless systems exhibit bursty channel losses, i.e., drop a prescribed mean number of Δ successive packets. Such a bursty channel can be modelled with the Gilbert-Elliot model, a two-state Markov chain model with a good channel state that does not drop packets and a bad channel state (with mean sojourn time Δ slots) that drops all packets. Tail RLNC includes all coded packets at the end of a generation of source symbols, and is thus not directly affected by the distribution of channel drops within a generation (uniform or bursty). On the other hand, PACE-Uniform intersperses individual coded packets within the source symbols. Each individual coded packet can only aid in the recovery of one additional dropped packet. Thus, a burst of multiple successively dropped packets would require multiple coded packets for recovery. PACE-Burst attempts to counter these bursty losses with bursts of coded packets.



(a) Delay

(b) Loss Probability

Fig. 10: Averages (over 500,000 simulation replications) of in-order delay [slots] and loss probability [%] for each of the $G = 100$ source symbols in a generation with coding ratio $C = 1.12$ for bursty channel losses (mean loss burst $\Delta = 3$ packets, overall channel drop probability $E = 0.08$): PACE-Burst with bursts of $B = 4$ coded packets, i.e., $\nu = 3$ sub-generations, reduces the loss probability at the expense of increased delay for the source symbols that are near the middle of a sub-generation.

B. PACE-Burst Specification

PACE-Burst groups the coded packets into bursts of B, $B \geq 1$, coded packets. The bursts of coded packets are distributed uniformly among the systematic source symbols. We define the number of coded packet bursts as

$$\nu = \left\lfloor \frac{\epsilon}{B} \right\rfloor . \tag{5}$$

We assign the ϵ coded packets uniformly to the ν coded packet bursts. If ϵ/ν is not an integer, then we assign the "extra" $\epsilon \bmod \nu$ coded packet to the last packet burst in order to provide added protection to the overall generation. In particular, the number μ_s of coded packets in coded packet burst s, $s = 1, 2, \ldots, \nu$, is

$$\mu_s = \begin{cases} B & s = 1, 2, \ldots, \nu - 1 \\ B + (\epsilon \bmod \nu) & s = \nu. \end{cases} \tag{6}$$

The number σ_s of source symbols in sub-generation s, $s = 1, 2, \ldots, \nu$, is evaluated with Eqn. (4) with ϵ replaced by the number of bursts ν. Sub-generation s consists of σ_s systematic source symbols followed by μ_s coded packets. Note that for the special case $B = 1$, PACE-Burst corresponds to PACE Uniform; whereas for $B = \epsilon$, PACE-Burst corresponds to tail RLNC.

C. Evaluation of PACE-Burst

1) Delay and Loss Probability for Individual Source Symbols: Fig. 10 shows the delay and loss probability for each individual source symbol i, $i = 1, 2, \ldots, 100 = G$, in a generation for bursty channel drops. Comparing PACE-Uniform for uniform channel drops in Fig. 4 with PACE-Uniform for bursty channel drops in Fig. 10 indicates that the delay exhibits the same overall shape; while for the loss probability, the clearly cut steps in the "staircase" in Fig. 4 have turned into ripples superimposed on an increasing slope in Fig. 10. That is, bursty channel drops tend to "wash out" the clear loss probability steps that occur for uniform channel drops.

TABLE III: Mean delay and loss probability (across a generation) for PACE-Burst with $B = 1$ (equivalent to PACE-Uniform), 2, 4, $\epsilon/2$, and $\epsilon = \lceil (C-1)G \rceil$ (equivalent to tail RLNC) coded packets in a burst for different coding ratios C for $G = 100$ source symbols in generation for bursty channel losses with $\Delta = 3$, $E = 0.08$.

		B				
		1	2	4	$\epsilon/2$	ϵ
$C = 1.12$	Delay [slots]	10.0	11.6	14.1	17.4	32.8
	Loss [%]	4.6	4.2	4.0	3.8	3.3
$C = 1.16$	Delay [slots]	7.9	8.7	10.0	15.4	33.7
	Loss [%]	3.7	3.7	3.3	2.9	2.3

We observe for the comparison of PACE-Burst with PACE-Uniform in Fig 10a, that PACE-Burst incurs higher delay for source symbols that are near the middle of a sub-generation. This is because any drop, say around source symbol $i = 20$ can only be recovered by the next coded packet burst. Thus, each sub-generation has delay dynamics that are similar to tail RLNC in Fig. 4a. For the loss probability, we observe from Fig. 10b that PACE-Burst has overall lower loss probabilities than PACE Uniform. This is mainly because the coded packets in PACE-Burst occur on average later in the generation, i.e., protect more source symbols than in PACE-Uniform.

2) Mean Delay and Loss Probability Across a Generation: Table III reports the mean delay and loss probabilities across a generation of $G = 100$ source symbols for sizes B of the coded packet bursts ranging from $B = 1$ coded packet, i.e., PACE-Uniform, to $B = \epsilon$, i.e., tail RLNC. We observe from Table III that the number B of coded packets in a burst provides a "tuning knob" that can adjust the mean delay and loss probability performance between PACE-Uniform and tail RLNC. Small B make PACE-Burst behave similar to PACE-Uniform, substantially reducing the average delays at the

expense of a relatively minor increase of the loss probability. Increasing B weakens the delay reduction while mitigating the increase in loss probability (relative to tail RLNC).

We also observe from Table III that the higher coding ratio $C = 1.16$ achieves generally lower loss probabilities and shorter mean delays than the lower $C = 1.12$ coding ratio, except for tail RLNC ($B = \epsilon$) which gives higher delays for the higher C. The lower loss probabilities are due to the generally stronger FEC protection provided by a higher coding ratio C. The lower delays are due to the recovery of more source symbols early in generation when more coded packets are interspersed among the source symbols (in conjunction with the considered datagram pool traffic model, which pulls datagrams from the network layer when it is their turn for transmission, see Fig. 6a). For tail RLNC, the higher C recovers more source symbols and thus reduces the loss probability, but requires waiting for more coded packets, which slightly increases the average delay.

VII. Conclusion

We have examined the scheduling (pacing) of coded packets among the uncoded source symbols in generation based systematic random linear network coding (RLNC). Our proposed PACE-Uniform approach to RLNC positions individual coded packets equidistantly among the uncoded source symbols in a generation. Our proposed PACE-Burst approach generalizes the PACE approach by positioning bursts of $B,\ B \geq 1$, coded packets in equidistant manner among the uncoded source symbols. Our extensive simulation evaluations indicated that the PACE approach substantially reduces the mean source symbol delay in an RLNC system compared to conventional systematic RLNC with coded packets at the tail end of a generation, while achieving nearly the same loss probability (of source symbols that cannot be decoded).

There are several interesting directions for future research on pacing coded packets in RLNC. One direction is to examine the PACE approach in the context of specific applications, such as multimedia applications, e.g., video streaming. Another direction is to investigate the PACE approach within the context of variations of generation based RLNC, e.g., for sliding window approaches, where the pacing could be applied within a given considered window. Yet another direction is to examine the PACE approach in conjunction with sparse RLNC which can further speed up the RLNC processing [100], [103], [104].

References

[1] R. Bassoli, H. Marques, J. Rodriguez, K. W. Shum, and R. Tafazolli, "Network coding theory: A survey," *IEEE Commun. Surv. & Tut.*, vol. 15, no. 4, pp. 1950–1978, 4th Qu. 2013.

[2] P. Chau, T. D. Bui, Y. Lee, and J. Shin, "Efficient data uploading based on network coding in LTE-Advanced heterogeneous networks," in *Proc. IEEE Int. Conf. on Adv. Commun. Techn. (ICACT)*, 2017, pp. 252–257.

[3] M. Z. Farooqi, S. M. Tabassum, M. H. Rehmani, and Y. Saleem, "A survey on network coding: From traditional wireless networks to emerging cognitive radio networks," *Journal of Network and Computer Applications*, vol. 46, pp. 166–181, Nov. 2014.

[4] X. Li, Q. Chang, and Y. Xu, "Queueing characteristics of the best effort network coding strategy," *IEEE Access*, vol. 4, pp. 5990–5997, 2016.

[5] J. S. Liu, C. H. R. Lin, and J. Tsai, "Delay and energy tradeoff in energy harvesting multi-hop wireless networks with inter-session network coding and successive interference cancellation," *IEEE Access*, vol. 5, pp. 544–564, 2017.

[6] H. Khamfroush, D. E. Lucani, P. Pahlevani, and J. Barros, "On optimal policies for network-coded cooperation: theory and implementation," *IEEE J. Sel. Areas in Commun.*, vol. 33, no. 2, pp. 199–212, Feb. 2015.

[7] A. Nessa, M. Kadoch, and B. Rong, "Fountain coded cooperative communications for LTE-A connected heterogeneous M2M network," *IEEE Access*, vol. 4, pp. 5280–5292, 2016.

[8] Y. Yang, W. Chen, O. Li, Q. Liu, and L. Hanzo, "Truncated-ARQ aided adaptive network coding for cooperative two-way relaying networks: Cross-layer design and analysis," *IEEE Access*, vol. 4, pp. 9361–9376, 2016.

[9] Y. Yang, W. Chen, O. Li, and L. Hanzo, "Joint rate and power adaptation for amplify-and-forward two-way relaying relying on analog network coding," *IEEE Access*, vol. 4, pp. 2465–2478, 2016.

[10] H. Kang, H. Yoo, D. Kim, and Y. S. Chung, "CANCORE: Context-Aware Network COded REpetition for VANETs," *IEEE Access*, vol. 5, pp. 3504–3512, 2017.

[11] 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.

[12] M. Rekab-Eslami, M. Esmaeili, and T. A. Gulliver, "Multicast convolutional network codes via local encoding kernels," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2017.

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

[14] 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.

[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 in 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 Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7892–7900, Dec. 2015.

[17] M. Sipos, J. Gahm, N. Venkat, and D. Oran, "Erasure coded storage on a changing network: The untold story," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2016, pp. 1–6.

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

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

[20] P. A. Chou and Y. Wu, "Network coding for the internet and wireless networks," *IEEE Signal Proc. Mag.*, vol. 24, no. 5, pp. 77–85, May 2007.

[21] Y. Benfattoum, S. Martin, and K. Al Agha, "QoS for real-time reliable multicasting in wireless multi-hop networks using a generation-based network coding," *Computer Netw.*, vol. 57, no. 6, pp. 1488–1502, Apr. 2013.

[22] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[23] K. Xiong, Y. Zhang, P. Fan, and H.-C. Yang, "Evaluation framework for user experience in 5G systems: On systematic rateless-coded transmissions," *IEEE Access*, vol. 4, pp. 9108–9118, 2016.

[24] J. Heide, M. V. Pedersen, F. H. Fitzek, and T. Larsen, "Network coding for mobile devices–systematic binary random rateless codes," in *Proc. IEEE ICC Workshops*, 2009, pp. 1–6.

[25] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Proc. IEEE Workshop on Network Coding, Theory and Applications (NetCod)*, 2008, pp. 1–6.

[26] D. E. Lucani, M. Médard, and M. Stojanovic, "Systematic network coding for time-division duplexing," in *Proc. IEEE Int. Symp on Information Theory Proceedings (ISIT)*, 2010, pp. 2403–2407.

[27] R. Prior and A. Rodrigues, "Systematic network coding for packet loss concealment in broadcast distribution," in *Proc. IEEE Int. Conf. on Information Networking (ICOIN)*, 2011, pp. 245–250.

[28] S. A. Khan, M. Moosa, F. Naeem, M. H. Alizai, and J. M. Kim, "Protocols and mechanisms to recover failed packets in wireless networks: History and evolution," *IEEE Access*, vol. 4, pp. 4207–4224, 2016.

[29] A. Eryilmaz, A. Ozdaglar, M. Médard, and E. Ahmed, "On the delay and throughput gains of coding in unreliable networks," *IEEE Trans. on Inform. Theory*, vol. 54, no. 12, pp. 5511–5524, Dec. 2008.

[30] X. Li, C.-C. Wang, and X. Lin, "Throughput and delay analysis on uncoded and coded wireless broadcast with hard deadline constraints," in *Proc. IEEE Infocom*, 2010, pp. 1–5.

[31] P. Parag and J.-F. Chamberland, "Queueing analysis of a butterfly network for comparing network coding to classical routing," *IEEE Trans. Inform. Theory*, vol. 56, no. 4, pp. 1890–1908, Apr. 2010.

[32] O. B. Rhaiem and L. Chaari, "Information transmission based on network coding over wireless networks: a survey," *Telecommunication Systems*, vol. 65, no. 4, pp. 551–565, Aug. 2017.

[33] A. A. Yazdi, S. Sorour, S. Valaee, and R. Y. Kim, "Optimum network coding for delay sensitive applications in WiMAX unicast," in *Proc. IEEE Infocom*, 2009, pp. 2576–2580.

[34] E. Drinea, L. Keller, and C. Fragouli, "Real-time delay with network coding and feedback," *Physical Commun.*, vol. 6, pp. 100–113, Mar. 2013.

[35] V. Roca, B. Teibi, C. Burdinat, T. Tran, and C. Thienot, "Block or convolutional AL-FEC codes? a performance comparison for robust low-latency communications," HAL Inria, Id: hal-01395937v2, Tech. Rep., 2017.

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

[37] S. Wunderlich, F. Gabriel, S. Pandi, and F. H. Fitzek, "We don't need no generation–a practical approach to sliding window RLNC," in *Proc. IEEE Wireless Days*, 2017, pp. 218–223.

[38] 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, 2014.

[39] A. Douik, M. S. Karim, P. Sadeghi, and S. Sorour, "Delivery time reduction for order-constrained applications using binary network codes," in *Proc. IEEE Wireless Commun. and Netw. Conf. (WCNC)*, 2016, pp. 1–6.

[40] A. Douik, S. Sorour, T. Al-Naffouri, and S. Alouini, "Instantly decodable network coding: From centralized to device-to-device communications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1201–1224, 2nd Qu. 2017.

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

[42] 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.

[43] J. K. Sundararajan, D. Shah, and M. Médard, "Online network coding for optimal throughput and delay-the three-receiver case," in *Proc. IEEE Int. Symp. on Inform. Theory and Its Appl. (ISITA)*, 2008, pp. 1–6.

[44] ——, "ARQ for network coding," in *Proc. IEEE Int. Symp. on Information Theory*, 2008, pp. 1651–1655.

[45] J. K. Sundararajan, P. Sadeghi, and M. Médard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *Proc. IEEE Workshop on Netw. Coding, Th., and Appl.*, 2009, pp. 1–6.

[46] J. K. Sundararajan, D. Shah, M. Médard, M. Mitzenmacher, and J. Barros, "Network coding meets TCP," in *Proc. IEEE Infocom*, 2009, pp. 280–288.

[47] Y. Lin, B. Liang, and B. Li, "SlideOR: Online opportunistic network coding in wireless mesh networks," in *Proc. IEEE Infocom*, 2010, pp. 1–5.

[48] J. Cloud, D. Leith, and M. Médard, "A coded generalization of selective repeat ARQ," in *Proc. IEEE Infocom*, 2015, pp. 2155–2163.

[49] M. Karzand, D. J. Leith, J. Cloud, and M. Medard, "FEC for lower in-order delivery delay in packet networks," *arXiv preprint arXiv:1509.00167*, 2015.

[50] A. Garcia-Saavedra, M. Karzand, and D. J. Leith, "Low delay random linear coding and scheduling over multiple interfaces," *IEEE Transactions on Mobile Computing, in print*, 2017.

[51] J. Barros, R. A. Costa, D. Munaretto, and J. Widmer, "Effective delay control in online network coding," in *Proc. IEEE Infocom*, 2009, pp. 208–216.

[52] R. A. Costa, D. Munaretto, J. Widmer, and J. Barros, "Informed network coding for minimum decoding delay," in *Proc. IEEE Int. Conf. on Mobile Ad Hoc and Sensor Systems*, 2008, pp. 80–91.

[53] A. Fu, P. Sadeghi, and M. Médard, "Dynamic rate adaptation for improved throughput and delay in wireless network coded broadcast," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6, pp. 1715–1728, Dec. 2014.

[54] P. Sadeghi, R. Shams, and D. Traskov, "An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels," *EURASIP J. Wireless Communications and Networking*, vol. 2010, no. 618016, pp. 1–14, Dec. 2010.

[55] S. Sorour and S. Valaee, "An adaptive network coded retransmission scheme for single-hop wireless multicast broadcast services," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 869–878, Jun. 2011.

[56] W.-L. Yeow, A. T. Hoang, and C.-K. Tham, "Minimizing delay for multicast-streaming in wireless networks with network coding," in *Proc. IEEE Infocom*, 2009, pp. 190–198.

[57] Y. Yuan, K. Wu, W. Jia, and Y. Peng, "On the queueing behavior of inter-flow asynchronous network coding," *Computer Communications*, vol. 35, no. 13, pp. 1535–1548, Jul. 2012.

[58] X. Li, C.-C. Wang, and X. Lin, "Optimal immediately-decodable inter-session network coding (IDNC) schemes for two unicast sessions with hard deadline constraints," in *Proc. IEEE Annual Allerton Conf. on Commun., Control, and Comp. (Allerton)*, 2011, pp. 784–791.

[59] Y. Li, P. Vingelmann, M. V. Pedersen, and E. Soljanin, "Round-robin streaming with generations," in *Proc. IEEE Int. Symp. on Network Coding (NetCod)*, 2012, pp. 143–148.

[60] I. Chatzigeorgiou and A. Tassi, "Decoding delay performance of random linear network coding for broadcast," *IEEE Transactions on Vehicular Technology, in print*, 2017.

[61] G. Joshi, Y. Kochman, and G. Wornell, "On throughput-smoothness trade-offs in streaming communication," *arXiv preprint arXiv:1511.08143*, 2015.

[62] A. Moreira, L. Almeida, and D. E. Lucani, "Merging network coding with feedback management in multicast streaming," *ACM SIGBED Review*, vol. 12, no. 3, pp. 49–52, Jun. 2015.

[63] C. W. Sung, K. W. Shum, L. Huang, and H. Y. Kwan, "Linear network coding for erasure broadcast channel with feedback: Complexity and algorithms," *IEEE Transactions on Information Theory*, vol. 62, no. 5, pp. 2493–2503, May 2016.

[64] D. E. Lucani, M. Medard, and M. Stojanovic, "On coding for delay–network coding for time-division duplexing," *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2330–2348, Apr. 2012.

[65] B. Shrader and A. Ephremides, "Queueing delay analysis for multicast with random linear coding," *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp. 421–429, Jan. 2012.

[66] M. Yu and P. Sadeghi, "Approximating throughput and packet decoding delay in linear network coded wireless broadcast," *arXiv preprint arXiv:1701.04551*, 2017.

[67] D. Zeng, S. Guo, H. Jin, and V. Leung, "Segmented network coding for stream-like applications in delay tolerant networks," in *Proc. IEEE Globecom*, 2011, pp. 1–5.

[68] J. Krigslund, F. Fitzek, and M. V. Pedersen, "On the combination of multi-layer source coding and network coding for wireless networks," in *Proc. IEEE Int. Workshop on Computer Aided Modeling and Design of Commun. Links and Netw. (CAMAD)*, 2013, pp. 1–6.

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

[70] V. N. Swamy, P. Rigge, G. Ranade, A. Sahai, and B. Nikolić, "Network coding for high-reliability low-latency wireless control," in *Proc. IEEE Wireless Commun. and Netw. Conf. (WCNC)*, 2016, pp. 1–7.

[71] K. Yu, J. Yue, Z. Lin, J. Åkerberg, and M. Björkman, "Achieving reliable and efficient transmission by using network coding solution in industrial wireless sensor networks," in *Proc. IEEE Int. Symp. on Industrial Electronics (ISIE)*, 2016, pp. 1162–1167.

[72] N. d. S. R. Júnior, R. C. Tavares, M. A. Vieira, L. F. Vieira, and O. Gnawali, "CodeDrip: Improving data dissemination for wireless sensor networks with network coding," *Ad Hoc Networks*, vol. 54, pp. 42–52, Jan. 2017.

[73] J. Claridge and I. Chatzigeorgiou, "Probability of partially decoding network-coded messages," *IEEE Commun. Letters, in print*, 2017.

[74] G. Cocco, T. de Cola, and M. Berioli, "Performance analysis of queueing systems with systematic packet-level coding," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, 2015, pp. 4524–4529.

[75] 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. Inform. Th.*, vol. 57, no. 2, pp. 1111–1123, Feb. 2011.

[76] M. Nistor, D. E. Lucani, T. T. Vinhoza, R. A. Costa, and J. Barros, "On the delay distribution of random linear network coding," *IEEE J. Sel. Areas in Commun.*, vol. 29, no. 5, pp. 1084–1093, May 2011.

[77] J.-P. Thibault, S. Yousefi, and W.-Y. Chan, "Throughput performance of generation-based network coding," in *Proc. IEEE Canadian Workshop on Information Theory (CWIT)*, 2007, pp. 89–92.

[78] K. Xu, B. Dai, F. Zhang, B. Huang, and B. Zhang, "Latency estimation for time-sensitive applications under wireless network coding scheme," in *Proc. IEEE Int. Conf. on Communications and Mobile Computing (CMC)*, vol. 2, 2009, pp. 263–266.

[79] W. Zeng, C. T. Ng, and M. Médard, "Joint coding and scheduling optimization in wireless systems with varying delay sensitivities," in *Proc. IEEE SECON*, 2012, pp. 416–424.

[80] B. Swapna, A. Eryilmaz, and N. B. Shroff, "Throughput-delay analysis of random linear network coding for wireless broadcasting," *IEEE Trans. on Inform. Theory*, vol. 59, no. 10, pp. 6328–6341, Oct. 2013.

[81] L. Yang, Y. E. Sagduyu, and J. H. Li, "Adaptive network coding for scheduling real-time traffic with hard deadlines," in *Proc. ACM Int. Symp. on Mobile Ad Hoc Netw. and Comp.*, 2012, pp. 105–114.

[82] X. Zhong, Y. Qin, and L. Li, "TCPNC-DGSA: Efficient network coding scheme for TCP in multi-hop cognitive radio networks," *Wireless Personal Communications*, vol. 84, no. 2, pp. 1243–1263, Sep. 2015.

[83] M. Halloush and H. Radha, "Network coding with multi-generation mixing: A generalized framework for practical network coding," *IEEE Trans. Wireless Commun.*, vol. 10, no. 2, pp. 466–473, Feb. 2011.

[84] ——, "The unequal protection of network coding with multi-generation mixing," in *Proc. IEEE Int. Conf. Innov. in Inform. Techn. (IIT)*, 2011, pp. 29–34.

[85] M. D. Halloush and H. Radha, "A framework for video network coding with multi-generation mixing," *Journal of Communications*, vol. 7, no. 3, pp. 192–201, Mar. 2012.

[86] J. Bhatia, A. Patel, and Z. Narmawala, "Review on variants of network coding in wireless ad-hoc networks," in *Proc. IEEE Nirma Univ. Int. Conf. on Engineering (NUiCONE)*, 2011, pp. 1–6.

[87] R. Liu, J. Hao, and Y. Guo, "An improved method of muti-generation mixing network coding," in *Proc. IEEE World Congress on Information and Commun. Technol. (WICT)*, 2012, pp. 1086–1091.

[88] A. J. Aljohani, S. X. Ng, and L. Hanzo, "Distributed source coding and its applications in relaying-based transmission," *IEEE Access*, vol. 4, pp. 1940–1970, 2016.

[89] G. Giacaglia, X. Shi, M. Kim, D. E. Lucani, and M. Médard, "Systematic network coding with the aid of a full-duplex relay," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, 2013, pp. 3312–3317.

[90] M. Esmaeilzadeh, N. Aboutorab, and P. Sadeghi, "Joint optimization of throughput and packet drop rate for delay sensitive applications in TDD satellite network coded systems," *IEEE Trans. Commun.*, vol. 62, no. 2, pp. 676–690, Feb. 2014.

[91] S. Teerapittayanon, K. Fouli, M. Médard, M.-J. Montpetit, X. Shi, I. Seskar, and A. Gosain, "Network coding as a WiMAX link reliability mechanism," in *Proc. Int. Workshop on Multiple Access Commun., Springer LNCS, Vol. 7642*. Springer-Verlag Berlin Heidelberg, 2012, pp. 1–12.

[92] T. D. Assefa, K. Kralevska, and Y. Jiang, "Performance analysis of LTE networks with random linear network coding," in *Proc. IEEE Int, Convention on Inform. and Commun. Techn., Electr. and Microelectr. (MIPRO)*, 2016, pp. 601–606.

[93] V. Patil, S. Gupta, and C. Keshavamurthy, "An enhanced network coding based MAC optimization model for QoS oriented multicast transmission over LTE networks," *Int. J. Computer Science and Information Security*, vol. 14, no. 12, pp. 843–851, Dec. 2016.

[94] A. Tassi, F. Chiti, R. Fantacci, and F. Schoen, "An energy-efficient resource allocation scheme for RLNC-based heterogeneous multicast communications," *IEEE Commun. Letters*, vol. 18, no. 8, pp. 1399–1402, Aug. 2014.

[95] Y. Li, W.-Y. Chan, and S. D. Blostein, "Systematic network coding for transmission over two-hop lossy links," in *Proc. IEEE Biennial Symposium on Communications (QBSC)*, 2014, pp. 213–217.

[96] M. Kwon and H. Park, "Analysis on decoding error rate of systematic network coding," in *Proc. IEEE Int. Conf. on Consumer Electr. (ICCE)*, 2017, pp. 258–259.

[97] H. Shin and J.-S. Park, "Optimizing random network coding for multimedia content distribution over smartphones," *Multimedia Tools and Applications, in print*, pp. 1–17, 2017.

[98] S. Wunderlich, J. Cabrera, F. H. Fitzek, and M. Reisslein, "Network coding in heterogeneous multicore IoT nodes with DAG scheduling of parallel matrix block operations," *IEEE Internet of Things Journal, in print*, 2017.

[99] M. V. Pedersen, J. Heide, and F. H. Fitzek, "Kodo: An open and research oriented network coding library," in *Proc. Networking Workshops, Springer LNCS, Vol. 6827*. Springer, Berlin, Heidelberg, 2011, pp. 145–152.

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

[101] J. Heide, M. V. Pedersen, and F. H. Fitzek, "Decoding algorithms for random linear network codes," in *Proc. Int. Conf. on Research in Networking*. Springer, Heidelberg, 2011, pp. 129–136.

[102] S. Feizi, D. E. Lucani, C. W. Sørensen, A. Makhdoumi, and M. Médard, "Tunable sparse network coding for multicast networks," in *Proc. IEEE Int. Symp. on Network Coding (NetCod)*, 2014, pp. 1–6.

[103] C. W. Sorensen, A. S. Badr, J. A. Cabrera, D. E. Lucani, J. Heide, and F. H. Fitzek, "A practical view on tunable sparse network coding," in *Proc. of VDE European Wireless*, 2015, pp. 1–6.

[104] A. Tassi, I. Chatzigeorgiou, and D. E. Lucani, "Analysis and optimization of sparse random linear network coding for reliable multicast services," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 285–299, Jan. 2016.

PLACE PHOTO HERE

Sreekrishna Pandi is a Ph.D. researcher at the Deutsche Telekom Chair of Communication Networks at TU Dresden. He has been working at the chair since January 2015. Born in the city of Chennai (India), he received his Bachelor of Engineering degree in the field of electronics and instrumentation engineering from Anna University in 2013. Right after graduation, he moved to Dresden, Germany to pursue his Masters studies in the field of Nano-electronic systems. He received his Masters degree from the Technical University of Dresden in October 2015.

PLACE PHOTO HERE

Frank Gabriel is a Ph.D. researcher at the Deutsche Telekom Chair of Communication Networks at TU Dresden. He received his Dipl.-Inf. degree in Computer Science from the Technical University Chemnitz, Germany, in 2011.

PLACE PHOTO HERE

Juan A. Cabrera is pursuing his Ph.D. degree at the Deutsche Telekom Chair of Communication Networks at the Technical University Dresden, Germany. He received his M.Sc. in Wireless Communication Systems from Aalborg University, Denmark, in 2015, and he obtained his B.Sc. degree in Electronics Engineering from Simon Bolivar University, Venezuela, in 2013. He is specially interested in the research areas of network coding, fog computing, distributed storage systems, and mobile edge cloud solutions.

PLACE PHOTO HERE

Simon Wunderlich is currently pursuing his Ph.D. degree in Electrical Engineering at the Technical University Dresden, Germany. He received his Dipl.-Inf. degree in Computer Science at Chemnitz Technical University, Germany, in 2009. He is co-author of the Wi-Fi Mesh software B.A.T.M.A.N. Advanced.

PLACE PHOTO HERE

Martin Reisslein (A'96-S'97-M'98-SM'03-F'14) is a Professor in the School of Electrical, Computer, and Energy Engineering at Arizona State University (ASU), Tempe. He received the Ph.D. in systems engineering from the University of Pennsylvania in 1998. He currently serves as Associate Editor for the *IEEE Transactions on Mobile Computing*, the *IEEE Transactions on Education*, and *IEEE Access* as well as *Computer Networks* and *Optical Switching and Networking*. He is Associate Editor-in-Chief for the *IEEE Communications Surveys & Tutorials* and chairs the steering committee of the *IEEE Transactions on Multimedia*.

PLACE PHOTO HERE

Frank H. P. Fitzek is a Professor and head of the Deutsche Telekom Chair of Communication Networks at the Technical University Dresden, Germany, coordinating the 5G Lab Germany. He received his diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology – Rheinisch-Westfälische Technische Hochschule (RWTH) – Aachen, Germany, in 1997 and his Ph.D. (Dr.-Ing.) in Electrical Engineering from the Technical University Berlin, Germany in 2002 and became Adjunct Professor at the University of Ferrara, Italy in the same year. In 2003 he joined Aalborg University as Associate Professor and later became Professor. Dr. Fitzek co-founded several start-up companies starting with acticom GmbH in Berlin in 1999. He was selected to receive the NOKIA Champion Award several times in a row from 2007 to 2011. In 2008 he was awarded the Nokia Achievement Award for his work on cooperative networks. In 2011 he received the SAPERE AUDE research grant from the Danish government and in 2012 he received the Vodafone Innovation prize. In 2015 he was awarded the honorary degree "Doctor Honoris Causa" from Budapest University of Technology and Economy (BUTE). 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.