Analysis and Optimization of Sparse Random Linear Network Coding for Reliable Multicast Services

Andrea Tassi, Ioannis Chatzigeorgiou and Daniel E. Lucani

***Abstract*—Point-to-multipoint communications are expected to play a pivotal role in next-generation networks. This paper refers to a cellular system transmitting layered multicast services to a multicast group of users. Reliability of communications is ensured via different Random Linear Network Coding (RLNC) techniques. We deal with a fundamental problem: the computational complexity of the RLNC decoder. The higher the number of decoding operations is, the more the user's computational overhead grows and, consequently, the faster the battery of mobile devices drains. By referring to several sparse RLNC techniques, and without any assumption on the implementation of the RLNC decoder in use, we provide an efficient way to characterize the performance of users targeted by ultra-reliable layered multicast services. The proposed modeling allows to efficiently derive the average number of coded packet transmissions needed to recover one or more service layers. We design a convex resource allocation framework that allows to minimize the complexity of the RLNC decoder by jointly optimizing the transmission parameters and the sparsity of the code. The designed optimization framework also ensures service guarantees to predetermined fractions of users. The performance of the proposed optimization framework is then investigated in a LTE-A eMBMS network multicasting H.264/SVC video services.**

***Index Terms*—Sparse network coding, multicast communication, ultra-reliable communications, green communications, mobile communication, resource allocation, LTE-A, eMBMS.**

I. Introduction

Among the major novelties likely to be implemented in next-generation networks, there is the possibility of providing services characterized by an availability level of almost 100%. In the literature, that emerging kind of services is usually referred to as *ultra-reliable services* [1]. The ultra-reliable way of conveying services is expected to be greatly useful in a plethora of applications, such as reliable cloud-connectivity, data harvesting from sensors, professional communications [2].

Among the possibilities, this paper refers to a system model where a Base Station (BS) transmits, in a multicast fashion, a Point-to-Multipoint (PtM) service to a Multicast Group (MG) of users. In particular, the multicast service is provided in an ultra-reliable way, hence, the service shall be received by predetermined fractions of users, and has to meet target temporal constraints. It is worth noting that the possibility of managing ultra-reliable multicast applications is pivotal, in any Professional Mobile Radio (PMR) standard [3]. Even though classic PMR standards, like Terrestrial Trunked Radio (TETRA) or Association of Public-Safety Communications Officials-Project 25 (APCO P25), refer to ad-hoc communication protocol stacks, the upcoming evolutions of those standards will rely on the 3GPP's Long Term Evolution-Advanced (LTE-A) standard and its extents [4]. As a result, next-generation PMR standards are expected to enable the deployment of PMR systems over pre-existing LTE-A networks.

In this paper, we consider a system model where the base station multicasts a scalable service composed by one *base layer* and multiple *enhancement layers*. The base layer provides a basic reconstruction quality that is gradually improved as one or more enhancement layers are progressively received. Because of the layered nature of the considered multicast service, it is natural to refer to service reliability constraints, which impose that at least a minimum number of users is able to recover predetermined sets of service layers, by a given temporal deadline. The layered service approach has been originally adopted in video communications [5]. However, as discussed in [1] and [6], the same principle is likely to go beyond the traditional boundaries of multimedia communications and be applied in other fields in order to achieve an *analog-like* service degradation.

Because of the ultra-reliable nature of the considered multicast service, users are required to acknowledge to the base station when they successfully recovered one or more service layers. Even though there exists Automatic Repeat-reQuest (ARQ) [7] and Hybrid ARQ error control protocols [8] suitable for PtM communications, the protocol complexity and the required amount of feedback quickly become intractable as the number of users increases. For these reasons, the reliability of PtM communications is ensured via Application Level-Forward Error Correction (AL-FEC) techniques based on Luby Transform (LT) or low-density parity-check codes. However, as noted in [9], these kind of codes require large block lengths to operate close to their capacity, and that could potentially be an issue, in the case of multimedia communications. In addition, the most recent evolutions of LT codes [10] usually rely on fixed degree distribution functions and, hence, the code sparsity cannot be optimized on-demand. To this end,

This work is part of the R2D2 project, which is supported by EPSRC under Grant EP/L006251/1, and by the TuneSCode project (No. DFF $1335-00125$) granted by the Danish Council for Independent Research.

A. Tassi was with the School of Computing and Communications, Lancaster University, UK. He is now with the Department of Electrical and Electronic Engineering, University of Bristol, UK (e-mail: a.tassi@bristol.ac.uk).

I. Chatzigeorgiou is with the School of Computing and Communications, Lancaster University, Lancaster, UK (e-mail: i.chatzigeorgiou@lancaster.ac.uk).

D. E. Lucani is with the Department of Electronic Systems, Aalborg University, Aalborg, DK (e-mail: del@es.aau.dk).

in order to mitigate those issues, our system model ensures reliability of multicast communications, via Random Linear Network Coding (RLNC) techniques [11], [12].

Given a source message of k source packets to be multicast, the RLNC principle generates and multicasts a stream of coded packets, where each of them is obtained as a linear combination of multiple source packets. A user recovers the source message as soon as it collects a number of linearly independent coded packets that is equal to k. RLNC schemes have been used in several wireless settings as a versatile solution for reliable service delivery [13], [14]. Among the literature contributions, M. Xiao *et al.* [15] refer to a system model where nodes are connected by a network that can be represented by a Direct Acyclic Graph (DAG); that network consists of one source node and several sinks. In [15], the RLNC principle takes place at the network layer and allows intermediate nodes to combine several incoming data flows; reliability of coded packet transmissions is ensured via a channel code operating at the physical layer. The size of coded packets and the channel code rate are jointly optimized to minimize the end-to-end delay at the network layer. In addition, multiple resource allocation approaches have been proposed to improve the reliability of layered services via different RLNC implementations [16]–[18]. In particular, [16] considers a multi-hop directed acyclic graph network topology where a scalable service is multicast to multiple receivers. That paper proposes to optimize the communication rate on each link, in order to improve reliability. Channel erasures are further mitigated via classic FEC techniques. Similarly to [16], [17] deals with multi-hop network topologies and layered services. However, in that case, reliability of end-to-end communications is improved via a specific implementation of RLNC, which achieves a ladder-shaped global coding matrix. Differently than [16] and [17], [18] applies RLNC to populate a distributed caching system, kept by intermediate network nodes. The communication-ends can take advantage of that while they retrieve the desired scalable service, via a reduced number of Point-to-Point sessions. In contrast to [15]–[18], this paper refers to a typical cellular network topology, where the source node transmits streams of coded packets to a set of users in a multicast fashion. In other words, this paper adopts RLNC to improve reliability over a one-hop broadcast network and not as a way to improve the end-to-end communication throughout across a multi-hop network topology [12, Ref. [14]-[16]] and [19, Ref. [26]].

We observe that the application of RLNC to one-hop broadcast networks has been also discussed in [20] and [21]. In both cases, the broadcasting of a set of source packets is split into multiple stages. During the first stage all the source packets are broadcast by the source node, then, in the following stages, the source node and/or an intermediate relay node broadcast streams of coded packets. Both [20] and [21] focus on different forms of Instantly Decodable Network Codes, which generate coded packets in a deterministic fashion, based on multiple user feedback. As a consequence, we observe that the user uplink traffic can quickly become non-negligible as the number of users increases. Given that we will refer to a system model composed by a source node multicasting services to a multicast group composed by a potentially great number of users, it is not appropriate to refer to the strategies as in [20], [21]. On the other hand, we will refer to classic decodable RLNC strategies (as in [19]) that are characterized by a significantly smaller user feedback footprint.

Unfortunately, as noted in [22], [23], the flip side of the considered RLNC techniques is represented by the complexity of the decoding operations that depends, amongst other code parameters, by the length k of the source message. As noted in [24], [25], the decoding complexity problem can be partially mitigated by the systematic implementation of RLNC (SRLNC). However, in case of poor propagation conditions, the performance of SRLNC coincides with that of RLNC [26]. Obviously, the more the decoding complexity grows, the more the processing footprint increases and, hence, the battery of mobile devices discharges. For these reasons, this paper addresses the following fundamental question: *Is there a way to minimize the RLNC decoding complexity of ultra-reliable layered multicast communications without altering the decoder currently onboard mobile devices?*

We will answer the aforementioned research question by referring to multiple sparse RLNC techniques. As will be clear in the following section, let us intuitively define the *sparsity* of the code as the number of source packets that on average are involved in the generation process of each coded packet [22]. To the best of our knowledge, the general expression of the decoding complexity as a function of the source message length and the sparsity is unknown. However, the decoding complexity decreases as the source message gets shorter [23] and/or the sparsity increases [22]. Intuitively, as the sparsity increases, the information content of each coded packet decreases. Hence, the average number of coded packets needed to recover a source message increases as the sparsity grows [22]. That leads us to further refine our research question as follows: *Are there any optimized sparse RLNC strategies ensuring: (i) a reduced decoding complexity, and (ii) a recovery of the source message with an average number of coded packet transmissions, which is close or equal to that provided by non-sparse RLNC techniques?*

The first contribution of the paper is that of providing an efficient performance modeling of sparse non-systematic and systematic RLNC techniques via a *unified* theoretical framework. In particular, in Section II, we characterize the user performance in terms of the average number of coded packet transmissions needed to recover a given service layer. It is well known in the literature that an exact expression for the aforementioned performance index is unknown [27]–[30]. That is caused by the lack of an analytical formulation of the probability of generating a full-rank sparse random matrix over a finite field [30]. In order to mitigate the aforementioned issue, X. Li *et al.* [27], [28], proposed a pioneering approach for upper-bounding and lower-bounding the probability of generating at random a sparse non-singular random matrix, based on the zero pattern of the random matrix. Unfortunately, the validity of the resulting bounds has been proven only for large finite fields. Apart from that, those bounds cannot be efficiently incorporated into an optimization model meant to be solved on-demand, before starting the transmission of a service. In fact, the bound expressions involve nested sums where each term is a product of several binomial coefficients, which could not be practically derivable, in the case of large source message lengths (Section II). Furthermore, it is also not straightforward to formally prove the convexity of the bounds as in [27], [28], because their definitions involve several non-differentiable points.

For these reasons, we rely on the results presented in [29] and extended in [30]. However, in [29], [30], authors only provide a lower-bound of the probability that a sparse $(t+1) \times k$ matrix is full-rank, given that the first t rows are linearly independent, for $0 \leq t \leq (k-1)$. It is worth mentioning, that the aforementioned result was provided without referring to any communication system or coding strategy. By building upon that result, we provide an upper-bound for the average number of coded packet transmissions needed to recover a service layer, via an Absorbing Markov Chain (AMC) with reduced complexity. In particular, Section II-B will show how our performance modeling does not involve any explicit matrix inversion, which is a common and computationally costly step in AMC-based analysis. As will be clear in the following sections, that desirable feature is achieved because of: (i) the nature of the aforementioned probability lower-bound and, (ii) the way we defined the states of the proposed AMC model.

The second contribution of the paper is made in Section III, where we answer to our research question by building upon an efficient user performance characterization and proposing a resource allocation framework for ultra-reliable layered multicast services. The proposed framework aims to maximize the code sparsity associated to each service layer, and hence, the overall decoding complexity is minimized. The optimization goal is fulfilled by a joint optimization of both the code sparsity and the Modulation and Coding Schemes (MCSs) used for multicasting each service layer. In addition, given the layered nature of the transmitted services, the optimization constraints ensure that the desired number of service layers are recovered by predetermined fractions of users, with an average number of coded packet transmissions that is smaller than or equal to a target value. We prove that the proposed resource allocation framework is convex and can be easily solved. Finally, we remark that the proposed resource allocation framework applies for several sparse RLNC techniques, in a complete RLNC decoder-agnostic fashion.

Even though our analysis deals with a generic cellular system model, Section IV inspects the effectiveness of the proposed optimized sparse RLNC techniques by referring to a LTE-A communication network. We chose that particular communication standard for two main reasons: (i) LTE-A is likely to play a leading role in the early-stage deployment of next-generation networks [31], and (ii) LTE-A provides the support to handle PtM communications at the radio access and core network level, by means of the evolved Multimedia Broadcast Multicast Service (eMBMS) framework [32]. In the proposed performance investigation, we refer to a MG targeted by non-real time multimedia multicast services compressed according to the widely used H.264 video encoding standard. In particular, we referred to the scalable extension of H.264, called Scalable Video Coding (H.264/SVC) [33]. In line with the considered system model, an H.264/SVC video stream consists of several layers such that the enhancement layers improve the reconstruction quality provided by the base video layer. Finally, Section V summarizes the main findings of the paper.

II. System Model and Performance Characterisation

We consider a one-hop broadcast communication system, which is composed by one source node and a MG of U users (hereafter called *multicast users*). In order to improve the reliability of PtM communications, the source transmits data streams encoded according to the RLNC principle. As a consequence, the source node transmits streams of network-coded packets (henceforth referred to as *coded packets*) to the MG. For the sake of generality, we assume that the transmission of a PtM communication occurs over a set of orthogonal broadcast erasure subchannels. Each subchannel consists of basic resource allocation units called *resource blocks*.

As mentioned in the previous section, our main goal is to design a general optimized service-provisioning paradigm for ultra-reliable multicast services, with a reduced decoding computational complexity. The following section will also clarify that the proposed theoretical modeling (Section II-B) and the resource allocation procedure (Section III) are easily applicable to any cellular system capable of multicasting multiple data streams at the same time. However, in order to effectively map user Quality of Service (QoS) constraints onto typical system performance metrics (e.g., delay, packet error rate, etc.), we will refer to an OFDM-based multicarrier communication system.

In the considered physical layer, the downlink phase is organized in radio frames. Resource blocks forming each subchannel are transmitted in one or more radio frames. Each frame can be modeled as a frequency $\times$ time structure where the frequency and time domains are discretized into OFDM subcarriers and OFDM symbols, respectively. Each resource block occupies a fixed time interval ($\hat{\tau}_{\mathrm{RB}}$) and frequency band, i.e., each resource block spans a fixed number of OFDM symbols and OFDM subcarriers. Since multicast users may experience heterogeneous propagation conditions, user diversity is exploited by assuming that the subcarriers used in a resource block are selected at random among all the available ones [34]. We also assume that users are static or characterized by low mobility, hence, the user channel conditions are considered constant within a resource block.

Each coded packet is always mapped onto one resource block and transmitted by means of a specific MCS that is identified by an index, which can take M possible values. We denote by $p_u(m)$ the Packet Error Rate (PER) experienced by a multicast user u, and by $r(m)$ the number of information bits carried by one resource block, when the MCS with index m is in use. Let us consider two MCSs with indexes a and b, where $a < b$. In our system model we assume that the MCS with index a is characterized by a smaller modulation order and/or a lower channel code rate than b. For the same user propagation

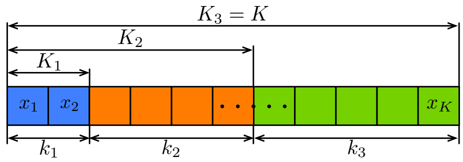


Fig. 1. Layered source message, in the case of $L = 3$.

conditions, we have $p_u(a) \leq p_u(b)$ and $r(a) < r(b)$. We also refer to a system where all the resource blocks belonging to the same subchannel shall adopt the same MCS. Coded packets associated with a PtM data service are transmitted via one or more broadcast erasure subchannels.

The source node transmits to the MG a *layered scalable* service consisting of one basic layer and $L-1$ enhancement layers. Each layer is characterized by different priority levels. The basic layer (also referred to as "layer 1") owns the highest priority, which decreases in the case of the enhancement layers (layers $2, \ldots, L$). In particular, layer L is characterized by the lowest priority. Because of that, it is natural to define the level of QoS achieved by a multicast user as the number of consecutive message layers, starting from the base layer, that can be recovered. Hence, a user shall achieve the QoS level ℓ, if *all* the layers $1, \ldots, \ell$ are successfully recovered. For instance, if a user successfully recovers message layers $\{1, 2, \ldots, \ell, \ell+2, \ell+3, \ldots, L\}$ then layers 2 to ℓ improve the information provided by layer 1. In that case, the QoS level achieved is equal to ℓ, and layers $\ell+2, \ldots, L$ do not provide any QoS improvements, as layer $\ell+1$ has not been received.

The considered multi-layer principle has been originally designed for video compression standards. In the case of H.264/SVC [33], it is possible to achieve different kinds of video scalability [5]. With the spatial scalability, the video frame resolution is gradually increased by each layer with the purpose to fit screens with different capabilities. In that case, the content provided by layer 1 allows a user, for instance, to recover a 352×288 px video stream. By following the same train of toughs, the spatial resolution can be boosted to 720×480 px and 1920×1080 px, by means of layers 1 and 2, and layers 1 to 3, respectively. It is worth mentioning that our analysis is generic enough to be applied to any layered scalable service that follows the previously mentioned hierarchical structure. It is beyond the scope of the paper to provide analytical and optimization frameworks dealing with the compression strategy used to generate a scalable service. For these reasons, the proposed analysis has been made independent of the way service layers are generated and the nature of the adopted service scalability.

As suggested in [12], [19], we model the transmitted service as a stream of information messages of the same size. The scalable nature of the service is reflected on each message. In particular, each message consists of L layers, where layer ℓ is a sequence of b_ℓ bits. We remark that coded packets associated with different message layers are transmitted by different subchannels. Therefore, the total number of occupied subchannels is L. In the rest of the paper, we will provide an analytical framework suitable for optimizing the transmission of each message and, hence, of the whole layered service.

Each layered message $\mathbf{x} = \{x_1, \ldots, x_K\}$ consists of K source packets, as shown in Fig. 1 for a 3-layer message. In particular, layer ℓ of $\mathbf{x}$ is defined by a fixed number k_ℓ of source packets, implying that $K = \sum_{\ell=1}^{L} k_\ell$. If the MCS adopted by the subchannel delivering coded packets of service layer ℓ is m_ℓ, the number of bits carried by each resource block will be equal to $r(m_\ell)$. Hence, we define $k_\ell = \lceil b_\ell / r(m_\ell) \rceil$. Without loss of generality we assume that the first source packets of $\mathbf{x}$ belong to the base layer ($\ell = 1$), and are progressively followed by packets defining the enhancement layers ($\ell = 2, \ldots, L$).

In the remaining part of the paper, we will characterize the performance of different network coding strategies. It will also become clear how the selection of MCS scheme and sparsity associated with each message layer can be jointly optimized.

A. Random Linear Network Coding Background

Let $K_\ell = \sum_{t=1}^{\ell} k_t$ be the number of source packets forming the first ℓ layers of a source message. In the classic implementation of RLNC, the source node linearly combines source packets $\{x_i\}_{i=K_{\ell-1}+1}^{K_\ell}$ forming message layer ℓ, in order to generate a stream $\{y_j\}_{j=1}^{n_\ell}$ of n_ℓ coded packets, where $y_j = \sum_{i=K_{\ell-1}+1}^{K_\ell} c_{j,i} \cdot x_i$. Each coding coefficient $c_{j,i}$ is uniformly selected at random over a finite field $\mathrm{GF}(q)$ of size q. The coding coefficients associated with y_j define the *coding vector* $\mathbf{c}_j = (c_{j,K_{\ell-1}+1}, \ldots, c_{j,K_\ell})$. Since each coding coefficient is obtained by the same Pseudo-Random Number Generator (PRNG), modern NC implementations are keen on representing $\mathbf{c}_j$ by the PRNG seed used to compute the first coding vector component $c_{j,K_{\ell-1}+1}$. The seed is transmitted along with the correspondent coded packet. Since each user is equipped by the same PRNG, it can incrementally recompute all the coding vector components, starting from the first one [11], [19]. The RLNC encoding process is then repeated for each message layer $\ell = 1, \ldots, L$. A multicast user can recover the source message layer ℓ, if it successfully receives k_ℓ linearly independent coded packets associated with that message layer.

Unlike classic RLNC, a coded packet stream obtained by SRLNC associated with layer ℓ generates k_ℓ *systematic packets* and one or more coded packets. The systematic packets are identical to the source packets $\{x_i\}_{i=K_{\ell-1}+1}^{K_\ell}$, while the coded packets are obtained as in the classic RLNC case. For the sake of the analysis, we define the coding vector associated with systematic packet i as a vector where: (i) the i-th component is equal to 1, and (ii) all the remaining components are equal to 0. For clarity, we will refer to a coding vector related to a systematic packet as *degenerate* coding vector in the rest of the paper. In our system model, we assume that users acknowledge to the source node, over a fully reliable channel, the successful recovery of a layer. Furthermore, the source node transmits a message layer until a predetermined fraction of multicast users has recovered it. Obviously, as will become clear in Section III, the transmission of each layer shall meet a temporal constraint.

The sparse versions of both the classic (S-RLNC) and systematic implementation of RLNC (S-SRLNC) are obtained as follows. Each component $c_{j,i}$ of a non-degenerate coding vector associated with source message layer ℓ is independently and identically distributed as follows [28]:

$$\Pr\left(c_{j,i}=v\right)=\begin{cases} p_\ell & \text{if } v=0\\ \dfrac{1-p_\ell}{q-1} & \text{if } v\in \mathrm{GF}(q)\setminus\{0\}\end{cases} \quad (1)$$

where p_ℓ, for $0<p_\ell<1$, is the probability of having $c_{j,i}=0$. The event $c_{j,i}\neq 0$ occurs with probability $1-p_\ell$. We remark that the average number of source packets involved in the generation of a non-degenerate coded packet, i.e., the *sparsity* of the code, can be controlled by tuning the value of p_ℓ, for any $\ell=1,\ldots,L$.

Since coding vectors are generated at random, there is the possibility of generating coding vectors where each coding coefficient is equal to 0. From a system implementation perspective, all-zero coded packets should be discarded and not transmitted. On the other hand, in the literature dealing with the performance characterization of RLNC, it is common to include the transmission of all-zero coded packets [35], [36]. In that way, the performance modeling is tractable and keeps a higher degree of generality. The same principle is adopted in this and the following sections. However, Section IV-A will show how the proposed analytical modeling can be applied to a practical communication system where all-zero coded packets are not transmitted.

In order to establish a link between the coding schemes presented in [12] and those discussed in this paper, the following sections will deal with the Non-Overlapping Window (NOW-RLNC) and the systematic NOW-RLNC strategies. We observe that the exact performance model of the Expanding Window RLNC (EW-RLNC) strategy is unknown, even for the non-sparse case. In fact, [12] proposes an upper-bound to the probability of recovering a source message, when the EW-RLNC is used. Since the reasoning behind that bound relies on a well-known result of classic non-sparse RLNC [37], its extension to the sparse case is not trivial. For these reasons, the sparse implementation of EW-RLNC is still an open research issue.

B. Markovian Modelling for Delay Performance

In this paper, user performance will be expressed in terms of the average number of coded packet transmissions after which a user u achieves a predetermined QoS level. For this reason, in the remainder of the section, we focus on user u and model the recovery of message layer ℓ as a Markovian process. In particular, the user decoding process is modeled via an AMC.

Let $\mathbf{C}_u$ be a matrix associated with the user u consisting of k_ℓ columns and variable number of rows. As user u successfully receives a coded packet associated with layer ℓ, the corresponding coding vector is extracted and added, as a new row, into matrix $\mathbf{C}_u$. Assume u already received $n_\ell \geq k_\ell$ coded packets, i.e., $\mathbf{C}_u$ is a $n_\ell \times k_\ell$ matrix. User u recovers layer ℓ when the rank of $\mathbf{C}_u$, denoted by $\mathrm{rank}(\mathbf{C}_u)$, is equal to k_ℓ or equivalently when the defect of the matrix, defined as $\mathrm{def}(\mathbf{C}_u)=k_\ell-\mathrm{rank}(\mathbf{C}_u)$, is zero. For these reasons, we define a state of the user AMC as follows.

Definition 2.1: The AMC associated with user u and message layer ℓ is in state $s_i^{(u,\ell)}$, if $\mathrm{def}(\mathbf{C}_u)=i$, for $i=0,\ldots,k_\ell$.

At first, when user u has not received any coded packet or coded packets associated with zero-coding vectors, the defect of $\mathbf{C}_u$ is k_ℓ, and hence, the AMC is in state $s_{k_\ell}^{(u,\ell)}$. The defect progressively decreases, i.e., the index of the AMC state decreases, as new linearly independent coded packets are received. As a consequence, in the case of layer ℓ, we have that the AMC consists of $k_\ell+1$ states. Furthermore, in order to define the probability transition matrix of the user AMC, we summarize here the proof of the following lemma, presented in [29, Theorem 6.3].

Lemma 2.1 ([29, Theorem 6.3]): Assume that matrix $\mathbf{C}_u$ consists of $(t+1)\times k_\ell$ elements, for $0<t\leq(k_\ell-1)$, and assume that t out of $t+1$ rows are linearly independent. The probability $\mathrm{P}_{\ell,t}$ that matrix $\mathbf{C}_u$ is not full-rank admits the following upper-bound:

$$\mathrm{P}_{\ell,t}\leq\left[\max\left(p_\ell,\frac{1-p_\ell}{q-1}\right)\right]^{k_\ell-t}. \quad (2)$$

Proof: Without loss of generality, assume that the first t rows of $\mathbf{C}_u$, denoted by $\mathbf{C}_{u,1},\ldots,\mathbf{C}_{u,t}$, are linearly independent. By resorting to basic row-wise operations, it is possible to transform $\mathbf{C}_u$ such that the first t rows and columns of $\mathbf{C}_u$ define the $t\times t$ identity matrix. Consequently, the first t rows of the transformed $\mathbf{C}_u$ generate the same vector space defined by $\mathbf{C}_{u,1},\ldots,\mathbf{C}_{u,t}$. The probability that $\mathbf{C}_u$ is not full-rank entirely depends on the last $k_\ell-t$ components of the last row $\mathbf{C}_{u,t+1}$ of $\mathbf{C}_u$. Hence, the probability that $\mathbf{C}_{u,t+1}$ does not belong to the vector space defined by $\mathbf{C}_{u,1},\ldots,\mathbf{C}_{u,t}$ is at least $1-\max\left(p_\ell,\frac{1-p_\ell}{q-1}\right)^{k_\ell-t}$. That completes the proof. ■

Because of (1), the exact QoS characterization is a challenging task [28]. In particular, to the best of our knowledge, the exact expression of $\mathrm{P}_{\ell,t}$ is not known. In the rest of the paper, owing to the lack of the exact expression of $\mathrm{P}_{\ell,t}$, we use (2) to approximate $\mathrm{P}_{\ell,t}$, that is

$$\mathrm{P}_{\ell,t}\cong\left[\max\left(p_\ell,\frac{1-p_\ell}{q-1}\right)\right]^{k_\ell-t}. \quad (3)$$

The following remark is immediate from (2) and (3).

Remark 2.1: If $p_\ell=q^{-1}$, each non-degenerate coding vector is equiprobable, for a given value of k_ℓ. Hence, a coding vector belongs to the vector space generated by t linearly independent coding vectors with probability $\mathrm{P}_{\ell,t}=q^t/q_\ell^k$. This result has been discussed in the literature [37] but is clearly not applicable to the sparse case, in contrast to (3). It is worth mentioning that the considered approximation (3) collapses to the exact expression of $\mathrm{P}_{\ell,t}$ and, hence, the relation $\mathrm{P}_{\ell,t}=\left[\max\left(p_\ell,(1-p_\ell)/(q-1)\right)\right]^{k_\ell-t}=q^t/q^{k_\ell}$ holds, for $p_\ell=q^{-1}$.

From (3), the transition probability matrix describing the AMC associated with user u and message layer ℓ can be derived by the following lemma.

Lemma 2.2: Assume layer ℓ is transmitted over a subchannel which adopts the MCS with index m. The probability

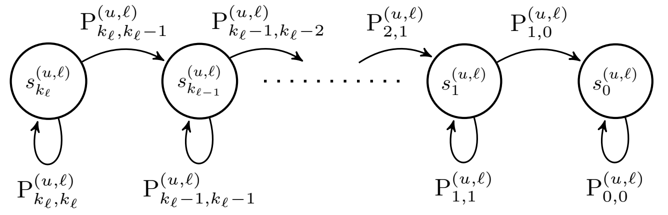


Fig. 2. State transition diagram for the AMC associated with user u and message layer ℓ.

$\mathrm{P}_{i,j}^{(u,\ell)}$ of moving from state $s_i^{(u,\ell)}$ to state $s_j^{(u,\ell)}$ is

$$\mathrm{P}_{i,j}^{(u,\ell)} = \begin{cases} (1-\mathrm{P}_{\ell,k_\ell-i})[1-p_u(m)] & \text{if } i-j=1 \\ \mathrm{P}_{\ell,k_\ell-i}[1-p_u(m)]+p_u(m) & \text{if } i=j \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Proof: Since the user AMC is in state $s_i^{(u,\ell)}$, user u has collected $k_\ell - i$ linearly independent coded packets, i.e., $\mathrm{rank}(\mathbf{C}_u) = k_\ell - i$. As a new coded packet associated with layer ℓ is transmitted, we have just two possibilities:

- The rank of $\mathbf{C}_u$ is increased to $k_\ell - i + 1$ - The coded packet is successfully received with probability $1 - p_u(m)$, and it is linearly independent of the previously received coded packets with probability $(1-\mathrm{P}_{\ell,k_\ell-i})$. This event occurs with a probability equal to $(1-\mathrm{P}_{\ell,k_\ell-i})[1-p_u(m)]$.
- The rank of $\mathbf{C}_u$ does not change - That may occur because the coded packet is not successfully received or because it is linearly dependent of the previously received coded packets. This event occurs with a probability equal to $\mathrm{P}_{\ell,k_\ell-i}[1-p_u(m)]+p_u(m)$.

■

From (29), we also understand that the probability of moving from state $s_0^{(u,\ell)}$ to another state is zero. Hence, $s_0^{(u,\ell)}$ represents the so-called *absorbing* state of the AMC. All the remaining states $s_1^{(u,\ell)}, \ldots, s_{k_\ell}^{(u,\ell)}$ are commonly referred to as *transient* states [38]. The state transition diagram of the resulting AMC can be represented as reported in Fig. 2.

From Lemma 2.2, it directly follows that the $(k_\ell+1)\times(k_\ell+1)$ transition matrix $\mathbf{T}^{(u,\ell)}$ describing the AMC of user u and associated with layer ℓ has the following structure in its *canonical form* [38]:

$$\mathbf{T}^{(u,\ell)} \doteq \left[\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{R}^{(u,\ell)} & \mathbf{Q}^{(u,\ell)} \end{array}\right], \quad (5)$$

where $\mathbf{Q}^{(u,\ell)}$ is the $k_\ell \times k_\ell$ transition matrix modeling the AMC process as long as it involves only transient states. The term $\mathbf{R}^{(u,\ell)}$ is a column vector of k_ℓ elements which lists all the probabilities of moving from a transient to the absorbing state. From [38, Theorem 3.2.4], let define matrix $\mathbf{N}^{(u,\ell)}$ as

$$\mathbf{N}^{(u,\ell)} = \sum_{t=0}^{\infty} \left(\mathbf{Q}^{(u,\ell)}\right)^t = \left[\mathbf{I} - \mathbf{Q}^{(u,\ell)}\right]^{-1}. \quad (6)$$

Element $\mathrm{N}_{i,j}^{(u,\ell)}$ at the location (i,j) of matrix $\mathbf{N}^{(u,\ell)}$ defines the average number of coded packet transmissions required for the process transition from state $s_i^{(u,\ell)}$ to state $s_j^{(u,\ell)}$, where both $s_i^{(u,\ell)}$ and $s_j^{(u,\ell)}$ are transient states. In particular, from Lemma 2.2, the following theorem holds

Theorem 2.1 ([38, Theorem 3.3.5]): If the AMC is in the transient state $s_i^{(u,\ell)}$, the average number of coded packet transmissions needed to get to state $s_0^{(u,\ell)}$ is

$$\tau_i^{(u,\ell)} = \begin{cases} 0 & \text{if } i=0 \\ \displaystyle\sum_{j=1}^{i} \mathrm{N}_{i,j}^{(u,\ell)} & \text{if } i=1,\ldots,k_\ell. \end{cases} \quad (7)$$

From (7) and Theorem 2.1, we prove the following corollaries.

Corollary 2.1: In the case of S-RLNC, the average number $\tau_{\text{S-RLNC}}^{(u,\ell)}$ of coded packets transmissions needed by user u to recover the source message layer ℓ is $\tau_{\text{S-RLNC}}^{(u,\ell)} = \tau_{k_\ell}^{(u,\ell)}$.

Proof: When the source node transmits the very first coded packet, user u is in state $s_{k_\ell}^{(u,\ell)}$. That follows from the fact that the source node has not previously transmitted any coded packets, and, hence, $\mathrm{rank}(\mathbf{C}_u)$ is always equal to 0. ■

We remark that, in the case of S-SRLNC transmission, at the end of the systematic phase, user u may have collected one or more source packets, implying that $\mathrm{def}(\mathbf{C}_u)$ may be smaller than k_ℓ. In particular, if $\mathrm{def}(\mathbf{C}_u) < k_\ell$, the AMC will start from any of the states $s_0^{(u,\ell)}, \ldots, s_{k_\ell-1}^{(u,\ell)}$.

Corollary 2.2: Consider S-SRLNC. If systematic and non-systematic coded packets associated with source message ℓ are transmitted by means of the MCS with index m, the considered average number $\tau_{\text{S-SRLNC}}^{(u,\ell)}$ of systematic and coded packet transmissions needed to recover layer ℓ is

$$\tau_{\text{S-SRLNC}}^{(u,\ell)} = \sum_{i=0}^{k_\ell} \pi_i^{(u,\ell)} \left(k_\ell - i + \tau_i^{(u,\ell)}\right) \quad (8)$$

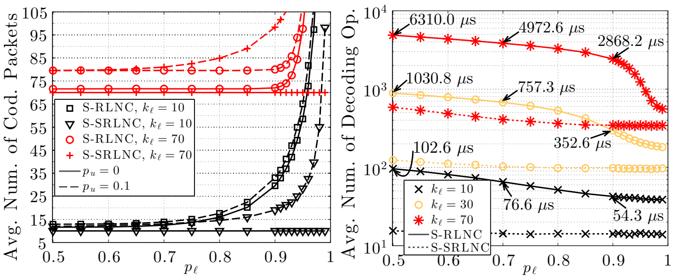
where $\pi_i^{(u,\ell)}$ is the probability that the process associated with user u starts from state $s_i^{(u,\ell)}$, given by

$$\pi_i^{(u,\ell)} = \binom{k_\ell}{i} p_u(m)^i \left[1-p_u(m)\right]^{k_\ell - i}, \quad i=0,\ldots,k_\ell. \quad (9)$$

Proof: Assume that u collects $k_\ell - i$ out of k_ℓ systematic packets. Hence, matrix $\mathbf{C}_u$ consists of $k_\ell - i$ linearly independent rows and, hence, the user AMC is in state $s_i^{(u,\ell)}$. In that case, from (7), we have that layer ℓ is recovered, on average, after $k_\ell - i + \tau_i^{(u,\ell)}$ packet transmissions, namely, $k_\ell - i$ systematic packets plus $\tau_i^{(u,\ell)}$ coded packets. At the end of the systematic packet transmission phase, the AMC is in state $s_i^{(u,\ell)}$ with probability $\binom{k_\ell}{i} p_u(m)^i \left[1-p_u(m)\right]^{k_\ell-i}$, for $i=0,\ldots,k_\ell$. Hence, the value of $\tau_{\text{S-SRLNC}}^{(u,\ell)}$ is obtained by simply averaging $k_\ell - i + \tau_i^{(u,\ell)}$ with the appropriate probability value of $\pi_i^{(u,\ell)}$, for $i=0,\ldots,k_\ell$, as provided in (8). ■

III. Sparse RLNC Optimization: Motivations and Resource Allocation Models

Among the most effective ways of decreasing the computational complexity of the RLNC decoding operations, we consider the reduction of the number of source packets, and the increase of the sparsity of the non-degenerate coding vectors per source message layer. As discussed in Section II, we remark that as the MCS index m_ℓ used to transmit layer ℓ increases, the number $r(m_\ell)$ of useful bits carried by one resource block or, equivalently, forming a coded packet, is



(a) No. of coded packet transmissions (b) No. of decoding operations

Fig. 3. Average number of coded packet transmissions and decoding operations, for $q = 2$. With regards the S-SRLNC scheme, the average number of decoding operations have been obtained by considering $p_u = 0.1$.

likely to increase. Given that coded and source packets have the same bit size, the value of k_ℓ is likely to decrease when m_ℓ increases. However, as m_ℓ increases, user PER related to the reception of subchannel ℓ is likely to increase, i.e., the fraction of multicast users regarding the reception of subchannel ℓ as acceptable is likely to decrease.

It is worth noting that both the value of k_ℓ and the probability p_ℓ of selecting a coding coefficient equal to zero determine the average number of coded packet transmissions and the average number of decoding operations needed to recover layer ℓ. With regards to the first aspect, Fig. 3a shows the value of $\tau^{(u,\ell)}_{\text{S-RLNC}}$ and $\tau^{(u,\ell)}_{\text{S-SRLNC}}$ as a function of p_ℓ, for $q = 2$, $k_\ell = \{10, 70\}$ and a packet error probability $p_u = \{0, 0.1\}$, when S-RLNC or S-SRLNC is used. Curves have been obtained by computer simulations. More details about the simulation environment will be given in Section IV.

In the case of S-SRLNC, as discussed in Section II-A, coded packets are transmitted after the systematic packets. Obviously, if $p_u = 0$, there is no need of transmitting coded packets as all the systematic packets are successfully received. That explains the reason way $\tau^{(u,\ell)}_{\text{S-SRLNC}}$ is always equal to k_ℓ, for $p_u = 0$. On the other hand, as the value of p_u increases, the number of coded packets to be transmitted increases, as well. We also observe that, for the same value of p_u, $\tau^{(u,\ell)}_{\text{S-SRLNC}}$ is smaller than or equal to $\tau^{(u,\ell)}_{\text{S-RLNC}}$. That is given by the fact that, in the case of S-SRLNC, there is aways the possibility for a user to collect some systematic packets, which are obviously linearly independent.

Both with S-RLNC and S-SRLNC (for $p_u > 0$), we observe that if p_ℓ approaches 1, then the average number of packet transmissions needed to recover layer ℓ increases. That is given by the fact that, coding vectors tend to be composed by all-zero. In addition, for a given value of p_ℓ, as k_ℓ and/or p_u decrease, the value of $\tau^{(u,\ell)}_{\text{S-RLNC}}$ decreases.

Fig. 3b shows the measured average number of decoding operations $\epsilon^{(\ell)}_{\text{S-RLNC}}$ and $\epsilon^{(\ell)}_{\text{S-SRLNC}}$ needed to recover layer ℓ, in the S-RLNC and S-SRLNC case, respectively. Results are provided as a function of p_ℓ, for $k_\ell = \{10, 30, 70\}$. Obviously, $\epsilon^{(\ell)}_{\text{S-RLNC}}$ does not depend on the user PER but just on k_ℓ and p_ℓ. In this paper, we will only refer to the fundamental finite field operations[1] performed by a network coding decoder based on the Gaussian Elimination principle, which represent the most computationally intensive part of the decoding process [14]. In particular, the more p_ℓ increases, the more the coding matrix $\mathbf{C}_u$ becomes sparser, and, consequently, the Gaussian Elimination requires a smaller number of iterations [22]. That behavior is confirmed by Fig. 3b, $\epsilon^{(\ell)}_{\text{S-RLNC}}$ decreases not only when k_ℓ decreases but also when p_ℓ increases.

[1]Let a, b, c be three elements in $\text{GF}(q)$, we will consider the following operations: $a \cdot b$, $a + b$, $a - b$, $a + (b \cdot c)$ and $a - (b \cdot c)$.

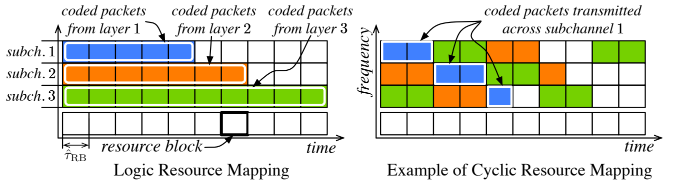


Fig. 4. Logic radio resource mapping (left-hand side) and an example of cyclic resource mapping (right-hand side), for $L = 3$.

In the case of S-SRLNC, the value of $\epsilon^{(\ell)}_{\text{S-SRLNC}}$ is indeed affected by the user PER. The more p_u increases, the more the number of successfully received systematic packets decreases and, the more the number of coded packets required to recover the layer increases. Hence, that corresponds to an increment in the value of $\epsilon^{(\ell)}_{\text{S-SRLNC}}$. In particular, Fig. 3b shows the value of $\epsilon^{(\ell)}_{\text{S-SRLNC}}$, for $p_u = 0.1$.

In the case of S-RLNC, in order to establish a link between the average number of decoding operations and the time needed to perform that number of decoding operations on a low-end device, Fig. 3b also reports the average processing time, for some (p_ℓ, k_ℓ) pairs. We have referred to a Gaussian Elimination-based decoder run on a Raspberry Pi Model B [39]. We note that there exists a linear relation between a reduction in the value of $\epsilon^{(\ell)}_{\text{S-RLNC}}$ and in the average processing time.

In the rest of the section, we will define a novel optimization model aiming to jointly optimize the sparsity of the code and the MCS index used to multicast each layer of the source message. The proposed model provides resource allocation solutions, which ensure that predetermined fractions of users recover sets of progressive layers, on average, within a given number of packet transmissions. In addition, the proposed model, at the same time, maximizes the sparsity and minimizes the total source message length.

A. Proposed Resource Allocation Models

From the *logic perspective*, we refer to the radio resource mapping presented in Fig. 4 (left-hand side). As the resource block is our fundamental resource allocation unit, the time duration of each radio frame shall be an integer multiple of the resource block time duration $\hat{\tau}_{\text{RB}}$. Every $\hat{\tau}_{\text{RB}}$ seconds, the source mode transmits *at most* one coded packet per-layer. We remark that the transmission of a message layer continues until the desired fraction of multicast users has recovered it (Section II-A). As a result, the average number of packet transmissions can be easily related to the average time needed to recover a layer.

Even though all the resource blocks forming the same subchannel are mapped onto time contiguous OFDM symbols, they could span a different set of OFDM subcarriers every $\hat{\tau}_{\text{RB}}$ seconds. For instance, subchannels could cyclically span different frequency sub-bands, as shown in Fig. 4 (right-hand side). In that way, the transmission of the same subchannel across the same set of OFDM subcarriers is avoided. Hence, users experiencing poor channel conditions across specific OFDM subcarriers will not always be prevented from receiving the same message layer.

In order to optimize m_ℓ and, indirectly, k_ℓ, the knowledge of the user propagation conditions is required. Obviously, the exact propagation conditions are unknown to the source node. However, modern communications standards allow users to periodically provide feedback about their average channel conditions across the whole transmission band[2]. Generally, the PER experienced by u is considered *acceptable* if it is smaller than or equal to a threshold $\hat{p}$. In the rest of the paper, we will refer to the principle adopted by the LTE-A standard, where any user u provides as propagation condition feedback the greatest MCS index M_u such that $p_u(\mathrm{M}_u) \le \hat{p}$, defined as [32]:

$$\mathrm{M}_u = \{m \,|\, m \in [1, M] \wedge p_u(m) \le \hat{p} \wedge p_u(m+1) > \hat{p}\}. \quad (10)$$

For these reasons, if layer ℓ is transmitted with MCS index $m_\ell \le \mathrm{M}_u$, $p_u(m_\ell)$ will be equal to or smaller than $\hat{p}$. Given the "aggregate nature" of the user channel feedback, relation $p_u(\mathrm{M}_u) \le \hat{p}$ is to be considered valid across the whole system band. Hence, the notion of M_u is independent to the way subchannels are actually transmitted across each frame.

Owing to the lack of knowledge of the user PER, during the resource allocation phase, the source node approximates the user PER as

$$p_u(m_\ell) \cong \begin{cases} \hat{p} & \text{if } m_\ell \le \mathrm{M}_u \\ 1 & \text{otherwise.} \end{cases} \quad (11)$$

In the case of S-RLNC, the proposed Sparsity-Tuning (ST) resource allocation model is defined as follows:

$$\text{ST} \quad \max_{\substack{p_1,\ldots,p_L \\ m_1,\ldots,m_L}} \; \|\mathbf{p}\|_1 \quad (12)$$

$$\text{s.t.} \quad \sum_{u=1}^{U} \delta\left(\sum_{t=1}^{\ell} \tau_{\text{S-RLNC}}^{(u,t)} \le \sum_{t=1}^{\ell} \hat{\tau}_t\right) \ge \sum_{t=1}^{\ell} \hat{U}_t, \quad \ell = 1, \ldots, L \quad (13)$$

$$q^{-1} \le p_\ell < 1 \qquad \ell = 1, \ldots, L \quad (14)$$

$$m_\ell \in \{1, \ldots, M\} \qquad \ell = 1, \ldots, L \quad (15)$$

where objective function (12) maximizes the 1-norm of vector $\mathbf{p} = \{p_1, \ldots, p_L\}$, which can be equivalently expressed as $\sum_{\ell=1}^{L} p_\ell$. Term $\delta(t)$ is an indication function that is equal to 1 if statement t is true, otherwise it is equal to 0. Parameters $\hat{\tau}_\ell$ and $\hat{U}_\ell$ represent the maximum number of coded packet transmissions needed to recover (on average) message layer ℓ and the minimum number of users that shall recover layer ℓ, respectively. For these reasons, the left-hand side of constraint (13) represents the number of multicast users that can recover layers $1, \ldots, \ell$, on average, in at most $\sum_{t=1}^{\ell} \hat{\tau}_t$ coded packet transmissions. As a result, constraint (13) ensures that the number of multicast users achieve *QoS level* ℓ is at least equal to $\sum_{t=0}^{\ell} \hat{U}_t$. Since user u can only achieve QoS level ℓ if all the layers $1, \ldots, \ell$ have been recovered, it would be pointless to recover layer ℓ before layer $\ell - 1$. For the same reasons, there is no point in having situations where the fraction of users recovering layer ℓ is greater than the fraction of users recovering $\ell - 1$. Hence, it is reasonable to assume that the relations $\hat{U}_{\ell-1} \ge \hat{U}_\ell$ and $m_{\ell-1} \le m_\ell$ hold, for $\ell = 2, \ldots, L$. Furthermore, constraint (14) avoids both dense coding vectors (i.e., $p_\ell < q^{-1}$) and all-zero coding vectors (i.e., $p_\ell = 1$). Then constraint (15) remarks that variable m_ℓ can only take values in range $1, \ldots, M$. The ST problem can also be defined for the case of S-SRLNC by simply replacing in constraint (13) the term $\tau_{\text{S-RLNC}}^{(u,t)}$ with $\tau_{\text{S-SRLNC}}^{(u,t)}$. We observe that the selection of parameters $\hat{\tau}_\ell$ and $\hat{U}_\ell$, for $\ell = 1, \ldots, L$, allow the ultra-reliable service to be delivered, by meeting the Service Level Agreements (SLAs) between the service provider and the users. In our case, SLAs imposes the minimum fraction of users that shall achieve target QoS levels and the maximum time needed (on average) to do so.

Because of constraint (13), the ST problem presents vast coupling constraints among the whole set of optimization variables. In spite of the apparent optimization complexity, we will show that the ST problem can be efficiently solved, both in the case of S-RLNC and S-SRLNC, by decomposing it into subproblems of a reduced complexity. In order to do so, it is worth solving the Layer Sparsity Maximization (LSM) problem associated with user u, MCS index m and layer ℓ. We will eventually refer to the LSM problem to solve the SM problem. In particular, the LSM problem is defined as follows:

$$\text{LSM-}(\ell, u, m) \quad \max_{p_\ell} \; p_\ell \quad (16)$$

$$\text{s.t.} \quad \tau_{\text{S-RLNC}}^{(u,\ell)} \le \hat{\tau}_\ell \quad (17)$$

$$q^{-1} \le p_\ell \le 1 \quad (18)$$

From Corollary 2.1, we have that $\tau_{\text{S-RLNC}}^{(u,\ell)}$ is defined as a sum of terms from matrix $\mathbf{N}^{(u,\ell)}$. In the following, we equivalently rewrite constraint (17) in order to avoid the explicit inversion of $\mathbf{I} - \mathbf{Q}^{(u,\ell)}$ in (6), and we prove the convexity of LSM-(ℓ, u, m).

We define the $k_\ell \times k_\ell$ matrix $\mathbf{W}^{(u,\ell)}$ as $\mathbf{W}^{(u,\ell)} = \mathbf{I} - \mathbf{Q}^{(u,\ell)}$. From (6), we have that $\mathbf{N}^{(u,\ell)} = (\mathbf{W}^{(u,\ell)})^{-1}$. Let $\mathrm{Q}_{i,j}^{(u,\ell)}$ be the (i,j)-th element of matrix $\mathbf{Q}^{(u,\ell)}$. From (29) and (5), we have that $\mathbf{Q}^{(u,\ell)}$ is a non-negative lower-triangular matrix with the following structure:

$$\mathbf{Q}^{(u,\ell)} = \begin{bmatrix} \mathrm{Q}_{1,1}^{(u,\ell)} & 0 & \cdots & 0 & 0 \\ \mathrm{Q}_{2,1}^{(u,\ell)} & \mathrm{Q}_{2,2}^{(u,\ell)} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \mathrm{Q}_{k_\ell,k_\ell-1}^{(u,\ell)} & \mathrm{Q}_{k_\ell,k_\ell}^{(u,\ell)} \end{bmatrix}. \quad (19)$$

Hence, for i and $j = 1, \ldots, k_\ell$, element (i,j) of $\mathbf{W}^{(u,\ell)}$ is

$$\mathrm{W}_{i,j}^{(u,\ell)} = \begin{cases} -(1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell-i}) & \text{if } i - j = 1 \\ (1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell-i}) & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \quad (20)$$

[2] 3GPP LTE and LTE-A standards refer to this kind of user channel feedback as wideband Channel Quality Indicators [32].

From (6), the following relation holds:

$$\mathbf{W}^{(u,\ell)} \cdot \mathbf{N}^{(u,\ell)} = \mathbf{I}. \tag{21}$$

Relation (21) defines a set of k_ℓ disjoint parametric systems of equations, where p_ℓ is the system parameter and the elements of $\mathbf{N}^{(u,\ell)}$ are the system unknowns. System s, for $s = 1, \ldots, k_\ell$, consists of $k_\ell - s + 1$ equations. In particular, the i-th equation of system s, for $i = s, \ldots, k_\ell$, is defined as:

$$\sum_{j=s}^{i} \mathrm{W}_{i,j}^{(u,\ell)} \, \mathrm{N}_{j,s}^{(u,\ell)} = \delta(i = s). \tag{22}$$

From (19), (20), the solution of system s can be expressed as

$$\mathrm{N}_{i,s}^{(u,\ell)} = \begin{cases} [(1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell - s})]^{-1} & \text{if } i = 1, \ldots, k_\ell \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

As a result we can prove the following lemma.

Lemma 3.1: The LSM-(ℓ, u, m) problem is convex. In addition, the optimum solution of the problem is the real root of

$$\sum_{i=0}^{k_\ell} [(1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell - i})]^{-1} - \hat{\tau}_\ell = 0, \tag{24}$$

which is greater than or equal to q^{-1} and smaller than 1.

Proof: From Corollary 2.1 and (23), we have that $\tau_{\text{S-RLNC}}^{(u,\ell)}$ can be equivalently rewritten as

$$\tau_{\text{S-RLNC}}^{(u,\ell)} = \sum_{i=0}^{k_\ell} [(1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell - i})]^{-1}. \tag{25}$$

Since we refer to the approximation as in (3), $\mathrm{P}_{\ell,k_\ell - i}$ is the non-negative power of a pointwise maximization of two convex functions. Hence, $\mathrm{P}_{\ell,k_\ell - i}$ is convex with respect to p_ℓ. Consider function $(1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell - i})$ of (25). Since $\mathrm{P}_{\ell,k_\ell - i}$ is convex, function $(1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell - i})$ is concave and, hence, $[(1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell - i})]^{-1}$ is convex. As a result, $\tau_{\text{S-RLNC}}^{(u,\ell)}$, expressed as in (25), is a non-negative weighted sum of convex functions, which is a convex function. For these reasons, it follows that the LSM-(ℓ, u, m) problem is convex [40]. From (25), we rewrite constraint (17) as $\sum_{i=1}^{k_\ell} [(1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell - i})]^{-1} \leq \hat{\tau}_\ell$. Because of the convexity of LSM-(ℓ, u, m), we have that the optimum solution of the problem is given by the real root of (24), which belongs to $[q^{-1}, 1)$. ■

The LSM-(ℓ, u, m) problem can be adapted to the S-SRLNC case by simply replacing constraint (17) with $\tau_{\text{S-SRLNC}}^{(u,\ell)} \leq \hat{\tau}_\ell$. The resulting optimization problem can be solved as follows.

Lemma 3.2: In the S-SRLNC case, the resulting LSM-(ℓ, u, m) problem is convex, and its optimal solution is the real root, greater than or equal to q^{-1} and smaller than 1, of the following equation:

$$\sum_{i=0}^{k_\ell} \pi_i^{(u,\ell)} (k_\ell - i) + \\ + \sum_{i=1}^{k_\ell} \pi_i^{(u,\ell)} \sum_{j=1}^{i} [(1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell - j})]^{-1} - \hat{\tau}_\ell = 0. \tag{26}$$

Proof: From Corollary 2.2 and (23), $\tau_{\text{S-SRLNC}}^{(u,\ell)}$ can be expressed as

$$\tau_{\text{S-SRLNC}}^{(u,\ell)} = \pi_0^{(u,\ell)} k_\ell + \sum_{i=1}^{k_\ell} \Bigg\{ \pi_i^{(u,\ell)} (k_\ell - i) + \\ + \pi_i^{(u,\ell)} \sum_{j=1}^{i} [(1 - p_u(m))(1 - \mathrm{P}_{\ell,k_\ell - j})]^{-1} \Bigg\}. \tag{27}$$

Likewise Lemma 3.1, $\tau_{\text{S-SRLNC}}^{(u,\ell)}$ is convex because it is the non-negative weighted sum of convex functions. Then the proof follows exactly the same reasoning as in the proof of Lemma 3.1. ■

Once more, consider the ST problem and the following remark.

Lemma 3.3: Constraint (13) of the ST problem can be equivalently rewritten as

$$\sum_{u=1}^{U} \delta \left(\tau_{\text{S-RLNC}}^{(u,\ell)} \leq \hat{\tau}_\ell \right) \geq \hat{U}_\ell, \qquad \ell = 1, \ldots, L. \tag{28}$$

or restated for the S-SRLNC case, in a similar way.

Proof: From Section III-A, relation $\tau_{\text{S-RLNC}}^{(u,t)} \leq \hat{\tau}_t$ shall hold, for at least $\hat{U}_t$ users. Hence, the complete statement of the argument of function $\delta(\cdot)$ in (13) is equivalent to the following system of inequalities

$$\begin{cases} \sum_{t=1}^{\ell} \tau_{\text{S-RLNC}}^{(u,t)} \leq \sum_{t=1}^{\ell} \hat{\tau}_t \\ \tau_{\text{S-RLNC}}^{(u,t)} \leq \hat{\tau}_t, & \text{for } t = 1, \ldots, \ell. \end{cases} \tag{29}$$

We observe that the first inequality is made redundant by the remaining ones. Hence, (13) can be rewritten as

$$\sum_{u=1}^{U} \delta \left(\bigwedge_{t=1}^{\ell} \tau_{\text{S-RLNC}}^{(u,t)} \leq \hat{\tau}_t \right) \geq \sum_{t=1}^{\ell} \hat{U}_t, \quad \text{for } \ell = 1, \ldots, L, \tag{30}$$

where the leftmost term still counts exactly the same number of users achieving QoS level ℓ as in (13). Consider layer t, it shall be received by at least $\hat{U}_t$ users, for $t = 1, \ldots, L$. Hence, the complete statement of (30), for a given ℓ, is

$$\begin{cases} \sum_{u=1}^{U} \delta \left(\bigwedge_{t=1}^{\ell} \tau_{\text{S-RLNC}}^{(u,t)} \leq \hat{\tau}_t \right) \geq \sum_{t=1}^{\ell} \hat{U}_t, \\ \sum_{u=1}^{U} \delta \left(\tau_{\text{S-RLNC}}^{(u,t)} \leq \hat{\tau}_t \right) \geq \hat{U}_t, & \text{for } t = 1, \ldots, \ell. \end{cases} \tag{31}$$

We remark that relations $\hat{U}_{\ell-1} \geq \hat{U}_\ell$ and $m_{\ell-1} \leq m_\ell$ hold, for $\ell = 2, \ldots, L$. In addition, from the considered PER model (11), we have that the set of users achieving QoS level ℓ entirely contains those achieving QoS levels $1, \ldots, \ell - 1$. Hence, the first inequality of (31) is made redundant by the following ones. That completes the proof. This proof can be similarly restated for the S-SRLNC case. ■

From Lemma 3.3, ST can be decomposed into L independent optimization problems ST-(1), …, ST-(L), where the ST-(ℓ) problem: (i) refers to the video layer ℓ, (ii) has the goal of maximizing p_ℓ, and (iii) refers to just the constraints of ST that are related to layer ℓ. ST-(ℓ) problem can be solved as follows.

Remark 3.1: From Lemmas 3.1 and 3.2, we have that $\tau^{(u,\ell)}_{\text{S-RLNC}}$ and $\tau^{(u,\ell)}_{\text{S-SRLNC}}$ are non-decreasing functions with respect to p_ℓ, for $q^{-1} \leq p_\ell < 1$. In addition, for a given value of p_ℓ, we remark that as m_ℓ increases, the value of $\tau^{(u,\ell)}_{\text{S-RLNC}}$ will decrease as well (Section II). Hence, ST-(ℓ) is solved by the pair (m_ℓ, p_ℓ) characterized by the greatest values of m_ℓ and p_ℓ such that relations $\tau^{(u,\ell)}_{\text{S-RLNC}} \leq \hat{\tau}_\ell$ or $\tau^{(u,\ell)}_{\text{S-SRLNC}} \leq \hat{\tau}_\ell$ hold, for at least $\hat{U}_\ell$ users. In particular, ST-(ℓ) can be solved by resorting to LSM problems as follows. For any $m_\ell = 1, \ldots, M$ and $\ell = 1, \ldots, L$, let $\mathcal{U}_{m_\ell}$ signify the set of users such that $\mathrm{M}_u \geq m_\ell$.

1. Let us solve LSM-(ℓ, u, m), for a user $u \in \mathcal{U}_{m_\ell}$ and $m = m_\ell$. Let p^*_{ℓ,m_ℓ} be the optimum solution of LSM-(ℓ, u, m_ℓ). If S-RLNC is in use then the value of p^*_{ℓ,m_ℓ} is derived as provided by Lemma 3.1. On the other hand, if S-SRLNC is in use then we will refer to Lemma 3.2, for the computation of p^*_{ℓ,m_ℓ}. Since $p_u(m)$ is approximated as in (11), the solution p^*_{ℓ,m_ℓ} will always be the same, for every user in $\mathcal{U}_{m_\ell}$.
2. For any $m_\ell = 1, \ldots, M$ such that $|\mathcal{U}_{m_\ell}| \geq \hat{U}_\ell$ and an optimum solution p^*_{ℓ,m_ℓ} exists, the pair $(m_\ell, p^*_{\ell,m_\ell})$ is an optimum solution of ST-(ℓ). Among the optimum solutions of problem ST-(ℓ), we choose the pair $(m_\ell, p^*_{\ell,m_\ell})$ associated with the greatest MCS index, i.e., we consider the solution that ensures the smallest value of k_ℓ (see Section II).

The process is repeated to solve any problem ST-(ℓ), for $\ell = 1, \ldots, L$ and, hence, to solve problem ST. We observe that, for a given value of m_ℓ, the pair $(m_\ell, p^*_{\ell,m_\ell})$ may not exist. That can happen because: (i) the value of $\hat{\tau}_\ell$ is too small and the average number of coded packet transmissions always exceed $\hat{\tau}_\ell$, for $q^{-1} \leq p_\ell < 1$, and/or (ii) the target user coverage $\hat{U}_\ell$ is too big (constraint (28) is not met), given the overall user propagation conditions and, hence, the MCSs that can be used in a considered scenario.

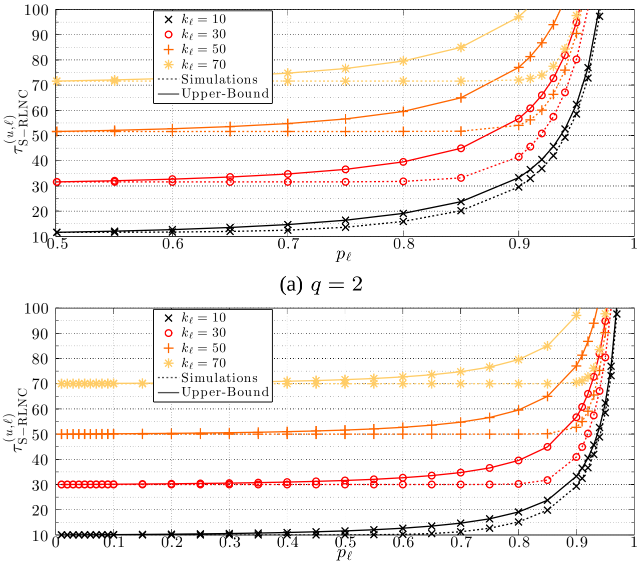
For these reasons and, in particular, from Lemmas 3.1, 3.2 and 3.3, it is immediate to prove the following theorem.

Theorem 3.1: Both in the S-RLNC and S-SRLNC cases, the resource allocation solution of ST problem derived by Remark 3.1, for any $\ell = 1, \ldots, L$, is optimal and characterized by the greatest MCS indexes, i.e., the derived optimal solution ensures the smallest values of k_ℓ, for $\ell = 1, \ldots, L$.

IV. Numerical Results

A. Assessment of the Performance Model

We recall from Section II-B that we mitigated the lack of an accurate expression of the probability $\mathrm{P}_{\ell,t}$ that a sparse random $(t+1) \times k_\ell$ matrix is not full-rank over GF(q), given that the first t rows are linearly independent. In particular, we upper-bounded the value of $\mathrm{P}_{\ell,t}$ by referring to the approximation in (3). Hence, the average user delay values $\tau^{(u,\ell)}_{\text{S-RLNC}}$ (Corollary 2.1) and $\tau^{(u,\ell)}_{\text{S-SRLNC}}$ (Corollary 2.2) are expected to be greater than or equal to the correspondent average user delay values obtained via computer simulations. In this paper, all



(a) $q = 2$

(b) $q = 2^8$

Fig. 5. Average number of coded packet transmissions vs. the average number of coded packet transmissions obtained by referring to the approximation as in (3), for $q = 2$ and 2^8.

the computer simulations rely on the encoders and decoders provided by the Kodo C++ network coding library [14].

Fig. 5 refers to a scenario, where a source message of $k_\ell \in \{10, 30, 50, 70\}$ source packets is transmitted to a user by means of S-RLNC, over a fully reliable channel (i.e., the user PER is equal to 0). In particular, Fig. 5a compares, for $q = 2$, the value of $\tau^{(u,\ell)}_{\text{S-RLNC}}$ as in Corollary 2.1 with that obtained by simulations, as a function of the probability p_ℓ of selecting a zero coding coefficient. Fig. 5b reports the same performance comparison, in the case of $q = 2^8$. Figs. 5a and 5b show that, for $p_\ell = q^{-1}$, simulation and our theoretical upper-bound of $\tau^{(u,\ell)}_{\text{S-RLNC}}$ overlap. In fact, from Remark 2.1, in that case, (3) no longer is an approximation. However, the gap between the theoretical upper-bound and simulation results increases, as p_ℓ becomes larger than q^{-1}.

Let us focus on S-RLNC such that $p_\ell \geq q^{-1}$, regardless of the value of q, we observe that the performance gap between the theoretical upper-bound and simulation results mainly depends only on the value of k_ℓ and p_ℓ. On the other hand, for large values of k_ℓ (such as, $k_\ell \geq 50$) and p_ℓ ($p_\ell \geq 0.93$), the value of the performance gap, normalized with respect to k_ℓ, is almost constant and equal to 0.53. In other words, the impact of q on the performance gap is not pivotal and, at the same time, it is mainly proportional to k_ℓ. Given that the simulation results reported in Section IV-B refer to values of k_ℓ and p_ℓ in the aforementioned ranges, that gives a clear upper-bound of the impact of our approximation onto the displayed performance, on a layer-basis. Since S-RLNC can be considered a special case of S-SRLNC, the aforementioned considerations also apply to the systematic case.

We observe that the theoretical upper-bound is no more than 33.4% higher than simulation results, in the considered cases. As one of the key aspects in our optimization framework is the enforcement of service coverage constraints, the adoption of the approximation as in (3) will indeed not violate those service constraints. Ideally, if the exact expression of $\tau^{(u,\ell)}_{\text{S-RLNC}}$

and $\tau_{\text{S-SRLNC}}^{(u,\ell)}$ were known, we would get ST solutions characterised by a greater level of sparsity. In addition, from an implementation perspective, it is not feasible to tabulate the exact values of $\tau_{\text{S-RLNC}}^{(u,\ell)}$ and $\tau_{\text{S-SRLNC}}^{(u,\ell)}$ as a function of p_ℓ and k_ℓ. In particular, we remark that the value of k_ℓ is given by the layer bit length and the adopted MCS, which cannot be determined in advance.

B. Performance Evaluation of the Proposed Resource Allocation Models

The performance of the proposed resource allocation modeling has been investigated in an LTE-A scenario composed by 19 base stations arranged in two concentric rings and centered on a *target base station*. Each base station manages three hexagonal sectors per cell. In addition, for the physical layer parameters, we referred to the 3GPP's benchmark Case 1 scenario [41], where base stations are characterized by an inter-site distance of 500 m. In order to meet the LTE-A physical layer constraints, each coded packet is mapped on resource blocks spanning a bandwidth of 540 kHz and 12 OFDM symbols (lasting for $\hat{\tau}_{\text{RB}} = 10$ ms). In accordance to a well documented best practice in the deployment of LTE-A networks [32], the reception of a resource block is regarded as acceptable when $\hat{p}$ is equal to 0.1. The reader who may want to have more details about the simulator and the considered low-level transmission parameters, can refer to [12, Section V]. Due to space limitations, all those details have been omitted.

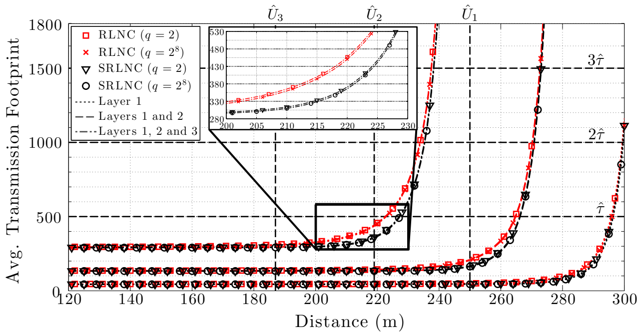
In our performance investigation, we referred to a network scenario where the target base station multicasts a layered video stream to a user MG, also known as Single Cell-eMBMS (SC-eMBMS) transmission mode. Furthermore, we considered a user distribution characterized by the maximum heterogeneity from the point of view of the channel conditions. In particular, we refer to a MG of $U = 80$ users that are regularly placed along the symmetry axis of one sector controlled by the target base station. The first user is 90 m apart from the center of the cell, and the distance between two consecutive users is 2 m.

In this section, we consider two different video sequences (Stream A and Stream B) of 30 s, compressed according to the H.264/SVC standard [33]:

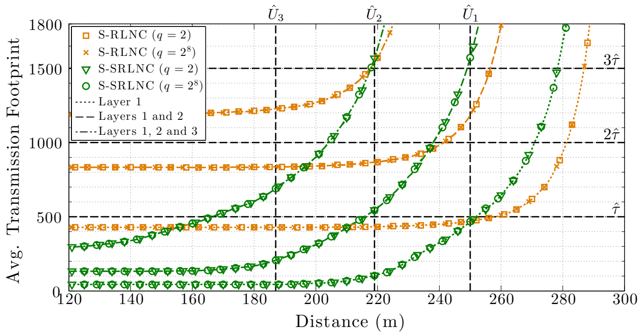
- Stream A [42] - is a $L = 3$ video trace characterized by $\{b_1, b_2, b_3\} = \{702, 4841, 20584\}$ KBytes per layer.
- Stream B [42] - is a $L = 4$ video trace such that $\{b_1, b_2, b_3, b_4\} = \{702, 2138, 6001, 19384\}$ KBytes per layer.

The video traces implement the coarse grain scalability principle, which is a form of spatial scalability such that the combination of consecutive layers enhances the frame resolution. In addition, both video traces belong to the database presented in [5], and developed for network performance evaluation purposes. The video traces have a resolution of 352×288, a Group of Picture size of 16 frames and a video frame rate of 30 fps.

In our numerical results, a 30 s video trace defines one layered source message. Each video layer has the same duration of the whole video trace. For simplicity, we impose that each



(a) RLNC and SRLNC



(b) S-RLNC and S-SRLNC

Fig. 6. Average transmission footprint $\tau^{(u,1:\ell)}$, expressed in terms of number of packet transmissions, in the case of Stream A, for $\ell = 1, \ldots, 3$, $q = 2$ and 2^8.

video layer shall be recovered by the same average number $\hat{\tau}$ of coded packet transmissions, i.e., $\hat{\tau}_1 = \hat{\tau}_2 = \ldots = \hat{\tau}_L = \hat{\tau}$. From constraint (13), we have that QoS level ℓ shall be achieved, on average, in $\sum_{t=1}^{\ell} \hat{\tau}_t = \ell\hat{\tau}$ coded packet transmissions. Since the time duration of each resource block is fixed, it is immediate to equivalently express $\hat{\tau}$ in seconds, denoted as "$\hat{\tau}$ (sec.)".

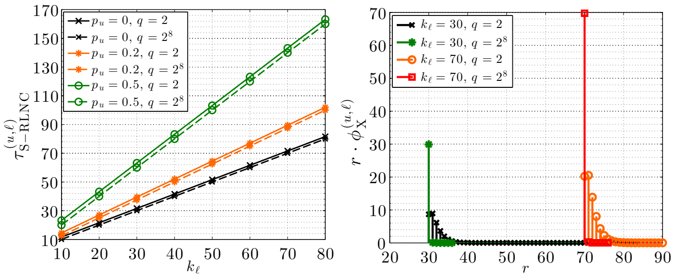
We compared the optimized version of S-RLNC and S-SRLNC (see Section III-A) against their non-sparse versions. In order to provide a fair comparison among the strategies, when either RLNC or SRLNC is used, the MCS indexes $m_1, \ldots, m_L$ associated to the transmission of each video layer are optimized such that the service constraints are met, for $p_\ell = q^{-1}$.

Let us define the average transmission footprint $\tau^{(u,1:\ell)}$ as

$$\tau^{(u,1:\ell)} = \begin{cases} \displaystyle\sum_{t=1}^{\ell} \tau_{\text{S-RLNC}}^{(u,t)}, & \text{for S-RLNC} \\ \displaystyle\sum_{t=1}^{\ell} \tau_{\text{S-SRLNC}}^{(u,t)}, & \text{for S-SRLNC.} \end{cases} \quad (32)$$

That definition can be easily extended to the non-sparse version of RLNC and SRLNC by considering a value of p_ℓ equal to q^{-1}. We remark that in the rest of this section, all the user performance investigation has been carried out via computer simulations. The approximated performance modeling of Section II-B is used only by the target base station during the resource allocation operations.

Figs. 6a and 6b show the value of $\tau^{(u,1:\ell)}$ provided by all the considered network coding schemes when Stream A is multicast, in the case of $q = \{2, 2^8\}$ and for $\hat{\tau}$ (sec.) equal to 0.5 s. Since users are regularly distributed along with a



(a) $\mathbb{E}[\text{X}]$ values (b) Terms $r \cdot \phi^{(u,\ell)}_{\text{X}}(r)$

Fig. 7. Expected value and scaled PMF of X, in the case of RLNC and for $q = 2$ and $p_\ell = 1/2$. The rightmost figure refers to case where $k_\ell = \{30, 70\}$ and $p_u = 0$.

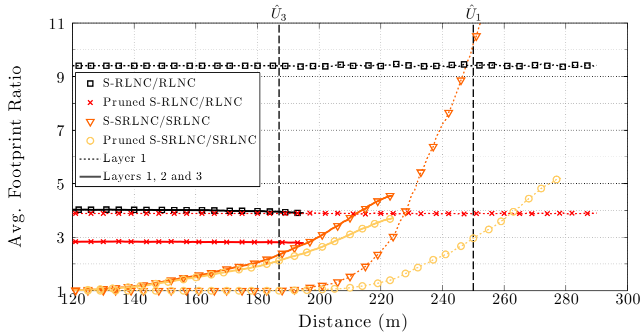
segment connecting the target base station with the cell edge, we have that: (i) $\tau^{(u,1:\ell)}$ can be equivalently expressed as a function of the distance form the center of the cell, and (ii) the target number of users $\hat{U}_\ell$ that shall receive video layer ℓ can be equivalently expressed in terms of distances from the center of the cell (vertical dashed lines in Figs. 6a and 6b). On the other hand, the horizontal dashed lines in Figs. 6a and 6b represents the maximum transmission footprint to achieve QoS level ℓ, namely, $\sum_{t=1}^{\ell} \hat{\tau}_t$.

From Fig. 6 and regardless of the network coding strategy in use, we observe that the values of $\tau^{(u,1:\ell)}$ when $q = 2$ are very close to those obtained when $q = 2^8$. In particular, the greatest performance gap is associated in the case of RLNC and it is smaller than 6 coded packets. In general, the performance differences between the case where $q = 2$ and the case $q = 2^8$ tend to vanish as we refer to the optimized S-RLNC or S-SRLNC strategies. The reasoning behind the aforementioned behaviour is given in the following remark.

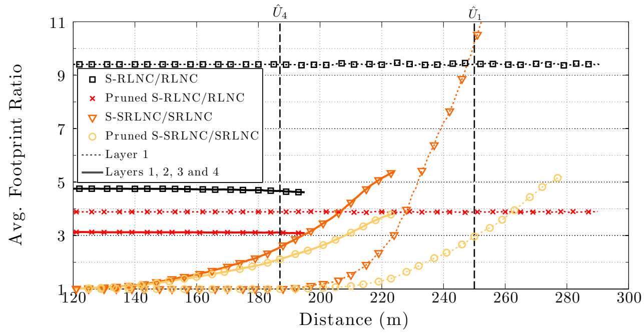
Remark 4.1: Let X be a random variable expressing the number of coded packet transmissions needed to recover a message layer composed of k_ℓ source packets transmitted via the RLNC principle, for $p_\ell = 1/q$. Fig. 7a shows, for different values of p_u, the expected value of X that is $\mathbb{E}[\text{X}] = \tau^{(u,\ell)}_{\text{S-RLNC}}$. In Fig. 7a, we observe that the values of $\mathbb{E}[\text{X}]$ derived when $q = 2$ are close to those obtained when $q = 2^8$, regardless of the value of p_u. For simplicity, let us refer to the case where $p_u = 0$ and $p_\ell = 1/q$. From [12, Eq. (6)], the Probability Mass Function (PMF) $\Pr[\text{X} = r] = \phi^{(u,\ell)}_{\text{X}}(r)$ of X can be expressed as follows:

$$\phi^{(u,\ell)}_{\text{X}}(r) = \begin{cases} \displaystyle\prod_{i=0}^{k_\ell - 1} \left[1 - q^{i-r}\right], & \text{if } r = k_\ell \\ \displaystyle\prod_{i=0}^{k_\ell - 1} \left[1 - q^{i-r}\right] - \prod_{i=0}^{k_\ell - 1} \left[1 - q^{i-r+1}\right], & \text{if } r > k_\ell. \end{cases} \tag{33}$$

Hence, in this case, the expected value of X can be alternatively expressed as $\mathbb{E}[\text{X}] = \sum_{r=k_\ell}^{\infty} r \cdot \phi^{(u,\ell)}_{\text{X}}(r)$. Fig. 7b shows the product of terms $r \cdot \phi^{(u,\ell)}_{\text{X}}(r)$ as a function of r, for $k_\ell = \{30, 70\}$ and $q = \{2, 2^8\}$. In the case of $q = 2$, we observe that the product $r \cdot \phi^{(u,\ell)}_{\text{X}}(r)$ and hence, the PMF of X is non-zero across several values of $r \geq k_\ell$, for both of the considered values of k_ℓ. On the other hand, the PMF of X is non-zero almost for $r = k_\ell$, when $q = 2^8$. Considering a target value of k_ℓ and $q = 2$, from Fig. 7b, we can infer that the sum of non-zero terms $r \cdot \phi^{(u,\ell)}_{\text{X}}(r)$, i.e., $\mathbb{E}[\text{X}]$ is barely greater than or equal to the correspondent sum of terms obtained by considering $q = 2^8$. In fact, in the case of $k_\ell = 30$ ($k_\ell = 70$), $\mathbb{E}[\text{X}]$ is equal to 31.6 and 30 (71.6 and 70), for $q = 2$ and 2^8, respectively. We thus observe that, even though the PMFs of X for $q = 2$ and $q = 2^8$ are significantly different, the corresponding average values of X are comparable. On the other hand, for a given $r \geq 0$, we observe that the probability value $\Pr[\text{X} \leq r]$ may vary significantly as the value of q changes. The same reasoning can be easily extended for different values of PER and applies to all the RLNC strategies discussed in this paper. In addition, that explains the reason way scenarios where $q = 2$ and $q = 2^8$ perform similarly from the point of view of the average transmission footprint.



(a) Stream A



(b) Stream B

Fig. 8. Average footprint ratio $\omega^{(u,1:\ell)}$ in the case of Stream A (Stream B), for $\ell = 1$ and 3 ($\ell = 1$ and 4) and $q = 2$.

Consider Fig. 6, we observe that all the considered network coding strategies can meet the services coverage constraints, for the considered values of $\hat{\tau}$ and $\hat{U}_\ell$. However, since the optimized S-RLNC and S-SRLNC strategies are characterized by values of p_ℓ (for any $\ell = 1, \ldots, L$) that are greater than q^{-1}, the probability of transmitting non-degenerate coded packets associated with all-zero coding vectors is likely to increase. Hence, the average transmission footprints provided by S-RLNC and S-SRLNC are greater than those associated with the optimized RLNC and SRLNC strategies.

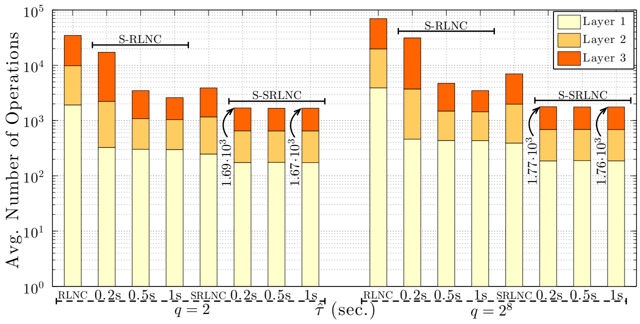
The aforementioned increment in the average transmission footprint has been investigated in Fig. 8a, where we reported the ratio $\omega^{(u,1:\ell)}$ (called "average footprint ratio") between the values of $\tau^{(u,1:\ell)}$ provided by the S-RLNC (S-SRLNC) and RLNC (SRLNC) strategies, for the QoS levels 1 and 3, and $q = 2$. We note that if $\omega^{(u,1:\ell)}$ is equal to 1, the considered sparse network coding strategy provides the same average transmission footprint of the correspondent non-sparse technique. Fig. 8a also shows the same performance metrics for two modified versions of S-RLNC and S-SRLNC, hereafter referred to as "Pruned S-RLNC" and "Pruned S-SRLNC". Those two strategies behave as the proposed optimized S-RLNC and S-SRLNC but in the pruned versions, the target base station does not transmit non-degenerate coded packets associated with all-zero coding vectors. It is straightforward to prove that if a non-pruned sparse RLNC strategy meets the optimization constraints, the correspondent pruned strategy will do the same. From Fig. 8a we observe that the $\omega^{(u,1:\ell)}$ values provided by the S-RLNC at the target distances associated with $\hat{U}_1$ and $\hat{U}_3$ are equal to 9.4 and 3.9, respectively. However, in the case of the Pruned S-RLNC, the average footprint ratios drop to 3.9 and 2.8, for the QoS levels 1 and 3, respectively. With regards to the optimized S-SRLNC strategy, the $\omega^{(u,1:\ell)}$ values associated with the QoS level 1 and 3 are equal to 10.2 and 2.4, respectively. However, also in this case, the Pruned S-SRLNC provides smaller average footprint ratios: 2.9 and 2.1, for the first and the third QoS levels, respectively.

We also observe from Fig. 8a that the $\omega^{(u,1:\ell)}$ values provided by (non-Pruned and Pruned) S-RLNC strategies tend to be constant, while the $\omega^{(u,1:\ell)}$ values related to the S-SRLNC strategies increase as the distance from the target base station grows. That behavior can be explained by the fact that non-systematic network coding strategies require to multicast coded packets from the beginning, while systematic techniques multicast coded packets only after the systematic packets have been transmitted. Hence, as the distance from the center of the cell increases, i.e., as the user propagation conditions get worse, the number of systematic packets successfully received decreases. In those cases, a user needs more coded packets to recover a video layer. However, it is worth noting that the optimized non-Pruned S-SRLNC and, specifically, the Pruned S-SRLNC strategies provide values of $\omega^{(u,1:\ell)}$ that drop below 1.6 for distances that are 22 m and 20 m smaller than the desired coverage, for the QoS level 1 and 3, respectively. The aforementioned analysis applies also in the case of Stream B, Fig. 8b.

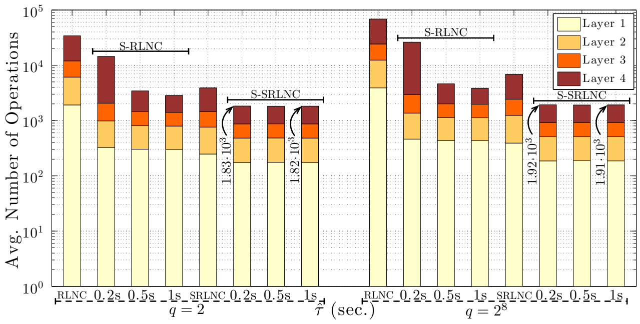
The performance of the considered network coded strategies has been also compared in terms of the complexity of the decoding operations. Likewise to the definition of $\tau^{(u,1:\ell)}$, we define the average number $\epsilon^{(1:\ell)}$ of decoding operations needed to recover the first ℓ video layers as follows:

$$\epsilon^{(1:\ell)} = \begin{cases} \displaystyle\sum_{t=1}^{\ell} \epsilon^{(t)}_{\mathrm{S-RLNC}}, & \text{for S-RLNC} \\ \displaystyle\sum_{t=1}^{\ell} \epsilon^{(t)}_{\mathrm{S-SRLNC}}, & \text{for S-SRLNC.} \end{cases} \tag{34}$$

Also in this case, the definition of $\epsilon^{(1:\ell)}$ can be extended to the non-sparse version of the RLNC and SRLNC, by referring to a value of $p_\ell = q^{-1}$, for $\ell = 1, \ldots, L$. In the case of non-sparse RLNC and S-RLNC, we remark that the value of $\epsilon^{(1:\ell)}$ is independent from the value of the user PER. On the other hand, in the non-sparse SRLNC and S-SRLNC, the number of decoding operations grows as the number of successfully received systematic packets decreases; that happens when the user PER increases. To this end, for all the systematic



(a) Stream A



(b) Stream B

Fig. 9. Average number of decoding operations $\epsilon^{(1:3)}$ and $\epsilon^{(1:4)}$ in the case of Stream A and B, for $q = 2$ and $q = 2^8$. For RLNC and SRLNC, $\hat{\tau}$ is set equal to 0.2. Different colours represents the contribution of each layer to the value of $\epsilon^{(1:3)}$ and $\epsilon^{(1:4)}$.

strategies, we evaluated $\epsilon^{(1:\ell)}$ by referring to a user PER equal to $\hat{p}$.

Fig. 9a shows the value of $\epsilon^{(1:L)}$ provided by all the considered strategies, in the case of Stream A, for $q = \{2, 2^8\}$. We recall from Section III that we refer to just the fundamental finite field operations performed by a Gaussian Elimination-based decoder. Hence, the reception of coded packets associated with all-zero coding vectors has no impact on the number of the considered operations. As a consequence, the Pruned and non-Pruned versions of S-RLNC and S-SRLNC are characterized by the same values of $\epsilon^{(1:\ell)}$.

Let us consider the S-RLNC strategy in Fig. 9a, it provides values of $\epsilon^{(1:L)}$ that are up to 92.5% and 97.08% smaller than those provided by the non-sparse RLNC, for $q = 2$ and $q = 2^8$, respectively. In particular, as expected, the value of $\epsilon^{(1:L)}$ reduces as the target service transmission time $\hat{\tau}$ (sec.) grows. On the other hand, the S-SRLNC strategy ensures values of $\epsilon^{(1:L)}$ that are up to 57% and 74.8% smaller than those associated with the non-sparse SRLNC, for $q = 2$ and $q = 2^8$, respectively. Regardless on the value of q, we observe that the systematic strategies provide values of $\epsilon^{(1:L)}$ that are significantly smaller than those given by non-systematic techniques. That is due to the fact that the decoder may rely on a subset of the systematic packets that have been successfully received and do not need to be decoded. We also observe that in the case of $q = 2^8$, both the non-sparse and sparse strategies are characterized by values of $\epsilon^{(1:L)}$ that can be up to 50.2% greater than in the case where $q = 2$. We repeated the same performance investigation for Stream B. However, due to space limitations, we provide results only in terms of $\epsilon^{(1:L)}$, in Fig. 9b. Also in this case, the discussion and conclusions that we provided for Fig. 9a also apply.

The key points of this discussion can be summarized as follows: (i) In the considered cases, the adoption of the finite field size $q = 2^8$ provides just a minimal reduction in terms of average transmission footprint and significantly increases the complexity of the decoding operations, if compared to the $q = 2$ case, (ii) The adoption of either the Pruned S-RLNC or Pruned S-SRLNC ensures a significant reduction in the average number of decoding operations, (iii) The Pruned S-SRLNC strategy ensures the best tradeoff between transmission footprint and decoding complexity.

V. Conclusions

In this paper, we addressed the issue of the complexity associated to a generic network coding decoder. In particular, we referred to a multicast network scenario where a layered service is transmitted to a set of users.

Based on the proposed modeling, we referred to a scenario where the layered service was delivered in an ultra-reliable fashion. By referring to both the S-RLNC and S-SRLNC strategies, we proposed a constrained convex resource allocation framework suitable for jointly optimizing both the MCS indexes and the code sparsity to be used in the multicasting of each service layer. The objective of the optimization model is that of maximizing the sparsity of the code associated with each layer, and hence, minimizing the number of operations performed by a generic network coding decoder employing Gaussian Elimination. We also showed that the aforementioned computation complexity reduction can be directly mapped onto a computational processing reduction, which allows to eventually prolong the battery life of mobile devices.

As shown by the provided numerical results, the average transmission footprint is likely to increase as the sparsity of the code grows. However, the average transmission footprint can be greatly improved by simply avoiding the transmissions of coded packets associated with all-zero coding vectors, as happens with the Pruned S-RLNC and Pruned S-SRLNC strategies. We observed that the proposed optimization ensures a reduction in the average number of decoding operations of at least 92% and 57%, if compared to the classic non-sparse S-RLNC and S-SRLNC techniques, respectively. We remark that the proposed decoding complexity reduction is obtained without altering the actual implementation of the decoder.

References

[1] P. Popovski, “Ultra-Reliable Communication in 5G Wireless Systems,” in *Proc. of the 1st International Conference on 5G for Ubiquitous Connectivity*, Levi, Finland, FI, Nov. 2014, pp. 1–6.

[2] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. Uusitalo, B. Timus, and M. Fallgren, “Scenarios for 5G mobile and wireless communications: The Vision of the METIS Project,” *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 26–35, May 2014.

[3] R. Ferrus, O. Sallent, G. Baldini, and L. Goratti, “LTE: The Technology Driver for Future Public Safety Communications,” *IEEE Commun. Mag.*, vol. 51, no. 10, pp. 154–161, Oct. 2013.

[4] L. Carlá, R. Fantacci, F. Gei, D. Marabissi, and L. Micciullo, “LTE enhancements for Public Safety and Security communications to support Group Multimedia Communications,” *IEEE Netw.*, to appear 2015. [Online]. Available: http://arxiv.org/abs/1501.03613

[5] P. Seeling and M. Reisslein, “Video Transport Evaluation With H.264 Video Traces,” *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1142–1165, Fourth 2012.

[6] J. Monserrat, G. Mange, V. Braun, H. Tullberg, G. Zimmermann, and Ö. Bulakci, “METIS Research Advances Towards the 5G Mobile and Wireless System Definition,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2015, no. 1, 2015.

[7] B.-S. Kim, S. W. Kim, and R. Ekl, “OFDMA-Based Reliable Multicasting MAC Protocol for WLANs,” *IEEE Trans. Veh. Technol.*, vol. 57, no. 5, pp. 3136–3145, Sep. 2008.

[8] J. Kim, H. Jin, D. K. Sung, and R. Schober, “Optimization of Wireless Multicast Systems Employing Hybrid-ARQ with Chase Combining,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 7, pp. 3342–3355, Sep. 2010.

[9] 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, 2013.

[10] T. Mladenov, S. Nooshabadi, and K. Kim, “Efficient GF(256) Raptor Code Decoding for Multimedia Broadcast/Multicast Services and Consumer Terminals,” *IEEE Trans. Consum. Electron.*, vol. 58, no. 2, May 2012.

[11] C. Khirallah, D. Vukobratović, and J. Thompson, “Performance Analysis and Energy Efficiency of Random Network Coding in LTE-Advanced,” *IEEE Trans. Wireless Commun.*, vol. 11, no. 12, pp. 4275–4285, Dec. 2012.

[12] A. Tassi, I. Chatzigeorgiou, and D. Vukobratović, “Resource Allocation Frameworks for Network-coded Layered Multimedia Multicast Services,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 2, pp. 141–155, Feb. 2015.

[13] D. Ferreira, R. Costa, and J. Barros, “Real-Time Network Coding for Live Streaming in Hyper-Dense WiFi Spaces,” *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 773–781, Apr. 2014.

[14] M. V. Pedersen, J. Heide, and F. H. Fitzek, “Kodo: An Open and Research Oriented Network Coding Library,” in *NETWORKING 2011 Workshops*, ser. Lecture Notes in Computer Science, V. Casares-Giner, P. Manzoni, and A. Pont, Eds. Springer Berlin Heidelberg, 2011, vol. 6827, pp. 145–152.

[15] M. Xiao, M. Medard, and T. Aulin, “Cross-Layer Design of Rateless Random Network Codes for Delay Optimization,” *IEEE Trans. Commun.*, vol. 59, no. 12, pp. 3311–3322, Dec. 2011.

[16] A. Ramasubramonian and J. Woods, “Video Multicast Using Network Coding,” in *Proc. of SPIE Conference on Visual Communications and Image Processing (VCIP)*, Jan. 2009, pp. 1–11.

[17] H. Wang, S. Xiao, and C.-C. J. Kuo, “Random Linear Network Coding with Ladder-shaped Global Coding Matrix for Robust Video Transmission,” *Elsevier J. Vis. Comun. Image Represent.*, vol. 22, no. 3, pp. 203–212, Apr. 2011.

[18] N. Thomos, J. Chakareski, and P. Frossard, “Prioritized Distributed Video Delivery With Randomized Network Coding,” *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 776–787, Aug. 2011.

[19] A. Tassi, C. Khirallah, D. Vukobratović, F. Chiti, J. Thompson, and R. Fantacci, “Resource Allocation Strategies for Network-Coded Video Broadcasting Services over LTE-Advanced,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 2186–2192, May 2015.

[20] L. Lu, M. Xiao, M. Skoglund, L. Rasmussen, G. Wu, and S. Li, “Efficient Network Coding for Wireless Broadcasting,” in *Proc. of IEEE WCNC 2010*, Sydney, Australia, AUS, Apr. 2010, pp. 1–6.

[21] L. Lu, M. Xiao, and L. Rasmussen, “Relay-Aided Broadcasting with Instantaneously Decodable Binary Network Codes,” in *Proc. of ICCCN 2011*, Maui, Hawaii, HI, Jul. 2011, pp. 1–5.

[22] S. Feizi, D. Lucani, C. Sørensen, A. Makhdoumi, and M. Médard, “Tunable Sparse Network Coding for Multicast Networks,” in *Proc. of NetCod 2014*, Aalborg, Denmark, DK, Jun. 2014.

[23] D. Andrén, L. Hellstrøm, and K. Markstrøm, “On the Complexity of Matrix Reduction Over Finite Fields,” *Advances in Applied Mathematics*, vol. 39, no. 4, pp. 428–452, 2007.

[24] J. Barros, R. Costa, D. Munaretto, and J. Widmer, “Effective Delay Control in Online Network Coding,” in *Proc. of IEEE INFOCOM 2009*, Rio de Janeiro, Brazil, BR, Apr. 2009, pp. 208–216.

[25] R. Prior and A. Rodrigues, “Systematic Network Coding for Packet Loss Concealment in Broadcast Distribution,” in *Proc. of ICOIN 2011*, Jan. 2011, pp. 245–250.

[26] A. Jones, I. Chatzigeorgiou, and A. Tassi, “Binary Systematic Network Coding for Progressive Packet Decoding,” in *Proc. of IEEE ICC 2015*, London, United Kingdom, UK, Jun. 2015, pp. 1–5.

[27] X. Li, W. H. Mow, and F.-L. Tsang, “Singularity Probability Analysis for Sparse Random Linear Network Coding,” in *Proc. of IEEE ICC 2011*, Kyoto, Japan, JP, Jun. 2011, pp. 1–5.

[28] ——, “Rank Distribution Analysis for Sparse Random Linear Network Coding,” in *Proc. of NetCod 2011*, Beijing, China, CN, Jul. 2011, pp. 1–6.
[29] J. Blömer, R. Karp, and E. Welzl, “The Rank of Sparse Random Matrices Over Finite Fields,” *Random Structures & Algorithms*, vol. 10, no. 4, pp. 407–419, 1997.
[30] C. Cooper, “On the Asymptotic Distribution of Rank of Random Matrices Over a Finite Field,” *Random Structures & Algorithms*, vol. 17, no. 3-4, pp. 197–212, 2000.
[31] ERICSSON AB, “5G Radio Access,” Tech. Rep., Feb. 2015. [Online]. Available: http://www.ericsson.com/res/docs/whitepapers/wp-5g.pdf
[32] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution*. John Wiley & Sons, 2011.
[33] ITU-T H.264, “Advanced Video Coding for Generic Audiovisual Services,” Tech. Rep., Nov. 2007.
[34] R. Afolabi, A. Dadlani, and K. Kim, “Multicast Scheduling and Resource Allocation Algorithms for OFDMA-Based Systems: A Survey,” *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 240–254, First 2013.
[35] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, “Efficient Broadcasting Using Network Coding,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 450–463, Apr. 2008.
[36] A. Cohen, B. Haeupler, C. Avin, and M. Médard, “Network Coding Based Information Spreading in Dynamic Networks With Correlated Data,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 2, pp. 213–224, Feb. 2015.
[37] O. Trullols-Cruces, J. 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.
[38] J. Kemény and J. Snell, *Finite Markov Chains*. Van Nostrand, 1960.
[39] E. Upton and G. Halfacree, *Raspberry Pi User Guide*. John Wiley & Sons, 2014.
[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
[41] 3GPP TR 36.814 v9.0.0, “Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects,” 2010.
[42] “Video Traces 2 for Terminator (DQP 15, 2 EL) and (DQP 10, 3 EL).” [Online]. Available: http://trace.eas.asu.edu/videotraces2/cgs/