Implementation of Network-Coded Cooperation for Energy Efficient Content Distribution in 5G Mobile Small Cells

Torre, Roberto; Leyva-Mayorga, Israel; Pandi, Sreekrishna; Salah, Hani; Nguyen, Giang T.; Fitzek, Frank

Published in:
IEEE Access

DOI (link to publication from Publisher):
10.1109/ACCESS.2020.3029601

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 23, 2020, accepted October 5, 2020, date of publication October 8, 2020, date of current version October 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3029601

Implementation of Network-Coded Cooperation for Energy Efficient Content Distribution in 5G Mobile Small Cells

ROBERTO TORRE[1], (Member, IEEE), ISRAEL LEYVA-MAYORGA[2], (Member, IEEE), SREEKRISHNA PANDI[1], (Graduate Student Member, IEEE), HANI SALAH[1], GIANG T. NGUYEN[1], (Member, IEEE), AND FRANK H. P. FITZEK[1,3], (Senior Member, IEEE)

[1]Deutsche Telekom Chair of Communication Networks, Technische Universität Dresden, 01062 Dresden, Germany
[2]Department of Electronic Systems, Aalborg University, 9100 Aalborg, Denmark
[3]Centre for Tactile Internet With Human-in-Loop (CeTI), 01062 Dresden, Germany

Corresponding author: Roberto Torre (roberto.torre@tu-dresden.de)

This work was supported in part by the European Union's H2020 Research and Innovation Program under Grant H2020-MCSA-ITN-2016-SECRET 722424.

ABSTRACT The continuous increase of mobile data traffic calls for the design of energy-efficient content distribution mechanisms, to be incorporated in the fifth generation of mobile networks, 5G. One of the biggest concerns of the companies and the research community is to reduce the energy consumption in both the user equipments (UEs) and the network equipment. In this article, we present a novel content distribution framework called Network-Coded Cooperative (NCC) Networks, which benefits from the interplay between mobile clouds (MC) and Random Linear Network Coding (RLNC) to reduce the overall energy consumption in the devices that take part in the communication. This novel framework leads to reduced energy consumption by offloading the cellular interface to a link with greater energy efficiency, for instance, WiFi, within the mobile small cell. We evaluate the performance of our framework analytically and in practical implementation (i.e., testbed) in terms of throughput, energy savings, packet decoding ratio, latency, and synchronicity. In comparison to the conventional content distribution system, for the case of four users, the analytical model and the testbed implementation show energy savings of more than 12% and 8%, respectively. Furthermore, network usage is reduced, losses are neutralized, and the content is synchronously distributed to all users.

INDEX TERMS Network coding, random linear network coding (RLNC), cooperative communication, content distribution, cellular networks, mobile small cells, energy efficiency, traffic offload.

I. INTRODUCTION

As a new mobile generation unfolds, new systems, structures, devices, and protocols are designed to match user's and company's expectations. The fifth generation of wireless networks (5G) presents a new networking paradigm where a massive number of mobile devices are connected anywhere and anytime. It has been reported by Cisco [2] that mobile data traffic will increase sevenfold between 2017 and 2022, reaching 77.5 exabytes per month by 2022. Moreover, wireless traffic will reach 71% of the total Internet Protocol (IP) traffic, and immersive media services will multiply their uses. For instance, IP video traffic will represent 82% of all IP traffic, live video will grow 15-fold, video surveillance cameras traffic will increase around 700%, and virtual reality (VR) and augmented reality (AR) traffic will have a Compound annual growth rate (CAGR) of 65%. Furthermore, the expected number of connected and connected devices will increase in the next years. During this time, technologies such as the Internet of Things (IoT) and Wireless Sensor Networks (WSN) will dominate the user's services with smartphones, tablets, machines, and sensors. Cisco reported in 2017 [3] that the number of connections will increase up to 29.3 billion in 2023. The number of connected devices will also increase up to 13.1 billion. In order to carry this increase in mobile traffic, micro, pico, and femtocells base stations are

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco Rafael Marques Lima.

expected to populate the network. These base stations will conform heterogeneous Ultra Dense Networks (UDNs) that provide ubiquitous connectivity.

All the aforementioned applications have in common the need for a massive content distribution system and infrastructure. The current state of the art of cellular content distribution establishes a cellular, unicast connection between the cellular gNodeB (gNB) and the user equipment (UE), regardless of the content requested by the users. If multiple users request the same information, the gNB establishes multiple replicated unicast connections, which leads to inefficient use of cellular resources. Examples of these situations can be found in stadiums, music festivals, or theaters where cameras can stream to the mobile phones of the audience, passengers in a train receiving news of connecting trains, weather news or delays, or players in AR mobile games such as Pokemon Go. Distributed ledger technologies (DLTs) (e.g., Blockchain) present another use case for massive content distribution, where every DLT node has to store a copy of a common timestamped and ordered database called ledger [4].

This problem has been in the spotlight of the industry and research community in the last years. Consequently, new architectures, protocols, and schemes have been developed with the target of providing a more efficient content distribution framework. The 3rd Generation Partnership Project (3GPP) proposed Multimedia Broadcast Multicast Service (eMBMS) [5], a framework that provides multicast capabilities to Long Term Evolution Advanced (LTE-A). However, a European Union report [6] detected several issues in eMBMS, such as reduced transmission range, high energy consumption, and poor spectral efficiency. Many other researchers developed in parallel different methods to provide massive content distribution, such as the use of cloudlets or device to device communication, combined with forward error correction (FEC) techniques such as network coding (NC). NC has been known for increasing throughput in wireless networks [7]. Random Linear Network Coding (RLNC) [8] is widely used in streaming systems due to the low latency it provides. Instead of using deterministic coefficients, the transmitter creates a random coding matrix. Consequently, the intermediate nodes do not have to wait until the generation is complete to decode and forward the information, but they only need to generate new coding coefficients and send them. The coding matrix is generated by a linear combination of the vector of source symbols with the vector of coefficient, which are taken from a Galois-field of size q, GF(q). In the end, the decoder only needs to receive enough linear independent packets to perform the Gaussian elimination in the decoding matrix and retrieve the source symbols.

Although a lot of research has been done in this field, it has not been found yet a framework that enables efficient massive content distribution in cellular networks. In particular, dissemination techniques that rely on cloudlets and mesh networks suffer from high jitter and latency in multipath communications. In this article, we propose a new framework called Network-Coded Cooperation (NCC) for efficient content distribution in cellular networks. NCC offloads traffic from the cellular network to a small cell short-ranged network. NCC leverages the massive UDNs to convert some of the smaller cells into Mobile Clouds (MCs). MCs are defined in [9] as ''a cooperative arrangement of dynamically connected nodes sharing opportunistically resources''. This means that nodes (i.e. UEs) in an MC communicate among each other to obtain a common or personal benefit. An example MC can consist of various UEs that communicate with each other through a short-range technology such as WiFi. Regarding data dissemination, it has been proved that MC can be a promising solution [10]. Similar cooperative relaying systems have been introduced to improve the performance and reliability of wireless networks [11]. NCC has three different agents: the first agent is a RLNC encoder that is placed inside the video server. The second agent is a NCC controller running in the edge cloud, and the third agent is a RLNC recoder placed inside the UE. Both the first and the second agents are enabled through Network Function Virtualization (NFV) and Software-Defined Networking (SDN). NFV enables them to be virtualized as Virtualized Network Functions (VNFs) so they can be flexibly deployed. SDN enables them to dynamically adjust the configuration to the events in the network. SDN and NFV together enable the placement of both the first and the second agent in a Mobile Edge Computing (MEC) server. Consequently, NCC can: (i) dynamically switch between configurations, (ii) dynamically adapt redundancies depending on the network's quality, (iii) dynamically create new NCC controllers to scale the service to new users, and (iv) seamlessly move between edge clouds to reduce latency, energy consumption, and network traffic [12]. The technology of SDN, NFV, and MEC, in combination with UDNs and the massive deployment of base stations, enable NCC to be viable in 5G.

NCC consists of two phases: the cellular phase and the MC phase. In the first phase, the gNB segments the content in packets and groups them in blocks of g packets; hereafter we refer to g as the generation size. The server in the gNB encodes the packets and sends them to the mobile users in a round-robin manner. At the end of this phase, each UE receives a part of the content that will be forwarded to the other UEs in the MC. In the MC phase, the UEs share their part of the content to the rest via WiFi multicast. NCC leverages the interplay between MCs and NC to increase network throughput, increase network resilience, and reduce energy consumption with small latency and complexity overhead.

In our previous works [13], [14], we proposed two different analytical models for NCC. The models estimated the optimal number of coded retransmissions in the MC for successful content delivery. The RLNC protocol used was systematic RLNC [15]. In this work, we present the analytical model of NCC for the protocol PACE-MG [16], which is an improved version of RLNC PACE [17]. Moreover, this is the first work to provide testbed results of NCC. We assess the

TABLE 1. Comparison of the related works. New acronyms (*) are listed here due to space constraints: Network Coding with Multiple Interfaces (NCMI), multicast (mcast), Instantly Decodable Network Coding (IDNC), Opportunistic Network Coding (ONC), Free Viewpoint Video (FVV), Hypertext Transfer Protocol (HTTP), acknowledgement (ACK).

Acronym	Ref.	Summary	Cellular	D2D	FEC	ACK*	Focus	Results
NCMI	[18]	Cooperative NC Framework that combines cellular and D2D communications.	Broadcasts all (uncoded)	1 UE broadcasts (coded)	Batch-RLNC, NCMI*-Instant	✗	Exploiting NC to recover from losses	Reduced number of transmission slots
ARNC	[19]	Hybrid mcast and D2D transmission scheme based on adaptive RLNC.	Broadcasts all (coded)	1 UE broadcasts (recoded)	Adaptive RLNC on the fly	Every L tranmissions	Reduce error rate, improve throughput	Improved network throughput
F-RANs	[20]	Opportunistic NC algorithm to offload using F-RANs	eNB mcast* using opportunistic IDNC*	✗	ONC*	✗	Cloud offloading in HetNets	Reduced complexity, improved throughput
MicroCast	[21]	Cooperative system for video streaming with RLNC	Unicasts a part (uncoded)	1 UE pseudo-broadcasts (coded)	NC Utils	✗	Cooperative cellular-D2D to improve streaming	Increased data rate without energy penalty
PAFV	[22]	Distributed NC-based efficient and scalable algorithm for live FVV* streaming	Unicast parts (coded)	Broadcasts the parts	Adapted RLNC for the protocol	Neighbor information exchange	Collaborative FVV* with multiple interfaces	Outperforms greedy algorithm with less distortion
NCVCS	[23]	NC for data distribution in LANs	✗	Each UE multicasts its part of data	RLNC	✗	Distribution of video-conference system	Achieves higher reliability and performance
eMBMS	[5]	Multicast and broadcast framework for LTE	eNB unicasts using FLUTE [24]	✗	Turbo-codes	HTTP* ACK*	Massive content delivery over LTE	Development of full-stack mcast/bcast protocol
NCC (RLNC)	[13]	Content distribution scheme with RLNC and cooperative relaying	Unicast parts from file	Multicast their own part via WiFi	Systematic RLNC [8]	No ACK*	Reliable disseminate data to UEs	Improved throughput and energy savings
NCC (PACE-MG)		Content distribution scheme with PACE-MG and cooperative relaying	Unicast parts from file	Multicast their own part via WiFi	PACE-MG [16]	ACK* added in [25]	Reliable disseminate data to UEs	Multicast resilience, high throughput, energy savings

validity of our framework with the comparison of the two aforementioned methods. We evaluate the analytical model presented to calculate the energy consumption at the UEs and compare our results with the hardware implementation. Please note that in this article the analytical model is used to find upper and lower bounds for the testbed and not for accuracy. Please refer to [14] if accuracy is the desired metric in the analytical model. The analytical results, for a MC of eight users, show energy savings of 30% when compared to single unicast. Implementation results confirm the analytical ones with an extra offset due to overprovisioning, by reducing the energy consumption at the UEs from 100% in the case of single unicast to a 27% in the case of eight nodes in the MC. Moreover, we reduce the packet loss rate in the implementation to a negligible value, while studying the latency and providing a high degree of decoding synchronicity.

The remainder of the paper is organized as follows. The state of the art is presented in Section II. Then, in section III we describe the content distribution scenario and we propose our solution for that specific scenario. In section IV we evaluate the proposed solution using two different approaches, namely an analytical model and simulations in a real testbed. Finally, section V concludes the paper.

II. RELATED WORK

Table 1 shows a comparison of the related works and the framework proposed in this article. The work in [26] presented a comparison of different systems for massive content distribution in cellular networks. On one hand, eMBMS [5] leveraged file delivery over unidirectional transport (same as HTTP, but unidirectional) and single frequency networks (SFN) to massively stream video over cellular networks. However, the performance obtained was inferior to the expectations [6]. On the other hand, cooperative relaying leveraged the use of mobile clouds [27] to improve network performance in terms of throughput and resilience, and, as a result, it reduces the energy consumption of the system [28]. For example, Shnaiwer *et al.* [29] proposed a NC broadcast offloading scheme in homogeneous cellular networks that offloads macrocell traffic into femtocells. The same authors expanded their work by adding ONC in the offloading scheme [20].

NC had a very important role in streaming since Ahlswede *et al.* [7] introduced it to increase throughput and resilience in wireless networks. In particular, the interplay between RLNC and cooperative relaying was first proposed in [30]. RLNC [31] is a popular NC protocol used in streaming because the intermediate nodes do not need to know the coding coefficients to forward the packets, but they can generate new coding coefficients as soon as the packets arrive, which reduces the latency of the system. This is known as coding *on the fly*. Consequently, RLNC has been used in many works [23], [32]–[34]. Wang *et al.* [23] proposed a dissemination scheme in local area networks (LANs) that uses

RLNC to improve the throughput and resilience in the system. Chau *et al.* [32] used RLNC in the relay nodes to improve the efficiency of the network during the uploading phase. Tassi *et al.* [33], [34] studied RLNC-based, energy-efficient resource allocation models for multicast scenarios in heterogeneous networks. RLNC and cooperative relaying have been used together several times.

Keshtkarjahromi *et al.* [18] developed a framework for cooperative mobile devices in a joint setup that consisted of cellular and device to device (D2D) links. MicroCast [35] is a novel cooperative system where each mobile device uses simultaneously two network interfaces, a cellular interface that connects to the video server and downloads parts of the video, and a WiFi interface to locally exchange the received data via unicast links. In their work, they firstly formulate the theoretical problem and they provide a distributed solution for it, and they prove their system by means of simulation and practical implementation. Yan *et al.* [36] contributed with a similar study but from a theoretical point of view. They formulated the problem of optimal collaborative transmission scheduling and they proposed a new weight function for measuring the quality of a coding pattern. Antonopoulos *et al.* [38] and Antonopoulos and Verikoukis [39] proposed and tested multi-player, non-cooperative game theoretic Medium Access Control (MAC) strategies to obtain the balance between saving energy and successfully transmitting the data. The authors used RLNC to eliminate the exchange of control packets. The Network-coding-based video conference system (NCVCS) [23] is a reliable framework for massive content distribution using WiFi multicast. Specifically, NCVCS leverages the benefits of RLNC and multicasting to provide a solid content distribution system inside a local area network. However, the cellular network as an overlay network of the NCVCS system was not considered [23]. Aymen *et al.* [39] offloaded cellular heterogeneous networks through cooperative RLNC-based video streaming with D2D. Li *et al.* [19] leveraged the benefits of adaptive RLNC and D2D communication to provide a hybrid multicast system in cellular heterogeneous networks. Similarly, Zhang *et al.* [40] proposed a cooperative multicast system based on adaptive RLNC. Both works used RLNC in combination with D2D communications to fix the losses produced in the cellular link. Le *et al.* [21] proposed a similar approach to the aforementioned ones, but in Android devices. They increased smartphone throughput while maintaining the same battery life. Zhang *et al.* [22] developed an algorithm based on encoded anchors (texture-plus-depth). The eNodeB (eNB) unicasted parts of the anchors to the UE, which broadcasted the encoded anchors. Firooz and Roy [41] studied the latency upper and lower bounds for content distribution using network coding, and Torre *et al.* [42] implemented and studied the latency of the scenario. The analytical model of RLNC has been studied in [43], who proposed a Newton method approximation to adjust the channel losses to optimize channel usage. Tsimbalo *et al.* [44] studied the decoding probability of one source and two destinations using multicast under RLNC. In their work, they have to deal with the correlation between packets received at each node. We have to cope with a similar problem in our analytical model.

Leyva-Mayorga *et al.* presented the NCC analytical model in [13]. This model was further extended in [14]. However, the results showed that the increase of complexity in the model in comparison to the extra accuracy achieved was not significant, and the initial model is sufficiently accurate in general scenarios. R. Torre developed a wired version of a demonstrator called *5G Nokia Stadium Experience*, shown in Mobile World Congress 2017. In this demonstrator, four clients were placed on the other side of a 5G network simulator, and they were connected to a video streaming server. The demonstrator was further extended into a wireless version, which was presented at CCNC and CES Las Vegas 2018 [45].

III. SYSTEM DESCRIPTION

In this section, we first introduce the scenario we want to address and then we propose our solution, NCC networks, to solve this scenario.

A. SCENARIO

The target scenario consists of multiple users (i.e., the UEs) inside a cell that requests the same information (e.g. a video file) through the cellular network to one or more servers far away from the UEs. These users are close enough to each other so these can communicate using a short-range wireless technology like WiFi, forming a complete network graph. The scenario is depicted in Fig. 1. The gNB establishes one unicast session per user requesting the video file. This scenario is likely to arise in situations where social events concentrate the users spatially and these share a common interest towards particular video streams, for example, stadiums, auditoriums, conferences, or even trains. As mentioned in the previous section, the traditional solution to this scenario is having the cellular gNB to establish multiple and redundant sessions to send the same data. This is inefficient in terms of energy consumption, bandwidth usage, and can also cause great levels of interference within the cell.

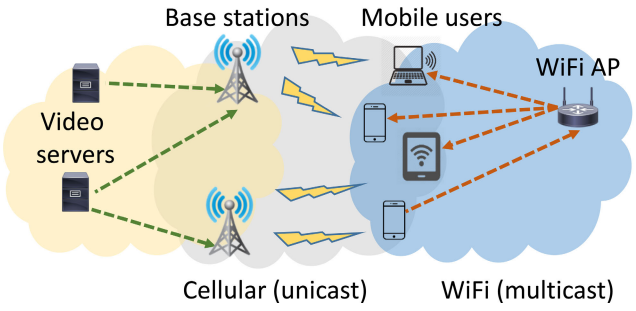


FIGURE 1. Graphic description of the scenario under evaluation.

B. PROPOSED SOLUTION

In our solution, NCC networks, we extend the concept of small cells to a more specific one called MCs. We define

an MC as a group of nodes inside a small cell that share their resources opportunistically, cooperating with each other to obtain a common benefit. This protocol uses small cells to offload the traffic from cellular communication into a short-range communication inside the small cell, like WiFi. The protocol comprises two different phases. In the first phase, communication occurs either from the gNB to the UEs, and, in the second phase, the UEs in the MC communicate with each other. Hence, these phases are called the cellular phase, and the MC phase, respectively. The cellular and MC phases can occur sequentially or in parallel, and communication in the MC phase can occur directly among the UEs or through an access point (AP).

NCC has three different agents: the first agent is a RLNC encoder that is placed inside the video server. The second agent is a NCC microservice running in the edge cloud, hereafter named NCC controller. The third agent is a RLNC recoder placed inside the UE.

We use RLNC in the video server to encode the video packets, and in each UE of the MC to recode and decode the packets. In RLNC, the encoder gathers the packets in blocks of size g, namely generation size. It linearly combines packets in the block multiplying them by random coefficients, creating new coded packets that are transmitted through the network. At the destination, the decoder only needs enough linear independent packets to fill a coding matrix of g columns until the matrix is full-rank. Then, the decoder performs Gaussian elimination to that matrix to obtain the original packets. The main benefit of using RLNC in this setup is that every NC coefficient is random, so during the recoding phase, the UEs do not need to wait for any specific coefficient. They simply recode the incoming packet with a new random coefficient and change the RLNC header. This reduces the latency that is inherent to coding, unlike in other coding protocols like Reed Solomon or turbo codes. Furthermore, in the case of lossy scenarios, the source only needs to send extra redundant packets to the destination, avoiding feedback techniques that would delay the communication.

The evaluations (analytical model and testbed) are done using a novel experimental RLNC protocol called PACE multigeneration [16]. We decided to use this protocol because it fulfills the NC protocol requirements of NCC. In particular, NCC needs high throughput and resilience to efficiently distribute video content, low latency to provide the content live (like in streaming services), resilience against high-jitter and multipath networks since the protocol is based on creating a mesh inside mobile clouds to distribute the content, low energy to extend the battery life of the devices. PACE [17] is an approach that focuses on lowering the latency per packet issue. It includes coded redundancies between the systematic packets of the generation. Therefore, the receiver will not have to wait until the end of the generation to recover from a loss. However, PACE, as well as many other RLNC protocol, provides in-order-delivery, which means that the protocol waits until the next incoming packet with the next source symbol arrives. PACE underperforms in high-jitter networks

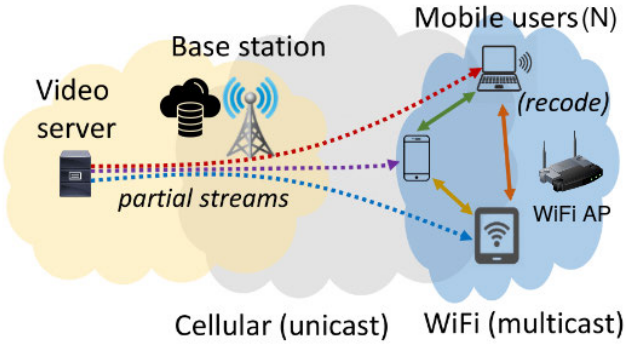


FIGURE 2. Concept of Network-Coded Cooperation. Adapted from: [45].

in terms of decoding ratio and latency. First, because it has no recovery mechanisms for unordered packet arrival and, as a result of it, packets stall in the decoding queue waiting for older packets, thus increasing the latency in the system. PACE multigeneration has a mechanism to handle multiple generations at the same time. It divides the original decoder into two layers: a controller that redirects the traffic and multiple subdecoders, which are entities that are created when the first packet of a new generation arrives. Each subdecoder stores the data of a single generation, and they are destroyed when the generation is decoded. More information about Pace multigeneration can be found in [16].

The NCC controller is the agent in change of monitoring the traffic and adapting NCC to the network conditions. The monitoring and adaptation must be fast, dynamic, and flexible, and thus, the NCC controller is preferably placed in the gNB, as close as possible to the user. If not, it can be placed in the 5G MEC server. It organizes MCs in periodic formation phases, where nodes can join or leave the cloud on the fly. The structure of each MC can only change during the formation phase. However, in case of failure of the network, the NCC controller can trigger an emergency formation phase (for example, if many nodes left a MC and the data does not arrive successfully). The NCC controller decides how many nodes are assigned per MC, based on the network conditions. Moreover, it is in charge of adjusting parameters during the application runtime. Parameters such as the coding ratio or the distribution of redundancies can be adjusted dynamically. We define N as the number of nodes in an MC; hereafter referred to as the cloud size. We assume that all members in the mobile cloud can have cellular communication with the gNB and they are close enough to communicate with the rest of the nodes in the MC through a short-range technology, namely WiFi. They all request the same content, and nodes can eventually join or leave the cloud.

NCC achieves the successful content distribution in two phases: the cellular phase and the MC phase.

1) CELLULAR PHASE

The gNB distributes the g packets to the UEs connected via time-multiplexed unicast sessions in a round-robin fashion.

Each of the N UEs is assigned an index, in the set $\mathcal{N} = \{n \in \mathbb{Z}_+ \mid i \leq N\}$, which defines the order in which they will receive the data packets from the gNB. That is, as illustrated in the left part of Fig. 3, the first packet is sent to the first (purple) UE, in the first time slot. In the second time slot, the gNB sends the second data packet to the second (blue) UE and so on. The gNB will send a second packet to first UE at time slot $(n + 1)$ after sending a data packet to the n-th UE; this will be the $(n + 1)$-th packet of the generation. In the specification of 5G New Radio (NR) [46], the data transmission takes place in a slotted channel, whose minimum scheduling unit is one subframe, with duration $d_s = 1$ ms. That is, the minimum unit for data transmission (downlink) in NR is the *physical resource block* (PRB), which is defined as the number of consecutive Orthogonal frequency-division multiplexing (OFDM) symbols in the time domain and the number of consecutive subcarriers in the frequency domain [46]; in the time domain, two PRBs fit in one subframe. Therefore, in the frequency domain, only one PRB is utilized simultaneously in each cluster.

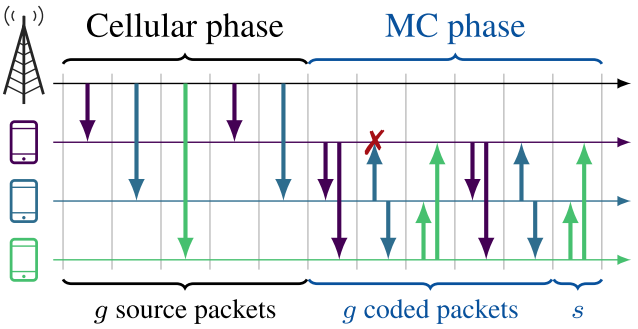


FIGURE 3. Timing diagram for the proposed NCC protocol given 3 nodes, $g = 5$ packets, and $s = 1$ time slot reserved for the transmission of a redundancy packet. The error that occurred in the second time slot is recovered with the first redundant packet [13].

Note that packet losses in the cellular phase may greatly impact the decoding after the MC phase. That is, if the aggregated packets in all the UEs in the MC are not sufficient to create a matrix of rank g at the end of the cellular phase, decoding becomes impossible. In the analytical model, we consider an upper bound in performance by assuming that no errors occur in the cellular communication. Specifically, this results in the minimum energy consumption in this phase and does not impact the communication in the MC phase. This latter statement is valid considering the low data rate (see Table 4) and NR has error correction algorithms such as Hybrid Automatic Repeat reQuest (HARQ) and modifies the modulation and coding scheme (MCS) if the packet error rate (PER) is higher than 0.1 [47]. Therefore, errors can be recovered at the cellular phase before initiating the MC phase. In the testbed, losses may occur at the cellular link. However, we mitigate their impact by adding extra redundancies. This places our scheme in the worst-case scenario for our protocol, since the current state of the art is to use only the cellular channel and the losses in this channel will be zero.

At the end of this phase, all g packets will be distributed over the UEs, where each UE will have either $\lceil g/n \rceil$ or $\lfloor g/n \rfloor$ packets depending on the order in which the connection between the gNB and the UE was established.

2) MC PHASE

When a UE receives a packet from the server in the gNB, it will be in charge of redistributing the packet to the rest of the nodes in the MC. Since no feedback messages are transmitted, the gNB must inform the number of time slots allocated for the content distribution within the MC to the UEs. Each UE will be assigned an index n to create the TDMA schedule in the cellular phase. At every time slot, a UE sends a WiFi multicast packet to the remaining UEs in the MC; each transmitted packet at this phase is coded using full-vector RLNC. The transmitting client is changed at every time slot to distribute all resources in the MC uniformly. The time slot in this phase does not need to be the same as in the cellular phase since a different data rate can be used.

A timing diagram of our NCC protocol is depicted in Fig. 3, where we show an example with three nodes, a generation of five packets, and one coded retransmission. In this diagram, we show how the data dissemination protocol on three devices and five packets. In the cellular phase, the packets are first sent in a round-robin fashion to the UEs. Then, each UE records and forwards the coded packets to the rest. The system recovers from an error that occurred in the second time slot, in the first coded retransmission.

However, there are still challenges arising when using NCC schemes. The first challenge is the modeling of multicasting when multiple sources are sending to the multicast group. The work in [43] uses an approximation algorithm to obtain the redundancies (i.e. the number of coded packets) needed for a certain RLNC protocol. The work in [44] models a single source and two destinations scenario under RLNC. We use the same lower bound used in [44] to solve our problem. The second challenge resides in modeling both incoming packets from the gNB and the multicast group at the same time. Finding a model for such a scenario is not straightforward due to the possible linear dependency of every coded transmission. Therefore, an upper bound is used in the analytical part. However, the implementation was accessible so we decided to use the multi-antenna approach to leverage throughput and latency in the UEs [48]. The third challenge is the nature of the application as all UEs need to be synchronous. Future 5G applications will require a synchronous content as it was observed in [49]. Our system can provide a synchronous content output when reducing the errors to zero. The fourth challenge of this system was introduced in [50]. The author identified that the large number of feedback messages needed to keep track of the state of the UEs is one of the main problems in existing cooperative systems. Suppressing the feedback messages in the MC and using an analytical model to calculate the number of coded transmissions the reliability of the system will not be affected and it will gain in cellular network usage and energy consumption.

IV. EVALUATION

In this section, we evaluate the proposed protocol using two different approaches. We first present the analytical model to obtain a lower bound for the number of coded transmissions needed to provide successful content distribution. We evaluate the analytical model in terms of packet loss and energy consumption. Then, we built a testbed that runs the proposed approach in real hardware. We evaluate the testbed in terms of the loss ratio, energy consumption, latency, and synchronicity.

A. MODEL AND TESTBED DESCRIPTION

In this subsection, we give an overview of the analytical model that we use to obtain the minimum number of coded transmissions needed to provide successful content distribution. Afterward, we introduce the aforementioned testbed.

1) ANALYTICAL MODEL

We are interested in using a simple, yet accurate, model to obtain the reliability of the system as a function of s, defined as the number of time slots allocated for the transmission of redundant packets in the MC. In our analytical model, we assume that one redundant transmission occurs at each allocated time slot. Hence, the total number of redundant packet transmissions is also s. Table 2 lists and describes the symbols used in the model.

TABLE 2. Table of symbols used for the analytical model.

Symbol	Description
N	Number of UEs in the MC
s	Number of time slots allocated for redundant transmissions in the MC
S	RV of the number of redundant transmissions needed to decode the generation in all UEs
g	Generation size
n	Index of the UEs
g_n	Total number of data packets received by the n-th UE in the cellular phase
s_n	Number of redundant transmissions towards the n-th UE
$X_s^{(n)}$	rank of the coding matrix of the n-th UE at time index $s \in \mathbb{Z}$
R_{ue}	Data rate of the UE
$\overline{E}_{\text{ue}}(N)$	Average energy consumption per UE for a given N

Let S be the random variable (RV) that defines the number of coded transmissions needed to make the coding matrix of each UE full-rank. That is, S has a phase-type distribution that describes the probability that every UE decodes the generation at each $s \in \mathbb{Z}$. We refer to S from now on as the probability of successful distribution.

In the cellular phase, the g source packets are distributed among the N UEs in a round-robin manner. Each UE $n \in \mathcal{N}$ will receive a part of those g packets that is different from the remaining $N-1$ UEs. We define the total number of data packets received by the n-th UE in the cellular phase as

$$g_n = \left\lceil \frac{g-(n-1)}{N} \right\rceil; \tag{1}$$

the latter is the rank of the UEs at the beginning of the MC phase. Note that the generation will not be decoded in all cases where $\sum_{n=1}^{N} g_n < g$. Hence, it is essential for the gNB to ensure the correct distribution of the packets among the UEs in the MC.

We model the MC as a complete graph, where the neighborhood of the n-th UE is $\mathcal{N}_n = \{j \mid j \in \mathcal{N} \setminus n\}$. We denote the packet erasure probability for the WiFi links between any two UEs in the MC as ϵ. Next, we define the stochastic process $X_s^{(n)}$ as the rank of the coding matrix of the n-th UE at time index $s \in \mathbb{Z}_{\geq 0}$, whose support for any s is $x = \{0, 1, \ldots, g\}$. Note that at each s, the number of redundant packet transmissions towards the n-th UE (i.e., from every $j \in \mathcal{N}_n$) may be different. Specifically, we denote the latter as s_n, which is defined as the function of s, n, and N given as

$$s_n = f(s, n, N) = s + g_n - \left\lceil \frac{g+s-(n-1)}{N} \right\rceil. \tag{2}$$

As described above, all the transmissions in the MC are encoded using full-vector RLNC. Please note that a correlation between the packets received at each UE exists during the MC phase due to the possible linear dependency of all of them. Besides, the coding matrix of the UEs is not full-rank until a sufficiently large number of coded transmissions s are performed. Hence, the probability of receiving a linearly independent packet depends on the rank of the receiver and the transmitter, as well as on the correlation between their coding matrices. These characteristics are extremely difficult to capture analytically [14], [44], hence, we follow a simplistic approach and assume that the coding matrix used for recoding at every transmitter is full-rank. Naturally, this leads to an upper bound in decoding probability for a given value of s. We have employed this approach previously with accurate results. The interested reader is invited to consult our previous work [13], [14] for details on the accuracy decrease due to this simplification, its implications, and on alternate approaches to capture the described characteristics.

Building on this, we approximate the probability of linear independence for the coded transmissions in the MC as

$$\mathbb{P}(x, g) = 1 - q^{x-g} \geq 1 - q^{x+z-g}; \tag{3}$$

the last term in (3) is the exact probability of linear independence for a given z, defined as the number of degrees of freedom (DOFs) that are missing from the receiver and the transmitter.

Also let $S^{(n)}$ be the RV that defines the number of redundant transmissions from the $j \in \mathcal{N}_n$ UEs needed for the coding matrix of the n-th UE to be full rank. $S^{(n)}$ also has a phase-type distribution whose domain is the set of values for the time index s_n. The latter can be calculated from a given s as

$$s_n \triangleq s + g_n - \left\lceil \frac{g+s-(n-1)}{N} \right\rceil. \tag{4}$$

Note that a minimum number of transmissions are needed for the N UEs to distribute the packets received from the gNB. Specifically, since every UE received g_n unique packets at

the cellular phase, every UE must transmit at least g_n times in the MC phase. Hence, the absolute minimum number of coded transmissions in the MC phase is g. Building on this, we denote $X_0^{(n)}$ as the RV that represents the rank of the coding matrix of the n-th UE at the end of these g transmissions.

To calculate the pmf of $X_s^{(n)}$, we first define $\mathbf{C}$ to be a coding matrix of size $r \times c$ s.t. $r \in \mathbb{Z}_{\geq 0}$ and $\{c \in \mathbb{Z}_+ \mid c \leq g\}$. The elements of $\mathbf{C}$ are selected uniformly at random from GF(q). The probability that the rank of a coding matrix matrix $\mathbf{C}$ is x is given as [44]

$$p_{X_\mathbf{C}}(x, r, c) = \Pr[X_\mathbf{C} = x \mid r, c] = \begin{cases} 0 & \text{for } x > \min\{r, c\}. \\ \dfrac{\prod_{j=0}^{x-1} \frac{(1-q^{j-r})(1-q^{j-c})}{(1-q^{j-x})}}{q^{(r-x)(c-x)}} & \text{otherwise.} \end{cases} \tag{5}$$

From there, it is easy to derive the probability that a coding matrix $\mathbf{C}$ is full-rank as

$$F(r, c) = p_{X_\mathbf{C}}(c, r, c) = \begin{cases} 0 & \text{for } r < c, \\ \prod_{j=0}^{c-1} \left(1 - q^{j-r}\right) & \text{otherwise.} \end{cases} \tag{6}$$

Then, we use (4) and (5) to obtain the pmf of the rank of the coding matrix after $g + s$ transmissions as

$$\begin{aligned} p_X(x, s_n; n, N) &= \Pr\left[X_s^{(n)} = x\right] \\ &= \sum_{u=0}^{g-g_n+s_n} \binom{g - g_n + s_n}{u} (1-\epsilon)^u \, \epsilon^{g-g_n+s_n-u} \\ &\quad \cdot p_{X_\mathbf{C}}(x - g_n, u, g - g_n). \end{aligned} \tag{7}$$

The closed-form expression for the probability of decoding the generation at a UE n for a given s is obtained by substituting x with g in (7). On the other hand, for the whole MC, the exact probability of decoding with s redundant transmissions is defined as

$$F_S^*(s; N) \triangleq \Pr\left[\bigcap_{n=1}^{N} X_s^{(n)} = g\right]. \tag{8}$$

However, capturing the correlation between coding matrices has proven to be a complex task that is intractable even for relatively small MC sizes. Instead, we calculate an upper bound by assuming $\left\{S^{(n)}\right\}$ to be a set of independent RVs. This allows us to calculate $F_S(s; N)$ as

$$\begin{aligned} F_S(s; N) &\triangleq \prod_{n=1}^{N} \Pr\left[X_s^{(n)} = g\right] \\ &= \prod_{n=1}^{N} p_X(g, s; n, N) \geq F_S^*(s; N) \end{aligned} \tag{9}$$

The latter upper bound is exact for $N = 2$.

From this point, it is easy to calculate diverse performance indicators. In the following, we calculate the throughput and the energy consumption per UE, along with bounds for the packet loss rate and packet latency. We consider that the same data rate is used at the cellular and at the WiFi links, so the throughput per UE is simply given as

$$\begin{aligned} T_{\text{ue}}(n; g, s, N) &= \frac{\mathbb{E}\left[X_s^{(n)}\right] \ell}{d_s(2g + s)} \\ &= \frac{\ell}{d_s(2g + s)} \sum_{x=1}^{g} x \, p_X(x, s_n; n, N) \end{aligned} \tag{10}$$

where ℓ is the packet length and d_s is the duration of the subframes in the cellular phase and of the time slots in the MC phase. In [14] we provided the closed-form expression for the upper bound in throughput per UE for different data rates at each interface.

Next, we calculate the average energy consumption per UE, $\overline{E}_{\text{ue}}$ as

$$\begin{aligned} &\overline{E}_{\text{ue}}(N, s) \\ &= \frac{1}{N d_s}\Bigg[g P_{\text{cel,rx}} + (g + s) P_{\text{wifi,tx}} \\ &\quad + \left((N-1)g + \sum_{n=1}^{N} \sum_{u=0}^{s_n} u p_X(g, s_n; n, N)\right) P_{\text{wifi,rx}}\Bigg] \end{aligned} \tag{11}$$

where $P_{\text{cel,rx}}$, $P_{\text{wifi,rx}}$, and $P_{\text{wifi,tx}}$ denote the power consumption during reception in the cellular link, and reception and transmission in the WiFi link, respectively.

Next, let $\hat{L}_{K,s}$ be RV of the packet loss ratio for a given s when K consecutive generations are transmitted. To calculate the latter, first, let $L_s^{(n)}$ be the RV of the number of lost packets at the n-th UE at the end of the MC phase, where s redundant packets were transmitted. Note that a packet is lost if it cannot be decoded from the received coded packets. Therefore, the rank of the coding matrix of a given UE x is an upper bound to the number of decoded packets, which gives $\Pr\left[L_s^{(n)} > g - x\right]$. Next, let $L_s(k)$ be the RV of the number of lost packets in the MC for the kth generation. Assuming that $\left\{L_s^{(n)}\right\}_{n \in \mathcal{N}}$ is a set of independent RVs, we have

$$L_s(k) = \sum_{n=1}^{N} L_s^{(n)}. \tag{12}$$

Analogously, for the transmission of K generations, assuming $\{L(k)_s\}_{k \in \mathbb{Z}}$ is a set of independent RVs, the pmf of the packet loss ratio $\hat{L}_{K,s}$ can be calculated as

$$\Pr\left[\hat{L}_{K,s} = \ell\right] = \Pr\left[\sum_{k=1}^{K} L_s(k) = \ell g K N\right] \tag{13}$$

Hence, the support for $\hat{L}_{K,s}$ is $\ell\{0, 1/gKN, 1/gKN, \ldots, 1\}$.

Both, $L_s(k)$ and $\hat{L}_{K,s}$ can be easily calculated by convolution. However, the pmf of $L_s^{(n)}$ is still unknown. In the following, we derive an upper and a lower bound for $L_s^{(n)}$.

We denote the lower bound for the number of lost packets as RV $\lambda_s^{(n)}$ with pmf

$$p_\lambda(x, s; n, N) = \Pr\left[L_s^{(n)} = x\right] = p_X\left(g - x, s_n; n, N\right). \quad (14)$$

That is, the lower bound is obtained by assuming that the UEs can decode as many packets as the rank of their coding matrices.

On the other hand, we denote the upper bound for the number of lost packets as RV $\Lambda_s^{(n)}$ with pmf

$$p_\Lambda(x, s; n, N) = \begin{cases} 1 & \text{if } x = g - g_n \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

That is, the upper bound on the number of lost packets is obtained by assuming that only the packets received at the cellular phase can be decoded for all the coding matrices that are not full-rank.

Finally, the maximum packet latency for a successful generation is simply given as $d_s(2g + s)$.

2) TESTBED DESCRIPTION

In this section, we introduce the testbed we developed to emulate a real scenario. This scenario consists of different UEs within the same cell, close enough so they can communicate with each other, that request the same video from a streaming server. We describe the hardware used, the NCC protocol, and the functionality of the proposed scheme.

The UEs are equipped with two wireless interfaces: an interface to request the video from the cellular communication in the cellular phase and WiFi to share and receive packets from the rest of UEs in the MC phase. Even though high-end mobile phones from the last generation can activate a feature to use the WiFi link to assist the cellular link and increase the download speed, this feature lacks flexibility in terms of configuration parameters. Hence, we use small portable computers that can carry two network interfaces, namely a WiFi interface and a cellular dongle with a SIM card for the cellular interface.

The architecture of the testbed with four users is the one previously illustrated in Fig. 1 on page 185967. We placed N Intel NUC6i5SYH (hereafter referred simply as NUCs) in our offices in Dresden, Germany, to work as clients that request a video from a video streaming server, which is located in an Amazon Web Service cloud in Frankfurt, Germany. A WiFi access point provides short-range connectivity in the MC phase. Finally, five-inch LCD screens are attached to each Intel NUCs to display the video. Table 3 lists the hardware specifications used for each element in the testbed.

The testbed works as follows. A server runs in the cloud, far away from the UEs. Each UE requests access to the server on its own. The server will grant the unicast connection and start sending the coded video stream. If there are more users connected, the server will send the packets in a round-robin manner to all the clients connected to it. Each user will receive g_n packets as in Eq. 1. The UEs will be connected in a multicast group, and a WiFi access point will provide short-range connectivity. Each UE will recode and forward the packets received to the remaining users in the MC. When the generation is complete, the UE will decode the coded packets and display them on the LCD screen.

TABLE 3. Testbed hardware specifications.

Element	Description	Specifications
User Equipment (UE)	Acts as the client who requests the video stream to the server.	Intel NUC6i5SYH 4 Core 8 GB
WiFi AP	Provides connectivity between the UEs in the MC phase	TP-Link Archer C9
Cloud Service Provider	Provides a location for the streaming server	Amazon Web Services
Cellular Dongle	Provides connectivity between the UEs and the server in the cellular phase	Huawei E3372
LCD Screens	Displays the video requested	Waveshare 5inch LCD (B)

The UEs always perform at least g_n multicast transmissions in the MC phase; this is the minimum to distribute the packets received from at the cellular phase. Then, the UEs select the number of coded redundancies based at the end of these transmissions based on the rank of their coding matrices, given by $X_0^{(n)}$ as defined in (7), and a design parameter $r \in \mathbb{R}$. Note that this approach enforces fairness in the MC, as the number of redundant transmissions from each UE depends on the packets received from the rest of the UEs in the MC. Specifically, the number of redundant multicast transmissions performed by the n-th UE whose rank at $s = 0$ is x is defined as

$$y_n(x) = \left\lfloor x \cdot \frac{\lceil gr \rceil}{g} \right\rfloor, \quad (16)$$

Note that the maximum value for $y_n(x)$ is $y_n^{\max} = \lceil gr \rceil$. Then, for a specific MC phase with y_n redundant transmissions for the n-th UE, he total number of redundant transmissions in the MC is

$$y = \sum_{n=1}^{N} y_n. \quad (17)$$

To provide some insights on the behavior of the testbed, we can define the RV $Y(N)$ as the total number of redundant transmissions in the MC phase with N UEs. To calculate the later, we first let $Y^{(n)}$ be the RV of the number of redundant multicast transmissions performed by the n-th UE, whose support is $y_n \in \mathbb{Z}$. The probability that the n-th UE performs y_n redundant transmissions is then defined as

$$\Pr\left[Y^{(n)} = y_n\right] = \Pr\left[X_0^{(n)} \leq \left\lceil \frac{g(y_n + 1)}{\lceil gr \rceil} \right\rceil - 1\right] - \Pr\left[X_0^{(n)} \leq \left\lceil \frac{gy_n}{\lceil gr \rceil} \right\rceil - 1\right] \quad (18)$$

Since the RVs $\left\{X_0^{(n)}\right\}$ are correlated, we can only define an upper bound for $Y(N)$ as

$$Y(N) = \sum_{n=1}^{N} Y^{(n)}, \tag{19}$$

which can be obtained by convolution.

B. RESULTS

In this subsection, we first present the parameters used to tune the model and the testbed, as well as discuss the reason why these parameters were chosen. Afterward, we introduce the metrics selected, and lastly discuss the results.

1) PARAMETER SELECTION

In this subsection, we introduce the parameters used for the analytical model and the testbed.

The testbed parameters were selected so that the demonstrator emulates a scenario as close as possible to reality. The parameters of the analytical model are selected to corroborate the testbed results, hence, the values correspond to the ones given in the testbed. In our previous works [13], [14], we selected a generation size g of 100 because more intuitive to understand. However, due to the on/off nature of computers, we decided to use 32 symbols (2^5) per generation. Regarding field size, we decided to use 2^8 because it reduces the chances of receiving linear dependent packets. We used a variable cloud size, ranging from 2 clients to 16 clients. We use an error rate of $\epsilon = 0.1$ since NR has error correction algorithms such as HARQ and modifies the modulation and coding scheme (MCS) if the packet error rate (PER) is higher than 0.1 [47]. However, it is expected to perceive higher losses in the testbed due to the inherent losses on the WiFi multicast channel, which make the end losses increase. In order to adjust the number of redundancies of the testbed, we define a design parameter (r) as the maximum number of redundancies per generation. This parameter defines a maximum value, however, the real value is variable and it is defined in 16. For example, with $g = 32$ and $r = 0.08$, the maximum number extra coded transmissions $y_n^{\max}$ that will be transmitted is 4. They will be inserted in between the other transmissions and then will be transmitted when the rank of the decoding matrix reaches 8, 16, 24, and 32. This means that if the rank of the matrix is 6, the number of coded redundancies transmitted is still 0, but if the rank of the matrix is 10, the number of coded redundancies transmitted is 1. In our experiments, we used different values of r, ranging from 0 to 0.2. However, the extreme cases had no interest because of their low performance, hence, we compare the results of $r = \{0.04, 0.08, 0.12, 0.16\}$. We define the payload size as the number of bytes each packet can carry. On top of the payload, different layers of encapsulation are applied, always respecting the MTU to avoid segmentation. Energy consumption models were obtained from [51] for NR and [52] for WiFi, respectively. Please note that Lauridsen *et al.* [51] models the energy consumption for LTE-A. However, the parameters in this article were selected in such a way that the specifications for LTE-A [53], [54] can be also applied to NR [46], [47] and hence, the power consumption model for LTE-A can be applied as well. Regarding the work of Sun *et al.* [52], please note that the authors assumed a negligible difference in energy per bit during transmission and reception over WiFi. Table 4 lists all the parameters used.

TABLE 4. Parameter settings.

Parameter	Symbol	Settings
Generation size	g	32 packets
Field size	q	2^8
Cloud size	N	$\{2, 4, \ldots, 16\}$ UEs
Design parameter	r	$\{0.04..0.16\}$
Packet erasure rate (PER)	ϵ	0.1
Subframe duration	d_s	1 ms
Packet length	ℓ	1470 bytes
Data rate at the NR and WiFi links	R	11.76 Mbps
Power consumption for NR reception	$P_{\text{cel,rx}}$	924.57 mW
Power consumption for WiFi transmission	$P_{\text{wifi,tx}}$	442.60 mW
Power consumption for WiFi reception	$P_{\text{wifi,rx}}$	442.60 mW

2) ANALYTICAL RESULTS

In this subsection, we show the results obtained from the analytical model.

a: SUCCESSFUL CONTENT DISTRIBUTION

We first analyze the complementary CDF (CCDF) of successful content distribution of our protocol, obtained with the analytical model with different parameters, for $N = \{2, 4, 8, 16\}$ in Fig. 4. We observe that smaller MCs need to send more coded transmissions than bigger ones if high reliability is needed, due to the fact that the frequency of transmissions in the MC phase increases with N. The analytical results for $\epsilon = 0.1$ indicate the need for around ten to eleven redundancies in order to provide a successful content distribution. As a result, we use these values to simulate variations of the coding ratio, i. e. the extra number of coded packets transmitted.

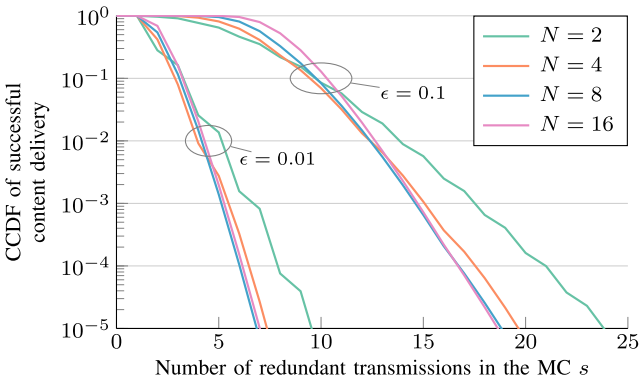
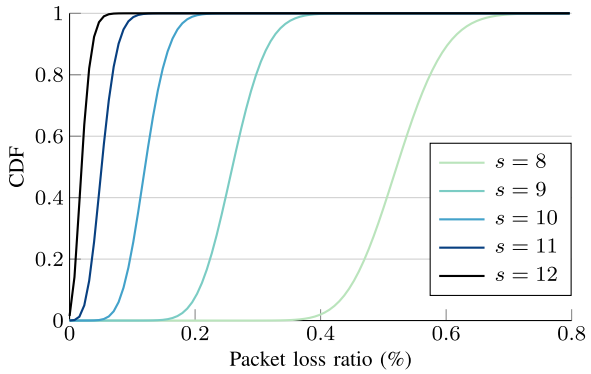
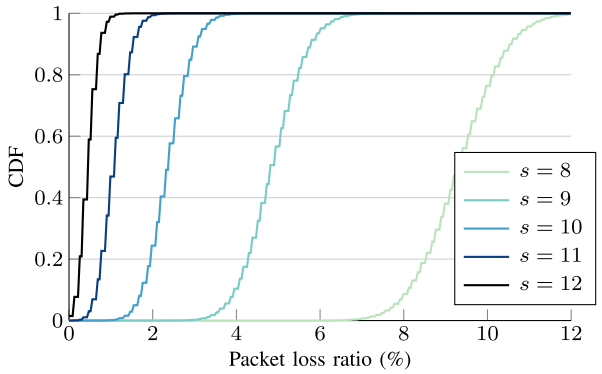


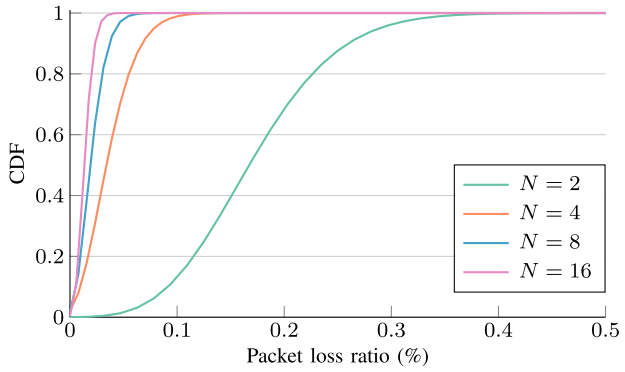
FIGURE 4. Complementary CDF (CCDF) of successful content distribution, S, for $N = \{2, 4, 8, 16\}$ in logarithmic scale.



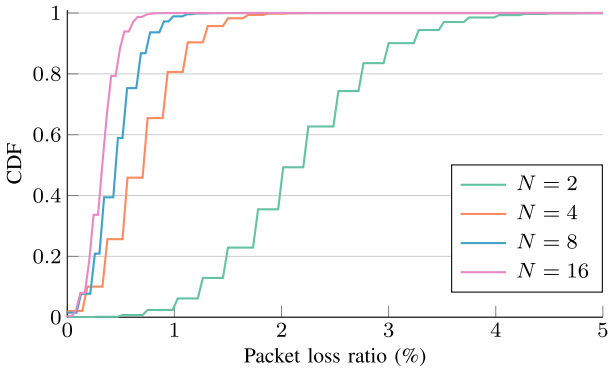
(a) Depending on the additional coded packet transmissions (lower bound)



(b) Depending on the additional coded packet transmissions (upper bound)



(c) Depending on the number of users; $s = 12$ (lower bound)



(d) Depending on the number of users; $s = 12$ (upper bound)

FIGURE 5. CDF of the packet loss probability given $\epsilon = 0.1$ and $q = 2^8$.

b: LOSS RATIO

The performance of the cloud in terms of loss ratio (the opposite of the decoding probability) for different cloud sizes and a different number of coded packet transmissions is presented in Fig. 5. In the upper figure, we change the additional coded packet transmissions while maintaining the cloud size $N = 8$. In the lower figure, we change the cloud size, while maintaining the additional coded packet transmissions $s = 11$. We observe a reduction in the packet loss probability while increasing both coded packets and users. The value selection of s and N is determined by the testbed results, in section IV-B3. More information about the explanation of the selected values and the comparison between the analytical model and the testbed can be found in the discussion section, in IV-B4.

c: ENERGY CONSUMPTION

Now we showcase the main benefit of our NCC protocol: the sharp reduction in energy consumption at the UEs. Fig. 6 plots the average energy consumption per UE for WiFi transmission, WiFi reception, and cellular reception for $\epsilon = 0.1$, as a function of the cloud size. A representation of the energy consumption in absolute values and relative values can be observed in the upper and the lower figures, respectively. We observe energy savings of more than 37% for small cloud sizes when the error rate is high. For example, the energy consumption for the direct transmission of the g packets in the cellular link is 29.586 mJ. On the other hand, the energy consumption per UE for $n = 16$ is 20.393 mJ, and it is further reduced as the cloud size increases.

Fig. 6 also shows that the main contributor to energy consumption is the energy that is used during the WiFi reception. As n increases, the WiFi antenna needs to be in an active state longer. Conversely, the reception in both cellular and WiFi has a less impact on energy consumption when the cloud size increases. The energy consumed in WiFi transmission becomes negligible for large cloud sizes, as well as the cellular reception. The energy consumed in the recoding and decoding processes was negligible and thus, it was not included in this study.

Finally, the CDF of $Y(N)$ can be observed in Fig. 7 for $N = 8$ and $r = \{0.04, 0.08, 0.12, 0.16\}$. The latter provides some insight on the upper bound of redundant packet transmissions in the MC for the given r. As can be seen, by assuming that the RVs of the rank of the coding matrices are independent, it is calculated that the number of redundant transmissions is close to $y_n^{\max}$.

3) TESTBED RESULTS

In this subsection, we plot the results obtained through the evaluation of the testbed and we validated them with the analitc model.

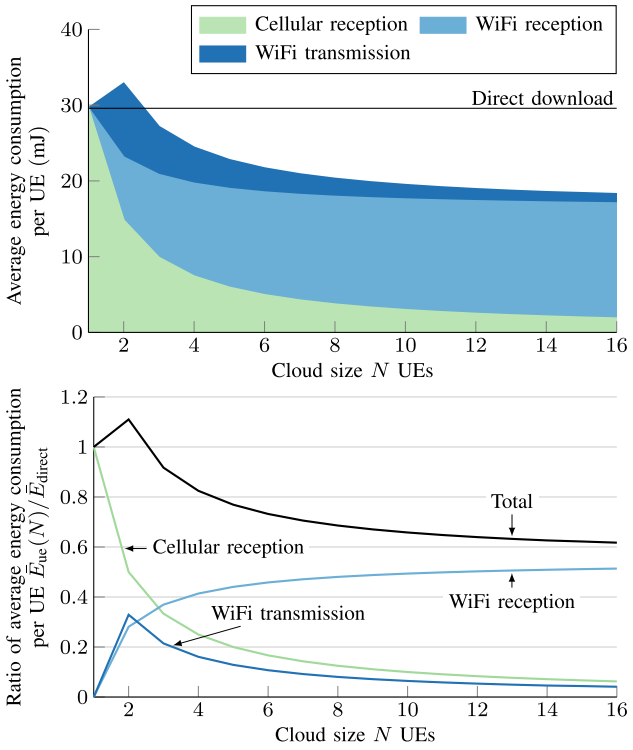


FIGURE 6. a) Average energy consumption and b) relative energy consumption per UE for different cloud sizes given $\epsilon = 0.1$.

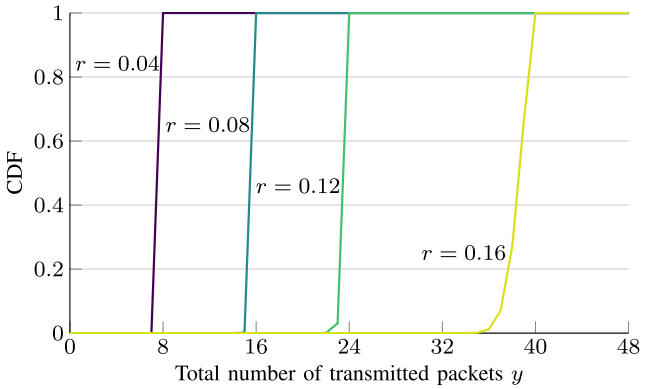
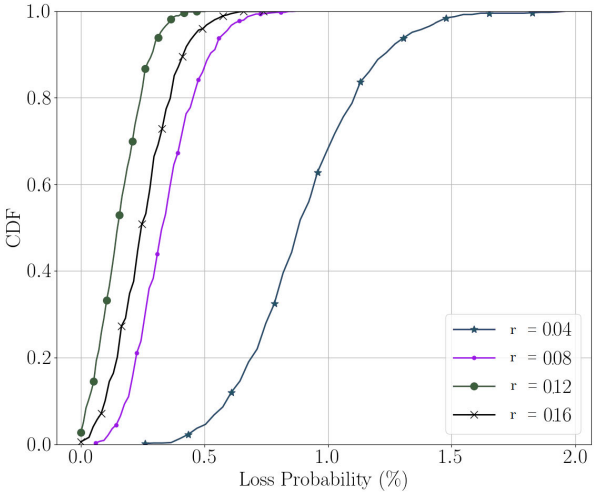
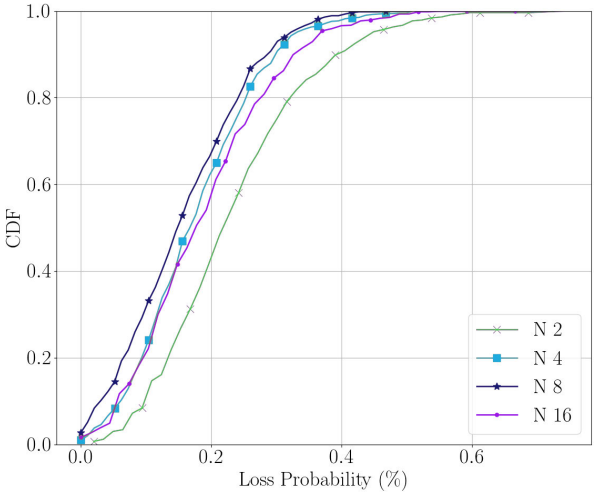


FIGURE 7. CDF of the maximum redundant multicast transmissions for a cloud size $N = 8$.



(a) Depending on the design parameter r



(b) Depending on the total number of users N

FIGURE 8. CDF of the average packet loss probability.

a: LOSS RATIO

We first focus on the decoding probability (i.e. losses). Fig. 8 shows the CDF of the average packet loss in the UEs. The upper picture shows the relation between the r and the loss probability, which is complementary to the decoding probability. The lower picture shows the optimal number of clients inside the MC. After retrieving the optimal number of clients for the parameters set, we use the testbed proposed in subsection IV-A2 to obtain different CDFs for different values of r. We observe that the values of $r = \{0.04, 0.08, 0.12, 0.16\}$ in the testbed correspond to values of $y_n^{\max} = \{2, 3, 4, 6\}$. We cannot know the real y unless we take a snapshot of the system at a specific point in time, but we can conclude that the real y is upper bounded by $y_n^{\max}$. In both cases, we can observe similar behavior in the results. This occurs due to the nature of RLNC. If r does not suffix the losses in the channel, none of the packets will be obtained (because they are coded). When the value of r gets closer to the channel erasure rate ϵ, the number of packets decoded increases drastically to the maximum decoding probability. Another important observation is that this scheme does not match the *the more, the better*, in both the design parameter r and the number of clients N in the cloud regarding losses. The performance of using $r = 0.16$ is worse than using $r = 0.12$, and a bigger number of clients ($N = 16$) does not reflect a better performance as in $N = 8$. If either r or N increases too much in such a way that there are too many packets in the channel, useful packets are delayed by useless redundancies, and by the time they arrive at the decoder is already too late.

We observe a trade-off between the useful packets sent in the multicast link, and the useless ones (due to linear dependence or network congestion).

b: ENERGY CONSUMPTION

Fig. 9 shows the energy consumption relative to the direct cellular download ($N = 1$). We observe a steady reduction in the overall energy consumption with the cloud size increase. This is mainly due to the reduction in the energy consumption of the cellular reception. The cellular power model contributes almost double to energy consumption. Hence, the reduction in energy consumption due to the cellular offload impacts more than the energy increase in the WiFi link. We observe an initial peak in the energy consumption for $N = 2$ (as well as in the analytical model). This happens because the energy overhead given by the transmission and reception of extra coded transmissions is higher than the energy savings from offloading. Conversely, this behavior changes for $N = 3$ and the following cloud sizes. In the case of $N = 8$, we maximize the energy reduction to 27% from the direct download value. Furthermore, we expect the trend in energy consumption for bigger cluster sizes to continue going down, i. e. the energy consumption for bigger cluster sizes than the ones shown in Fig. 9 will be lower.

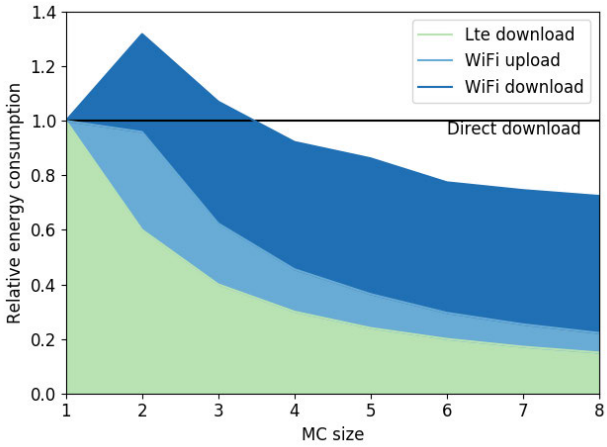


FIGURE 9. Average relative energy consumption (NCC vs. direct download).

c: PACKET LATENCY

Fig. 10 shows the CDF of average latency per packet observed in all eight UEs. Since the upper part cannot be clear, we decided to zoom it. Therefore, the colored part represents exactly the colored part of the graph (from 0.95 to 1.0). Since Fig. 10 is a little bit complex we divide it into five different parts. Hereunder a description of each part is described:

(A) The distance between the 0 ms mark and Line A represents the minimum latency the packets can have due to the transmission delay. The height of Line A represents the percentage of packets that are decoded right after being received, without waiting in the coder of the queues.

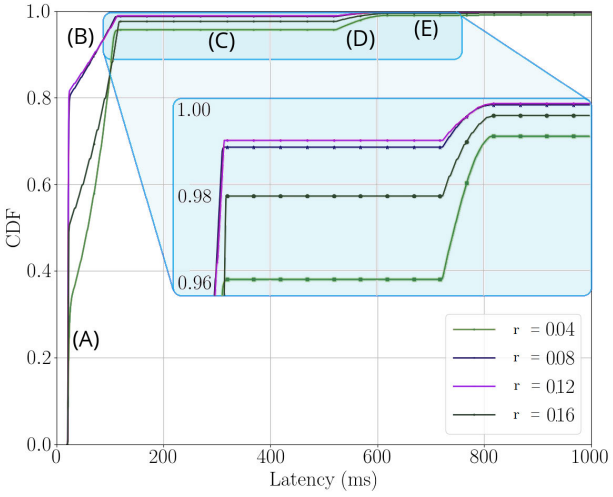


FIGURE 10. Average packet latency.

(B) Line B represents the cases where a packet is lost or corrupted, but it is recovered within its generation. The protocol we use provides in-order delivery, which means that if a packet is lost, the rest of the packets that arrive later will wait until the lost packet is recovered. Hence, the first packet sent after the lost one will have a higher delay than the last packet sent before the error was corrected. This generates a linear slope, as observed in Fig. 10. The explanation of this behavior is extended in Fig. 3 from the work of Pandi *et al.* [17].

(C) Line C shows an internal timeout. This timeout is triggered in the case the protocol cannot recover the loss. The tunable timeout starts when the first packet is decoded, and it is refreshed every time a new packet is decoded. In our protocol, we set a timeout of 500ms, which can be observed in the Line C. No packets will arrive until the timeout (500ms plus 20ms of transmission delay) is reached. We are aware that the value of this timeout is not ideal, and it depends on the losses of the channel and the architecture of the system.

(D) Line D shows the latency of the packets that arrive after an unrecovered error. If an unrecovered error occurs, the protocol will wait and store the packets that arrive during the waiting interval. The waiting interval is characterized by the timeout flag, which is our case is 500 ms. That means that, after an error, the protocol will wait for 500 ms to see if the error can be recovered. When the timeout is triggered, the protocol will forget the lost packet and the stored packets will be decoded. In Fig. 10, can be observed that line D is not linear. The explanation of this shape lies in the in-order delivery nature of the protocol. A loss can occur at the beginning, at the middle, or the end of the generation. When the

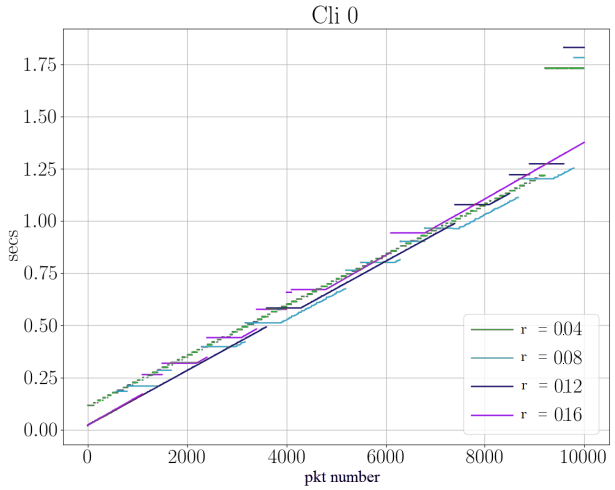


FIGURE 11. Packet decoding times depending on r for the first UE (Client 0).

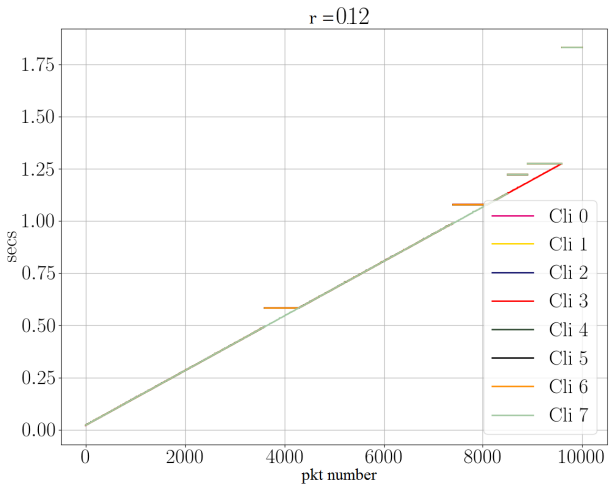


FIGURE 12. Packet decoding times for all clients with $r = 0.12$.

timeout is triggered, the packets of the generation that arrived after the loss (which is a smaller number than the generation size) will be decoded, as well as all the packets of all the generations next. This means that the number of packets that wait for longer (the ones that have the same generation of the lost packet) will be less than the number of packets that wait for less.

(E) Finally, line E shows an infinite latency for the rest of the packets. This gives us the percentage of unrecovered packets, i.e., the loss ratio.

d: PACKET SYNCHRONICITY

We discuss now how synchronous packets are decoded in the UEs. Synchronicity in streaming services is very important. For instance, two unsynchronized devices in a sports bar could trigger that a part of the audience receives the video and cheers before the other part. Moreover, security systems and ultra-reliable systems need to have consistent data in their databases, and sending the updates synchronously is vital.

Due to the constraint that one UE will need the information of the rest UEs to decode the information, we expect our system to be synchronous. The source node sends 10, 000 packets with their respective timestamps attached, marking down when the packet is decoded. Fig. 11 depicts the packet decoding time of the 10, 000 packets for 10% of the channel erasure rate in the first UE. For lower values of r, we observe a stepped line. This occurs because r is not high enough to cope with the channel erasure rate, so the decoder is continuously flushing every generation. As r increases, the graphs become more linear with eventual failures. However, the more redundancy we add in the channel, the later the packets arrive. This can be appreciated by observing the differences in the slopes of the lines. At the end of the simulation, it can be observed that the decoder has to wait for the timeout to retrieve the remaining packets in case of previous losses, due to the fact that the transmission ends. This UE can be considered as a representative of the others since the behavior is similar.

Fig. 12 showcases the synchronicity of the decoding in all UEs. It prints the same information from Fig. 11 with the optimal $r = 0.12$ for all UEs. The results show eight straight lines superimposed to each other, which demonstrates that all nodes decode the information at the same time, keeping the system synchronous. Eventual failures make some of the nodes deviate from the trend in punctual moments when a packet is lost and the decoder needs to be flushed. Please note this figure is a representative example since the behavior with different values of r is similar.

4) DISCUSSION

In this subsection, we discuss and compare the results. In the work presented above, we performed the following evaluations:

- In the *analytical model*, we obtained the optimal packets to provide successful content distribution over a certain threshold. Then, we evaluated the loss ratio of the scheme for different coded packet transmissions and different cluster sizes. Finally, we evaluated the energy savings for the proposed scheme.
- In the *testbed*, we evaluated the loss ratio of the scheme for different values of r and different cluster sizes. Then, we evaluated the energy savings for the proposed scheme. We evaluated the latency overhead caused by the introduction of our model and the packet synchronicity.

To provide a fair comparison of both evaluation methods, it is important to take into consideration the impact of the network in the testbed, for example, network congestion, interferences, WiFi scheduling, etc. We expect that the evaluation in the testbed provides worse results and that the testbed is upper bounded by the analytical model. Moreover, these irregularities in the network do not allow a fine adjustment of

the number of coded transmissions in the testbed. As a result, we overprovision the network with more redundant packets and we evaluate it for different values of r. We observe similar behavior in the shape of the loss rate plots in the analytical model (Fig. 5) and in the testbed (Fig. 8). We observe that the values of $r = \{0.04, 0.08, 0.12, 0.16\}$ in the testbed correspond to values of $y_n^{\max} = \{2, 3, 4, 6\}$. We cannot know the real y unless we take a snapshot of the system at a specific point in time, but we can conclude that the real y is upper bounded by $y_n^{\max}$. In the analytical model, the more redundant packets, the lower the loss ratio is. The same behavior occurs with different cluster sizes. However, the testbed results do not follow the same trend. We observe that after $r = 0.12$ and $N = 8$ the decoding probability decreases. The reason for this behavior lies in the addition of multiple useless packets (either by increasing N or r), which congests the network, fills the receiving queues, and ultimately forces the nodes to discard packets. This behavior is not considered in the analytical model.

Regarding the comparison in terms of energy savings, the shapes of both the analytical model (Fig. 6) and the testbed (Fig. 9) are very similar. However, the energy savings in the testbed appear to be slightly smaller in comparison to the ones observed in the model. The reason behind this behavior lies in the overprovisioning given in the testbed to deal with possible congestion bottlenecks in the short-range channel. For example, the energy overhead for $N = 2$ and the energy savings for $N = 8$ in the analytical model correspond to 15.8% and 28.17% respectively. In the case of the testbed results, these values correspond to 31.8% and 27.55%. The MSE between the analytical model and the testbed is 0.61%.

The latency evaluation (Fig. 10) in the testbed provides the minimal latency overhead of the NCC protocol, which is around 20 *ms*. However, it also shows that the impact of channel losses will highly impact the latency, increasing it to 150 *ms*. In error-prone channels such as WiFi, it is expected to receive most of the packets below the 150 *ms* mark. These latency values are observable by the human eye. Therefore, this technology is not optimal for applications that require low latency, such as VR or AR. However, 150 *ms* is acceptable in streaming services. Moreover, the synchronicity evaluation (Fig. 11 and Fig. 12) indicates a synchronous packet delivery within the MC in reliable channels, which is optimal for streaming services.

V. CONCLUSION

In this article, we presented a novel NCC protocol to disseminate data in cellular networks reliably and efficiently when co-located clients request the same content to a server. We assess the validity of our scheme with an analytical model and a testbed. The analytical model provides an upper bound for the testbed. We obtain the number of coded transmissions needed for successful content distribution given a certain QoS threshold. Then, we evaluated our system in terms of decoding ratio, energy consumption, latency, and synchronicity.

The results can be summarized as follows: Important energy savings were achieved, for example, in the case of four nodes, more than 12%, and 8% for the analytical model and the testbed, respectively. Moreover, decoding probability increases to more than 99.5% with an initial channel erasure rate $\epsilon = 0.1$. A significant decrease in cellular network usage was also achieved through NCC. The evaluation done in the testbed showed a reduction in the multicast channel error rate up to 93.75% when using our system for 16 nodes. Furthermore, the results also showed that UEs decode the information synchronously to the human eye. The main overheads of our protocol were the decrease of network throughput in situations where the WiFi channel is slower than the cellular channel and a latency increase due to the in-order-delivery nature of our RLNC protocol. However, a gNB can only serve a limited number of high data rate unicast sessions in parallel and no errors were considered in the cellular channel. Hence, in a real scenario with hundreds of UEs connected to the gNB the throughput would be higher and the error rate in the cellular phase not negligible.

There are still unknowns that were left out of the scope of this research and will be further studied. The formation of the MCs, the maximum number of nodes per cloud, the study of interferences between MCs, the mobility and handover between MCs, are some examples of them. Another challenge appears when UEs need to carry active cellular and WiFi antennas. This feature is present in the last generation of high-end mobile devices. However, we should not expect that every node in the MC is able to send and receive from two active antennas at the same time. Nevertheless, we expect that by the moment this technology is ready for deployment, UEs with two active antennas will be the state of the art in the mobile phone market.

REFERENCES

[1] J. Rodriguez, A. Radwan, C. Barbosa, F. H. Fitzek, R. A. Abd-Alhameed, J. M. Noras, S. M. Jones, I. Politis, P. Galiotos, G. Schulte, and A. Rayit, "Secret-secure network coding for reduced energy next generation mobile small cells," in *Proc. ITA Conf.*, Sep. 2017, pp. 329–333.
[2] Cisco Technologies, "Cisco visual networking index: Forecast and methodology, 2017 to 2022," Cisco Technol., Cisco, San Francisco, CA, USA, Cisco Annu. Internet Rep. (2017Ű2022) White Paper, Nov. 2018.
[3] Cisco Technologies, "Cisco visual networking index: Forecast and methodology, 2016 to 2021," Cisco Technol., Cisco, San Francisco, CA, USA, Cisco Annu. Internet Rep. (2018Ű2023) White Paper, Mar. 2017.
[4] P. Danzi, A. E. Kalør, R. B. Sørensen, A. K. Hagelskjær, L. D. Nguyen, Ä. Stefanović, and P. Popovski, "Communication aspects of the integration of wireless iot devices with distributed ledger technology," Dept. Electron. Syst., Aalborg Univ., Aalborg, Denmark, 2019.
[5] "LTE multimedia broadcast multicast services (MBMS)," Viavi Solutions, San Jose, CA, USA, White Paper, 2015.
[6] Operating Eurovision and Euroradio, "Delivery of broadcast content over lte networks," EBU, Grand-Saconnex, Switzerland, Delivery Bradcast Content LTE Netw., Tech. Rep. 027, Jul. 2014.
[7] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
[8] T. Ho, M. Medard, J. Shi, M. Efiros, and D. R. Karger, "On randomized network coding," in *Proc. 41st Annu. Allerton Conf. Commun., Control, Computi*, 2003, pp. 11–20.
[9] F. H. Fitzek and M. D. Katz, *Exploiting Distributed Resources in Wireless, Mobile and Social Networks*. Hoboken, NJ, USA: Wiley, 2014.

[10] M. V. Pedersen and F. H. P. Fitzek, "Mobile clouds: The new content distribution platform," *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1400–1403, May 2012.

[11] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[12] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[13] I. Leyva-Mayorga, R. Torre, S. Pandi, G. T. Nguyen, V. Pla, J. Martinez-Bauset, and F. H. P. Fitzek, "A network-coded cooperation protocol for efficient massive content distribution," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[14] I. Leyva-Mayorga, R. Torre, V. Pla, S. Pandi, G. T. Nguyen, J. Martinez-Bauset, and F. H. P. Fitzek, "Network-coded cooperation and multi-connectivity for massive content delivery," *IEEE Access*, vol. 8, pp. 15656–15672, 2020.

[15] D. E. Lucani, M. Medard, and M. Stojanovic, "Systematic network coding for time-division duplexing," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 2403–2407.

[16] R. Torre, S. Pandi, and F. H. P. Fitzek, "Network-coded multigeneration protocols in heterogeneous cellular networks," in *SpringerLink Digital Library*. Faro, Portugal: Springer, 2018.

[17] S. Pandi, F. Gabriel, J. A. Cabrera, S. Wunderlich, M. Reisslein, and F. H. P. Fitzek, "PACE: Redundancy engineering in RLNC for low-latency communication," *IEEE Access*, vol. 5, pp. 20477–20493, 2017.

[18] Y. Keshtkarjahromi, H. Seferoglu, R. Ansari, and A. Khokhar, "Device-to-device networking meets cellular via network coding," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 370–383, Feb. 2018.

[19] B. Li, H. Li, X. Li, H. Jiang, W. Tang, and S. Li, "Hybrid multicast and device-to-device communications based on adaptive random network coding," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2071–2083, Mar. 2019.

[20] Y. N. Shnaiwer, S. Sorour, T. Y. Al-Naffouri, and S. N. Al-Ghadhban, "Opportunistic network coding-assisted cloud offloading in heterogeneous fog radio access networks," *IEEE Access*, vol. 7, pp. 56147–56162, 2019.

[21] A. Le, L. Keller, H. Seferoglu, B. Cici, C. Fragouli, and A. Markopoulou, "MicroCast: Cooperative video streaming using cellular and local connections," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2983–2999, Oct. 2016.

[22] B. Zhang, Z. Liu, S.-H.-G. Chan, and G. Cheung, "Collaborative wireless freeview video streaming with network coding," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 521–536, Mar. 2016.

[23] L. Wang, Z. Yang, L. Xu, and Y. Yang, "NCVCS: Network-coding-based video conference system for mobile devices in multicast networks," *Ad Hoc Netw.*, vol. 45, pp. 13–21, Jul. 2016.

[24] J. Park, J.-N. Hwang, Q. Li, Y. Xu, and W. Huang, "Optimal DASH-multicasting over LTE," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4487–4500, May 2018.

[25] R. Torre, C. C. Sala, S. Pandi, H. Salah, G. T. Nguyen, and F. H. P. Fitzek, "A random linear network coded HARQ solution for lossy and high-jitter wireless networks," in *Proc. IEEE Global Commun. Conf., Commun. QoS, Rel. Modeling (IEEE Globecom CQRM)*, Taipei, Taiwan, Dec. 2020.

[26] R. Torre and F. H. P. Fitzek, "A study on data dissemination techniques in heterogeneous cellular networks," in *Broadband Communications, Networks, and Systems*. Faro, Portugal: Springer, 2019.

[27] F. Fitzek, M. Katz, and Q. Zhang, "Cellular controlled short-range communication for cooperative P2P networking," *WirelessWorld Res. Forum*, vol. 48, no. 1, pp. 141–155, 2006.

[28] F. Rossetto and M. Zorzi, "Mixing network coding and cooperation for reliable wireless communications," *IEEE Wireless Commun.*, vol. 18, no. 1, pp. 15–21, Feb. 2011.

[29] Y. N. Shnaiwer, S. Sorour, P. Sadeghi, N. Aboutorab, and T. Y. Al-Naffouri, "Network-coded macrocell offloading in femtocaching-assisted cellular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2644–2659, Mar. 2018.

[30] M. Di Renzo, M. Iezzi, and F. Graziosi, "On diversity order and coding gain of multisource multirelay cooperative wireless networks with binary network coding," *IEEE Trans. Veh. Technol.*, vol. 62, no. 3, pp. 1138–1157, Mar. 2013.

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

[32] P. Chau, T. Duc Bui, Y. Lee, and J. Shin, "Efficient data uploading based on network coding in LTE-advanced heterogeneous networks," in *Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2017, pp. 252–257.

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

[34] A. Tassi, C. Khirallah, D. Vukobratovic, F. Chiti, J. S. 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.

[35] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, "Microcast: Cooperative video streaming on smartphones," in *Proc. MobiSys*, Low Wood Bay, U.K., Jun. 2012, pp. 57–70.

[36] Y. Yan, B. Zhang, and C. Li, "Network coding aided collaborative real-time scalable video transmission in D2D communications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6203–6217, Jul. 2018.

[37] A. Antonopoulos, J. Bastos, and C. Verikoukis, "Analogue network coding-aided game theoretic medium access control protocol for energy-efficient data dissemination," *IET Sci., Meas. Technol.*, vol. 8, no. 6, pp. 399–407, Nov. 2014.

[38] A. Antonopoulos and C. Verikoukis, "Multi-player game theoretic MAC strategies for energy efficient data dissemination," *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 592–603, Feb. 2014.

[39] L. Aymen, B. Ye, and T. M. T. Nguyen, "Offloading performance evaluation for network coding-based cooperative mobile video streaming," in *Proc. 7th Int. Conf. Netw. Future (NOF)*, Nov. 2016, pp. 1–5.

[40] R. Zhang, D. Ban, B. Li, and Y. Jiang, "The cooperative multicasting based on random network coding in wireless networks," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–5.

[41] M. H. Firooz and S. Roy, "Data dissemination in wireless networks with network coding," *IEEE Commun. Lett.*, vol. 17, no. 5, pp. 944–947, May 2013.

[42] R. Torre, H. Salah, G. T. Nguyen, and F. H. P. Fitzek, "Evaluating the latency overhead of network-coded cooperative networks for different cloud sizes," in *Proc. IEEE 2nd 5G World Forum (5GWF)*, Sep. 2019, pp. 253–258.

[43] R. Torre, S. Pandi, G. T. Nguyen, and F. H. P. Fitzek, "Optimization of a random linear network coding system with Newton method for wireless systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[44] E. Tsimbalo, A. Tassi, and R. J. Piechocki, "Reliability of multicast under random linear network coding," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2547–2559, Jun. 2018.

[45] S. Pandi, R. T. Arranz, G. T. Nguyen, and F. H. P. Fitzek, "Massive video multicasting in cellular networks using network coded cooperative communication," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2018, pp. 1–2.

[46] *Physical Channels and Modulation*, Standard TS38.211., 3GPP, Jul. 2020.

[47] *Physical Layer Procedures*, Standard TS38.214., 3GPP, Jul. 2020.

[48] M. Toemoeskoezi, F. H. Fitzek, D. E. Lucani, M. V. Pedersen, P. Seeling, and P. Ekler, "On the packet delay characteristics for serially-connected links using random linear network coding with and without recoding," in *Proc. Eur. Wireless 21th Eur. Wireless Conf.*, May 2015, pp. 1–6.

[49] G. Wunder, M. Kasparick, S. ten Brink, F. Schaich, T. Wild, I. Gaspar, E. Ohlmer, S. Krone, N. Michailow, A. Navarro, G. Fettweis, D. Ktenas, V. Berg, M. Dryjanski, S. Pietrzyk, and B. Eged, "5GNOW: Challenging the LTE design paradigms of orthogonality and synchronicity," in *Proc. IEEE 77th Veh. Technol. Conf. (VTC Spring)*, Jun. 2013, pp. 1–5.

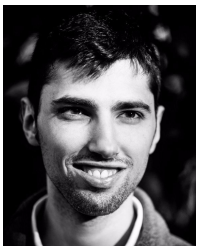
[50] Z. Chang, S. Zhou, T. Ristaniemi, and Z. Niu, "Collaborative mobile clouds: An energy efficient paradigm for content sharing," *IEEE Wireless Commun.*, vol. 25, no. 2, pp. 186–192, Apr. 2018.

[51] M. Lauridsen, L. Noël, T. B. Sørensen, and P. Mogensen, "An empirical LTE smartphone power model with a view to energy efficiency evolution," *Intel Technol. J.*, vol. 18, no. 1, pp. 172–193, Mar. 2014.

[52] L. Sun, H. Deng, R. K. Sheshadri, W. Zheng, and D. Koutsonikolas, "Experimental evaluation of WiFi active Power/Energy consumption models for smartphones," *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 115–129, Jan. 2017.

[53] *Physical Channels and Modulation*, Standard TS36.211., 3GPP, Apr. 2017.

[54] *Physical Layer Procedures*, Standard TS36.213., 3GPP, May 2016.



ROBERTO TORRE (Member, IEEE) received the bachelor's and master's degrees in telecommunications from the University of Valladolid, Spain, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Deutsche Telekom Chair of Communication Networks, TU Dresden, working on the Europa SECRET Project. He has been working with the chair since October 2016. His research interests include neural networks, GPU computing, network coding, and mesh networks. His current research interests include mobile edge computing, software defined networking, and network function virtualization.

ISRAEL LEYVA-MAYORGA (Member, IEEE) received the M.Sc. degree (Hons.) in mobile computing systems from the Instituto Politécnico Nacional, Mexico, in 2014, and the Ph.D. degree *(cum laude)* in telecommunications from the Universitat Politècnica de València, Spain, in 2018. He was a Visiting Researcher with the Department of Communications, UPV, in 2014, and the Deutsche Telekom Chair of Communication Networks, Technische Universität Dresden, Germany, in 2018. He is currently a Postdoctoral Researcher with the Department of Electronic Systems, Connectivity Section (CNT), Aalborg University, Denmark. His research interests include 5G and beyond, satellite networks, random access protocols, and massive machine-type communications.

SREEKRISHNA PANDI (Graduate Student Member, IEEE) was born in Chennai, India. He received the B.E. degree in electronics and instrumentation engineering from Anna University, in 2013, and the master's degree in nanoelectronic systems from Technical University (TU) Dresden, Dresden, Germany, in 2015, where he is currently pursuing the Ph.D. degree with the Deutsche Telekom Chair of Communication Networks.

HANI SALAH received the B.Sc. degree in computer systems engineering from Palestine Polytechnic University, the M.Sc. degree in internetworking from the Royal Institute of Technology (KTH), Sweden, and the Ph.D. (Dr.Ing.) degree in computer science from TU Darmstadt, Darmstadt, Germany, in 2015, specialized in measurements, performance, and security of large-scale distributed systems, under supervision of Prof. Dr. Thorsten Strufe. From September 2016 to June 2018, he has worked as a Lecturer and a Researcher with Palestine Polytechnic University, Hebron, Palestine. He has worked as a Research and Development Employee at TU Darmstadt and flexOptix GmbH, DietzeNBach, Germany, for 1.5 years, a Research Assistant at TU Darmstadt for four years, and a Lecturer and a Technical Assistant at Palestine Polytechnic University for about 5.5 years. He is currently a Senior Researcher with the Deutsche Telekom Chair of Communication Networks, TU Dresden, Dresden, Germany. His current research interests include distributed systems and modern networking technologies (SDN, NFV, and ICN) with emphasis on their performance, security, and monitoring.

GIANG T. NGUYEN (Member, IEEE) received the M.Eng. degree in telecommunications from the Asian Institute of Technology (AIT), Thailand, in 2007, and the Ph.D. (Dr.Ing.) degree in computer science from the Technical University of Dresden, Germany, in 2016. He is currently a Senior Researcher with the Deutsche Telekom Chair of Communication Networks, Technical University of Dresden. His research interests include 5G and highly adaptive and energy-efficient computing (HAEC). His research interests also include network coding and low-latency aspects of network function virtualization and software-defined networking.

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