Network Coding for Cooperation in Wireless Networks

Ph.D. Dissertation

Néstor Javier Hernández Marcano

Dissertation submitted October 15, 2016

Thesis submitted:	October 15, 2016
PhD Supervisor:	Assoc. Prof. Daniel E. Lucani Aalborg University
Assistant PhD Supervisor:	Dr. Janus Heide Steinwurf ApS, Denmark
Assistant PhD Supervisor:	Prof. Frank H.P. Fitzek Technische Universität Dresden, Germany
PhD Committee:	Assoc. Prof. Troels Bundgaard Sørensen, Aalborg University Assoc. Prof. Ming Xiao, KTH Royal Institute of Technology, Sweden Prof. Frank Yong Li, University of Adger, Norway
PhD Series:	Faculty of Engineering and Science, Aalborg University

ISSN: xxxx-xxxx
ISBN: xxx-xx-xxxx-xxx-x

Published by:
Aalborg University Press
Skjernvej 4A, 2nd floor
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 20XX

Normalsider: XXX sider (á 2.400 anslag inkl. mellemrum).
Standard pages: XXX pages (2,400 characters incl. spaces).

Abstract

English abstract

Resumé

Danish Abstract

Thesis Details

Thesis Title:	Network Coding for Cooperation in Wireless Networks
Ph.D. Student:	Néstor Javier Hernández Marcano
Supervisors:	Assoc. Prof. Daniel R. Lucani, Aalborg University Dr. Janus Heide, Steinwurf ApS Prof. Frank H.P. Fitzek, Aalborg University

The main body of this thesis consist of the following papers:

[A] Néstor J. Hernández Marcano, Janus Heide, Daniel E. Lucani, Frank H.P. Fitzek, "On the Throughput and Energy Benefits of Network Coded Cooperation", *2014 IEEE 3rd International Conference on Cloud Networking (IEEE Cloudnet)*, pp. 138–142, 2014.

[B] Néstor J. Hernández Marcano, Janus Heide, Daniel E. Lucani, Frank H.P. Fitzek, "Throughput, Energy and Overhead of Multicast Device-to-Device Communications with Network Coded Cooperation", *Wiley Transactions on Emerging Telecommunications Technologies (former European Transactions on Telecommunications). Special Issue: Emerging Topics in Device to Device Communications as Enabling Technology for 5G Systems*, pp. 1–17, 2016.

[C] Néstor J. Hernández Marcano, Péter Vingelmann, Morten V. Pedersen, Janus Heide, Daniel E. Lucani, Frank H.P. Fitzek, "Getting Kodo: Network Coding for the ns-3 Simulator", *ACM The Workshop in ns-3 (WNS3)*, pp. 101–107, 2016.

[D] Néstor J. Hernández Marcano, Janus Heide, Daniel E. Lucani, Frank H.P. Fitzek, "On Transmission Policies for Multihop Device-to-Device Communications with Network Coded Cooperation", *IEEE 22th International Conference on European Wireless*, pp. 350–354, 2016.

In addition to the main papers, the following publications have also been made:

[1] Néstor J. Hernández Marcano, Janus Heide, Daniel E. Lucani, Frank H.P. Fitzek, "On the Overhead of Telescopic Codes in Network Coded Cooperation", *IEEE 3rd International Conference on Cloud Networking (IEEE Cloudnet)*, pp. 138–142, 2014.

[2] Néstor J. Hernández Marcano, Jeppe Pihl, Janus Heide, Jeppe Krigslund, Péter Vingelmann, Morten V. Pedersen, Daniel E. Lucani, Frank H.P. Fitzek, "Wurf.it: A Network Coding Reliable Multicast Streaming Solution - NS-3 Simulations and Implementation", *ACM The Workshop in ns-3: Posters, Demos and Short-Talks Session (WNS3)*, Available online at the ns-3 website: https://www.nsnam.org/workshops/wns3-2016/posters/hernandez-demo-paper.pdf, 2016.

[3] Néstor J. Hernández Marcano, Chres W. Sørensen, Juan A. Cabrera Guerrero, Simon Wunderlich, Daniel E. Lucani, Frank H.P. Fitzek, "On Goodput and Energy Measurements of Network Coding Schemes in the Raspberry Pi" *Accepted to MDPI, Journal of Electronics. Special Issue for the Raspberry Pi (Indexed by Elsevier Scopus and Thomson-Reuters ESCI Web of Science).*

[4] Chres W. Sørensen, Néstor J. Hernández Marcano, Juan A. Cabrera Guerrero, Simon Wunderlich, Daniel E. Lucani, Frank H.P. Fitzek, "Easy as Pi: A Network Coding Raspberry Pi Testbed" *Accepted to MDPI, Journal of Electronics. Special Issue for the Raspberry Pi (Indexed by Elsevier Scopus and Thomson-Reuters ESCI Web of Science).*

This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Contents

Abstract iii

Resumé v

Thesis Details vii

Preface xiii

I Introduction 1

Introduction 3

1 Introduction 3

1.1 Including the Bibliographies 3

1.2 Formatting Guidelines 3

2 Section 2 name 6

2.1 Examples 7

2.2 How does Subsections and Subsubsections Look? 7

3 Conclusion 7

References 8

II Papers 9

A On the Throughput and Energy Benefits of Network Coded Cooperation 11

B Throughput, Energy and Overhead of Multicast Device-to-Device Communications with Network Coded Cooperation 19

C Getting Kodo: Network Coding for the ns-3 Simulator 39

D On Transmission Policies in Multihop Device-to-Device Communications with Network Coded Cooperation 49

Todo list

🟧 I think this word is mispelled 6

Figure: We need a figure right here! 6

🟧 Is it possible to add a subsubparagraph? 7

🟩 I think that a summary of this exciting chapter should be added. . . 7

Preface

Here is the preface

Name
Aalborg University, October 6, 2016

Part I

Introduction

Introduction

1 Introduction

I do not want to write a lengthy introduction on how to use this template. If you are familiar with LaTeX 2ε, it should be fairly simple to use. Note, however, that you should pay attention to one detail regarding the compilation of the various bibliographies in the template.

1.1 Including the Bibliographies

I have received numerous emails asking me why their bibliographies for the introduction and the individual papers are not included in the thesis when they compile it. The reason is always that they blindly trust that their LaTeX 2ε editor of choice can generate the bibliographies for them automatically. However, this is not true for this thesis template since the introduction and the included papers have their own separate bibliography.

Normally, an editor will run `bibtex master` in the background when asked to compile the bibliography for a master document named `master.tex`. Since the template uses the package `chapterbib` for generating the multiple bibliographies, the editor should instead run `bibtex introduction/introduction` to generate the bibliography for the introduction, `bibtex papers/paperA/paperA` to generate the bibliography for paper A, and so on. You can see how I compile the template by looking in the included `Makefile`. Note that you can run this `Makefile` on both OS X and on a Linux distribution. On Windows, however, you will have to either translate the `Makefile` into a bat-file, install Cygwin, or make your LaTeX 2ε editor run the same commands.

1.2 Formatting Guidelines

This template should comply with the formatting guidelines for an Aalborg University Ph.D. thesis. However, it is *your* responsibility to ensure that this is also the case. You can see the guidelies below. Please note that these guidelines have not been formulated by the Aalborg University Press so any questions should be directed to them. See the email address below.

Minimum requirements regarding thesis format before submission

- Page format: 170 mm x 240 mm.
- Margins /top, bottom, right, left): 25 mm.
- Page number and header: Placed centrally top and bottom.
- All pages in the pdf must be placed vertically (portrait), otherwise they cannot be printed.
- Tables, pictures etc. must be rotated clockwise/counter clockwise for portrait format. Take care that the page number are in the right place afterwards!
- Font type: If fonts are not embedded use only Windows/MAC standard fonts (e.g. Arial, Verdana, Times New Roman, Minion Pro, Baskerville, Garamond etc.).
- Colophon: Fill in as much as you can. The University press will fill in Serial title, ISBN, ISSN and reviewing committee.
- You are welcome to include an author CV, preferably with picture (see template). Maximum 1350 key-strokes (no spacing).
- You can also include a back cover text when you submit the thesis in Pure/VBN if you want. Maximum 1200 keystrokes (no spacing).

REMEMBER to open your pdf and do a thorough check of the thesis page by page before you submit it in VBN. Once you have submitted your thesis in VBN, you must send an email to Aalborg University Press (aauf@forlag.aau.dk) with the following information:

- Your choice of cover.
- If you want your own pictures on the cover, please attach 1–2 pictures size 17.5 cm x 8.5 cm in 300 dpi/ppi.
- Information regarding the number of copies of the thesis. Remember to clear the number with your department and PhD School.

Formatting tips

- Body text: Times New Roman 10pt / 12pt spacing.
- Headline level 1: Arial bold 18pt / 21 pt - Capital letters.
- Headline level 2: Arial bold 11pt / 21 pt - Capital letters.
- Headline level 3: Arial bold 10pt / 21 pt - Capital letters.
- Quotations: Times New Roman Italic 10pt / 12pt.

Please notice that chapter must start on a right-hand page.

Hyphenation

Please do a thorough check of the thesis page by page to ensure a proper hyphenation. For additional info see wikibooks.

CMYK color model

All text and figures should be in the CMYK color model.

Images/photos Images in RGB should be converted to CMYK using for instance rgb2cmyk.org or convert_cmyk.bat. For more info see stackexchange: how-do-i-make-sure-images-are-cmyk. Note that PNG or Portable Network Graphic format is a graphic file format that uses lossless compression algorithm to store raster images. It is frequently used as web site images rather than printing as it supports only the RGB color model. So CMYK color images cannot be saved as PNG image.

Text in color It is recomended not to use colors for text. However, if you want to use colors they must use the CMYK model. If you use fontspec, for font definitions, using the [Color=...] options as fontspec doesn't support CMYK colours. instead you will need to use the
color command from xcolor instead. For more information see for instance stackexchange: pdf-colour-model-and-latex.

Black colors Please ensure that black colors are truly black and not a dark gray tone, i.e. black text including headers and footers should be 100 % K(eycolor). This goes for both text and figures.

Include fonts in the PDF

Please make sure that all fonts are embedded in the PDF. To check this using the normal Adobe Reader (or Foxit if you prefer) select File - Properties, on the resulting Dialog choose the Font tab. You will see a list of fonts. The ones that are embedded will state this fact in ( ) behind the font name. For more information see stackoverflow: how-to-find-out-which-fonts-are-referenced-and-which-are-embedded-in-a-pdf.

If the fonts are not embeded: Using pdflatex to create a PDF from LaTeX make sure pdftexDownloadBase14 is set to true in the updmap config file. For more information see chapter "3. LaTeX => PDF" in EmbedLaTeX-fonts.pdf.

Additional information on the subject:

- how-to-embed-fonts-at-compile-time-with-pdflatex
- getting-pdflatex-to-embed-all-fonts.
- embedding-fonts-for-ieee.

How to embed missing fonts in pdf?

If you need to include a pdf in your thesis where the fonts are missing please have a look at some of the resources below.

- Adobe: Reembedding fonts in a pdf
- prepressure.com: Fonts in PDF files
- Stackoverflow: how-do-i-embed-fonts-in-an-existing-pdf

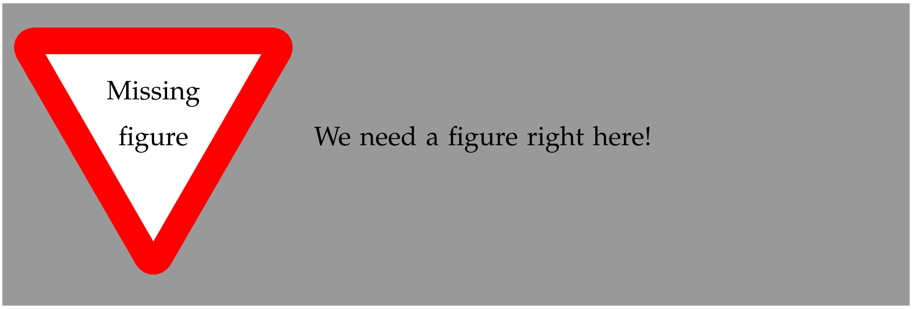
Graphics

- Make sure that no lines have a thickness less than 0.15 points.
- Images should be approximately 300 dpi/ppi but no more than 400 dpi/ppi.

2 Section 2 name

I think this word is mispelled

Here is section 2. If you want to leearn more about LaTeX 2$_{\varepsilon}$, have a look at [1], [2] and [3].



We need a figure right here!

2.1 Examples

You can also have examples in your document such as in example 2.1.

> **Example 2.1 (An Example of an Example)**
> Here is an example with some math
>
> $$0 = \exp(i\pi) + 1\,. \tag{1}$$
>
> You can adjust the colour and the line width in the `macros.tex` file.

2.2 How does Subsections and Subsubsections Look?

Well, like this

This is a Subsubsection

and this.

A Paragraph You can also use paragraph titles which look like this.

A Subparagraph Moreover, you can also use subparagraph titles which look like this. They have a small indentation as opposed to the paragraph titles.

Is it possible to add a subsubparagraph?

I think that a summary of this exciting chapter should be added.

3 Conclusion

In case you have questions, comments, suggestions or have found a bug, please do not hesitate to contact me. You can find my contact details below.

Jesper Kjær Nielsen
jkn@es.aau.dk
http://kom.aau.dk/~jkn
Niels Jernes Vej 12, A6-302
9220 Aalborg Ø

References

[1] L. Madsen, “Introduktion til LaTeX,” http://www.imf.au.dk/system/latex/bog/, 2010.

[2] T. Oetiker, “The not so short a introduction to LaTeX2e,” http://tobi.oetiker.ch/lshort/lshort.pdf, 2010.

[3] F. Mittelbach, *The LATEX companion*, 2nd ed. Addison-Wesley, 2005.

Part II

Papers

Paper A

On the Throughput and Energy Benefits of Network Coded Cooperation

Néstor J. Hernández Marcano, Janus Heide, Daniel E. Lucani,
Frank H.P. Fitzek

The paper has been published in the
Proceedings of the 2014 IEEE 3rd International Conference on Cloud Networking (IEEE Cloudnet), pp. 138–142, 2014.

On the Throughput and Energy Benefits of Network Coded Cooperation

Néstor J. Hernández Marcano*, Janus Heide*, Daniel E. Lucani† and Frank H.P. Fitzek†
*Steinwurf ApS, Aalborg, Denmark. Mail: {nestor|janus}@steinwurf.com
†Department of Electronic Systems. Aalborg University, Denmark. Mail: {nh|del|ff}@es.aau.dk

***Abstract*—Cooperative techniques in wireless mobile networks typically leverage short-range communication technologies, e.g., WiFi, to allow data exchange between devices forming a mobile cloud. These mobile clouds have been considered as a key to reduce the cost of multicast services for the network operators as well as a means to deliver a better quality to the users. In fact, LTE-A includes Device-to-Device communication capabilities to enable such a direct communication between devices. The underlying assumption for attaining the throughput gains in mobile clouds is that the communication rate between devices is typically larger than the data rate from the base station to a receiver. However, while the data rates on cellular technologies have been steadily increasing, short-range communication speeds have remained largely unchanged calling into question these assumptions. This work's goal is to assess the operating regions where the use of cooperation results in a higher throughput and/or energy saving. We consider a multicasting and a cooperative scheme with network coded mechanisms, as they typically outperform uncoded approaches. Our analysis and numerical results show that gains of several fold can be attained even if the data rate of the short-range technologies is moderately larger, e.g., 2x larger, than the cellular link data rate.**

Keywords*—4G, cooperation, energy, mobile clouds, network coding, throughput

I. Introduction

Data traffic is expected to grow by an order of magnitude for wireless mobile devices due largely to video services. This presents significant technical challenges for mobile operators to provide high quality of experience to the network users at high data rates, with low delay, while maintaining a low energy consumption in the mobile devices. Thus, mechanisms that can offload infrastructure networks have gathered significant interest from both academia and industry.

To address some of these challenges in multicast transmissions, wireless cooperation between receivers leveraging a separate communication channel to exchange missing data packets (instead of requesting them directly from the cellular infrastructure as in Fig. 1) are known to provide large gains over simply broadcasting the data [1].

This potential for cooperation has resulted in the inclusion of device-to-device (D2D) communication in the 3rd Generation Partnership Project (3GPP) standardization efforts. Beyond offloading the network operator, these cooperative techniques can result in increased reliability, coverage extension, and even increased throughput to end receivers.

In this context, network coding (NC) [2] provides not only a faster and more efficient approach to broadcast the data to the users, as shown by [3], but it simplifies the cooperation process since (i) devices need not know the specific packets missing at other devices, only the number of linear combinations available; and (ii) transmissions from a single device during the cooperation process can be used to *heal* a packet at multiple receivers, i.e., each transmission in the cooperation phase can have a larger impact for the end-receivers. This intuition has been exploited in previous work ranging from analysis to optimal policies and practical mechanisms, e.g., [4], [5].

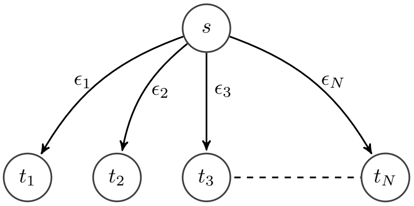


Fig. 1: Single source multiple sink topology of N receivers

However, the conventional wisdom of such cooperative techniques is that the secondary channel is considerably faster than the channel to the base station. Although this assumption was reasonable in the context of 2G and 3G communications, the much higher data rates achievable in LTE-A (4G) calls this assumption into question. The reason is that alternative technologies for device to device communications, e.g., WiFi, may no longer be faster than LTE-A as their data rates have stayed moderately constant over time. Additionally, if the devices cooperate using D2D of LTE-A the data rate for cooperation will also be limited by the common channel and could be the same data rate in some cases. Thus, the goal of this paper is to revisit the problem of device cooperation focusing on the specific regions of operation where it can bring gains in throughput and energy.

Some of the analysis of mean performance for cooperative schemes has been carried out before, e.g., [4], [6], however this paper provides an in-depth study of the distributions of the number of transmitted packets of different broadcast and cooperative schemes with NC (Section II). Leveraging these distributions, we derive the throughput and energy performance of the various schemes (Section III). In particular, we introduce the natural concept of stable throughput for cooperative schemes. To the best of our knowledge, this has not been considered before because of the conventional assumption that the cellular data rate is the bottleneck in the communication process. Our analysis allows us to determine the regions where cooperation provides gains over broadcasting (Section IV).

978-1-4799-2730-2/14/$31.00 ©2014 IEEE

II. Model and Transmission Schemes

We consider the problem of reliably transmitting a batch of packets from a source to N receivers using various transmission schemes. The batch constitutes a generation of g packets which we code using random linear network coding (RLNC) with field size q [7]. We assume independent heterogeneous erasure rates on the links from the source to the receivers, $\epsilon_j, j = [1, N]$, to derive the expressions.

We review two transmission schemes namely broadcast RLNC and cooperation with NC. For the cooperative scheme the receivers communicate among themselves to locally repair missing packets. We model the number of transmission as random variables using the geometric distribution as a building block to derive the probability mass functions (pmf) in order to obtain a complete description of the transmission process. We first give a new expression of the pmf for RLNC with no erasures and then compute the pmf for the schemes.

A. RLNC Probabiblity Mass Function

Consider the case of a single source - destination pair without erasures. Let $\mathbb{T}_{RLNC,i}$ be a r.v. for the number of transmissions needed to receive a linearly independent (l.i.) coded packet in a stage of RLNC, i.e. once $i-1$ l.i. packets have been received. This is a geometric distribution with success probability given by $p_i = 1 - q^{-g+(i-1)}, i \in [1, g]$. Following, $\mathbb{T}_{RLNC} = \sum_{i=1}^{g} \mathbb{T}_{RLNC,i}$ transmissions are necessary to decode g packets. Therefore, the code pmf can be computed using a characteristic function approach to make the analysis more tractable. Consequently, we obtain the pmf for RLNC without erasures in (1), where $P_g = \prod_{i=1}^{g} p_i = \Pr[\mathbb{T}_{RLNC} = g]$ is the probability of decoding in exactly g transmissions, $\gamma_i = 1 - p_i$ is the probability of receiving a linearly dependent coded packet and $a_i = \prod_{m=1, m \neq i}^{g} (1 - q^{m-i})^{-1}$ is a scaling factor for γ_i that quantifies the effect of the linear dependence in the decoding probability.

$$f_{\mathbb{T}_{RLNC}}(t; q, g) = \Pr[\mathbb{T}_{RLNC} = t] = \qquad (1)$$
$$P_g \sum_{i=1}^{g} a_i {\gamma_i}^{t-g}, \;\; t \in [g, \infty)$$

B. Broadcast RLNC

As an approximation, we consider the case of finding the required transmissions for the maximum of N independent unicast sessions which makes the results an upper bound since we are excluding the transmissions accounting common coded packets. For each unicast session, we first model a single source - destination pair with erasure ϵ with RLNC and then proceed to calculate the broadcast case. Here, we need to account for g l.i. received packets in t transmissions. Hence, we need to consider all the cases where i l.i. packets are received (with the final success in t, which [8], [9] do not consider) and $t-i$ packets were lost or linearly dependent. For this, we review two main probabilities in the same way as [10]. First, let $\Pr[\mathbb{T}_{S_i} = t]$ be the probability for receiving i coded packets in t transmission (only considering the erasures), then $\mathbb{T}_{S_i} \sim \mathrm{NB}(i, 1-\epsilon)$. Second, the probability that g coded packets are l.i. in i slots, is $\Pr[\mathbb{T}_{RLNC} = i]$. Subsequently, the probability of decoding in exactly t slots for a single user with RLNC based unicast with erasure ϵ, $\mathbb{T}_{U,cod}$, is:

$$f_{\mathbb{T}_{U,cod}}(t; \epsilon, q, g) = \Pr[\mathbb{T}_{U,cod} = t] = \qquad (2)$$
$$\sum_{i=g}^{t} \binom{t-1}{i-1} (1-\epsilon)^i \epsilon^{t-i} f_{\mathbb{T}_{RLNC}}(i; q, g), \;\; t \in [g, \infty)$$

Since each receiver just needs to collect different linear combinations to decode the packets, the number of transmissions will be bounded by the receiver that performs the worst in terms of retransmissions, i.e. $\mathbb{T}_{B,cod} = \max_{j=[1,N]} \mathbb{T}_{U_j,cod}$, which we calculate by a c.d.f. approach. For the probability of the maximum being less than or equal to t transmissions, this must occur for every receiver. Then, under the independence assumption, we can compute the c.d.f for broadcast RLNC, e.g. $\Pr[\mathbb{T}_{B,cod} \leq t] = \prod_{j=1}^{N} \Pr[\mathbb{T}_{U_j,cod} \leq t]$ from (2) with the resulting pmf in (3).

$$f_{\mathbb{T}_{B,cod}}(t; N, \epsilon_1, \dots, \epsilon_N, q, g) = \qquad (3)$$
$$\prod_{j=1}^{N} \left(\sum_{k=g}^{t} \sum_{i=g}^{k} \binom{k-1}{i-1} (1-\epsilon_j)^i \epsilon_j^{k-i} f_{\mathbb{T}_{RLNC}}(i; q, g) \right)$$
$$- \prod_{j=1}^{N} \left(\sum_{k=g}^{t-1} \sum_{i=g}^{k} \binom{k-1}{i-1} (1-\epsilon_j)^i \epsilon_j^{k-i} f_{\mathbb{T}_{RLNC}}(i; q, g) \right)$$
$$, \;\; t \in [g, \infty)$$

We notice that the expression in (3) is the general case for low field sizes of the randomized broadcast coding scheme reviewed in [3], since if we let $q \to \infty$ in (3), then the c.d.f. used to compute (3) tends to the c.d.f. used to compute the mean and variance in section III-B of [3].

C. Cloud Cooperation with Coding

For the cooperation scheme, we consider a mobile cloud composed of H receivers ($H < N$) with cellular connection, the *heads* and $N - H$ receivers the *non-heads*, with a local connection to the heads. Packet transmissions takes place in two stages: (i) between source and heads and (ii) between heads and non-heads, which we label the cellular and local stage respectively. For the cellular stage, the source broadcast a coded packet to the heads which receive it *collectively*, i.e. it is enough that one head gets it for the cloud to acknowledge reception, with the stage finishing once the heads get the generation as a group. In the local stage, the heads broadcast recoded packets in a round robin fashion to the non-heads. The local stage finishes once all receivers have decoded the generation.

Under this condition, the distribution of receiving g packets in the cellular stage for the heads, $\mathbb{T}_{C,cod}$ is modeled as $\mathbb{T}_{U,cod}$ but with a success probability given by $1 - \prod_{j=1}^{H} \epsilon_j$ because all links need to fail for a packet to not be received. In the local stage the heads takes turn to broadcast to the non-heads, which is a particular case of (3). The total number of transmissions for this scheme, $\mathbb{T}_{CC,cod}$, is given by $\mathbb{T}_{CC,cod} = \mathbb{T}_{C,cod}(H, \prod_{j=1}^{H} \epsilon_j, g, q) + \mathbb{T}_{B,cod}(N-H, \epsilon, g, q)$ where the parentheses notation indicates the evaluation of the pmf with the given parameters.

III. Performance Metrics

With the pmf for each scheme from section II, we calculate the moments for the number of transmissions which allows us to compute the throughput and energy.

A. Throughput

We define the throughput in the cloud cooperation scheme for a given erasure rate, generation and field size in the following way:

$$R_{eff,CC} = \frac{g}{\max(T_{s,cel}\mathbb{T}_{C,cod}(H), T_{s,loc}\mathbb{T}_{B,cod}(N-H))} \tag{4}$$

In (4), $T_{s,cel}$ and $T_{s,loc}$ are the duration of a time slot in the cellular and local stages, respectively. The effective rate perceived by a user will be the information sent divided by the completion time. For broadcast RLNC, the throughput is $R_{eff,B} = \frac{g}{T_{s,cel}\mathbb{T}_{B,cod}(N)}$.

B. Energy Consumption

We review the energy spent for the BS and average energy per receiver for the cooperation and broadcast schemes on the coded cases for a given erasure and code parameters. First, the energy consumption for broadcast is as follows:

$$E_{T_x} = E_{cel}\mathbb{T}_{B,cod}(N) \; ; E_{R_x} = E_{cel}\mathbb{T}_{B,cod}(N) \tag{5}$$

Where $E_{cel} = N_B E_B$ is the energy cost per packet in the cellular stage, N_B is the number of bytes per packet and E_B is the energy per byte proportional to the energy per bit. In a similar way, the energy expenditure for the cooperation schemes is shown in (6).

$$\begin{aligned} E_{T_x} &= E_{cel}\mathbb{T}_{C,cod}(H) \\ E_{R_x} &= E_{cel}\left(\frac{H}{N}\right)\mathbb{T}_{C,cod}(H) + E_{loc}\mathbb{T}_{B,cod}(N-H) \end{aligned} \tag{6}$$

C. Cellular vs. Local Links

The performance of cooperation will depend on the throughput and energy use per bit on the local links vs. that on the cellular links. Therefore, we define the r_t as the ratio between cellular and local throughput, and r_e as the ratio between the cellular and local energy.

$$r_t = \frac{T_{s,loc}}{T_{s,cel}} = \frac{R_{s,cel}}{R_{s,loc}} \quad ; \quad r_e = \frac{E_{b,cel}}{E_{b,loc}} \tag{7}$$

D. Gain Regions

For the analysis with different erasure rates per stage, we define the throughput and energy gains of cloud cooperation against broadcast RLNC from (5) and (6) as shown in (8).

$$\begin{aligned} G_t &= \frac{E\{\mathbb{T}_{B,cod}(N,\epsilon_{cel})\}}{\max(r_t E\{\mathbb{T}_{C,cod}(H,\epsilon_{cel})\}, E\{\mathbb{T}_{B,cod}(N,\epsilon_{cel})\})} \\ G_e &= 1 - \frac{r_e\left(\frac{H}{N}\right)E\{\mathbb{T}_{C,cod}(H,\epsilon_{cel}) + \mathbb{T}_{B,cod}(N-H,\epsilon_{loc})\}}{r_e E\{\mathbb{T}_{B,cod}(N,\epsilon_{cel})\}} \end{aligned} \tag{8}$$

We define throughput gain as the ratio of the cloud cooperation and broadcast RLNC throughputs. The energy gain of cooperation over broadcast is defined as the saving in energy for the devices, since cooperation always save energy at the BS.

IV. Numerical Results

With the obtained expressions, we can evaluate broadcast and cooperation to study the impact on the throughput and energy at the receivers, as we vary the number of users, the ratio between the cellular and local costs, and the erasure rates on the cellular and local links. We use a set of parameters in the following ranges $1 \leq N \leq 50$, $g = \{64, 128\}$, $q = 2^8$ and $0 \leq \epsilon \leq 0.6$. The timeslot duration is set to $T_{s,cel} = 0.5$ ms to conform to the LTE-A E-UTRA [11] and its set of D2D specifications. For the energy, we extracted the energy per bit cost from the energy model in [12] and use a packet size N_B of 500 B.

Fig. 2 shows the throughput as defined in (4) for the different cooperation schemes and broadcast RLNC when the cellular and local data rate are identical. Generally as the number of users increase the sustainable throughput to each receiver decreases. The highest throughput is obtained when the majority are heads, as this reduces the work in the local phase. As the number of non-heads increases the throughput with cooperation tends to that of broadcast, because the transmissions on the local stage becomes the dominating cost.

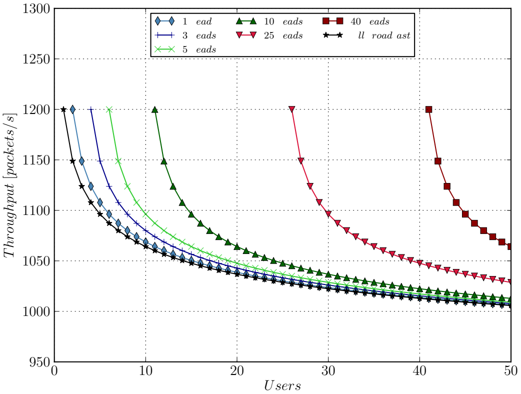


Fig. 2: Schemes throughput for equal date rate costs in the cellular and local link. Used parameters: $g = 64,\ q = 2^8,\ \epsilon = 0.4,\ r_t = 1$

Fig. 3 shows the energy spent per device where the energy costs are the same on the cellular and local links for both schemes. For a low amount of users, the energy consumption for the cooperation scheme is higher than broadcast because the amount of transmissions in the cellular and local links are comparable. As the number of user increases, the number of transmissions in the cellular link tends to g while the transmissions in the local link increases reducing the difference in performance.

Fig. 4 shows how the throughput varies depending on the ratio of the cellular and local data rate. The ratios are

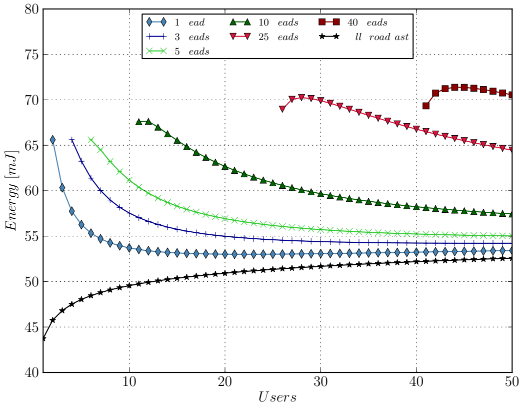


Fig. 3: Schemes energy consumption for equal energy costs in the cellular and local link. Used parameters: $g = 64$, $q = 2^8$, $\epsilon = 0.4$, $r_e = 1$

obtained by fixing the cellular data rate and varying the local data rate. When the local data rate is lower than the cellular rate the cooperative schemes provides lower throughput than the broadcast scheme. Conversely, when the local data rate is higher than the cellular data rate, the cooperative schemes delivers a higher throughput than broadcast. The throughput is highest when the local links rate are twice as faster as the cellular ones. The number of heads controls how much gain can be obtained and where it occurs for a given ratio. When the number of heads decreases, the throughput also diminishes because there are fewer heads each with an independent chance of receiving the packet.

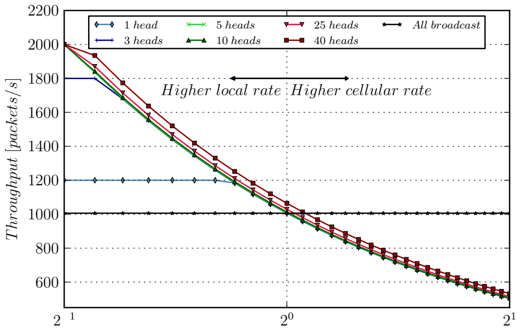


Fig. 4: The throughput of broadcast and cooperation with different number of heads, for different ratios between the data rate on the cellular and local link. Used parameters: $g = 64$, $q = 2^8$, $\epsilon = 0.4, N = 50$.

Fig. 5 shows how the energy for the devices changes as ratio between the cellular and local energy per bit changes. The energy cost in the cellular link is fixed and cost on the local link is changed to obtain the different ratios, consequently the energy per bit for broadcast is constant. When the energy cost for the local links is higher than the cellular energy cost, the cooperative scheme expends more energy than the broadcast scheme. The additional consumption for cooperation comes from the transmissions in the local stage. Contrarily, when the cost of the local links is lower than the cost of the cellular links, then the cooperation scheme uses less energy than broadcast. For the cooperation schemes, the consumption is determined by the number of heads on the cellular stage. For a low number of heads, energy consumption is the lowest because the transmissions on the cellular links are for a few devices only.

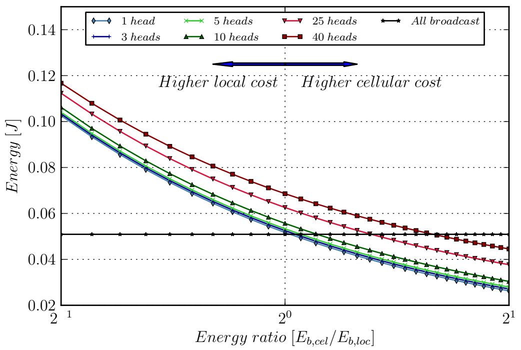


Fig. 5: The energy per generation of broadcast and cooperation with different number of heads, for different ratios between the energy per bit on the cellular and local link. Used parameters: $g = 64$, $q = 2^8$, $\epsilon = 0.4, N = 50$.

Fig. 6 shows the regions where cooperation provides a gain in terms of throughput for a wide range of erasure rates on the cellular and local links. The lines show where broadcast and cooperation performs the same, for $r_t = [0.5, 0.8, 1, 1.5, 2]$. In the region below each line, cooperation provides higher throughput than broadcast for that particular r_t. Above the line broadcast performs better. E.g. in the case of a fast local link $r_t = 0.5$ then cooperation provides a gain for almost all considered erasure rates, even in cases where the local erasure rate is much higher than the cellular.

Fig. 7 shows the regions where cooperation provides a gain in terms of energy saving on the devices for various erasure rates on the cellular and local links. The lines show where broadcast and cooperation performs the same, for $r_e = [0.5, 0.75, 1, 1.5, 2]$. In the region below each line, cooperation provides a lower energy per bit than broadcast for that particular r_e. Above the line broadcast performs better.

V. Conclusions

This work revisits the problem of wireless cooperation with network coding on cellular systems for multicast sessions in light of the increased data rates of current 4G and future 5G mobile networks and the stagnant data rates in short-range technologies, e.g., WiFi. This is particularly relevant because it breaks with the common assumption that the cooperative cluster can communicate locally at much higher data rates than the direct link to the cellular base station.

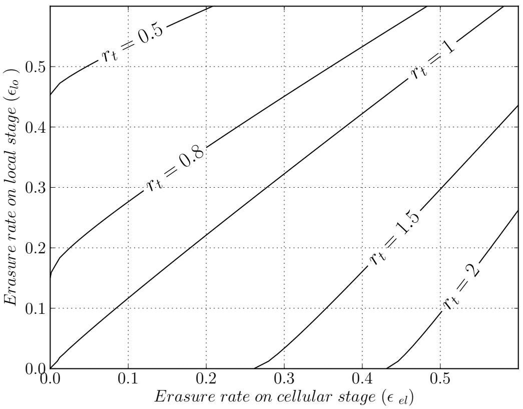


Fig. 6: For different values of r_t the lines indicate where cooperation and broadcast provide the same thoughput for various erasure rates on the cellular and local links. Below the each line, cooperation performs better for the respective r_t. Used parameters: $g = 128,\ q = 2^8,\ H = 40,\ N = 50$.

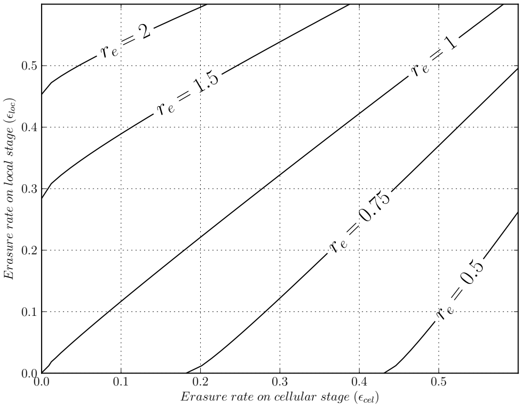


Fig. 7: For different values of r_e the lines indicate where cooperation and broadcast provide the same energy per bit for various erasure rates on the cellular and local links. Below the each line, cooperation performs better for the respective r_e. Used parameters: $g = 128,\ q = 2^8,\ H = 40,\ N = 50$.

More specifically, we presented an in-depth study of the specific operating regions where cooperation provides gains in throughput and energy over coded broadcasting techniques. Our numerical results showed that gains can be achieved even if the long-range and short-range technologies transmit at comparable data rates. More importantly, we showed that cooperation can provide several fold gains to the best broadcasting option (network coded broadcast) as long as the short-range link is at least twice as fast as the long-range one. Finally, our results showed that a moderate number of heads (e.g., three or more) per cooperative cluster is enough to yield the high throughput gains while maintaining a low energy consumption at the receivers. The latter is not possible if a large fraction of the cooperative cluster is actively receiving directly from the base station with only a few exchanges needed during the cooperation process.

Future work shall focus on protocol design for cooperative schemes in highly-dense scenarios as well as implementation and evaluation of the most promising schemes in Aalborg University's Raspberry Pi testbed [13].

ACKNOWLEDGMENTS

This research has been partially financed by the CROSS-FIRE MITN Marie Curie project (317126) from the European Comission FP7 framework and the Green Mobile Cloud project (Grant No. DFF - 0602- 01372B) granted by the Danish Council for Independent Research.

REFERENCES

[1] F. H. Fitzek and M. D. Katz, *Cooperation in Wireless Networks: Principles and Applications - Real Egoisic Behaviour is to Cooperate!* Springer, 2006.

[2] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, 2000.

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

[4] J. Heide, F. H. Fitzek, M. V. Pedersen, and M. Katz, "Green mobile clouds: Network coding and user cooperation for improved energy efficiency," in *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on.* IEEE, 2012, pp. 111–118.

[5] H. Khamfroush, D. E. Lucani, and J. Barros, "Minimizing the completion time of a wireless cooperative network using network coding," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on.* IEEE, 2013, pp. 2016–2020.

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

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

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

[9] X. Zhao, "Notes on "exact decoding probability under random linear network coding"," *Communications Letters, IEEE*, vol. 16, no. 5, pp. 720–721, 2012.

[10] D. E. Lucani, M. Médard, and M. Stojanovic, "Random linear network coding for time division duplexing: Field size considerations," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE.* IEEE, 2009, pp. 1–6.

[11] 36.913, "Requirements for further advancements for e-utra (lte-advanced)," 3GPP, Tech. Rep. Release 8, March 2009.

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

[13] A. Paramanathan, P. Pahlevani, S. Thorsteinsson, M. Hundebøll, D. Lucani, and F. Fitzek, "Sharing the pi: Testbed description and performance evaluation of network coding on the raspberry pi," in *2014 IEEE 79th Vehicular Technology Conference*, 2014.

Paper B

Throughput, Energy and Overhead of Multicast Device-to-Device Communications with Network Coded Cooperation

Néstor J. Hernández Marcano, Janus Heide, Daniel E. Lucani, Frank Fitzek

The paper has been published in the
2016 Wiley Transactions on Emerging Telecommunications Technologies (former European Transactions on Telecommunications). Special Issue: Emerging Topics in Device to Device Communications as Enabling Technology for 5G Systems, pp. 1–17, 2016.

TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES
Trans. Emerging Tel. Tech. (2016)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/ett.3011

SPECIAL ISSUE PAPER

Throughput, energy and overhead of multicast device-to-device communications with network-coded cooperation

Néstor J. Hernández Marcano[1,2]*, Janus Heide[1], Daniel E. Lucani[2] and Frank H. P. Fitzek[2]

[1] Steinwurf ApS, Aalborg, Denmark
[2] Department of Electronic Systems, Aalborg University, Aalborg, Denmark

ABSTRACT

Cooperation strategies in mobile networks typically rely in short-range technologies, like LTE-A device-to-device communications, for data exchange between devices forming mobile clouds. These communications provide a better device experience because the clouds offload the network. Nevertheless, this assumes that the throughput gains and energy savings in multicasting are much larger between devices than the base station to the receivers. However, current mobile networks suffer from many different issues varying the performance in data rates, which calls into question these assumptions. Therefore, a first objective of this work is to assess the operating regions where employing cooperation results in higher throughput and/or energy savings. We consider multicast scenarios with network-coded mechanisms employing random linear network coding (RLNC). However, although RLNC is good for low amount of transmissions in multicast, it has an inherent overhead from extreme high or low field-related caveats. Thus, as a second objective, we review and propose the application of new network codes that posses low overhead for multicasting, by having a short representation and low dependence probability. We provide an analytical framework with numerical results showing (i) gains of several fold can be attained even if the in-device data rates are moderately larger (2×) than the cellular link data rate and (ii) that is feasible to attain less than 3% total mean overhead with the proposed codes. This is fairly lower than what can be achieved with RLNC schemes in most of the considered cases and achieving at least 1.5–2× gains. Copyright © 2016 John Wiley & Sons, Ltd.

***Correspondence**
N. J. Hernández Marcano, Steinwurf ApS, Aalborg, Denmark.
E-mail: nestor@steinwurf.com, nh@es.aau.dk

Received 5 July 2015; Revised 14 October 2015; Accepted 25 November 2015

1. INTRODUCTION

Data traffic is expected to grow by an order of magnitude for wireless mobile devices to support many data demanding services as shown in Figure 1 [1]. Common services of this type are video multicasting or local multimedia content sharing. From the perspective of the mobile users, high perceived quality and a low battery drain are important. For operators, their goal is to serve the highest number of users with the least amount of network resources and energy consumption from their infrastructure. In this scenario, both multicast and cooperation provide better performance than unicast, because several devices are served with the same communication resources.

Then, mechanisms that can offload network infrastructures have gathered significant interest from both academia and industry. Thus, there is a general interest from both academia [2, 3] and industry [4] in finding strategies that reduce mobile network usage by offloading the infrastructure to other types of short-range communications like device-to-device (D2D) or Wi-Fi. For this purpose, wireless cooperative *mobile clouds* [5, 6] are formed by receivers helping the cellular network by locally exchanging missing data packets instead of directly requesting them from it. Thus, cooperative techniques result in increased reliability, coverage extension and even increased throughput to end receivers. This potential has resulted in the inclusion of D2D communications in the 3rd Generation Partnership Project standardisation efforts. To recover from packet erasures in the wireless medium, typically rateless codes are employed as a forward error correction technique. Nevertheless, although they provide benefits for a broadcast scenario, they cannot be deployed for cooperative communications without affecting their

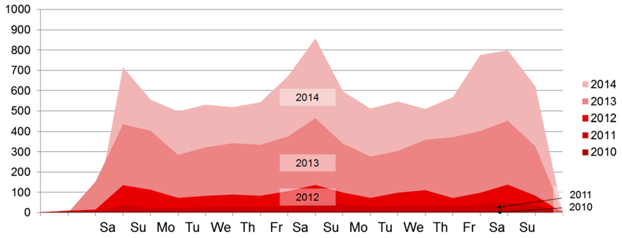


Figure 1. Average Oktoberfest data load for the 2010–2014 period (generic data units).

performance or decoding the data for each hop. Thus, there is a need for code schemes to overcome these drawbacks.

In this context, network coding [7], and particularly random linear network coding (RLNC) [8], not only provides a faster and more efficient approach to broadcast the data to the users, as shown by [9], but also simplifies the cooperation process because (i) the devices only need to know the number of linear combinations available and (ii) transmissions from a single device during the cooperation process can have a larger impact for the end receivers, given that a single transmissions can help different devices at the same time. This intuition has been exploited in previous works ranging from analysis to optimal policies and practical mechanisms, for example, Khamfroush *et al*. [10, 11].

However, the underlying assumption is that the cooperative channel is considerably faster than the channel to the base station and that energy costs are much lower because of proximity. Although this assumption was reasonable long ago, now the much higher data rates achievable in Long Term Evolution-Advanced (LTE-A) call this assumption into question. Also, if the devices cooperate using D2D of LTE-A, the data rate for cooperation will also be limited by the common channel and could be the same data rate in some cases. Thus, one goal of this paper is to review the problem of device cooperation focusing on the specific regions of operation where it can bring gains in throughput and energy.

Some of the analysis of mean performance for cooperative schemes has been carried out before, for example, Heide *et al.* [6, 12]; however, this paper provides a comprehensive study of the distributions of the number of transmitted packets for various scenarios. We define the natural concept of stable throughput for cooperative schemes. To the best of our knowledge, this has not been considered before because of the conventional assumption that the cellular data rate is the bottleneck in the communication process.

A second important factor that may affect the performance of the network is the coding scheme choice. The reason being that if a code does not employ resources properly, it reduces even more the benefits of throughput and energy in the short-range links. Here, RLNC considers creating linear combinations from a single network flow to distribute data between many devices. In order to reduce complexity and delay, transmissions of packet batches from the original data, called generations, were introduced in [13] as a technique to accomplish this. RLNC's flexibility to adapt to different network topologies makes it an interesting choice for upcoming networking protocols. Although, given that the random coefficients are picked from a single field, RLNC-based techniques have the caveat of introducing overhead due to mainly two reasons in the coding process. First, transmissions of linearly dependent (l.d.) packets occur due to the random selection of the coefficients [14, 15]. Second, in order to later perform decoding, typically the easiest goes for the coding coefficients to be appended to each coded packet before being sent through the network.

Different works have been made to observe the effects of RLNC parameters, for example, generation and field size [16], not only for the overhead but also for other metrics such as energy consumption and processing speed [17] among others. Variants of RLNC have been proposed to exploit a particular code structure to obtain a low overhead without compromising other metrics [18]. More recently, in [19], *telescopic codes* (TC) were introduced applying composite extension fields within a single generation, with the goal of reducing the total overhead while preserving recoding. Its parameters and performance were analysed for an ideal unicast scenario, but their potential for unreliable multicast scenarios was not explored.

In [20] and [21], the authors develop new optimised transmission and coding schemes based on RLNC for content distribution. However, they only consider broadcast and exclude cooperation. Also, these studies have a strong focus on video streaming, while in our case, we deal with distributing data in general. Furthermore, they do not consider the overhead of the code. In [22–25] can be found the advantages of employing multicast schemes with D2D capabilities. Although, these works do not consider both cooperation and network coding at the same time, making it difficult to evaluate the benefits of network coding in these scenarios. In this work, we pursue two goals: we (i) analyse the throughput and energy gains of network-coded cooperation and (ii) propose the employment of TC as a technique for minimising the overhead in heterogeneous, unreliable, cooperative networks. We provide a full analysis with a set of numerical results for both the region gains with RLNC and the use of TC for low overhead. We make this analysis for three scenarios in multicast sessions, namely, broadcast, single cloud cooperation and multiple clouds cooperation for D2D communications, because they provide more benefits than others like unicast. Moreover, for the employment of TC, we compute the set of composite fields required to minimise the total overhead, which we define for broadcast and single cloud cooperation.

Our work is organised as follows: Section 2 introduces our system and general assumptions. Section 3 describes the code schemes and scenarios that we review indicating how the information is conveyed through the network. Analysis of the proposed coding schemes and transmission scenarios is made in Section 4. Later, Section 5 shows the considered metrics in our study with the numerical results provided in Section 6. Final conclusions are presented in Section 7. The proofs of the used lemmas and corollaries in our study are in the Appendix.

2. SYSTEM MODEL

We consider the problem of reliably transmitting a set of packets from a source to N receivers in a cellular network under various transmission scenarios. The set constitutes a generation of g packets, which we code using RLNC with field size q. We consider a general topology as shown in Figure 2 where receivers might form D2D-based mobile clouds with multicast capabilities. Devices that are relatively close together create fully interconnected clouds.

In general, we may have C clouds, where each cloud has N_n users with $n \in [1, C]$ and $N = \sum_{n=1}^{C} N_n$. Here, in each cloud, we differentiate two types of devices. First, we refer as *heads* the devices with both cellular connectivity to the source and all others in the cloud. Then, we have $H_n \leqslant N_n$ heads per cloud. Second, the *non-heads* are the devices without a cellular connection but only to all others, for example, $N_n - H_n$ non-heads per cloud. For our study, we briefly observe the dominating regimes for heterogeneous cloud sizes to notice that the homogeneous cloud size provides the best performance in terms of total transmission time. Therefore, we review the case of having a homogeneous cloud size, for example, the same number of users in each cloud $N_n = N_{uc}\ \forall\ n$, giving $N = C \times N_{uc}$, because this is the ideal operational regime.

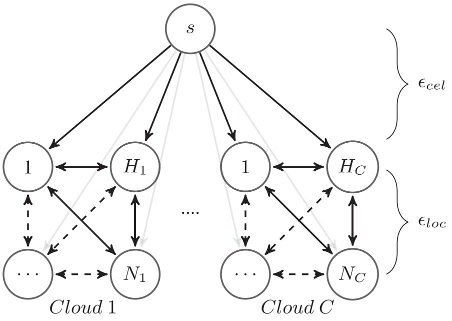


Figure 2. Topology of C device-to-device mobile clouds.

We consider independent packet erasure rates on the connectivity links from the source to the heads, ϵ_{jn}, $j \in [1, H_n]$, $n \in [1, C]$; for example, the packet reception distribution of receiver j is *Bernoulli*$(1 - \epsilon_{jn})$ and is independent from all others. We will provide the general expressions for scenario distributions with heterogeneous cellular links, but for evaluation purposes, we will consider all these erasures equal to ϵ_{cel}. Similarly, inside any cloud, all the connections can be regarded as bidirectional symmetric channels via D2D. These links have homogeneous, independent and identically distributed erasure rates, which we consider equal to ϵ_{loc} for simplicity.

To guarantee interference management and proper content delivery, we assume an underlay D2D mobile network. In this way, we review the transmission scenarios once a previous and arbitrary cellular network resource allocator has assigned the communications resources in it. Once network management has been addressed, packets are either sent under a broadcast ($N_{uc} = 1$), multiple cloud ($1 < N_{uc} < N$) or single cloud ($N_{uc} = N$) cooperation transmission scenarios.

Thus, in any scenario, for convenience, we consider that $C + 1$ physical communication resources have already been allocated. Nevertheless, if $K < C + 1$ are only available, standard cellular techniques such as frequency reuse patterns can be employed. Then, two clouds using the same frequency will be geographically separated long enough to ensure the reuse of communication resources without causing interference. We consider this to perform the transmissions between the source and all the clouds through *multicast groups*. First, a multicast group is created between the source and all cloud heads to transmit the content to each cloud collectively. Second, C multicast groups are created between the heads of each of the C clouds and their respective non-heads, to cover missing packets. In this stage, the network controller is in charge of the coordination regarding on how the nodes must share their content into its own cloud.

Inter-cloud interference is defined as the reception of a transmission from a cloud where a receiver does not belong, which occurs when a transmission of that receivers cloud has been made during the same time slot. The interference generates a reduction in the signal-to-interference-plus-noise ratio, which increases the erasure rate and degrades the reception, possibly forcing retransmissions. Hence, we assume that cloud transmissions take place in orthogonal resources to ensure that inter-cloud interference does not occur. In case there are no available resources, techniques such as frequency reuse planning from mobile networks help to avoid this interference. For example, two clouds using the same frequency will be geographically separated enough to ensure there is no interference, making possible to reuse communications resources.

We consider the scenario where an acknowledgment (ACK) is sent only in two cases: first, when a receiver has collected g linearly independent (l.i.) coded packets and second, for the cooperative case, when a cloud has g l.i. coded packets as a group. The ACK is sent through a reliable communications channel. Thus, only one ACK per generation of packets per receiver is sent. This makes a total of N feedback packets, differing from the case where ACKs are sent on a per-packet and per-receiver basis, which would result in $g \times N$ feedback packets. In other words, this reduces the feedback transmissions by a factor of g. If feedback is not wanted (or possible), a fixed number of extra transmissions can be sent ensuring some target reliability level, for example, 99%. However, because the number of transmissions is fixed, such an approach will lead to an additional transmission overhead particularly if the channel is time varying.

3. CODING SCHEMES AND TRANSMISSION SCENARIOS

In this section, we first consider and describe two coding schemes. Afterwards, we describe the three transmission scenarios considered in our study, namely, broadcast, single cloud cooperation and multiple clouds cooperation. We review them going from the simplest to the most elaborated scenario. Regarding the coding schemes, first, in all the scenarios, the source or the heads employ RLNC as a coding scheme. Second, for only broadcast and single cloud cooperation, we employ TC [19]. These types of codes possess very low overhead. The reason is that they are tailored for a given network to provide the best trade-off between both the overhead caused by (i) transmissions of l.d. packets and (ii) sending the coding coefficients for any receiver to be able to decode.

To accomplish this, TC relies on *composite extension finite fields*. This type of fields enables an encoder to create coded packets in an RLNC fashion but with some differences. The key idea for TC with composite fields is the following: Instead of picking all coding coefficients from a single field q inside a generation of size g, the encoder selects each coding coefficient $\boldsymbol{v}_j$, $j \in [1, g]$ in the generation from $GF(\boldsymbol{q}_j)$. Here, $\boldsymbol{q}$ is a vector that contains all the field sizes employed for each of the coding coefficients. Each of the field sizes follows a specific pattern. The chosen pattern makes compatible finite fields arithmetics from different fields in the same generation. In the subsequent sections, we give a brief description of RLNC because it is well known in the network coding literature. However, we provide further details for TC because they were recently introduced. Also, as we will see in our framework, RLNC will be regarded as a special case of TC that occurs when $\boldsymbol{q}_i = q \ \forall i \in [1, g]$. Thus, we will focus the following section on describing TC and its operations.

3.1. Coding schemes

3.1.1. Random linear network coding.

For this conventional coding scheme, we may create an encoded packet, create a recoded packet from a previous set or decode any set g coded packets performing Gaussian elimination on the $g \times g$ coding matrix. In general, we consider a field size q from which we take the coding coefficients and perform predefined Galois Field (GF) arithmetics. As follows, we will see that this is a special case of TC.

3.1.2. Telescopic codes.

We consider coding g packets, $\boldsymbol{m}_j, j \in [1, g]$ in the generation, each of size B bits. We define $\boldsymbol{q}$ as a vector containing each of the field sizes $\boldsymbol{q}_j$, we assume $\boldsymbol{q}_j \leqslant \boldsymbol{q}_{j+1}, \forall j \in [1, g]$ without loss of generality. Then, coded packets are generated in a similar fashion as with RLNC. However, each coding coefficient in an encoding vector $\boldsymbol{v}$ is chosen uniformly at random from a finite field $GF(\boldsymbol{q}_j), j \in [1, g]$ differing from RLNC. To keep valid field arithmetics, TC are based on composite extension finite fields of the form $\mathbb{F}_{2^k}$ with $k \in [1, 2, 4, 8, \ldots]$ in general. The key idea in composite extension fields is to design the arithmetic operations of a new extended finite field from the operations of a base field. With the extended field, we can continue the process and create another extension and so on. These fields are designed to allow compatibility for the operations performed among them. The composite fields are defined and described in [19]. Nevertheless, we give a brief overview of its operations to have a description for our analysis in Section 4.

3.1.3. Encoding.

A generic coded packet, $\boldsymbol{c}_i$, is generated by an encoder by mixing linear combinations of g packets. Here, packet $\boldsymbol{m}_j$ is regarded as an element from $GF(\boldsymbol{q}_j)$ and is multiplied by a coding coefficient $\boldsymbol{v}_{i,j}$ chosen uniformly at random from $GF(\boldsymbol{q}_j)$. Thus, a resulting coded packet of size B and its coding can be expressed as:

$$\boldsymbol{c}_i = \bigoplus_{j=1}^{g} \boldsymbol{v}_{i,j} \otimes \boldsymbol{m}_j \tag{1}$$

$$|\boldsymbol{v}_i| = \sum_{j=1}^{g} |\boldsymbol{v}_{i,j}| = \sum_{j=1}^{g} \lceil \log_2(\boldsymbol{q}_j) \rceil \text{ [bits]} \tag{2}$$

3.1.4. Decoding.

To perform decoding, we define $\boldsymbol{c} = \left[\boldsymbol{c}_1 \ldots \boldsymbol{c}_g\right]^T$ and $\boldsymbol{m} = \left[\boldsymbol{m}_1 \ldots \boldsymbol{m}_g\right]^T$. Decoding reduces to solve the linear system $\boldsymbol{c} = \boldsymbol{V} \cdot \boldsymbol{m}$ using Gaussian elimination [26]. Here, the coding matrix $\boldsymbol{V}$ contains any set of g l.i. packets $\boldsymbol{c}_i$ as rows as follows:

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 \\ \hline \vdots \\ \hline \boldsymbol{v}_g \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_{1,1} & \ldots & \boldsymbol{v}_{1,g} \\ \vdots & \ddots & \vdots \\ \boldsymbol{v}_{g,1} & \ldots & \boldsymbol{v}_{g,g} \end{bmatrix} \tag{3}$$

The decoder begins removing the contributions of the smallest field pivot elements , for example, leftmost elements in the main diagonal of (3). Once that information is known to the decoder, it proceeds in the same way for the next upper field in the composition using operations of the current field and so on to obtain the original set of packets [19].

3.1.5. Recoding.

Given the picking of elements from different fields, recoding needs to be properly defined to ensure that the coefficients of a recoded packet cannot be differentiated from a single coded one. Different approaches are described in [19]. For the scope of this study, we considered recoding in the lowest available field in the generation because it preserves recoding with low overhead in the encoding vector. Nevertheless, the major problem in this approach is the increase of more l.d. packets for the last transmissions in a given generation. In this way, let us define a generic recoded packet as $\tilde{\boldsymbol{c}}_i$ and its corresponding encoding vector as $\tilde{\boldsymbol{v}}_i$ as follows:

$$\tilde{\boldsymbol{c}}_i = \bigoplus_{j=1}^{g} \boldsymbol{w}_{i,j} \otimes \boldsymbol{c}_j \tag{4}$$

$$\tilde{\boldsymbol{v}}_i = \bigoplus_{j=1}^{g} \boldsymbol{w}_{i,j} \otimes \boldsymbol{v}_j \tag{5}$$

In (4) and (5), $\boldsymbol{w}_{i,j}$ is the coding coefficient that multiplies $\boldsymbol{c}_j$, uniformly and randomly chosen from $\min(\boldsymbol{q})$. Any decoder that collects $\tilde{\boldsymbol{c}}_i, i \in [1, g]$ l.i. packets, with their $\tilde{\boldsymbol{v}}_i$, will be able to decode the whole generation as described in Section 3.1.4.

3.2. Transmission scenarios

3.2.1. Broadcast.

For broadcast, $N_{uc} = 1$, the source generates an encoded packet and later attaches the coding coefficients values and sends it to all receivers through the erasure channels described in Section 2. When a packet successfully arrives at a receiver, it checks if the packet is l.i. from all its previous. If not, it discards it. In case of being l.i., the receivers add it to its coding matrix. This process is repeated until all receivers have collected their required combinations. An ACK is sent through the feedback channel from the last receiver after it obtains its final combination. In this scheme, recoding is not used.

3.2.2. Single cloud cooperation.

In a single cloud cooperation scenario, $N_{uc} = N$ and packet transmissions take place in two stages. First, the source broadcasts coded packets to the heads through the cellular network, that is, the cellular stage, and second, internally between receivers, which have missing packets, in a network-coordinated fashion, that is, the local stage. For the cellular stage, the source broadcast coded packets for the cloud heads, because it is enough to obtain coded packets collectively to later recode them. Once the cellular stage has ended, for example, the g l.i. packets are in the cloud, there will be receivers that do not exactly have this quantity of packets to decode. To manage this, in the local stage, each head broadcasts recoded packets in a coordinated way to ensure all receivers obtain their remaining packets. For the case of TC, recoding will be performed in the lowest field. This stage finishes once all receivers have decoded the generation and any device in the cloud sends and an ACK through the feedback channel to the sender.

3.2.3. Multiple clouds cooperation.

For a cloud cooperation scenario, $1 < N_{uc} < N$, packet transmissions go in a similar way as before. However, two main differences exist. First, in the cellular stage, packets are broadcasted to the heads multicast group, instead of all in a single cloud. Hence, we use one multicast channel. Second, in the local stage, the C set of heads broadcasts recoded packets to all the other users inside their respective multicast groups. Again, this performed in a network-coordinated way to ensure all receivers obtain their remaining packets. This stage finishes once all receivers in all clouds have g l.i. coded packets and an ACK through the feedback channel to the sender.

4. SCENARIOS ANALYSIS

We proceed to study the underlying probability distributions for the number of transmissions required to decode either RLNC or TC within their respective transmission scenarios. With the statistical description of the transmissions, we perform two types of studies. First, we identify the regions where cooperation performs better than broadcast in terms of throughput and energy when RLNC is used. Second, for TC, we perform an overhead analysis for the transmission scenarios and evaluate them for three schemes that can be used with TC.

For the code and each transmission scenario, we model the number of transmissions to decode as a random variable to derive its probability mass function (pmf). We perform this in order to obtain reasonable approximations of the linear independence, erasure and transmission processes and also to separate the effect of the code from the scenario. We first give an expression of the pmf for RLNC, later we incorporate erasures in the process and finally compute the pmf for the transmission scenarios.

4.1. Coding scheme distributions

To calculate the distribution and pmf for either an RLNC or TC scheme in a generic fashion, we derive a framework for the pmf of TC distribution, and we will make the proper evaluations to differentiate with RLNC.

We first consider a single source-destination link without erasures. The process for obtaining each new l.i. packet can be modelled by the Markov Chain in Figure 3. This chain comprises $g + 1$ states. First, state i with $i \in [1, g]$ is the case where the i-th l.i. coded packet has not been received

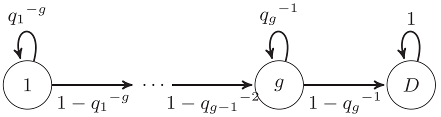


Figure 3. Absorbing Markov Chain for telescopic codes.

by the destination. Then, D is the absorbing state where decoding is performed.

The transition probabilities of each state depend not only on the amount of previously received l.i. combinations but also on the field chosen to represent the i-th coding coefficient. The model in Figure 3 is a reasonable approximation as described in [19] because, in the decoding process, non-received pivots from the lowest fields dominate the transition probabilities and tend to appear first.

We use a probability-generating function (pgf) approach with the key idea that each stage of the Markov Chain of TC in Figure 3 can be modelled as a geometric distribution and then compute the pmf from operations in the more tractable pgf domain.

Lemma 4.1 (*Telescopic codes probability mass function*). *Consider the case of a single source-destination link without erasures that employs TC of generation size g field size* $\boldsymbol{q}$*. Let* $\mathbb{T}_{TC}$ *be the random number of transmissions required to decode. Then, the pmf of* $\mathbb{T}_{TC}$ *for the probability of decoding in exactly t transmissions is*

$$f_{\mathbb{T}_{TC}}(t;g,\boldsymbol{q}) = \Pr[\mathbb{T}_{TC}=t] = P_g \sum_{l=1}^{L}\sum_{n=1}^{m_l} a_{l,n}\binom{t-g+n-1}{t-g}{\gamma_l}^{t-g},\ t\in[g,\infty) \tag{6}$$

where in (6), $P_g = \Pr[\mathbb{T}_{TC}=g]$ *is the probability of decoding in exactly g transmissions,* $\binom{n}{k}$ *is the binomial coefficient defined as* $\frac{n!}{k!(n-k)!}$ *with* $n \geq k$ *and* $\gamma_l,\ l\in[1,L]$ *is one of the L distinct probabilities from the Markov Chain, which is repeated* m_l *times in the resulting Markov Chain. The* $a_{l,n},\ l\in[1,L],\ n\in[1,m_l]$ *are the residues of the pgf of (6) given by*

$$a_{l,n} = \frac{\lim_{z\to\gamma_l}\frac{d^{m_l-n}}{d(z^{-1})^{m_l-n}}\left(\frac{G_{\mathbb{T}_{TC}}(z^{-1})(1-\gamma_l z^{-1})^{m_l}}{P_g z^{-g}}\right)}{(m_l-n)!(-\gamma_l)^{m_l-n}} \tag{7}$$

Proof. The proof is in the Appendix. □

Corollary 4.1.1 (*RLNC* Distribution). *The distribution and pmf for RLNC is given by*

$$\begin{aligned} f_{\mathbb{T}_{RLNC}}(t;g,q) &= \Pr[\mathbb{T}_{RLNC}=t] \\ &= P_g\sum_{i=1}^{g} a_i{\gamma_i}^{t-g},\ t\in[g,\infty) \end{aligned} \tag{8}$$

where in (8), the a_i *are given as follows:*

$$a_i = \prod_{m=1,m\neq i}^{g}\frac{1}{1-q^{m-i}} \tag{9}$$

Proof. The proof is in the Appendix. □

We include the analysis for a generic erasure ϵ by following a procedure, as made in [27], that considers all the disjoint cases by the law of total probability, where i l.i. coded packets where received in t transmissions and $t-i$ coded packets were received and l.i. The successful receptions are accounted by the negative binomial distribution[†] and the linear independence by (6). Then, the pmf for a unicast session with TC and erasures, $\mathbb{T}_U$, can be expressed as follows:

$$f_{\mathbb{T}_U}(t;g,\boldsymbol{q},\epsilon) = \Pr[\mathbb{T}_U=t] = \sum_{i=g}^{t}\binom{t-1}{i-1}(1-\epsilon)^i\epsilon^{t-i}f_{\mathbb{T}_{TC}}(i;g,\boldsymbol{q}),\ t\in[g,\infty) \tag{10}$$

4.2. Broadcast distribution

For broadcast, we can regard its pmf as the case of finding the required transmissions for the maximum of N independent parallel unicast sessions. This results in an upper bound because we exclude other policies that take advantage of common coded packets. Because each receiver just needs to collect different linear combinations to decode the packets, the number of transmissions will be bounded by the receiver that performs the worst. In this way, the number of transmissions for broadcast is given by $\mathbb{T}_B = \max_{j=[1,N]}\mathbb{T}_{U_j}$, where $\mathbb{T}_{U_j}$ is the distribution on which pmf is obtained from (10). We review this and prove it in the following lemma.

Lemma 4.2 (*Broadcast TC probability mass function*). *Consider the case of a broadcast scenario with heterogeneous packet erasure rates* ϵ_j , $j\in[1,N]$ *that employs TC of generation size g and field sizes* $\boldsymbol{q}$*. Let* $\mathbb{T}_B$ *be the random number of transmissions required to decode. Then, the pmf of* $\mathbb{T}_B$ *for the probability of decoding in exactly t transmissions is*

[†] We use the definition of the negative binomial distribution that accounts for the random number $\mathbb{T}$ of Bernoulli trials of success probability p required to attain n successes. Then, $\mathbb{T}\sim NB(n,p) \implies \Pr[\mathbb{T}=t] = \binom{t-1}{n-1}p^n(1-p)^{t-n}$, $t\in[n,\infty)$.

$$\begin{aligned}&f_{\mathbb{T}_B}(t;N,g,\boldsymbol{q},\epsilon_1,\ldots,\epsilon_N) = \\ &\prod_{j=1}^{N}\left(\sum_{k=g}^{t}\sum_{i=g}^{k}\binom{k-1}{i-1}(1-\epsilon_j)^i\epsilon_j^{k-i}f_{\mathbb{T}_{TC}}(i;g,\boldsymbol{q})\right) \\ &-\prod_{j=1}^{N}\left(\sum_{k=g}^{t-1}\sum_{i=g}^{k}\binom{k-1}{i-1}(1-\epsilon_j)^i\epsilon_j^{k-i}f_{\mathbb{T}_{TC}}(i;g,\boldsymbol{q})\right) \\ &\quad ,\ t\in[g,\infty)\end{aligned} \tag{11}$$

where in (11), $f_{\mathbb{T}_{TC}}(t;g,\boldsymbol{q})$ is the TC pmf given by (6).

Proof. The proof is in the Appendix. □

Corollary 4.2.1 (*Homogeneous erasures in broadcast*)**.** *The pmf of broadcast TC with homogeneous erasure rates, $\epsilon_1,\ldots,\epsilon_N=\epsilon$, is*

$$\begin{aligned}&f_{\mathbb{T}_B}(t;N,g,\boldsymbol{q},\epsilon) = \\ &\left(\sum_{k=g}^{t}\sum_{i=g}^{k}\binom{k-1}{i-1}(1-\epsilon)^i\epsilon^{k-i}f_{\mathbb{T}_{TC}}(i;g,\boldsymbol{q})\right)^N \\ &-\left(\sum_{k=g}^{t-1}\sum_{i=g}^{k}\binom{k-1}{i-1}(1-\epsilon)^i\epsilon^{k-i}f_{\mathbb{T}_{TC}}(i;g,\boldsymbol{q})\right)^N \\ &\quad ,\ t\in[g,\infty)\end{aligned} \tag{12}$$

Proof. It can be easily verified that evaluating (11) with homogeneous erasure rates, $\epsilon_j=\epsilon\ \forall j$, and doing the corresponding algebra, one obtains (12). □

4.3. Single cloud cooperation distribution

For single cloud cooperation, we consider its random number of transmissions for decoding, $\mathbb{T}_{SCC}$, as the sum of two random number of transmissions. First, we consider the random number of transmissions for the cloud to obtain g l.i. coded packets in the cellular stage, $\mathbb{T}_{SCC,cel}$. Second, we add the number of transmissions in the local stage required for all the devices to share their content, $\mathbb{T}_{SCC,loc}$, which is a particular case of broadcast under the proper evaluation. Hence, $\mathbb{T}_{SCC}=\mathbb{T}_{SCC,cel}+\mathbb{T}_{SCC,loc}$. We give a formal definition of this distribution and its proof in the following lemma.

Lemma 4.3 (*Single cloud cooperation TC distribution*)**.** *Consider the case of a single cloud cooperation scenario that employs TC of generation size g and field sizes $\boldsymbol{q}$. The cloud is composed of H heads from N devices with $H\leqslant N$. The heads have heterogeneous packet erasure rates ϵ_j , $j\in[1,H]$ for the links between the source and them. All devices inside the cloud have a homogeneous packet erasure rate ϵ_{loc} for all their $\frac{N(N-1)}{2}$ connection links between them. Then, the distribution for the random number of transmissions required for decoding is given by $\mathbb{T}_{SCC}=\mathbb{T}_{SCC,cel}+\mathbb{T}_{SCC,loc}$ where each term is given as follows:*

$$\mathbb{T}_{SCC,cel}=\mathbb{T}_U\left(g,\boldsymbol{q},\prod_{j=1}^{H}\epsilon_j\right) \tag{13}$$

$$\mathbb{T}_{SCC,loc}=\begin{cases}\mathbb{T}_B(N-H,g,\boldsymbol{q},\epsilon_{loc}),\ H<N\\ \mathbb{T}_B(N_{loc},g_{loc},\boldsymbol{q},\epsilon_{loc})\ ,\ H=N\end{cases} \tag{14}$$

$$N_{loc}=N-\left\lfloor P_g\sum_{j=1}^{N}(1-\epsilon_j)^g\right\rfloor \tag{15}$$

$$g_{loc}=\max_{j\in[1,N]}\left(g-\left\lfloor(1-\epsilon_j)\sum_{i=1}^{g}p_i\right\rfloor\right) \tag{16}$$

where in (13) and (14) the parenthesis notation indicates the evaluation of the respective distribution with the given parameters. In (15) and (16), P_g and p_i are respectively the probabilities of linear independence at g transmissions and in each stage of the TC Markov Chain. Both are defined in the proof of Lemma 4.1.

Proof. The proof is in the Appendix. □

Corollary 4.3.1 (*Homogeneous conditions in single cloud cooperation*)**.** *The distribution of single cloud cooperation TC with homogeneous erasure rates, $\epsilon_1,\ldots,\epsilon_H=\epsilon_{cel}$, is*

$$\mathbb{T}_{SCC,cel}=\mathbb{T}_U\left(g,\boldsymbol{q},\epsilon_{cel}^{H}\right) \tag{17}$$

$$\mathbb{T}_{SCC,loc}=\begin{cases}\mathbb{T}_B(N-H,\boldsymbol{q},\epsilon_{loc}),\ H<N\\ \mathbb{T}_B(N_{loc},g_{loc},\boldsymbol{q},\epsilon_{loc})\ ,\ H=N\end{cases} \tag{18}$$

$$N_{loc}=N-\left\lfloor P_gN(1-\epsilon_{cel})^g\right\rfloor \tag{19}$$

$$g_{loc}=g-\left\lfloor(1-\epsilon_{cel})\sum_{i=1}^{g}p_i\right\rfloor \tag{20}$$

Proof. By evaluating Equations 13–(16) with $\epsilon_j=\epsilon_{cel}$, $\forall j$ □

4.4. Multiple clouds cooperation distribution

For multiple clouds cooperation, we consider its random number of transmissions for decoding, $\mathbb{T}_{MCC}$, again as the sum of two random number of transmissions in a cellular and local fashion as before. Hence, $\mathbb{T}_{MCC}=\mathbb{T}_{MCC,cel}+\mathbb{T}_{MCC,loc}$. However, in this scenario, for the local stage, consider the behaviour of the average user. We first derive the distribution for all the cloud and then evaluate for the average user. We give a formal definition of this distribution and its proof in the following lemma.

Lemma 4.4 (*Multiple clouds cooperation RLNC distribution*)**.** *Consider the case of a multiple clouds cooperation scenario that employs RLNC of generation size g and field* *size q. Each cloud is composed of H_n heads from N_{uc} devices with $H_n \leqslant N_{uc}$. The heads have heterogeneous packet erasure rates ϵ_{jn} , $j \in [1, H_n]$, $n \in [1, C]$ for the links between the source and them. All devices inside each of the C clouds have a homogeneous packet erasure rate ϵ_{loc} for all their $\frac{N(N-1)}{2}$ connection links between them. Then, the distribution for the random number of transmissions required for decoding each cloud is given by $\mathbb{T}_{MCC} = \mathbb{T}_{MCC,cel} + \mathbb{T}_{MCC,loc}$ where each term is given as follows:*

$$\mathbb{T}_{MCC,cel} = \max_{n \in [1,C]} \left(\mathbb{T}_U \left(g, q, \prod_{j=1}^{H_n} \epsilon_{jn} \right) \right) \tag{21}$$

$$\mathbb{T}_{MCC,loc} = \begin{cases} \mathbb{T}_B(N_{uc} - H_n, g, q, \epsilon_{loc}), H_n < N_{uc} \\ \mathbb{T}_B(N_{uc,loc,n}, g_{loc,n}, q, \epsilon_{loc}), H_n = N_{uc} \end{cases} \tag{22}$$

$$N_{uc,loc,n} = N_{uc} - \left\lfloor P_g \sum_{j=1}^{N_{uc}} (1 - \epsilon_{jn})^g \right\rfloor \tag{23}$$

$$g_{loc,n} = \max_{j \in [1,N_{uc}]} \left(g - \left\lfloor (1 - \epsilon_{jn}) \sum_{i=1}^{g} p_i \right\rfloor \right) \tag{24}$$

Where in (21) and (22), the parenthesis notation indicates the evaluation of the respective distribution with the given parameters. In (23) and (24), P_g and p_i are respectively the probabilities of linear independence at g transmissions and in each stage of the RLNC Markov Chain. Both are defined in the proof of Lemma 4.1. Also important in the local stage, the number of transmissions depends on the n-th cloud being considered.

Proof. The proof is in the Appendix. □

With the previous distributions, we simply find a proper operation to observe the behaviour of the system. In our case, we consider the average cloud. To find the trends of the system, we simply evaluate the previous distributions in the homogeneous regime, for example, $\epsilon_{jn} = \epsilon_{cel}$, $\forall j, n$ and $H_n = H_c$, $\forall n$, which makes the average cloud equal to any cloud.

Corollary 4.4.1 (*Homogeneous conditions in multiple clouds cooperation*). *The distribution of multiple clouds cooperation RLNC with homogeneous erasure rates, $\epsilon_{jn} = \epsilon_{cel}$, $\forall j, n$ and $H_n = H_c$, $\forall n$, is the following:*

$$\mathbb{T}_{MCC,cel} = \mathbb{T}_B \left(CH_c, g, q, \epsilon_{cel}^{H_c} \right) \tag{25}$$

$$\mathbb{T}_{MCC,loc} = \begin{cases} \mathbb{T}_B(N_{uc} - H_c, g, q, \epsilon_{loc}),\ H_c < N_{uc} \\ \mathbb{T}_B(N_{uc,loc}, g_{loc}, q, \epsilon_{loc})\ ,\ H_c = N_{uc} \end{cases} \tag{26}$$

$$N_{uc,loc} = N_{uc} - \lfloor P_g N_{uc} (1 - \epsilon_{cel})^g \rfloor \tag{27}$$

$$g_{loc} = g - \left\lfloor (1 - \epsilon_{cel}) \sum_{i=1}^{g} p_i \right\rfloor \tag{28}$$

Proof. The proof is performed by performing the mentioned evaluations in Equations 21–(24). Moreover, we notice that for the cellular stage, all erasure rates are the same, and the distribution for this stage reduces to employing a similar reasoning as the one in Corollary 4.2.1. □

5. PERFORMANCE METRICS

With the pmf for each scenario in Section 4, we compute the mean for the number of transmissions, which allows us to compute the throughput and energy. In the cloud cooperation scenarios, the results are relatively equivalent; however, the pmf in both cellular and local stages will depend on the scenario employed as shown in the previous section. Given this, for notation purposes, we omit the difference between $\mathbb{T}_{cel}$ and $\mathbb{T}_{loc}$ in single cloud cooperation and multiple clouds cooperation unless it is a necessary exception.

5.1. Throughput

We define the throughput in the cloud cooperation scenario for a given set of parameters in the following way:

$$R_{CC} = \frac{g}{\max(t_{cel} E[\mathbb{T}_{cel}], t_{loc} E[\mathbb{T}_{loc}])} \tag{29}$$

In (29), t_{cel} and t_{loc} are the durations of a time slot in the cellular and local stages, respectively. The effective rate perceived by a user will be the information sent divided by the completion time multiplied by a cost. For broadcast RLNC, the throughput is $R_B = \frac{g}{t_{cel} E[\mathbb{T}_B]}$.

5.2. Energy consumption

From the energy point of view, we only consider how much energy is required to transmit and receive a packet due to channel erasures. We do not consider the computational energy consumption because they are up to one order of magnitude below the energy expenditure for transmitting and receiving packets for moderate generation (up to 128) and field sizes (up to 2^8) [17], which we employ in our study. We consider that the energy for being idle is the same as for receiving a packet, because we have also observed that they are very close in practice [17].

To compute the energy consumption, we assume that a packet transmission and a packet reception spend the same energy because we observed this for commercial mobile devices in [17]. Then, we consider that the energy cost depends only on the type of connection employed. Therefore, we have two energy costs: E_{cel} for the cellular interface, and E_{loc} for the local interface. Each cost depends on the packet size and the energy per byte. The latter is proportional to the energy per bit for each type of connection. Then, naming the packet size p_s, the energy per byte E_B and the energy per bit E_b for the cellular stage, we obtain $E_{cel} = p_s E_{B,cel}$ with $E_{B,cel} = 8E_{b,cel}$. Similarly, for the local stage, we obtain $E_{loc} = p_s E_{B,loc}$ with $E_{B,loc} = 8E_{b,loc}$. The energy per bit values are extracted from [28].

For our study, we compute the energy spent for the BS and the average energy spent per device for each transmission scenario. In the following, E_x indicates an energy value, and $E[]$ is the expected value operator for random variables. The energy expenditure for the BS, E_{BS}, is total number of cellular transmissions necessary before the heads in each cloud can decode the content jointly, multiplied by the energy cost of transmitting a packet on the cellular link. The average energy spent by a device, E_D, is computed from the following: (i) the reception of the heads in the cellular stage; (ii) the transmission of the heads in the local stage; and (iii) the reception of the non-heads in the local stage. The average is computed by dividing the previous total energy by the number of devices.

First, the energy consumption for broadcast is as follows:

$$E_{BS} = E_{cel}E[\mathbb{T}_B]\,, E_D = E_{cel}E[\mathbb{T}_B] \tag{30}$$

Second, the energy expenditure for the single cooperation scenario is shown in (31).

$$\begin{aligned} E_{BS} &= E_{cel}E[\mathbb{T}_{cel}] \\ E_D &= E_{cel}\left(\frac{H}{N}\right)E[\mathbb{T}_{cel}] + E_{loc}E[\mathbb{T}_{loc}] \end{aligned} \tag{31}$$

For the multiple clouds cooperation scenario, the result is equivalent with the number of heads and users per cloud equal to H_c and N_{uc}:

$$\begin{aligned} E_{BS} &= E_{cel}E[\mathbb{T}_{cel}] \\ E_D &= E_{cel}\left(\frac{H_c}{N_{uc}}\right)E[\mathbb{T}_{cel}] + E_{loc}E[\mathbb{T}_{loc}] \end{aligned} \tag{32}$$

5.3. Cellular versus local links

The performance of cooperation will depend on the slot rate and energy use per bit costs on the local and the cellular links. Therefore, we define the r_t as the ratio between cellular and local slot rate costs, and r_e as the ratio between the cellular and local energy cost.

$$r_t = \frac{t_{loc}}{t_{cel}} = \frac{R_{cel}}{R_{loc}} \;\; ; \;\; r_e = \frac{E_{b,cel}}{E_{b,loc}} \tag{33}$$

5.4. Gain regions

For the analysis with different erasure rates per stage, we define the throughput and energy gains of cloud cooperation against broadcast RLNC from (30) and (31) as shown in (35).

$$\begin{aligned} G_t &= \frac{E[\mathbb{T}_{cel}]}{\max(r_t E[\mathbb{T}_{cel}], E[\mathbb{T}_{loc}])} \\ G_e &= 1 - \frac{r_e\left(\frac{H}{N}\right)E[\mathbb{T}_{cel} + \mathbb{T}_{loc}]}{r_e E[\mathbb{T}_{loc}]} \end{aligned} \tag{34}$$

We define throughput gain as the ratio of the cloud cooperation and broadcast RLNC throughputs. The energy gain of cooperation over broadcast is defined as the saving in energy for the devices, because cooperation always save energy at the BS.

5.5. Optimal cloud size

All the studied scenarios can be regarded as the spectrum of cooperation, where broadcast is the case of no cooperation, single cloud cooperation the case of full cooperation and multiple clouds cooperation the in-between. Then, we define the optimal cloud size as the size that all the clouds should have in order to minimise the total transmission time. We consider this because we have observed that in the case of having the same erasure rate for all the links in the cellular stage and the same erasure rate for all the links in the local stage, employing the same cloud size for all the clouds, two main situations dominate either the cellular or the local transmission time. First, the cellular transmission time mostly depends on the smallest cloud, for example, the one with the least amount of devices. Second, the local transmission time depends on the remaining devices to be served, which is proportional to the biggest cloud. We verify this in the results section and then focus on the homogeneous case because it gives the best performance. For this case, there will be a trade-off for the number of transmissions in each stage. For the broadcast case, we simply consider that $N_{uc} = 1$ and $\mathbb{T}_{MCC,loc} = 0$. Thus, the optimal cloud size is defined as follows:

$$N_{uc}^* = \min_{N_{uc}} \mathbb{T}_{MCC} \tag{35}$$

5.6. Overhead

We calculate the performance of TC against RLNC by reviewing the optimal field choices that minimises an overhead-related cost function and compare them against the performance of both RLNC with $GF(2)$ and $GF(2^8)$ given that they represent opposite extremes.

We define the overhead for a field scheme s and transmission scenario t as

$$\mathbb{O}_{s,t} = (B + |\boldsymbol{v}|_{s,t})(\mathbb{T}_{s,t} - \mathbb{T}_{s_{min},t}) + |\boldsymbol{v}|_{s,t}\mathbb{T}_{s_{min},t} \text{ [bits]} \tag{36}$$

In (36), $\mathbb{O}_{s,t}$ is the overhead viewed as random variable depending on a given scheme and transmission scenario. $|\boldsymbol{v}|_{s,t}$ is the coding coefficients overhead for the given scheme and scenario. $\mathbb{T}_{s,t}$ is the (random) number of transmissions of the given scheme and scenario and $\mathbb{T}_{s_{min},t}$ is a random variable for the minimum amount of transmissions that the scheme might take in the given scenario. A reasonable approximation for this variable is $\mathbb{T}_{GF(2^8),t}$. Given that we are evaluating the overhead, our choice for the cost function to obtain the optimal field scheme is the mean overhead. Then, the optimal field scheme for a given scenario is the one that minimises the following cost function (after rearranging terms):

$$\begin{aligned} \min_{\boldsymbol{q}} \quad & (B + |\boldsymbol{v}|_{\boldsymbol{q},t}) E[\mathbb{T}_{\boldsymbol{q},t}] \\ \text{s.t.} \quad & \boldsymbol{q}_i = 2^{2^{k_i}},\ i \in [1, g],\ k_i \in \mathbb{Z}^+ \end{aligned} \tag{37}$$

In the nonlinear integer problem defined in (37), we have substituted the scheme subscript to highlight the dependence on the fields of the mean overhead minimisation because the optimal scheme is a particular choice of fields. For a given solution of (37), we evaluate its cost in (36) to review the performance of the given scheme.

6. NUMERICAL RESULTS

We use a set of parameters in the following ranges $1 \leqslant N \leqslant 50$, $g = \{64, 128\}$, $q = 2^8$ and $0 \leqslant \epsilon_{cel} = \epsilon_{loc} = \epsilon \leqslant 0.6$. The time slot duration is set to $t_{cel} = 0.5$ ms to conform to the LTE-A E-UTRA [29] and its set of D2D specifications. For the energy, we extracted the energy per bit cost from the energy model in [28] and use a packet size N_B of 500 B. For RLNC, in case of increasing the packet size (e.g. 1.5 KB) while keeping the total amount of data, the overhead contribution from the coding coefficients will be low because the amount of bits required to send the coding coefficients will be less than the required packet size, reducing the required signalling. For the case of a low packet size (e.g. 100 B), the overhead due to the coding coefficients per packet increases because it is comparable with or even higher than the packet size. Then, for very low packet sizes, most of the information sent is mainly signalling reducing the throughput. For the overhead of TC, we evaluate broadcast and single cloud cooperation with a set of parameters in the following ranges $N = \{1, 30, 50\}$, $g = \{16, 32, 64, 128\}$, $\epsilon_{cel} = \epsilon_{loc} = \epsilon = \{0.1, 0.3, 0.5\}$. We use a representative wireless network packet size of 1.6 KB ($B = 12\,800$ bits). For the optimal field scheme, we obtain the solutions that minimise the cost function in (37) by performing a search for the solutions in the feasible set of (37) and verifying which minimises the cost function. The considered field sizes for the feasible set were 2, $2^2, 2^4, 2^8, 2^{16}$ and 2^{32} as in [19], because current computer data types can easily represent these values. Following, we make the comparison in percentage value obtained as $E[\mathbb{O}_{s,t}]/gB \times 100\%$.

Figure 4 shows how the throughput varies depending on the ratio of the cellular and local data rate. The ratios are obtained by fixing the cellular data rate and varying the local data rate. When the local data rate is lower than the cellular rate, the cooperative schemes provide lower throughput than the broadcast scheme. Conversely, when the local data rate is higher than the cellular data rate, the cooperative schemes deliver a higher throughput than broadcast. The throughput is highest when the local links rate are twice as faster as the cellular ones. The number of heads controls how much gain can be obtained and where it occurs for a given ratio. When the number of heads decreases, the throughput also diminishes because there are fewer heads each with an independent chance of receiving the packet.

Figure 5 shows how the energy for the devices changes as the ratio between the cellular and local energy per bit changes. The energy cost in the cellular link is fixed and the cost on the local link is changed to obtain the different ratios; consequently, the energy per bit for broadcast is constant.

When the energy cost for the local links is higher than the cellular energy cost, the cooperation performs worse than broadcast. The extra consumption for cooperation comes from the transmissions in the local stage. Contrarily, when the cost of the local links is lower than the cost of

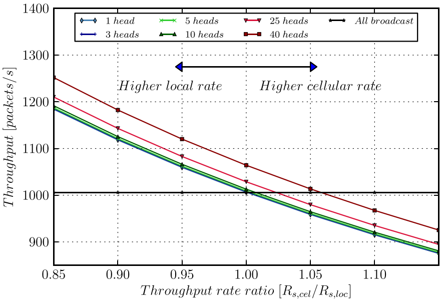


Figure 4. The throughput of broadcast and cooperation with different number of heads, for different ratios between the data rate on the cellular and local link. Used parameters: $g = 64$, $q = 2^8$, $\epsilon = 0.4$, $N = 50$.

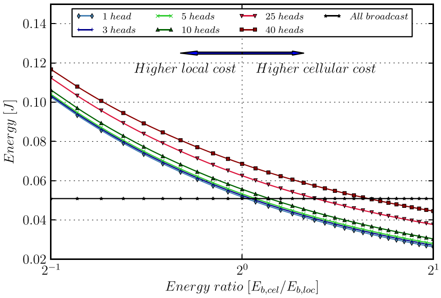


Figure 5. The energy per generation of broadcast and cooperation with different number of heads, for different ratios between the energy per bit on the cellular and local link. Used parameters: $g = 64$, $q = 2^8$, $\epsilon = 0.4$, $N = 50$.

the cellular links, single cooperation uses less energy than broadcast. In these cooperation scenarios, the consumption is determined by the number of heads on the cellular stage.

For a cloud of a determined size, in the case of having the same erasure rate for all the links in the cellular stage, there is a trade-off between throughput and energy expenditure that depends on the number of heads. The higher number of heads, the higher the throughput. When a coded packet is broadcasted to the cloud in the cellular stage, having more heads cooperating with each other rapidly increases the probability that at least one of them obtains it, to later share this knowledge with all the devices. This in turn increments the probability of having all the packets inside the cloud in g transmissions, enhancing the throughput. However, this comes at the expense of higher energy consumption because more energy is spent when receiving the packets to the cloud. For a low amount of heads, the energy expenditure for the devices is low, but the same for the throughput given that more transmission is required because less devices are cooperating. At the end, it is a design decision because both benefits cannot be achieved at the same time.

Figure 6 shows the regions where cooperation provides a gain in terms of throughput for a wide range of erasure rates on the cellular and local links. The lines show where broadcast and cooperation perform the same, for $r_t = [0.5, 0.8, 1, 1.5, 2]$. In the region below each line, cooperation provides higher throughput than broadcast for that particular r_t. Above the line, broadcast performs better. For example, in the case of a fast local link $r_t = 0.5$, then cooperation provides a gain for almost all considered erasure rates, even in cases where the local erasure rate is much higher than the cellular.

Figure 7 shows the regions where cooperation provides a gain in terms of energy saving on the devices for various erasure rates on the cellular and local links. The lines show where broadcast and cooperation performs the same, for $r_e = [0.5, 0.75, 1, 1.5, 2]$. In the region below each line, cooperation provides a lower energy per bit than broadcast for that particular r_e. Above the line, broadcast performs better.

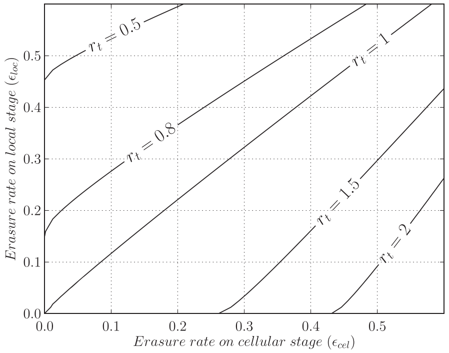


Figure 6. For different values of r_t, the lines indicate where cooperation and broadcast provide the same throughput for various erasure rates on the cellular and local links. Below each line, cooperation performs better for the respective r_t. Used parameters: $g = 128,\ q = 2^8,\ H = 40,\ N = 50$.

Figure 8 shows the results for two simulations. The first simulation shows the cellular transmission time for two different scenarios. In the first scenario, for the same losses in the cellular stage, $\epsilon_{cel} = 0.3$, we show the cellular transmission time for two clouds where the first cloud size is fixed to six devices and the second cloud size ranges from one to 20 devices. In the second scenario, for the same losses in the cellular stage, we show the transmission time for five clouds where all the cloud sizes are the same. We vary the size of all these clouds from one to 20 devices.

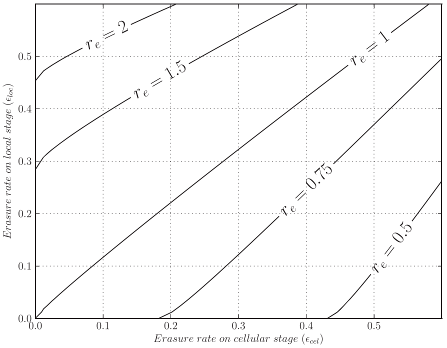


Figure 7. For different values of r_e, the lines indicate where cooperation and broadcast provide the same energy per bit for various erasure rates on the cellular and local links. Below each line, cooperation performs better for the respective r_e. Used parameters: $g = 128,\ q = 2^8,\ H = 40,\ N = 50$.

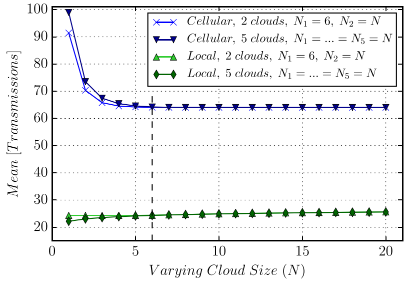


Figure 8. For the cellular transmission time, the smallest cloud dominates the transmissions below the homogeneous size (dashed line). Above it, it remains mostly constant. For the local transmission time, it remains constant below the homogeneous size (dashed line). Above it, the biggest cloud starts to slowly dominate the transmissions. Used parameters: $g = 64,\ q = 2^8,\ \epsilon_{cel} = 0.3,\ \epsilon_{cel} = 0.1,\ H = N,\ N$ varies.

For the first scenario, we observe that as the number of devices in the second clouds is less than six (dashed line), then transmission time in the cellular stage increases given that this cloud requires more transmissions than the first cloud. For the second scenario, as we increase the number of devices in the five clouds, the error between one varying cloud and five varying clouds becomes very small.

The second simulation shows the trends for the local transmission time similarly in two scenarios. In the first scenario, assuming equal losses in the local stage, $\epsilon_{loc} = 0.1$, we present the local transmission time for two clouds keeping fixed the size of the first one to six devices and varying the size of the second from one to 20 devices. In the second scenario, for equal losses in the local stage, we present the local transmission time where all the clouds have the same size, and we vary them from one to 20 devices. Here, for the first scenario, we notice that the local transmission time is higher when second cloud size is bigger than the first one. The reason being that the second cloud becomes the biggest one and dominates the local transmission time. Similarly, for the second scenario, we observe that the local transmission time matches for both cases after six devices (dashed line).

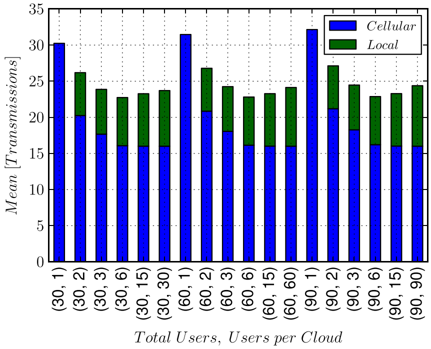


Figure 9. Optimal cloud size for $g = 16$, $q = 2^8$, $\epsilon_{cel} = 0.3$ and $\epsilon_{loc} = 0.1$.

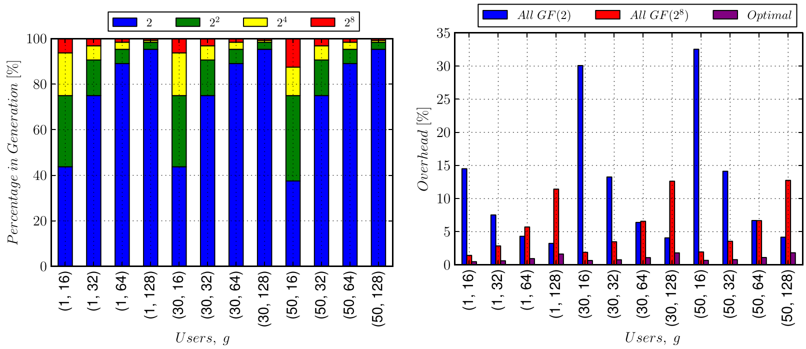
At the end, we see that the homogeneous is the best case that we could obtain. In any other case, one of the following two situations occurs: (i) the cellular transmission time is high, while the local transmission time is constant, or (ii) the cellular transmission time remains constant, while local transmission time is high. The homogeneous case is simply the boundary between the previous two.

Figure 9 shows the trade-off for the number of users per cloud for different given erasure rate in the cellular and local stages. We clearly observe that there is a cloud size N_{uc} for which the total number of transmissions $\mathbb{T}_{MCC}$ is minimum, thus minimising the energy consumption and maximising the throughput.

We study the overhead performance for the three schemes for both scenarios to compare the optimal scheme performance.

6.1. Optimal telescopic codes configuration and performance for broadcast

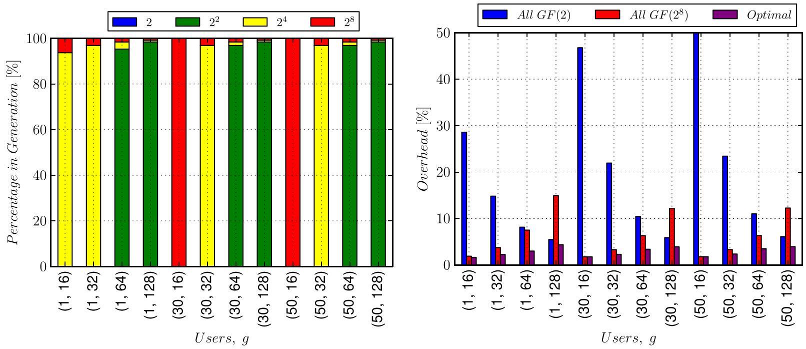
Figure 10(a) first shows the optimal scheme obtained from solving (37). Second, Figure 10(b) shows the corresponding optimal scheme overhead against the overhead from the other two code schemes. Both results are for 30% losses in all the links of the remote stage. Both the optimal field scheme and overhead are presented versus the combinations of users, N, and generation sizes, g. Each bar in Figure 10(a) indicates the amount of coding coefficients for the given fields as percentages in the generation. For all shown combinations in Figure 10(a), as g increases for a fixed amount of users, most of the coefficients are drawn from $GF(2)$ with a diminishing percentage being chosen from other fields. Nevertheless, for a fixed generation size and increasing number of users, only for low values of g it can be observed a tendency to use more coding coefficients in high fields. Figure 10(b) exhibits the overhead mean of the three schemes. For all the cases, we see that the optimal



(a) Optimal field scheme, q^*

(b) Overhead mean (%)

Figure 10. Schemes performance for broadcast. $\epsilon = 0.3$.



(a) Optimal field scheme, q^*

(b) Overhead mean (%)

Figure 11. Schemes performance for cloud cooperation. $\epsilon = 0.3$.

Table I. Scenarios and schemes comparison.

			Transmission time				
Transmission scenario	Code scheme	Overhead	Cellular	Local	Total	*Energy	*Throughput
Broadcast	RLNC	Med	High	None	High	Med	Low
	Telescopic	Low	High	None	High	Low	Low
Single cloud cooperation	RLNC	Med	Low	Med	Med	High	Med
	Telescopic	Low	Low	Med	Med	High	Med
Multiple clouds cooperation	RLNC	Med	Low	Low	Low	Med	High
	Telescopic	Low	Low	Low	Low	Low	High

$\epsilon_{cel} = 0.3, \epsilon_{loc} = 0.1,\ N = 60,\ H = N,\ r_e \leqslant 1,\ r_t \geqslant 1.$
RLNC, random linear network coding.

field scheme outperforms both $GF(2)$ and $GF(2^8)$ achieving a less than 2% total overhead in most of the cases. For $g \leqslant 32$, increasing the number of receivers greatly affects the total overhead, almost doubling it in the case of $g = 16$. Although, for $g \geqslant 64$, the overhead is less sensitive to the number or receivers.

6.2. Optimal telescopic codes configuration and performance for single cloud cooperation

Correspondingly, Figure 11 exhibits the performance for cloud cooperation. Figure 11(a) presents the optimal scheme obtained from solving (37) for cloud cooperation and 30% losses for both the remote and local links. In this case, for increasing g and fixed users amount, more than 90% of the coding coefficients belong to a single field. For $g \leqslant 32$, the scheme distribution goes mostly to either $GF(2^4)$ or $GF(2^8)$. For $g \geqslant 64$, most of the coefficients are chosen from $GF(2^2)$. For a fixed generation size and varying amount of users, the coding coefficients distribution appreciably changes for $g = 16$ and slightly for $g = 64$. Figure 11(b) displays the overhead mean; the optimal field scheme provides a lower mean overhead than both $GF(2)$ and $GF(2^8)$, except for the cases of 30 and 50 receivers with 16 packets as generation size, where the optimal scheme and subsequently mean overhead are the same as $GF(2^8)$. Still, for all the cases, total overhead mean does not exceed 2.5% with some cases being approximately less than 1%.

Finally, in Table I, we present a solution comparison to show our major results. We consider a packet erasure rate of 30% for the cellular links and 10% for the local links. Similarly, we consider a large fixed amount users, $N = 60$, with all of them being heads. Also, the energy cost in the cellular stage is equal or higher as the energy cost in the local stage ($r_e \leqslant 1$). Similarly, we consider that duration of a time slot in the cellular stage is the same as in the local stage ($r_t \geqslant 1$). Under this scenario, the multiple clouds cooperation approach provides the best performance in terms of throughput. Thus, we expect the highest gains from cooperation and coding in this regime.

7. CONCLUSIONS

In this work, we presented an in-depth study of the specific operating regions where cooperation provides gains in throughput and energy over coded broadcasting techniques. Our numerical results showed that gains can be achieved even if the long-range and short-range technologies transmit at comparable data rates or there are few differences in the erasure rates in each stage. More importantly, we showed that cooperation can provide several fold gains broadcast as long as the short-range link is at least twice as fast as the long-range one. Finally, our results showed that a moderate number of heads (e.g. three or more) per cooperative cluster is enough to yield the high throughput gains while maintaining a low energy consumption at the receivers. Also, for multiple clouds cooperation, we observed that the cloud size should be around six nodes in most of the cases.

We also proposed the use of TC for network-coded cooperative to reduce the total system overhead. We review the performance of TC against classical RLNC systems using either $GF(2)$ or $GF(2^8)$ with two common multicast scenarios: broadcast and single cloud cooperation. For a broadcast scenario, we observed that the optimal field scheme always outperforms both traditional RLNC schemes where in some cases total overhead is less than 0.5%. For single cloud cooperation, the total overhead was less than approximately 3% in all the cases. In a single cloud cooperation scenario, the penalty for including coefficients from low fields becomes significant because of the presence of the hops. However, this becomes less critical as the generation size increases.

Future work shall focus on protocol design for cooperative schemes in highly dense scenarios as well as implementation and evaluation of the most promising schemes in Aalborg University's Raspberry Pi testbed [30]. Regarding optimal overhead codes, future work in this area should consider the inclusion of other scenarios such as multiple clouds cooperation and multihops.

APPENDIX

Proof of Lemma 4.1

Let $\mathbb{T}_{TC,i}$ be a random variable for the number of transmissions needed to receive an l.i. coded packet in a stage of the Markov Chain for TC in Figure 3. This is a geometric distribution[†] with success probability given by $p_i = 1 - \boldsymbol{q}_i{}^{-g+(i-1)}, i \in [1, g]$. In general, $\boldsymbol{q}_i = 2^{2^{k_i}}, k_i \in \mathbb{Z}^+$. Later, the distribution for TC is given as follows:

$$\mathbb{T}_{TC} = \sum_{i=1}^{g} \mathbb{T}_{TC,i} \tag{A.1}$$

A direct computation of (A.1) is analytically intractable due to the requirement of computing $g-1$ discrete convolutions. Nevertheless, in the pgf domain, this operation turns into a product making the analysis fairly easier.

First, let us define the pgf of a discrete random variable $\mathbb{T}$ from its pmf as $G_{\mathbb{T}}(z) = E[z^{\mathbb{T}}] = \sum_{t=-\infty}^{\infty} \Pr[\mathbb{T} = t]z^t = \sum_{t=-\infty}^{\infty} f_{\mathbb{T}}(t) z^t$. Second, the bilateral Z-transform of the pmf as $F(z) = \mathcal{Z}\{\Pr[\mathbb{T} = t]\} = \sum_{t=-\infty}^{\infty} \Pr[\mathbb{T} = t] = \sum_{t=-\infty}^{\infty} f_{\mathbb{T}}(t) z^{-t}$. Doing little algebra with the previous two definitions, it can be seen that $F(z) = G_{\mathbb{T}}(z^{-1})$. Thus, the Z-transform of (A.1) becomes

$$F_{\mathbb{T}_{TC}}(z) = \prod_{i=1}^{g} G_{\mathbb{T}_{TC,i}}(z^{-1}) \tag{A.2}$$

Calculating the pgf of $\mathbb{T}_{TC,i}$ with the previous definitions and inserting in (A.2), we obtain the following:

$$\begin{aligned} F_{\mathbb{T}_{TC}}(z) &= G_{\mathbb{T}_{TC}}(z^{-1}) \\ &= \prod_{i=1}^{g} \frac{p_i z^{-1}}{1-(1-p_i)z^{-1}} \ , \ |z| > \max(1-p_i) \end{aligned} \tag{A.3}$$

In (A.3), we notice $\prod_{i=1}^{g} p_i$ is simply the probability of decoding in exactly g transmissions, $\Pr[\mathbb{T}_{TC} = g]$, which we relabel as P_g. Also, $\gamma_i = 1 - p_i$ is probability of obtaining an l.d. in stage i. Including this (A.3), we obtain

$$F_{\mathbb{T}_{TC}}(z) = P_g z^{-g} \prod_{i=1}^{g} \frac{1}{1-\gamma_i z^{-1}} \ , \ |z| > \max(\gamma_i) \tag{A.4}$$

From (A.4), we analytically observe the following: First, as expected, the pmf will be a right handed, causal and stable sequence from the signal processing perspective. The reason being that the pmf Z-transform Region of Convergence includes $|z| = \infty$ and $|z| = 1$ because $\max(\gamma_i) < 1$ always. Also, the pmf sequence will begin at g because of the delay term z^{-g}, which makes reference to the fact that g transmissions are required to receive g l.i. packets.

At this point, we make the following observation: Depending on the field distribution considered, at least some of the l.d. probabilities may be equivalent to each other. Thus, in general, the Z-transform of the TC, (A.4), may have repeated roots in its pgf. Therefore, let us consider that we have $\gamma_l,\ l \in [1, L]$ distinct l.d. probabilities in the Markov Chain in Figure 3, each repeated m_l times with $\sum_{l=1}^{L} m_l = g$. Then, (A.4) becomes (A.5):

$$F_{\mathbb{T}_{TC}}(z) = P_g z^{-g} \prod_{l=1}^{L} \frac{1}{(1-\gamma_l z^{-1})^{m_l}} \ , \ |z| > \max(\gamma_l) \tag{A.5}$$

Afterwards, we perform a partial fraction expansion on the product term in (A.5), which is rational, which turns it into

$$F_{\mathbb{T}_{TC}}(z) = P_g z^{-g} \sum_{l=1}^{L} \sum_{n=1}^{m_l} \frac{a_{l,n}}{(1-\gamma_l z^{-1})^n} \ , \ |z| > \gamma_{l_{max}} \tag{A.6}$$

In (A.6), we have splitted the product as a sum of the contributions of each of its poles of the Z-transform in (A.6). Here, RLNC appears as a special subcase due to the

[†] We use the definition of the geometric distribution that accounts for the random number $\mathbb{T}$ of Bernoulli trials of success probability p required to obtain the first success. Later, if $\mathbb{T} \sim Geom(p) \implies \Pr[\mathbb{T} = t] = (1-p)^{t-1}p \ , \ t \in [1, \infty)$

linear dependence probabilities being unique in that case. Hence, we can obtain the pmf for RLNC as a subcase of the for TC.

From (A.6), the $a_{l,n}$ coefficients $a_{l,n}$, $l \in [1,L]$, $n \in [1,m_l]$ are the residues of the complex function $\left[\prod_{m=1}^{L}(1-\gamma_m z^{-m_l})\right]^{-1}$ at the poles $z = \gamma_l$,$l \in [1,L]$. These residues are calculated using the expression:

$$a_{l,n} = \frac{\lim_{z\to\gamma_l} \frac{d^{m_l-n}}{d(z^{-1})^{m_l-n}} \left(\frac{G_{\mathbb{T}_{TC}}(z^{-1})(1-\gamma_l z^{-1})^{m_l}}{P_g z^{-g}}\right)}{(m_l-n)!(-\gamma_l)^{m_l-n}} \quad \text{(A.7)}$$

Performing this evaluation and doing inverse Z-transform algebra, the one obtains (6), which concludes the proof.

Proof of Corollary 4.1.1 The proof comes by letting $\boldsymbol{q}_i = q \; \forall i \in [1,g]$, which gives different unique l.d. probabilities in the Markov Chain, which in turns gives all simple poles (A.4). This makes quite easy to evaluate the residues as $a_i = \lim_{z\to\gamma_i}\left[\prod_{m=1}^{g}(1-\gamma_m z^{-1})\right]^{-1} \times \left[(1-\gamma_i z^{-1})\right]$ and noting that the relationship $\gamma_m\gamma_i^{-1} = q^{m-i}$ between the l.d. probabilities giving the result in (8), and this concludes the proof.

Proof of Lemma 4.2 For the probability of the maximum number of transmissions of the worst receiver to be less than or equal to t transmissions, then necessarily all other receivers must also have this (or in general less) transmissions than the worst receiver.

Therefore, under the packet erasures independence assumption, we can compute the cumulative density function (CDF) for broadcast with TC, as $F_{\mathbb{T}_B}(t) = \Pr[\mathbb{T}_B \leqslant t] = \prod_{j=1}^{N}\Pr[\mathbb{T}_{U_j} \leqslant t]$ with $\Pr[\mathbb{T}_{U_j} \leqslant t]$ being the CDF obtained from the pmf in (10) with the resulting CDF in (A.8).

$$F_{\mathbb{T}_B}(t; N, g, \boldsymbol{q}, \epsilon_1, \ldots, \epsilon_N) = \\ \prod_{j=1}^{N}\left(\sum_{k=g}^{t}\sum_{i=g}^{k}\binom{k-1}{i-1}(1-\epsilon_j)^i\epsilon_j^{k-i}f_{\mathbb{T}_{TC}}(i;g,\boldsymbol{q})\right) \quad \text{(A.8)}$$

Finally, to obtain the pmf for broadcast, we simply compute $f_{\mathbb{T}_B}(t) = F_{\mathbb{T}_B}(t) - F_{\mathbb{T}_B}(t-1)$, which gives (11).

Proof of Lemma 4.3 In the cellular stage, each coded packet is acknowledged as received if at least one head obtains it, regardless if it is l.i. or not. This event occurs with probability $1 - \prod_{j=1}^{H}\epsilon_j$ because all links need to fail for a packet to not be received. Therefore, the distribution of the number of transmissions to obtain a coded packet is $Geom(1 - \prod_{j=1}^{H}\epsilon_j)$. Because we need to account for g packets, the distribution for receiving all the packets without considering the coding scheme is $NB(g, 1 - \prod_{j=1}^{H}\epsilon_j)$ given that it is the sum of g geometric distributions. Later, we couple the resulting pmf with the coding scheme one by following the same procedure as in the proof of Lemma 4.1 to obtain (10). Doing the calculations, we obtain (13).

In the local stage, the heads take turns to broadcast their content between all devices in the cloud. Given that we have assumed coordination among the heads, any temporal transmitting head behaves as a source broadcasting recoded packets. Then, the pmf for this stage is a particular case of (12).

At this point, we differentiate two cases: not all heads and all heads. For the former, we make the approximation that the non-heads will become the dominant factor in the broadcast pmf, given that they have no collected packets from the previous stage. In case of the latter, there is no dominant set of devices with a particular number of packets. So, in general, we need to exclude from the total amount of devices, N, the devices that may have finished before (although depending on the conditions this number might be very low) and account that some devices have some packets already.

To obtain an average number of devices that may have finished in the cellular stage for the all heads case, N_f, we approximate it as the mean for the random number of devices that have g l.i. packets after exactly g transmissions have occurred. We round down this to provide an integer result, so $N_f = \lfloor E[\mathbb{N}_f] \rfloor$. Assuming that g transmissions occur in the cellular stage is reasonable because the mean of the distribution in (13) is $g/(1 - \prod_{j=1}^{N}\epsilon_j)$. This mean tends to g rapidly for practical values of the ϵ_j and N.

To calculate the distribution of the devices that have finished, $\mathbb{N}_f$, we notice that each device meeting the previous condition can be regarded as a Bernoulli trial with success probability $P_g(1-\epsilon_j)^g$ because all packets must be received and l.i. for each device independently. So, we consider $\mathbb{N}_f = \sum_{j=1}^{N} Bernoulli(P_g(1-\epsilon_j)^g)$ and taking the mean to this expression gives $E[\mathbb{N}_f] = \sum_{j=1}^{N} E[Bernoulli(P_g(1-\epsilon_j)^g)] = P_g\sum_{j=1}^{N}(1-\epsilon_j)^g$. Later, $N_f = \lfloor E[\mathbb{N}_f] \rfloor = \lfloor P_g\sum_{j=1}^{N}(1-\epsilon_j)^g \rfloor$ from which we obtain $N_{loc} = N - N_f$, which gives (15).

To obtain how many packets we need to transmit in the local stage, we calculate how many packets does each device j, $j \in [1,N]$ has on average after g transmission have occurred rounded down, $\lfloor E[\mathbb{G}_j] \rfloor$. Then, we will transmit as many coded packets as required to ensure that the device that has the less number of packets from the cellular stage acquires the whole generation, for example, $g_{loc} = \max_j(g - \lfloor E[\mathbb{G}_j] \rfloor)$.

For the j-th device, in g transmissions, it will have received $\mathbb{G}_j$ l.i. packets. The i-th transmission with $i \in [1,g]$ can be regarded as a Bernoulli trial that has success probability $p_i(1-\epsilon_j)$. Thus, $\mathbb{G}_j = \sum_{i=1}^{g} Bernoulli(p_i(1-\epsilon_j))$ for which its mean is $(1-\epsilon_j)\sum_{i=1}^{g} p_i$. Then, doing the remaining algebra, we obtain (16), and this concludes the proof.

Proof of Lemma 4.4 This proof is similar as the preceding one, but some differences occur. First, instead of broadcasting to a single cloud in the cellular stage, we broadcast to $1 < n < C$ clouds, so the number of transmissions will be upper bounded by the worst cloud, hence the computation as described in (21). Second, in the local stage, we calculate the new parameters for evaluating the distributions in each cloud depending on the cloud being considered in general because the parameters depend on the erasure rates of cellular stage. Nevertheless, the results from the proof of Lemma 4.3 still hold.

ACKNOWLEDGEMENTS

This research has been partially financed by the CROSS-FIRE MITN Marie Curie project (317126) from the European Commission FP7 framework and the Green Mobile Cloud project (grant no. DFF - 0602- 01372B) granted by the Danish Council for Independent Research. Also, we would like to thank former Vodafone CTO, Eng. Hartmut Kremling, for providing data consumption trends in current well-deployed mobile networks.

REFERENCES

1. Kremling H. *Innovation at Vodafone: 5G and the Internet of Things*, Vodafone Annual Meeting: Dresden, Germany, 2015.
2. Fitzek FHP, Katz MD. *Cooperation in Wireless Networks: Principles and Applications—Real Egoistic Behaviour is to Cooperate!* Springer: Dordrecht, The Netherlands, 2006. https://books.google.dk/books?id=iSWmPZS_yvkC [available on 15 December 2015].
3. Lin X, Andrews JG, Ghosh A. A comprehensive framework for device-to-device communications in cellular networks. *arXiv preprint ArXiv:1305.4219* 2013.
4. 3GPP. Feasibility study for proximity services (prose). *TR 22.803. Release 12*, 3rd generation partnership project; technical specification group: system aspects (SA), 2012. http://www.3gpp.org/ftp/Specs/html-info/22803.htm [available on 15 December 2015].
5. Fitzek FHP, Katz MD. *Mobile Clouds: Exploiting Distributed Resources in Wireless, Mobile and Social Networks*. Wiley: Hoboken, New Jersey, USA, 2013. https://books.google.dk/books?id=s2lXAgAAQBAJ [available on 15 December 2015].
6. Heide J, Fitzek FH, Pedersen MV, Katz M. Green mobile clouds: network coding and user cooperation for improved energy efficiency. In *Cloud Networking (Cloudnet), 2012 IEEE 1St International Conference on*, Paris, France, 2012, IEEE; 111–118.
7. Ahlswede R, Cai N, Li S-YR, Yeung RW. Network information flow. *Information Theory, IEEE Transactions on* 2000; **46**(4): 1204–1216.
8. Ho T, Médard M, Koetter R, Karger DR, Effros M, Shi J, Leong B. A random linear network coding approach to multicast. *Information Theory, IEEE Transactions on* 2006; **52**(10): 4413–4430.
9. Eryilmaz A, Ozdaglar A, Médard M, Ahmed E. On the delay and throughput gains of coding in unreliable networks. *Information Theory, IEEE Transactions on* 2008; **54**(12): 5511–5524.
10. Khamfroush H, Lucani DE, Barros J. Minimizing the completion time of a wireless cooperative network using network coding. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, London, UK, 2013, IEEE; 2016–2020.
11. Khamfroush H, Lucani DE, Pahlevani P, Barros J. On optimal policies for network coded cooperation: theory and implementation. *IEEE Journal on Selected Areas in Communications* 2015; **33**(2): 199–212.
12. Heide J, Pedersen MV, Fitzek FH, Larsen T. Network coding for mobile devices-systematic binary random rateless codes. In *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, Dresden, Germany, 2009, IEEE; 1–6.
13. Chou PA, Wu Y, Jain K. Practical network coding, 2003.
14. Trullols-Cruces O, Barcelo-Ordinas JM, Fiore M. Exact decoding probability under random linear network coding. *Communications Letters, IEEE* 2011; **15** (1): 67–69.
15. Zhao X. Notes on exact decoding probability under random linear network coding. *Communications Letters, IEEE* 2012; **16**(5): 720–721.
16. Heide J, Pedersen MV, Fitzek FH, Médard M. On code parameters and coding vector representation for practical RLNC. In *Communications (ICC), 2011 IEEE International Conference on*, Kyoto, Japan, 2011, IEEE; 1–5.
17. Paramanathan A, Pedersen MV, Lucani DE, Fitzek FH, Katz M. Lean and mean: network coding for commercial devices. *Wireless Communications, IEEE* 2013; **20** (5): 54–61.
18. Li X, Mow WH, Tsang FL. Singularity probability analysis for sparse random linear network coding. In *Communications (ICC), 2011 IEEE International Conference on*, Kyoto, Japan, 2011, IEEE; 1–5.
19. Heide J, Lucani D. Composite extension finite fields for low overhead network coding: telescopic codes. In *IEEE International Conference on Communications (ICC)*, London, UK, 2015.
20. Chiti F, Fantacci R, Schoen F, Tassi A. Optimized random network coding for reliable multicast communications. *Communications Letters, IEEE* 2013; **17** (8): 1624–1627.
21. Tassi A, Chatzigeorgiou I, Vukobratovic D. Resource-allocation frameworks for network-coded layered multimedia multicast services. *Selected Areas in Communications, IEEE Journal on* 2015; **33**(2): 141–155.
22. Militano L, Condoluci M, Araniti G, Molinaro A, Iera A, Fitzek FH. Wi-Fi cooperation or D2D-based multicast

content distribution in LTE-A: a comparative analysis. In *Communications Workshops (ICC), 2014 IEEE International Conference on*, Sydney, Australia, 2014, IEEE; 296–301.

23. Militano L, Condoluci M, Araniti G, Molinaro A, Iera A. When D2D communication improves group oriented services in beyond 4G networks. *Wireless Networks* 2014; **21**(4): 1363–1377.
24. Condoluci M, Militano L, Araniti G, Molinaro A, Iera A. Multicasting in LTE-A networks enhanced by device-to-device communications. In *Globecom Workshops (GC Wkshps), 2013 IEEE*, Atlanta, Georgia, USA, 2013, IEEE; 567–572.
25. Militano L, Condoluci M, Araniti G, Molinaro A, Iera A, Muntean M. Single frequency-based device-to-device-enhanced video delivery for evolved multimedia broadcast and multicast services. *IEEE Transactions on Broadcasting* 2015; **61**(2): 263–278.
26. Fragouli C, Le Boudec JY, Widmer J. Network coding: an instant primer. *ACM SIGCOMM Computer Communication Review* 2006; **36**(1): 63–68.
27. Lucani DE, Médard M, Stojanovic M. Random linear network coding for time division duplexing: field size considerations. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, Honolulu, Hawaii, USA, 2009, IEEE; 1–6.
28. Lauridsen M, Noël L, Sørensen TB, Mogensen P. An empirical LTE smartphone power model with a view to energy efficiency evolution. *Intel Technology Journal* 2014; **18**(1): 172–193.
29. 3GPP. Requirements for further advancements for e-utra (lte-advanced). *TR 36.913. Release 8*, 3rd generation partnership project; technical specification group: radio access plenary (RAP), 2009. http://www.3gpp.org/ftp/Specs/html-info/36913.htm [available on 15 December 2015].
30. Paramanathan A, Pahlevani P, Thorsteinsson S, Hundebøll M, Lucani D, Fitzek F. Sharing the pi: testbed description and performance evaluation of network coding on the raspberry pi. In *2014 IEEE 79th Vehicular Technology Conference*, Seoul, South Korea, 2014; 1–5.

Paper C

Getting Kodo: Network Coding for the ns-3 Simulator

Néstor J. Hernández Marcano, Morten V. Pedersen, Péter Vingelmann, Janus Heide, Daniel E. Lucani, Frank H.P. Fitzek.

The paper has been published in the
2016 Proceedings of the ACM Workshop on ns-3 (WNS3), pp. 101–107, 2016.

The layout has been revised.

Getting Kodo: Network Coding for the ns-3 Simulator

Néstor J. Hernández M.
Steinwurf ApS, Aalborg University
Aalborg, Denmark
nestor@steinwurf.com

Morten V. Pedersen
Steinwurf ApS
Aalborg, Denmark
morten@steinwurf.com

Péter Vingelmann
Steinwurf ApS
Dunaújváros, Hungary
peter@steinwurf.com

Janus Heide
Steinwurf ApS
Aalborg, Denmark
janus@steinwurf.com

Daniel E. Lucani
Aalborg University
Aalborg, Denmark
del@es.aau.dk

Frank H. P. Fitzek
Techn. Universität Dresden
Dresden, Germany
frank.fitzek@tu-dresden.de

ABSTRACT

Network Coding (NC) has been shown to improve current and upcoming communication systems in terms of throughput, energy consumption and delay reduction. However, today's evaluations on network coding solutions rely on home-grown simulators that might not accurately model realistic systems. In this work, we present for the first time the steps to use Kodo, a C++11 network coding library into the ns-3 simulator and show its potential with basic examples. Our purpose is to allow ns-3 users to use a flexible and reliable set of network coding functionalities together with the technologies simulated in ns-3. Therefore, in this paper we (i) show how to set up the Kodo library with ns-3, (ii) present the underlying design of the library examples, and (iii) verify the performance of key examples with known theoretical results.

CCS Concepts

•**Networks** → **Network simulations;** *Packet-switching networks;* •**Mathematics of computing** → **Coding theory;** •**Computing methodologies** → **Simulation tools;** •**Software and its engineering** → **Software libraries and repositories;**

Keywords

Network Coding, C++, ns-3, simulator

1. INTRODUCTION

Since its inception, network coding [14] has been a disruptive technology that allows intermediate network nodes to combine packets, instead of just routing them, resulting in increased throughput, reliability, and lower delay. NC implementations have also corroborated these promised gains under specific scenarios [15, 23, 24, 27, 29, 30].

In most previous implementations, the Kodo C++11 network coding library [31] was used. Kodo is intended to make network coding implementations available to both researchers and commercial entities, in particular those developing protocols. Kodo provides fast implementations of finite field arithmetics and the encoding, decoding and recoding fnctionalities for a variety of network codes, including Random Linear Network Coding (RLNC) [22], Perpetual [21] and Fulcrum network codes [28]. The library is continuously tested to support a large number of operating systems, compilers and architectures with hardware acceleration (SIMD) [6]. Hence, Kodo has been designed to ensure performance, testability and flexibility.

An important part of the evaluation process for these protocols is the simulation stage that aids developers to verify analytical results, rethink the modeling process by including unobserved system effects or proceed with a given design. Through the research community, the ns-3 project [8] aims to develop and establish an open network simulation environment for research. Among the project's goals are: simulation of standard technologies, simple usage and debugging, code testing and documentation that caters to the needs of the simulation workflow. Although there has been various initiatives to develop simulations tools in the network coding environment, [1, 9, 12, 18], most of these simulators: (i) may not be continuously maintained and tested, (ii) may rely on former functionalities of its components and (iii) are hard to integrate with standard technologies. Thus, to date there are no accurate network coding libraries that are well-tested and maintained to interact with deployable network simulation environments. Hence, in this work we provide for the first time, a set of examples compliant with ns-3 using Kodo as an external library for network coding where we verify know and expected results from the NC literature.

Our work is organized in the following way: Section 2 provides the theoretical aspects regarding the encoding, decoding and recoding of RLNC packets indicating some application scenarios. Section 3 shows how to get the Kodo library for ns-3 in an easy and rapid fashion. Section 4 describes the design and implementation details of our examples. Section 5 provides known verifiable results in the NC literature using several ns-3 simulations to validate the examples. Final conclusions of our work are drawn in Section 6.

WNS3, June 15-16, 2016, Seattle, WA, USA
© 2016 ACM. ISBN 978-1-4503-4216-2/16/06...$15.00
DOI: http://dx.doi.org/10.1145/2915371.2915389

2. NETWORK CODING BASICS

Kodo implements core functionalities of intra-session NC (i.e., where data packets from a single flow are combined with each other). In this type of network coding, the original data $P_j, j \in [1, g]$, each of B bits, is used to create coded packets. In the following subsections, we describe the basic functionalities of RLNC [22], namely encoding, decoding and recoding. Later, we mention applications that could potentially benefit from including RLNC as a coding scheme. More complex code variants available in Kodo are described in more detail in [3].

2.1 Encoding

With RLNC, each coded packet is a random linear combination of the original set of packets. Hence, a linearly independent (l.i.) set of g coded packets, $C_i,\ i \in [1, g]$ is required in order to get the original information. Each original packet is considered as a concatenation of elements from a Galois Field (GF) of a given size q, which we denote $GF(q)$. To create a coded packet, a coding coefficient $v_{i,j}$, is chosen at random from $GF(q)$ for every packet P_j and multiplied and added following the respective GF arithmetics. In this way, a coded packet is:

$$C_i = \bigoplus_{j=1}^{g} v_{i,j} \otimes P_j,\ \forall i \in [1, g] \tag{1}$$

To indicate which packets were used to generate a coded packet, one form is to append its coding coefficients. In this case, the overhead included for $C_i,\ \forall i \in [1, g]$ by the coding coefficients is given by:

$$|v_i| = \sum_{j=1}^{g} |v_{i,j}| = g \times \lceil \log_2(q) \rceil \text{ [bits]} \tag{2}$$

2.2 Decoding

To perform decoding, we define $\mathbf{C} = [C_1 \dots C_g]^T$ and $\mathbf{P} = [P_1 \dots P_g]^T$. Then, decoding reduces to solve the linear system $\mathbf{C} = \mathbf{V} \cdot \mathbf{P}$ using Gaussian elimination [19]. Here, the coding matrix $\mathbf{V}$ contains any set of g linearly independent packets C_i as rows as follows:

$$\mathbf{V} = \begin{bmatrix} \underline{\ v_1\ } \\ \vdots \\ \overline{\ v_g\ } \end{bmatrix} = \begin{bmatrix} v_{1,1} & \dots & v_{1,g} \\ \vdots & \ddots & \vdots \\ v_{g,1} & \dots & v_{g,g} \end{bmatrix} \tag{3}$$

The decoder begins to compute and remove the contributions from each of the pivot elements, e.g. leftmost elements in the main diagonal of (3), to reduce $\mathbf{V}$ to reduced echelon form. In this way, it is possible to recover the original set of packets.

2.3 Recoding

Network coding allows intermediate nodes in a network to recombine (or recode) packets from their sources whether they are coded or not. In general, a recoded packet should be indistinguishable from a coded one. Thus, we define a recoded packet as R_i and its corresponding encoding vector as w_i with coding coefficients $[w_{i,1} \dots w_{i,g}]$, as follows:

$$R_i = \bigoplus_{j=1}^{g} w_{i,j} \otimes C_j,\ \forall i \in [1, g] \tag{4}$$

$$\tag{5}$$

In (4), $w_{i,j}$ is the coding coefficient that multiplies C_j, uniformly and randomly chosen from $GF(q)$. Any decoder that collects $R_i, i \in [1, g]$ linearly independent coded packets, with their respective w_i, will be able to decode the data as mentioned before.

2.4 Network Coding Applications

There are numerous situations where NC provides benefits over conventional routing schemes. Basic gain descriptions and practical use cases for network coding can be found in [19, Sections 3,4] covering various areas. Among different benefits for communications, network coding can achieve the capacity for networks with multicast flows [26], improve content distribution in peer-to-peer networks [20] or enhance throughput in conventional Transmission Control Protocol (TCP) protocols for reliable communication [25]. For distributed storage systems, network coding has found applications in scenarios where it could incur less redundancy for data protection than simple replication [16].

3. GETTING KODO FOR NS-3

In this section, we explain how to get the Kodo up and running. The procedure helps to quickly add new coding functionalities in ns-3. The project with the examples is available in [4] under a GPLv2 license and it tracks the latest stable revision of the ns-3 development repository, `ns-3-dev`, to get the most recent changes. For research purposes, Kodo uses a free research license detailed in [10].

A more detailed setup guide can also be found at [4]. A descriptive tutorial for the project is available at [5]. We strongly encourage any developer to follow the setup guide. As a reference for this guide, we assume that the ns-3 project is in the `~/ns-3-dev` folder on the developer's system.

1. To get access to Kodo, it is necessary to submit a request at [11] for a research license.
2. Build the local ns-3 repository with its examples since the Kodo examples need the ns-3 binaries in order to build itself. Execute in the local `ns-3-dev` folder:
   (a) `python waf configure --enable_examples`
   (b) `python waf build`
3. Go to ~ and clone the `kodo-ns3-examples` git repository. At this point, a confirmed license is necessary to get the Kodo dependencies.
4. Go to the new `kodo-ns3-examples` folder and configure with `python waf configure` (Kodo also uses the `waf` [13] build system) to set and compile the project and its dependencies.
5. Build the `kodo-ns3-examples` and install all the needed files for ns-3 in the `~/ns-3-dev/examples/kodo` folder with `python waf build install --ns3_path="PATH"`. In this case, `"PATH"` would be `~/ns-3-dev`.

6. Get back to ns-3 folder and build the local ns-3 project with `python waf build`. At this point, the examples should be available to run as any ns-3 simulation.

4. KODO EXAMPLES FOR NS-3

In this section we describe our design implementation and criteria for creating the examples, an overview of what do the examples simulate and the design of two helpers that provide the coding operations for the system represented in the examples. The helpers function is to serve as an interface between ns-3 and the Kodo C++ bindings [2]. These are high level wrappers for the core functionalities of Kodo.

4.1 Examples Implementation

To create our examples, we consider an approach where we perform intra-session network coding, between the application and transport layer of the User Datagram Protocol (UDP) / Internet Protocol (IP) model as shown in Fig. 1. Although other approaches apply coding between the transport and Medium Access Control (MAC) layer [15,24,27,32], we implement it below the application layer to not alter other layers within the protocol stack and keep the implementation simple.

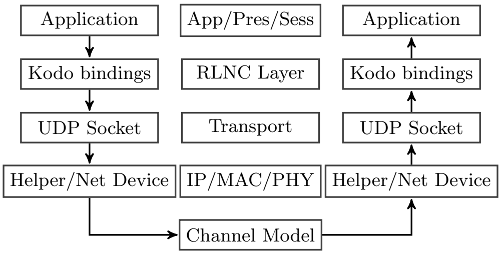


Figure 1: ns-3 + Kodo Implementation Protocol Stack based in a simple UDP/IP model.

In Fig. 1, we consider an application that generates a batch of g packets of some content. For practical reasons, we consider Hyper Text Transfer Protocol (HTTP) traffic and represent it by sending RLNC coded packets through port 80 of a UDP socket from the `UdpSocketFactory`. To encode, recode (if necessary) and decode NC packets, we employ the Application Programming Interface (API) provided by the bindings. We employ UDP datagrams because we consider best effort traffic. For the IP layer, we employ IPv4. For address assignment and routing tables, we use the `InternetStackHelper`, the `Ipv4AddressHelper` and the `Ipv4GlobalRoutingHelper` from ns-3. The details of the MAC, Physical Layer (PHY) and channel models depend on the considered example as we will see.

4.2 Examples Description

With a defined protocol stack, we describe the networks implemented in the examples to evaluate NC performance providing the details for the layers not described previously.

4.2.1 kodo-wifi-broadcast

This example, shown in Fig. 2, simulates a source broadcasting a generation of RLNC packets with generation size g and field size q to N sinks with an IEEE 802.11b WiFi ad-hoc channel. For the MAC we regard it without Quality of Service (QoS) implemented through `NqosWifiMacHelper`. We pick a WiFi MAC without QoS since in principle we are simulating connectionless best-effort traffic. Thus, the ns-3 net devices are constructed through the `WifiHelper`. Also, we turn off unnecessary MAC parameters, namely: frame fragmentation for frames larger than 2200 bytes and RTS / CTS frame collision protocol for the less than 2200 bytes. Although not required within the example, these parameters need to be included in order for the WiFi MAC to work.

For the PHY of this example, we use the `YansWifiPhyHelper`. The considered PHY includes a channel model that accounts for channel delay, path loss and receiver signal strength in dBm. We employ the `FixedRssLossModel` where the receiver signal (`rss`) is set to a fixed value. We set the broadcast data rate to be the same as unicast for the given `phyMode`. As a transmission policy, the sender keeps transmitting coded packets until all the receivers have g l.i. coded packets, even if some receivers are able to decode the whole generation.

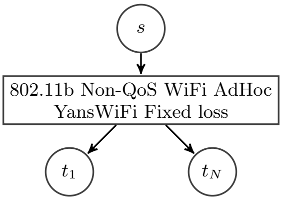


Figure 2: kodo-wifi-broadcast example network.

4.2.2 kodo-wired-broadcast

The example shown in Fig. 3, is similar as in Fig. 2 but instead, we evaluate a basic time-slotted wired system where a node either transmits or receives a single packet in a given time slot with the aid of the `PointToPointHelper`. To model a network with erasures, we consider the `RateErrorModel` for the PHY and channel model. In this case, packets sent from the transmitter could be lost or useless before arriving at the receiver. To control the amount of losses, an `ErrorRate` attribute is included at the `ReceiveErrorModel` attribute of the `RateErrorModel` to indicate the frequency of erasures within a given channel. The resulting topology is a basic representation for packet erasure networks which is akin for network coding applications. The transmission policy is the same as before. For simplicity, all devices are assumed to have the packet erasure rate, $0 \leq \epsilon < 1$. The erasure rate can be introduced as a command-line argument to set the `ErrorRate` attribute from the wired topology as we will see.

4.2.3 kodo-recoders

This example shows the gain of RLNC with recoding in a 2-hop line wired network consisting of a source, N recoders and a sink with different erasure rates. All the links between the sender and the recoders have the same packet erasure rate, $0 \leq \epsilon_{S \to R} < 1$. Equivalently, the packet erasure rate for the links between the recoders and the receivers is the

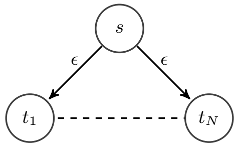


Figure 3: kodo-wired-broadcast example network.

same, $0 \leq \epsilon_{R \to D} < 1$. Again, both recoding and the erasure rates can be modified by command-line parsing. The transmission policy for this case, is as follows: First, packets are sent to each of the recoders. The transmitter stops if the decoder or all the recoders are full rank, e.g. have g l.i. coded packets. Second, a recoder retransmits packets in another scheduled time slot if l.i. packets to transmit and it stops only if the decoder is full rank.

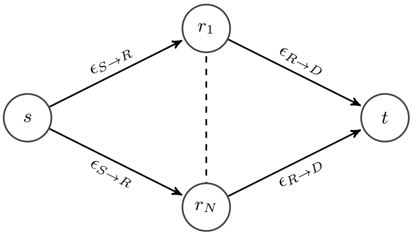


Figure 4: kodo-recoders example network.

4.3 Simulation Workflow and Helpers

In Fig. 5, we show the workflow of the example's simulation source program. This workflow is standard for ns-3 simulations and consists in defining the network (nodes, net-devices) with ns-3 helpers according the required layer functionality described in Section 4.1. Once the socket connections are defined, we call the topology helper which provides the application and coding layers. A receive callback is set to trigger an action whenever a packet is received in a decoder socket. When an encoding or decoding action has been performed, a new event is scheduled through the `ns3::Simulator` class. Events are scheduled until a generation originated in the source is decoded by the sink(s) in the evaluated example.

At the core of each example implementation resides a topology helper which contains all the encoding, recoding and decoding parameters and functionalities of the RLNC layer, the transmission policy and eases the socket connections made in each source file. The helpers are classes that serve as interfaces between the bindings and ns-3. To accomplish this, the helpers are included in ns-3, but its basic elements are objects from the Kodo C++ bindings. For our case, we use two helpers. For `kodo-wifi-broadcast` and `kodo-wired-broadcast`, we use the `Broadcast` topology helper . For the `kodo-recoders` example, we utilize the `Recoders` topology helper. In this section we present the API of these helpers in order to show the interface between Kodo and ns-3. To do so, we elaborate an Unified Modeling Language (UML) class diagram to visualize the relationships between our bindings and the helpers. We make this review only for the `Broadcast` topology helper, since the analysis for the `Recoders` topology would be similar.

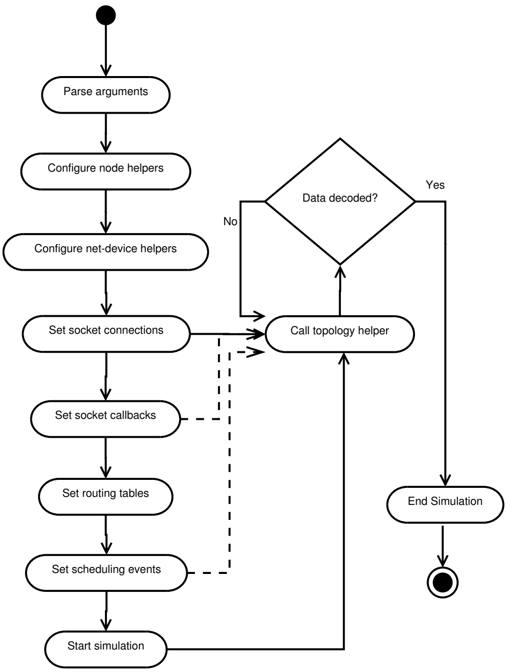


Figure 5: Examples Simulations workflow.

Fig. 6 shows the UML class diagram for the `Broadcast` topology. We have indicated the most important classes that have a type of dependency with the bindings. Also, we employ the UML package notation to indicate the namespace where all the bindings reside. We describe the topology members where the links with `kodocpp` occur. Later, we give an overview of other members whose type are natively contained in the C++ standard library or ns-3. We list the members with dependency on `kodocpp` according to their functionality.

4.3.1 Code Parameter Members

First, `m_codeType` stands for the type of erasure correcting codes utilized. In our implementation, an instance of `kodocpp::codec` is passed to the source program. The available codecs in the bindings are: `full_vector`, `on_the_fly`, `sliding_window`, `sparse_full_vector`, `seed`, `sparse_seed`, `perpetual`, `fulcrum` and `reed_solomon`. A complete description of each codec can be found in the overview section of the Kodo documentation [3]. Second, `m_field` indicates the finite field of the coding scheme. An instance of `kodocpp::field` is passed to the source program. For the available fields: `binary`, `binary4` and `binary8` represent $GF(2)$, $GF(2^4)$ and $GF(2^8)$ respectively.

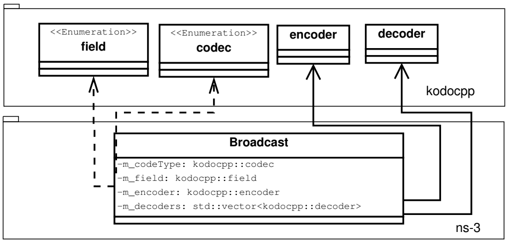


Figure 6: UML class diagram for the Broadcast topology helper interface.

4.3.2 Encoder / Decoder Object Members

The encoder data type is `kodocpp::encoder`, which is provided by the bindings. However, the encoder is not aware of the topology on its own, thus the uni-directional association link to indicate this in Fig. 6. The encoder class is a child class of the more general `kodocpp::coder` abstract class. In this way, the encoder class contains both own and general functionalities, inherited from `kodocpp::coder`, to configure its basic parameters and generate coded data. Similarly, we employ `std::vector<kodocpp::decoder>` to get a local decoder instance for each sink socket. As before, it contains functionalities to configure itself, read coded data and signal when to stop transmissions.

4.3.3 Sockets and Transmission-State Members

For packet transmissions and receptions, we use the native `ns3::Ptr<ns3::Socket>` class. Only the policies for packet transmissions/receptions are implemented through the methods `SendPacket` and `ReceivePacket`. Both of them receive the intended socket for tranmission or reception. In case of the transmitter, the packet interval time (`ns3::Time pktTime`) is also given because this will indicate the transmitter the scheduling time for next transmissions. Finally, other members like the number of users to serve, generation, packet sizes and storage buffers are considered too as standard types.

5. SIMULATIONS

To verify the accuracy of the results provided by the examples, we execute a set of ns-3 simulations to observe the behavior of RLNC in well-known scenarios. For the simulations, we compute the distribution of the number of transmissions required to decode a set of g packets with RLNC.

We only consider this metric since, typically, the time cost for the encoding and decoding operations is much lower when compared to the time spent in conveying the information from a transmitter to a receiver. Still, information regarding encoding and decoding speeds for RLNC can be easily obtained by running the benchmarks in [7] for a given platform. Similar benchmarks exist for other codes as well in their respective repositories. In our scenarios, we consider that an ideal feedback scheme is employed, where the source is aware when any destination has acknowledged all its required coded packets. To get this information, we simply call the bindings API required functions in the topology helpers.

We obtain the distribution in two scenarios. First, we consider the case of one transmitter-receiver pair. Second, we review the scenario of single-hop broadcast for N receivers. We examine these scenarios under two conditions, without packet erasures and with packet erasures. Hence, for the broadcast case, we regard the packet erasure distribution of receiver $j \in [1, N]$ as $Bernoulli(1 - \epsilon_j)$ where ϵ_j is the packet erasure probability. For evaluation purposes, we compute the distribution under a homogeneous packet erasure for all the receivers, $\epsilon_j = \epsilon \; \forall j$.

To accomplish this, we run the `kodo-wired-broadcast` example and get the number of transmissions required to decode the data in 10^4 runs. To get independent runs, the pseudo-random number generator is set to use the default seed and the `RngRun` parameter is changed in the `RngSeedManager` class by command line parsing.

For the single transmitter and receiver, we use the following parameters: `users` $= 1$, `generationSize` $= 30$, `errorRate` $= 0, 0.1$, and `field` $=$ `binary`, `binary8`. For the broadcast case, we evaluate with `users` $= 10$, `generationSize` $= 50$, `errorRate` $= 0.1, 0.2$, and `field` $=$ `binary`, `binary8`. To verify our simulations, we compare the practical results with analytical ones. To do so, we compute the Probability Mass Function (pmf) as [33, Eqs. 11-12] for the single receiver and [17, Eq. 3, Sec. III-B] for the broadcast case. Then, we plot the pmf of the analytical distributions against the simulation results.

5.1 RLNC Probability Mass Function

Fig. 7 shows the result fors the pmf of the single receiver for the evaluated parameters. We present the results for $g = 30$, $\epsilon = [0, \; 0.1]$ with $GF(2)$ and $GF(2^8)$ to observe the effect of linear independence in packets transmissions. We also evaluate the consequences of packet erasures in the number of transmissions required for decoding. In all the results, it can be clearly seen that the analytical calculations matches the simulations obtained from ns-3. For the case of no erasures, employing RLNC with $GF(2)$ requires more transmissions compared with $GF(2^8)$ since the possibilities for selecting the coding coefficients are much reduced for the last packets. For the erasure case, the transmissions alos increase given that packets might be lost regardless of linear dependency, but still are less that when employing a higher field size.

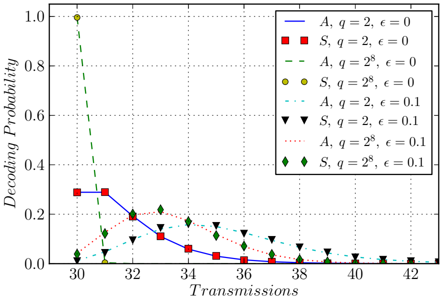


Figure 7: Analytical (A) vs. Simulation (S) for Unicast with 1 receiver and 30 packets.

5.2 RLNC Broadcast Probability Mass Function

Fig. 8 shows the result for the pmf of broadcast with RLNC for the case of 10 receivers and the evaluated parameters. In this scenario, $g = 50$. The selected fields are $GF(2)$ and $GF(2^8)$. We present the results for two erasure rates in all the links, $\epsilon = [0.1, 0.2]$.

Again, we observe the theoretical computations fit the simulations results. A difference that can be noticed with the single receiver case is the number of transmissions required to decode increases much more. Excluding the field and the erasure effects, the difference arises from all the receivers being required to get g l.i. coded packets in order to be able to decode. This is the main reason why the pmfs do not start to show a significant non-zero probability of decoding at g transmissions and shortly afterwards.

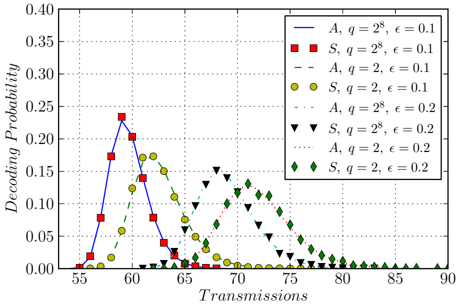


Figure 8: Analytical (A) vs. Simulation (S) for Broadcast with 10 receivers and 50 packets.

6. CONCLUSIONS

Given the increasing amount of NC applications from both academia and industry, we introduced a framework for using the Kodo library with ns-3. We hope that our contribution helps to cover the need for NC simulation capabilities in ns-3. With a set of examples where NC provides known gains, we show that our library complies with the expected results. Although the examples are made for particular topologies, the deployment of different topologies or scenarios could be easily extended by the user as detailed in [4]. Future work will be to simulate RLNC with other technologies, such as Long Term Evolution Advanced (LTE-A) within ns-3.

7. ACKNOWLEDGMENTS

This research has been financed by the CROSSFIRE MITN Marie Curie project (317126) from the European Comission FP7 framework, the Green Mobile Cloud project (Grant No. DFF - 0602 - 01372B) and the TuneSCode project (Grant No. DFF - 1335-00125) both granted by the Danish Council for Independent Research.

8. REFERENCES

[1] Inter-session network coding simulator for matlab. http://www.mathworks.com/matlabcentral/fileexchange/53750-network-coding-simulator.

[2] Kodo c++ bindings git repository. https://github.com/steinwurf/kodo-cpp.

[3] Kodo documentation read-the-docs codecs overview. http://kodo-docs.steinwurf.com/en/latest/overview.html.

[4] Kodo examples for the ns-3 simulator git repository. https://github.com/steinwurf/kodo-ns3-examples.

[5] Kodo-ns3-examples documentation read-the-docs tutorial. http://kodo-ns3-examples.readthedocs.org/en/latest.

[6] Kodo platform support. http://steinwurf.com/kodo-specifications.

[7] Kodo-rlnc git repository. https://github.com/steinwurf/kodo-rlnc.

[8] ns-3 website. https://www.nsnam.org.

[9] Software related to network coding. http://www.ifp.illinois.edu/~koetter/NWC/Software.html.

[10] Steinwurf research license. http://steinwurf.com/research-license.

[11] Steinwurf research license webpage. http://steinwurf.com/license.

[12] Universidad de cantabria network coding implementation on ns-3.13. https://github.com/dgomezunican/network-coding-ns3.

[13] Waf. the metabuild system webpage. https://waf.io.

[14] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, 2000.

[15] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. *SIGCOMM Comput. Commun. Rev.*, 37(4):169–180, 2007.

[16] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Trans. Inf. Theor.*, 56(9):4539–4551, 2010.

[17] A. Eryilmaz, A. Ozdaglar, M. Médard, and E. Ahmed. On the delay and throughput gains of coding in unreliable networks. *Information Theory, IEEE Transactions on*, 54(12):5511–5524, 2008.

[18] D. Ferreira, L. Lima, and J. Barros. Neco: Network coding simulator. ICST, 5 2010.

[19] C. Fragouli, J.-Y. Le Boudec, and J. Widmer. Network coding: an instant primer. *ACM SIGCOMM Computer Communication Review*, 36(1):63–68, 2006.

[20] C. Gkantsidis and P. Rodriguez. Network coding for large scale content distribution. In *IEEE INFOCOM*, number MSR-TR-2004-80, page 12, 2005.

[21] J. Heide, M. V. Pedersen, F. H. P. Fitzek, and M. Médard. Perpetual codes for network coding. *CoRR*, abs/1509.04492, 2015.

[22] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *Information Theory, IEEE Transactions on*, 52(10):4413–4430,

2006.

[23] M. Hundebøll, J. Leddet-Pedersen, J. Heide, M. Pedersen, S. Rein, and F. Fitzek. *CATWOMAN: Implementation and Performance Evaluation of IEEE 802.11 based Multi-Hop Networks using Network Coding*, pages 1–5. IEEE Press, 9 2012.

[24] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air: Practical wireless network coding. *IEEE/ACM Trans. Netw.*, 16(3):497–510, 2008.

[25] M. Kim, T. Klein, E. Soljanin, J. Barros, and M. Médard. Modeling network coded tcp: Analysis of throughput and energy cost. *Mobile Networks and Applications*, 19(6):790 – 803, December 2014.

[26] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Trans. Netw.*, 11(5):782–795, 2003.

[27] J. Krigslund, J. Hansen, M. Hundebøll, D. Lucani, and F. Fitzek. *CORE: COPE with MORE in Wireless Meshed Networks*, pages 1–6. IEEE, United States, 2013.

[28] D. E. Lucani, M. V. Pedersen, J. Heide, and F. H. P. Fitzek. Fulcrum network codes: A code for fluid allocation of complexity. *CoRR*, abs/1404.6620, 2014.

[29] A. Paramanathan, P. Pahlevani, S. Thorsteinsson, M. Hundebøll, D. Lucani, and F. Fitzek. Sharing the pi: Testbed description and performance evaluation of network coding on the raspberry pi. In *2014 IEEE 79th Vehicular Technology Conference*, 2014.

[30] A. Paramanathan, M. Pedersen, D. Lucani, F. Fitzek, and M. Katz. Lean and mean: Network coding for commercial devices. *IEEE Wireless Communications Magazine*, 20(5):54 – 61, 2013.

[31] M. Pedersen, J. Heide, and F. Fitzek. Kodo: An open and research oriented network coding library. In *Networking 2011 Workshops*, volume 6827 of *Lecture Notes in Computer Science*, pages 145–152. Valencia, Spain, 2011.

[32] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan. I2nc: Intra- and inter-session network coding for unicast flows in wireless networks. In *INFOCOM*, pages 1035–1043. IEEE, 2011.

[33] O. Trullols-Cruces, J. M. Barcelo-Ordinas, and M. Fiore. Exact decoding probability under random linear network coding. *Communications Letters, IEEE*, 15(1):67–69, 2011.

Paper D

On Transmission Policies in Multihop Device-to-Device Communications with Network Coded Cooperation

Néstor J. Hernández Marcano, Janus Heide, Daniel E. Lucani,
Frank H.P. Fitzek.

The paper has been published in the
2016 Proceedings of the IEEE 22th European Wireless Conference (EW 2016),
pp. 350–354, 2016.

On Transmission Policies in Multihop Device-to-Device Communications with Network Coded Cooperation

Néstor J. Hernández Marcano[*†], Janus Heide[*], Daniel E. Lucani[†] and Frank H.P. Fitzek[‡]
[*]Steinwurf ApS, Aalborg, Denmark. Mail: {nestor|janus}@steinwurf.com
[†]Department of Electronic Systems. Aalborg University, Denmark. Mail: {nh|del}@es.aau.dk
[‡]Deutsche Telekom Chair of Communication Networks.
Technische Universität Dresden, Germany. Mail: frank.fitzek@tu-dresden.de

***Abstract*—Due to the expected amount of interconnected devices in the near future, a frequent communication setting will be the case where the end user is connected to the network through short range communication protocols to other mobile users. Therefore, there is an interest in introducing new mechanisms that provide reliable content distribution in these scenarios. Thus, in this work, we present ideal network coded transmission policies to reduce the number of transmissions in simple multihop networks. We propose two recoding schemes with a collision avoidance mechanism to reduce the number of transmission required to convey a batch of packets from a source to a destination through several non-interconnected relays. Our findings indicate benefits that when employing relays with a recoding scheme and different ideal medium access probabilities, reductions of at least 50% in the total number of transmissions might be attained.**

Keywords*—*Cooperation, network coding, multihop, device-to-device

I. Introduction

In the following years, an exponential growth in data consumption for new services using telecommunication technologies is expected for future communication systems [1]. A standard assumption in former networks was that a single hop was sufficient to reach an end-user. However, due to this growth in the expected amount of connected devices and services, an end-user might not have connectivity directly from the source of information, but instead through other devices in the network that could aid in conveying information to it. Then, short range based mechanisms that can help to relay data in future network infrastructures have gathered significant interest from both academia and industry [2]–[4].

Thus, there is a major interest in finding decentralized schemes that extend connectivity and coverage in cellular systems while still providing high data rate and reliability to the end user. To achieve this, current alternatives are Device-to-Device (D2D) [5] or WiFi. For this purpose, multihop topologies with D2D communications might be formed to *cooperate* in conveying information to a receiver out of each of the cellular network [6], [7]. In this type of networks, different paths without inter-communication might be formed to reach the end receiver. These paths benefit from spatial diversity to forward the intended data to the final receiver since a loss in a path might be recovered from the correct reception in another. Hence, cooperative techniques result in increased reliability, coverage extension and throughput to end receivers. This potential has resulted in the inclusion of D2D communications in the 3rd Generation Partnership Project (3GPP) standardization efforts. To recover from packet erasures in the wireless medium, typically rateless codes are employed as a Forward Error Correction (FEC) technique. Nevertheless, although they provide benefits for single hop scenarios, they can not be deployed for cooperative communications without affecting their performance or decoding the data for each hop. Thus, rateless schemes seem an unsuitable coding choice against erasures in cooperative networks.

In this context, Network Coding (NC) [8], and particularly Random Linear Network Coding (RLNC) [9], provides not only an effective, faster and more efficient approach to relay data in multihop networks, but it simplifies the cooperation process since: (i) the information is not simply replicated, but distributed in a useful representation in the network and (ii) the final receiver only needs to get a number of linear combinations from any of the middle devices. This intuition has been exploited in previous works ranging from analysis to optimal policies and practical mechanisms, e.g., [10]–[14]. However, previous work has focused mostly in: topologies where all the cooperating devices inter-communicate with each other to coordinate the information, other scenarios like multicast or cooperation with fully connectivity in small clusters.

Thus, in this work, we present two simple decentralized transmission policies to reduce the total mean number of transmissions required to decode a batch of packets in a two-hop single source, single destination topology with various relays. To avoid collisions from the relays to the destination, we consider a collision avoidance mechanism at the Medium Access Control (MAC) layer that permits to allocate simultaneous transmissions from different nodes. Under this mechanism, we review the impact of a variable relay medium access probability in the number of transmissions to search for medium access probabilities that helps to minimize this metric in order to reduce the redundancy sent in this network. We present a set of ns-3 [15] simulations showing that at least a 30% reduction in the total transmissions, is possible for only enabling recoding at the relays. We also find that for our giving scenarios, an ideal medium access probability permits to reduce even more total packet transmissions. The paper is organized as follows: Section II defines the system model in this study. Section III gives a description of the transmission policies considered. Section IV shows ns-3 simulations to evaluate the policies. Final conclusions and future work are proposed in Section V.

ISBN 978-3-8007-4221-9

II. Model

We consider the problem of reliably transmitting a batch of g packets in a time-slotted system from a source S to a destination D, through $R_1, \ldots, R_N$ relays in a 2-hop network as shown in Fig. 1. Each packet has a length of B bits. We model the channel between transmitter $X \in [S, R_i]$ and receiver $Y \in [R_i, D]$ as a packet erasure channel, e.g. packets are sent from X to rY a might be erased (lost) with an erasure probability of $\epsilon_{X \to Y}$. We consider there is not any inter-relay connectivity nor between the source and the destination, thus $\epsilon_{R_i \to R_j} = 1, \ \forall i,j \in [1, N]$ and $\epsilon_{S \to D} = 1$. We consider independent heterogeneous packet erasure rates for each of the connectivity links from the source to the relays, $\epsilon_{S \to R_i}, \ i \in [1, N]$ and from the relays to the destinations, $\epsilon_{R_i \to D}, \ i \in [1, N]$. Hence, the packet reception distribution of receiver Y from transmitter X is $Bernoulli(1 - \epsilon_{X \to Y})$ and is independent from all others.

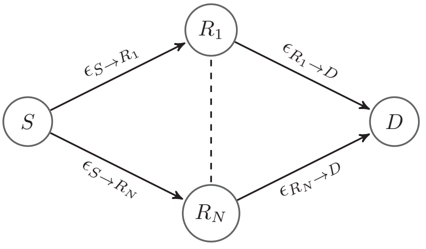


Fig. 1: 2-hop topology of a source (S), N relays ($R_1, \ldots, R_N$) and a destination (D) with a packet erasure rate in each respective link.

Tranmissions are performed through two hops. In the first hop, packets are transmitted from the source to the relays in a broadcast with RLNC fashion. The sender stops once each of the relays has g linearly independent (l.i.) coded packets. Through our study, we may refer to a l.i. coded packet as a degree of freedom. A newly received coded packet (or degree of freedom) will be called an innovative packet. Regarding the stopping condition, we may consider other stopping conditions to reduce the sender transmissions, but we consider this one since it is an upper bound for the transmissions in the first hop. Nevertheless, this condition permits to decode the data at the relays with inter-communication, if the content is also of their interest. For the second hop, the relays cooperate as a group to convey the information using either one of two possible recoding schemes, (i) recoding with RLNC or (ii) uncoded random forwarding which both will be detailed in Section III.

For any of these recoding schemes, it does not make sense for all the relays to transmit at the same time since there will be collisions at the receiver. Instead, each relay transmits with probability $p_i, \ i \in [1, N]$ only if it receives a packet from the sender in a given time slot. For simplicity, we consider $p_i = p, \ \forall i$. If two or more relays happen to transmit at the same time, we assume that a MAC layer mechanism allocates sequential non-colliding smaller time slots in a local network for the requesting relays. Therefore, in this case, we count the transmissions of the colliding relays as correctly received, regardless if the packets are innovative or not.

To take advantage of the information in the relays and not await for all to have the batch, once a packet is received from the source by the relays, they attempt to access the local medium with their respective probabilities. If several access the medium, they are scheduled by the MAC and make their transmissions. Thus, the destination benefits from receiving various packets in a single transmission. Finally, we assume that an ideal instantaneous feedback channel exists for any transmitter to know when its intended receivers are able to decode the data for stop sending packets.

III. Transmission Policies

In this Section, we give a description of the transmission policies employed to send the data. First, we provide a short description for RLNC as a coding scheme considered to broadcast the data from the source to the receivers. Later, we describe the recoding schemes employed at the relays and the MAC mechanism to avoid collisions of simultaneous transmitting relays.

A. Source to Relays: Broadcast with RLNC

In this type of network coding, the original data $P_j, j \in [1, g]$, each of B bits, is used to create coded packets. In the following subsections, we describe the basic functionalities of RLNC [9], namely encoding and recoding.

1) Encoding: With RLNC, each coded packet is a random linear combination of the original set of g packets. Each original packet is considered as a concatenation of elements from a Galois Field (GF) of a given size q, which we denote $GF(q)$. To create a coded packet, a coding coefficient $v_{i,j}$, is chosen at random from $GF(q)$ for every packet P_j and multiplied and added following the respective GF arithmetics. In this way, a coded packet is:

$$C_i = \bigoplus_{j=1}^{g} v_{i,j} \otimes P_j, \ \forall i \in [1, g] \tag{1}$$

To indicate which packets were used to generate a coded packet, one form is to append its coding coefficients. In this case, the overhead included for $C_i, \ \forall i \in [1, g]$ by the coding coefficients is given by:

$$|v_i| = \sum_{j=1}^{g} |v_{i,j}| = g \times \lceil \log_2(q) \rceil \ \text{[bits]} \tag{2}$$

2) Decoding: To perform decoding, at each relay we define $\mathbf{C} = [C_1 \ldots C_g]^T$ and $\mathbf{P} = [P_1 \ldots P_g]^T$. Then, decoding reduces to solve the linear system $\mathbf{C} = \mathbf{V} \cdot \mathbf{P}$ using Gaussian elimination [16]. Here, the coding matrix $\mathbf{V}$ contains any set of g linearly independent packets C_i as rows as follows:

$$\mathbf{V} = \begin{bmatrix} \underline{v_1} \\ \vdots \\ \overline{v_g} \end{bmatrix} = \begin{bmatrix} v_{1,1} & \ldots & v_{1,g} \\ \vdots & \ddots & \vdots \\ v_{g,1} & \ldots & v_{g,g} \end{bmatrix} \tag{3}$$

The decoder begins to compute and remove the contributions from each of the pivot elements, e.g. leftmost elements in the main diagonal of (3), to reduce $\mathbf{V}$ to reduced echelon form. In this way, it is possible to recover the original set of packets. When a packet successfully arrives at a receiver, it checks if the packet is l.i. from all its previous. If not, it discards it. In case of being l.i., the receivers adds it to its coding matrix as mentioned before. This repeats until all receivers have collected their required combinations. An Acknowledgment (ACK) is sent through the feedback channel from the last relay after it gets its final combination and the source stops sending packets.

B. Relays to Destination I: Recoding Schemes

If a packet arrives at a relay, it will proceed to send the data to the destination according to a given recoding scheme. In our study, we consider two recoding schemes which we describe subsequently.

1) RLNC Recoding Scheme: Network coding allows intermediate nodes in a network to recombine (or recode) packets obtained from their sources whether they are coded or not. Thus, we define a recoded packet as R_i and its corresponding encoding vector as w_i with coding coefficients $[w_{i,1} \dots w_{i,g}]$, as follows:

$$R_i = \bigoplus_{j=1}^{g} w_{i,j} \otimes C_j, \ \forall i \in [1, g] \tag{4}$$

In (4), $w_{i,j}$ is the coding coefficient that multiplies C_j, uniformly and randomly chosen from $GF(q)$. Notice that C_j is a packet received previously which might be coded already. However, this does not affect the original encoding since a recoded packet is again a (new) linear combination of the previous ones. Any destination that collects $R_i, i \in [1, g]$ linearly independent coded packets from all the relays, appended with their respective w_i similarly as in (2), will be able to decode the data as mentioned before. In this scheme, a relay sends recoded packet only if the rank of its coding matrix is greater than zero. Otherwise, it will always generate linearly dependent packets which may introduce overhead in the network. Still, some redudant packets might be sent given that, particularly at the beginning of the transmission process, a relay might have few coded packets to combine. However, as more l.i. coded packets are received, this redundancy tends to diminish.

2) Random Forwarding Scheme: For this case, all the packets received by a relay before acknowledging decoding are stored by it. Then, once a packet arrives at a relay, it simply forwards uniformly at random one of the currently stored packets. Although storage resource consuming, forwarding any of the previous packets nulls the possibility that any pair of relay always send two inter-dependent coded packets. Still, in this scheme, a relay is constrained to send distingueshable packets, reducing the total amount combinations that could possibly be sent. Any destination that collects g l.i. coded packets from all the relays will be able to decode the data. Same as before, a relay tries to forward a previously received packet if its rank is greater than zero.

C. Relays to Destination II: Collision Avoidance Mechanism

Once a packet is generated with any of the previous recoding schemes, a relay senses the wireless local medium and access with probability p. If two or more relaying devices coincide in a packet transmission, we assume (without loss of generality) a MAC mechanism that allocates non-colliding time slots for each coinciding relay in order to avoid collisions. The detection time of a possible collision is considered to be ideal. Thus, coinciding relay nodes do not abort the current transmission and incur in retransmissions. Hence, a single transmission is accounted for each of the coinciding nodes.

IV. Simulation Results

To analyze the performance of our proposed transmission policies, we execute a set of ns-3 [15] simulations to observe the effect of the recoding scheme, code parameters and number of relays under a given combination of packet erasure rates in the links.

We consider the number of transmissions as a metric since other metrics such as the energy or throughput, which affect performance of cellular and wireless networks, depend directly on the number of transmissions for data decoding. We evaluate this metric at the source, the relays and the total amount of transmissions required to get the content at the destination. For evaluation purposes, we make this computations under homogeneous source-relays and relays-destination packet erasure probabilities for all the relays, e.g. $\epsilon_{S \to R_i} = \epsilon_{S \to R} \ \forall i \in [1, N]$ and $\epsilon_{R_i \to D} = \epsilon_{R \to D} \ \forall i \in [1, N]$.

To accomplish this, we employ the Kodo C++11 network coding library [17] with ns-3 through a project stored in a Git repository [18] that contains a set of examples using a set of Kodo C++ bindings with ns-3. A descriptive tutorial for this project can be found in [19]. From the repository, we run the `kodo-wired-broadcast` example and get the number of transmissions required to decode the data in 10^3 runs. To get independent runs, the pseudo-random number generator is set to use the default seed and the `RngRun` parameter is set equal to the run number in the `ns3::RngSeedManager` class. With the previous data, we compute the distribution for the number of transmissions in each of the nodes and later extract the mean of it.

For the simulations, we use the following parameters: $N = [1,\ 2,\ 3]$, $g = [8,\ 16,\ 32,\ 64]$, $q = [2,\ 2^8]$, $\epsilon_{S \to R} = [0.1,\ 0.3]$, $\epsilon_{R \to D} = [0.1,\ 0.3]$, $p = [0.01,\ 0.05,\ 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 1]$ for the medium access probability and the recoding schemes considered in Section III.

A. The effect of the Number of Relays (N)

Fig. 2 shows the effects of the number of users in the total amount of transmissions from both the sender and the relays. We show it for the case of the random forwarding scheme, packet erasure probabilities $\epsilon_{S \to R} = 0.3$, $\epsilon_{R \to D} = 0.1$ and code parameters $g = 32$, $q = 2^8$. In this scheme, we observe that for more than one relay, there is a reduction in the total amount of transmissions required for decoding. Including more relays permits to have more sources of possible l.i. coded packets for the destination. However, in this case, always increasing the number of relays is not the optimal strategy because they may share various degrees of freedom. The ideal number of relays depends on the medium access probability. For a low access probability in the random forwarding scheme, the relays transmission attempts are reduced helping them collect different sets of degrees of freedom. For a high medium access probability, more transmissions at the relays of non-innovative packets tend to occur, given that they forward similar set of packets at the beginning. Hence, fewer relays are useful in this scenario. Notice that in this case, new coded packets are only introduced by the source.

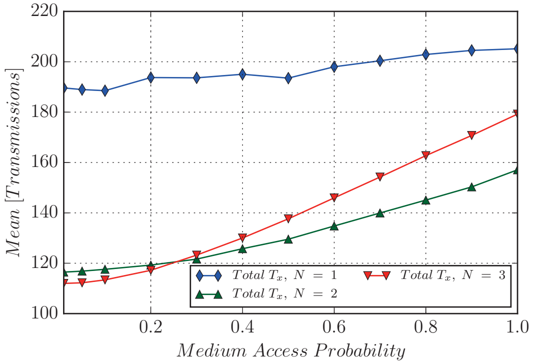


Fig. 2: Mean total number of transmissions, $Total\ T_x$, for decoding at the destination with a different amount of relays using a Random Forwarding Scheme. Scenario: $\epsilon_{S \to R} = 0.3$, $\epsilon_{R \to D} = 0.1$. Parameters: $g = 32$, $q = 2^8$

B. The effect of the Recoding Scheme and Field Size (q)

Fig. 3 shows the effects in the total number of transmissions by using a RLNC Recoding Scheme with different field sizes. We show it for the case of $g = 32$ packets, packet erasure probabilities $\epsilon_{S \to R} = 0.3$, $\epsilon_{R \to D} = 0.1$ and field sizes $q = 2, 2^8$. Allowing the relays to recode packets from their received degrees of freedom, reduces the total amount of transmissions for decoding by at least 30%, when comparing the results for $GF(2^8)$ in Fig. 2 and Fig. 3. The inherent recoding capability of RLNC makes each recoded packet indistingueshable from others, removing the restriction of receiving specific packets as in the forwarding scheme. Also, in Fig. 3, it can be observed the effect of the field size. Here, using a high field provides the advantage of requiring less transmissions than using a lower one, regardless of the number of relays. The reason is that, in the high field case, innovative packets are generated with very high probability.

We also observe there is an optimal medium access probability that minimizes the total number of transmissions for a given number of relays. From Fig. 3, a low medium access probability increments the number of transmissions required in the first hop as we consider more relays. The lower the access probability, the higher amount of time slots that a relay needs to wait for attempting a transmission and the higher amount of transmissions that the source makes since it stops transmitting once all the relays have all the degrees of freedom. For a high access probability, the relays access more frenquently the medium to transmit data, reducing the amount of transmissions from the source. However, if the access probability is too high, various redudant transmissions are made near the end.

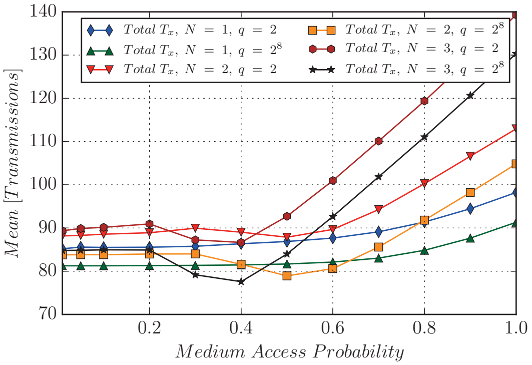


Fig. 3: Mean total number of transmissions, $Total\ T_x$, for decoding at the destination with different field sizes using a RLNC Recoding Scheme. Scenario: $\epsilon_{S \to R} = 0.3$, $\epsilon_{R \to D} = 0.1$. Parameters: $g = 32$, $q = 2, 2^8$.

C. Source and Relay Transmissions, Generation Size Effect (g)

Fig. 4 and Fig. 5 show the number of sender, relay and total transmissions employing a RLNC Recoding Scheme with two generation sizes. Fig. 4 shows the case of two relays, while Fig. 5 the case of three relays.

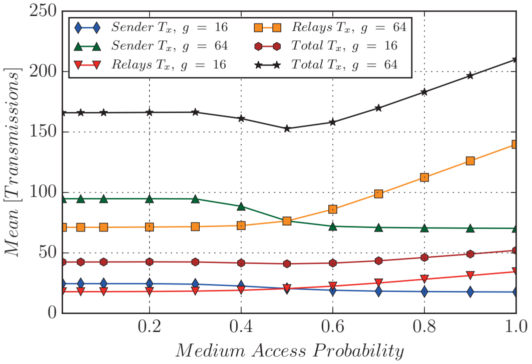


Fig. 4: Mean Sender (Source), Relays and Total number of transmissions, $Sender\ T_x$, $Relays\ T_x$, $Total\ T_x$ for 2 relays with different generation sizes using a RLNC Recoding Scheme. Scenario: $\epsilon_{S \to R} = 0.3$, $\epsilon_{R \to D} = 0.1$. Parameters: $N = 2$, $q = 2^8$

Using a higher generation size simply requires more transmissions given that more degrees of freedom are needed to be sent to the destination. It occurs independently of the number of relays to aid the source since it only varies with the generation size. By separating the sender and relays transmissions, we observe how the optimal medium access probability arises and where does it occur. As mentioned previously with Fig. 3, a low access probability increases the number of transmissions from the sender whereas a high access probability does the proper for the relays. We again observe these effects in both Fig. 4 and Fig. 5. Moreover, the total amount of transmissions is minimal particularly when the medium access probability

ISBN 978-3-8007-4221-9

approaches $p = 1/N$ approximately, e.g. a uniform medium access probability in all of the cases.

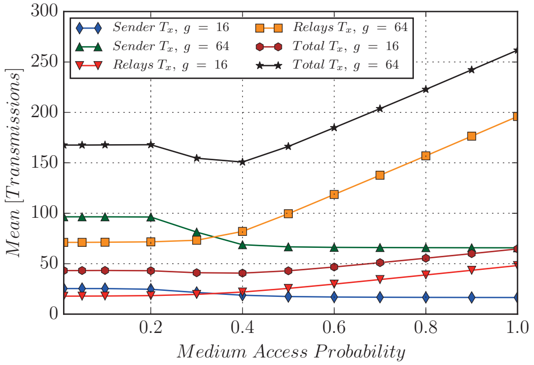


Fig. 5: Mean Sender (Source), Relays and Total number of transmissions f, $Sender\ T_x$, $Relays\ T_x$, $Total\ T_x$ for 3 relays with a different generation sizes using a RLNC Recoding Scheme. Scenario: $\epsilon_{S\rightarrow R} = 0.3,\ \epsilon_{R\rightarrow D} = 0.1$. Parameters: $N = 3,\ q = 2^8$

V. Conclusions

In this work, we propose different transmission policies to reduce the mean number of transmissions in network coded cooperative systems, since this a key metric that controls other relevant ones such as the energy consumption or the throughput. Through extensive system simulations, we could observe the benefits of a RLNC recoding scheme with a MAC collision avoidance mechanism to exploit the benefit of spatial diversity with a set of relays in multihop communications, observing a reduction in the number of transmission for various medium access transmission probabilities. Future work in this are should focus on evaluating ideal policies for minimum completing time as evaluated in [20] and [10] for similar topologies.

Acknowledgments

This research has been financed by the CROSSFIRE MITN Marie Curie project (317126) from the European Comission FP7 framework, the Green Mobile Cloud project (Grant No. DFF - 0602 - 01372B) and the TuneSCode project (Grant No. DFF - 1335-00125) both granted by the Danish Council for Independent Research.

References

[1] "Cisco visual networking index: Global traffic forecast 2015-2020, white paper," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ visual-networking-index-vni/mobile-white-paper-c11-520862.html, accessed: 2016-04-15.

[2] F. H. Fitzek and M. D. Katz, *Cooperation in Wireless Networks: Principles and Applications - Real Egoistic Behaviour is to Cooperate!* Springer, 2006.

[3] F. Fitzek and M. Katz, *Mobile Clouds: Exploiting Distributed Resources in Wireless, Mobile and Social Networks*. Wiley, 2013. [Online]. Available: https://books.google.dk/books?id=s2lXAgAAQBAJ

[4] X. Lin, J. G. Andrews, and A. Ghosh, "A comprehensive framework for device-to-device communications in cellular networks," *arXiv preprint ArXiv:1305.4219*, 2013.

[5] 3GPP, "3rd generation partnership project; technical specification group sa; feasibility study for proximity services (prose) (release 12)," *TR22.803V1.0.0*, 2012.

[6] A. Abrardo, G. Fodor, and B. Tola, "Network coding schemes for d2d communications based relaying for cellular coverage extension," *Transactions on Emerging Telecommunications Technologies*, pp. n/a–n/a, 2015. [Online]. Available: http://dx.doi.org/10.1002/ett.2994

[7] J. Heide, F. H. Fitzek, M. V. Pedersen, and M. Katz, "Green mobile clouds: Network coding and user cooperation for improved energy efficiency," in *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*. IEEE, 2012, pp. 111–118.

[8] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, 2000.

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

[10] H. Khamfroush, D. E. Lucani, and J. Barros, "Minimizing the completion time of a wireless cooperative network using network coding," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*. IEEE, 2013, pp. 2016–2020.

[11] H. Khamfroush, D. E. Lucani, P. Pahlevani, and J. Barros, "On optimal policies for network coded cooperation: Theory and implementation," *IEEE Journal on Selected Areas in Communications*, 2015.

[12] A. Paramanathan, M. V. Pedersen, D. E. Lucani, F. H. Fitzek, and M. Katz, "Lean and mean: network coding for commercial devices," *Wireless Communications, IEEE*, vol. 20, no. 5, pp. 54–61, 2013.

[13] L. Militano, A. Orsino, G. Araniti, A. Molinaro, and A. Iera, "A constrained coalition formation game for multihop d2d content uploading," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 2012–2024, 2015.

[14] G. Fodor, S. Parkvall, S. Sorrentino, P. Wallentin, Q. Lu, and N. Brahmi, "Device-to-device communications for national security and public safety," *IEEE Access*, vol. 2, pp. 1510–1520, 2014.

[15] "ns-3 website," https://www.nsnam.org, accessed: 2016-04-15.

[16] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, 2006.

[17] M. Pedersen, J. Heide, and F. Fitzek, "Kodo: An open and research oriented network coding library," in *Networking 2011 Workshops*, ser. Lecture Notes in Computer Science. Valencia, Spain, 2011, vol. 6827, pp. 145–152.

[18] "Kodo examples for the ns-3 simulator git repository," https://github.com/steinwurf/kodo-ns3-examples, accessed: 2016-04-15.

[19] "Kodo-ns3-examples documentation read-the-docs tutorial," http://kodo-ns3-examples.readthedocs.org/en/latest/, accessed: 2016-04-15.

[20] H. Khamfroush, P. Pahlevani, D. E. Lucani, M. Hundeboll, and F. H. Fitzek, "On the coded packet relay network in the presence of neighbors: Benefits of speaking in a crowded room," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1928–1933.