# Bridging inter-flow and intra-flow network coding in wireless mesh networks

*From theory to implementation*

Hansen, Jonas; Krigslund, Jeppe; Lucani, Daniel E.; Pahlevani, Peyman; Fitzek, Frank H.P.

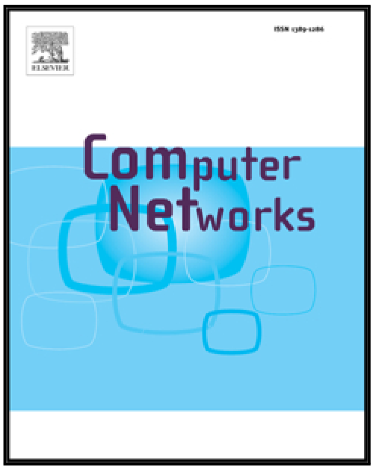
*Published in:*
Computer Networks

*DOI (link to publication from Publisher):*
10.1016/j.comnet.2018.07.014



*Publication date:*
2018

*Document Version*
Accepted author manuscript, peer reviewed version




# Bridging Inter-flow and Intra-flow Network Coding in Wireless Mesh Networks: From Theory to Implementation

Jonas Hansen, Jeppe Krigslund, Daniel E. Lucani,
Peyman Pahlevani, Frank H.P. Fitzek

PII: S1389-1286(18)30518-8
DOI: https://doi.org/10.1016/j.comnet.2018.07.014
Reference: COMPNW 6544

To appear in: *Computer Networks*

Received date: 10 May 2017
Revised date: 1 July 2018
Accepted date: 8 July 2018

Please cite this article as: Jonas Hansen, Jeppe Krigslund, Daniel E. Lucani, Peyman Pahlevani, Frank H.P. Fitzek, Bridging Inter-flow and Intra-flow Network Coding in Wireless Mesh Networks: From Theory to Implementation, *Computer Networks* (2018), doi: https://doi.org/10.1016/j.comnet.2018.07.014

# Bridging Inter-flow and Intra-flow Network Coding in Wireless Mesh Networks: From Theory to Implementation

Jonas Hansen[a], Jeppe Krigslund[a,b], Daniel E. Lucani[c], Peyman Pahlevani,[d,], Frank H. P. Fitzek[a,b]

[a] Department of Electronic Systems, Aalborg University, Denmark
[b] acticom GmbH, Germany
[c] Department of Engineering - Communications Systems, Arhus University, Denmark
[d] Department of Computer Science and Information Technology, IASBS, Zanjan, Iran

**Abstract**

This paper presents and characterizes the performance of CORE, a protocol that brings together the efficiency in spectrum usage of inter–session network coding schemes and the robustness against packet losses of intra–session network coding. We provide in–depth mathematical analysis of the gains of CORE followed by protocol design and implementation details needed for CORE's successful deployment in practice. Finally, we provide extensive measurements with off–the–shelf wireless nodes under various channel and system conditions comparing CORE to other state–of–the–art approaches, namely, forwarding (no coding) and COPE (inter–session network coding). These measurements support our theoretical findings, showing that CORE not only outperforms COPE and forwarding in general, but that order of magnitude gains are possible for cases with high packet losses. Specifically, CORE has a throughput gain of more than $10x$ over a COPE–like scheme and $7x$ over forwarding when the error ratio is 50 % on all links. Beyond these gains over other protocols, our measurements show that our CORE implementation can achieve close to optimal performance with a gap of less than 0.43 dB.

*Keywords:* Network Coding, Wireless Mesh Networks, Ad-hoc Network, Implementation on Real Devices, Performance Comparison, Real-life Measurements



*Corresponding author, Tel: +98 24 33 15 34 36
*Email addresses:* jonas09@es.aau.dk (Jonas Hansen), jepkri@es.aau.dk (Jeppe Krigslund), daniel.lucani@eng.au.dk (Daniel E. Lucani), pahlevani@iasbs.ac.ir (Peyman Pahlevani,), ff@es.aau.dk (Frank H. P. Fitzek)

## 1. Introduction

The emergence and widespread adoption of devices with wireless communication capabilities, from mobile phones and laptops to sensors, has spawned a variety of solutions for providing end–users with higher data rates and quality of service. Although more traditional networks are based on an infrastructured approach, i.e., each device is connected directly to a base station or an access point, network operators' resource limitations or lack of infrastructure in some regions require alternative solutions. Wireless Mesh Network (WMN) provides such an alternative by allowing devices to connect to other devices. Future large scale operator driven networks such as 5G communication networks indicate the inclusion of mesh topologies like device–to–device communications, which can be used for range extension or have Wi–Fi complementing the cellular-network in high dense areas. WMN is also used for sensor networks, car–to–car networks and Internet of things, but the applications are not limited to these cases. The inherently dynamic structure of WMN imposes challenges in providing proper throughput and Quality of Service (QoS). The design of current and future WMN protocols should address these issues.

Network Coding (NC) [1] has emerged as a compelling solution to improve throughput performance and reliability in wireless networks. Research in this area has focused on two key approaches, namely inter–flow and intra–flow network coding, which consider coding packets from different flows or coding only across packets of the same flow. Since the former constitutes a complex problem in general [2], a typical approach to achieve some of its benefits has been to use opportunistic coding. In essence, the protocol identifies specific *coding regions*, e.g., shaded regions in Figure 1, where data from two or more flows intersect at a relaying node. This relay then can perform a bit–by–bit XOR of packets judiciously and broadcasted to all its neighbors in order to reduce the number of transmissions. A common goal of these protocols is to guarantee that the packets coming out of the coding region do not have contributions from more than one flow, which guarantees that sources or destinations are unaware of the operations in the coding regions. This idea was originally proposed in COPE [3], but it has seen a lot of attention since then from the research community, e.g., [4, 5, 6, 7]. Although suited to reduce the number of transmissions from relays in the coding region, these approaches do not provide protection against packet losses.



ACCEPTED MANUSCRIPT

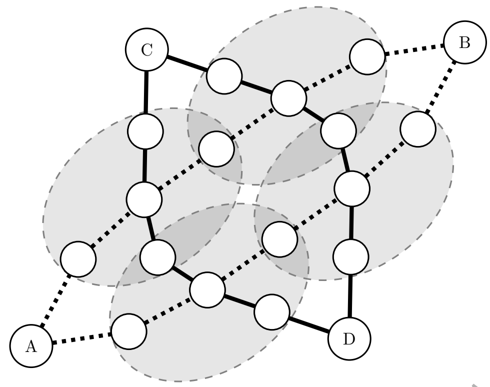



Figure 1: An example mesh network. The network contains multiple nodes, multiple data paths and two flows from $A$ to $B$ and from $C$ to $D$. Each coding region is marked with a gray ellipse, note that some nodes are included in more than one coding region.

On the other hand, intra–flow NC pursues advantages by coding packets of the same flow. One of the key goals is to provide higher resilience to packet losses. An important feature of intra–flow NC is the possibility to recode packets at intermediate nodes from packets in their buffers, thus constructing new linear combinations. This feature allows the system to compensate for losses on a link by link basis, instead of compounding losses end–to–end and, hence, attain higher throughput. Random Linear Network Coding (RLNC) provides a distributed approach to perform intra–flow NC, regardless of the traffic characteristics or network topology. The key is that the source as well as intermediate nodes generate linear combinations of packets in their respective buffers using random coefficients. Operations are performed using finite fields $\mathbf{F_q}$ of size $q$ and coding coefficients are chosen uniformly at random from the $q$ elements of the field. MORE [8] is a protocol that combines RLNC with opportunistic routing to provide higher reliability as well as a higher end–to–end throughput.

Our goal in the paper is to provide solutions that exploit the broadcast nature of the wireless channel and network coding principles to both cope with losses and provide a more efficient use of the wireless resource at intermediate nodes. To do so, we proposed CORE [9], a network coding protocol that bridges the concepts of intra–flow and inter–flow network coding. Although, [10, 11] suggested a combination of intra–flow and inter–flow NC, their proposals did not merge them, but rather used them at different protocol layers. That is, intra–flow was used for reliability between TCP and IP while COPE was used in lower layers. This means that redundancy is provided end–to–end, but the potential limitations of COPE can still limit the overall system performance.

The original paper [3], suggested reverting to forwarding if the probability of all nodes being able to decode XOR packets is less than 80 %. Our results in [9] showed that this in fact coincided with the region where both forwarding and COPE had the same performance and where COPE started performing worse than forwarding.

In this paper, and as an extension to the our work presented in [9], we present an implementation in a testbed that characterises the performance of CORE, COPE, and forwarding under similar conditions. Furthermore, a theoretical analysis based on probability theory is provided. Both analysis and testbed supports the findings in our original paper [9]. In this paper we present both main and novel features of the CORE protocol (Section 4) and testbed (Section 7) as well as a performance comparison to the different approaches, both in theory (Section 5) and in practice (Section 8). CORE is shown not only to outperform forwarding and COPE by several orders of magnitude, but also is shown that CORE can provide near optimal throughput performance even in the presence of high losses.

## 2. Related Work

The idea of inter session network coding was originally proposed in COPE and it has received a lot of attention from the research community, e.g., [7, 4, 12, 13]. For example, BEND [4] is a MAC layer solution to practical network coding in multi-hop wireless networks. It is a first exploration of the broadcasting nature of wireless channels to proactively capture more coding opportunities. The CATWOMAN protocol [7] provides an implementation of network coding on top of an existing routing scheme known as B.A.T.M.A.N. to utilize the routing to detect coding opportunities. Finally, FENC [12] utilizes division and conquer method to find a coded packet with maximum benefit for all receivers. Although COPE-like approaches are suited to reduce the number of transmissions from relays in the coding region, these approaches do not provide protection against packet losses. Authors in [14] provide analytical expressions to this problem to qualify the performance of the COPE in presence of packet losses. It also proposes a mechanism to provide the reliability for overhearing nodes based on traditional ARQ approaches that uses acknowledgment packet to signal the sender. However, using ARQ based approaches increases the overhead in the network. Losses add an additional dimension to network coding problems and, when losses are present, even single unicast connection may benefit for specific scenarios, e.g., long latency [15]. Tracy Ho et al. [8] introduced random linear network coding (RLNC) to provide robustness against random packet losses. This approach is also known as an intra-session network coding approach where the coded packets are created within the same session. Authors in MORE [8] showed that RLNC can reduce signaling overhead between different nodes by avoiding coordination between different nodes in the network. The source calculates and assigns a transmission redundancy to the relays, computed from loss rate measurements. Unlike MORE, in CCACK[16] each relay node decides how many packets to transmit in an online fashion and oblivious to link loss rates. Although these approaches increase the network performance by introducing novel network coding based routing protocols, they are not compatible with existing routing protocols already deployed in wireless networks. PlayNCool [17] advocates for the use of a network coded protocol that is independent of the system's routing protocol, thus allowing to exploit existing routing protocols such as AODV, OLSR, and DSDV [aodv, olsr, dsdv] to select the best next hop. Recent demonstrators have shown the potential of network coding to deliver low latency wireless mesh networking [18]. I2NC [10] suggests a combination of intra-flow and inter-flow that has two key benefits. The erasure-correcting capabilities of intra-session network coding make the scheme resilient to losses, while the redundancy introduced by the relays also allows intermediate nodes to operate without knowledge of the decoding buffers of their neighbors. This proposal used intra- and inter-session network coding at different protocol layers and in separation from each other. That is, intra–flow was used for reliability between TCP and IP (compensate for packet losses), while COPE was used in lower layers (spectral efficiency). This means that redundancy is provided end–to–end, but the potential limitations of COPE can still limit the overall system performance. Our proposed CORE protocol will combine both techniques in a single, coherent solution that is compatible with different routing protocols.

## 3. The Need for Bridging Intra- and Inter–Flow Coding

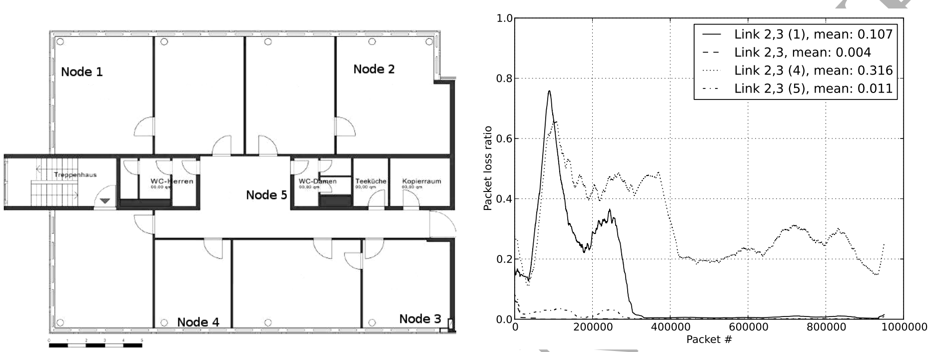
Wireless networks are inherently unreliable, both due to external interference and the increased mobility among wireless nodes. Although the effects and limitations introduced by channel fading are inevitable, an intelligent combination of channel modulation and repetition coding has been commonly used to provide a sufficiently reliable communication channel. Such mechanisms are oriented towards direct communication between two wireless nodes and have been widely used in cellular networks and in IEEE 802.11.

However, this reliable point–to–point approach has been shown to be ineffective in more complex wireless topologies/networks and recent research efforts have advocated for the use of opportunistic overhearing as a first strategy to exploit the broadcast nature of wireless communications [3, 8, 10]. Since the channel modulation and repetition coding are optimized specifically for the intended receiver, the reliability of overhearing will vary considerably and potentially limiting the impact of the proposed opportunistic approaches.

In order to elucidate this issue, we conduct a series of communication channel measurements in an IEEE 802.11 ad–hoc network with off–the–shelf devices. We investigate the packet loss probability from a sender to all receivers in three scenarios. First, when the source node uses the broadcast mode of IEEE 802.11, i.e., no specific node is targeted for reception of the packet. Second, when the source node uses the unicast mode of IEEE 802.11, which implements rate adaptation and Medium Access Control (MAC) packet retransmissions for channel reliability to the intended node. Third, a similar setting to the second case but without MAC retransmissions. In the last two, the loss probability both on the direct link, i.e., at the intended receiver, and on overhearing receivers is measured. The difference between unicast without retransmissions and broadcast resides in the channel modulation, where unicast can optimise the modulation to the intended receiver, while broadcast must fall back to a default modulation, which is usually slow but robust. Furthermore, unicast may utilise RTS/CTS mechanisms to virtually eliminate the 'hidden node problem'.

To compare the introduced three scenarios, we placed five laptops in an office environment and we set to transmit packets to each other over the course of a full day. The setup is illustrated with a floor plan of the office in Figure 2a. Overhearing was enabled by using libpcap. The average measured packet loss ratio for the transmission links is stated in Tables 2a, 2b and 1. Our measurements demonstrate not only the opportunistic overhearing is inherently challenged due to the lack of reception guarantees, but also the packet loss ratio can vary dramatically depending both on the channel qualities and adaptation mechanisms deployed for improving the direct communication link. While overhearing seems to benefit from MAC retransmissions, it does not provide any packet delivery guarantees. Despite the retransmissions, we observe packet loss above 80 % on multiple occasions.

Figure 2b shows the packet loss ratio throughout a day's measurement for the case of unicast without MAC retransmissions, illustrating that the packet loss ratio varies substantially over time even when the nodes themselves are stationary. This can be explained in part by changes in the external interference from other devices. This result further supports the fact that overhearing is extremely volatile and reinforces the need for protocols that can naturally and seamlessly exploit the benefits of overhearing for inter–flow network coding, but without relying on assumptions of good or moderate quality in the overhearing links to do so. Our goal is to propose and demonstrate one such protocol.




(a) (b)

Figure 2: Packet loss measurement setup and packet loss over time.

| Tx | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | x | 0.05 | 0.04 | 0.06 | 0.01 |
| 2 | 0.13 | x | 0.03 | 0.05 | 0.01 |
| 3 | 0.21 | 0.08 | x | 0.04 | 0.01 |
| 4 | 0.20 | 0.07 | 0.01 | x | 0.01 |
| 5 | 0.07 | 0.01 | 0.02 | 0.05 | x |

Table 1: Average packet loss ratio (floating-point notation) for broadcast transmissions between nodes in the measurement setup. Tx indicates the transmitting node, while $1-5$ indicates the receiver.

## 4. CORE

CORE combines the idea of inter–flow network coding with intra–flow network coding and merges the benefits from both approaches into one robust and efficient network protocol for meshed networks. The CORE protocol characterizes and generalizes the mechanisms described in [9]. These features together with new enhanced features are presented in Section V. The new features allow us to provide a more robust practical implementation of the original idea.

The CORE protocol uses RLNC for each unicast flow, potentially over multiple routes [9]. Route selection for each flow can be performed using standard WMN routing algorithms. CORE identifies relays where two or

| Tx | Dx | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 | 2 | x | 0.01 | 0.94 | 0.98 | 0.52 |
| 1 | 3 | x | 0.03 | 0.01 | 0.61 | 0.20 |
| 1 | 4 | x | 0.02 | 0.31 | 0.01 | 0.01 |
| 1 | 5 | x | 0.09 | 0.89 | 0.98 | 0.01 |
| 2 | 1 | 0.00 | x | 0.84 | 0.82 | 0.01 |
| 2 | 3 | 0.11 | x | 0.00 | 0.32 | 0.01 |
| 2 | 4 | 0.06 | x | 0.42 | 0.00 | 0.01 |
| 2 | 5 | 0.22 | x | 0.93 | 0.92 | 0.00 |
| 3 | 1 | 0.01 | 0.03 | x | 0.03 | 0.02 |
| 3 | 2 | 0.92 | 0.01 | x | 0.20 | 0.17 |
| 3 | 4 | 0.98 | 0.79 | x | 0.01 | 0.22 |
| 3 | 5 | 0.97 | 0.57 | x | 0.10 | 0.01 |
| 4 | 1 | 0.01 | 0.02 | 0.02 | x | 0.01 |
| 4 | 2 | 0.98 | 0.01 | 0.55 | x | 0.02 |
| 4 | 3 | 1.00 | 0.80 | 0.01 | x | 0.62 |
| 4 | 5 | 1.00 | 0.17 | 0.62 | x | 0.01 |
| 5 | 1 | 0.00 | 0.01 | 0.35 | 0.33 | x |
| 5 | 2 | 0.18 | 0.00 | 0.41 | 0.36 | x |
| 5 | 3 | 0.13 | 0.02 | 0.00 | 0.28 | x |
| 5 | 4 | 0.16 | 0.03 | 0.26 | 0.00 | x |

(a) With MAC-layer retransmissions.

| Tx | Dx | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 | 2 | x | 0.69 | 0.90 | 0.72 | 0.66 |
| 1 | 3 | x | 0.66 | 0.70 | 0.66 | 0.66 |
| 1 | 4 | x | 0.68 | 0.87 | 0.69 | 0.66 |
| 1 | 5 | x | 0.79 | 0.95 | 0.81 | 0.67 |
| 2 | 1 | 0.43 | x | 0.50 | 0.50 | 0.43 |
| 2 | 3 | 0.44 | x | 0.51 | 0.51 | 0.43 |
| 2 | 4 | 0.68 | x | 0.80 | 0.66 | 0.62 |
| 2 | 5 | 0.42 | x | 0.48 | 0.48 | 0.40 |
| 3 | 1 | 0.53 | 0.52 | x | 0.38 | 0.46 |
| 3 | 2 | 0.94 | 0.68 | x | 0.60 | 0.76 |
| 3 | 4 | 0.99 | 0.79 | x | 0.63 | 0.81 |
| 3 | 5 | 0.90 | 0.70 | x | 0.59 | 0.66 |
| 4 | 1 | 0.55 | 0.53 | 0.43 | x | 0.44 |
| 4 | 2 | 0.55 | 0.52 | 0.42 | x | 0.43 |
| 4 | 3 | 0.56 | 0.54 | 0.44 | x | 0.45 |
| 4 | 5 | 0.53 | 0.52 | 0.41 | x | 0.41 |
| 5 | 1 | 0.39 | 0.37 | 0.44 | 0.37 | x |
| 5 | 2 | 0.38 | 0.36 | 0.43 | 0.36 | x |
| 5 | 3 | 0.41 | 0.39 | 0.46 | 0.39 | x |
| 5 | 4 | 0.58 | 0.58 | 0.72 | 0.61 | x |

(b) Without MAC-layer retransmissions.

Table 2: Average packet loss ratio (floating-point notation) for unicast transmissions between nodes in the measurement setup. Tx indicates the transmitting node, Dx the intended destination, while $1-5$ indicates the actual receiver.

more unicast flows intersect forming a potential coding region. CORE performs inter–flow NC within these coding regions to provide greater spectral efficiency in the region by reducing the number of necessary transmissions from the relay. The relay XORs the payload of received RLNC packets, and remaps coding coefficients to preserve original coefficients for each flow. This essentially creates a new coded packet from a combined RLNC generation, as XORing is equivalent to coding with coefficients of 1 regardless of the Galois field $GF(2^k)$ in use. Similar to COPE, if a receiver in the coding region overhears a packet and receives an XORed packet containing the overheard packet, the receiver will be able to recover the intended data packet from its own flow simply by XORing the two payloads and extracting the corresponding coding coefficients. Once transmitted, the relay stores the coded packets of each flow to be used later. To guarantee XORing of the packets, the relay defines an appropriate holding time for outgoing packets that is reset once a packet is received from all active flows. If this is exceeded, the relay will XOR the novel packets with recoded packets from the remaining flow(s) using the RLNC packets stored in its buffer. This provides a way to compensate possible losses of the other flow while still transmitting the intended new packet. In other words, a CORE relay can implement recoding, a standard feature of RLNC and one that provides additional throughput and robustness against losses.

Receivers in CORE can then exploit the structure of RLNC to speed up packet recovery and attain higher robustness against losses, particularly due to unsuccessful packet overhearing. Receivers accomplish this not only by overhearing transmissions corresponding to other flows to be used immediately, but store them for later use. Receivers are required to do at least partial [9] decoding as CORE creates combinations of RLNC packets from multiple flows, which should be separated into individually coded flows before leaving the coding region.

CORE can implement simple signalling mechanisms. Receivers in a coding region can request more coded packets from the relay. Similarly, the relay can signal the sources to transmit additional coded packets. The relay will signal a source to stop transmissions if it has received enough coded packets to decode. Of course, the relay has no intention to decode, but signaling the upstream node to stop transmitting reduces overall transmissions and collisions in the coding region. As with any signalling mechanisms, this may have an impact on performance due to the increased medium access. This impact is counted in by our test-bed setup, but omitted in simulations and theoretical analysis. A benefit of RLNC is that the relay may send packet requests to the sources and only specify the amount of desired packets, without knowing the specific packets received. In essence, RLNC reduces the amount of signalling needed to provide consistent communication.

CORE can provide three different types of relays depending on the desired trade–off between performance and computational effort [9]. Sources and destinations are independent on the relay choices, which means that on–the–fly changes can occur in the relays without knowledge from destinations and sources. For this work, we focus on two extreme cases and explain them in the following.

• *CORE with no recoding, no feedback:* This is the lightest CORE scheme, with minimal requirements on the relay operations. The relay does not recode or implement any link–by–link signaling functionalities. The sources and destinations use RLNC, and the relay uses XOR every time a coding opportunity occurs, i.e., when the relay receives at least two RLNC packets from two different flows within a prespecified holding time. After a period without receiving any packets, the receivers will time–out and send packet requests to the senders. In the implementation and throughout this paper, the relay used in this configuration is denoted the *non–recoding relay*.

• *CORE with no feedback:* An intermediate version of the CORE scheme where recoding is enabled at the relay. This provides the relay with functionality to create redundancy from the previously received packets and include these in XOR operations if necessary. This naturally requires additional capabilities from the relay: processing power for the recoding and memory for storage of previous packets. This configuration of the relay is denoted the *recoding relay*.

• *CORE:* The full CORE scheme. The relay performs full decoding of the flows from the sources, this allows the relay to know the rank of each recoder. When the relay knows the rank of its recoders, it is able to request only the needed amount of packets from the sources. The relay will inform a source to stop sending packets, when the corresponding recoder reaches full rank. The benefit of this feedback is two fold, as the sources do not send redundant packets, thereby increasing the network efficiency, they will not access the wireless medium allowing the relay to use a higher amount of bandwidth. The relay used in this configuration is denoted the *CORE relay*.

## 5. Theoretical Analysis of Transmission Schemes

This section characterizes the theoretical efficiency of CORE in terms of transmissions per packet under different system conditions. In particular, we study the performance (i) without recoding and without link–by–link feedback, (ii) with recoding but without link–by–link feedback, and (iii) with recoding and feedback (full CORE scheme). Furthermore, we provide a similar analysis for forwarding and a COPE–like protocol under similar assumptions.

The efficiency is defined as the total amount of transmissions in the cross topology per total amount of source packets, where source $S_i$ holds $M_i$ original packets. Efficiency of the COPE–like and CORE schemes are calculated based on the perspective of one destination receiving its intended data. This allows us to calculate a simple and tight lower–bound on the efficiency of these schemes. A full characterization is possible with our proposed techniques, but requiring additional dimensions to be considered in our Markov model.

Our analysis focuses on the cross topology, where the source and destination for flow $i$ within the coding cluster are labeled $S_i$ and $D_i$, respectively, and the relay is labeled $R$. Channels are independent and packet erasures on a wireless link are assumed to be independent and identically distributed (i.i.d.). The erasure probability of the link between node $i$ and $j$ is denoted $e_{i,j}$, where $i, j \in \{S_1, S_2, R, D_1, D_2\}$. To reduce the number of variables in our current derivation, we consider the packet loss probabilities from and to the relay to be equal $e_{S_1,R} = e_{S_2,R} = e_{R,D_1} = e_{R,D_2} = e_X$, while the overhearing failure probabilities between sources and destinations, namely, $e_{S_1,D_2} = e_{S_2,D_1} = e_{OH}$, are different to $e_X$. An extension that incorporates all variables is possible, but cumbersome. We focus on the current case as it allows us to illustrate the effects of overhearing channels separately from the main channels. Finally, we assume that our system uses MAC layer retransmissions for forwarding and COPE–like, with the allowed amount of retransmissions to be infinite, i.e. $r_{max} = \infty$.

### *5.1. Forwarding*

There are two possible cases for forwarding. The first corresponds to "unconstrained forwarding", i.e., the relay is provided with sufficient resources to forward the two flows successfully. The second is called "constrained forwarding" which is a relevant case for wireless networks using a fair MAC protocol, e.g., CSMA/CA used in Wi–Fi when the system is highly loaded and the relay becomes a bottleneck for both flows as the relay is competing with both sources for the channel access.

For unconstrained forwarding, the number of transmissions needed to successfully deliver a packet from flow $i$ are the sum of the expected transmissions needed to deliver a packet on each of the hops, $S_i \to R$ and $R \to D_i$. The expected number of transmissions for an arbitrary packet loss probability $e_X$ is given by:

$$E_{TX}\left[S_i \to R\right] = \sum_{k=1}^{\infty} k \cdot {e_X}^{k-1} \cdot (1 - e_X) = (1 - e_X) \sum_{k=1}^{\infty} k \cdot {e_X}^{k-1}. \tag{1}$$

Geometric convergence is used to reduce the expression:

$$E_{TX}\left[S_i \to R\right] = \ (1 - e_X) \frac{\partial}{\partial e_X} \left( \sum_{k=0}^{\infty} {e_X}^{k} \right) = \frac{1}{1 - e_X}. \tag{2}$$

Due to the symmetry ($E_{TX}\left[S_i \to R\right] = E_{TX}\left[R \to D_i\right]$) in the packet loss probabilities for the channels involved in each flow. The efficiency of forwarding is then calculated as the sum of all the expected packets divided by the number of source packets delivered:

$$\eta_{Forwarding,\ unconstrained} = \frac{\frac{1}{1-e_X} + \frac{1}{1-e_X} + \frac{1}{1-e_X} + \frac{1}{1-e_X}}{2} = \frac{2}{1 - e_X}. \tag{3}$$

For constrained forwarding, the sources need to compensate for the packets dropped at the relay by resending those packets. The expected number of transmissions per packet at each source is then based on both the probability of the packet being lost in the transmission, $e_X$, and the probability of the packet being dropped at the relay, $P_{drop}$. Eq. (2) is then expanded to:

$$E\left[S_i \to R\right] = \left((1 - e_X) \cdot (1 - P_{drop})\right)^{-1}. \tag{4}$$

In the X topology, using a CSMA/CA algorithm and high load conditions $P_{drop} = 0.5$. By expanding Eq. (3) with Eq. (4), the efficiency for the constrained forwarding scheme is:

$$\eta_{Forwarding,\ constrained} = \left((1 - e_X) \cdot (1 - P_{drop})\right)^{-1} + (1 - e_X)^{-1}. \tag{5}$$

*5.2. COPE–like*

To calculate the efficiency of the COPE–like transmission scheme, we first derive an expression for the probability of the destinations overhear the traffic from the sources in Eq. (6). This is a sum of probabilities of overhearing, when $n$ consecutive transmissions are used between $S_i$ and $R$, weighted by the probability of that event occurring. In Eq. (6) we consider $n$ consecutive transmissions until a packet is successfully received. The probability of success after $n-1$ failures is then $e_X^{(n-1)}.(1 - e_X)$. Since the probability of overhearing failure is denoted by $e_{OH}$, the probability of successful overhearing of a packet within $n$ consecutive transmissions is $1 - (e_{OH})^n$.

$$\Pr(S_i \to D_j) = \sum_{n=1}^{\infty} (1 - e_{OH}{}^n) \cdot e_X{}^{n-1} \cdot (1 - e_X), \tag{6}$$

where $i \neq j$. For each coded packet, the destination receives the coded packet either on the direct link, i.e., at the intended receiver, or on the overhearing link. By using the direct link, the source keeps retransmitting the packet until the destination receives the packet. Thus, the destination always receives the transmitted packets after couple of retransmissions. For the case of overhearing, the probability of successful transmissions are calculated by using Eq. (6). Thus, the probability of receiving a packet from the relay, either by direct transmission or overhearing is then given by:

$$\Pr(R \to D_i) = \frac{1}{2} + \frac{1}{2} \sum_{n=1}^{\infty} (1 - e_{OH}{}^n) \cdot e_X{}^{n-1} \cdot (1 - e_X). \tag{7}$$

The probability of receiving a usable packet at destination $D_i$ is then the probability of receiving both the combined packet and the corresponding overheard packet:

$$\Pr(D_i) = \Pr(R \to D_i) \cdot \Pr(S_j \to D_i), \quad i \neq j. \tag{8}$$

The expected total amount of packets needed to be sent from the sources is then the expected packet overhead:

$$E_{pkts}\left[S_i \to R\right] = \Pr(D_i)^{-1}. \tag{9}$$

The total amount of transmissions from each source is then the total amount of packets needed to be sent multiplied by the transmissions per packet:

$$E_{\Sigma TX}\left[S_i \to R\right] = E_{pkts}\left[S_i \to R\right] \cdot E_{TX}\left[S_i \to R\right], \tag{10}$$

where $E_{TX}\left[S_i \to R\right]$ is derived in the unconstrained forwarding case, Eq. (2). The number of transmissions from the relay is the amount of packets transmitted from a source, weighted by the expected transmissions per packet:

$$E_{\Sigma TX}\left[R \to D_i\right] = E_{pkts}\left[S_1 \to R\right] \cdot E_{TX}\left[R \to D_i\right]. \tag{11}$$

Again, $E_{TX}\left[R \to D_i\right]$ is derived in Eq. (2) in the unconstrained forwarding case. The efficiency is then:

$$\eta_{COPE--like} = E_{\Sigma TX}\left[S_i \to R\right] + E_{\Sigma TX}\left[R \to D_i\right]. \tag{12}$$

### *5.3. CORE, No Feedback, No Recoding*

The CORE schemes incorporate RLNC with generation size $M_i$ for flow $i$. For the theoretical analysis, we assume no linear dependency, i.e infinite field size. Due to the similarities to the COPE–like transmission scheme, the expected amount of transmissions per packet between sources and relay is the same. Furthermore, the probability of a destination overhearing a packet between source and relay, Eq. (6), is also valid for this scheme, as well as Eq. (7) describing the probability of a destination receiving a packet from the relay. Due to the assumption of large field size, each overheard packet grants one additional degree of freedom (independent linear combination) to the receiver and only limited by the generation size of its corresponding RLNC generation. The expected amount of degrees of freedom received from overhearing is:

$$E\left[S_i \to D_j \mid p_{S_i}\right] = \begin{cases} \Pr(S_i \to D_j) \cdot p_{S_i}, & \Pr(S_i \to D_j) \cdot p_{S_i} \leq M_i \\ M_i, & \text{otherwise} \end{cases} , i \neq j.$$

where $p_{S_i}$ is the amount of packets sent from $S_i$. The number of degrees of freedom received in combined packets from the relay are found similarly to that of overhearing:

$$E\left[R \to D_i \mid p_R\right] = \begin{cases} \Pr(R \to D_i) \cdot p_R, & \Pr(R \to D_i) \cdot p_R \leq M_1 + M_2 \\ M_1 + M_2, & \text{otherwise} \end{cases} .$$

where $p_R$ is the amount of packets sent from the relay. Due to symmetric traffic $p_{S_i} = p_R = p$. The amount of packets needed to be transmitted from each of the sources, $p'$, can be found by solving the optimisation problem in Equation (13).

$$p' = \min(p) \quad \text{s.t. } M_1 + M_2 \geq E\left[R \rightarrow D_j \mid p\right] + E\left[S_i \rightarrow D_j \mid p\right] \tag{13}$$

The total amount of transmissions are then found by applying the channel overhead per packet between the nodes, which are equal to that derived in the unconstrained forwarding case.

$$E_{\Sigma TX}\left[S_1, S_2, R\right] = \left(E_{TX}\left[S_1 \rightarrow R\right] + E_{TX}\left[S_2 \rightarrow R\right] + E_{TX}\left[R \rightarrow D_i\right]\right) \cdot p' \tag{14}$$

The efficiency is then the total amount of transmissions needed to deliver the combined generation size in packets:

$$\eta_{CORE,\ no\ feedback,\ no\ recoding} = \frac{E_{\Sigma TX}\left[S_1, S_2, R\right]}{M_1 + M_2}. \tag{15}$$

### *5.4. CORE, No Feedback*

Due to recoding, the simplification of a degree of freedom per received packet does not hold: the relay and destination may share common degrees of freedom, thus received packets may be non–innovative. The amount of degrees of freedom from flow 1 and 2 is termed $i_1$ and $i_2$ at relay. Likewise, the degrees of freedom at $D_i$ are termed $i'$ and $i_O$ for the combined generation and the overheard generation respectively.

Only when $i' < i_1 + i_2$ can $i'$ be incremented. Thus, when receiving a packet after a transmission from each of the transmitters, the probability of increasing $i'$ is two–sided: if the relay received a packet, and if $i' \leq i_1 + i_2$ is a valid prior state. Though, if the relay failed to receive, only the case of $i' < i_1 + i_2$ can increase $i'$. The probabilities of these inequalities being true after $tx$ transmissions from each transmitter, are given in Eq. (16) and (17). The probability of increasing $i_O$ requires that $i_O < M_j$, where $j$ is the index of the overheard flow. The probability of this inequality being true after $tx$ transmissions is given in Eq. (18).

$$\Pr(i' < i_1 + i_2 \mid tx) = \sum_{k=0}^{M_1} \left( \Pr(i_1 = k \mid tx) \cdot \sum_{l=0}^{M_2} \Pr(i_2 = l \mid tx) \cdot \Pr(i' < l + k \mid tx) \right). \tag{16}$$

$$\Pr(i' \leq i_1 + i_2 \mid tx) = \sum_{k=0}^{M_1} \left( \Pr(i_1 = k \mid tx) \cdot \sum_{l=o}^{M_2} \Pr(i_2 = l \mid tx) \cdot \Pr(i' \leq l + k \mid tx) \right). \tag{17}$$

$$\Pr(i_O < M_2 \mid tx) = \sum_{k=0}^{M_2 - 1} \binom{tx}{k} (1 - e_{OH})^k \cdot {e_{OH}}^{tx-k}. \tag{18}$$

Eventually, $i'$ and $i_O$ will share degrees of freedom, due to the combined generation including the overheard generation. We assume that the common degrees of freedom, $c$, is minimised:

$$\min(c) \quad s.\ t. \quad i' + i_O - c \ \leq M_1 + M_2. \tag{19}$$

To simplify notation, Eq. (20), (21), (22) and (23) are defined. Eq. (20) and (21) are the probabilities of $i_1$ and $i_2$ being equal to a value $G$ given $tx$ transmissions from the transmitters. Eq. (22) and (23) are the probabilities of $i' < G$ and $i' \leq G$ given $tx$ transmissions from the relay.

$$\Pr(i_1 = G \mid tx) = \binom{tx}{G} (1 - e_X)^G \cdot {e_X}^{tx-G}. \tag{20}$$

$$\Pr(i_2 = G \mid tx) = \binom{tx}{G} (1 - e_X)^G \cdot {e_X}^{tx-G}. \tag{21}$$

$$\Pr(i' < G \mid tx) = \sum_{k=0}^{G-1} \binom{tx}{k} \cdot (1 - e_X)^k \cdot {e_X}^{tx-k}. \tag{22}$$

$$\Pr(i' \leq G \mid tx) = \Pr(i' < G + 1 \mid tx). \tag{23}$$

The sum of the two probabilities of increasing $i'$, Eq. (16) and (17), weighted by the probability of said events, yields the probability of adding a degree of freedom to $i'$ at the $k$–th transmission from the relay, thus also the mean increases in degrees of freedom in $i'$ at the $k$–th transmission from the relay, stated in Eq. (24). The same approach is used for $i_O$ in Eq. (25).

$$\begin{aligned} Pr(i' \to i' + 1 \mid tx) = &\big((1 - e_X)^3 + 2 \cdot (1 - e_X)^2 \cdot e_X\big) \cdot \Pr(i' \leq i_1 + i_2 \mid tx) \\ &+ (1 - e_X) \cdot {e_X}^2 \cdot \Pr(i' < i_1 + i_2 \mid tx) \end{aligned} \tag{24}$$

$$\Pr(i_O \to i_O + 1 \mid tx) = (1 - e_{OH}) \cdot \Pr(i_O < M_2 \mid tx). \tag{25}$$

In Eq. (26), $R(tx)$ is the sum of degrees of freedom from both $i'$ and $i_O$, after $tx$ transmissions from the relay.

$$R(tx) = \sum_{k=1}^{tx} \big( \Pr(i' \rightarrow i'+1 \mid k) + \Pr(i_O \rightarrow i_O+1 \mid k) \big). \tag{26}$$

The required amount of transmissions, $tx'$, from the relay before $R(tx) = M_1 + M_2$, i.e., full rank, is computed in the optimisation problem of Eq. (27). As each source and the relay transmit an equal amount of packets, the expected total amount of packets is $3 \cdot tx'$. The theoretical efficiency of CORE with no feedback is then given by the ratio between transmitted and original source packets, as in Eq. (28).

$$tx' \;=\; \min(tx) \quad s.\ t. \quad R(tx) = M_1 + M_2. \tag{27}$$

$$\eta_{CORE,\ no\ feedback} = \frac{3 \cdot tx'}{M_1 + M_2}. \tag{28}$$

*5.5. CORE*

The CORE scheme stops the source when the relay has reached full rank. The probability of the relay not having reached full rank in $i_1$ or $i_2$ is then the probability that another packet needs to be transmitted:

$$\Pr(i_j < M_j \mid tx) = \sum_{k=0}^{M_j - 1} \Pr(i_j = k \mid tx). \tag{29}$$

In Eq. (30), $R(tx)$ denotes the sum of degrees of freedom contributed from $i'$ and $i_O$, after $tx$ transmissions from the transmitters. Eq. (30) differs from (26) as CORE stops the respective source when the relay reaches full rank, thus the probability of increasing $i_O$ is weighted by the probability of the overheard source $S_j$ transmitting another packet:

$$R(tx) = \sum_{k=1}^{tx} \big( \Pr(i' \rightarrow i' + 1 \mid k) + \Pr(i_O \rightarrow i_O + 1 \mid k) \cdot \Pr(i_j < M_j \mid k) \big). \tag{30}$$

The amount of transmissions needed from the relay is then found by the optimisation problem:

$$tx' \;=\; \min(tx) \quad s.\ t. \quad R(tx) = M_1 + M_2. \tag{31}$$

Eq. (31) only describes the amount of transmissions needed from the relay, in order for the destination to receive all information. The total amount of transmissions on the network is given by Eq. (32), incorporating the probability of the sources being stopped:

$$tx_{\Sigma} = \sum_{k=1}^{tx'} \big( \Pr(i_1 < M_1 \mid k) + \Pr(i_2 < M_2 \mid k) + 1 \big). \tag{32}$$

The efficiency of the full CORE scheme is then given by the ratio between the total transmissions and the amount of original source packets:

$$\eta_{CORE} = \frac{tx_{\Sigma}}{M_1 + M_2}. \tag{33}$$

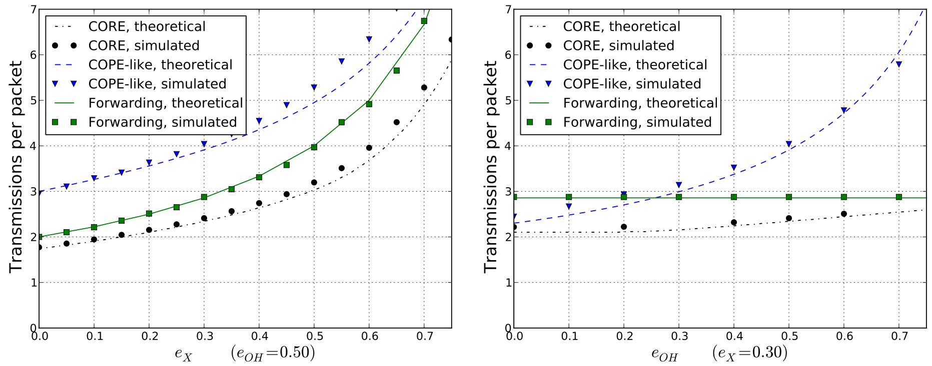
### *5.6. Theoretical Throughput*

The analytical expressions derived throughout this section are used to calculate the throughput for a few selected schemes given the offered load by the sources. The maximum throughput is calculated as the reciprocal efficiency. It is assumed that when this throughput is reached, it saturates, with the exception of the forwarding scheme. The calculations are based on an ideal setup, where we assume that all packets are of equal length, the flows offer the same load (symmetric), that all nodes are synchronised and is only valid in the investigated topology. These assumptions make the calculations reflect an upper–bound on the throughput.

The theoretical throughput for unconstrained forwarding is valid until the network reaches congestion, which is at 50 % and 33 % relative offered load at packet loss ratios of 0 % and 33 %, respectively. When the relative offered load gets higher, the throughput for the constrained forwarding is valid. However, this amount of throughput is not reached until the 802.11 MAC–fairness is fully employed, which is at 67 % and 44 % relative offered load at loss ratios of 0 % and 33 % respectively. The throughput for the COPE–like and CORE schemes is the same in a loss free environment. However, this is not the case when using lossy channels. Figure 5 shows the theoretical throughput for the forwarding, COPE–like and the CORE schemes showing the gain of CORE with respect to COPE–like.

### *5.7. Numerical Results*

Our original paper [9] presented a series of numerical results produced through simulations. We compare the theoretical efficiency with our previous simulations in Figure 3. The simulation setup is based on a cross




(a) Increasing loss ratio on the direct links. Error ratio for the overhearing fixed on 0.50.

(b) Increasing loss ratio for overhearing. Error ratio on the direct links fixed on 0.30.

Figure 3: Comparison of theoretical analysis of efficiency and the simulated results. For the CORE simulations a RLNC generation size of 64 is used with coding coefficients from $GF(2^8)$.

topology as shown in Figure 6. The main performance metric is the number of transmissions per successfully received packet. Our simulation assumes a given number of packets per source and stops when both destinations have received their respective source packets. Packet losses are assumed to be i.i.d. RLNC is performed using the Kodo C++ library.

The results show that overhearing in the COPE–like should only be used opportunistically and, particularly, for low packet loss probability. On the other hand, CORE shows a great resilience towards losses on both direct and overhearing channels, with better performance over both forwarding and the COPE–like transmission scheme under all channel conditions. It should also be noted that the forwarding efficiency stated in Figure 3 corresponds to a MAC unconstrained case, i.e., where the relay gets additional priority to be able to transmit twice as often as the sources. This efficiency will go down in IEEE 802.11 systems at high rates, when MAC fairness will limit the transmissions of the relay to just one time per two transmissions of the sources (one each).

Because the RLNC provides a coding overhead that is not included in the theoretical analysis, Figure 4 is provided to show the impact of varying coding parameters for RLNC. Choosing parameters that results in a high coding overhead (low generation size, low field size) may degrade the efficiency of CORE. However, even moderate field sizes and generation sizes provide a better performance to forwarding.

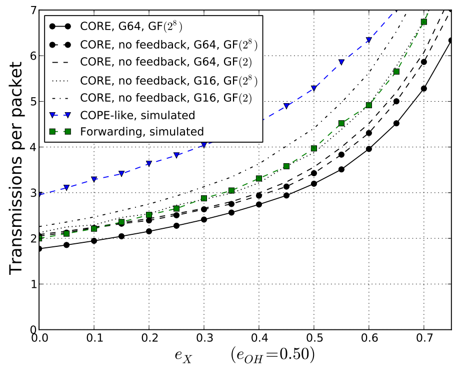



Figure 4: Impact of generation size and field size for CORE.

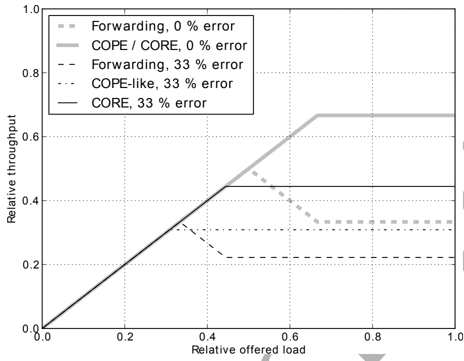



Figure 5: Theoretical throughput in a cross structure with the forwarding, COPE–like and CORE schemes, and error ratios of 0 % and 33 %.

## 6. Protocol Implementations

As CORE uses RLNC each generation must be acknowledged. In contrast, forwarding and COPE, where a Negative Acknowledgement (NACK) will be used for missing packets in our implementation.

### *6.1. Synchronisation of Generations*

Synchronising generations from flows refers to have the sources start transmitting packets from a generation at the same time, and then wait for a signal to start transmitting from the next generation. Decoding of the generation at the destinations can naturally not be synchronised. As shown in [9], partial decoding can be used even though the destinations may have to do some decoding of the flows intended for the other destination. However, by using synchronised generations the destinations will at most have to decode two generations in order to retrieve one. Synchronising the generations may also introduce some delay when changing generation, however the benefits of the simplified implementation surpass the draw back of this delay. The relay handles the synchronisation, i.e., the destinations inform the relay when they have decoded the current generation, and when all destinations have decoded the relay will inform the sources to start the next generation. If the generations are not synchronised several different generations may be mixed in the relay significantly complicating the decoding procedure, and possibly introduce decoding delay. In an unfortunate situation a destination may have to decode several generations just to retrieve one desired. When more and more generations are needed to complete the decoding of just one generation the decode complexity increases because of the complexity of Gaussian elimination.

### *6.2. Measuring Packet Loss*

Additionally, the packet loss information is also required for providing the credit based scheme. The cross topology has six links, where as the packet loss on the links from the sources to the relay is of interest together with the packet loss from the sources and relay to the destinations. The packet loss information is measured simply by counting the amount of received packets divided by the amount of sent packets. However, as the implementation uses libpcap to received packets, some packets may be overheard several times and they shall not be counted more than once. That is, the links 3 and 5 are combined into one virtual link for $D_2$, the same goes for $D_1$ and the links 4 and 6. The packet loss on the virtual links are defined as the mean packet loss over all the included links. The success ratio of the virtual channel to $D_2$ is calculated as:

$$r_{d2} = \frac{R_3 + R_5}{S_3 + S_5}, \tag{34}$$

where $r_{d2}$ is the success ratio of the virtual channel to $D_2$, $R_x$ is the amount of packets received on link $x$, and $S_y$ is the amount of packets send on link $y$. The success ratio of the virtual channel to $D_1$ is calculated in a similar fashion. The links are shown on Figure 6.

When the relay sends combined packets they will be counted on both link 3 and 4. When $S_1$ is transmitting packets to the relay via link 1 they will also be counted on link 5 as this link is only for overhearing. In order for both the relay and sources to calculate the packet loss, they need to receive counting reports from the other nodes. The counting reports covers both packets sent and packets received. Since the implementation uses synchronised generations, the counting reports can be piggybacked on the generation synchronisation messages, and thus minimize the overhead needed for exchange of the counting reports. When the counting reports are only exchanged at generation shifts, changes in the error ratio cannot be detected faster than on generation–by–generation basis. As the generations may not complete the decoding procedure at the same time, one of them will receive packets after completion, this amount of packets is then reported together with the next counting report. This makes it necessary to average over some generations to get an accurate packet loss.

*6.3. Credit Based Packet Scheduling*

Credit based packet scheduling is a method designed to transmit a judicious amount of packets to compensate for channel losses. Packets are only sent when the transmitter has credits available. The way of acquiring credits depends on the type of transmitter. The basics of the method is described in [16, 8], however the implementation uses a method that is different due to the managing of inter–flow coding.

*6.3.1. Sources*

For the forwarding and COPE approaches, the source obtains one credit per packet. For CORE, sources obtain credits each time a symbol is added to the encoder. More specifically, $s$ new symbols add $C_s = \frac{s}{r} \cdot x$ credits, where $C_s$ is the amount of new credits, $r \in (0, 1)$ is the average success ratio of the wireless channel, and $x \geq 1$ is an extra credit factor to increase the probability of decoding before triggering a time–out event and resulting in the first acknowledgement. The sources keep sending coded packets as long as they have at least one credit. Otherwise, they wait for new packet requests from the relay, for new packets to be added to the encoder, or to a packet indicating completion of the generation. In the latter, the source proceeds to send a new generation if packets are available.

Since triggering a time–out and requesting additional packets can be costly, using a $x = 1$ may delay the decoding of the generation significantly as the $C_s$ is only adding the expected number of transmissions to deliver $s$ but leaving a significant probability of not decoding. Using $x > 1$ results in a smaller number of generations to trigger additional packet requests. An adequate amount of additional redundancy using NC is preferred over feedback when the latency of the channel is large, e.g., due to a large Round-Trip Time (RTT) [19, 20]. To determine $x$, we assume in this work that losses in the network are independent and identically distributed (i.i.d.) Bernoulli trials, with the $r$ estimated from transmissions previous generations. Thus, the probability of receiving enough packets from the source at the relay is:

$$p = 1 - \sum_{i=0}^{g-1} \left( \binom{\left\lfloor \frac{g}{r} \cdot x \right\rfloor}{i} \cdot r^i \cdot (1-r)^{\left\lfloor \frac{g}{r} \cdot x \right\rfloor - i} \right). \tag{35}$$

Figure 7 shows the probability of the relay receiving $g$ packets with increasing $x$. This shows that setting $x = 1.1$, i.e., adding an extra 10 % redundancy, when $g = 60$ and $r = 0.7$, increases $p$ from $p = 0.53$ to $p = 0.93$

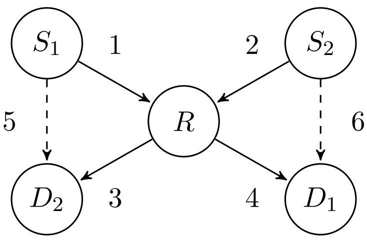



Figure 6: The links in the cross topology. Overhearing links are marked with a dashed line. Each destination has multiple links to receive packets from. The packet loss for the destinations is calculated as the mean packet loss over all their links.

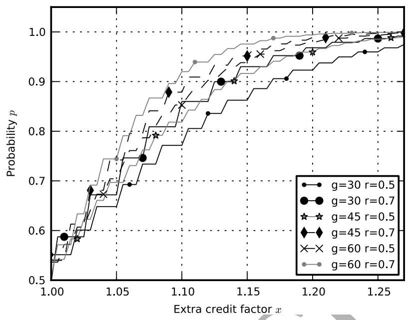



Figure 7: Probability of the relay receiving enough packets as a function of $x$, for $g = \{30, 45, 60\}$ and $r = \{0.5, 0.7\}$.

guaranteeing that another 40 % of the generations can be transmitted without a request for additional packets. The value of $x$ can be determined optimally using [19, 20]. For our current implementation, we use $x = 1.06$ unless stated otherwise.

*6.3.2. Relay*

For forwarding, COPE, and CORE with no recoding and no feedback, the relay essentially gets one new credit for each flow once it has received a new packet. For the former, one credit per flow is decreased once the packet is sent. For the latter, two credit pools are setup one per flow combined. If both credit pools have at least one credit each, an XORed packet is sent and two credits are substracted (one per flow). On the other hand, if only a single credit pool has more than one credit, then packets are forwarded and only the credit pool of the corresponding flow is decreased. As long as any credit pool has at least one credit, the relay will continue to send packets.

For CORE with a recoding relay, still two credit pools are used. $C = \frac{x}{r}$ new credits are obtained for each packet when the success ratio for the intended destination is $r$ and $x$ is the extra credit factor. In contrast with the other relay types, the recoding relay will recode a new packet from flow $i$ in the event that it only has credits in the credit pool of the other flow. Thus, it always sends combined packets. The only exception is when the relay has only received packets from one source in the beginning of a generation, since no recoding is possible for the other flow. As the other relays, the recoding relay will transmit packets as long as there are sufficient credits in any credit pool. For all cases, if all credit pools have less than one credit, the relay will either time–out or receive a `NEXT GENERATION` request

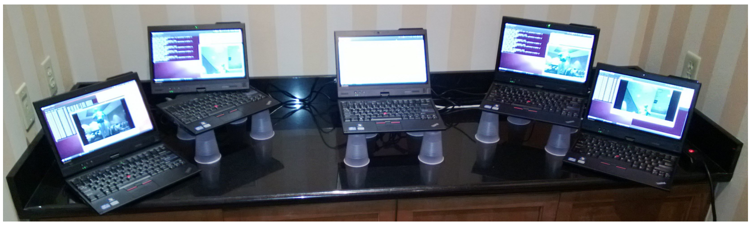


Figure 8: Picture of the testbed at IETF 86 – Orlando, FL, USA.

from the destinations. If the relay times out, it will send a packet request to the sources that have uncompleted destinations.

## 7. Description of the Testbed

The implementation for the testbed is written in C++11 and runs on five laptops connected through a Wi–Fi ad hoc network and setup to imitate the cross topology, see Figure 8. The network coding mechanisms are integrated with the kodo (9.0.9) C++ coding library [21], which also supplies the necessary RLNC operations. Ordinary UDP sockets are used for data transmission. In order to enable packet overhearing, i.e., receiving packets meant for other terminals, we set the Wireless Network Interface Controllers (WNICs) in promiscuous mode and use libpcap (`tcpdump.org`) for data reception. A drawback of using libpcap is that all packets sent to the used UDP port will be received, irrespective of the source of the packets. Thus, some packet filtering is needed. Feedback and other control messages are sent using TCP to ensure reliability. Nagle's algorithm is disabled in TCP via the socket option `TCP_NODELAY` in order to send control packets as soon as possible. TCP delayed acknowledgments are also disabled to guarantee a fast response for the control messages. The testbed consists of five Lenovo ThinkPad X220 4291–AP8, equipped with a Intel Core i5 2540M processor and a the WNIC Intel® Centrino® Advanced–N 6205, and running Debian 7.0. The WNIC driver is compat–wireless v. 3.3–2–n. We use the cross topology in Figure 6 for deployment of two sources, two destinations, and a relay. To ensure this topology, we guarantee no direct link exists between each source–destination pair using a filter in libpcap at the destinations to discard the packets from their intended source. In the future, this more complex scenario will be studied. The same filter is used to discard all but UDP traffic on the correct port. In the following, we describe the specifics of each type of node.

• *Sources:* Sources generate data randomly. The sources implement two different settings: no coding and RLNC. In the former, all the original packets

are transmitted without any NC. This is used for COPE and forwarding. The last setting implement intra–flow coding using RLNC. Regardless of coding setting, the source generates data in generations. The generation size may be constant or chosen at random.
• *Destinations:* Just like the sources, the destinations implement two settings: no coding and RLNC. The former is used for COPE and forwarding. All data are extracted when a generation is complete, regardless of using no coding or RLNC.
• *Relay:* The relay is key to the coding region and supports four schemes: forwarding, COPE, CORE "with no recoding, no feedback", and CORE.

Like the data flows, all control messages go through the relay. A control message consists of an operational code (opcode) and some parameters associated to the opcode. Control messages have a size of 28 bytes without counting the TCP/IP headers. The relay listens on TCP port 11000, while sources and destinations connect to it. When the TCP connection is made, the connecting node will send a handshake message informing the relay about the node type, ID, and if the node is a source, the ID of its destination. If the node connecting is a destination, the relay will send a `START` message, when receiving the handshake message. When the relay is connected to enough nodes, default is four, it will send a `START` message to all sources. As the relay is listening for connections, it must be started before any source or destination. Control messages are used for generation synchronisation, packet count reports, packet requests, and NACKs.

## 8. Performance Evaluation

In order to evaluate the performance of CORE and "CORE with no recoding, no feedback", later known as "CORE no recoding", the two schemes: forwarding and COPE–like has been implemented in the testbed. To make a fair comparison, the forwarding and COPE–like schemes uses generations, just like RLNC, this includes rate control ensuring that the relay is allowed to transmit twice as much as the sources. We assume symmetric losses in all the links for our performance evaluation generated synthetically. Additionally, losses will affect the overhearing channels.

Note that the measured throughput corresponds to the end-to-end throughput for the different flows at the application layer. This means that the overheads from intra- and inter-session coding introduced by CORE or COPE, respectively, are already taken into account in our comparison. This is a fair metric as it focuses on the perceived throughput for the users/devices in the system.

| | |
|---|---|
| Stop | When the relay reaches full rank in an encoder, it sends this opcode to the appropriate source to stop transmissions of the current generation. |
| Start | This opcode is sent from the relay to all other nodes, making them start the appropriate encoding and decoding processes. |
| Negative acknowledgement | This opcode is used to send NACKs when RLNC is not used. The destinations send it to the relay, which then forwards it to the sources. Up to five packets can be requested in a single NACK. |
| Next Generation | The destinations send this opcode when they have decoded the intended generation. When the relay has received this opcode from all destinations, it will send this opcode to the sources to start transmission of the next generation. |
| Handshake | The sources and destinations send this opcode after connecting to the relay |
| Terminate | Sent out from the relay to all nodes when a test is finished |
| Packet request | The relay sends this opcode to the sources if it needs more RLNC packets. |
| Packet count report | This opcode is used by the sources to inform the relay about how may packets have been sent in the last generation. |

Table 3: List of opcodes used in the testbed.

The performance evaluation is focused on the throughput of the four different schemes, measured as the data rate. The rate is calculated as the amount of bytes in a generation over the number of generations delivered per unit time. Figure 9 shows the throughput with varying error ratios. The two CORE schemes outperform forwarding and COPE for the entire range but particularly for high packet loss ratios. Already at an error ratio of 10 % , CORE has a throughput gain of more than 42 % over both the COPE–like scheme and forwarding. When the error ratio is at 50 %, CORE has a throughput gain of more than 1060 % over the COPE–like scheme and more than 698 % over forwarding. This confirms the findings in [9], which showed that COPE is highly dependent on the quality of the overhearing channels. For each measurement point each scheme is tested at least ten times, with each test consisting of 1000 generations with a generation size of 64 bytes and symbol size of 1100 bytes. Figure 10 shows the distribution of transmission time per generation with 50 % losses on each link, for each of the four schemes. The distribution is significantly wider for COPE and

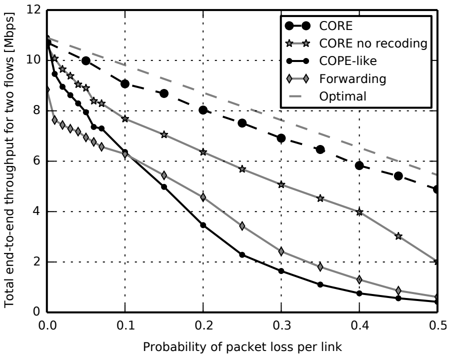



Figure 9: Throughput of the different schemes as a function of the erasure ratio and the optimal throughput.

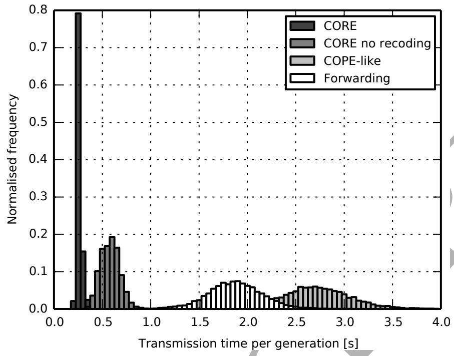



Figure 10: The distribution of transmission time per generation, with 50 % losses on each link.

forwarding than for the two CORE schemes. Additionally it is also clear that the average transmission time per generation is lowest for the two CORE schemes, this also corresponds to the higher data rate. Due to the high transmission time per generation, COPE–like and forwarding is not suitable for time–sensitive applications.

In the COPE–like implementation, the coding requirement has been relaxed compared to [22], this results in a higher throughput in high loss scenarios as the scheme converges towards forwarding. This is in line with the solution of switching to forwarding in high loss scenarios described in [3].

The optimal normalised throughput is defined as $R = 1 - e = r$, where $R$ is the maximum normalised throughput, $e$ is the packet loss ratio, and $r$ is the success ratio of the channel. This simple expression is possible using RLNC arguments and the fact that we explored symmetric channel losses. The optimal curve on Figure 9, is defined as $R \cdot s$, where $s$, the scaling factor, is set to the maximum throughput for any scheme. Under the observed channel conditions, CORE follows the same trend as an optimal scheme.

## 9. Conclusion

This paper showed a practical design and implementation in a testbed of the CORE protocol. Special attention was paid to practical design issues concerning signaling between nodes. Measured throughput performance of CORE in the testbed was shown to be higher than that of forwarding and COPE, even when packet losses were negligible. More importantly, it was shown that the performance of COPE degrades to unacceptable levels for time–sensitive applications even with moderate losses in the links, confirming the results presented in [9], which analyzed the problem by counting packet transmissions. Furthermore, it is revealed that the throughput of CORE scales following the same trend as an optimal scheme, with a gap of less than 0.43 dB. Our results not only show that a CORE implementation is possible, but also that it provides significant gains with respect to the state–of–the–art.

### Acknowledgment

This work was partially financed by the Aarhus Universitets Forskningsfond Starting Grant Project AUFF-2017-FLS-7-1, the TuneSCode project (Grant No. DFF - 1335-00125) granted by the Danish Council for Independent Research, and the DIGIT Centre.

### References

[1] R. Ahlswede, N. Cai, S.-Y. Li, R. Yeung, Network information flow, IEEE Trans. Info. Theory 46 (4) (2000) 1204 –1216.

[2] R. Dougherty, C. Freiling, K. Zeger, Insufficiency of linear coding in network information flow, IEEE Trans. on Info. Theory, 51 (8) (2005) 2745–2759.

[3] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, J. Crowcroft, XORs in the Air: Practical Wireless Network Coding, IEEE/ACM Trans. on Net. 16 (3) (2008) 497–510.

[4] J. Zhang, Y. P. Chen, I. Marsic, MAC-Layer Proactive Mixing for Network Coding in Multi-Hop Wireless Networks, Comput. Netw. 54 (2) (2010) 196–207.

[5] H. Seferoglu, A. Markopoulou, M. Médard, NCAPQ: Network Coding-Aware Priority Queueing for UDP Flows over COPE, in: Int. Symp. on Network Coding (NetCod), 2011, pp. 1 –8.

[6] H. Seferoglu, A. Markopoulou, Network Coding-Aware Queue Management for Unicast Flows over Coded Wireless Networks, in: IEEE Int. Symp. on Network Coding (NetCod), 2010, pp. 1 –6.

[7] M. Hundeboll, J. Leddet-Pedersen, J. Heide, M. V. Pedersen, S. A. Rein, F. Fitzek, CATWOMAN: Implementation and Performance Evaluation of IEEE 802.11 based Multi-Hop Networks using Network Coding, 2012, 76th IEEE Veh. Tech. Conf. (VTC).

[8] S. Chachulski, M. Jennings, S. Katti, D. Katabi, Trading Structure for Randomness in Wireless Opportunistic Routing, in: Conf. on App., tech., archi., and prot. for comp. comm. (SIGCOMM), ACM, New York, NY, USA, 2007, pp. 169–180.

[9] J. Krigslund, J. Hansen, M. Hundebøll, D. Lucani, F. Fitzek, CORE: COPE with MORE in Wireless Meshed Networks, in: IEEE Veh. Tech. Conf. (VTC), Dresden, Germany, 2013.

[10] H. Seferoglu, A. Markopoulou, K. Ramakrishnan, $I^2$NC: Intra- and Inter-session Network Coding for Unicast Flows in Wireless Networks, in: IEEE INFOCOM, 2011, pp. 1035 –1043.

[11] X. Zhu, H. Yue, Y. Wang, C and M: A New Network Coding Scheme for Wireless Networks, in: Fifth Int. Conf. on Info. Assu. and Sec. (IAS), Vol. 2, 2009, pp. 432 –436.

[12] P. Pahlavani, V. Derhami, A. M. Z. Bidoki, FENC: fast and efficient opportunistic network coding in wireless networks, TIIS 5 (1) (2011) 52–67. doi:10.3837/tiis.2011.01.003.
URL `https://doi.org/10.3837/tiis.2011.01.003`

[13] H. Seferoglu, A. Markopoulou, M. Medard, Ncapq: Network coding-aware priority queueing for udp flows over cope, in: 2011 International Symposium on Networking Coding, 2011, pp. 1–8. doi:10.1109/ISNETCOD.2011.5979091.

[14] P. Pahlevani, A. Paramanathan, M. Hundebøll, J. Heide, S. A. Rein, F. H. P. Fitzek, Reliable communication in wireless meshed networks using network coding, in: 2012 IEEE Vehicular Technology Conference (VTC Fall), 2012, pp. 1–5. doi:10.1109/VTCFall.2012.6399118.

[15] D. E. Lucani, M. Medard, M. Stojanovic, On coding for delay–network coding for time-division duplexing, IEEE Transactions on Information Theory 58 (4) (2012) 2330–2348. doi:10.1109/TIT.2011.2177562.

[16] D. Koutsonikolas, C.-C. Wang, Y. Hu, Efficient Network-Coding-Based Opportunistic Routing Through Cumulative Coded Acknowledgments, IEEE/ACM Trans. on Net. 19 (5) (2011) 1368–1381.

[17] P. Pahlevani, H. Khamfroush, D. E. Lucani, M. V. Pedersen, F. H. Fitzek, Network coding for hop-by-hop communication enhancement in multi-hop networks, Comput. Netw. 105 (C) (2016) 138–149.

doi:10.1016/j.comnet.2016.05.012.
URL `https://doi.org/10.1016/j.comnet.2016.05.012`

[18] S. Pandi, S. Wunderlich, F. H. P. Fitzek, Reliable low latency wireless mesh networks–from myth to reality, in: 2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC), 2018, pp. 1–2.

[19] D. Lucani, M. Stojanovic, M. Médard, Random Linear Network Coding For Time Division Duplexing: When To Stop Talking And Start Listening, in: IEEE INFOCOM, 2009, pp. 1800–1808.

[20] D. Lucani, M. Médard, M. Stojanovic, On coding for delay - network coding for time-division duplexing, IEEE Trans. on Info. Theory 58 (4) (2012) 2330–2348.

[21] M. V. Pedersen, J. Heide, F. Fitzek, Kodo: An open and research oriented network coding library, Lecture Notes in Computer Science 6827 (2011) 145–152.

[22] J. Hansen, J. Krigslund, D. Lucani, F. Fitzek, Bridging inter-flow and intra-flow network coding for video applications: Testbed description and performance evaluation, in: Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2013 IEEE 18th International Workshop on, 2013, pp. 7–12.

**Jonas Hansen** is an industrial Ph.D. student at Bang & Olufsen, Struer, Denmark and Aalborg University, Aalborg, Denmark working on network coding, code design and wireless distribution protocols for audio streaming. He received his B.S. degree in electrical engineering from Aalborg University in 2012. In 2013 he was a visiting student at Massachusetts Institute of Technology's Research Laboratory of Electronics, where he collaborated with the Network Coding and Reliable Communications Group. He graduated (*with cum laude*) from the Elite Masters Programme in Wireless Communication at Aalborg University in 2014. His research interests include network coding for wireless communication and multimedia transmission with an emphasis on low–latency traffic and applications.

**Jeppe Krigslund** is a software developer at Steinwurf ApS working on network coding protocols for wireless video multicast. He was a student in the Elite Masters Programme in Wireless Communication at Aalborg University (AAU), Denmark, where he also received his B.S. degree in electrical engineering in 2012. His research interests revolve around wireless communications and multimedia transmission with work including a mix of wireless mesh networks, network coding, video streaming, and cooperative protocol design.

**Daniel E. Lucani** is an Associate Professor at the Department of Engineering at Aarhus University since April 2017 as well as the CEO and lead scientist of the start-up company Chocolate Cloud ApS since June 2014. Dr. Lucani is an IEEE Senior member and the recipient of the IEEE ComSoc Outstanding Young Researcher Award for the EMEA region in 2015 (First time the award goes to Denmark). Dr. Lucani received the B.S. summa cum laude (2005) and M.S. with Honors (2006) degrees in electronics engineering from Universidad Simn Bolvar, Caracas, Venezuela, and the Ph.D. degree in electrical engineering (2010) from the Massachusetts Institute of Technology (MIT). Dr. Lucani has published more than 135 scientific papers in international journals and top-ranked international conferences as well as 7 patents and patent applications. His research interests include communications and networks, network coding, information theory, coding theory, distributed storage and computation, and their applications to cloud computing technologies necessary to enable IoT, BigData and 5G applications and services. He was the General Co-Chair of the 2014 International Symposium on Network Coding (NetCod2014) and he is an Associate Editor of EURASIP Journal on Wireless Communications and Networking. Dr. Lucani was an Associate Professor at Aalborg University (2012-2017) and an Assistant Professor at the University of Porto (2010-2012).

**Frank H. P. Fitzek** is a Professor and head of the Deutsche Telekom Chair of Communication Networks at TU Dresden coordinating the 5G Lab Germany. He received his diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology Rheinisch-Westflische Technische Hochschule (RWTH) Aachen, Germany, in 1997 and his Ph.D. (Dr.-Ing.) in Electrical Engineering from the Technical University Berlin, Germany in 2002 and became Adjunct

Professor at the University of Ferrara, Italy in the same year. In 2003 he joined Aalborg University as Associate Professor and later became Professor.
He co-founded several start-up companies starting with acticom GmbH in Berlin in 1999. He has visited various research institutes including Massachusetts Institute of Technology (MIT), VTT, and Arizona State University. In 2005 he won the YRP award for the work on MIMO MDC and received the Young Elite Researcher Award of Denmark. He was selected to receive the NOKIA Champion Award several times in a row from 2007 to 2011. In 2008 he was awarded the Nokia Achievement Award for his work on cooperative networks. In 2011 he received the SAPERE AUDE research grant from the Danish government and in 2012 he received the Vodafone Innovation prize. In 2015 he was awarded the honorary degree Doctor Honoris Causa from Budapest University of Technology and Economy (BUTE).
His current research interests are in the areas of wireless and mobile 5G communication networks, mobile phone programming, network coding, cross layer as well as energy efficient protocol design and cooperative networking.

**Peyman Pahlevani** is an Assistant Professor in the department of Computer Science and Information Technology of Institute for Advanced Studies in Basic Science (IASBS), Iran. He received his PhD in Wireless Communication from Aalborg University, Denmark in 2014. Prof. Pahlevani was a visiting researcher at the Computer Science Department of University of California, Los Angeles (UCLA). Moreover, He has been collaborated with different institutes and universities such as MIT, and Porto universities. Beside his academic backgrounds, he extended his skills to solve practical challenges in the area of video communication in his previous carrier, as a Researcher Engineer at AIRTAME company. His research interests are in the area of wireless communication, network coding and its applications, vehicular communications, cooperative networking, and video streaming over WiFi links. Dr. Pahlevani has also served as TPC member for international conferences and as reviewer for high impact journals, such as IEEE Transactions on Vehicular Technology.