OPPORTUNISTIC NETWORK CODING IN WIRELESS MESH NETWORKS

Kemal Alič

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia, December 2015

Evaluation Board:

Prof. Dr. Mihael Mohorčič, *Chairman, Jožef Stefan Institute, Ljubljana, Slovenia*

Prof. Dr. Gorazd Kandus, Member, Jožef Stefan Institute, Ljubljana, Slovenia

Prof. Dr. Andrej Košir, Member, Faculty of Electrical Engineering, University of Ljubljana, Slovenia

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL



Kemal Alič

OPPORTUNISTIC NETWORK CODING IN WIRELESS MESH NETWORKS

Doctoral Dissertation

PRILOŽNOSTNO OMREŽNO KODIRANJE V BREZŽIČNIH ZANKASTIH OMREŽJIH

Doktorska disertacija

Supervisor: Prof. Dr. Aleš Švigelj

Ljubljana, Slovenia, December 2015

Index

Abstract ..... **VII**

Povzetek ..... **IX**

Abbreviations ..... **XI**

Symbols ..... **XIII**

1 Introduction **1**
1.1 Network Coding 1
1.2 Motivation and Hypothesis 2
1.3 Contributions 3
1.4 Organization of the Dissertation 3

2 Background **5**
2.1 Network Coding 5
2.1.1 Encoding and decoding 6
2.2 Opportunistic Network Coding 7
2.2.1 Information required in the coding process 8
2.3 Wireless Mesh Networks 9
2.4 The Network Layer 10
2.5 The Link Layer 10
2.6 Terminology 10

3 Bearing Opportunistic Network Coding **13**
3.1 The Algorithm 14
3.2 BON Design Details 19
3.2.1 Packet queues 19
3.2.2 Coding procedure 19
3.2.3 Signalization 21
3.2.4 Packet reception 21
3.2.5 Packet retransmission 22
3.2.6 Pseudo broadcast 23
3.3 Tolerance Angle 23
3.3.1 Self-adaptation 23

4 Methodology for Performance Evaluation **27**
4.1 Simulation Model 27
4.1.1 Traffic management 27
4.1.2 The network layer 29
4.1.3 The network coding layer 29

4.1.4 The link layer 30
4.1.5 The network topology 31
4.2 Performance Metrics 31
4.3 Simulations Settings 33
4.3.1 Network coding settings 33
4.3.2 Traffic, topology and variable parameters settings 34

5 BON Performance Evaluation 39
5.1 BON Procedure Performance 39
5.2 Influence of Tolerance Angle on BON Performance 44
5.3 Summary of the BON Performance 46

6 Network and Traffic Design Aspects in Network Coding Enabled Wireless Networks 47
6.1 Topology Design Impact on Network Coding 47
6.2 Traffic Distribution Impact On Network Coding 49
6.3 Packet Length Evaluation 53
6.4 Analysis of Output Queue Length 55
6.5 Design recommendations for network coding-supported wireless networks . 56
6.6 End user perception of Network coding 57
6.6.1 Quality of service parameters 57
6.6.2 Analysis of Coding Fairness 59
6.6.3 Summary on the end user perception of network coding 61

7 Conclusions 63
7.1 Summary of Contributions 63
7.2 Future Work 64

Acknowledgements 65

References 67

Index of Figures 71

Index of Tables 73

Index of Algorithms 75

Appendix A: Simulation Model Parameter Setting Overview 77

Appendix B: Node Implementation in OPNET Modeler 79

Appendix C: State Diagrams for BON and COPE Procedures 81

Bibliography 83

Abstract

Wireless Internet traffic is growing at staggering rates. For these reasons there is an ongoing pursue for using radio resources as rationally as possible. Network coding provides strong means in utilising the full network capacity.

In general, network coding takes advantage of the broadcasting capabilities of the shared wireless medium to provide benefits in terms of bandwidth, delay and transmission power, or to conserve scarce resources, such as energy. In this thesis we are interested in opportunistic network coding used to improve the network goodput and delay. The term opportunistic describes the opportunistic nature of this type of coding, as packets are encoded only if the opportunity arises and there are no mechanisms to increase the number of coding opportunities.

In this thesis, we look at the opportunistic network coding as an independent communication layer in the communications stack. With this we need to create a system that searches for coding opportunities independent of other layers. We argue that in practice, there are too many possibilities to provide reliable estimation on which node pairs provide good coding matches. Hence, the possibilities subset should be made in order to provide a reliable decision process.

In order to substantiate our claim, we propose a network coding layer completely transparent to layers above and below, namely Bearing Opportunistic Network coding (BON). It is an effective procedure that introduces little overhead to the network and its coding decisions are based solely on guesswork. Further, BON dynamically adapts to changes in traffic patterns and continuously changing link conditions. We show that the proposed approach increases goodput in all the analysed networks and traffic conditions in comparison to the similar network coding approaches for *2–25 %* while maintaining the same Quality of service parameters.

When designing the procedure we were looking for, a balance between the number of coded packets and the decoding success rate. Efficient coding algorithm does not only attempt to encode as many packets as possible, but rather makes sure that the encoded packets are very likely to be decoded. Making incorrect coding decisions leads to unnecessary retransmissions, which results in a waste of the wireless resources. The high goodput is achieved with BON due to a fine compromise between finding coding opportunities and keeping the overhead low.

Use of custom developed simulations instead of models allows us to analyse many aspects of opportunistic network coding. We systematically investigate opportunistic network coding in the scope of the construction of efficient and robust large scale network-coding-enabled wireless mesh networks. The influence of several wireless network and traffic design parameters on the performance of opportunistic network coding in terms of quantitative measures is examined. We show that opportunistic network coding can improve the performance of different networks and supported applications. Depending on the traffic distribution and topology characteristics, the goodput in comparison to case where coding is not used can be increased up to *330 %*, though numbers in more general cases are between *10* and *40 %*.

Povzetek

Priča smo skokovitemu porastu brezžičnega internetnega prometa. Fizične zakonitosti brezžičnih komunikacij nam narekujejo racionalno uporabo spektra in nenehno iskanje izboljšave tehnologij. Med slednjimi uporaba omrežnega kodiranja ponuja vrsto rešitev, ki lahko prispevajo k povečanju prenosnih kapacitet.

Omrežno kodiranje izkorišča lastnosti razpršene oddaje v deljenem mediju in izboljšuje prenosne kapacitete, zakasnitve ali nižjo porabo moči. V tem delu se osredotočimo na priložnostno omrežno kodiranje, ki ga uporabimo za povečevanje prenosne kapacitete in nižanje zakasnitev pri prenosu. S terminom priložnostno opisujemo samo naravo takega načina omrežnega kodiranja, ker kodiranje izvajamo samo v primeru, ko se pojavi priložnost, medtem ko mehanizmov za povečevanje le-teh ne uporabljamo.

V tem delu priložnostno mrežno kodiranje predstavimo kot neodvisno komunikacijsko plast v komunikacijskem sistemu. Sistem avtonomno sprejema odločitve o kodiranju paketov brez zanašanja na informacije, ki prihajajo iz drugih plasti. Trdimo, da dober kodirni postopek potrebuje omejitev kodnih možnosti, saj je v praksi kodirnih možnosti preveč, da bi jih lahko kvalitetno ovrednotili.

Predlagamo komunikacijsko plast, s katero podpremo naše trditve. Smerno priložnostno omrežno kodiranje (BON - Bearing Opportunistic Network) je zadolženo za priložnostno omrežno kodiranje in je popolnoma neodvisno od višje in nižje ležečih komunikacijskih plasti. Postopek je efektiven in vnaša malo komunikacijske režije v komunikacijo. Kodirni postopek je še posebej enostaven, saj temelji izključno na ugibanju stanja, vendar se kljub temu prilagaja trenutnemu stanju omrežja in prometnim tokovom. Pokažemo, da s predlaganim postopkom v analiziranih omrežjih z različnimi prometnimi obremenitvami dosežemo *2–25 %* povečanje pretoka koristnih informacij, s tem da dosegamo enako kakovost storitve.

V fazi razvoja postopka smo iskali ravnotežje med številom kodiranih paketov in uspešnostjo dekodiranja le-teh na sprejemnih vozliščih. Efektiven kodirni postopek mora skrbeti, da se čim več kodiranih paketov uspešno dekodira in ne zgolj kodira. Neprimerno izbran set kodiranih paketov se na enem ali več vozliščih ne bo odkodiral, kar pomeni, da moramo ponoviti pošiljanje in zato nekoristno pouporabiti skupni medij. S kodirnim postopkom BON dosegamo dobre rezultate, saj smo našli dobro ravnotežje med uspešnim kodiranjem in režijskimi stroški.

V delu smo analizirali več aspektov priložnostnega omrežnega kodiranja s pomočjo razvitega simulacijskega modela. Sistematično pristopimo h gradnji robustnih, velikih zazankanih omrežij, ki upošteva zakonitosti omrežnega kodiranja in z upoštevanjem zakonitosti le-tega omogoča večje prenosne kapacitete. Podamo pregled vplivov mrežnih in prometnih konstrukcijskih zakonitosti na priložnostno omrežno kodiranje s stališča kvantitete storitve. Pokažemo, da z uporabo priložnostnega omrežnega kodiranja lahko izboljšamo delovanje omrežja in podprtih storitev. Pretok koristnih informacij z analiziranimi postopki v posebnih primerih lahko povečamo do *330 %*, medtem ko je povečanje pretoka v bolj splošnih primerih med *10* in *40 %*. Dobitki zavisijo predvsem od prometne razporeditve in lastnosti omrežij.

Abbreviations

ACK	=	Acknowledgement
BON	=	Bearing Opportunistic Network Coding
COPE	=	Coding Opportunistically
ETX	=	Expected Transmission Count Matrix
MAC	=	Media Access Control
OSI	=	Open Systems Interconnection
Tx	=	Transmission
XOR	=	Exclusive OR operation

Symbols

$\oplus$	=	XOR, Exclusive OR operation
$\vec{b}_{ij}$	=	bearing for packet on the relay node
ε_i	=	tolerance angle
ETE	=	End-to-end delay
J	=	Jain's Index
g	=	goodput
G	=	gain
$NNWT$	=	Normalised Number of Wireless Transmissions
$MaxGain$	=	*MaxGain* as defined with Eq. (4.7)
P_{ij}	=	probability of successful packet delivery between nodes i and j
p_i	=	i-th packet
SNC	=	success in packet coding
V_i	=	i-th node
V_R	=	relay node

1 Introduction

Studies on Internet traffic trends show staggering growth rates of between *70* and *115* percent per year [1]. This remarkable expansion in growth is the result of new applications, modifications in user habits, and changing trends:

- New industries and operator types such as data centre providers and cloud operators are present in the market.
- Devices that enable "non-stop" content generation and consumption, such as tablet computers and smartphones, have found their way into our everyday lives.
- New technologies have enabled a rapid change in user behaviour patterns (e.g., instead of watching linear broadcast content via classic television, the same or personalized and self-selected content can be accessed via YouTube).
- DVD stores or rental outlets seem to be forgotten as video content can be accessed through online streaming services such as Netflix.

At the moment mobile Internet traffic represents *15* percent of total Internet traffic. However, the trends suggest that mobile Internet traffic is increasing at *50* percent per year, and that tablet computer users generate *2.5* times more traffic than those using smartphones. In addition, the migration to the fourth-generation (4G) mobile communications technology standards i.e. Long Term Evolution (LTE) promise *19* times more traffic than we have had to cope with in the 3G mobile communications technology standards. According to the predictions, by 2017 there will be *70* percent of traffic originating from mobile devices, with mobile traffic increasing by *13* times and that of video by *16* times [2]. For these reasons there is an ongoing pursuit for using the radio spectrum as rationally as possible where lately technologies such as cognitive networks and network coding are often mentioned.

1.1 Network Coding

Network Coding is now a recognised method that can increase the capacity of a single-source communication, such as a multicast stream [3], in order to achieve the maximum data-transfer rate in multicast networks [4], thus adding to the rational spectrum usage.

With network coding, the conventional store-and-forward paradigm in relaying scenarios is replaced with a process-and-forward approach. Instead of forwarding packets in the same form as they are received, the intermediate nodes combine the received outgoing packets using an algebraic function. Hence, a typical network coding mechanism is composed of two main operations: (1) coding of outgoing packets on intermediate nodes and (2) decoding of incoming packets upon their reception on nodes [5].

In general, network coding is taking advantage of the broadcasting nature of the shared wireless medium to provide benefits in terms of bandwidth, delay and transmission power [6], or to conserve scarce resources, such as energy [7, 8]. Moreover, network coding can be used in mobile networks [9] to increase reliability with respect to channel errors [10, 11], in wireless networks over TCP [12], or in Delay Tolerant Networks, such as deep-space links [13], with all demonstrating promising results.

Depending on the application of the network coding the implementation affects different OSI layers. For example, in multicast scenarios network coding is typically implemented

in the application layer [14], while a two-stage network coding is deployed in the physical layer [15-17] for increased spectrum efficiency. In this work, we are interested in opportunistic network coding [5, 6, 18] for static wireless mesh networks, such as metropolitan Wi-Fi networks [19] or Wireless Sensor Networks (WSNs) [20] which is usually addressed between network and link layers.

Opportunistic network coding is used to improve the network goodput and delay. The term opportunistic describes the opportunistic nature of this type of coding, as packets are encoded only if the opportunity arises and there are no mechanisms within it to increase the number of coding opportunities.

Routing-path selection, traffic-load balancing, and network coding decisions are the three main issues tackled in the development of opportunistic network coding schemes [5]. Traffic flows need to be routed through the network in order to provide as many coding possibilities as possible [21-26]. Since coding opportunities are more likely to appear on nodes that are already fully loaded, and routing new traffic through highly loaded nodes might lead to packet dropping due to overflown queues [27], load balancing needs to be considered along with routing. The coding schema should be able to exploit the wireless medium to find coding opportunities so that it provides nodes with encoded packets that can be decoded at the recipient nodes. Making incorrect coding decisions leads to wasting of network bandwidth and therefore poorer performance.

Optimal integration of the opportunistic network coding schema into the existing architecture of a network is not straightforward. Its implementation influences several functional components in the existing network protocol stack, e.g., scheduling, routing, and congestion control [12, 27-29]. As a result, the interaction with all these components increases the implementation difficulty and possibly also the computational complexity. For example, Rule Based network coding (RNC) [5], COPE [6, 18] and CATWOMAN [30, 31] all rely on information gathered in the network layer; the RNC also relies on scheduling information from the link layer; and the well-known COPE coding procedure has been introduced with numerous improvements, boosting its performance at the expense of an increased complexity [32-35]. Furthermore, network coding schemes often maintain the state of traffic flows to search for coding opportunities. Network coding also brings an additional overhead into networks in terms of additional packets and headers added to the regular packets. To boost the network coding performance, the overhead has to be kept low, as analysed in [36]. However, as shown in BEND [37], the additional overhead can be compensated with increased coding opportunities.

1.2 Motivation and Hypothesis

The work has been driven by two major motives:

1. Find the main influences that impact the performance of opportunistic network coding.
2. Construct a coding process that balances coding opportunities and decoding success rate.

The working hypothesis of the presented dissertation is that an opportunistic network coding procedure can be developed that can operate in a completely independent manner from other layers, relying only on the information that it can collect on its own. This information should be sufficient for a good and general estimation on which packets to encode, and compensate for the possible introduced overhead, to obtain similar or better performance gains as provided with the existing comparable network coding procedures.

1.3 Contributions

In this thesis we investigated opportunistic network coding for wireless mesh networks. Our aim was to improve their performance in terms of increased goodput, while maintaining the quality of experience for all the users. The results presented in this dissertation are expected to contribute to further research and development of methods, models, procedures, and algorithms of the next generation wireless networks.

The main contributions of the thesis are as follows:

C1 A comprehensive analysis of the parameters influencing opportunistic network coding performance, resulting in the definition of a systematic approach to constructing efficient and robust large scale network coding-enabled wireless mesh networks.

C2 A novel general procedure for network coding named Bearing Opportunistic Network Coding (BON) that operates over multiple unicast sessions and it introduces very low overhead to the network.

C3 Reference implementation of the proposed network coding procedure BON into the existing network stack for performance evaluation and comparison with existing network coding algorithms.

Further contributions that are also of high importance to the researchers working in the area include the following:

F1 The conceptual design of a simulation model for the performance evaluation and comparison of opportunistic network coding algorithms and their respective supporting mechanisms.

F2 A new metric *MaxGain* used to compare performance of network coding algorithm(s) in different environments.

1.4 Organization of the Dissertation

Chapter 2 provides the background necessary for understanding the remainder of the thesis. We introduce network coding in general and give more details on the opportunistic network coding. Wireless mesh network properties required for understanding network coding discussions are also given.

Chapter 3 introduces a novel opportunistic network coding procedure. The core of the procedure is the algorithm for making decisions on which packets to encode (C2). In addition, all the implementations details for the supporting mechanisms are described (C3).

Methodology for performance evaluation is presented in Chapter 4. It is divided into two main parts, i.e. a simulation tool developed for evaluation purposes (F1), and metrics used in the evaluation (F2).

The proposed procedure, namely BON evaluation is presented in Chapter 5. The evaluation is carried out in three representative synthetic networks. Main characteristics of the proposed algorithm can be seen in the presented results.

In Chapter 6 we investigate the influence of several wireless network and traffic design parameters on the performance of opportunistic network coding in terms of quantitative measures (C1). In addition, we also carry out the analysis of the influence that network coding has on the delay, jitter and coding fairness i.e. qualitative network performance measures.

Finally, Chapter 7 concludes the thesis by highlighting the main findings and pointing out directions for future work.

2 Background

2.1 Network Coding

The main idea behind network coding is simple. Rather than just simply storing and forwarding packets, the coding operations are performed over packets. With the wide emergence of wireless technologies the network coding idea can be used to improve network performance.

We adopt the definition for network coding from [38] that says that network coding is coding at a node in a packet network (where data is divided into packets and network coding is applied to the contents of packets), or more generally, coding above the physical layer. This is different than the definition by the network information theory, which is generally concerned with coding at the physical layer.

Network coding benefits are best demonstrated over the throughput increase. The famous butterfly given in [3] considers multicast in a wireline network. The case is presented in Figure 2.1 and features two destinations (V_{D1} and V_{D2}) and two source nodes (V_{S1} and V_{S2}). Both destinations are interested in the packet streams from both sources. Let us assume that all links have the same capacity e.g. 1 packet per second. If we consider a traditional communication procedure, where only storing and forwarding packets is considered, the link between two relay nodes (V_{R1} and V_{R2}) is a bottleneck and each of the destinations receives one stream in full and only half of the second stream. If the intermediate node R_1 breaks from the traditional routing paradigm of packet networks and performs coding operation over packets from different streams (e.g. forms a new packet by performing linear operation such as *XOR* over the two packets) and outputs the resulting packet ($p_1 \oplus p_2$), the full content of both streams can be delivered to both destinations. The two destinations must perform additional coding operations to get hold of both packets i.e. V_{D1} *XORs* the encoded packet ($p_1 \oplus p_2$) with p_1 thus also obtaining packet p_2. Likewise, the V_{D2} *XORs* the encoded packet ($p_1 \oplus p_2$) with p_2 thus also obtaining packet p_1.

The network coding benefits are not limited to either wireline networks or multicast applications. The opportunistic network coding which is investigated in this work shows, how benefits can be achieved also in the wireless networks and with multiple unicast streams. Prior to having a look at the opportunistic network coding, the encoding and decoding needs to be explained.

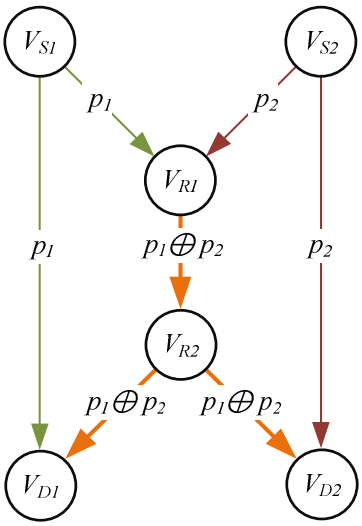


Figure 2.1: The butterfly network.

2.1.1 Encoding and decoding

We suppose that, at the source node, we have K packets $p_1, p_2, ..., p_K$, that we want to encode together and each consists of L bits. When the packets to be combined are not of the same size, the shorter ones are padded with zeros. With linear network coding, outgoing packets are linear combinations of the original packets. Hence we can represent any packet p as a linear combination of packets $p_1, p_2, ..., p_K$, namely:

$$p = \sum_{k=1}^{K} z(k) p_k \tag{2.1}$$

where $z(k)$ is the code vector associated with p_k.

A destination node collects packets and if it has K packets with linearly-independent global encoding vectors it is able to recover the message packets. Decoding can be done by Gaussian elimination.

In this dissertation we assume that linear operation for coding packets is *XOR* operation, hence the code vector is known in advance and its distribution is not required. With the style of coding and the supporting retransmission schema the code vector selection does not affect results, hence the best solution is to make it known in advance to eliminate the overhead.

We can also see that with network coding all the information that can be obtained is important and carries some useful information. We may claim that information placed in a packet is not a separate entity, but rather part of a system. Nevertheless, with network coding the nodes are required to listen to all the transmissions within their range, and hold the information for the decoding purposes.

2.2 Opportunistic Network Coding

The term opportunistic network coding describes the opportunistic nature of this type of coding, as packets are encoded only if the opportunity arises and there are no mechanisms to increase the number of coding opportunities. Its benefits can be best explained with the help of the simple example depicted in Figure 2.2. Consider a three-node chain topology where node V_1 has a packet p_1 destined for node V_2, and node V_2 has a packet p_2 for the destination node V_1. The two nodes can exchange their packets through an intermediate relay node V_R. Hence, nodes V_1 and V_2 send packets p_1 and p_2 to V_R. In a conventional system where network coding is not used, V_R first transmits p_1, which is then followed by the p_2. In the case where a network coding procedure is applied, the V_R performs an algebraic operation over the two packets (e.g., *XOR*) and sends out an encoded packet $p_R = p_1 \oplus p_2$. Upon reception, node V_1 *XORs* the received encoded packet p_R with the sent packet p_1 and obtains the packet p_2. The same procedure is used on node V_2. V_2 is familiar with the content of p_2 and *XORs* it with the just-received packet p_R, thus obtaining p_1 officially destined to it. By using network coding the number of transmissions that all three nodes need to perform in order to deliver both packets to their destinations has been reduced from *4* to *3*.

The example in Figure 2.2 presents a topology that is suitable for the chain-structure coding, i.e., a topology where the two packets are travelling in the opposite directions and get encoded in the intermediate node(s). In such a case, all the recipient nodes have sufficient information for successful packet decoding without packet overhearing. Furthermore, packet retransmission may appear only due to the unsuccessful delivery of the encoded packet to one of the recipient nodes.

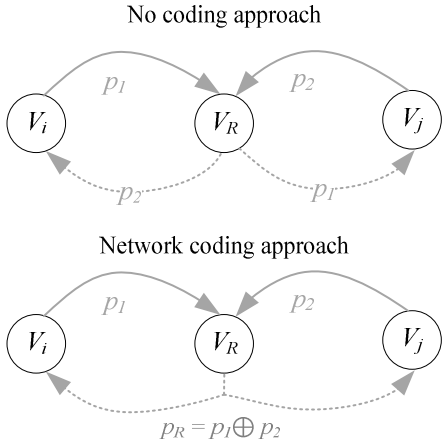


Figure 2.2: Example of how opportunistic network coding increases the throughput.

A more interesting case arises with a more general network deployment. In such a case, the question is which packets the coding node should encode together in such a way that the recipient nodes will be able to decode the encoded packet. When a node has a packet to forward, it needs to know whether encoding of this packet with other queued packet(s) might save bandwidth, i.e., it needs to determine whether the receivers can decode the encoded packets. Consider the situation in Figure 2.3. Nodes V_1 and V_2 are sending packets p_{13} and p_{24} via the relay node V_R to the nodes V_3 and V_4, respectively. If V_R creates an encoded packet from p_{13} and p_{24}, e.g., $p_R = p_{13} \oplus p_{24}$, the receiving node V_3 will be able to decode the encoded packet p_R if V_3 has overheard the p_{24} transmission from V_2 to V_R. The situation is similar for V_4, which needs to overhear the p_{13} transmission from V_1 to R.

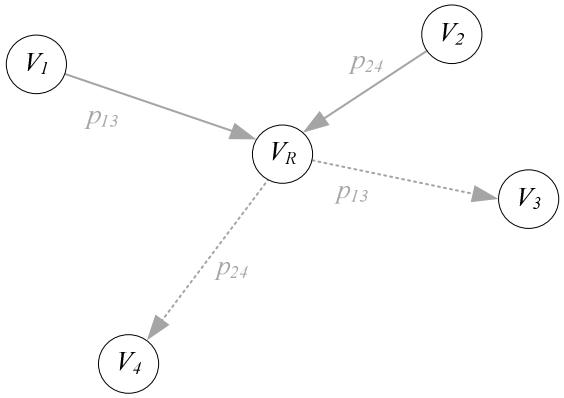


Figure 2.3: General relaying scenario in which network coding can be used.

Let us highlight why the coding packets that travel in the opposite directions are preferred. Assume the situation presented in Figure 2.2 where packets p_n and p_m travel in the opposite directions, i.e. p_n from V_j via V_R to node V_i and p_m via versa. If we describe the links between nodes with packet delivery probabilities, i.e. probability that packets are successfully transmitted between two links, then we can estimate the chances that the encoded packet $p_n \oplus p_m$ will be successfully received and decoded on the two nodes is

$$P = P_{V_R V_j} P_{V_R V_i}. \tag{2.2}$$

Where $P_{V_R V_j}$ and $P_{V_R V_i}$ are the probabilities that packets are successfully transmitted from node V_R to node V_j and to node V_i, respectively.

In a more general situation where packet p_n travels from V_i to V_j and p_m from V_k to V_l, then the probability that the encoded packet $p_n \oplus p_m$ will be successfully received and decoded on the two recipient nodes is:

$$P = P_{V_R V_j} P_{V_R V_i} P_{V_i V_l} P_{V_k V_j}. \tag{2.3}$$

The difference to the first case is that also two overhearing events need to take place for each of the receiving nodes to have sufficient information to perform decoding.

2.2.1 Information required in the coding process

The more packets we encode together or looser the conditions in coding process are, the higher is the possibility that receiving nodes will not be able to decode those packets. However, if coding conditions are hard to meet, then there are only few coding opportunities and low bandwidth saving benefits. Good coding algorithm does not only attempt to encode as many packets as possible, but rather makes sure that the encoded packets are very likely to be decoded. Making incorrect coding decisions leads to unnecessary retransmissions, which results in wasting of the wireless resources. Therefore, an important development issue is finding a balance between the number of coded packets

and the successful decoding rate, by using small network coding overhead.

Coding decisions can be based on knowledge, e.g. each node records the state of the traffic for its neighbouring nodes, although as shown in [6] this is only sufficient for a portion of coding decisions. Thus, guessing mechanism needs to be used in the coding decision process. COPE, for instance, uses information gathered through routing for estimating the probability that a certain packet will be present at a certain node. The drawback is that the access to the network layer is required which is not always possible. An alternative is to acquire an estimation of coding possibilities through own network coding evaluation. This could be performed through (i) transmitting the probe (measurement) packets; or (ii) by estimating the coding success of the regularly encoded packets. The first option introduces additional overhead to the network and is therefore not in our favour. The second option is difficult to be set for general network performance since the number of coding options (combinations of packet's previous hop and packet's next hop) grows with the number of neighbours a node has faster than exponentially (i.e. $N!/2$). In practice, there are too many possibilities to provide a reliable estimation on which pairs provide good coding matches. In addition, the ongoing measurements should be able to detect the changes in wireless links conditions and react fast in case of their degradation. The subset of possible coding matches should be made in order to provide reliable decision process and react to changes in wireless environment.

2.3 Wireless Mesh Networks

Wireless mesh networks describe networks where nodes connect to each other via multi-hop wireless links. Usually we consider them to be self-sustainable, i.e. adaptive and self-organised in order to be able to maintain connectivity.

Wireless mesh networks are very limited in capacity. E.g. the 802.11 family of products are advertised to support data rate of up to *54 Mbps*. Still, “protection” mechanisms such as binary exponential backoff, rate adaptation, and protocol overheads, cut the throughput by *50 %*. Moreover, owing to backward compatibility with 802.11b and 802.11g is encumbered with legacy issues that reduce throughput by an additional *~20 %*. Besides, the actual bandwidth available to individual clients can be even much lower due to the shared nature of the wireless medium. Also, mesh networks usually operate in the unlicensed *2.45 GHz* Industrial, Scientific and Medical (ISM) frequency band. Hence, bandwidth is shared also with other networks or network devices, e.g. Bluetooth peripheral devices, spread-spectrum cordless phones, or microwave ovens. Interference affects the quality of a wireless link and, consequently, its error rate and achievable capacity [39].

Regardless of their capacity limitations wireless mesh networks are appealing in areas with scarce wired infrastructure. Such places can be found in less developed areas or in places such as tunnels, battlefields, on board of public transportation etc. Furthermore, due to low operational and deployment costs commercial deployment can also be found in urban areas with the main application of offering cheap Internet connectivity.

In a mesh network, each router can be seen as part of a backbone infrastructure and also as an access point for end users. In this work we see mesh routers as access points that offer end-users access, hence traffic generated on nodes can be considered aggregated from multiple users.

2.4 The Network Layer

The network coding procedures discussed in this work are positioned between the network and the link layer. The well-known COPE algorithm relies on the information gathered in the network layer, i.e. the expected transmission count matrix (ETX):

$$ETX = \frac{1}{P_{V_i V_l} P_{V_k V_j}} \tag{2.4}$$

With this metric each link between two nodes is described with the inverse of the packet delivery probability and its successful acknowledgement. In other words, two successful transmissions need to take place.

Routing-path selection also influences the performance of network coding. Routing that takes into account also network coding can further boost the network performance, as can poorly selected routing in combination with network coding result even in poorer network performance. The proposed BON procedure works as an independent layer, but its performance is indirectly still related to the performance of routing.

In this work we are using link-state routing protocol that is widely adopted in the wireless mesh networks. The influence of the routing protocol selection on the performance results is balanced out by comparing results also to COPE coding procedure. We adopted symmetric routing (with symmetric we refer to path selection of traffic flows that travel in the opposite directions). Routing that takes into account also network coding is not considered in this work, though it has been investigated inside the workgroup and the main results and findings can be found in [40, 41].

2.5 The Link Layer

Wireless mesh networks were initially based on the IEEE 802.11 standard. Therefore, we adopt general properties of the standard important from the opportunistic network coding perspective.

The standard offers mechanisms for packet distribution to the nodes within their communication range. Unicast inherently assure reliable delivery of packets to the next hop while broadcast offers distribution of data to multiple users. With network coding we want all the nodes to listen to all the transmissions. To this end, the broadcast mechanism in the link layer seems a suitable solution at a first glance. However, due to the absence of collision detection and backoff on the source node the reliability of broadcast is questionable. For this purpose we adopt pseudo broadcast first presented in [6] and discussed in Section 3.2.6 .

2.6 Terminology

Terminology related to network coding is still not very widely accepted in the broader communications community. We try to adopt terminology that has been established in prior work and summarise it in Table 2.1.

Table 2.1: Definitions of terms used in this work.

Term	Definition
Native packet	Packet that came into network coding layer from the upper layer and is a candidate to be encoded.
Encoded packet	Packet that carries information of more than one packet.
Coded packet	Native packet that has been coded into encoded packet.
Decoded packet	Coded packets that have been successfully decoded.
Undecodable packet	Encoded packet that with a given information in particular node cannot be decoded.
Packet pool	Pool where the node saves all the heard or overheard native packets for a limited period of time.
Packet source	The node that created a packet.
Packet destination	The node with packet's final destination.
Packet previous hop	The node that transmitted the packet to the current node.
Packet next hop	The node to which the packet is headed in the next transmission.
Overhearing	With overhearing we refer to actions when nodes successfully receive packets that are not destined to them, but store them for decoding purposes.

3 Bearing Opportunistic Network Coding

This section describes the design and implementation details of Bearing Opportunistic Network coding (BON), a novel opportunistic network coding procedure that performs network coding on the packet level and can significantly improve the network throughput. In the same way as with other similar approaches, the procedure is positioned between the network and link layers. The core of the procedure is the coding algorithm which identifies coding opportunities and provides benefits to the system by forwarding multiple packets in a single transmission. The nature of the wireless media, where packet transmissions are overheard by multiple clients, is embraced. At no additional costs the system exploits redundancy that takes place when packets are delivered to multiple nodes.

The main benefit of BON is simplicity, as it introduces little overhead to the network, and nodes do not need to keep track of the received traffic for their neighbouring nodes. We assume that each node has positions of all of its neighbouring nodes, and that nodes have fixed locations. Algorithm makes coding decisions based solely on the information about the packet's previous and next hop node position, hence, it acts completely independently of all other communication layers, and is completely protocol agnostic.

The BON main modules are presented in Figure 3.1. All the outgoing packets that arrive from the Network layer on the node are distributed into three outgoing queues (listed according to priority) for the acknowledgement packets, retransmitted (native) packets, and regular outgoing (native) packets. In the Packet coding module, the coding algorithm searches for coding opportunities and looks for opportunities to attach acknowledgement messages to outgoing packets. The self-adaptation process observes the coding process and based on its success adopts the coding-related parameter. On the receiving side, received acknowledgements are used to cancel scheduled packet retransmissions. If possible, encoded packets are decoded, or destroyed otherwise. All native packets are stored into the packet pool for decoding purposes and sent to the upper layers for further processing.

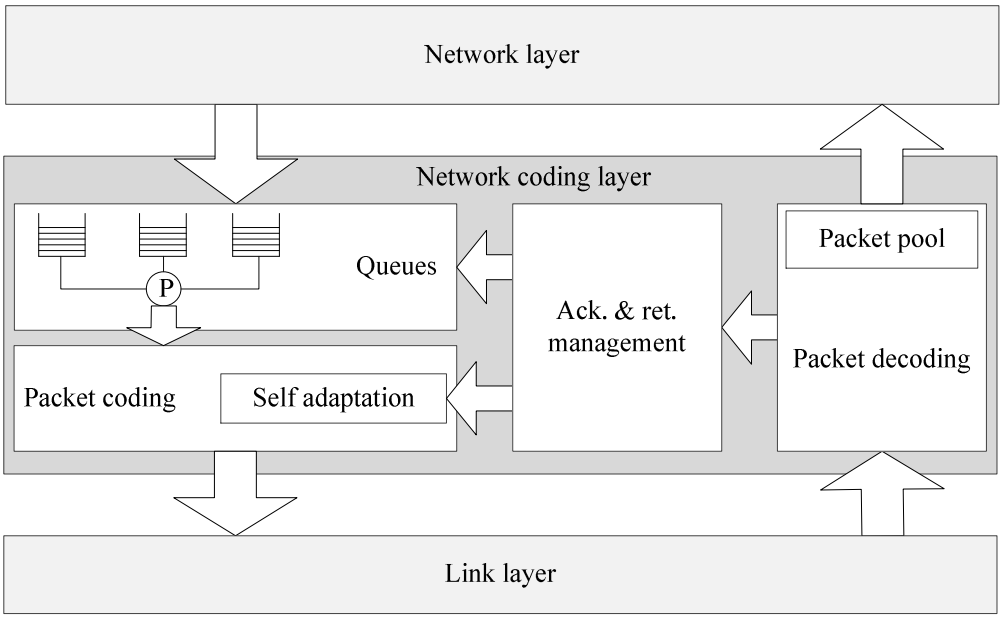


Figure 3.1: BON procedure main modules.

3.1 The Algorithm

The decision making on which packets to encode together can be based on the information already available on the node or the required information needs to be gathered. The COPE and CATWOMAN for example build on the information already available on the node and acquire the information gathered by the routing protocol. This is an excellent option as no new additional overhead is introduced to the network. The drawback with this solution is that the routing information has to be available in the layer where coding procedure is implemented. With communication components available on the market this is not always possible. By collecting the required information through measurements scheduled periodically (e.g. such as for example routing link delivery probability measurements) we introduce additional overhead, thus lowering goodput and benefits of network coding. Data can also be collected by just observing the traffic going through the network coding module e.g. collecting the statistics on which packets have been encoded and how many retransmissions were needed. The main drawback of such an approach is that there is a rather small portion of packets that get encoded. Collecting them over longer periods of time is not an option as this would make the system slow and unable to adapt to the fast changes on wireless links.

The BON coding process is based on the local bearing (as used in the navigation) of the packet, which is defined on the relay node and depends on the positions of the packet's previous hop and the packet's next hop. Packets that have not travelled at least one hop are not codable, and so a bearing is not defined for them.

Let us use the example from Figure 3.2 to explain the definition of bearing. A packet p_m has been transmitted from the node V_i with location (X_i, Y_i) to the relay node V_R. The next hop for the packet p_m is the node V_j with location (X_j, Y_j). We define bearing for the packet p_m on the relay node V_R as a unit vector calculated as:

$$\vec{b}_{ij} = \frac{(X_j - X_i,\ Y_j - Y_i)}{\left\|(X_j - X_i,\ Y_j - Y_i)\right\|} \tag{3.1}$$

and translated into three dimensional space:

$$\vec{b}_{ij} = \frac{(X_j - X_i,\ Y_j - Y_i,\ Z_j - Z_i)}{\left\|(X_j - X_i,\ Y_j - Y_i,\ Z_j - Z_i)\right\|} \tag{3.2}$$

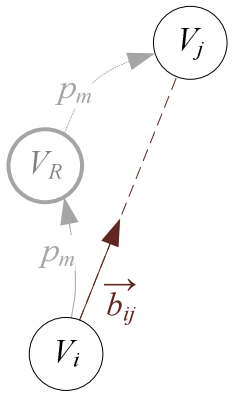


Figure 3.2: Graphical presentation of a bearing.

In the coding process we are primarily interested in packets that are travelling in opposite directions. Without the loss of generality we build on a two-dimensional example presented in Figure 3.3 where the packet p_m has the same next hop as was the packet p_n's previous hop and vice versa (the case of the so-called chain topology). In such a case, the bearings of the two packets are $\vec{b}_{ij} - \vec{b}_{ji} = \vec{0}$. The two packets can be seen as representatives of the two flows that are travelling in the opposite directions. Coding of such packets is an ideal combination as no overhearing between the nodes is needed and the recipient nodes can always decode the encoded packet.

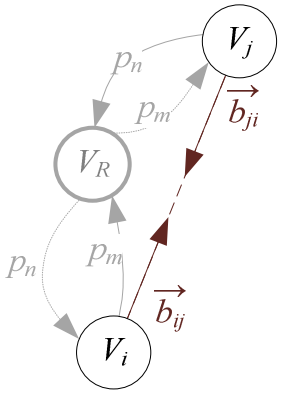


Figure 3.3: Bearings for two packets travelling in the opposite direction.

Since such opportunities (i.e. when packets travel in exactly opposite directions) do not provide a sufficient number of coding opportunities, the coding opportunities need to be searched for also amongst other packets. We can assume that in general the nodes that are located in the *vicinity* of the transmitting node have a better chance of successfully overhearing the packet transmission than nodes that are located far from it, hence, also provide a possibly good coding opportunity. With the BON coding algorithm we are looking for packets (p_i) that are codeable with the transmitting packet (p_t) on the relay node that have been transmitted from the *vicinity* of the node that sent the packet p_t to the relay node and vice versa.

Definition 1: With BON the *vicinity* of the node is described as an area within the shape of an infinite cone with the apex in the packet source, and cone axis with the direction $\vec{b}_{ij}$ and the aperture $2\varepsilon_i$.

Example of such an area for a packet p_m that is on the node V_R, with the previous hop V_i and the next hop V_j is shown in Figure 3.4, where the cone has the apex in node V_i and the aperture $2\varepsilon_i$. The node V_j lies directly on the cone axes. With ε_i we define how large the *vicinity* is, hence we refer to it as tolerance angle.

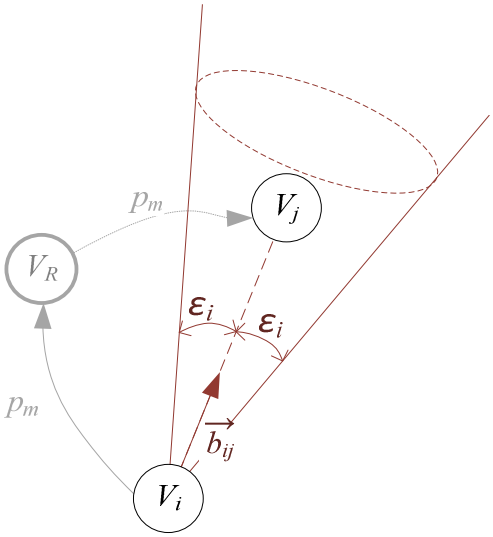


Figure 3.4: Graphical presentation of the *vicinity* in the shape of an infinite cone.

Definition 2: With BON the packets p_n and p_m are codeable on the relay node if the next hop of p_n is in the *vicinity* of the previous hop of p_m and vice versa, where *vicinity* for p_n and p_m is set according to the Definition 1 with ε_i and ε_j accordingly.

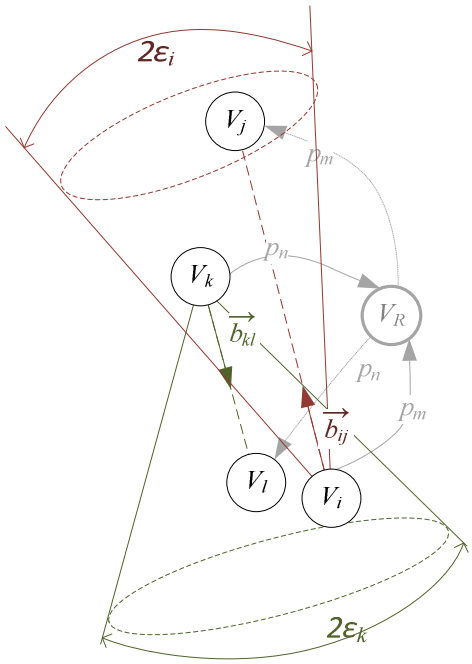


Figure 3.5: Graphical presentation of general coding case for a matching packet pair.

Let us illustrate using general two-dimensional examples from Figure 3.5 and Figure 3.6. One example illustrates a coding for two packets and the second example illustrates a non-matching packet pair. In both presented cases the initial situation is the same. Let us point out that the location of the V_R has no impact on the coding decision, and its location is changed only for the reasons of graphical presentation. Packets p_n and p_m are both on the relay node V_R. V_i and V_j are the previous and the next hop for p_m and V_k and V_l are the previous and the next hop for p_n. ε_i and ε_j are tolerance angles that relate to packets' previous hops on the relay V_R.

In Figure 3.5 we can see that the node V_i is located in the *vicinity* of the node V_l and the similar goes for the node V_k, which is in the *vicinity* of the node V_j. Since we want that the condition in Definition 2 is met for both situations in the given example, the two packets are codeable.

In Figure 3.6 we can see that the node V_i is located in the *vicinity* of the node V_l and the node V_k is outside the *vicinity* of the node V_j. Since we want that the condition is met for both situations in the given example the two packets are not codeable. Packets p_n and p_m would be codeable if ε_i was larger. By increasing the tolerance angle we cover a larger area and thus increase probability that packets will meet the conditions. However by increasing the tolerance angle parameter, the probability that the receiver will not be able to decode the packet is also increased. By reducing the parameter ε towards zero, the coding opportunities are reduced, but the probability of a successful packet decoding on the receiving nodes is increased.

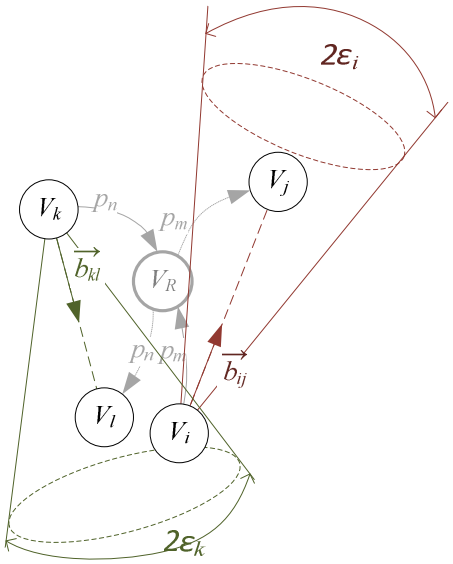


Figure 3.6: Graphical presentation of a general coding case for a non matching coding pair.

By further generalization the BON procedure allows also coding of multiple packets. Multiple packets are encoded into one encoded packet when all packet pairs and their corresponding nodes are within the *vicinity* area. The higher the ε values, the higher the possibility of coding multiple packets on the other hand, if at least one $\varepsilon=0$ only two packets can be encoded. In BON the balance between coding opportunities and successful packet decoding is set through the parameter ε, which is adapted automatically on every node and is discussed in Section 3.3.1 .

3.2 BON Design Details

In order to make use of the BON network coding process, several supporting mechanisms are required to be implemented on the nodes. Using BON, the nodes change the way packets are processed after being received and a new signalization is required. Moreover, BON requires that the nodes are put in a broadcast receiving mode, while a pseudo-broadcast mechanism (see Section 3.2.6) is used for sending the packets. In addition, the BON process is self-adaptable to follow the network's dynamic changes and modifications in the traffic patterns.

The heart of the BON procedure is the part where the decisions on which packets to encode are made. We strived to make the process as independent as possible from the other OSI layers, as this makes the implementation easier and more straightforward. Furthermore, the decision process and the supporting process need to remain simple. We are not attempting only to encode as many packets as possible, but rather to make sure that the encoded packets are very likely to be decoded. Making incorrect coding decisions leads to unnecessary retransmissions, which results in waste of the wireless resources that we are trying to use wisely. Therefore, finding a balance between the number of coded packets and the successful decoding rate is of great importance in network coding procedures.

All the nodes in the network listen to all the transmissions and try to overhear as many packets as possible – including the ones that are not addressed to them. All the overheard packets are saved into the packet pool for decoding purposes. A copy of a received packet (i.e., a packet headed to the node) is also stored in the packet pool, while the original is forwarded to the network layer.

3.2.1 Packet queues

As already mentioned, BON coding procedure uses three outgoing queues for three different packet types i.e.:

- ACK packets. These are standalone acknowledgement messages that are created only when no opportunity has been found to add them as a header to regular outgoing packets. This is explained in detail in Section 3.2.3 .
- Retransmitted (native) packets are packets that have been already sent out as part of the encoded packet and acknowledgement message has not been received. This may happen if acknowledgement message has been lost, or has not arrived yet, the encoded packet was lost or receiving node was not able to decode the encoded packet.
- Regular outgoing (native) packets that have arrived from the network layer and have arrived from one of the neighbouring nodes or have their origin on this node.

When there are packets present in one of the output queues, the BON procedure signals the link layer that the access to the wireless media is required. When the link layer signals that it has successfully gained access to the channel, the queue is selected depending on its priority. If we are dealing with retransmitted or regular queue, the coding procedure is initiated.

3.2.2 Coding procedure

The coding process using the FIFO system always takes the packet p_0 that is at the head of the output queue (retransmission or regular) and searches for possible coding opportunities with the other packets in the regular outgoing queue. If there are no coding options found, the packet is sent out as-is.

The complete coding procedure is presented in Figure 3.7. The coding algorithm first

tries to find coding opportunities among packets of the same size, but if none are found, then opportunities are sought for in the other subqueue. In the worst case, all of the packets in the output queue are evaluated in combination with p_0 using Definition (2). If the condition is met for the packet p_n, then the two packets will be coded together. The process continues the evaluation with the remaining packets in the output subqueue, however each packet candidate from the output queue is now evaluated twice with respect to once to p_0 and p_n. Definition 2 has to be true for both packet pairs for all three packets to be encoded. If such a packet is found, the process continues with the packets in the output-queue evaluation, using the evaluation Definition 2 for all the packet pairs.

All the selected packets are coded into one encoded packet using a linear operation, e.g., *XOR*. When encoding packets of different sizes, the shorter packets are padded with zeros. The receiving nodes can remove the padded zeros by checking the packet-size field in the IP header.

For each of the native packets coded into the encoded packet a retransmission event is scheduled and is cancelled only when the acknowledgement message is received. Before sending out the packet to the wireless device, the acknowledgement headers are possibly added to the packet.

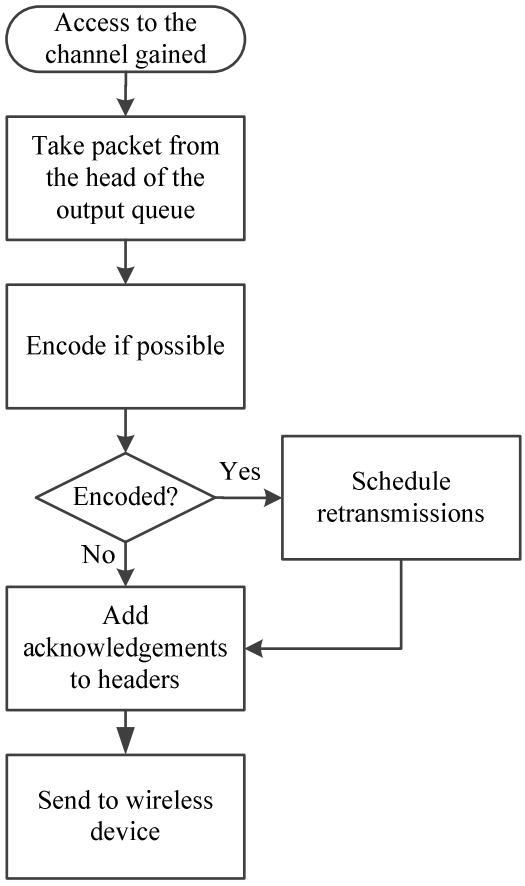


Figure 3.7: A flow chart of the BON coding process.

3.2.3 Signalization

Since only static nodes are used and under investigation, acknowledgement messages are the only signalization packets required in BON. We foresee acknowledgement messages in the network coding layer only for native packets that have been received as part of the encoded packet. Acknowledgement messages are generated by the receiving node, designated as a packet's next hop. Please note that we are discussing only the acknowledgement messages in the network coding layer, while acknowledgements in the other layers remain as they were. The acknowledgement message in the network coding layer is required to indicate the success in the packet decoding procedure.

Depending on the application and the expected benefits, different ways of sending acknowledgement messages can be used. We use cumulative acknowledgement in a structure proposed in [6]. Acknowledgement massages are sent in a bulk every time an opportunity arises and when there are new messages in the pool. The latest received packet sequence number is used for reference. This is followed by a sequence of Boolean values that also indicates status of the received packets for the packets with lower sequence number. E.g. an entry {*A, 50, 01011111*} confirms that packets *50, 48, 46-42* have been received and that packets *49* and *47* are still missing.

To make this operate, packet sequence number needs to be recorded. Each node indexes outgoing packet per each neighbour individually and the same holds true for all the received packets. We use 16-bit counters and 8-bit sequence for the previously received packets.

Acknowledgement messages confirm reception for every received native packet that has been received as part of the encoded packet. Cumulative acknowledgement report messages are broadcasted periodically, every T_u (referred to as acknowledgement packets). If the opportunity arises, the acknowledgement messages are attached to the regular outgoing packets. In this case cumulative acknowledgement can be seen only as an additional header, thus introducing less overhead. In case that the upper limit of acknowledged packets per node (*9* packets in current configuration) is reached, the acknowledgement process is triggered and acknowledgement massages are sent out as individual packet. Cumulative acknowledgement messages reduce the overhead compared to the individual acknowledgement messages.

3.2.4 Packet reception

In addition to handling acknowledgement packets, the recipient node needs to be able to also handle native and encoded packets, which both can carry additional header that contains acknowledgement massages.

Upon packet reception in the network coding module, further actions depend on whether the packet is encoded or native. In the case when an encoded packet (consisting of *M* native packets) is received, the process checks the packet pool where all the received, sent and overheard packets are stored for decoding purposes. The encoded packet can be decoded if a node has in its packet pool at least *M-1* packets. The process has to determine whether it has already received *M-1* of the packets encoded in the encoded packet. If not, the encoded packet cannot be decoded and it is simply dropped. If the node has at least the required *M-1* packets, i.e., sufficient information, it decodes the encoded packet using these packets with the *XOR* operations, thus obtaining a native packet that has not been received before. From here on the process is the same as upon receiving a native packet. If the packet is new (i.e. newer received before), its copy is inserted into the packet pool for decoding purposes. It does so for every received native packet, as all the received packets are potentially needed for further decoding purposes. The process checks whether the node is the next hop of the native packet. If so, and if the packet has been a part of the encoded packet, an

acknowledgement message is scheduled and the packet is sent to the network layer for further processing.

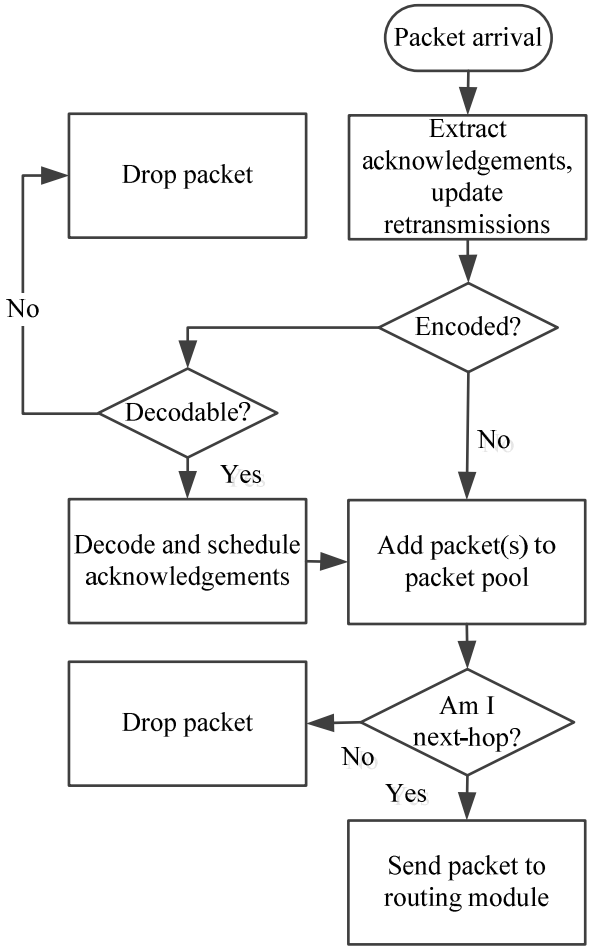


Figure 3.8: A flow chart of the BON decoding process.

3.2.5 Packet retransmission

If an ACK message is not received within a predetermined time, the retransmission event is triggered. A native packet that has been sent out as part of the encoded packet and has not been acknowledged, is placed in the retransmission output queue.

Packets require retransmissions for two reasons. Packets cannot be decoded on the recipient node (i) because there are not enough packets in the packet pool, and (ii) because packets get lost in the transmission/reception procedure.

The packets in the retransmissions queue can be coded again. However, coding opportunities are searched for only within the packets in the regular output queue. This mechanism ensures that it is not possible for the same set of packets to be coded again. Hence, the situation where the same set of packets is encoded again is avoided.

With every packet retransmission we want to increase the probability of successful packet decoding on the recipient node. As discussed in Section 3.1 , we can do so by decreasing the parameter ε. Hence, we foresee that with every packet retransmission parameter ε is decreased.

3.2.6 Pseudo broadcast

With network coding we need to overhear as many transmissions as possible for the network coding mechanisms to seize their full potential. Hence, a natural approach would be to use a broadcast mechanism. As shown in [6] this does not work due to poor reliability and lack of backoff. A mechanism that is compatible with or introduces little change to the protocol stack was searched. An approach proposed in [6] has been adopted. They propose a pseudo broadcast mechanism which unicasts packets that are meant for the broadcast.

The link-layer destination field is set to the MAC address of one of the intended recipients. Since all the nodes are set in the promiscuous mode, they can overhear packets not addressed to them. When a node receives a packet with a MAC address that is identical to its own, it sends an ACK message to the sender. Regardless of the address of the next hop of the packet, the node sends the packet to the network coding module.

3.3 Tolerance Angle

With the tolerance angle we can set the *vicinity* area that influences coding opportunities (see Section 3.1). The tolerance angle is set for every node individually.

Following motivation and desire to keep the overhead to the minimum, the information on which the tolerance angle is regulated is collected locally with resources that are available on the node itself, i.e. regular outgoing packets, retransmissions count, number of encoded packets, number of outgoing packets.

This section is based on the heuristic observation of the network and main finding on which the adaptation algorithm has been developed and its main parameters set are:

- Since there is a limited number of outgoing packets and there are several neighbours, it is unfeasible to set the tolerance angle for every neighbouring node (similar to discussion in Section 2.2.1). Hence, there is only one tolerance angle on each node, i.e. ε_i and ε_k from Definition 2 are $\varepsilon_i = \varepsilon_k = \varepsilon$.
- Where there are many coding opportunities, the tolerance angle should encode only packets that have high possibility of decoding. In other words, on the nodes with the highest traffic load the tolerance angle should in general be low.
- Where there are no coding opportunities found, the media should be tested if coding opportunities are being overlooked. In practice this means that on nodes with low traffic load the tolerance angle should be high.

3.3.1 Self-adaptation

In the decision making process, the tolerance angle ε must be known following Definition 2. By setting the parameter ε to too low values, the chances are that the algorithm will not find all the possible coding opportunities, which results in poorer performance. Setting the parameter ε to too high values results in a greater probability that the recipient nodes are not able to decode the packet content and therefore the packets need to be retransmitted, which results in poorer performance. Furthermore, due to the very variable quality of link quality in wireless networks, the algorithm should be able to adapt the parameter ε quickly and automatically.

The tolerance angle ε is adjusted based on the parameter representing the success in packet coding (*SNC*), which is on node the V_i calculated as:

$$SNC_{V_i} = \frac{K_{RV_i}}{K_{CV_i} K_{PV_i}} \tag{3.3}$$

where $K_{CV_i} > 0$ is the number of native packets that were sent out as part of the encoded packets, i.e. coded packets, on the node, and $K_{PV_i} > 0$ is the number of packets that were passed from the network layer to the network coding layer. K_{RV_i} is the number of network coding layer retransmitted packets on the node. High *SNC* value indicates high share of packet retransmissions that took place due to an unsuccessful packet decoding or unsuccessful packet delivery. In this case we may assume that conditions are not good and we want to consider tightening the coding conditions, i.e. lowering ε.

Low *SNC* value is obtained when K_{RV_i} / K_{CV_i} is low or when K_{PV_i} is high enough to compensate for even for high K_{RV_i} / K_{CV_i} ratio. In the first case we are dealing with good coding conditions as there is low number of retransmissions in comparison to number of coded packets and therefore we want to consider losing the coding conditions, i.e. increasing ε. The second situation when K_{PV_i} is high may take place when there is actually little coding going on and the node is processing many packets. In general we consider this situation as an opportunity to test the media and thus we want to loosen the coding conditions. Still, as discussed later on we want to limit this situations by properly setting parameters related to the self-adaptation process.

The concept of the parameter ε update process is presented in Figure 3.9. At the beginning, the initial value of the parameter ε is provided. The process counts K_R, K_C and K_P for the update period $T_{\varepsilon U}$. After that the *SNC* is calculated for the update period:

$$SNC(mT_{\varepsilon U})_{V_i} = \frac{K_{RmV_i}}{K_{CmV_i} K_{PmV_i}} \tag{3.4}$$

where $m = \{1, 2, \ldots\}$ and K_{RmV_i}, K_{CmV_i} and K_{PmV_i} are measured in the time interval [$(m-1)\ T\varepsilon U,\ m\ T\varepsilon U$). The obtained value is compared to the maximum threshold SNC_{max} and the minimum threshold SNC_{min}. Based on the comparison outcomes, the process provides a decision on increasing, decreasing or not changing the parameter ε. At the end of the process the K_R, K_C and K_P counters are reset.

The reasoning is straightforward. If network conditions are in favour of coding and there are few retransmissions, the ε is increased. In this manner, even more coding opportunities are found. If the decision was incorrect, the number of packet retransmission will increase and ε will be decreased again in the next iteration of self-adaptation. Such continuous changes of ε need to be taken as regular outgoing packets are also used for testing the media and for finding new opportunities.

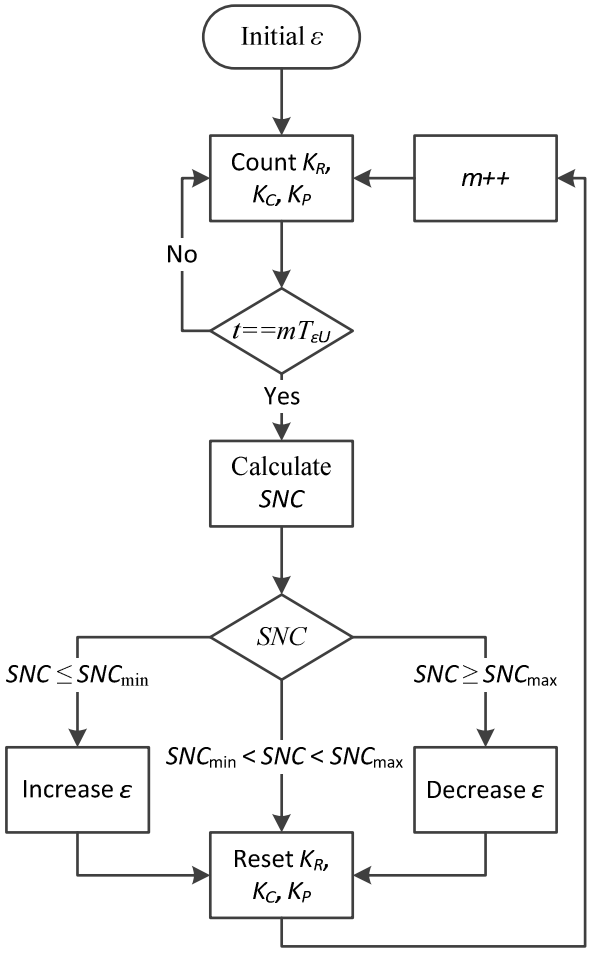


Figure 3.9: A flow chart for adapting the tolerance angle.

The whole detailed algorithm for the parameter ε update is presented in Algorithm 3.1 In addition to described basic process, we also take into account the number of processed native packets K_P, which needs to be greater than K_{PMIN} to apply any changes. We also increase the parameter ε, if for a longer period no coding opportunities are found (*CounterTurnsWithoutCoding* counts the times the algorithm has been run, and no packet encoding has been recorded). If the *No_Coding_Threshold* is reached, parameter ε is increased. On the nodes where no coding opportunities are found chances are that we are missing them due to too low ε. Thus, by increasing the ε value, we increase the probability that coding algorithm will encode packets.

We also pay attention to the sampling period for K_R, K_C and K_P, since retransmissions due to a modified parameter ε will first appear only after the scheduled time for retransmissions, T_{RET}. The sample period until $t + T_{RET}$ will record the previous state of the parameter ε. As already defined, parameter ε has to be greater than or equal to *0*. Furthermore, we foresee that the parameter ε cannot be larger than ε_{max}. The functions *increaseEpsilon*(ε, $\Delta\varepsilon$, ε_{max}) and *decreaseEpsilon*(ε, $\Delta\varepsilon$) take that into account when increasing or decreasing ε with the step $\Delta\varepsilon$. The function *scheduleUpdate*(T) schedules an event at time T that starts the procedure for an ε update.

Algorithm 3.1: Update tolerance angle ε.

```
1: if (KP > KPMIN) do
2:     if (KC > 0) do
3:         if (ε increased in the last update) do
4:             CounterTurnsWithoutCoding = 0;
5:             scheduleUpdate(t + TRET)
6:         else do
7:             scheduleUpdate(t + TεU)
8:             if (SNC ≤ SNCmin) do
9:                 ε = increaseEpsilon(ε, Δε, εmax)
10:            else if (SNC ≥ SNCmax) do
11:                ε = decreaseEpsilon(ε, Δε)
12:            end if
13:        end if
14:    else do
15:        scheduleUpdate(t + TεU)
16:       if (CounterTurnsWithoutCoding > No_Coding_Threshold) do
17:            ε = increaseEpsilon(ε, Δε, εmax)
18:        else do
19:            CounterTurnsWithoutCoding ++
20:        end if
21:    end if
22: else do
23:    scheduleUpdate(t + TεU)
24: end if
```

4 Methodology for Performance Evaluation

Analysis of network coding is difficult to perform in real environment due to its complexity and high dependency of other network and traffic parameters. Performance evaluation focused on the observation of the influence of changing parameters that can only be done in the controlled environment, either by using simulation models or analytical studies. In this work we selected simulations for the evaluation and analysis purposes as it allows analysis on non-trivial networks, and explore the scalability of the procedures. Furthermore, the use of simulations (instead of models for instance) allows to analyse many aspects of network coding (e.g. parameters, topology).

4.1 Simulation Model

Simulation model was developed using OPNET Modeler 17.1 and fully relies on custom developed models. The work was presented in [42-44].

The simulation model is divided into five functional layers as presented in Figure 4.1. The traffic management is responsible for creating the network load and it also provides an end point for packets that have reached their destination.

The network layer takes care of routing the packets through the network. The Link layer assures reliable packet delivery. Network topology layer holds information in the current situation on the network. The network coding layer is the core of our simulation model and is responsible for coding multiple packets into one packet. For the purposes of evaluation statistics is also collected in each functional layer.

4.1.1 Traffic management

The traffic management is responsible for loading the network with traffic. It allows evaluation of quality of service by measuring end-to-end delay, packet delivery rate, the number of lost packets, throughput, the number of hops the traffic travelled, etc.

Traffic distribution has a big impact on the network coding performance and in this work we also try to highlight how this affects the performance results. Hence, the module supports different traffic distributions. Traffic can be distributed throughout the entire network, where all nodes generate traffic and all nodes are destinations and traffic intensity is set at each node individually. More unique cases can be studied through traffic matrix usage, where each element in the matrix represents the traffic intensity and the row and column elements correspond to the source and destination of traffic. With the term distant traffic flows we refer to the flows that make the largest number of hops between their source and destination. This traffic distribution is useful for opportunistic network coding, since only packets that make at least two hops between their source and destination can get encoded.

When coding packets of different sizes, the smaller packets are padded with zeros, thus not fully seizing the full potential of coding. Furthermore, the overhead that we are dealing with is less notable when larger packets are used. To evaluate the influence of this parameter, the simulation model supports different packet size settings such as: average size, distribution, etc., or sampled distribution can be imported through the vector. Values

in the vector represent packet sizes. The more frequently the packet size appears in the vector, the higher the chance of selecting that size for a generated packet.

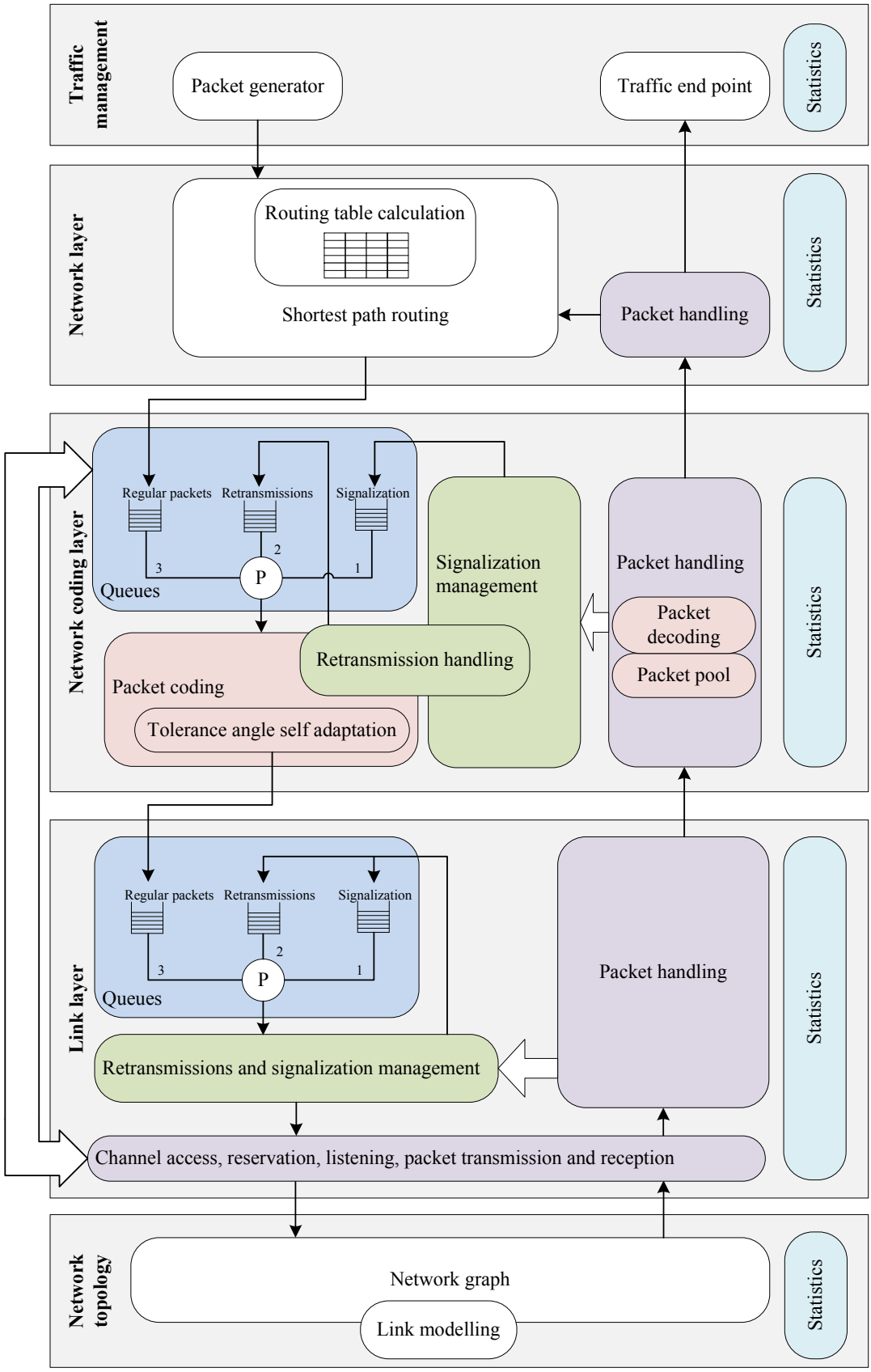


Figure 4.1: Simulation model architecture.

4.1.2 The network layer

Routing module routes packets through the network. Looking at the packet's destination and given current location the module sets the packet's next hop or sends the packet to the traffic management.

Routing tables are calculated using Dijkstra's algorithm taking into account distances between nodes, ETX metrics or hop count. In addition, our group also developed the routing algorithms that consider network coding. The ANT-based algorithm takes into account local conditions in the network, looking to reroute the traffic to create new coding opportunities [45]. The CANCAR (Congestion-Avoidance Network Coding-Aware Routing) routing approach distributes the traffic through the network differently, taking into account current traffic conditions in the bottlenecks and the number of coding opportunities seized over particular traffic flow [41].

4.1.3 The network coding layer

The main goal of the network coding layer according to the selected algorithm. In the module two coding algorithms are already supported, namely BON and the well-known COPE algorithm [6], however further algorithms can be added in a modular manner. The COPE algorithm is the first algorithm developed in the area of opportunistic network coding. It is typically used as a benchmark for all other algorithms. Furthermore, as also discussed in Section 1.1 in its basic principle BON best compares to COPE as other approaches look for performance boosts also with the cross-layer optimization, which is not within the scope of this work.

The network coding layer implements the functionalities laid down in Section 3 , and those that are also required in COPE. The main building blocks, their functionalities and implementation details are summarised below:

- Queues handle packets received from the higher layers (coding queues), and queues where packets wait to be sent to the layers below network coding (outgoing queues)
    - Retransmission and
    - regular packets queue are the two coding queues. All the packets that arrive from the network layer are placed in the regular outgoing queue. The retransmissions queue has higher priority and here the packets that are to be retransmitted are waiting. Both queues are divided into two subqueues, one for large and one for small packets. Coding algorithm selects the first packet according to priority and FIFO (first in first out) approach and searches for coding opportunities only within the regular outgoing queue (please note that with COPE only regular queue is used where retransmitted packets are placed at the head of this queue).
    - Signalization packets namely cumulative ACKs and reports packets (the latter are used only with COPE, see signalization management in network coding description) are all placed into signalization queue. This is the highest priority queue and packets placed here are not subdue to coding process.
- All the received packets are go through packet handling process. Encoded packets are decoded, where decoding procedure is based on the knowledge on which packets have been received. Hence all the received (and sent) packets are for decoding purposes saved into the packet pool. Due to memory limitations the packets are held in the packet pool only for a limited period of time. This

parameter can be set in terms of a number of packets and the time duration.

- Retransmissions handling is not as simple as in regular networks. We adopted cumulative acknowledgements (see Section 3.2.3), and this requires that each node has two counters for each neighbouring node, counting outgoing and incoming packets from that node for the identification purposes. Furthermore, the procedure also seeks opportunities to attach ACK messages to regular outgoing packets and the ACK is sent out as an individual packet only if none appears and time for acknowledgements runs out.

  In addition to the mentioned procedure used in BON and COPE procedures, negative acknowledgement and acknowledgement is also implemented for individual packets. The latter is also available for all packets and not only for encoded packets (in the case where broadcast or multicast mechanism is used in the link layer). These mechanisms are useful for different applications such as streaming applications.
- Signalization management handles acknowledgements and with COPE procedure also reports. As explained in [6] "the block starts with the number of reports in the packet. Each report specifies the source of the reported packets. This is followed by the IP sequence number of the last packet heard from that source, and a bit-map of recently heard packets. For example, a report of the form {*193.2.4.8, 50, 10000001*} means that the last packet this node has heard from source is packet *50*, and it has also heard packets *42* and *49* from that source but none in between".
- Packet coding is a decision making process on which packets to code together. With BON there is also a tolerance angle self-adaptation process running, as all the information required are available there.

4.1.4 The link layer

The link layer takes care of transmitting packets over the wireless medium. The resources are provisioned as in Carrier Sense Multiple Access (CSMA) and where required also Collision Detection (CD) is supported. Several transmission mechanisms are supported:

- broadcast,
- unicast,
- pseudo broadcast as already presented in Section 3.2.6
- all unicast which is an extension to pseudo broadcast.

All unicast was developed in our group as a result of studying network coding algorithms. We set all the next hop addresses into the MAC header. When all unicast is used all the nodes that are next hops for the encoded packet (and are therefore next hops for one of the native packets coded into encoded packet) are required to acknowledge successful reception of the packet.

Communication between the Network coding layer and the link layer is required. We want to keep packets in the network coding layer for as long as possible, as the number of coding opportunities increases with the time the packet is waiting in the queue. Hence, the network coding layer signals the link layer that it has packets in its queue for transmission, and only when the link layer obtained access to the wireless channel the network coding procedure is initiated.

In Figure 4.1 we represent the link layer main functionalities with the following blocks:

- In queues outgoing packets are sorted and we are dealing with three main queues,

namely regular packets queue has the lowest priority and fills with packets coming from the network coding layer. Retransmissions queue holds packets that have been lost in the transmission. Signalization queue has the highest priority and holds all the packet related to signalization.

- Retransmissions and signalization management handles acknowledgement packets and initiates retransmission when required.
- Packet handling reads headers from the received packets and if required signals the retransmissions and signalization to cancel scheduled retransmissions (if acknowledgement packet is received) or to send out acknowledgement packet (if regular packet requiring acknowledgement has been received).
- Channel access, reservation, listening, packet transmission and reception, listens to the channel and handles packet transmissions and receptions. In addition it can also reserve the channel.

4.1.5 The network topology

The network topology layer holds the information on how nodes connect to each other and what is the link quality amongst them. Furthermore, it also makes decisions (independent of other modules) on which packets are lost during transmission (i.e. dropped). Information on which nodes currently hold rights to access the channel are also available here (although access is obtained on a local level).

An important functionality of this layer is the wireless link model which mimics wireless transmission, both at the level of physical layer techniques and wireless channel. It considers packet delay due to signal propagation through wireless medium and transmission delay due to packet size and bandwidth use. Detailed description of models can be found in [46]. It also incorporates packet loss that occur in wireless packet transmissions.

4.2 Performance Metrics

Network coding is a relatively new approach, whose fundaments we are still learning. With opportunistic network coding we may claim the same, as knowledge in what kind of networks and with which traffic distributions we can expect benefits and how does the new layer affect the performance of applications with high Quality of Service demands has not yet been obtained. Depending on what we wanted to show, we adopted different metrics, and on the high level we can group them into two categories:

- Performance metrics that capture the detailed network response on the use of network coding. Corresponding set of results is shown against the network load, which is always measured over longer periods. Network load was selected as the key influence for the network coding performance as it can be measured in a straight-forward manner and there is a strong correlation between performance benefits and load.
- Performance metrics that allow higher level conclusions such as the influence of the network topology on the performance of the network. The main issue in showing such results is in defining a performance metric that is only dependant on the parameter under scope.

As the elementary metric reflecting the quantity of service we observe the network goodput (g), which is the number of useful information bits delivered by the network to a certain destination per unit of time. The goodput in our graphs is defined as a sum of all the goodput on all the network nodes at particular network load in *i-th* simulation run, i.e. $g(i)$.

We further define the gain *G(i)* in *i-th* simulation run as the relative increase of goodput obtained with network coding with respect to goodput without network coding:

$$G(i) = \frac{g_{NC}(i) - g_{no\ coding}(i)}{g_{no\ coding}(i)} 100\% \tag{4.1}$$

where gain can be observed for the whole traffic in the network or just on the individual flow level. In the latter case we refer to it as flow gain.

As a typical Quality of Service (QoS) metric we measure End-to-End Delay and jitter at the application layer. Both ETE delay and jitter are measured for each particular flow separately and for all the flows. Regardless of the case, we can write the following:

$$ETE(i) = \frac{\sum_{n=1}^{K_a(i)} d_n}{K_a(i)} \tag{4.2}$$

where d_n is the *ETE* delay of the *n*-th packet and $K_a(i)$ is the number of packets received in the application layer.

Jitter is the standard deviation from true periodicity of a presumed periodic signal. We measure jitter only for flows with constant interarrival packet rates:

$$jitter(i) = \frac{1}{F(i)} \sum_{f=1}^{F(i)} \frac{1}{K_f(i)} \sum_{n=1}^{K_f(i)} (d_{nf} - ETE_f(i))^2 \tag{4.3}$$

where *F* is the number of flows, K_f is the number of packets received in the application layer belonging to the *f-th* flow, and ETE_f is *ETE* for *f-th* flow. Where we are interested in jitter of only one particular flow, this same equations is used for calculation and *F(i) = 1*.

The overhead introduced by the network coding layer in terms of individual packets is observed by the Normalised Number of Wireless Transmissions (*NNWT*), which is defined as follows:

$$NNWT = \frac{K_{nc}}{K_a} \tag{4.4}$$

K_{nc} is the number of packets (encoded, native, and signalization) received from the network coding layer by the wireless layer. *NNWT* shows the number of packet transmissions needed in the link layer without its own overhead for every goodput packet. In case that all the application packets need to travel only one hop to their destination, no network coding is to take place, and with BON the *NNWT* value is going to be exactly one. In more general cases where packets travel several hops to their destinations the *NNWT* is approximately the same as the sum of the average hop count traffic distribution distance, and the overhead and network coding retransmitted packets per application packet.

The efficiency of coding algorithms is also evaluated through the number of individual overhead packets (i.e. overhead that is attached to regular packets as headers is not taken into account in this measure).

The number of packet retransmissions in the network coding layer K_R is a counter for the retransmitted native packets and indicates how well the packets were encoded via network coding algorithms. The higher the number of retransmissions, the higher the possibility that incorrect coding decisions were made and the receiving nodes were not able to decode the received encoded packets.

We also consider Jain's index [47] which is primarily used for identifying underutilized channels or fairness in load distribution. We use it to identify underutilised streams.

$$J(i) = \frac{(\sum_{f=1}^{F} g_f(i))^2}{F \sum_{f=1}^{F} g_f(i)^2} \tag{4.5}$$

where F is the number of flows in the network and g_f is the goodput over the f-th stream.

In the results where we show the dependency of gain on other parameters (e.g. queue length, packet length, topology, etc.), we depict the *MaxGain*, which is calculated as the average of the *3* highest gain values obtained among results for a given set of parameter values. We first sort vector *G(i)* where $i = \{1, 2, ..., I\}$ and I is the number of simulation runs from the highest to lowest value:

$$G_s = sort\{G(1), G(2), \quad \ldots, G(I)\} \tag{4.6}$$

and then the *MaxGain* is:

$$MaxGain = \frac{1}{3}\,(G_s(1) + G_s(2) + G_s(3)) \tag{4.7}$$

The metric is also explained using example in the Section 6.1 .

4.3 Simulations Settings

4.3.1 Network coding settings

Performance evaluation was carried out to investigate three aspects of opportunistic network coding. Firstly, we are interested in the performance of the proposed BON procedure. Secondly, we investigated the influence of some fundamental characteristics of the network and traffic load on the opportunistic network coding performance. And thirdly, we were interested how the users perceive the introduction of such coding to the network by studying QoS parameters and fairness.

Each investigated characteristic was evaluated in a separate scenario. To this end two opportunistic network coding procedures, COPE [6] and BON, as well as the reference scenario where network coding was not used (no coding), were considered in the evaluation. Purposely developed simulation model, presented in Section 4.1 was used. Each

simulation run took *190 s*. The results were collected between the *90th* and the *150th* second to observe the steady-state conditions only. Each scenario was evaluated using *30* different network loads. For each network load we made four simulation runs, using different seeds.

The COPE and BON parameters remained fixed in all the scenarios, unless otherwise stated, and are summarised in Table 4.1. The COPE parameters were set to the values as used in the test-bed presented in [6]. The BON parameters that are common to the COPE procedures were set to the same values. Both, BON and COPE used up to two retransmissions in the network coding layer. Both procedures stored packets in the packet pool for *30 s* after the packet was received for the first time. BON and COPE sent out cumulative acknowledgements at least every *0.5 s*. The network coding layer packet retransmissions were scheduled at *0.6 s* after the initial packet transmission. If the opportunity arose, cumulative acknowledgments were attached to the regular outgoing packets. COPE reports were sent out at least every *0.5 s*. When possible, the report packets were also attached to the regular outgoing packets or to the cumulative acknowledgements. All the parameters for both coding algorithms are summarised in Table 4.1.

4.3.2 Traffic, topology and variable parameters settings

In the origin nodes, the UDP-like traffic was generated. Traffic intensity was regulated through packet interarrival time and packet lengths. We used several different constant packet lengths (*2, 3, 4, 5, 6, 7, 8, 9*, and *10 kbits*) and variable packet length between *360* and *12000 bits*. The packet size distribution had two peaks and followed the measured internet traffic, as presented in [48]. Interarrival time between packets was constant or calculated using the exponential distribution. Different number of traffic flows was generated between selected nodes in the network. All the flows were selected in pairs (i.e. from node V_x to V_y and from node V_y to V_x) where each node was selected only once. We selected n node pairs from the lowest to the highest hop-count distance, thus obtaining long flows where packets had several opportunities to get encoded on the path. In the case of the "*all2all*" traffic flows distribution all the nodes generated traffic flows and sent them to all the nodes in the network resulting in $F = N_N(N_N - 1)$ traffic flows, where N_N is the number of nodes in the observed topology. The symmetry between the two flows in the load pair was set through traffic symmetry ratio.

All nodes used the same channel with the capacity of *2 Mbit/s*. The packet delivery probability between the nodes changed dynamically throughout the simulations and throughout each simulation run, and depended on the amount of traffic between the neighbouring nodes and the distance between the nodes. In simulation runs with high load the packet delivery probability was between *0.7* and *0.97*.

Table 4.1: Parameter setting in the performance evaluation.

	Parameter	Value	Parameter full name
BON	$T_{\varepsilon U}$	0.5 s	ε evaluation period
	SNC_{max}	0.002	maximum threshold for success in packet coding
	SNC_{min}	0.000005	minimum threshold for success in packet coding
	$\Delta\varepsilon$	2°	Decrease and increase step for parameter ε
	ε_{max}	45°	Maximum ε that can be used in the algorithm
	$\varepsilon_{initial}$	30°	Initial ε value
	No_Coding_Threshold	5	Maximum number of evaluations performed where no coding was detected
	K_{PMIN}	100	Minimum number of native packets processed to perform the evaluation
COPE	T_{report}	0.5 s	Longest duration at which reports are sent out
	G	0.8	Threshold that defines the probability that each of the next hops will be able to decode the encoded packet
Common	K_{ret}	2	Number of packet retransmissions in the network coding layer
	T_{pool}	30 s	The period of packet storing in the packet pool
	T_{ret}	0.6 s	Retransmissions timeout
	T_{ACK}	0.5 s	Longest duration at which acknowledgement reports are sent out

Routing tables were calculated at the beginning of each simulation run and were updated every *30 s*. Dijkstra's algorithm was used for the route calculation, taking into account the ETX metric.

BON performance and also influence of the parameters on the network coding performance was studied in different topologies to also show that performance improvement does not depend on optimal parameter settings. We used two node distributions with a different number of nodes scattered over the selected area. Several node transmission (Tx) ranges were used in order to study different network topologies with the same node distributions. Examples of the two node distributions are presented in Figure 4.2 with two different transmission ranges (i.e. shortest and longest). Several parameters that are often used in describing network topologies are summarised for all considered topologies in Table 4.2. Node transmission range indicates how far the signal from the node can be detected, which reflects in the connectivity and wireless links between nodes. Network diameter is the longest distance between two nodes in hop count over ideal links. Average hop count is the average hop count distance between all node pairs. Average number of neighbours tells with how many nodes can each node communicate directly.

We were interested in several parameters that influence the network coding performance. All the variable parameters with included reference to the section where they were used are presented in Table 4.3. In general for each case only one parameter was varied and all the other remained fixed.

Table 4.2: Network topologies characteristics.

Node distribution index	Number of nodes	Node Tx range (m)	Network diameter	Average hop count	Average number of neighbours
1	40	800	14	5,56	3,20
		1200	6	2,59	7,50
2	99	280	14	5,40	7,04
		300	14	5,00	7,65
		320	13	4,69	8,25
		340	12	4,39	9,04
		360	10	4,05	10,07
		380	9	3,56	11,63
		400	9	3,38	12,27
		420	9	3,22	13,14

Table 4.3: Simulation parameters for investigated characteristic.

	Topology parameters		Nodes parameters		Traffic parameters			
Section	**Number of nodes**	**Area**	**Nodes transmission (Tx) range (m)**	**Queue length (packets)**	**Packet length (kbits)**	**Traffic symmetry ratio (%)**	**Number of traffic flows**	**Packet interarrival time distributio**
BON performance evaluation (5.1)	40	4 km x 4 km	800, 1200	100	10	100	2, 'all2all'	exponential
Influence of tolerance angle on BON	40, 99	4 km x 4 km, 2 km x 2 km	800, 1200, 280, 300, 320, 340, 360, 380, 400	100	10	100	'all2all'	exponential
Topology design (6.1)	99	2 km x 2 km	280, 300, 320, 340, 360, 380, 400	100	10	100	'all2all'	exponential
Traffic distribution impact (6.2)	99	2 km x 2 km	300	100	10	0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100	2, 10, 20, 30, 40, 50, 60, 70, 80, 90, 'all2all'	exponential
Packet length evaluation (6.3)	99	2 km x 2 km	300	100	2, 3, 4, 5, 6, 7, 8, 9, 10, *variable*	100	'all2all'	exponential
Analysis of output queue length (6.4)	99	2 km x 2 km	300	10, 20, 50, 80, 100, 150, 200	10	100	'all2all'	exponential
QoS analysis (6.6.1)	99	2 km x 2 km	300	100	10	100	2, 20	constant
Coding fairness (6.6.2)	99	2 km x 2 km	300	100	10	100	20, 90	exponential

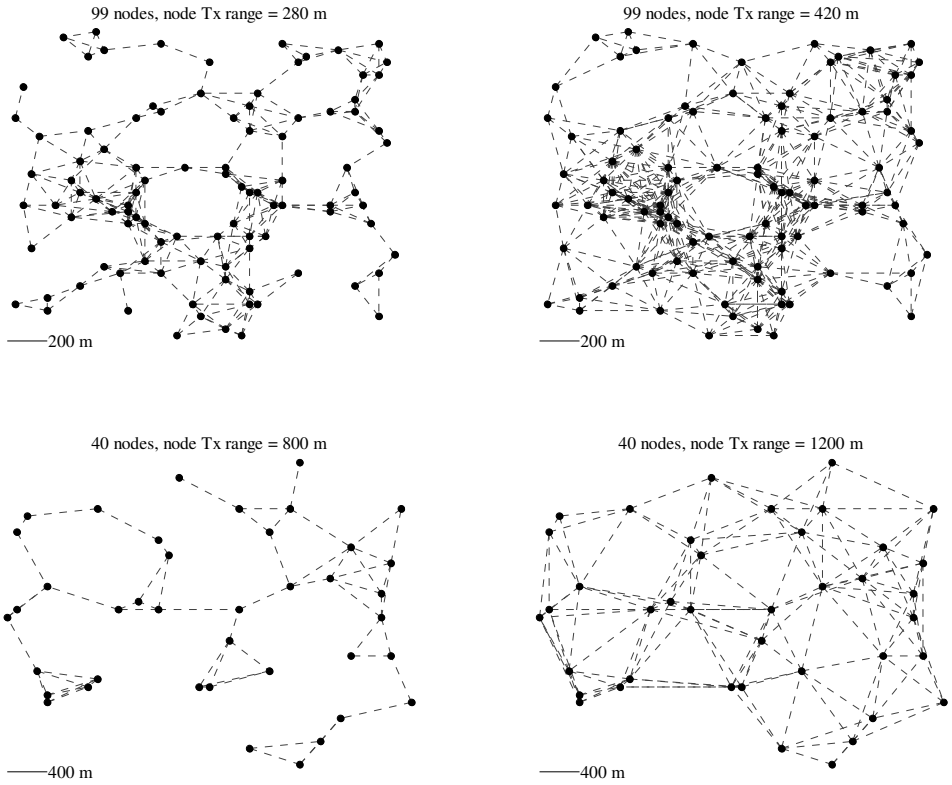


Figure 4.2: A graphical presentation of selected network topologies used in the evaluation.

5 BON Performance Evaluation

In the first set of results we analyse BON performance and compare its performance to COPE and to the reference system where coding is not used. For the evaluation purposes traffic distribution and topologies were selected in a way that the differences between BON and COPE are highlighted. Furthermore, we analyse the impact of the tolerance angle on the BON performance. Here we stress out the importance of the self-adaptation and further show what are the expected benefits of searching for coding opportunities on a wider scale.

5.1 BON Procedure Performance

In this subsection we use two topologies for the evaluation purposes. Both have *40* nodes distributed within the observed area and nodes have transmission range *800* and *1200* m respectively. All other parameters were set according to Table 4.2. We selected '*all2all*' traffic distribution as we consider it the most critical for BON performance evaluation, as traffic is not structured and self-adaptation process for the tolerance angle is thoroughly tested.

For the two selected topologies we observe the goodput in Figure 5.1. Evidentially, the proposed BON procedure can deliver the largest amount of traffic to its destinations in both topologies. As expected, both COPE and BON greatly improve the goodput of the network compared to the no coding case. The relative goodput improvement in comparison to the no coding case is also shown in Figure 5.2. Both coding procedures do not impose any degradation on the network performance with low loads. Under low loads, coding opportunities are scarce; on average there are few packets in the node queues, hence there are few coding opportunities. Furthermore, in the no coding case the system is able to deliver the entire load to its destination and differences compared to network coding are hard to detect. As the load increases, so do the coding opportunities. Due to multiple encoded packets fewer transmissions are needed to deliver the same amount of traffic to the destinations. Hence, congestion is reduced and the possibility for packet collisions reduces, thus further improving the network performance.

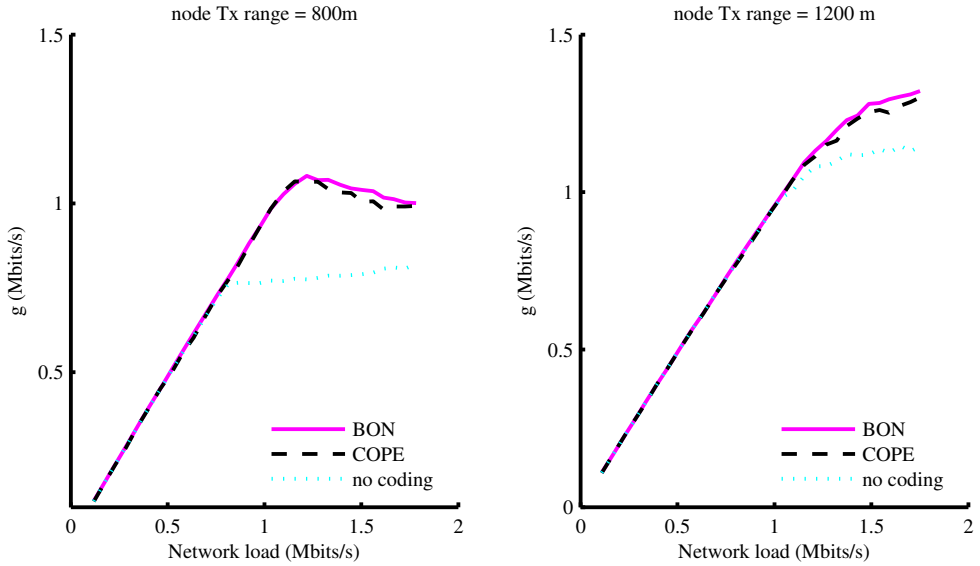


Figure 5.1: Network goodput (g) with respect to the network load for topologies with node transmission (Tx) range *800* and *1200 m*..

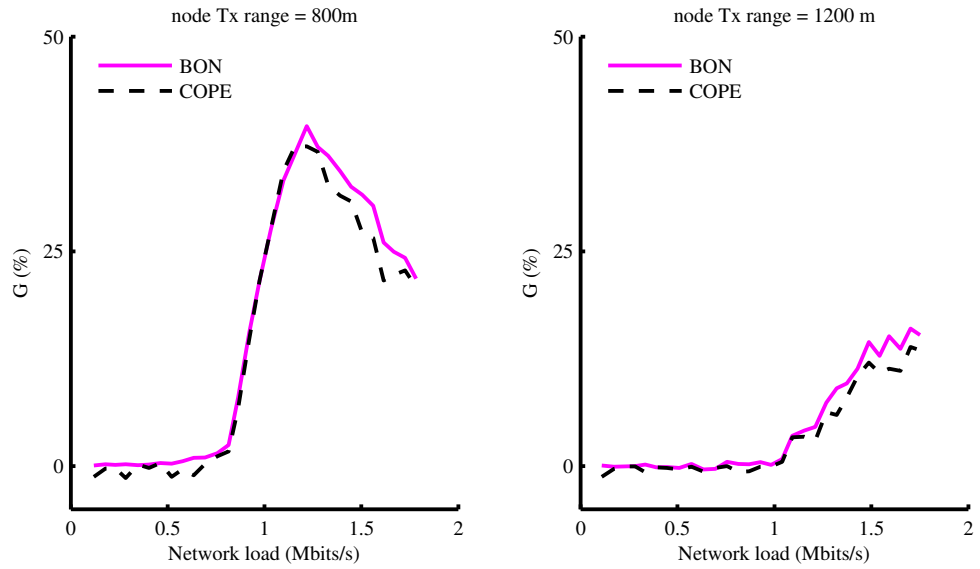


Figure 5.2: Gain (G) with respect to the network load for topologies with node transmission (Tx) range *800* and *1200 m*..

One might expect even higher goodput benefits with network coding. Compared to theoretical studies, we are dealing with non-ideal links, which results in imperfect overhearing. Secondly, there is some bandwidth loss due to network coding overhead. Thirdly, an important reason for such performance lies in the traffic distribution. With symmetrical flows we can expect packets to code on multiple hops, while with a given traffic distribution the coding mainly happens on the nodes in the centre of the network.

In Figure 5.3 we can see that in addition to increasing the goodput, BON and COPE also reduce the average ETE delay in comparison to the no coding case. At low loads the delay is approximately the same for BON, COPE and the no coding case. The differences appear with an increased load. In the no coding case, the delay increases the fastest and remains

the highest for all the observed loads. Eventhough BON can deliver higher goodput than COPE, it still manages to keep the delay slightly lower than COPE.

In simulations the parameters for network coding were set in such a way as to optimize the network goodput. With network coding the average delay was kept low, although the encoded packets that were not successfully decoded on their path at least once, arrived at their destination with delays that were much higher than average.

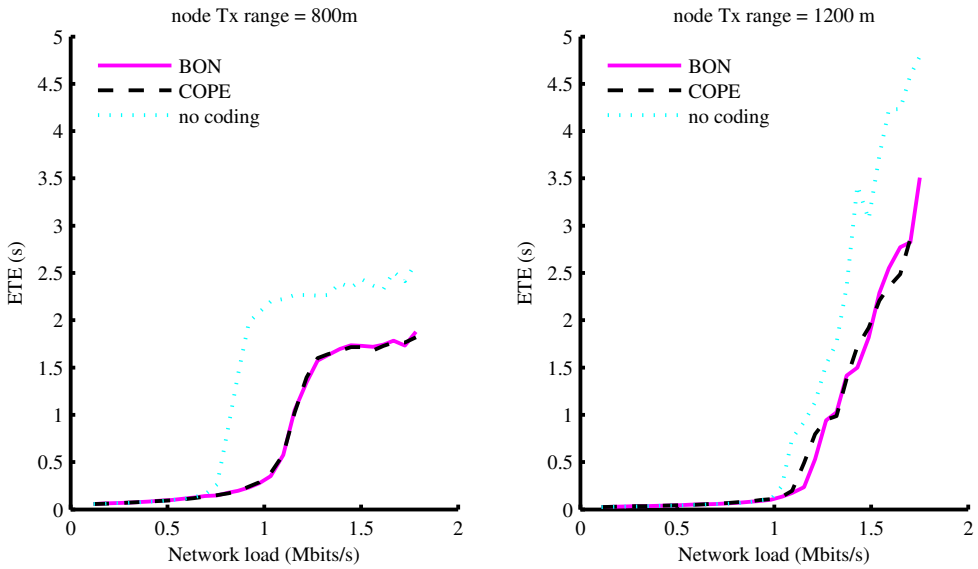


Figure 5.3: Network End-to-End delay (*ETE*) for topologies with node transmission (Tx) range *800* and *1200 m*.

Higher goodput and lower delay for BON in comparison to COPE can be explained with the help of the results for packet retransmissions, presented in Figure 5.4. We can see that BON is more successful at finding the appropriate coding opportunities than COPE, resulting in fewer packet retransmissions. Better decision making in the coding procedure lies in the self- adaptation process. In COPE, packet coding depends on the probability of packet decoding being larger than the threshold and this value is fixed. In BON the tolerance angle ε is calculated on each individual node taking into account current and past local traffic and link conditions. This means that nodes that represent network bottlenecks and perform the majority of coding have ε set to low values. Amongst plenty of coding opportunities the algorithm selects those options that are very likely to get decoded. Thus, few retransmissions are needed due to incorrect coding decisions on network bottlenecks. On the nodes that perform little coding (e.g. located at the network edges) the ε is set to high values and packets get encoded in a very opportunistic manner. Hence, retransmissions due to incorrect coding decisions are more likely to appear there. The packet high delivery rate is assured through lower ε values with retransmissions and through the queue intended for retransmitting packets as explained in Section 3.2 .

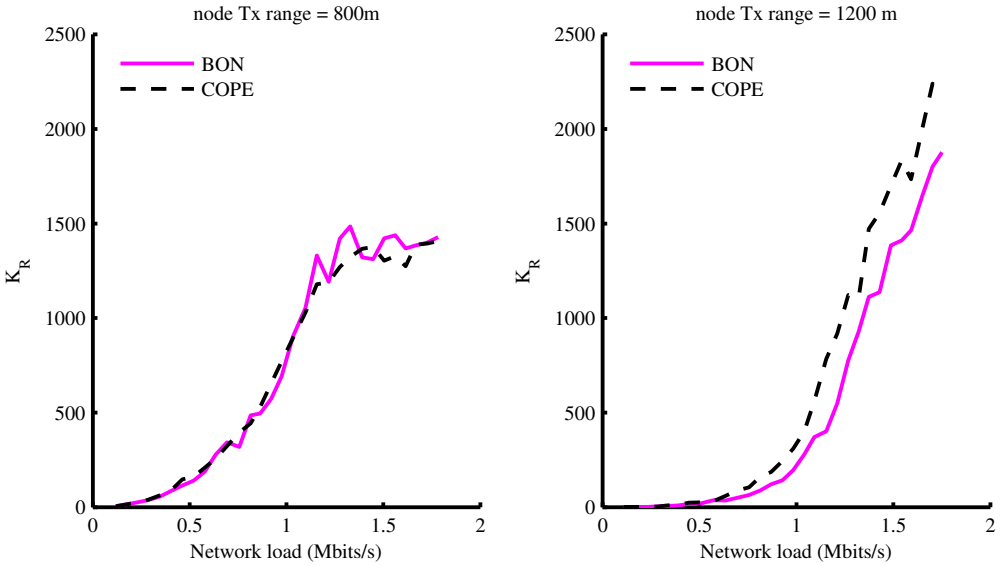


Figure 5.4: Number of packet retransmissions (K_R) in the network coding layer for topologies with node transmission (Tx) range *800* and *1200 m*.

Figure 5.5 depicts *NNWT* (see Section 4.2) to evaluate the use of overhead brought by the network coding procedures. With the lowest loads the link layer needs to transfer approximately the same number of packets for every goodput packet in BON and the no coding case: in both cases the number is approximately the same as the network average-hop count, which decreases with an increased node-transmission range. In contrast, with COPE the link layer handles more packets. These are update packets that inform the neighbouring nodes about the status of the received packets.

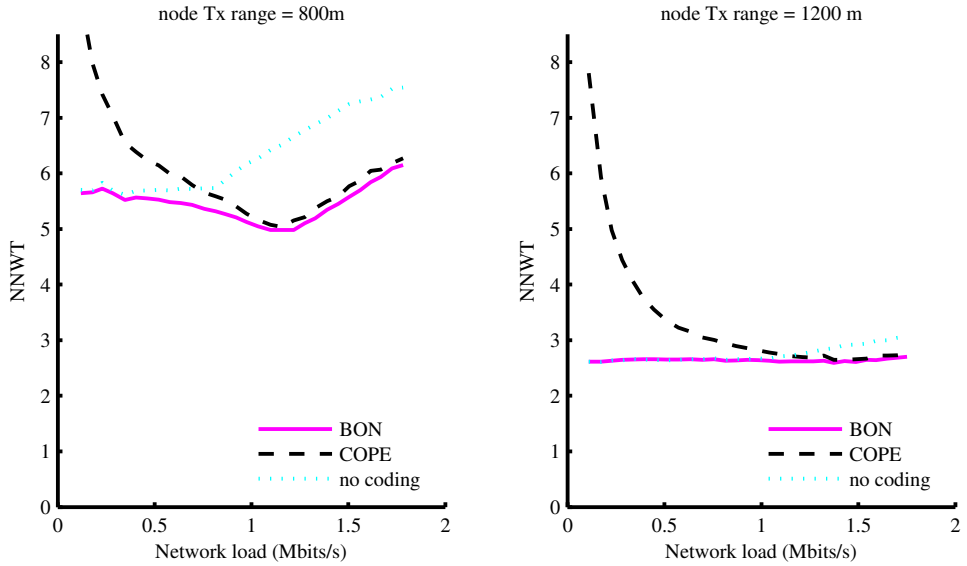


Figure 5.5: Normalised number of wireless transmissions (*NNWT*) for topologies with node transmission (Tx) range *800* and *1200 m*..

Let us illustrate the phenomena of increased number of packets with COPE at low traffic loads using a simple example. Consider a case from Figure 2.3, with ideal links and that node V_R makes decision to encode packets p_m and p_n into the encoded packet $p_m \oplus p_n$. Node V_i overhears all transmissions from node V_l; V_j overhears all transmissions from node V_k;

V_i, V_j, V_l and V_k overhear transmissions from V_R and vice versa. Packets p_m and p_n represent the only load. In this case nodes V_j and V_l each receive and successfully decode the packet $p_m \oplus p_n$ (i.e. p_m is destined to V_j and p_n is destined to V_l). Both nodes (V_j and V_l) confirm successful packet reception and decoding with the network coding layer acknowledgement messages in both procedures, namely COPE and BON. In COPE procedure, nodes (V_i, V_j, V_l, V_k and V_R) also prepare update messages informing their neighbours on the status of newly received packets. With the used mechanisms the update messages can be attached to regular outgoing packets as additional headers or as standalone packets, if there are no opportunities to attach them as headers. So in the simplest case where nodes exchange only packets p_m and p_n the overhead for COPE needed in the network coding layer would be:

- V_R transmits one update message informing that it has received p_n and p_m in a form of an additional header with the encoded packet.
- V_j and V_l confirm a successful packet ($p_n \oplus p_m$) reception and decoding to node V_R, with network coding acknowledgement messages (i.e. two acknowledgement messages are transmitted).
- V_j and V_l inform neighbours that they have new packets (p_n and p_m) in their packet pool. This information is added to the acknowledgement message as a header.
- V_i and V_k send update messages informing neighbours that they have received packets p_n and p_m (i.e. two messages in total in this case).

In contrast, BON performs only the second operation. This means that BON needs to transmit two standalone packets, while COPE transmits four messages in this particular case. While the number of acknowledgement messages depends only on the number of the encoded packets, the number of update packets depends on the number of the nodes overhearing the transmission (i.e. the number of update packets is equal to the number of neighbour nodes).

In low load conditions, the situation is similar as in the described example, hence the *NNWT* value is high. Still, the update packets do not affect the results in terms of goodput or delay as the overall network load is low and the network can easily handle the additional load.

For a given traffic distribution (i.e. '*all2all*') with a load increase the value of *NNWT* for COPE drastically decreases. Update packets can be attached to regular outgoing packets, as there is a sufficient number of opportunities generated by all nodes generating traffic, and the decrease in goodput only due to additional overhead would be hardly notable. Depending on the topology, the *NNWT* for BON decreases or remains the same with the load increase. Encoded packets reduce the required number of packet transmissions, while the acknowledgement messages can, in most of the cases, be attached to regular outgoing packets. *NNWT* for the no coding increases constantly after the load reaches the threshold value, where the first nodes start dropping packets due to full queues. This is because fewer application packets are delivered to the final destinations. With high traffic loads, when several nodes have full queues, *NNWT* also increases for BON and COPE for the same reasons as with the no coding case, but it remains lower than in the no coding case for all the loads. Eventhough BON is able to deliver more packets to its final destination, it keeps *NNWT* lower or at approximately the same level as COPE.

The influence of the COPE additional overhead on the performance can also be analysed in scenarios with fewer traffic flows. In Figure 5.6 we show the gain for BON and COPE in two topologies where there are only two symmetric traffic flows. COPE produces lower gains primarily because nodes that overhear transmissions but do not forward the packets need to create their own report packets, which take the common wireless channel. Furthermore, for the coding process they do not provide any useful information.

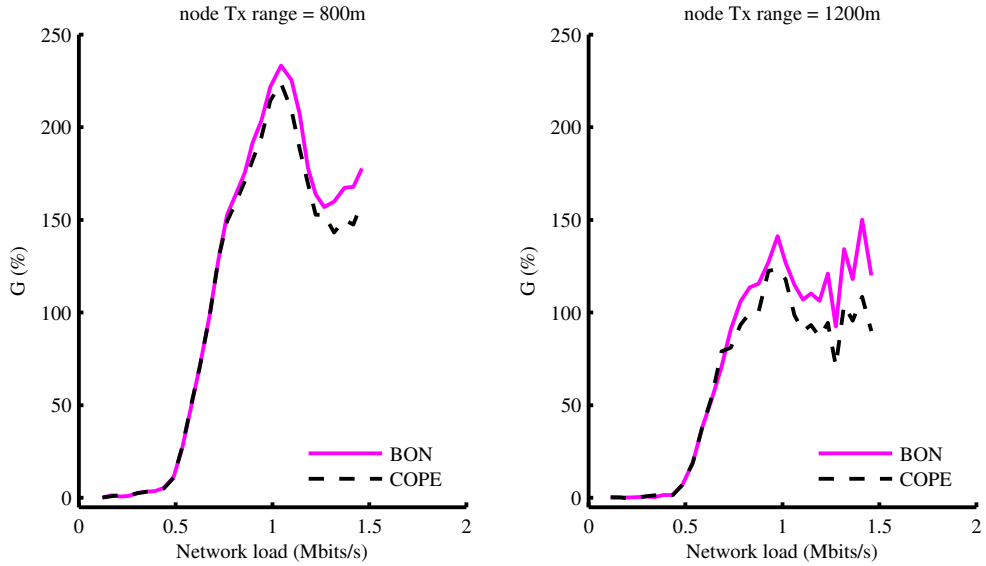


Figure 5.6: Gain (*G*) for t topologies with node transmission (Tx) range *800* and *1200 m* loaded with only two flows.

5.2 Influence of Tolerance Angle on BON Performance

Incorrectly set tolerance angle can result in degradation of service, or in the worst case in the loss of entire flow. In the following we provide an insight on how the setting of the tolerance angle influences the BON performance. We examined all the topologies used for the evaluation purposes in this work. We used *'all2all'* traffic distribution and constant packet size (*10 kbits*). The tolerance angle ε was fixed in each simulation run, and we compared results to the case when tolerance angle is self-adaptable. The results are presented in Figure 5.7 and Table 5.1.

The results show that the ε setting has a big impact on the performance, although gains in comparison to no coding scenario can still be expected. We can also see that for a given traffic distribution the best gains in the analysed topologies were obtained for ε ranging from 15° to 40°. The gains reached with self-adaptation are slightly lower than with the best fixed settings, but, the usability advantages speak for the adoption of self-adaptation algorithm. Furthermore, in highly dynamic wireless environment the adaptation to current topology conditions is necessary, as we also indirectly show in the following sections.

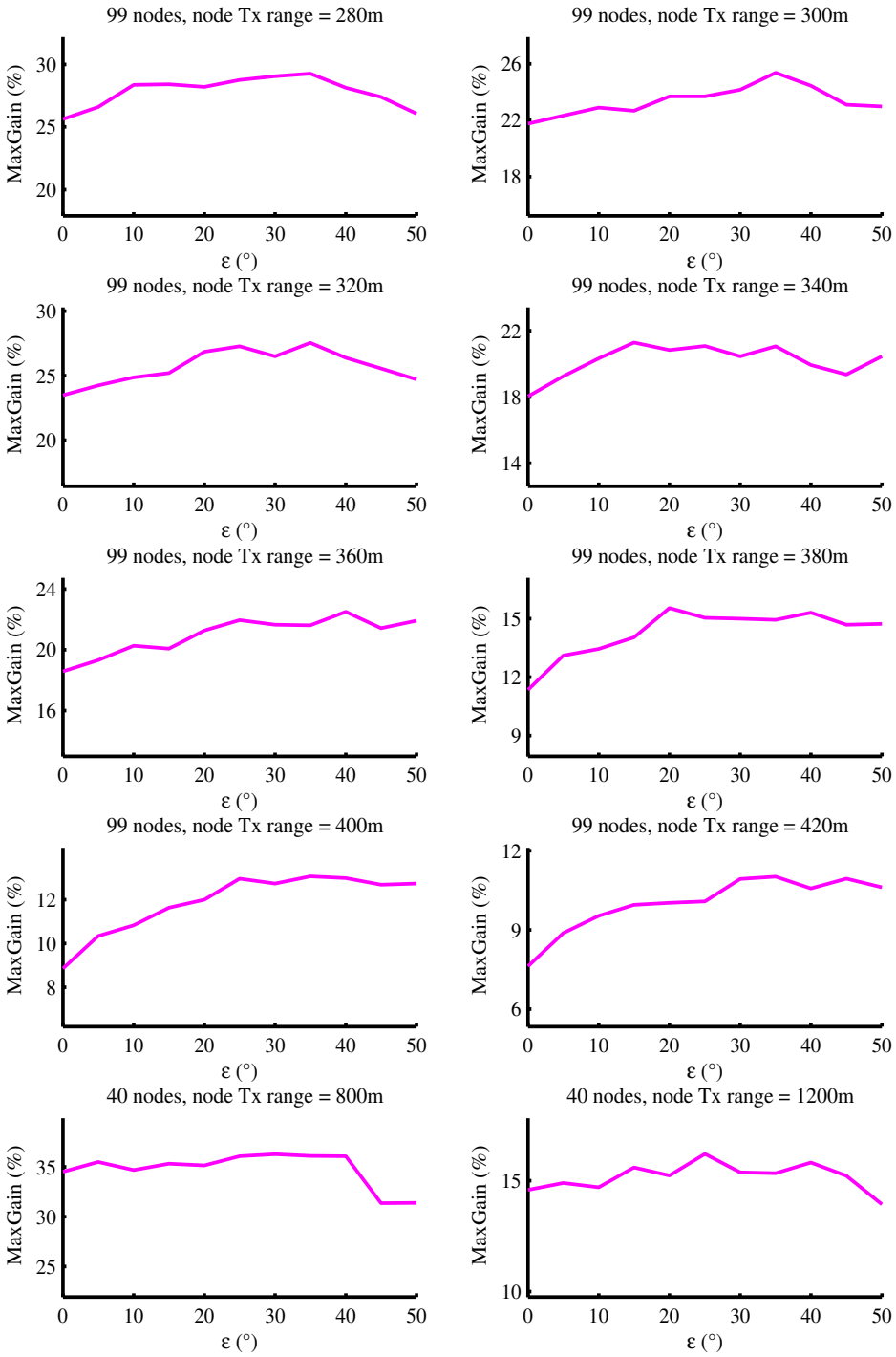


Figure 5.7: *MaxGain* in dependency of epsilon (ε) for all evaluated topologies used in this thesis.

Table 5.1: Best fixed ε settings for analysed topologies.

Number of nodes	Nodes transmission (Tx) range (m)	Best ε (°)	Highest recorded *MaxGain (%)*	*MaxGain* obtained with adaptable ε *(%)*
99	280	35	29.3	28.8
99	300	35	25.4	24.1
99	320	35	27.6	27.1
99	340	15	21.3	20.8
99	360	40	22.5	22.3
99	380	20	15.5	14.8
99	400	35	13.1	12.2
99	420	35	11.0	10.2
40	800	30	36.3	35.9
40	1200	25	16.2	15.5

5.3 Summary of the BON Performance

BON is self-adaptive, general-purpose network coding procedure applicable in static wireless mesh networks, such as metropolitan Wi-Fi networks, or wireless sensor networks. It is routing independent and it introduces very little overhead to the network, thus using less radio resources and consequently less energy. For nodes without any coding opportunities or generally in networks with low loads, the overhead is the same as in the reference scenario without network coding.

The BON decisions, i.e., which packets to encode together, are based solely on guesswork, based on bearing of the packet. This means that nodes do not record which packets their neighbours have already received, which further reduces the complexity. Analysis showed that COPE finds more coding opportunities in complex networks and traffic distributions, but by keeping the overhead low, BON still outperforms COPE.

BON dynamically adapts on one hand to changes in traffic patterns and on the other hand to continuously changing wireless link conditions by looking for a balance in decoding success and number of coding opportunities. Routing protocol defines paths through the network and BON looks for coding opportunities only in routes selected by the routing (the next and previous hop that are taken into account in coding process are set by the routing protocol). This means that BON avoids low quality links as good as the accompanying routing protocol.

6 Network and Traffic Design Aspects in Network Coding Enabled Wireless Networks

The core of this section is the analysis and the evaluation of the influence of different network and traffic characteristics on the performance efficiency of the network coding in wireless mesh networks. The investigated characteristics include (i) network topology (ii) traffic distribution, (iii) packet length, and (iv) packet queue length.

We are not optimising the parameters of any particular network coding algorithm or one of the support mechanisms, but we are rather investigating the impact of different network characteristics on the performance of network coding procedures. Thus, our interest goes to traffic distributions and patterns, different network topologies and node properties. As these parameters depend on the design and the purpose of the network, it is of paramount importance for the developers of wireless mesh networks and applications to know what benefits they can expect from the use of opportunistic network coding.

6.1 Topology Design Impact on Network Coding

We used one synthetic node distribution of *99* nodes and changed the topology graph by increasing the node transmission range, and thus the number of nodes, which can opportunistically hear each other and communicate. Two representative topologies with the shortest (*280* m) and the longest (*420* m) considered node transmission ranges are depicted in Figure 4.2. Other parameters were set according to Table 4.3.

Figure 6.1 depicts gain with respect to the increasing network load for network topologies with node transmission range set to *280*, *340*, and *400* m. The highest network coding benefits were obtained in the topology where nodes have the shortest transmission range. This might be counterintuitive to certain extent as by larger coverage area one might think that more nodes overhear transmissions, thus resulting in more coding opportunities and higher decoding success. We wanted to present gain in dependency of node transmission range so the same results are presented in Figure 6.2 as *MaxGain* as obtained for all topologies. Coding benefits decrease with higher node transmission range. There are several reasons for that. The diameter and average hop count distance of each network decreases with increased nodes transmission range. With this there are fewer possibilities for packets to get encoded as they need fewer hops from their sources to destinations. In networks where nodes have shorter node transmission range there are typically few nodes that handle large portions of traffic. Nodes that handle a larger portion of load is where normally the majority of coding takes place. With larger node transmission range the number of possible paths between source and destination pairs increases, hence the load is more evenly distributed amongst nodes. It often happens that flows between two neighbouring nodes on one side of the network and two neighbouring nodes on the other side of the network use entirely disjoint paths, yet often still competing for the same wireless resources, but not being able to get encoded in the intermediate nodes. This can be improved for example by using network-coding aware routing which could in such a case increase the coding opportunities.

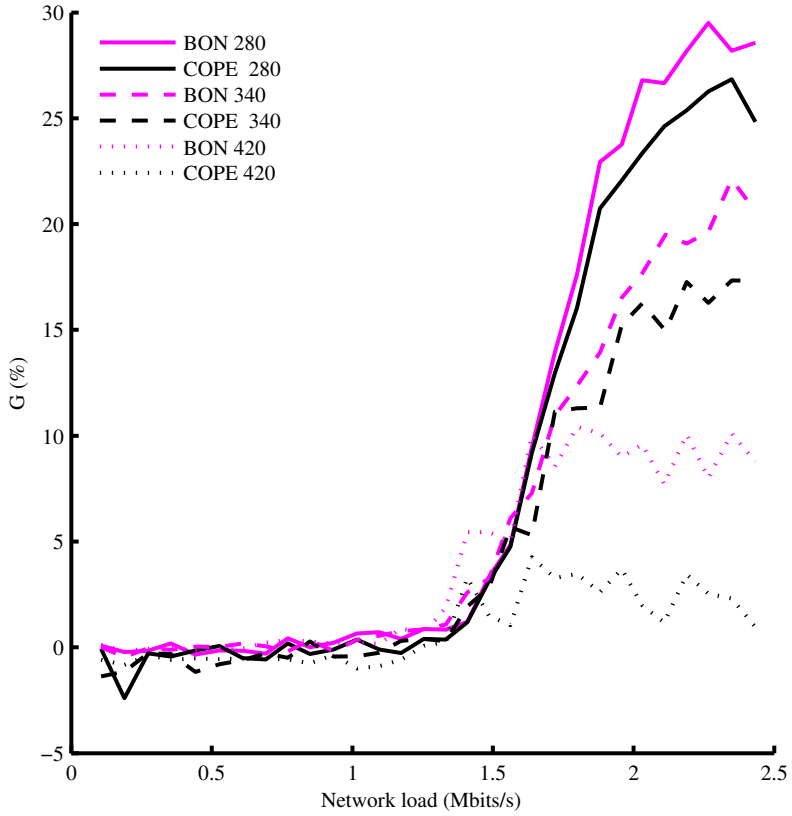


Figure 6.1: Gain (*G*) in dependency of network load for topologies with *280, 340,* and *400 m* node transmission range.

Furthermore, with higher node transmission range the competition to access the wireless channel is stiffer. Hence, the time needed to receive the network coding layer acknowledgement message increases, especially when the network has to deal with higher traffic loads. Thus, the acknowledgement messages may arrive to their destinations after the retransmission procedure was initiated. To some extent this can be monitored by adjusting (i.e. increasing) the waiting time for the network coding layer acknowledgement messages, but this might affect the streaming applications in terms of unacceptable jitter.

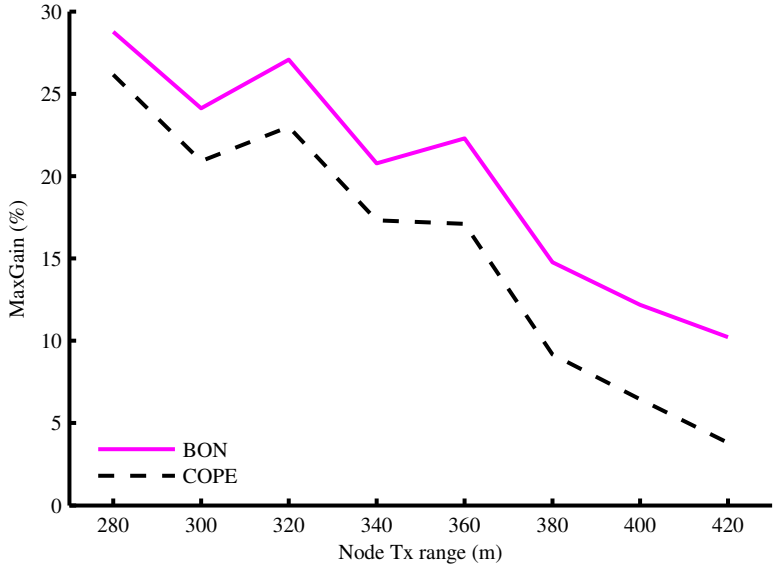


Figure 6.2: *MaxGain* in dependency of node transmission (Tx) range.

Based on the above analysis we can conclude that the goodput of networks where the node transmission range is high can be increased only for few percent using basic opportunistic network coding approaches. If the network coding is considered for the usage in a mesh network, where nodes have many neighbours, the support of network-coding-aware routing protocol should be considered, or a network coding procedure with different approach should be selected such as e.g. BEND [37].

6.2 Traffic Distribution Impact On Network Coding

Potential benefits of using network coding are strongly depending on the traffic distribution. We first examined the influence of traffic symmetry and later on the traffic flows distribution on the performance of opportunistic network coding.

For investigating the influence of traffic symmetry we deployed *10* flows between *5* node pairs and varied the traffic symmetry ratio as defined in Table 4.3 where also all other parameter settings are stated.

In Figure 6.3 we present results for the *MaxGain* in dependency of traffic symmetry ratio. As expected, the gains increase as traffic within traffic flows becomes more symmetric, essentially increasing the network coding opportunities. With the lowest ratio (i.e. traffic symmetry ratio = *0*) small gains have been noted. Since coding happens between the flows that cross their paths in opposite directions, the benefits of network coding increase with increasing traffic symmetry ratio. We can see that the network coding procedures bring best benefits when the traffic symmetry is above *60 %*.

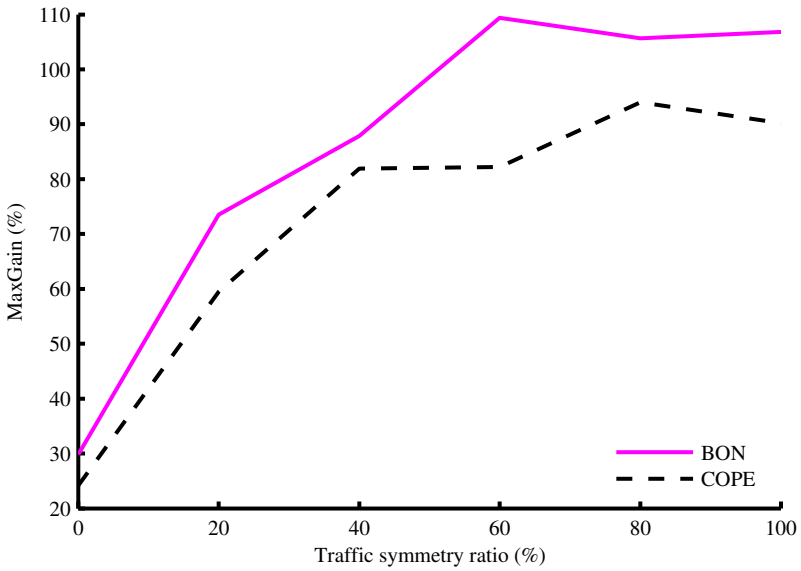


Figure 6.3: *MaxGain* in dependency of traffic symmetry ratio.

In the next set of results we evaluated the dependency of network coding benefits with respect to the number of traffic flows, i.e. traffic flows distribution. In Figure 6.4, ETE delay in dependency of goodput for *20* and *90*, traffic flows and *'all2all'* traffic distribution is presented. By increasing the number of traffic flows the network goodput increased with the unchanged ETE delay. This is expected as with the higher number of traffic flows the same overall network load is more evenly distributed across the network. Furthermore, the network coding decreases delay in comparison to no coding scenario while increasing the network goodput, but the delay is increasing due to the increased network load.

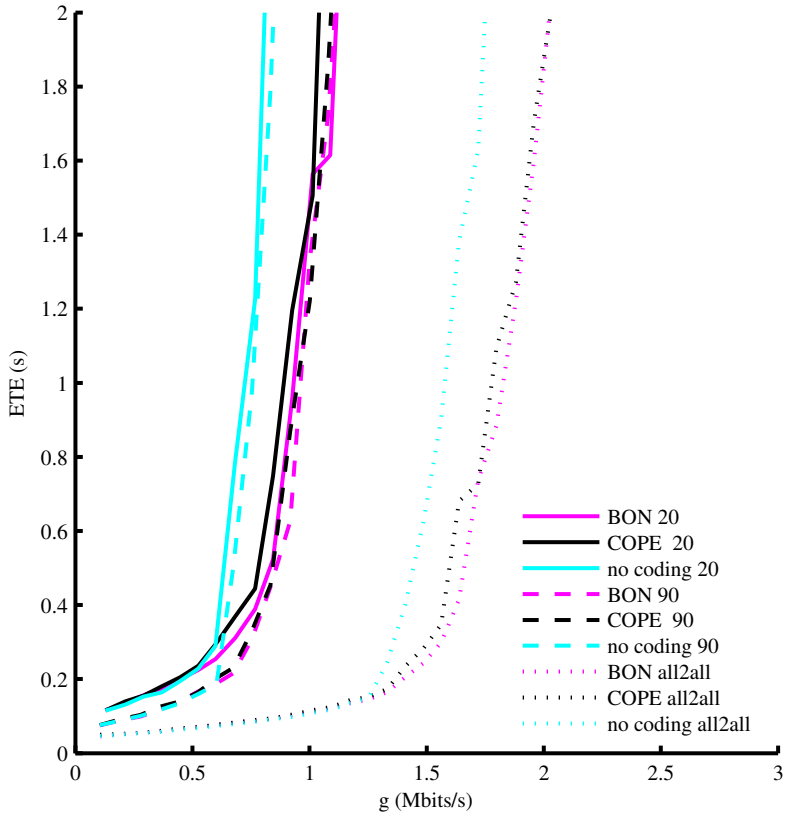


Figure 6.4: ETE delay (*ETE*) in dependency of goodput (*g*) for 20 and 90 traffic flows and *'all2all'* traffic distribution.

In Figure 6.5 *MaxGain* in dependency of the number of traffic flows is depicted. With very few flows in the network (i.e. *10*) *MaxGain* was higher than *100 %*. By increasing the number of flows also shorter flows were injected into the network and the shorter the traffic flow, the fewer coding opportunities its packets experience along the path. The lowest gains were obtained in the '*all2all*' traffic flows distribution (depicted as separate points in the graph), where an in-depth analysis also revealed that fewest coding opportunities are found. Nevertheless, in the '*all2all*' distribution of traffic flows, where network coding benefits were the lowest, the network coding gain with both procedures were still higher than *20 %*, and in addition both procedures decreased the delay significantly.

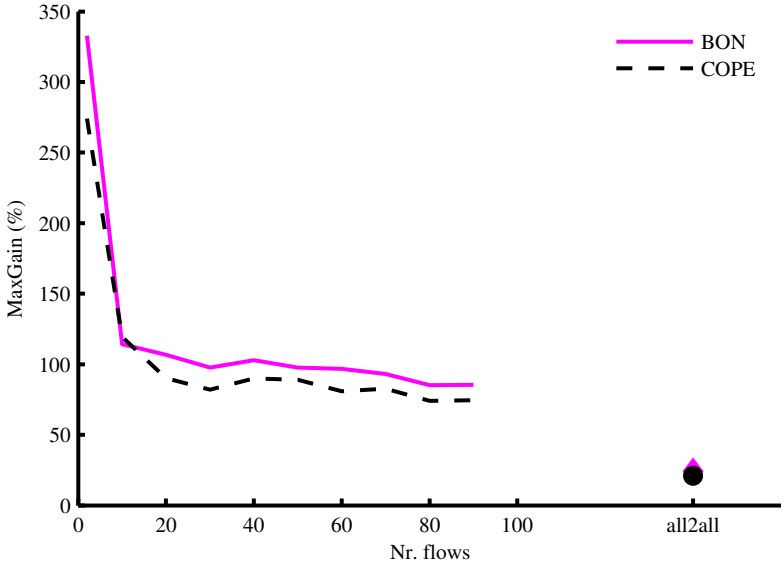


Figure 6.5: *MaxGain* in dependency of number of traffic flows.

In topologies where nodes have higher node transmission range also the signalization overhead becomes more notable. In Figure 6.6 the overhead caused by BON and COPE in terms of the number of individual packets is presented for two extreme traffic distributions i.e. *2* flows and *all2all*, and topologies with node transmission ranges *300* and *400 m*. COPE uses so called report messages with which nodes inform their neighbours which packets they have received. With the increased node transmission range there are more nodes that can overhear each other's transmissions, which means that report messages increase in numbers and in the size (each node composes reports for each of the neighbours from which it has received new, unreported) packets). Higher number of traffic flows means higher number of COPE reports as there are more nodes that can overhear packet transmission. In low traffic load conditions there are few opportunities to attach reports to regular outgoing packets, hence high number of individual report messages. By increasing the traffic load the number of report messages depends also on traffic flow distributions. In '*all2all*' traffic flow distribution the number of standalone individual messages quickly decreases, as there are plenty of regular outgoing packets on all the nodes to which report messages can be attached, hence fewer individual packets. The opposite happens in the case of only two traffic flows. There is a constant number of nodes that relay packets to which report messages can be attached. Due to compact format of report messages the number of individual messages remains almost constant regardless of the traffic load.

Using BON, the number of overhead packets increases with load. With higher loads there are more encoded packets and thus more acknowledgement messages which generate the only potential individual overhead packets in BON. Similarly as with COPE, the acknowledgement messages are attached to regular outgoing packets, if possible. In traffic distribution with *2* flows, interarrival time of packets within the same flow is low enough so that acknowledgement messages are attached to regular outgoing packets, hence there are almost no individual acknowledgement packets. It can be seen that the overhead increases also with increased transmission range. This indicates that there can be more coding opportunities found in topologies with larger transmission range. Nevertheless, the overhead produced by BON is significantly lower than by COPE.

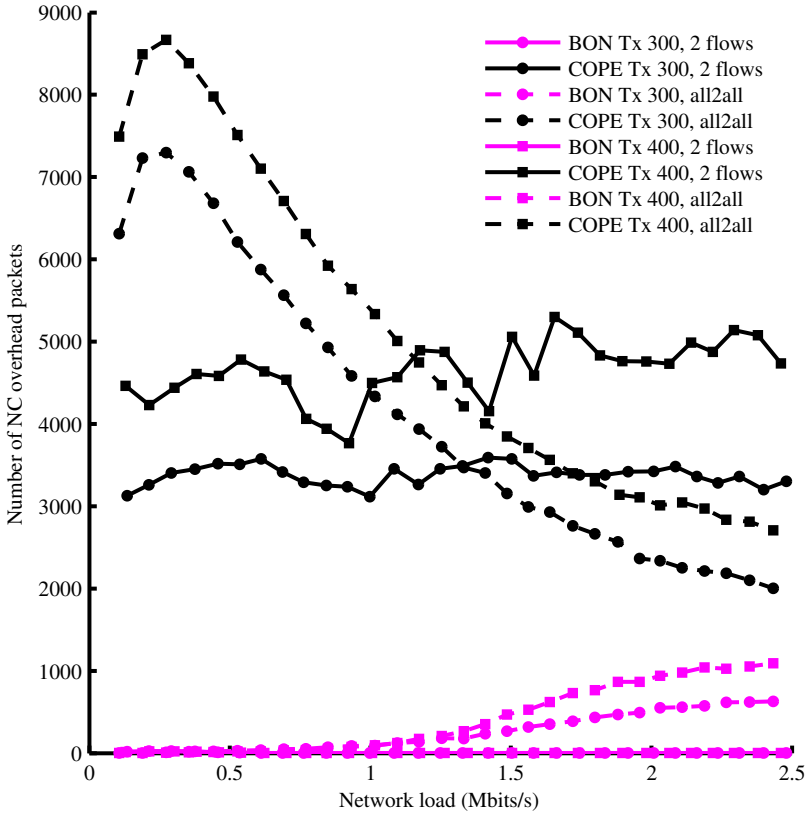


Figure 6.6: Network coding overhead in terms of the number of individual packets for *2* flows and '*all2all*' traffic distributions and topologies with node transmission (Tx) range *300* and *400 m*.

6.3 Packet Length Evaluation

The influence of the packet length on the performance of opportunistic network coding reveals the overhead caused by the network coding and how well the network coding procedures encode packets together. In this evaluation parameters were set according to specifications in Table 4.3.

In Figure 6.7 gain in dependency of network load for packets of *2000* and *5000* bits is shown (also results for variable packet length are shown, but they are discussed later on). We can see that the highest gains were obtained with BON for packet length of *2000 bits*. BON self-adaptation process is based solely on information gathered in network coding layer. The more packets the network coding layer processes the better the decision process, hence there are fewer retransmissions indicating better coding decisions.

However, with BON lowest gains are obtained with the largest packets, where, although the influence of the overhead is the lowest, but incorrect coding decisions result in high cost of retransmission. In addition, the impact on signalization is higher than when smaller packets are in use. The channel is occupied for longer periods and acknowledgement message time out can be affected, resulting in additional packet retransmissions.

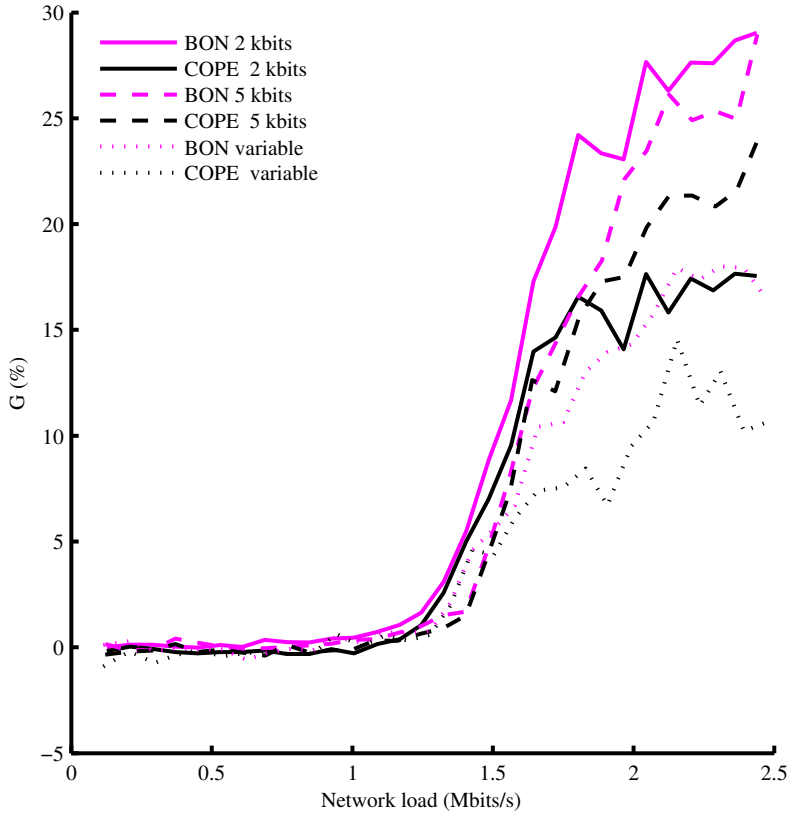


Figure 6.7: Gain (G) in dependency of network load with packets of *2000* and *5000* bits, and of variable length.

MaxGain in dependancy of packet length is presented in Figure 6.8 for BON and COPE. With BON *MaxGain* was decreasing with the increased packet length since the overhead was less important and because the network coding layer handled fewer packets. The latter resulted in poorer coding decisions as BON self-adaptation process depends also on the number of processed packets. COPE generated higher overhead than BON, hence the *MaxGain* was lower with short packets. By increasing the packet length, the gains also grew to the point where signalization did not affect results and *MaxGain* shows no dependency on the packet length size.

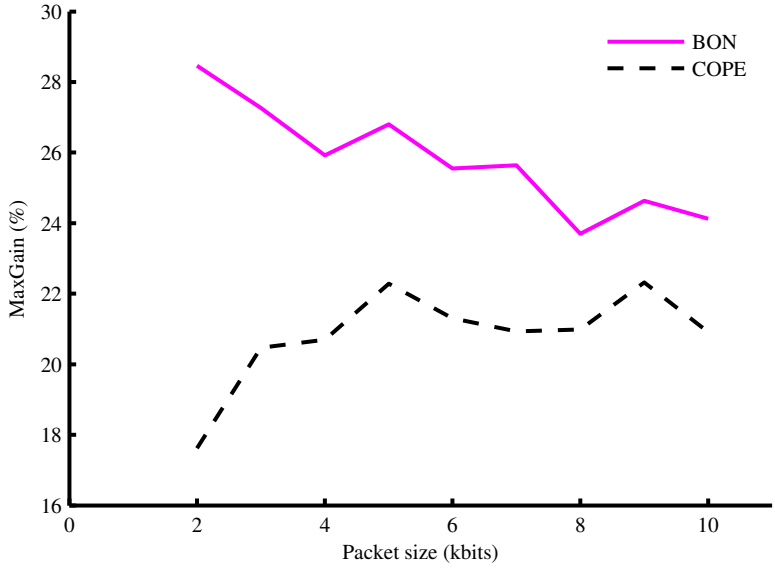


Figure 6.8: *MaxGain* in dependency of packet size.

We were also interested in the network performance considering variable packet lengths (also presented in Figure 6.7, and Figure, and noted as "variable"). Thus, additional experiments were carried out with empirical packet length distribution presented in [48] where packet lengths varied between *360* and *12000 bits*. When encoding packets with different lengths the shorter packets are padded with zeroes, thus adding irrelevant information to the payload. As expected, the results for variable packet length distribution show the lowest gains. Both coding algorithms favour coding of packet of approximately the same length, but the degrading impact of packet extension cannot be compensated.

6.4 Analysis of Output Queue Length

In this subsection we examine the influence of the output queue length on the network coding performance. The output queue length is important because COPE and BON do not deliberately delay packets in order to find more coding opportunities. Instead, both analysed algorithms search for coding opportunities only amongst packets present in the output queue. For this analysis parameters were set according to the specifications in Table 4.3.

In Figure 6.9 *MaxGain* in dependency of queue length ranging from *10* to *200* packets is shown. As expected, both network coding procedures provided the lowest gain with the shortest queue. Although one would expect that more packets in the output queue will increase coding opportunities, the results show that initially the *MaxGain* grew very fast with the queue length, but then the increase of the queue length had no effect on the performance of network coding.

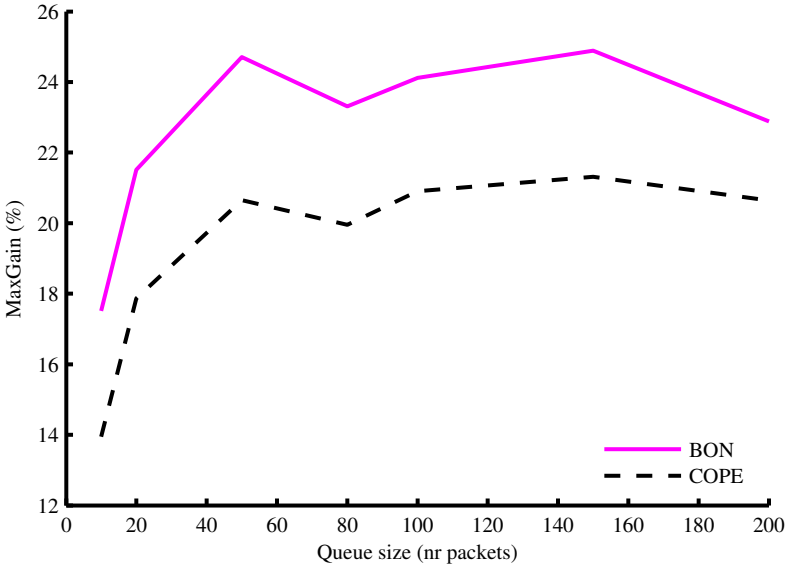


Figure 6.9: *MaxGain* in dependency of the output queue length.

6.5 Design recommendations for network coding-supported wireless networks

Two opportunistic network coding algorithms COPE and BON were used for the evaluation purposes. Both procedures are similar in several aspects, but they differ in their core, i.e. the coding decision algorithm, which is in case of COPE using exact distributed information about coding opportunities, whereas in case of BON it is based only on local information available in a node. Both procedures show similar tendencies, thus observations and conclusions gained can be generalised to other similar types of algorithms. The performance of both coding algorithms could be further improved for specific scenarios by optimising the particular parameters or using additional support mechanisms. However, in this work we focused on the impact of the network and traffic characteristics on the efficiency of network coding procedures, thus the network coding parameters remained fixed throughout the evaluation study.

In the following we summarise the observed and discussed design principles for wireless networks using network coding and their applications.

Traffic distribution and foremost traffic symmetry have a significant impact on the performance of opportunistic network coding. In the case where only one direction flows (completely asymmetric traffic) are considered, opportunistic network coding will not improve the overall network performance. By increasing the symmetry factor the coding gains will increase, reaching the maximum benefits for completely symmetric traffic load. Furthermore, the highest gains are obtained when there is relatively small number of flows in the network. By increasing the number of flows the network coding gains decrease.

The average length of packets in the network is also an important parameter for the selection of the network coding procedure. The overhead induced by the network coding procedure has bigger influence in the network with smaller packets. When packets of the same length are encoded the gains obtained are higher than when packets vary in length. In addition, the enlarging of the output queues improves the gains, but only to certain level, which also depends on the number of flows in the network. Hence, the output queue length should primarily be adjusted to handle the expected peak network loads.

When planning the network topology, additional transmission power and thus increased nodes transmission range in network coding enabled networks will not necessarily improve the network performance. The results show that with the increased nodes' transmission range the benefits of opportunistic network coding fade, requiring cautious planning of the transmission power.

6.6 End user perception of Network coding

6.6.1 Quality of service parameters

One particularly important consideration when introducing network coding in wireless communication systems is its potential effect on the quality of service. BON and COPE use asynchronous acknowledgement mechanisms to confirm successful transmission and decoding of encoded packets. Furthermore, cumulative acknowledgements are used to avoid high overhead and to increase the goodput. This reflects in relatively long time periods used for packet retransmission scheduling (i.e. in our case *0.6 s*). Such long waiting period is expected to affect streaming applications in terms of delay and jitter (i.e. QoS). We analysed this by observing a traffic flow between the two nodes with the highest hop-count distance under two traffic flow distributions. In the considered network topology with node transmission range set to *300 m* these two nodes were *14* hops apart. We first investigated the case of two symmetric traffic flows between the two nodes. In such traffic distribution the coding happened only between flows travelling in the opposite directions and retransmissions in the network coding layer were only due to packet loss and never due to inability to decode the encoded packet. Next we created twenty symmetric traffic flows between ten node pairs with the highest hop-count distance but we still observed the conditions on the same node pair as before. Other parameters were set according to Table 4.3.

Figure 6.10 shows ETE delay in dependency of the increasing network load with two and twenty symmetric flows. The delay was not affected at lower loads in comparison to no coding scenario. However, with the higher loads the delay obtained in the case of network coding was notably lower than in the no coding scenario. As shown in Figure 6.11, jitter also remained lower or approximately the same as in the no coding scenario for both network coding procedures, especially in the network with only two flows. With high loads there were occasions when jitter increased, though this was in the case when the network was congested and packets were being dropped out of packet queues. An in-depth analysis also revealed that with high loads there were rare occasions when the packet travelled the complete path between two nodes without getting encoded on at least one of the hops. For the most of the cases on a *14*-hop path packets were encoded on several hops.

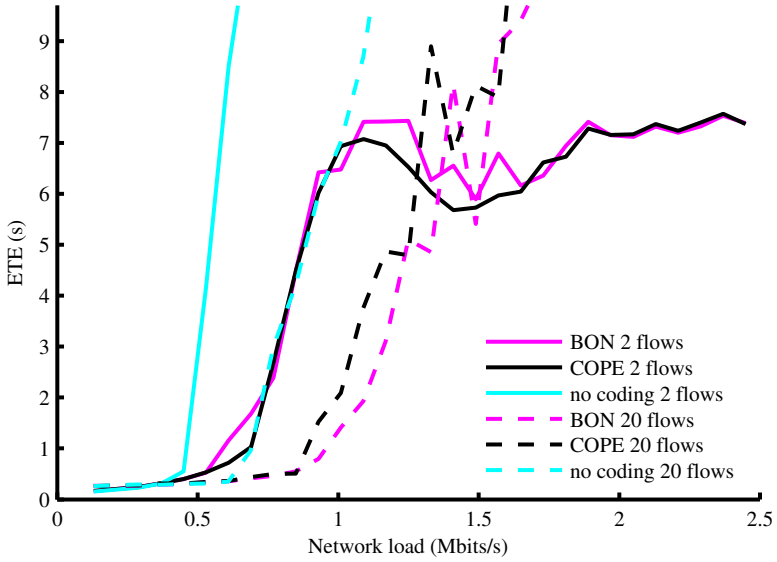


Figure 6.10: ETE delay (*ETE*) for the longest flow in dependency of network load with *2* and *20* symetric traffic flows.

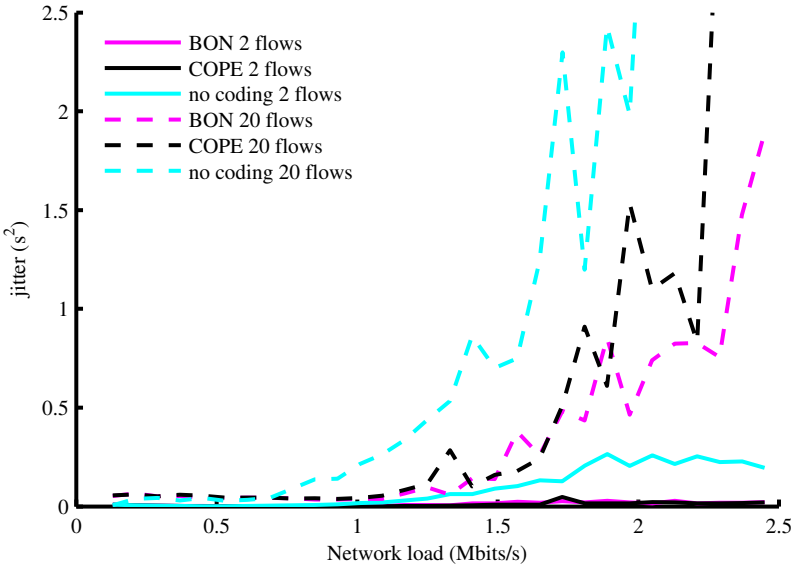


Figure 6.11: Jitter for the longest flow in dependency of network load with *2* and *20* symetric traffic flows.

Jitter for the network load with *20* symmetric flows shows similar results. Under small network load conditions jitter obtained for both BON and COPE was slightly higher than for the no coding scenario. With low number of packets in the network the impact of every encoded packet that was unsuccessfully delivered or decoded is highly notable. With the increased load jitter generally remained lower than with the no coding scenario, although there were few occurrences when jitter for the case of using network coding, especially COPE, was high. This happened when higher number of opportunistically coded packets got lost during transmissions and arrival order of packets was significantly affected.

Generally we can conclude that network coding has positive impact on jitter. The benefits of network coding with respect to the no coding scenario appear at high network loads, when jitter is significantly affected in the no coding scenario due to congestion of most loaded nodes. It is worth pointing out that the impact of network coding was observed on a large scale network using traffic flows that travel long distances. Such networks are not primarily built for streaming applications with high QoS demands, but were selected then for the presented analysis so that packets were encoded several times on their path to the destination, hence the effects of network coding would have been observed with higher reliability.

6.6.2 Analysis of Coding Fairness

Important consideration for network coding procedure is also its fairness, i.e. how different traffic flows benefit from network coding. In general traffic conditions only a small part of the native packets gets encoded, thus observing the performance of network coding for the entire network may blur the effects on the level of individual traffic flows. The aim of this section is to analyse the network coding fairness with respect to traffic flows in the network, i.e. if it improves the network performance by significantly improving some traffic flows at the expense of some others. It is desirable that the network coding algorithm improves the performance of all the traffic flows or at least that it does not degrade their performance in comparison to the no coding scenario.

We carried out a detailed analysis of individual traffic flows with respect to how successful the network was in transferring individual flows from the source to the destination. Other parameters were set according to Table 4.3.

At first let us consider Jain's index, which is primarily used for identifying underutilized channels. In our case we are establishing whether users or applications were receiving a fair share of system resources. In Figure 6.12 Jain's index for *20* flows traffic distribution is presented. Jain's index indicates that network coding procedures treat traffic at least as good as or better than when there is no coding, however it does not reveal how coding gains were distributed amongst flows.

In addition to Jain's index, we use in the following the gain of the particular traffic flow as the performance metric, so we refer to it as flow gain. The flow gain results for traffic distribution consisting of *20* traffic flows are shown in Figure 6.13. The flow gain was collected at the overall network load equal to *0.768* Mbit/s which is the maximum load at which the delay for the no coding scenario was lower than *2 s*. At such network load intensity the goodput for the no coding scenario was already affected by full queues on the nodes, i.e. the network was congested and several nodes were dropping packets due to full queues. At the same traffic load, the COPE and BON network coding procedures did not have a negative effect on any of the flows in terms of goodput, as in this case the highest observed flow gain was above *30%*.

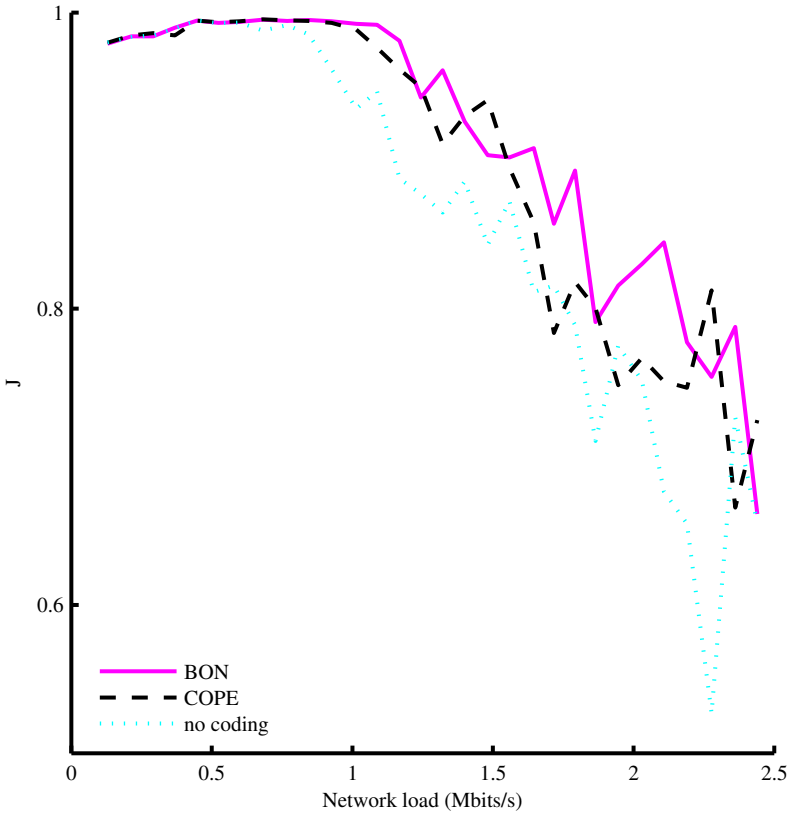


Figure 6.12: Jain's index (*J*) for the distribution with *20* traffic flows.

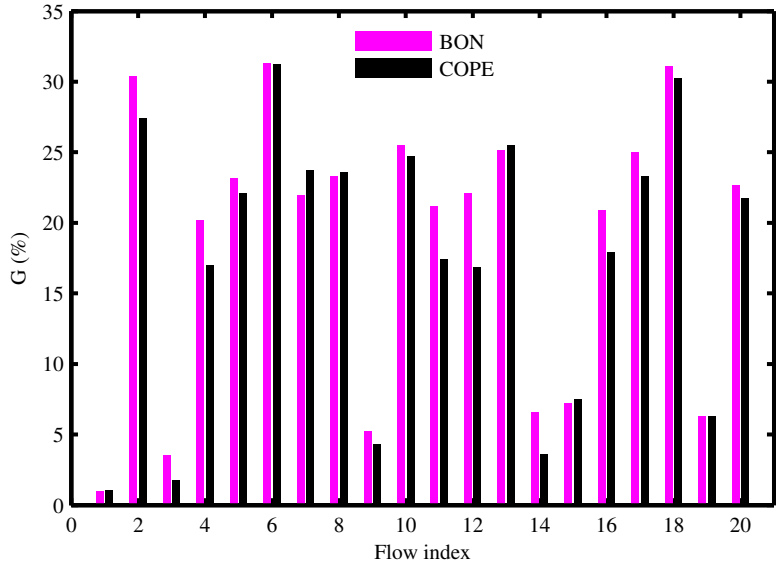


Figure 6.13: Gain (*G*) per individual flows for the traffic distribution with *20* traffic flows.

In Figure 6.14 we present the results for the case with even higher number of traffic flows (i.e. *90*). The results are presented as histograms for the overall network load equal to *1.24* Mbit/s, which was the maximum load at which the delay for COPE was lower than *3* s. Again, the network was in a state where packets on several nodes were dropped due to full output queues. On the *x*-axis we have flow gain while on the *y*-axis is the number of flows within a given flow gain interval. For example, there were *20* flows for which BON achieved flow gain between *10 %* and *40 %*. Although the goal of network coding procedures is to successfully transfer many flows with high flow gain to their destinations, it is desirable that none of the flows are degraded. Both coding procedures were able to improve the goodput of almost all the flows and there were only few flows where gains were low. Detailed analysis reveals that there were *2* out of *90* flows with BON with negative coding gain, where the lowest value was *-1.15 %*. With COPE there were *4* flows with negative gain, with the lowest value *-16.7 %*. Although we present only few representative results, the same conclusions can be drawn also for different traffic distributions and higher traffic loads.

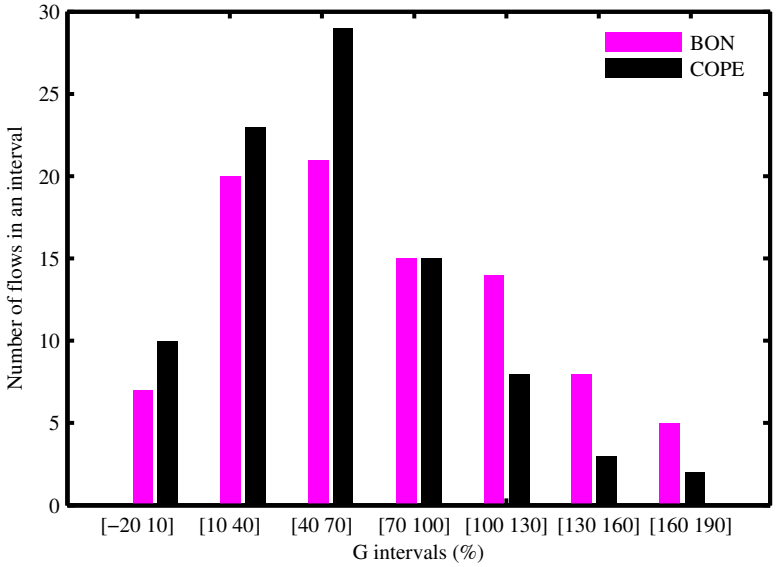


Figure 6.14: Histogram for gain (*G*) per individual flow for the traffic distribution with *90* traffic flows.

6.6.3 Summary on the end user perception of network coding

The detailed delay and jitter analysis of individual flows revealed no negative effects on QoS due to the usage of network coding procedures. Although jitter might have slightly increases (almost negligible in absolute numbers) at rare occurrences with low loads, the network performance in terms of delay and jitter with network coding in heavily loaded network significantly outperformed the no coding scenario where no network coding was used. It is worth noting that jitter and delay were analysed on a flow that travelled *14* hops to its final destination and that there were only few occasions where packets did not get encoded at least one time along the path, and still we did not detect any degradation of QoS.

Finally, the coding fairness analysis showed that both network coding procedures considered in this study improved or at least maintained the throughput of all traffic flows in the network i.e. they were not improving the overall performance at the expense of some less favoured traffic flows.

7 Conclusions

In this thesis we systematically analysed opportunistic network coding and showed what benefits in terms of throughput, delay and jitter can be achieved in different networks and applications. We showed how network coding can be implemented into current communication stack and proposed an original procedure that is completely transparent to layers above and below. Two opportunistic network coding procedures, namely COPE and proposed BON, were used for the evaluation purposes. Both procedures showed similar tendencies, thus the observations and conclusions gained can be generalised to other similar types of network coding procedures.

The proposed procedure BON is routing independent and it introduces very little overhead to the network, thus using less radio resources. For nodes without any coding opportunities or generally in networks with low loads, the overhead was the same as when coding was not used. The BON decisions, i.e., which packets to code together, are based solely on guesswork, based on bearing of the packet. This means that nodes do not record which packets their neighbours have already received, which further reduces the complexity compared to COPE.

BON dynamically adapts to changes in traffic patterns and continuously changing link conditions. A good coding algorithm does not only attempt to encode as many packets as possible, but rather makes sure that the encoded packets are very likely to be decoded. Making incorrect coding decisions leads to unnecessary retransmissions, which results in waste of the wireless resources. A balance between the number of coded packets and the decoding rate success is an important attribute of BON algorithm.

7.1 Summary of Contributions

A comprehensive analysis of the parameters influencing the performance of the opportunistic network coding, resulted in the definition of a systematic approach to constructing efficient and robust large scale network coding-enabled wireless mesh networks. We showed that the opportunistic network coding can improve the performance of different networks and supported applications. We investigated the influence of several wireless network and traffic design parameters on the performance of the opportunistic network coding in terms of quantitative measures.

Based on the analysis, we proposed a novel Bearing Opportunistic Network coding (BON) procedure which is a self-adaptive, general-purpose network coding algorithm applicable in static wireless mesh networks, such as metropolitan Wi-Fi networks or wireless sensor networks. In its operation it is routing independent and it introduces very little overhead to the network.

BON is a completely independent procedure and can be seen as an independent layer in the communication stack. Our work gives wireless network designers the flexibility to deploy network coding as the independent layer that is transparent to layers above and below.

The use of simulations (instead of mathematical models for instance) allowed us to analyse many aspects of network coding. For performance evaluation we conducted extensive simulations, using purposely built simulation model, on non-trivial networks, and

explored the scalability of the opportunistic network procedures. A new metric *MaxGain* was proposed to compare the performance of network coding algorithm(s) in different environments.

7.2 Future Work

We see the work presented in this thesis to have several possible research and development directions:

- The presented algorithm could be moved closer to the physical layer, where it would serve as a support to the two-way analog network coding. This area of research is predominantly occupied with spectrum shaping of different signals on trivial networks. BON could serve as an algorithm that helps scheduling access to the media in a way that all involved parties would be able to decode the received signal.
- The performance of network coding could be further improved with the support of routing. Work in this area has already shown promising results and has also been worked on within our group.
- The adaptation of opportunistic network coding procedures for special cases, such as support for retransmissions, also hold potential for further improvements. In cooperation with methods currently in use such as hybrid automatic repeat request retransmitting combinations of lost packets could further benefit the system performance.
- Further integration of proposed procedure into current communication stack is required. Network coding provides best results in heavily loaded (close to congestion) network condition. Still, current mechanisms running in the network often try to avoid congestion, e.g. Transmission Control Protocol (TCP), by offloading the most loaded nodes. Looking for compromises between the two approaches for the benefit of the improved network performance should be better understood.
- The procedure could also be adopted for mobile networks where wireless network routers require updating of their positions to their neighbouring nodes. The update interval of the new positions depends on the speed of mobile routers, where faster movements require updates that are more frequent. Similar to the realization of acknowledgement messages these updates could be attached to regular outgoing packets, thus keeping the additional overhead low.

Acknowledgements

Upon my arrival to the Jožef Stefan institute I was not really sure what PhD in terms of dedication and work is really about. When earning the salary on applied projects that rarely result in scientific contributions, pursuing scientific work is a constantly interrupted process. For helping me schedule different activities and finding joy in research work I am deeply indebted to my supervisor Ales Švigelj. Long hours of patient listening and insightful advising has made this work possible. Let us not forget friendly chats sometimes required to break away, and his constant raise of expectations, is what helped me push further.

I am also indebted to Mihael Mohorčič, primarily for finding time whenever required in his overbooked schedule. His scepticism, insistence on details and sharp questions forced me to rethink fundamentals all over again and think harder about the system.

I would also like to thank my colleague Erik Pertovt with whom we collaborated side by side for several years on the network coding supported routing related topics.

I would also like to express my gratitude towards my group of co-authors and collaborators, and my colleagues from the Jožef Stefan Institute for many valuable discussions and contributions.

There are no words of gratitude that can thank for endless patience, support and a stream of supporting thoughts and words that I received from my wife. Thank you!

References

[1] (2013, May). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012–2017*. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf

[2] A. Svigelj, R. Sernec, and K. Alic, "Network traffic modeling for load prediction : a user-centric approach,," *IEEE network,* vol. 29, pp. 88-96.

[3] R. Ahlswede, N. Cai, S.-y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory,* vol. 46, pp. 1204 - 1216 July 2000.

[4] T. Matsuda, T. Noguchi, and T. Takine, "Survey of Network Coding and Its Applications," *IEICE Transactions on Communications,* vol. E94-B, pp. 698-717, January 2011.

[5] S.-Y. Wang, C.-C. Lin, and Y.-C. Chang, "A rule-based inter-session network coding scheme over IEEE 802.16(d) mesh CDS-mode networks," *Computer Networks,* vol. 56, pp. 661-685 February 2012.

[6] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *IEEE/ACM Transactions on Networking,* vol. 16, pp. 497 - 510, June 2008.

[7] P. Achuthan, U. W. Rasmussen, M. Hundeboll, S. A. Rein, F. H. P. Fitzek, and G. Ertli, "Energy consumption model and measurement results for network coding-enabled IEEE 802.11 meshed wireless networks," in *IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Barcelona, Spain, 2012, pp. 286 - 291.

[8] T. Cui, L. Chen, and T. Ho, "Energy Efficient Opportunistic Network Coding for Wireless Networks," in *IEEE INFOCOM The 27th Conference on Computer Communications*, Phoenix, USA, 2008, pp. 361-365.

[9] Y. Liu, B. Guo, and C. Zhou, "Network-coded cooperative information recovery in cellular/802.11 mobile Networks," *Journal of Network and Computer Applications,* vol. In press, 2014.

[10] C. Stefanovic, P. Popovski, and D. Vukobratovic, "Frameless ALOHA Protocol for Wireless Networks," *IEEE Communications Letters,* vol. 16, pp. 2087 - 2090, December 2012.

[11] M. Susanto, F. Y. Hu, and P. Pillai, "Joint Random Linear Network Coding and Convolutional Code with Interleaving for Multihop Wireless Network," presented at the 27th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2013, Barcelona, Spain, 2013.

[12] J. Chen, L. Liu, X. Hu, and W. Tan, "Effective Retransmission in Network Coding for TCP," *Int. J. of Computers, Communications & Control,* vol. 6, pp. 53-62, March 2011.

[13] S. Haoliang, L. Lixiang, and H. Xiaohui, "A Network Coding based DTN Convergence Layer Reliable Transport Mechanism over InterPlaNetary Networks," *Int. J. of Computers, Communications & Control,* vol. 6, pp. 236-245, June 2011.

[14] K. Chia, X. Jiangb, and S. Horiguchib, "Network coding-based reliable multicast in wireless networks," *Computer Networks,* vol. 54, pp. 1823–1836, August 2010.

[15] K. Yasami, A. Razi, and A. Abedi, "Analysis of Channel Estimation Error in Physical Layer Network Coding " *IEEE Communications Letters,* vol. 15, pp. 1907-1910, October 2012.

[16] R. Alegre, N. Alagha, S. Gheorghiu, and M. Á. Vázquez-Castro, "Multicasting optimization methods for multi-beam satellite systems using network coding," presented at the 29th AIAA International Communications Satellite Systems Conference 2011, Nara, Japan, 2011.

[17] A. H. Mohammeda, B. Daia, B. Huanga, M. Azhara, G. Xua, P. Qina*, et al.*, "A survey and tutorial of wireless relay network protocols based on network coding," *Journal of Network and Computer Applications,* vol. 36, pp. 593–610, March 2013.

[18] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, "The importance of being opportunistic: Practical network coding for wireless environments," presented at the 43rd Allerton Conference on Communication, Control, and Computing, 2005.

[19] S. Chieochan and E. Hossain, "Network coding for unicast in a WiFi hotspot: Promises, challenges, and testbed implementation," *Computer Networks,* vol. 56, August 2012.
[20] D. Zeng, S. Guo, V. Leung, and J. Hu, "The Exploration of Network Coding in IEEE 802.15.4 Networks," *International Journal of Digital Multimedia Broadcasting,* vol. 2011, p. 9 pages, April 2011.
[21] R. Bruno and M. Nurchis, "Survey on diversity-based routing in wireless mesh networks: Challenges and solutions," *Computer Communications,* vol. 33, pp. 269-282 February 2010.
[22] M. A. Iqbala, B. Daia, B. Huanga, A. Hassana, and S. Yub, "Survey of network coding-aware routing protocols in wireless networks," *Journal of Network and Computer Applications,* vol. 34, pp. 1956-1970, May 2011.
[23] J. Le, J. C. S. Lui, and D.-M. Chiu, "DCAR: Distributed Coding-Aware Routing in Wireless Networks," *IEEE Transactions on Mobile Computing,* vol. 9, pp. 596 - 608, April 2010.
[24] J. Le, J. C. S. Lui, and D.-M. Chiu, "On the Performance Bounds of Practical Wireless Network Coding," *IEEE Transactions on Mobile Computing,* vol. 9, pp. 1134 - 1146, August 2010.
[25] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "CORE: a coding-aware opportunistic routing mechanism for wireless mesh networks," *IEEE Wireless Communications,* vol. 17, pp. 96 - 103, June 2010.
[26] Y. Yang, C. Zhong, Y. Sun, and J. Yang, "Network coding based reliable disjoint and braided multipath routing for sensor networks," *Journal of Network and Computer Applications,* vol. 33, pp. 422–432, July 2010.
[27] L. You, L. Ding, P. Wu, Z. Pan, H. Hu, M. Song*, et al.*, "Cross-layer optimization of wireless multihop networks with one-hop two-way network coding," *Computer Networks,* vol. 55, pp. 1747-1769 June 2011.
[28] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," presented at the 13th annual ACM international conference on Mobile computing and networking, 2007.
[29] A. Khreishah, C.-C. Wang, and N. B. Shroff, "Cross-layer optimization for wireless multihop networks with pairwise intersession network coding," *IEEE Journal on Selected Areas in Communications,* vol. 27, pp. 606-621, June 2009.
[30] M. Hundeboll, J. Ledet-Pedersen, J. Heide, M. V. Pedersen, S. A. Rein, and F. H. P. Fitzek, "CATWOMAN: Implementation and Performance Evaluation of IEEE 802.11 Based Multi-Hop Networks Using Network Coding," presented at the IEEE Vehicular Technology Conference (VTC Fall), Quebec City, Canada, 2012.
[31] G. Ertli, A. Paramanathan, S. A. Rein, D. E. L. Roetter, and F. H. P. Frank, "Network Coding in the Bidirectional Cross: A Case Study for the System Throughput and Energy," presented at the IEEE Vehicular Technology Conference (VTC Spring), Dresden,Germany, 2013.
[32] K. Chi, X. Jiang, B. Ye, and Y. Li, "Flow-oriented network coding architecture for multihop wireless networks," *Computer Networks,* vol. 55, pp. 2425–2442, July 2011.
[33] N. D. Coppi, J. Ning, G. Papageorgiou, M. Zorzi, S. V. Krishnamurthy, and T. L. Porta, "Network Coding Aware Queue Management in Multi-Rate Wireless Networks," presented at the 21st International Conference on Computer Communications and Networks (ICCCN), Munich, Germany, 2012.
[34] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, "Loss-aware network coding for unicast wireless sessions: design, implementation, and performance evaluation," in *2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2008.
[35] B. Scheuermann, W. Hu, and J. Crowcroft, "Near-optimal co-ordinated coding in wireless multihop networks," in *2007 ACM CoNEXT conference*, 2007.
[36] T. Yazane, H. Masuyama, S. Kasahara, and Y. Takahashi, "Effect of network-coding overhead on end-to-end throughput for multihop wireless networks," *Performance Evaluation,* vol. 70, pp. 14–27, January 2013.
[37] J. Zhang, Y. P. Chen, and I. Marsic, "MAC-layer proactive mixing for network coding in multi-hop wireless networks," *Computer Networks,* vol. 54, pp. 196-207, February 2010.
[38] T. Ho and D. Lun, *Network Coding: An Introduction*: Cambridge University Press, 2008.
[39] M. Junuzovic, "Network-coding-assisted retransmission scheme for video streaming applications in wireless access networks," Master thesis, Jožef Stefan International Postgraduate School, 2015.
[40] E. Pertovt, K. Alic, M. Mohorcic, and A. Svigelj, "ETX-based Metrics and Adapted Routing Algorithms for Network Coding," presented at the 20th Electrotechnical and Computer Science Conference (ERK 2011), Portorož, Slovenija, 2011.
[41] E. Pertovt, K. Alic, A. Svigelj, and M. Mohorcic, "CANCAR - Congestion-Avoidance Network Coding-Aware Routing for Wireless Mesh Networks," *submitted to Computer Communications,* 2015.
[42] K. Alic, E. Pertovt, and A. Svigelj, "Simulation environment for network coding," in *Mosharaka International Conference on Communications, Networking and Information Technology (MIC-*

CNIT2011), Dubai, UAE, 2011.

[43] K. Alic, E. Pertovt, and A. Svigelj, "Network coding simulation model in OPNET modeler," in *OPNETWORK 2012*, Washington, USA, 2012.

[44] E. Pertovt, K. Alic, A. Svigelj, and M. Mohorcic, "Implementation and performance evaluation of ant-based control routing in wireless mesh networks using OPNET modeler," presented at the OPNETWORK 2013, Washington D.C., USA, 2013.

[45] E. Pertovt, K. Alic, A. Svigelj, and M. Mohorcic, "Simulation model for ant-based control algorithm in wireless mesh networks," in *Enaindvajseta mednarodna Elektrotehniška in računalniška konference ERK 2012*, Slovenia, 2012.

[46] Pertovt Erik, "Dynamic network coding-aware routing in wireless networks" doctoral dissertation, Ljubljana, 2015.

[47] R. Jain, D.-M. Chiu, and W. Hawe. (1984, February). *A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems*. Available: http://www.cse.wustl.edu/~jain/papers/fairness.htm

[48] A. Svigelj, M. Mohorcic, G. Kandus, A. Kos, M. Pustisek, and J. Bester, "Routing in ISL Networks Considering Empirical IP Traffic," *IEEE Journal on Selected Areas in Communications,* vol. 22, February 2004.

Index of Figures

Figure 2.1: The butterfly network. 6
Figure 2.2: Example of how opportunistic network coding increases the throughput. 7
Figure 2.3: General relaying scenario in which network coding can be used. 8
Figure 3.1: BON procedure main modules. 14
Figure 3.2: Graphical presentation of a bearing. 15
Figure 3.3: Bearings for two packets travelling in the opposite direction. 16
Figure 3.4: Graphical presentation of the *vicinity* in the shape of an infinite cone. 17
Figure 3.5: Graphical presentation of general coding case for a matching packet pair. 17
Figure 3.6: Graphical presentation of a general coding case for a non matching coding pair. 18
Figure 3.7: A flow chart of the BON coding process. 20
Figure 3.8: A flow chart of the BON decoding process. 22
Figure 3.9: A flow chart for adapting the tolerance angle. 25
Figure 4.1: Simulation model architecture. 28
Figure 4.2: A graphical presentation of selected network topologies used in the evaluation. 38
Figure 5.1: Network goodput (g) with respect to the network load for topologies with node transmission (Tx) range *800* and *1200 m.*. 40
Figure 5.2: Gain (G) with respect to the network load for topologies with node transmission (Tx) range *800* and *1200 m.*. 40
Figure 5.3: Network End-to-End delay (*ETE*) for t topologies with node transmission (Tx) range *800* and *1200 m.* 41
Figure 5.4: Number of packet retransmissions (K_R) in the network coding layer for topologies with node transmission (Tx) range *800* and *200 m.* 42
Figure 5.5: Normalised number of wireless transmissions (*NNWT*) for topologies with node transmission (Tx) range *800* and *1200 m.*. ... 42
Figure 5.6: Gain (G) for t topologies with node transmission (Tx) range *800* and *1200 m* loaded with only two flows. 44
Figure 5.7: *MaxGain* in dependency of epsilon (ε) for all evaluated topologies used in this thesis. 45
Figure 6.1: Gain (G) in dependency of network load for topologies with *280, 340,* and *400 m* node transmission range. 48
Figure 6.2: *MaxGain* in dependency of node transmission (Tx) range. 49
Figure 6.3: *MaxGain* in dependency of traffic symmetry ratio. 50

Figure 6.4: ETE delay (*ETE*) in dependency of goodput (*g*) for 20 and 90 traffic flows and *'all2all'* traffic distribution. 51
Figure 6.5: *MaxGain* in dependency of number of traffic flows. 52
Figure 6.6: Network coding overhead in terms of the number of individual packets for *2* flows and '*all2all*' traffic distributions and topologies with node transmission (Tx) range *300* and *400 m*. 53
Figure 6.7: Gain (*G*) in dependency of network load with packets of *2000* and *5000* bits, and of variable length. 54
Figure 6.8: *MaxGain* in dependency of packet size. 55
Figure 6.9: *MaxGain* in dependency of the output queue length. 56
Figure 6.10: ETE delay (*ETE*) for the longest flow in dependency of network load with *2* and *20* symetric traffic flows. 58
Figure 6.11: Jitter for the longest flow in dependency of network load with *2* and *20* symetric traffic flows. 58
Figure 6.12: Jain's index (*J*) for the distribution with *20* traffic flows. 60
Figure 6.13: Gain (*G*) per individual flows for the traffic distribution with *20* traffic flows. 60
Figure 6.14: Histogram for gain (*G*) per individual flow for the traffic distribution with *90* traffic flows. 61

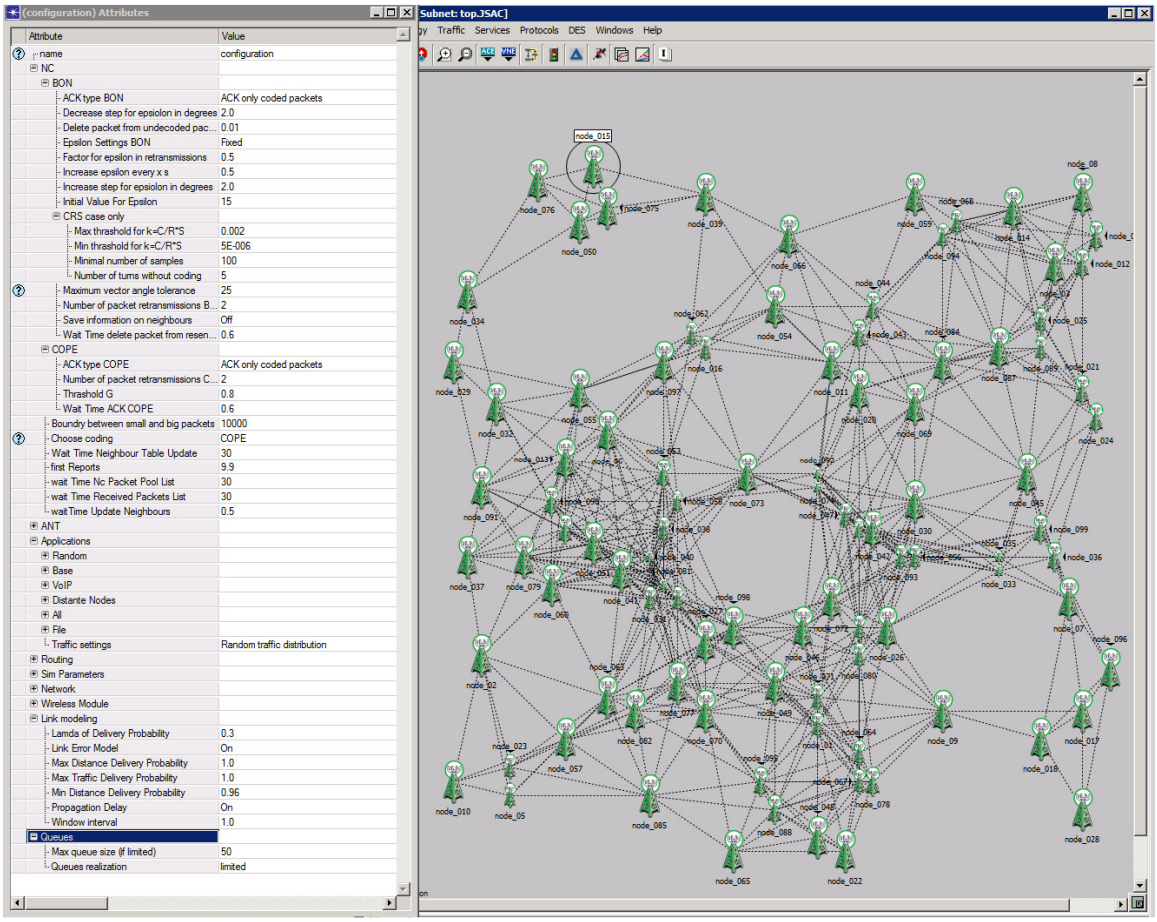
Index of Tables

Table 2.1: Definitions of terms used in this work. 11
Table 4.1: Parameter setting in the performance evaluation. 35
Table 4.2: Network topologies characteristics. 36
Table 4.3: Simulation parameters for investigated characteristic. 37
Table 5.1: Best fixed ε settings for analysed topologies. 46

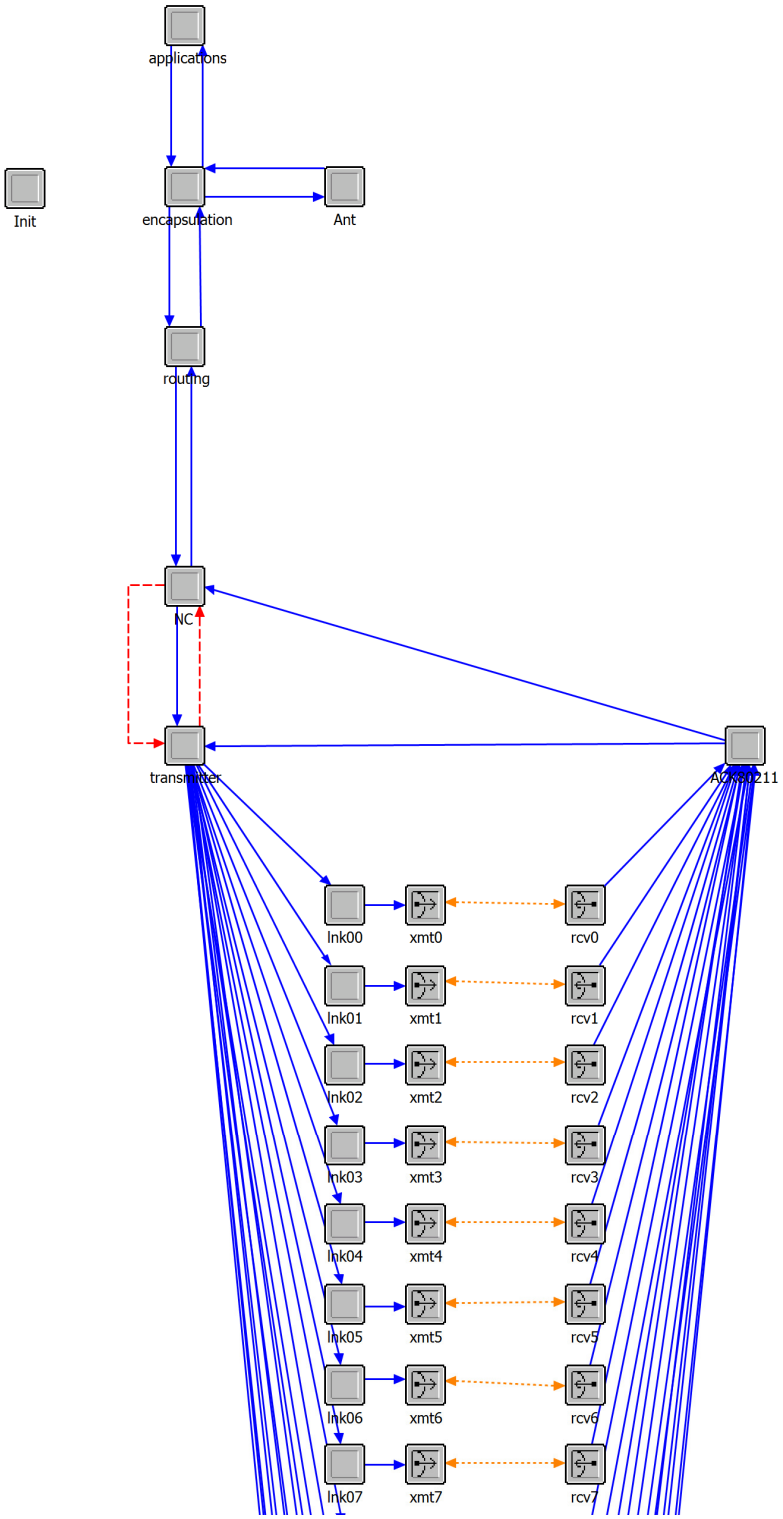
Index of Algorithms

Algorithm 3.1: Update tolerance angle ε.....26

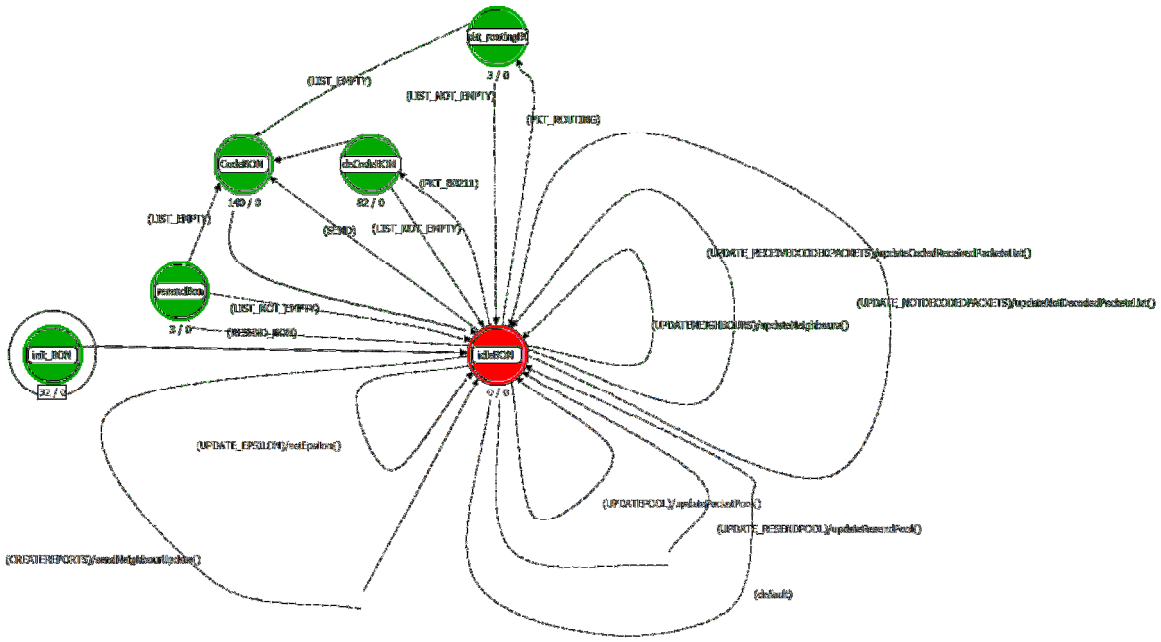
Appendix A: Simulation Model Parameter Setting Overview

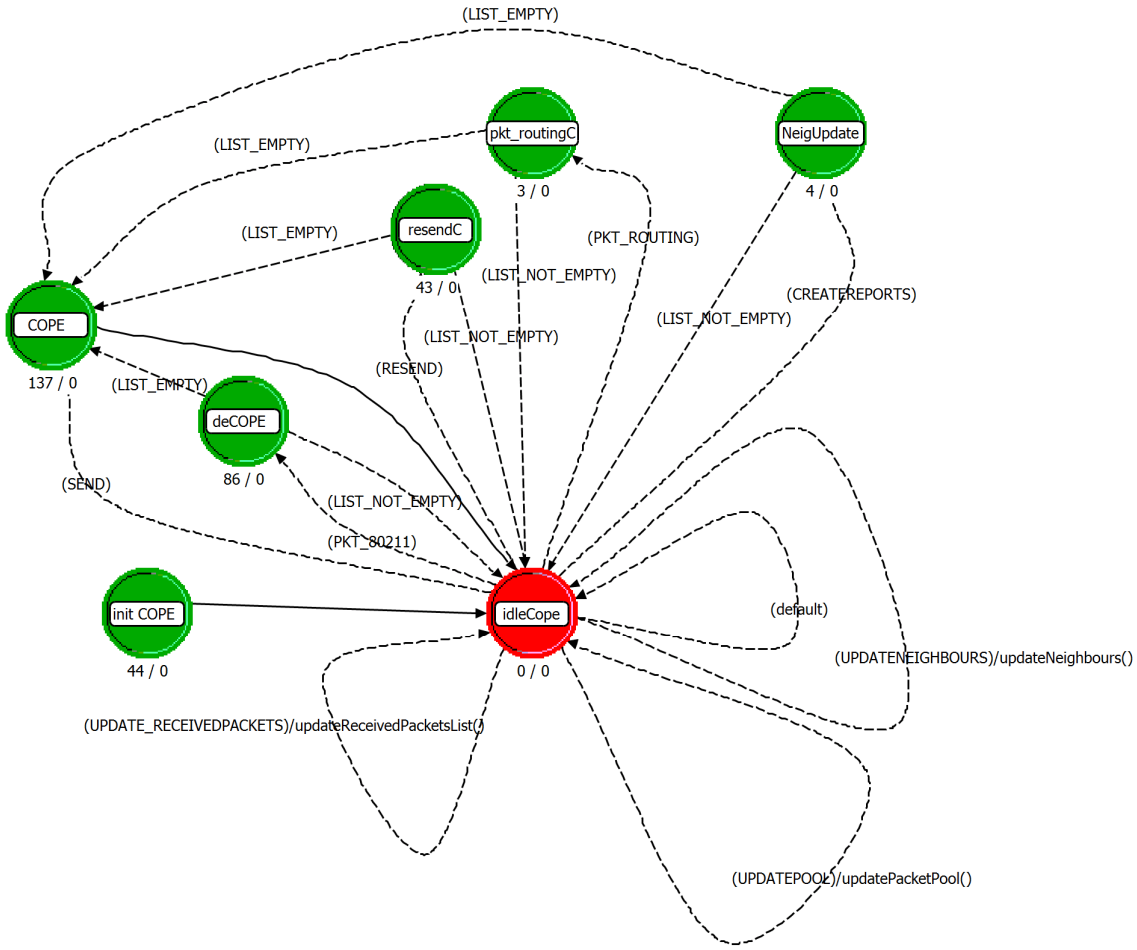


Appendix B: Node Implementation in OPNET Modeler



Appendix C: State Diagrams for BON and COPE Procedures




(LIST_EMPTY)
pkt_routingC
3 / 0
NeigUpdate
4 / 0
(LIST_EMPTY)
resendC
43 / 0
(LIST_EMPTY)
(PKT_ROUTING)
(LIST_NOT_EMPTY)
(CREATEREPORTS)
COPE
137 / 0
(LIST_NOT_EMPTY)
(LIST_NOT_EMPTY)
(RESEND)
(LIST_EMPTY)
deCOPE
86 / 0
(SEND)
(LIST_NOT_EMPTY)
(PKT_80211)
init COPE
44 / 0
idleCope
0 / 0
(default)
(UPDATENEIGHBOURS)/updateNeighbours()
(UPDATE_RECEIVEDPACKETS)/updateReceivedPacketsList()
(UPDATEPOOL)/updatePacketPool()

Bibliography

Publications related to the thesis

Journals

1. Alič, Kemal, Pertovt, Erik, Švigelj, Aleš. Bearing-opportunistic network coding. *International Journal of Computers, Communications & Control*, ISSN 1841-9836, 2015, vol. 10, no. 2, pp. 154-164.
2. Švigelj, Aleš, Sernec, Radovan, Alič, Kemal, "Network traffic modeling for load prediction : a user-centric approach," *IEEE network*, ISSN 0890-8044, 2015, vol. 29, no. 4, pp. 88-96.
3. Junuzović, Melisa, Alič, Kemal, Švigelj, Aleš. Wireless broadcast transmission scheme for reliable video-streaming service. WSEAS transactions on communications, ISSN 1109-2742, 2015, vol. 14, pp. 256-266.
4. Pertovt, Erik, Alič, Kemal, Švigelj, Aleš, Mohorčič, Mihael. Voice over internet protocol in wireless mesh networks with opportunistic network coding. International journal of communications, ISSN 1998-4480, 2013, vol. 7, no. 2, pp. 42-53.
5. Pertovt, E., Alič, K., Švigelj, A., & Mohorčič, M. (V recenziji). CANCAR - Congestion-Avoidance Network Coding-Aware Routing for Wireless Mesh Networks. Computer Communications. in submission

Conference papers

6. Junuzović, Melisa, Alič, Kemal, Švigelj, Aleš. Reliable video-streaming delivery in wireless access network using network coding. V: ZHUANG, Xiaodong (ur.). Recent researchers in applied computer science : proceedings of the 15th Conference on Applied Computer Science, (ACS '15), proceedings of the16th International Conference on Mathematics and Computers in Business and Economics (MCBE '15), proceedings of the 6th International Conference on Education and Educational Technologies (EET '15, May 20-22, 2015, Konya, Turkey, (Recent advances in computer engineering series, ISSN 1790-5109, vol. 30). [S. l.]: WSEAS = World Scientific and Engineering Academy and Society, 2015, pp. 19-25.
7. Junuzović, Melisa, Alič, Kemal, Švigelj, Aleš. Network-coding-based retransmission scheme for real time streaming applications in wireless broadcast networks. V: 6. študentska konferenca Mednarodne podiplomske šole Jožefa Stefana = 6th Jožef Stefan International Postgraduate School Students' Conference, 20.-22. 05. 2014, Ljubljana. TRDIN, Nejc (ur.), et al. Zbornik : 1. del = Proceedings : part 1. Ljubljana: Mednarodna podiplomska šola Jožefa Stefana, 2014, pp. 131-140. http://ipssc.mps.si/2014/proceedingsIPSSC2014Papers.pdf.
8. Pertovt, Erik, Alič, Kemal, Švigelj, Aleš, Mohorčič, Mihael. Implementation and performance evaluation of ant-based control routing in wireless mesh networks using OPNET modeler. V: OPNETWORK 2013: August 26-29, 2013, Washington, USA. [S. l.]: OPNET = Application and Network Performance, 2013, 7 pp.

9. Pertovt, Erik, Alič, Kemal, Švigelj, Aleš, Mohorčič, Mihael. Performance evaluation of VoIP codecs over network coding in wireless mech networks. V: LONG, Charles A. (ur.), MASTORAKIS, Nikos E. (ur.), MLADENOV, Valeri (ur.). Recent advances in electronics and communication systems: proceedings of the 2013 International Conference on Electronics and Communication Systems, (ECS 2013), July 16-19, 2013, Rhodes Island, Greece, (Recent advances in electrical engineering series, 15). [S. l.: s. n.], 2013, pp. 43-49.
10. Pertovt, Erik, Alič, Kemal, Švigelj, Aleš, Mohorčič, Mihael. Analysis of the basic topology structure for neighbourhood network coding aware routing. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik dvaindvajsete mednarodne Elektrotehniške in računalniške konference ERK 2013, 16.-18. September 2013, Portorož, Slovenija, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., ISSN 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2013, zv. A, pp. 37-40.
11. Pertovt, Erik, Alič, Kemal, Švigelj, Aleš, Mohorčič, Mihael. Network coding aware routing for performance boost in wireless-mesh-network. V: 5. študentska konferenca Mednarodne podiplomske šole Jožefa Stefana = 5th Jožef Stefan International Postgraduate School Students Conference, 23. May 2013, Ljubljana, Slovenija. TRDIN, Nejc (ur.), et al. Zbornik = Proceedings. Ljubljana: Mednarodna podiplomska šola Jožefa Stefana, 2013, pp. 168-173.
12. Alič, Kemal, Pertovt, Erik, Švigelj, Aleš. Network topologies and traffic distribution evaluation for network coding. V: Netware 2012: August 19-24, 2012, Rome, Italy. [S. l.]: IARIA, cop. 2012, pp. 20-27.
13. Alič, Kemal, Pertovt, Erik, Švigelj, Aleš. Network coding simulation model in OPNET modeler. V: OPNETWORK 2012: August 13-17 2012, Washington, USA. [S. l.]: OPNET = Application and Network Performance, 2012, pp. 1-7.
14. Alič, Kemal, Pertovt, Erik, Švigelj, Aleš. Routing-independent practical network coding algorithm. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik enaindvajsete mednarodne Elektrotehniške in računalniške konference ERK 2012, 17.-19. September 2012, Portorož, Slovenija, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., ISSN 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2012, zv. A, pp. 65-68.
15. Pertovt, Erik, Alič, Kemal, Švigelj, Aleš, Mohorčič, Mihael. Simulation model for ant-based control algorithm in wireless mesh networks. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik enaindvajsete mednarodne Elektrotehniške in računalniške konference ERK 2012, 17.-19. September 2012, Portorož, Slovenija, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., ISSN 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2012, zv. A, pp. 69-72.
16. Pertovt, Erik, Alič, Kemal, Švigelj, Aleš, Mohorčič, Mihael. Improving performance of wireless mesh networks with network coding. V: 4. študentska konferenca Mednarodne podiplomske šole Jožefa Stefana = 4th Jožef Stefan International Postgraduate School Students Conference, 25. May 2012, Ljubljana, Slovenija. PETELIN, Dejan (ur.), TAVČAR, Aleš (ur.), KALUŽA, Boštjan (ur.). Zbornik = Proceedings. Ljubljana: Mednarodna podiplomska šola Jožefa Stefana, 2012, pp. 150-156.
17. Alič, Kemal, Pertovt, Erik, Švigelj, Aleš. Simulation environment for network coding. V: MAHASNEH, Jaser (ur.). Proceedings of Mosharaka International Conference on Communications, Networking and Information Technology, (MIC-CNIT2011), 16-18 December 2011, Dubai, UAE. [S. l.: s. n.], 2011, pp. 1-6.

18. Pertovt, Erik, Alič, Kemal, Mohorčič, Mihael, Švigelj, Aleš. ETX-based metrics and adapted routing algorithms for network coding. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik dvajsete mednarodne Elektrotehniške in računalniške konference ERK 2011, 19.-21. September 2011, Portorož, Slovenija, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., ISSN 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2011, zv. A, pp. 124-127.
19. Alič, Kemal, Pertovt, Erik, Švigelj, Aleš. COPE simulation model. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik dvajsete mednarodne Elektrotehniške in računalniške konference ERK 2011, 19.-21. September 2011, Portorož, Slovenija, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., ISSN 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2011, zv. A, pp. 128-131.

Other publications

20. Istenič Starčič, Andreja, Alič, Kemal, Zajc, Matej. The Delphi Technique as a Participatory Methodology in Design, Development and Evaluation of T-Learning : E-learning Evaluation as a Social Practice. V: PSAROMILIGKOS, Y. (ur.). Evaluation in e-learning, (Advances in Operations Research). New York: Nova Science Publishers, 2012, pp. 175-188.
21. Proske, Herwing, Antić, Boris, Tsagaris, Vassilis, Alič, Kemal, Solakov, Dimcho. Integrated potential and gap analysis in the wider Balkan region. V: ANTIĆ, Boris (ur.), CRNOJEVIĆ, Vladimir (ur.). Overview of main project results, (Balkan GEO network = B.GN, FP7-265176). Novi Sad: University of Novi Sad, 2013.
22. Alič, Kemal, Švigelj, Aleš, Javornik, Tomaž. System performance evaluation of the IEEE 802.16e pico-base station in multihop-relay and mesh networks. WSEAS transactions on communications, ISSN 1109-2742, 2010, vol. 9, no. 10, pp. 615-625.
23. Alič, Kemal, Istenič Starčič, Andreja, Tasič, Jurij F., Zajc, Matej. Razvoj interaktivnih multimedijskih televizijskih učnih vsebin v okolju DVB-T = Conceptual interactive multimedia television learning-content production. Elektrotehniški vestnik, ISSN 0013-5852. [Slovenska tiskana izd.], 2009, vol. 76, no. 5, pp. 269-274.
24. Kuhar, Urban, Alič, Kemal, Medved, Anže, Novak, Roman, Švigelj, Aleš. Low-voltage-grid state estimation testbed. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik štiriindvajsete mednarodne Elektrotehniške in računalniške konference ERK 2015, 21. - 23. September 2015, Portorož, Slovenija, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., ISSN 1581-4572, 24). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2015, zv. A, pp. 225-228.
25. Alič, Kemal, Švigelj, Aleš, Javornik, Tomaž. Simulation model for multihop relay network based on mobile WiMAX standard. V: 14th WSEAS International Conference on Communications, (Part of the 14 WSEAS CSCC Multiconference), Corfu Island, Greece, July 23-25, 2010. MASTORAKIS, Nikos E. (ur.), MLADENOV, Valeri (ur.), BOJKOVIĆ, Zoran (ur.). Latest trends on communications, (Recent advances in electrical engineering, ISSN 1792-4243). [S. l.]: WSEAS, 2010, pp. 149-154.
26. Alič, Kemal, Kastelic, Marko, Škrabl, Denis, Smolnikar, Miha, Javornik, Tomaž, Mohorčič, Mihael. Versatile sensor node in environmental applications. V: EcoSense 2011 The Second International Workshop on Sensing Technologies in Agriculture, Forestry and Environment, 6-7 April 2011, Belgrade, Serbia. Workshop papers. [S. l.: s. n.], 2011, 4 pp. http://agrosense.org/images/stories/ecosense11/04%20Versatile%20Sensor%20Node%20in%20Environmental%20Applications.pdf. 24877095]

27. Alič, Kemal, Kastelic, Marko, Škrabl, Denis, Smolnikar, Miha, Javornik, Tomaž, Mohorčič, Mihael. Uporaba vsestranskega senzorskega vozlišča za podporo okoljskim aplikacijam. V: Petindvajseta delavnica o telekomunikacijah, 12. in 13. maj 2011, Brdo pri Kranju. SIMIČ, Nikolaj (ur.), et al. Internet stvari : zbornik referatov, (VITEL, ISSN 1581-6737). Ljubljana: Elektrotehniška zveza Slovenije, cop. 2011, pp. 67-72.
28. Alič, Kemal, Švigelj, Aleš, Javornik, Tomaž. 802.16e pico base station for multihop relaying. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik devetnajste mednarodne Elektrotehniške in računalniške konference ERK 2010, Portorož, Slovenija, 20.-22. September 2010, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., ISSN 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2010, zv. A, pp. 165-168.
29. Alič, Kemal, Švigelj, Aleš, Smolnikar, Miha, Mohorčič, Mihael, Škrabl, Denis, Javornik, Tomaž, Kandus, Gorazd. On remote sensing techniques in agriculture. V: The First International Workshop on Sensing Technologies in Agriculture, Forestry and Environment, 14-15 October 2009, Novi Sad, Serbia. BioSense09. [S. l.]: AgroSense, 2009, 4 pp.
30. Alič, Kemal, Zajc, Matej, Tasič, Jurij F. Delivering personalized content in narrowband return channel t-learning environment. V: International Conference Interactive Computer Aided Learning ICL, September 24- 26 2008, Villach, Austria_. AUER, Michael E. (ur.). The future of learning - globalizing in education. Kassel: University Press, cop. 2008, pp. 1-9.
31. Alič, Kemal, Zajc, Matej, Tkalčič, Marko, Burnik, Urban, Tasič, Jurij F. Development of interactive television t-learning course. V: 14th IEEE Mediterranean Electrotechnical Conference, Ajaccio, France, May 5-7, 2008. Proceedings. [Piscataway]: IEEE, cop. 2008, pp. 139-144.
32. Tkalčič, Marko, Alič, Kemal, Zajc, Matej, Dobravec, Štefan, Trnkoczy, Jernej, Meža, Marko, Tasič, Jurij F. Scalability of the "Peer-to-Peer for Major Events" platform. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik sedemnajste mednarodne Elektrotehniške in računalniške konference ERK 2008, 29. september - 1. oktober 2008, Portorož, Slovenija, (Zbornik Elektrotehniške in računalniške konference ERK, ISSN 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2008, zv. A, pp. 133-136.
33. Markelj, Tomaž, Alič, Kemal, Burnik, Urban, Zajc, Matej, Tasič, Jurij F. Postavitev iDTV DVB-T/C okolja za pripravo interaktivnih vsebin. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik sedemnajste mednarodne Elektrotehniške in računalniške konference ERK 2008, 29. September - 1. Oktober 2008, Portorož, Slovenija, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., ISSN 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2008, zv. A, pp. 137-140.
34. Alič, Kemal, Zajc, Matej, Tkalčič, Marko, Tasič, Jurij F. Interaktivna digitalna televizija kot izobraževalni medij. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik sedemnajste mednarodne Elektrotehniške in računalniške konference ERK 2008, 29. September - 1. Oktober 2008, Portorož, Slovenija, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., ISSN 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2008, zv. B, pp. 345-348.
35. Zajc, Matej, Alič, Kemal, Battelino, Irena, Tasič, Jurij F. Challenges of interactive digital television for t-learning. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik Šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, 24. -26. September 2007, Portorož, Slovenija. Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2007, zv. B, pp. 314-317.

36. Alič, Kemal, Sernec, Radovan, Zajc, Matej, Tasič, Jurij F. Interconnection network switch fabric architectures. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). Zbornik petnajste mednarodne Elektrotehniške in računalniške konference ERK 2006, 25. - 27. September 2006, Portorož, Slovenija, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., ISSN 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2006, zv. A, pp. 199-202.
37. Švigelj, Aleš, Kažič, Blaž, Alič, Kemal, Škrjanc, Maja, Kuhar, Urban. State estimation: the key to distribution anagement system: presented at 1st SUNSEED Workshop on Advanced Control, Communications and Algorithms for the Smart Grid, 25 March 2015, Aalborg, Denmark.