INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(3), 293-310, June 2019.

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

K. Alic, M. Mohorcic, A. Svigelj

Kemal Alic
Jozef Stefan International Postgraduate School
Jamova cesta 39, 1000 Ljubljana, Slovenia
kemalic@gmail.com

Mihael Mohorcic
1. Department of Communication Systems, Jozef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
2. Jozef Stefan International Postgraduate School
Jamova cesta 39, 1000 Ljubljana, Slovenia
miha.mohorcic@ijs.si

Ales Svigelj*
1. Department of Communication Systems, Jozef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
2. Jozef Stefan International Postgraduate School
Jamova cesta 39, 1000 Ljubljana, Slovenia
*Corresponding author: ales.svigelj@ijs.si

Abstract: Practical experience of using opportunistic network coding has already been gained in several real network deployments, indicating the influence of some of the fundamental characteristics of the network and the traffic load. However, these aspects have not been systematically investigated in the scope of the construction of efficient and robust large-scale network-coding-enabled wireless mesh networks. In this paper we focus on these aspects using an example of two opportunistic network-coding procedures: the well-known COPE and the Bearing Opportunistic Network coding (BON). In addition, the design aspects for network-coding-enabled wireless mesh networks and applications are discussed. We have shown that opportunistic network coding can improve the performance of different networks and supported applications in terms of throughput, delay and jitter, although the benefits are not significant in all the cases. Thus, the use of opportunistic network coding should be considered upfront during the wireless network design phase in order to obtain the greatest benefits.

Keywords: opportunistic network coding, network coding algorithm, wireless multihop networks, simulations; traffic design.

1 Introduction

Network coding has been successfully used to improve the performance of large, wireless, multi-hop networks. With network coding the conventional store-and-forward paradigm in relaying scenarios is replaced by 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. The broadcast nature of wireless media is embraced as all the packets are distributed, for free, to all the neighbours, i.e., the nodes that can overhear the transmission. All the nodes store all the received packets for possible packet-decoding purposes. Hence, a typical network-coding mechanism is composed of two main operations: (1) the coding of the outgoing packets on (intermediate) nodes and (2) the decoding of incoming packets upon their reception on nodes [19].

Network coding was originally proposed to increase the throughput of a single-source communication, for instance, a multicast stream [1], in order to achieve the maximum data-transfer rate in multicast networks [12]. In this paper, we are interested in opportunistic network coding, where the packets for transmission are selected in a classic manner and the coding opportunities are searched for only amongst the packets currently present in the output queue [10, 13, 19]. We further focus our interest in fixed, wireless mesh networks, such as metropolitan Wi-Fi networks [6] or Wireless Sensor Networks (WSNs) [22] with unicast traffic, where the nodes can be seen as fixed wireless access points for the end users.

The performance gains of opportunistic network coding in wireless mesh networks have already been proven in testbed deployments [8, 10]. The gains have also been analysed from the theoretical perspective [24] and compared to practical gains [25]. Opportunistic network coding has also been considered for streaming applications. Th code selection that takes into account video-stream characteristics, such as the distortion values and the play-out deadlines of the video packets, was studied in [15]. The well-known and widely adopted COPE [10] has been under the spotlight, while so-called support mechanisms such as scheduling [21], queue management [7,16,17] and routing support [11,14,20] were used to bring about additional gains. Most of these support mechanisms can also be used with other opportunistic network-coding approaches, such as [3, 8, 19].

In this paper we are not optimising the parameters of any particular network-coding algorithm or one of the support mechanisms, rather we are investigating the impact of different network characteristics on the performance of network-coding procedures. Thus, our interest is in 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. As a particular example, this paper deals with the design and construction of an efficient and robust, large-scale (99 nodes) network-coding-enabled, wireless mesh network that ensures the delivery of high-data-rate and low-delay services for different applications used by a very large number of distributed users. We analyse and explain the major impacts on network performance in terms of a throughput and delay analysis of the opportunistic network-coding using two different opportunistic network coding procedures, namely, the well-known COPE [10] as a representative of a procedure based on distributed information, and the recently proposed BON (Bearing Opportunistic Network Coding) [3–5], representing procedures that make coding decisions based only on local knowledge, thus reducing the overhead. Both selected network-coding procedures are positioned between the MAC and the network layer and are performing single-hop packet coding. Support mechanisms that further improve the benefits of network coding are not considered.

This article is organised as follows. In Section 2 we briefly summarize the main characteristics of the COPE and the BON. In Section 3 the methodology for a subsequent performance comparison is presented. Section 4 analyses the influence of wireless-network design parameters on the performance of opportunistic network coding from the quantity-of-service perspective (e.g., goodput). The end user perception of network coding in terms of delay and jitter is presented in Section 5. Next, coding fairness for different traffic conditions is evaluated in Section 5 before Section 6 concludes the article.

2 Considered network-coding procedures

In this study we consider COPE [10] and BON [4] for use in wireless mesh networks. Both procedures are inter-session network-coding schemes that use the XOR operation for the coding of packets. They are fully distributed, able to discover coding opportunities and can be seen as an independent layer in the communication stack.

After receiving an encoded packet, the BON and COPE processes try to decode it. Decoding is successful if there are enough packets in the packet pool. An ACK message is sent for the native packet, but only if the recipient node is determined as one of the next hops for the packet. If there is not enough information to decode the packet, the encoded packet is dropped.

Both procedures use cumulative ACK reports, which were first presented in [10]. Cumulative ACK-report messages are broadcasted periodically, every T_u seconds. If the opportunity arises the ACK reports are attached to the regular outgoing packets. In this case cumulative ACKs can be seen as an additional header. The cumulative ACK messages reduce the overhead compared to the individual ACK messages, thus optimizing the network coding from the throughput point-of-view.

If an ACK message is not received on the sending node within a predetermined time, the re-transmission event is triggered. A native packet that has been sent out as part of the encoded packet and has not been ACKed is to be re-transmitted. In the COPE the packet is placed at the head of the output queue. In the BON the packet is placed in the re-transmission output sub-queue. This queue has a lower priority than the signalization queue, but a higher priority than the queue with regular outgoing packets. The packets in the retransmissions queue can be coded again. However, coding opportunities are only searched for within the packets in the regular output queue. This mechanism ensures that it is not possible for the same set of native packets that already failed in the decoding to be coded again.

The COPE also uses so-called reports. With these each node informs all its neighbours about which new packets it has received. The reports are broadcasted periodically, every T_u seconds. If possible, the reports are attached to regular outgoing packets or ACK messages as additional header.

At the MAC layer the BON and COPE use a pseudo-broadcast mechanism, presented in [10]. Given that the BON and COPE rely on the same mechanisms of the underlying OSI layers, in this paper we assume the use of the same MAC layer that mimics the provision of resources as in CSMA-CD. The COPE and BON coding algorithms are described in the following subsections.

2.1 COPE

In the COPE the coding process depends of the nodes' knowledge on what information (which packets) its neighbouring nodes have, through listening to the neighbours' broadcasts (i.e., regular packets or reports). Based on this knowledge the coding process is straightforward and the decoding process will have a high success rate. Information arriving through particular messages and through listening to all the broadcasted messages provides only a few coding opportunities. In the case that the information about a packet's presence at a particular neighbour's node is not available, the coding needs to make a guess regarding the situation. The guessing is made through a delivery probability that is calculated in all of the ETX-based (Expected Transmissions Count) routing protocols. The node estimates the probability that the node N has the packet p by looking at the delivery probability for the link between packet's previous hop and node N.

With all the required information, the node can code together as many packets (p_1, ..., p_n) as possible provided that none of the packets were created on the coding node, all the packets have different next hops and that there is a strong possibility that all the next hops will be able to decode the packet. The next hop can decode the packet if it has already received all but one of the packets coded together. Let the probability that a next hop has heard packet p_m be P_m. Then, the probability, P_D, that it can decode its native packet is equal to the probability that it has heard all of the $n - \mathit{1}$ native packets encoded with its own, i.e.,

$$P_D = P_1 P_2 \dots P_{n-1} \tag{1}$$

The coding algorithm ensures that the decoding probability P_D for all the next hops for a given combination of encoded packets is above the threshold G.

2.2 BON

The BON-coding process [4] relies only on information gathered locally in the network-coding layer and does not record the state of the traffic flows. The BON-coding process is based on the local bearing of the packet $p_{ij}^{(R)}$ which is defined on the relay node V_R and depends on the positions of the $p_{ij}^{(R)}$ previous hop $V_i(X_i, Y_i)$ and the next hop $V_j(X_j, Y_j)$. The bearing for the packet $p_{ij}^{(R)}$ on the relay node is defined as a unit vector, calculated as:

$$\overrightarrow{b}_{ij} = \frac{(X_j - X_i, Y_j - Y_i)}{\|(X_j - X_i, Y_j - Y_i)\|} \tag{2}$$

With the BON the packets $p_{ij}^{(R)}$ and $p_{kl}^{(R)}$ are codeable on the relay node if the previous hop of $p_{ij}^{(R)}$ (V_i) is in the *vicinity* of the next hop of ${p_{kl}}^{(R)}$ (V_l) and vice versa. This is where the *vicinity* of the node V_j for the packet $p_{ij}^{(R)}$on the relay node V_R is described as an area within the shape of an infinite cone, with the apex in the packet source V_i, and the cone axis with the direction $\overrightarrow{b_{ij}}$ and the aperture $2\epsilon_{ij}^{(R)}$.

ϵ is referred to as the tolerance angle parameter and $\epsilon \geq 0$. There is one tolerance angle on each node, i.e., $\epsilon_{ij}^{(R)} = \epsilon_{kl}^{(R)} = \epsilon^{(R)}$. By increasing the parameter ϵ, the possibility of encoding multiple packets and the probability that one of the receivers will not be able to decode the packet is 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. In the case of $\epsilon = 0$ only two packets can be coded.

On each node the tolerance angle ϵ is adjusted based on the retransmissions-ratio parameter (RR), reflecting the success of the packet coding, which is calculated on the node V_i as:

$$RR(m)_{V_i} = \frac{K_{RmV_i}}{K_{CmV_i} K_{PmV_i}} \tag{3}$$

where $m = \{1, 2, ...\}$, $K_{CmV_i} > 0$ 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_{PmV_i} is the number of packets that were passed from the network layer to the network coding layer. K_{RmV_i} is the number of network-coding-layer retransmitted packets on the node. All the values are measured in the time interval $[(m-1)T_{\epsilon U},\ m\, T_{\epsilon U})$. The obtained value is compared to the maximum threshold RR_{max} and the minimum threshold RR_{min}. Based on the comparison outcomes, the process provides a decision about increasing, decreasing or leaving the parameter ϵ unchanged.

3 Methodology for evaluating the performance

In this section we define the performance metrics and the simulation parameters for a subsequent investigation of the impact that the network and traffic characteristics have on the efficiency of the network coding.

3.1 Performance metrics

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 the sum of all the goodput on all the network nodes at a particular network load in the i-th simulation run, i.e., $g(i)$.

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

$$G(i) = \frac{g_{NC}(i) - g_{nocoding}(i)}{g_{nocoding}(i)} 100\% \tag{4}$$

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

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

$$ETE(i) = \frac{\sum_{n=1}^{K_a(i)} d_n}{K_a(i)} \tag{5}$$

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 the true periodicity of a presumed periodic signal. We only measure the jitter for flows with constant inter-arrival packet rates:

$$jitter(i) = \frac{1}{F(i)} \sum_{f=1}^{F(i)} \frac{1}{K_f(i)} \sum_{n=1}^{K_f(i)} (d_{nf} - ETE_f(i))^2 \tag{6}$$

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 the ETE for the f-th flow. When we are interested in the jitter of only one particular flow, this same equation is used for the calculation and $F(i) = 1$.

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

We also consider Jain's index [9], which is primarily used to identify underutilized channels or fairness in the load distribution. We use it to identify underutilised streams.

$$J(i) = \frac{(\sum_{f=1}^{F} g_f(i))^2}{F \sum_{f=1}^{F} g_f(i)^2} \tag{7}$$

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

$$G_s = sort\{G(1), G(2), ..., G(I)\} \tag{8}$$

and then the $MaxGain$ is:

$$MaxGain = \frac{1}{3}(G_s(1) + G_s(2) + G_s(3)) \tag{9}$$

3.2 Network-coding settings

Each investigated characteristic was evaluated in a separate scenario. To this end, two opportunistic network-coding procedures, COPE [10] and BON [4], as well as the reference scenario where network coding was not used (no coding), were considered in the evaluation. A purposely developed simulation model, presented in [2], built in the Riverbed Modeler simulation tool, was used. Each simulation run took 150 s. The results were collected between the 90th and the 150th second to observe only the steady-state conditions. 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, as in this study we are focusing on the impact of the network and the traffic characteristics on the efficiency of the network-coding procedures, rather than scaling (i.e., optimising) the network-coding parameters. Both the BON and COPE use up to two retransmissions in the network-coding layer. Both procedures store packets in the packet pool for 15s after the packet has been received for the first time. The BON and COPE send out cumulative ACKs at least every 0.5 s. The network-coding-layer packet re-transmissions are scheduled for 0.6 s after the initial packet transmission. If the opportunity arises, cumulative ACKs are attached to the regular outgoing packets. The COPE reports are sent out at least every 0.5 s. When possible, the report packets are also attached to the regular outgoing packets or to the cumulative ACKs. The COPE parameters were set to the values used in the test-bed presented in [10]. The BON parameters that are common to the COPE procedures are set to the same values. The other BON parameters were set to the values used in [4].

3.3 Traffic, topology and variable parameters settings

The UDP-like traffic was generated in the origin nodes. The traffic intensity was regulated through the packet inter-arrival time and the 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 12,000 bits. The packet size distribution had two peaks and followed the measured internet traffic, as presented in [18]. The inter-arrival time between the packets was constant or calculated using an exponential distribution. Different numbers of traffic flows were 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 nodeV_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 the packets had several opportunities to be 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 by the traffic-symmetry ratio.

All the nodes used the same channel with a capacity of 2 Mbit/s. The packet-delivery probability between the nodes changed dynamically during 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 loads the packet-delivery probability was between 0.7 and 0.97.

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

The BON performance and the influence of the parameters on the network-coding performance were studied in different topologies to also show that the performance improvement does not depend on optimal parameter settings. We randomly deployed 99 fixed nodes over an area of $2\,km \times 2\,km$. Several node-transmission (Tx) ranges were used in order to study different network topologies with the same node distributions. The node-transmission range indicates how far the signal from the node can be detected, which is reflected in the connectivity and the wireless links between the nodes. As an example, in Fig. 1 two topologies with the shortest (280 m) and the longest (420 m) considered node-transmission ranges are depicted.

We were interested in several parameters that influence the network-coding performance. All the variable parameters, with an included reference to the section where they were used, are presented in Table 1. In general, for each case only one parameter was varied, and all the others remained fixed.

Table 1: Simulation parameters for investigated characteristic

	Transmiss. range	**Number of traffic flows**	**Traffic-symm. ratio (%)**	**Packet length (kb) Int. time dist.**	**Queue length (packets)**
Topology design (4.1)	280 - 420 m	’*all2all*’	100	10; exp	100
Traffic distribution impact (4.2)	300 m	10 - 90, ’*all2all*’	0-100	10; exp	100
Packet length evaluation (4.3)	300 m	’*all2all*’	100	2 - 10, *var*; exp	100
Analysis of output queue length (4.4)	300 m	’*all2all*’	100	10; exp	10 - 200
QoS analysis (5.1)	300 m	2, 20	100	10; const.	100
Coding fairness (5.2)	300 m	20, 90	100	10; exp	100

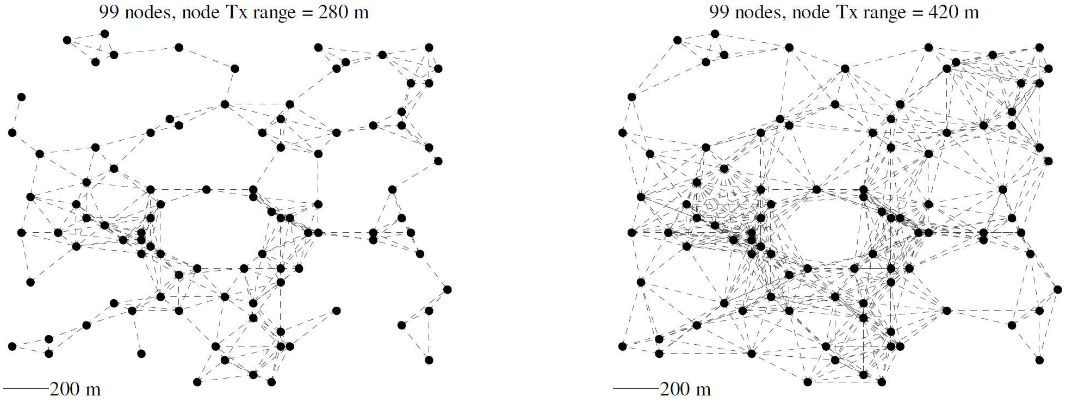
99 nodes, node Tx range = 280 m
200 m
99 nodes, node Tx range = 420 m
200 m

Figure 1: A graphical presentation of selected network topologies used in the evaluation

4 Quantitative impact of the network and the traffic characteristics on the network coding

The core of this section is the analysis and evaluation of the influence of the 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.

4.1 Topology design

Fig. 2 depicts the gain with respect to the increasing network load for network topologies with the node-transmission range set to 280, 340 and 400 m. The highest network-coding benefits were obtained for the topology where the nodes have the shortest transmission range. This might be counter-intuitive to a certain extent, as by a larger coverage area, it is possible to think that more nodes overhear transmissions, thus resulting in more coding opportunities and a higher decoding success. We wanted to present the gain, with its dependence on the node-transmission range, so the same results are presented in Fig. 3 as *MaxGain*, as obtained for all the topologies. The coding benefits decrease with a higher node-transmission range. There are several reasons for this. The diameter and the average hop-count distance of each network decreases with an increased nodes-transmission range. With this there are fewer possibilities for the packets to be encoded, as they need fewer hops from their sources to their destinations. In networks where the nodes have a shorter node-transmission range there are typically a few nodes that handle large portions of the traffic. Nodes that handle a larger portion of the load are where normally the majority of the coding takes place. With a larger node-transmission range the number of possible paths between the source and the destination pairs increases, hence the load is more evenly distributed among the nodes. It often happens that the flows between two neighbouring nodes on one side of the network and two neighbouring nodes on the other side of the network use entirely disjointed paths, while often still competing for the same wireless resources, but not being able to be 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.

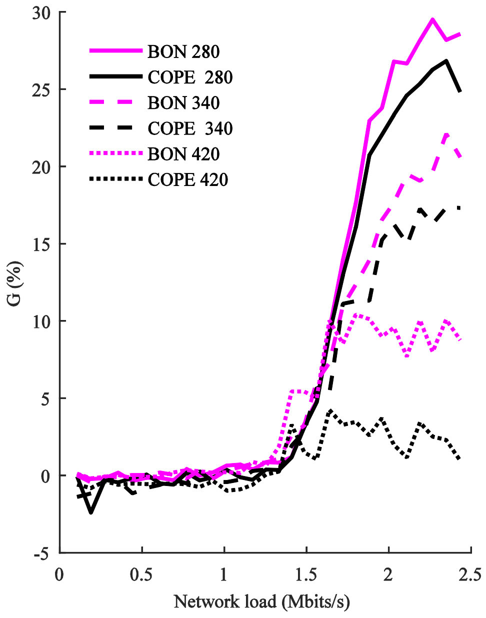

Figure 2: Dependence of the gain (G) on the network load for topologies with a 280, 340, and 400 m node-transmission range

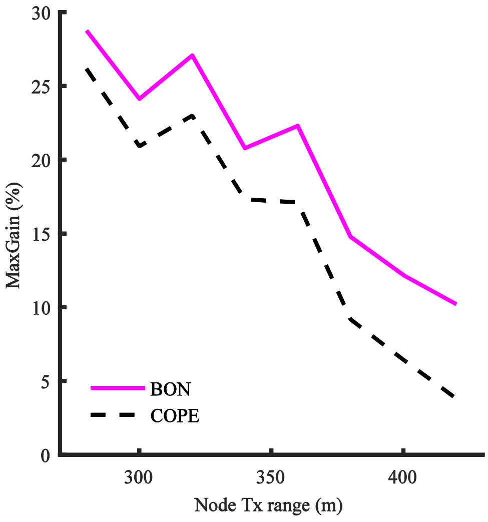

Figure 3: Dependence of the *MaxGain* on the node-transmission (Tx) range

Furthermore, with a 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 can arrive at 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.

Based on the above analysis we can conclude that the goodput of networks where the node transmission range is high can only be increased by a few percent using basic opportunistic network-coding approaches. If the network coding is considered for use in a mesh network, where the nodes have many neighbours, the support of a network-coding-aware routing protocol should be considered, or a network-coding procedure with a different approach should be selected such as e.g. BEND [23].

4.2 Impact of the traffic distribution

The potential benefits of using network coding are strongly dependent 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 1, where also all the other parameter settings are stated.

In Fig. 4 we present the results for the dependence of $MaxGain$ on the traffic-symmetry ratio. As expected, the gains increase as the traffic within the traffic flows becomes more symmetric, essentially increasing the network-coding opportunities. With the lowest ratio (i.e., a traffic symmetry ratio $= 0$) small gains were noted. Since the coding happens between the flows that cross their paths in opposite directions, the benefits of network coding increase with an increasing traffic-symmetry ratio. We can see that the network-coding procedures bring the most benefits when the traffic symmetry is above 60 %.

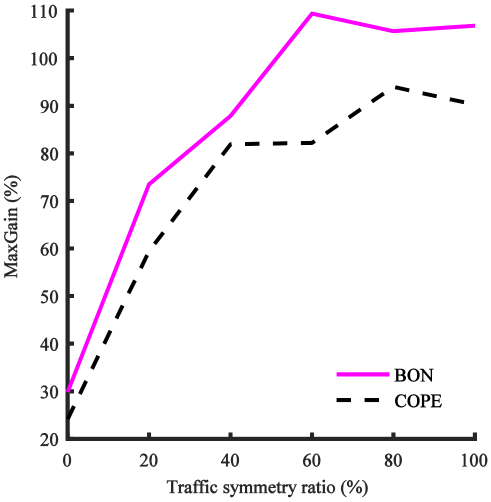

Figure 4: Dependence of the $MaxGain$ on the traffic-symmetry ratio

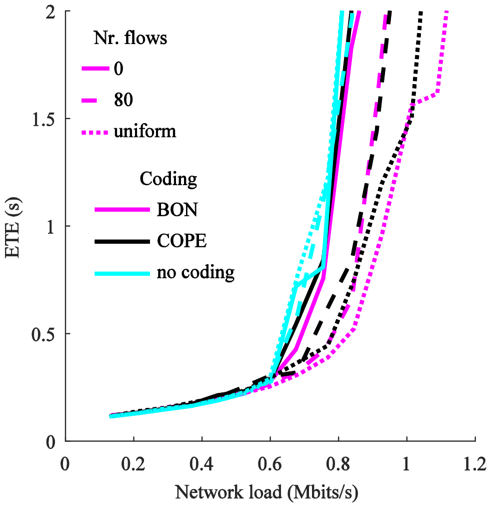

Figure 5: Dependence of the ETE delay on the goodput (g) for 20 and 90 traffic flows and '$all2all$' traffic distribution

In the next set of results, we evaluated the dependency of the network-coding benefits with respect to the number of traffic flows, i.e., the traffic-flows distribution. In Fig. 5, The dependence of the ETE delay on the goodput for 20 and 90 traffic flows and the '$all2all$' traffic distribution is presented. By increasing the number of traffic flows the network goodput increased with an unchanged ETE delay. This is expected, as with the larger number of traffic flows, the same overall network load is more evenly distributed across the network. Furthermore, the network coding decreases the delay in comparison to the no-coding scenario, while increasing the network goodput, but the delay is increasing due to the increased network load.

In Fig. 6 the dependence of $MaxGain$ on the number of traffic flows is shown. With very few flows in the network (i.e., 10) the $MaxGain$ 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 (shown as separate points in the graph), where an in-depth analysis also revealed that the fewest coding opportunities are found. Nevertheless, in the '*all2all*' distribution of traffic flows, where the network-coding benefits were the lowest, the network-coding gain with both procedures was still higher than 20 %, and in addition both procedures decreased the delay significantly.

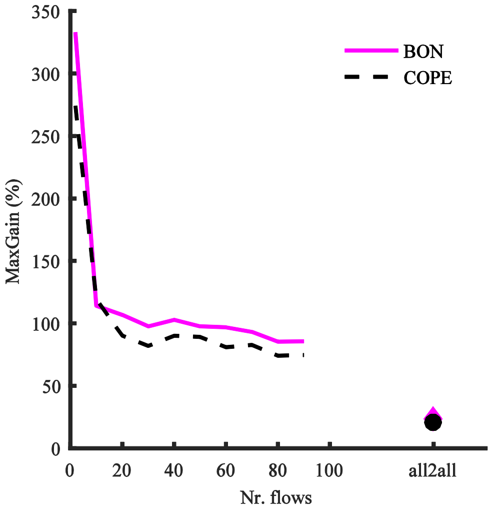

Figure 6: Dependence of the $MaxGain$ on the number of traffic flows

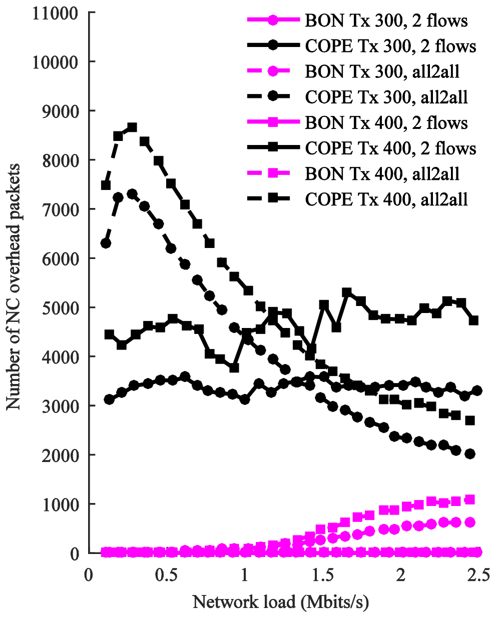

Figure 7: Network coding overhead in terms of the number of individual packets for 2 flows and '*all2all*' traffic distributions and topologies with node transmission (Tx) range 300 and 400 m

In topologies where the nodes have a higher node-transmission range, the signalization overhead also becomes more significant. In Fig. 7 the overhead caused by the 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 of 300 and 400 m. The COPE uses the so-called report messages, with which the nodes inform their neighbours about 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 number and in the size (each node composes reports for each of the neighbours from which it has received new, unreported) packets). A higher number of traffic flows means a higher number of COPE reports, as there are more nodes that can overhear the packet transmission. In low traffic-load conditions there are few opportunities to attach reports to regular outgoing packets, hence a high number of individual report messages. By increasing the traffic load the number of report messages also depends on the traffic-flow distributions. In an '*all2all*' traffic-flow distribution the number of stand-alone individual messages quickly decreases, as there are plenty of regular outgoing packets on all the nodes to which the report messages can be attached, and 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 the compact format of report messages the number of individual messages remains almost constant, regardless of the traffic load.

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

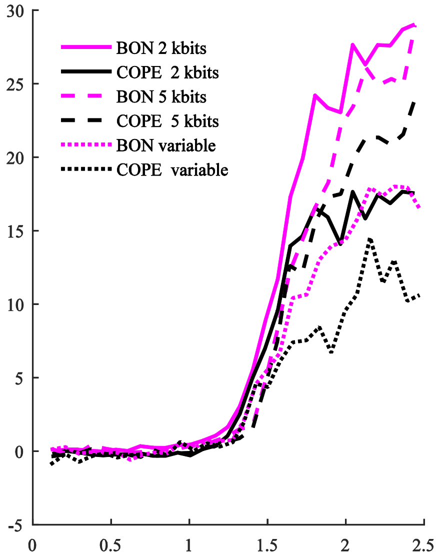

Figure 8: Dependence of the gain (G) on the network load with packets of 2000 and 5000 bits, and of variable length

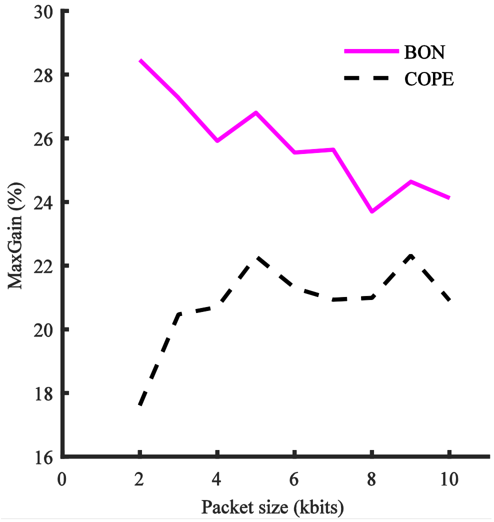

Figure 9: Dependence of the $MaxGain$ on the packet length

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 the parameters were set according to the specifications in Table 1.

In Fig. 8 the dependence of gain on the network load for packets of 2000 and 5000 bits is shown (the results for a variable packet length are also shown, but they are discussed later). We can see that the highest gains were obtained with the BON for a packet length of 2000 bits. The BON self-adaptation process is based solely on information gathered in the network-coding layer. The more packets the network-coding layer processes, the better the decision process, and hence there are fewer retransmissions, indicating better coding decisions.

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

The dependence of $MaxGain$ on the packet length is presented in Fig. 9 for the BON and COPE. With the BON the $MaxGain$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 the BON self-adaptation process also depends on the number of processed packets. The COPE generated a higher overhead than the BON, and hence the $MaxGain$ was lower with short packets. By increasing the packet length, the gains also grew to the point where the signalization did not affect the results and the $MaxGain$ shows no dependence on the size of the packet length .

We were also interested in the network performance in terms of the variable packet lengths (also presented in Fig. 8, and noted as "*variable*"). Thus, additional experiments were carried out with the empirical packet-length distribution presented in [18], where the 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 a variable packet-length distribution show the lowest gains. Both coding algorithms favour the coding of a packet of approximately the same length, but the degrading impact of the packet extension cannot be compensated.

4.3 Analysis of the length of the output queue

In this subsection we examine the influence of the output-queue length on the network-coding performance. The output-queue length is important because The COPE and BON do not deliberately delay packets in order to find more coding opportunities. Instead, both analysed algorithms only search for coding opportunities among the packets present in the output queue. For this analysis the parameters were set according to the specifications in Table 1.

In Fig. 10 the dependence of $MaxGain$ on the 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 we would expect that more packets in the output queue will increase the coding opportunities, the results show that initially the $MaxGain$ grew very quickly with the queue length, but then the increase in the queue length had no effect on the performance of the network coding.

5 End-user perception of the network coding

5.1 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. The BON and COPE use asynchronous acknowledgement mechanisms to confirm the successful transmission and decoding of encoded packets. Furthermore, cumulative acknowledgements are used to avoid a high overhead and to increase the goodput. This is reflected in the relatively long time periods used for packet-retransmission scheduling (in our case 0.6 s). Such a long waiting period is expected to affect the streaming applications in terms of the 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 the 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 a 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 an 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 1.

Fig. 11 shows the dependence of ETE delay on the increasing network load with two and twenty symmetric flows. The delay was not affected at lower loads in comparison to the no-coding scenario. However, with the higher loads, the delay obtained in the case of network coding was significantly lower than in the no-coding scenario. As shown in Fig. 12, the 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 the jitter increased, though this was in the case when the network was congested and packets were being dropped out of the 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 most of the cases on a 14-hop path the packets were encoded on several hops.

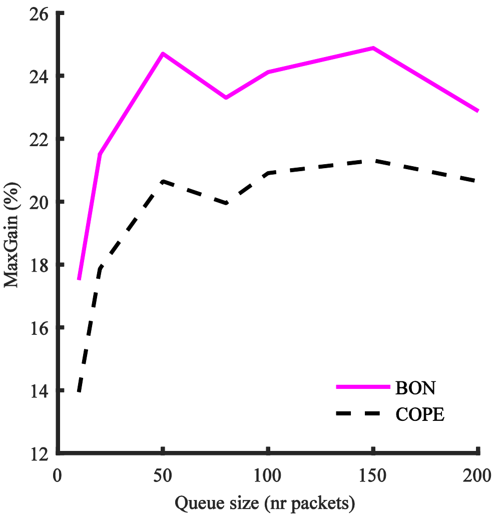

Figure 10: Dependence of the $MaxGain$ on the queue length

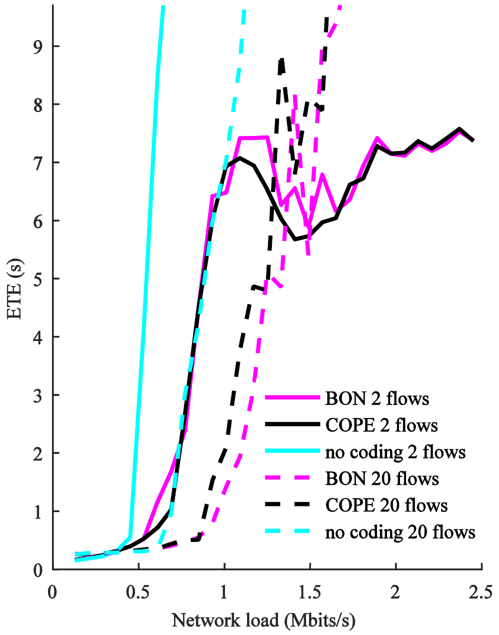

Figure 11: Dependence of the ETE delay for the longest flow on the network load with 2 and 20 symmetric traffic flows.

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

Generally, we can conclude that network coding has a positive impact on the jitter. The benefits of network coding with respect to the no-coding scenario appear at high network loads, when the jitter is significantly affected in the no-coding scenario due to the congestion of the 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 for the presented analysis so that packets were encoded several times on their path to the destination, and hence the effects of network coding would be observed with higher reliability.

5.2 Analysis of coding fairness

An important consideration for the network-coding procedure is also its fairness, i.e., how different traffic flows benefit from the network coding. In general traffic conditions only a small part of the native packets are encoded, thus observing the performance of the network coding for the entire network may blur the effects at the level of individual traffic flows. The aim of this section is to analyse the fairness of the network coding with respect to the 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.

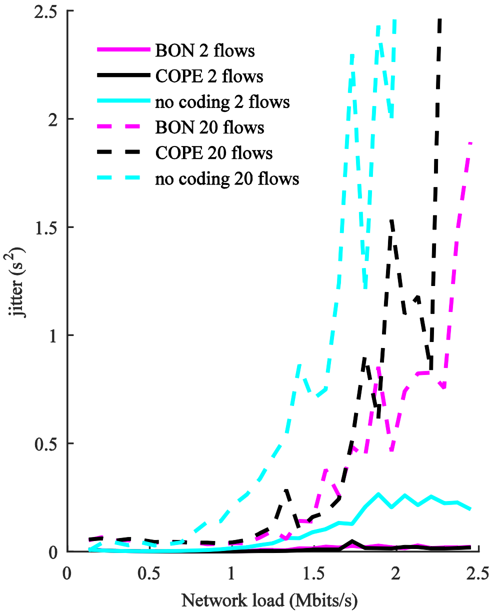

Figure 12: Dependence of the jitter for the longest flow on the network load with 2 and 20 symmetric traffic flows

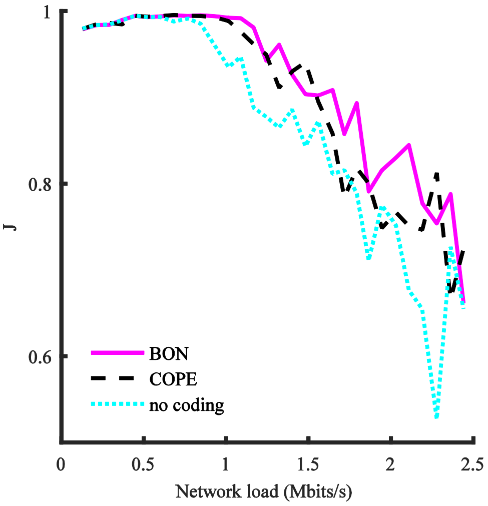

Figure 13: Jain's index (J) for the distribution with 20 traffic flows

We carried out a detailed analysis of the 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 1.

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 the system resources. In Fig. 13 Jain's index for 20 flows of traffic distribution is presented. Jain's index indicates that the network-coding procedures treat the traffic at least as well as or better than when there is no coding; however, it does not reveal how the coding gains were distributed among the flows.

In addition to Jain's index, in the following we use the gain of the particular traffic flow as the performance metric, so we refer to it as the flow gain. The flow gain results for a traffic distribution consisting of 20 traffic flows are shown in Fig. 14. The flow gain was collected at an 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 a 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 the full queues. For 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%.

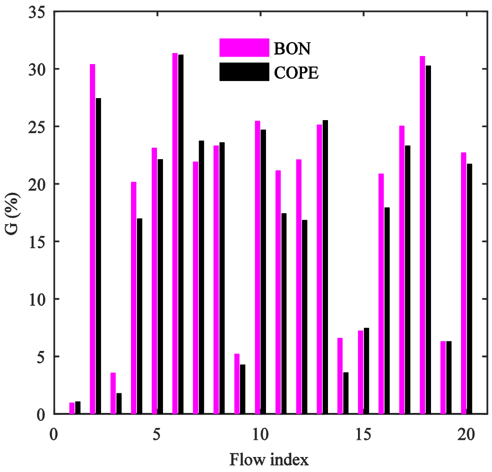

Figure 14: Gain (G) per individual flows for the traffic distribution with 20 traffic flows

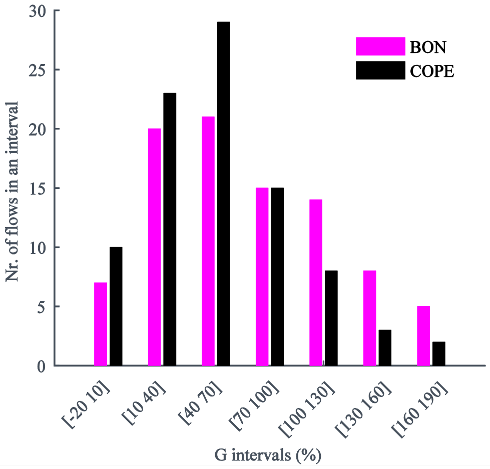

Figure 15: Histogram for gain (G) per individual flow for the traffic distribution with 90 traffic flows

In Fig. 15 we present the results for the case with an 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 the COPE was lower than 3 s. Again, the network was in a state where packets on several nodes were dropped due to the full output queues. On the x-axis we have the 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 the BON achieved a flow gain between 10 % and 40 %. Although the goal of the network-coding procedures is to successfully transfer many flows with a 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 a few flows where the gains were low. A detailed analysis revealed that there were 2 out of 90 flows with the BON having a negative coding gain, where the lowest value was -1.15 %. With the COPE there were four flows with a negative gain, with the lowest value being -16.7 %. Although we present only a few representative results, the same conclusions can also be drawn for different traffic distributions and higher traffic loads.

6 Conclusions

Two opportunistic network-coding algorithms the COPE and BON were used for evaluation purposes. Both procedures are similar in several aspects, but they differ in their core, i.e., the coding decision algorithm, which in case of the COPE using exact distributed information about coding opportunities, whereas in the case of the BON it is based only on the 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 investigation we focused on the impact of the network and traffic characteristics on the efficiency of the network-coding procedures, and thus the network-coding parameters remained fixed throughout the evaluation.

In the following we summarise the observed and discussed design principles for wireless networks using network coding and their applications. Table 2 summarises the winning parameters and protocol with an included reference to the section where they are analysed.

Table 2: Summary of winning parameters for analysed scenarios

	Parameter	**Values**	**Win. value**	**Win. protocol**
Topology design (4.1)	Transmission range	280 m - 420 m	280 m	BON
Traffic distribution impact (4.2)	Traffic-symm. ratio	0% - 100%	> 60%	BON
Traffic distribution impact (4.2)	Number of traffic flows	10 - 90, '*all2all*'	10	BON
Packet length Evaluation (4.3)	Packet length	2 kb - 10 kb, '*var*'	2 kb	BON
Analysis of queue length (4.4)	Queue length	10 p - 200 p	50 p	BON
QoS Analysis - ETE delay (5.1)	Number of traffic flows	2, 20	20	BON ≈ COPE
QoS Analysis - Jitter (5.1)	Number of traffic flows	2, 20	2	BON ≈ COPE
Coding Fairness Jain's index (5.2)	Number of traffic flows	20	20	BON ≈ COPE

Traffic distribution, and in particular, 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 a completely symmetric traffic load. Furthermore, the highest gains are obtained when there is a relatively small number of flows in the network. By increasing the number of flows the network-coding gains decrease.

The average length of the 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 a larger influence on the network with smaller packets. When packets of the same length are encoded the gains obtained are higher than when the packets vary in length. In addition, the enlarging of the output queues improves the gains, but only to a certain level, which also depends on the number of flows in the network. Hence, the length of the output queue should primarily be adjusted to handle the expected peak network loads.

When planning the network topology, additional transmission power and thus an increased nodes-transmission range in network-coding-enabled networks will not necessarily improve the networks' 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.

The detailed delay and jitter analysis of individual flows revealed no negative effects on the QoS due to the use of network-coding procedures. Although the jitter might have slightly increased (almost negligibly in absolute terms) on rare occasions with low loads, the network performance in terms of delay and jitter with network coding in a heavily loaded network significantly outperformed the no-coding scenario where no network coding was used. It is worth noting that the jitter and delay were analysed on a flow that travelled 14 hops to its final destination and that there were only a few occasions where packets were not encoded at least one time along the path, and still we did not detect any degradation in the 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 the traffic flows in the network, i.e., they were not improving the overall performance at the expense of some less-favoured traffic flows.

Funding

This work has been funded by the Slovenian Research Agency through grants J2-4197 and P2-0016.

Bibliography

[1] Ahlswede, R.; Cai, N.; Li, S.-y. R.; Yeung, R. W. (2000). Network Information Flow, *IEEE Transactions on Information Theory*, 46, 1204-1216, 2000.

[2] Alic, K.; Pertovt, E.; Svigelj, A. (2012). Network coding simulation model in OPNET modeler, *In procc. of OPNETWORK 2012*, Washington, USA, 2012.

[3] Alic, K.; Pertovt, E.; Svigelj, A. (2015). Bearing-Opportunistic Network Coding, *International Journal of Computers, Communications & Control,* 10, 2015.

[4] Alic, K.; Svigelj, A. (2016). A One-Hop Opportunistic Network Coding Algorithm for Wireless Mesh Networks, *Wireless Networks, https://doi.org/10.1007/s11276-016-1384-y, 2016.*

[5] Alic, K.; Svigelj, A. (2017). Self-adaptive practical opportunistic network-coding procedure for static wireless mesh networks, *Ad Hoc & Sensor Wireless Networks,* 36, 87-105, 2017.

[6] Chieochan, S.; Hossain E. (2012). Network coding for unicast in a WiFi hotspot: Promises, challenges, and testbed implementation, *Computer Networks*, 56, 2012.

[7] Coppi, N. D.; Ning, J.; Papageorgiou, G.; Zorzi, M.; Krishnamurthy, S. V.; Porta, T. L. (2012). Network Coding Aware Queue Management in Multi-Rate Wireless Networks, *in Procc. of the 21st International Conference on Computer Communications and Networks (ICCCN)*, Munich, Germany, 2012.

[8] Hundeboll, M.; Ledet-Pedersen, J.; Heide, J; Pedersen, M. V.; Rein, S. A.; Fitzek, F. H. P. (2012). CATWOMAN: Implementation and Performance Evaluation of IEEE 802.11 Based Multi-Hop Networks Using Network Coding, *Procc. of the IEEE Vehicular Technology Conference (VTC Fall)*, Quebec City, Canada, 2012.

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

[10] Katti, S.; Rahul, H.; Hu, W.; Katabi, D.; Médard, M.; Crowcroft, J. (2008). XORs in the Air: Practical Wireless Network Coding, *IEEE/ACM Transactions on networking*, 16, 2008.

[11] Le, J.; Lui, J. C. S.; Chiu, D.-M. (2010); DCAR: Distributed Coding-Aware Routing in Wireless Networks, *IEEE Transactions on Mobile Computing*, 9, 596-608, 2010.

[12] Matsuda, T.; Noguchi, T.; Takine, T. (2011). Survey of Network Coding and Its Applications, *IEICE Transactions on Communications*, E94-B, 698-717, 2011.

[13] Omiwade, S.; Zheng, R.; Hua, C. (2008). Butteries in the Mesh: Lightweight Localized Wireless Network Coding, *Procc. of Fourth Workshop on Network Coding, Theory and Applications (NetCod)*, Hong Kong, 2008.

[14] Pertovt, E.; Alic, K.; Svigelj, A.; Mohorčič, M. (2018). CANCAR - Congestion-Avoidance Network Coding-Aware Routing for Wireless Mesh Networks, *KSII Transactions on Internet and Information Systems*, 12, 9, 2018.

[15] Seferoglu, H.; Markopoulou, A. (2009). Video-aware opportunistic network coding over wireless networks, *IEEE Journal on Selected Areas in Communications,* 27, 713-728, 2009.

[16] Seferoglu, H.; Markopoulou, A., Medard, M. (2011). NCAPQ: Network Coding-Aware Priority Queueing for UDP Flows over COPE, *In procc. of the Int. Symp. on Network Coding (NetCod)*, Beijing, China, 2011.

[17] Seferoglu, H.; Markopoulou, A. (20013). Network Coding-Aware Queue Management for TCP Flows Over Coded Wireless Networks, *IEEE/ACM Transactions on Networking,* 2013.

[18] Svigelj, A.; Mohorcic, M.; Kandus, G.; Kos, A.; Pustisek, M.; Bester, J. (2004). Routing in ISL Networks Considering Empirical IP Traffic, *IEEE Journal on Selected Areas in Communications* , 22, 2004.

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

[20] Yan, Y.; Zhang, B.; Zheng, J.; Ma, J.(2010). CORE: a coding-aware opportunistic routing mechanism for wireless mesh networks, *IEEE Wireless Communications*, 17, 96-103, 2010.

[21] You, L.; Ding, L.; Wu, P.; Pan, Z.; Hu, H.; Song, M. (2011). Cross-layer optimization of wireless multihop networks with one-hop two-way network coding, *Computer Networks,* 55, 1747-1769, 2011.

[22] Zeng, D.;Guo, S.; Leung, V.; Hu, J. (2011). The Exploration of Network Coding in IEEE 802.15.4 Networks, *Int. Journal of Digital Multimedia Broadcasting,* vol. 2011, 2011.

[23] Zhang, J.; Chen, Y. P.; Marsic, I. (2010). MAC-layer proactive mixing for network coding in multi-hop wireless networks, *Computer Networks,* 54, 196-207, 2010.

[24] Zhao, F; Medard, M. (2010). On analyzing and improving COPE performance, *Proc. of Information Theory and Applications Workshop*, Italy, 2010.

[25] Zhao, F; Medard, M.; Hundeboll, M.; Ledet-Pedersen, J.; Rein, S. A.; Fitzek, F.H.P. (2012). Comparison of Analytical and Measured Performance Results on Network Coding in IEEE 802.11 Ad-Hoc Networks, *Int. Symp. on Network Coding (NetCod), 2012*, USA, 2012.