Network Topologies and Traffic Distribution Evaluation for Network Coding

Kemal Alic, Erik Pertovt, Ales Svigelj
Department of Communication Systems,
Jožef Stefan Institute,
Ljubljana, Slovenia
kemal.alic@ijs.si, erik.pertovt@ijs.si, ales.svigelj@ijs.si

Abstract— **In this article, we evaluate network coding algorithm COPE using different topologies, traffic intensities and distributions in order to search for guidelines for network deployments and for better exploring the advantage of network coding opportunities. Using different network topologies we identify nodes that perform well with coding and nodes that do not find any coding opportunities. Results show that in all analyzed networks, although the COPE performs well in decreasing average network delay, there are only few nodes that perform coding operations and up to 25% of nodes that find no coding opportunities at all. Also in, randomly distributed traffic, there is only a fragment of packets that find coding pairs, thus showing place for significant improvements either in used algorithms or in designing networks.**

Keywords-Network coding; COPE; simulation model; performance evaluation

I. INTRODUCTION

In the past few years, network coding (NC) has become popular in both, wired and wireless networks as a mechanism for increasing network throughput. Proposed by Ahlswede et al. [1], NC is a paradigm for encoding multiple packets either from the same or from different streams into the same packet in order to increase the throughput of the network. In wireless networks, NC can be used to exploit the broadcast nature of the wireless medium to enhance the effectiveness of several wireless channels and thus increasing the throughput of unicast traffic in wireless multihop (scenario) networks.

Whilst it is still not clear in which OSI layer the NC will find its place, in our opinion, it offers the largest potential in the network layer in strong collaboration with routing. This might lead to significant modification of routing concepts, as no longer shortest paths, or paths that avoid congestions will be looked for, but rather paths where NC principles can be fully exploited, thus increasing the network throughput capacity while maintaining the desired Quality of Service (QoS).

Presented paper deals with well known COPE algorithm described in details in [2, 3] as it is with MORE [4] the only algorithm that is considered to bring innovation to NC [5]. COPE is an intra-session NC algorithm. It codes packets for one hop, where packet decoding is done. The coding process depends on the nodes knowledge on what information (which packets) neighbouring nodes have. In case the node knows which information neighbours have (through listening to neighbours broadcasts (packets and ACKs) or receiving their updates) the coding process is straightforward and the decoding process will have a high success rate. Information arriving through particular massages and through listening to all the broadcast, is not sufficient and provides only few coding opportunities. In the case that the information on the packet presence at specific neighbour's node is not available the coding needs to guess on the situation. The node estimates probability that the node A has packet P, by looking at the delivery probability between packet's previous hop and node A. With all the needed information the node can code together as many packets as possible, as long as none of the packets have been created on this node, all the packets have different next hops and we know that there is a strong possibility that each next hop (all the neighbouring nodes that we are encoding packets in for) will be able to decode the packet. The next hop can decode the packet if it has already received all except one of the packets coded together.

We observe NC in different wireless network topologies and different traffic distributions and try to highlight where in the NC opportunities appear. We build networks around the nodes that can communicate with different number (i.e., 4, 6, 8) of neighbours, to see whether nodes with higher number of neighbours really have more coding opportunities as theory suggests. Such analysis has to best of our knowledge not been performed yet and it provides insightful information on the NC effects, where and why coding opportunities appear, thus providing a base for NC algorithm and protocol developers.

In the COPE presentation article [2, 3], the algorithm has been evaluated in the testbed in the outside environment using 20 nodes. Network was loaded using packet streams. The tests clearly showed that COPE significantly improves networks throughput, especially when the network is loaded with higher loads (congested network). Though, the testbed placed in the hardly controllable environment did not reveal the underlying process, thus constraining the chances for possible improvements on the algorithm or defining its possible lines of usage. Furthermore, the experiment is described insufficiently, thus not allowing repeating the results. Nevertheless, an accurate performance analysis of COPE is extremely challenging [6].

COPE principle has been adapted and extended also for covering other ideas in NC: for example, in [7], noCoCo algorithm specializes in bidirectional traffic flows. It is trying to maximize the number of coding opportunities for the two opposite direction routes. CLONE [8] generalized COPE to address multiple unicast sessions. The system takes into account lossy links and highlights specific situations where COPE provides no coding gain. In [9] the MORE and COPE principles have been joint in search of benefits of the two at the same time.

Making routing aware of COPE coding opportunities has been investigated in [10] and [11]. In the first case, CORE, a coding-aware opportunistic (hop-by-hop) routing mechanism for wireless mesh networks, combines opportunistic routing by each route hop and localized inter-flow NC, increasing coding opportunities. In the second case, NJCAR, a network joint coding-aware (source) routing for wireless mesh networks, explore entire potential routes from source to destination in search for multiple hop decoding opportunities. Moreover, a new interference-avoid and coding-aware (ICM) routing metric is investigated in [12] for making tradeoffs between increased interference (due to multiple flows gathered together for creating coding opportunities) and increased coding gain.

While a significant amount of research has been done on better usage of the COPE there has been little words on its usability, e.g., in what kind of networks it is usable, where and why coding opportunities appear. The main reason for this is probably in rather unusual development and presentation of this excellent NC algorithm, which has initially been tested in close to real deployment environment.

This paper is organised as follows. Our NC simulation model used for obtaining the results is presented in Section II. In Section III, we present simulation parameters and briefly discuss simulation model. In Section IV, the results are presented and explained, while in Section V, we discuss results and give introduction into our further work.

II. NETWORK CODING SIMULATION MODEL

Simulation model [13, 14] consists of two major parts: the supporting network topology generator that reduces the manual work and main simulation model in OPNET [15] The network description program describes the desired topology, nodes, links and parameters for the communication stack (e.g., throughputs, number of packet retransmissions, loads, etc.). Network and its settings are then imported into the main simulation that takes place in the OPNET Modeler simulation tool.

As depicted in Fig. 1, the main simulation model is divided into five functional layers. Traffic Generation and traffic sink module is responsible for creating the network load and it also provides an end point for packets that have reached their destination. Routing module takes care of routing the packets through the network. The wireless module takes care of successful packet distribution through the wireless channel to the right address taking into account links conditions. Network coding module is the core of our simulation model and is responsible for coding multiple packets into one packet.

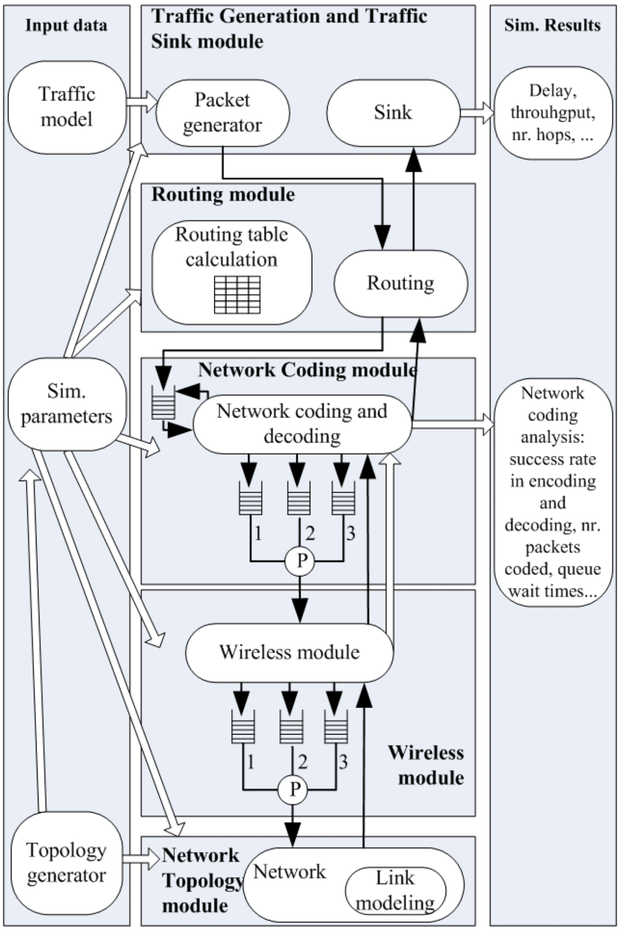

Figure 1. Network coding simulation model architecture.

A. *Traffic module*

Traffic module is responsible for loading the network with traffic. It allows evaluation of quality of service through measuring delay and amount of lost packets and quantity of service through throughput measurements.

The module supports two traffic distributions. In the first case, traffic is distributed throughout the entire network, where all nodes generate traffic and all nodes are destinations, while in the second case all nodes except their neighbours are destinations. Traffic intensity is set at each node individually.

B. *Routing module*

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

Routing tables are calculated using Dijkstra's algorithm [16] taking into account distances between nodes, Expected Transmission Count (ETX) metrics or hop count.

C. *Network coding module*

In network coding module, COPE algorithm is implemented. The module has two functions. Its goal is to code as many packets as possible into one packet. When a module receives an encoded packet it tries to decode it. The module's success on decoding depends on the packets content and on the packets the module already has. The module has to own at least N-1 packets that have been coded into an encoded packet, where N is the number of all packets coded into the encoded packet. If the node can decode the packet it saves all the packets for later decoding purposes and it forwards to routing layer only packets destined to node's address.

In case module cannot decode the encoded packet, it discards the non-decoded packet and waits for the retransmission from the sender's side.

D. *Wireless module*

Wireless module takes care of successfully transmitting packets over the wireless medium. Using acknowledgement mechanisms it takes care of successful packet transmission.

In addition to using unicast transmissions where node that is the packet's recipient confirms successful reception with an ACK, we have also implemented a mechanism that boosts up effects of NC. It is a pseudo-broadcast, which has first been introduced by Katti et. all in [3]. This mechanism unicasts packets that are meant for broadcast. The link-layer destination field is set to the MAC address of one of the intended recipients (next hop of one of the packets coded into an encoded packet). 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 identical to its own, it sends an ACK massage to the sender. Regardless of packet's next hop address, the node sends the packet to the Network coding module.

E. *Network topology module*

The topology generator is developed in MATLAB [17] and it is able to generate random wireless topologies built around the arbitrary number of nodes that can communicate with different number of neighbours. The distribution of nodes is made in a random fashion within the predefined area. The connectivity between nodes is enabled based on nodes positions and transmission "range" of nodes. Selected topologies are then imported into the OPNET [15] simulation model with all the corresponding network parameters.

Nodes are connected with Point-to-Point receivers and transmitters to allow better topology control which is an important aspect in studying NC procedures. Even though, the connections are wired, they are modelled as wireless medium.

F. *Link model*

Link model can simulate various effects of wireless transmission. It can consider packet delay due to signal propagation through radio medium and transmission delay due to packet size and bandwidth used.

III. SIMULATION PARAMETERS

In our analysis, we assume that all network nodes have the same configuration as, e.g., in a homogeneous network. Networks with 20 nodes have been investigated, where each node has been given a random location within a given area. Node locations remain the same in all simulation scenarios.

Nodes that are connected (i.e., can communicate) to each other are presented with dashed lines between nodes in order to allow better topology control in the simulation. For each simulation scenario all the nodes in the scenario have exactly predefined number of neighbours i.e., $N_N = \{4, 6, 8\}$, while their position remains unchanged. As more scenarios (i.e., more different N_N) would be difficult to present and yielding no additional benefit, the representative scenarios with three different N_N were selected for presentation in this paper. A set of presented regular topologies above are useful in understanding the real advantage of NC. Two network topologies for N_N=4 and N_N=8 are presented in Figs. 2 and 3 respectively. Neighbour selection is mainly based on the node positions. For the simulation purposes all the links are symmetrical 1Mbit/1Mbit and are ideal, meaning that no packets get lost during transmissions and there is no additional delay due to propagation.

Traffic load is generated on all the nodes with the same intensity using exponential distribution of inter-arrival times and constant packet lengths (10 Kbit). Traffic distribution differs for Case 1 and Case 2, (results of both are presented in Section IV). In Case 1, all the nodes direct traffic to all nodes evenly (d1- distribution 1), while in Case 2 each node destines traffic to all nodes except to their neighbours (d2). By eliminating neighbouring nodes, as traffic destinations, a

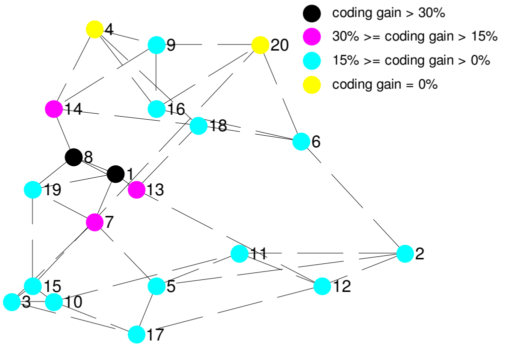

Figure 2. C4 network (N_N=4) presentation, with node categorization based on their coding success (d1, L1).

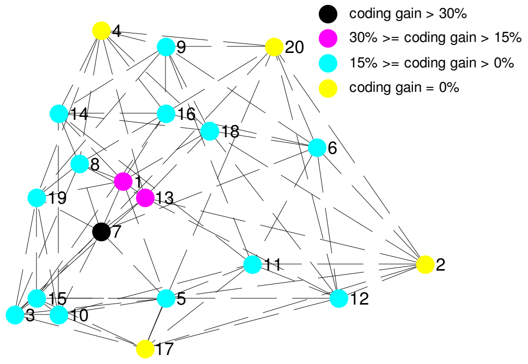

Figure 3. C8 network (N_N=8) presentation, with node categorization based on their coding success (d1, L1).

traffic distribution that is expected to affect coding opportunities has been introduced. Normalised network loads for Case 1 and Case 2 are presented in Table I and II, respectively. All the packets generated are upper layer packets (ULP) coming from layers above COPE.

TABLE I. HOP COUNTS (HC), DIAMETERS (D) AND NETWORK LOADS FOR ALL SIMULATION SCENARIOS FOR CASE 1

				Load(d1) (Mbits/s)			
		HC(d1)	***D***	***L1***	***L2***	***L3***	***L4***
Scenario	***C4***	2.76	7	6.9	4.6	3.4	2.7
	C6	2.01	5	9.4	6.3	4.7	3.8
	C8	1.76	4	10.7	7.2	5.4	4.3

TABLE II. HOP COUNTS (HC), DIAMETERS (D) AND NETWORK LOADS FOR ALL SIMULATION SCENARIOS FOR CASE 2

				Load(d2) (Mbits/s)			
		HC(d2)	***D***	***L1***	***L3***	***L5***	***L6***
Scenario	***C4***	3.23	7	5.9	3.9	2.9	2.3
	C6	2.48	5	7.6	5.1	3.8	3.1
	C8	2.32	4	8.1	5.5	4.1	3.3

Since coding opportunities depend on the amount of traffic in the network the load between the scenarios has been normalized in order to make scenarios comparable. The network with nodes that have more connections has smaller average hop distance to its neighbours. In scenarios described as d2, the packets make a longer trip to their destination. That means that with the same packet generator settings in the network with more connections per nodes would result in lower network load and in a case with different traffic distribution we would end up with different amount of network load. Thus, load (L) generated by traffic generators in the scenarios has been balanced according to the:

$$L_j * HC_{Ci} = const. \quad (1)$$

where HC is an average hop distance of traffic between sources and destinations in the network, j={1, 2, 3, 4, 5, 6} represents different loads, i=N_N and, Ci denotes scenario with different topologies (i.e., C4, C6 or C8) and stands for connection per node. In Table I and II, the six used network average hop counts (HC) are presented. In addition, we are presenting also network diameters (D). Ci stands for connections per node, i.e., N_N in the particular scenario.

Tables I and II also show the average traffic load generated in the upper layers at all the nodes together for a particular scenario. This resulted in the comparable network throughputs in scenarios using different N_N.

Routing of packets through the network was done using static tables which were calculated ahead of simulation runs. Routing tables are calculated using Dijkstra's algorithm taking into account distances between nodes.

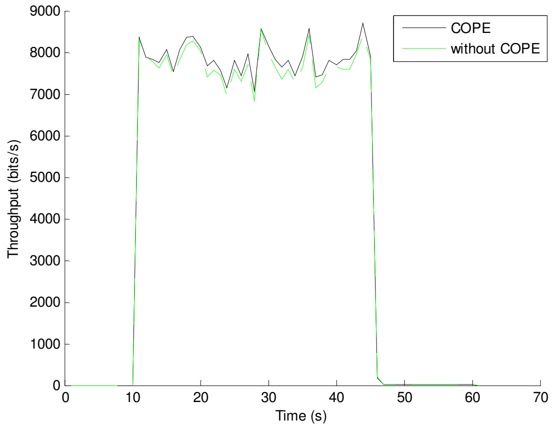

Figure 4. Network throughput for C6, L3, in case of using COPE and without COPE.

As seen from Fig. 4, every simulation run took 60 seconds. The network initialization phase took 10 seconds. The traffic was generated between the 10th and 45th second. Results were collected only between 11th and 44th second in order to observe steady state conditions. All the packets arriving after that time were not included into results. Running simulations for a longer period than loading the network with traffic allows easier recognition of a steady state conditions.

IV. SIMULATION RESULTS

The NC simulation has been used to obtain numerous simulation results considering different combinations of simulation parameters described in Section III. In the following, the most representative results are displayed, obtained for COPE scenarios and also for no-NC case scenarios as a reference (i.e., without COPE). With the help of all the presented results, one can also make a lot of conclusions on its own.

The results are presented in Figs. 5 and 6 for quality of service (user) point of view, and grouped in Tables III and IV according to their relevance for quantity of service (operators' point of view) and evaluation of COPE algorithm itself. As described in Section III, we have three different networks that differ in number of neighbouring nodes: 4 ('C4' in table), 6 ('C6'), and 8 ('C8'), namely. Each network has been loaded with four different traffic loads (L1 > L2 > L3 > L4) in a way that a complete network is handling the same network load for different number of connections (see Tables I and II).

A. Case 1 Traffic Distribution

First, we present results for the Case 1. As already mentioned in Section II, in this case, all the nodes send packets to all nodes evenly.

1) Average End-to-End delay

From the quality of service point of view we measured the average End-to-End delay in the network. Each packet that arrived to its destination node (between 10th and 44th second) was included in the statistics. We measured the delay between source node and destination node for every ULP packet. We have done so also in the scenarios without COPE. Delay results are represented graphically for all combinations of networks and loads in Fig. 5 (note logarithmic scale), where measurements are marked with signs, while connections between them have been made only for visualization purposes.

2) *Coding Gain*

From the COPE point of view we looked at COPE packet distribution and we have arranged COPE nodes according to their success in coding.

Packet distribution shows how much (%) ULP packets were encoded into the COPE packet (1 to 4 packets - there were no occasions of coding more than four packets into one COPE packet).

Node successfulness in coding further explains coding opportunities. We have divided nodes into four categories based on their success in coding packets together. Coding gain (G) for each node defined as the ratio between the number of source packets (without coding) N_S and the number of packets required to send source packets with coding N_C [3] has been used as a measurement:

$$G = \frac{N_S}{N_C} \tag{2}$$

For the gain representation, thresholds of coding gains have been set at 1.3, 1.15 and 1, representing 30 (higher G), 15 (medium G), and 0 (no gain) percent of packets being coded on particular node.

3) *Network Throughput*

Quantity of service results deal with network throughputs. We present the traffic throughputs as observed in the 802.11 layer. In Fig. 4, the network throughput for C6,

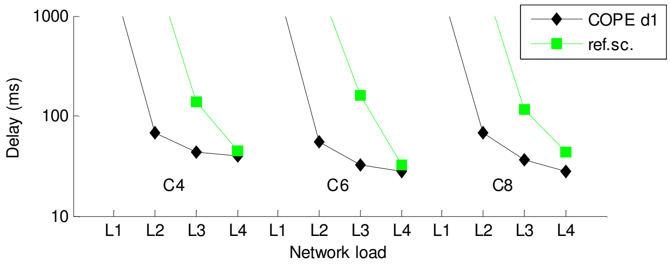

Figure 5. QoS of Case 1 - End–to–End delay for different loads in all of the observed topologies for nodes using COPE and for reference without COPE.

TABLE III. SIMULATION RESULTS FOR CASE 1

		COPE d1							**Quantity of service**			
		Packet distribution (%)			***Node successfulness in coding (Nr.)***				***Distribution of average network load (802.11) η(%)***			
		1 ULP	***2 ULP***	***3 or 4 ULP***	***G>30% for d1***	***15%<G<=30% for d1***	***0%<G<=15% for d1***	***G=0 for d1 (no coding)***	***COPE d1***	***without COPE***	***COPE d1***	***without COPE***
									(high)		***(medium)***	
C4	***L1***	78.8	21.2	0.0	2	3	13	2	60	65	20	10
	L2	85.8	14.2	0.0	2	2	15	1	35	40	25	20
	L3	92.7	7.3	0.0	0	2	14	4	15	25	30	20
	L4	95.6	4.4	0.0	0	0	16	4	5	15	20	10
C6	***L1***	80.4	19.6	null	3	4	9	4	70	65	25	25
	L2	90.4	9.6	null	1	2	13	4	40	40	25	20
	L3	95.2	4.8	null	0	1	13	6	15	15	30	30
	L4	97.2	2.8	null	0	0	14	6	5	10	15	15
C8	***L1***	81.5	16.7	1.8	3	4	9	4	70	70	30	30
	L2	89.6	10.0	0.4	1	2	13	4	30	35	30	25
	L3	94.9	5.0	0.1	0	1	13	6	20	20	15	15
	L4	97.1	2.9	0.0	0	0	14	6	10	10	15	15

L3 (random) scenario is presented as it varies thorough time during the simulation. Similar, the time variation in other case scenarios is behaving. In Tables III and IV, in “Distribution of average network load” column 802.11 load measurement results on all the links in both directions are presented. We have been interested in two types of links, those with high load (average load more than 90% of link capacity) and those with mint load (average load more than 70% and less than 90% of the link capacity). To be able to compare results between the scenarios we present results as ratio of links that meet criteria and all the links in the network:

$$\eta = \frac{Nr\ of\ Links\ to\ meet\ criteria}{Nr\ all\ links} \quad (3)$$

4) Results interpretation

From Fig. 5, we can see that in given conditions the delay is significantly smaller when COPE algorithm is used. This is especially notable when in the scenarios without COPE the network is not able to handle so high network load (noted also in figure with no mark, representing infinite delay). This is expected as with higher loads there are more packets in queues, increasing the chance for algorithm to find co-codable packets, thus taking them from the queue ahead their turn.

We can also notice that end-to-end delay is decreasing while increasing the number of connections nodes have. This is once more expected as the average hope distance between nodes is smaller in networks where nodes have more connections. Equally distributed load (from each node to all the nodes) per average has to take a shorter path to its destination, thus resulting in lower delay.

The results show that there are only few coding opportunities for a given traffic distribution. Even though in all scenarios for L1 the network was congested, only a fragment of packets from upper layers is codable with other

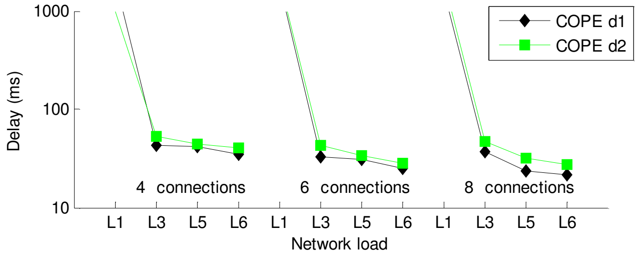

Figure 6. QoS of Case 2 - End–to–End delay for different loads in all of the observed topologies for nodes using COPE with two different traffic distributions.

TABLE IV. SIMULATION RESULTS FOR CASE 2

		COPE d1		**COPE d2**		***Node successfulness in coding (Nr.) for L1***						**Quantity of service**			
		Packet distribution (%)										***Distribution of average network load (802.11) η(%)***			
		2 ULP	***3 or 4 ULP***	***2 ULP***	***3 or 4 ULP***	***G>30% for d1***	***G>30% for d2***	***15%<G<=30% for d1***	***15%<G<=30% for d2***	***G=0 for d1***	***G=0 for d2***	***COPE d1***	***COPE d2***	***COPE d1***	***COPE d2***
												(high)		***(medium)***	
C4	***L1***	21.2	null	22.2	null	2	4	3	2	2	3	60	55	20	20
	L3	7.3	null	9.2	null	0	0	2	3	4	3	15	20	30	30
	L5	3.0	null	4.0	null	0	0	0	0	5	7	0	0	15	20
	L6	1.6	null	2.1	null	0	0	0	0	9	9	0	0	0	5
C6	***L1***	19.6	null	22.9	null	3	3	4	4	4	4	70	60	25	20
	L3	4.8	null	6.3	null	0	0	1	1	6	6	15	15	30	25
	L5	1.8	null	2.5	null	0	0	0	0	7	6	0	0	15	15
	L6	1.0	null	1.3	null	0	0	0	0	9	8	0	0	0	5
C8	***L1***	16.7	1.8	21.2	2.6	3	3	4	4	4	4	70	60	30	10
	L3	5.0	0.1	8.7	0.2	0	0	1	1	6	6	20	20	15	15
	L5	1.7	0.0	3.6	0.0	0	0	0	0	7	6	0	5	20	15
	L6	0.9	0.0	1.8	0.0	0	0	0	0	9	8	0	0	0	10

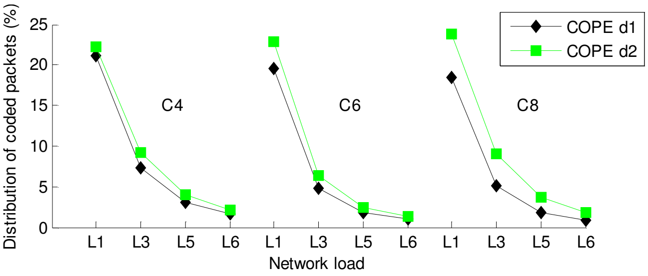

Figure 7. Case 1 and Case 2 - Share of COPE packets carrying more than one ULP packet.

packets (21.2 % is the highest rate in all scenarios). Also there were only few opportunities to code together more than two packets into one COPE packet. The results clearly show that coding opportunities for coding more than two packets together appear more often in networks where nodes have more connections (higher N_N). Furthermore, while increasing the number of connections per node in the network, share of packets carrying more than one upper layer packet decreases. This can to a certain point be explained with the load distribution: the average hop distance is lower, making average packet path shorter, while knowing that packets are in the first hop not codable. Though, more connections mean more overheard packets, more neighbours and thus higher chance of finding packets that can be coded together.

B. *Case 2 traffic distribution*

In the Case 2, we have tried to decrease the effect of smaller average network HC with networks that have nodes with more connections by introducing different traffic distribution (d2); all nodes send packets to all nodes except to their neighbours (see also Section III). In Fig. 6, we can see that end-to-end delay is increased with the d2 distributions. This is expected as the packets per average had to take a longer path than in d1 traffic distribution.

In Table IV, the results show that there are only few coding opportunities also for d2 traffic distribution, as compared to d1. In all scenarios for L1, the network was again congested, but only almost 23 % is the highest fragment of packets from upper layers that were codable with other packets. Furthermore, we can see in Fig. 7 that the load distribution has little effect on the coding opportunities. Even though, all the packets sent into the network are codable at least half their way there has been only a small increase of coded packets. The difference becomes apparent only with higher loads (e.g., L1). If compared to d1 in d2 there is almost no variation in coding success when number of node neighbours is changed, thus indicating that there is almost no difference in finding coding opportunities for different network topologies as used in this article.

C. *Summary of the results for Case 1 and Case 2*

For both traffic distributions we can see that in majority of scenarios there are only few nodes that perform very well or above average in coding. The number of these nodes slowly decreases with the number of connections nodes have. And quite the opposite with networks with higher number of connections per node there are more nodes that do not perform any coding at all (COPE packets carried only one ULP packet) or manage to code packets only occasionally. The situation can further be analysed in the Figs. 2 and 3 for d1, where nodes that are darker had more coding opportunities, while the lightest coloured nodes (i.e., yellow) have not coded any packets together. In Fig. 3, we have a network where each node has eight connections. Around the “node 7” (black node) there are two more nodes that perform majority of coding, that is also the part of the network that presents the bottleneck. Nodes that do not perform any coding are stationed on the network edges. Fig. 2 shows the same situation in case of network where nodes have four connections (N_N =4). There are two nodes that do well in coding packets and there are less nodes that do not perform any coding. If we divide the network into “subnetworks” we can once more claim that nodes that do not perform coding are placed at the network edges, while nodes that perform better, act as links between individual “subnetworks”. For d2, the results of analysis are in line with d1 analysis.

Even though the total network throughput is increased the COPE helps “balancing” the network load. This can be seen from Tables’ III and IV column “Distribution of average network loads”. Results show that in case of higher loads, COPE is able to distribute the traffic amongst more nodes and thus keep the network congestion free. There are more nodes in COPE case with medium loads than in case without COPE. This means that congestions happen when coding is not used and traffic is delayed, thus making links that are not congested even less loaded. COPE, in the case of sudden bursts, finds more coding opportunities, thus coding more packets together, while using the same bandwidth and thus avoiding congestion.

V. CONCLUSIONS AND FUTURE WORK

We evaluated NC algorithm COPE using different topologies, traffic intensities and distributions. The results show that COPE NC algorithm improves the network throughput in all analysed networks with given traffic distributions. Most importantly, in the high load situations when in case without COPE the network can not carry the load and thus the delay is infinitive the network still handles its load. Results once more show that this intuitive algorithm COPE really helps improving the network capacity.

Still, our results show that there is a lot of space for improvement. In our random traffic load conditions only a fragment of packets found pairs to be coded to. Even more, cases when more than two packets got coded together hardly ever happened.

We have been looking for correlation between success ratio in coding and number of neighbours that nodes have in the network. Theoretically nodes with more neighbours should be able to code together more packets, thus reaching higher coding gains. Results of the simulations show the opposite; networks where nodes had fewer neighbours had more success in coding, by comparable network load conditions. This happens as hearing more neighbours means having more connections per node when distributing network traffic. In such situation, nodes have lower average hop distance to destinations and thus fewer chances to code packets together.

Analysis of individual nodes and their success in coding also reveals that there are only few nodes that perform coding operations, while there were a lot of nodes that did not manage to find any co-codable packets. Moreover, nodes that are placed in the heart of the network and do have many neighbours did not find coding options for coding more than three packets into one COPE packet. Overall, we had one to three nodes per network that were very successful in coding but up to 30% of nodes that have not managed to perform any coding operations.

Overall, results show that coding opportunities arise with increased traffic. In highly loaded network, there is significantly less coding opportunities than in congested network.

In depth analysis of NC can help us better understand where and why coding opportunities appear. The results imply that implementation of NC can result in modifications and possibly new approaches in NC, affect heavily routing concepts, or just provide guidelines on network planning is still a question.

In our opinion, mainly based on presented results the biggest opportunity for NC lies in strong collaboration with routing. If traffic is acting randomly as in our case, NC can not reach its full potentials, therefore different approaches in traffic organisation should be looked for, thus making routing be aware of coding opportunities, rather than being oblivious to it.

As future work, we consider implementing other existing NC algorithms for further evaluation of NC principles. Furthermore, on the basis of the results and analyses presented in this article we intend to develop new NC algorithms and routing schemes for NC. With respect to the simulation model, additional modules, which will take into account also more realistic wireless links, will be implemented.

REFERENCES

[1] 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 2000.

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

[3] 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, June 2008.

[4] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *2007 conference on Applications, technologies, architectures, and protocols for computer communications*, 2007.

[5] D.-C. Tomozei, T. Salonidis, L. Massoulie, L. Tassiulas, L. Georgiadis, I. Broustis, et al., "Wireless protocols using network coding: report on on state-of-the-art," retrieved July, 2009.

[6] K. Chi, X. Jiang, and S. Horiguchi, "Network Coding Opportunity Analysis of COPE in Multihop Wireless Networks," in *IEEE Wireless Communications and Networking Conference, 2008 (WCNC 2008)* Las Vegas, NV, USA, 2008.

[7] B. Scheuermann, W. Hu, and J. Crowcroft, "Near-optimal co-ordinated coding in wireless multihop networks," in *2007 ACM CoNEXT conference*, 2007.

[8] 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.

[9] C. Qin, Y. Xian, C. Gray, N. Santhapuri, and S. Nelakuditi, "I2MIX: Integration of Intra-flow and Inter-flow Wireless Network Coding," in *5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2008. SECON Workshops '08*, 2008.

[10] 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, June 2010.

[11] Z. Zhou and L. Zhou, "Network Joint Coding-Aware Routing for Wireless Ad Hoc Networks," in *2010 IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*, 2010, pp. 17-21.

[12] L. Yifei, S. Cheng, X. Qin, and T. Jun, "ICM: a novel coding-aware metric for multi-hop wireless routing," in *WiCOM'09 Proceedings of the 5th International Conference on Wireless communications, networking and mobile computing* NJ, USA: IEEE Press, 2009.

[13] K. Alic, E. Pertovt, and A. Svigelj, "COPE Simulation model," in *20th Electrotechnical and Computer Science Conference (ERK 2011)* Portoroz, Slovenija: IEEE Region 8, IEEE Slovenian section, 2011.

[14] K. Alic, E. Pertovt, and A. Svigelj, "Simulation Environment for Network Coding," in *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies 2011 (AEECT 2011)* Amman, Jordan, 2011.

[15] "OPNET web page," retrieved July, 2012.

[16] D. Bertsekas and R. Gallager, *Data Networks*: Prentice-Hall International Editions, 1987, pp. 322-323.

[17] "Matlab web page," retrieved July, 2012.