

## Bearing-Opportunistic Network Coding

K. Alic, E. Pertovt, A. Svirgelj

**Kemal Alic\***, Erik Pertovt, Ales Svirgelj

Department of Communication Systems, Jozef Stefan Institute

Jozef Stefan International Postgraduate School

Jamova cesta 39, 1000 Ljubljana, Slovenia

kemal.alic@ijs.si, erik.pertovt@ijs.si, ales.svirgelj@ijs.si

\*Corresponding author: kemal.alic@ijs.si

**Abstract:** This paper presents a novel, practical, routing-independent network-coding algorithm: BON-Bearing opportunistic network coding. Simplicity is its main benefit as it introduces little overhead to the network since nodes do not need to keep track of received traffic for their neighbouring nodes. Algorithm makes coding decisions based solely on the information about the packet previous and next hop node position. Algorithm functions between the MAC and link layers, with small modifications made only to the MAC layer. Using different topologies and different traffic loads and distributions in the simulation model we evaluated algorithm performance and compared it to a well-known COPE algorithm.

**Keywords:** Algorithms, network layer network coding, network throughput, performance evaluation.

## 1 Introduction

Network coding (NC) introduced by Ahlswede [1] brings a promising approach to improving network throughput and performance. The classical communication networks paradigm of nodes only forwarding packets is metamorphosed by a simple and important premise that nodes also process the incoming packets [2].

NC was originally proposed in order to achieve multicast data delivery at the maximum data transfer rate in single-source multicast networks [3]. We soon began to see NC ideas being put to use in problems other than multicast networks. Increasing throughput in satellite networks [4, 5] and P2P networks [6, 7], improving delivery reliability over the lossy links either in wireless networks over TCP [8] or in Delay Tolerant Networks such as deep space links [9] all bring promising results. Depending on the application of NC the implementation affects different OSI layers. In multicast scenarios NC is typically implemented in the application layer while two stage NC for increased spectrum efficiency is deployed in the physical layer [10].

We are interested in opportunistic NC for static wireless mesh networks such as metropolitan WiFi networks [11] for unicast traffic. Benefits of opportunistic NC can be best explained with a help of a simple example depicted in in Figure 1. Consider the following situation: Node 1 (N1) has a packet P1 for Node 2 (N2) and Node 2 (N2) has a Packet P2 for N1. The nodes can exchange their packets through intermediate node (N3). N1 and N2 send packets P1 and P2 to N2. Without using NC the N3 first sends out P1 and later on P2. By using the NC procedure N3 performs a linear operation over the two packets (e.g. XOR) and sends out a coded packet  $P1 \oplus P2$ . N1 and N2 XOR the received packet with the sent packet (P1 and P2) and obtain the packets headed to them. With NC, the number of transmissions nodes need to perform in order to deliver both packets to their destinations has been reduced from 4 to 3.

The question which packets the coding node can encode together in general network deployment in order that the receiving nodes will be able to decode the coded packet, which is the key issue in this paper. Optimal integration of NC into the existing architecture of a network is not straightforward and its implementation influences all functional components in current

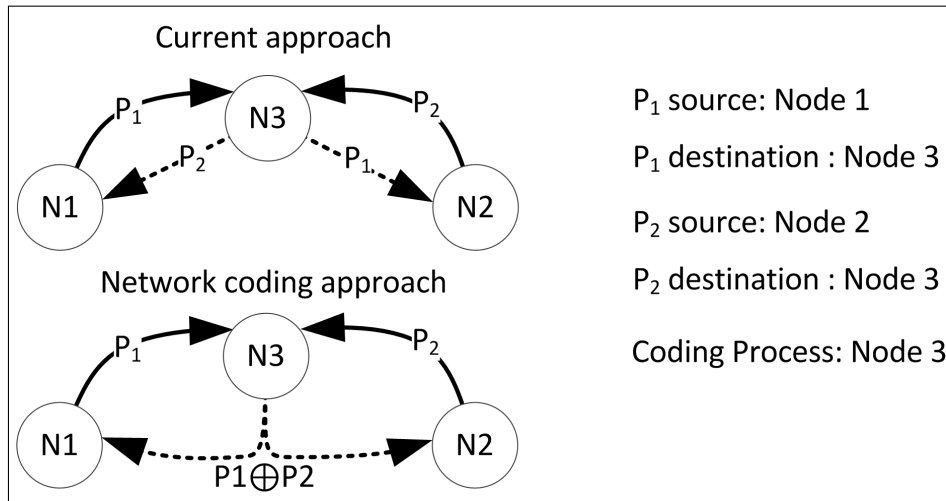


Figure 1: Simple example on how opportunistic NC increases throughput.

network protocol stack e.g. scheduling, routing and congestion control [12]. Though, interaction with all these layers increases the implementation difficulty and possibly also the computational complexity. Henceforth, a trade-off needs to be made between coding and complexity. Hence, the objective of this paper is to propose a novel inter-session NC schema that is easily implementable in the real world, can be used in the current OSI structure and improves the network performance.

Thus, in this article we propose a novel Bearing-Opportunistic Network coding (BON) algorithm, which main advantages are the following: (i) it can be seen as an individual OSI layer between the MAC and the network layers; (ii) it introduces little overhead to the network; (iii) it works for all traffic types; (iv) it is distributed; (v) it does not need to record information about the received packets on individual neighbouring nodes and it is (vi) routing independent.

The rest of this paper is organized as follows. In Section 2 related work is described. In Section 3 the BON coding process is explained and its implementation into OSI model is explained in Section 4. Performance evaluation results are given in Section 4. Section 5 concludes the article and gives insight into further work.

## 2 Related work

Practical NC has mainly been influenced by an excellent and intuitive algorithm COPE (Coding Opportunistically) [13] which principles have been adapted and extended also for covering other ideas in NC: for example in [14] noCoCo algorithm specializes in bidirectional traffic flows. It is trying to maximize the number of coding opportunities for the two opposite direction routes. CLONE [15] generalized COPE to address multiple unicast sessions. The system takes into account lossy links and highlights specific situations where COPE provides no coding gain. COPE coding decisions are classified and improved thus making better coding decisions by K. Chi et.al [16]. In [17] 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 [18] and [19], while [20] investigated new metric schemas for coding aware wireless routing. In DCAR architecture [21] coding is based on COPE, though extended and supported by route discovery.

COPE procedure introduced NC between the MAC and network layers with small modifications made only to the MAC layer. COPE takes advantage of the broadcast nature the wireless

channel to perform opportunistic listening and encoded packet broadcasting so that the number of packet transmissions can be reduced.

In COPE, every node keeps information for each of its neighbours about packets already received. Information can be gathered through ACK packets and through update packets. The later introduces additional overhead as update packets are broadcasted by individual nodes to all their neighbours. Owing to these facts, nodes require additional hardware functionalities and the network requires handling of additional packets which on one hand results in lower network capacity, and is on the other hand undesirable in energy constrained networks such as Wireless Sensor Networks (WSN). A fair share of coding decisions in COPE is based on the delivery probabilities between the nodes. These are calculated in the routing process in the routing protocols based on the ETX metrics. Hence for COPE implementation access to the routing layer is needed, which is not always possible, especially when combining equipment from different vendors.

### 3 Network Coding Process

Essentially the Coding Process is about making decisions on which packets the coding node can encode together in order that the receiving nodes will be able to decode the coded packet. The more packets we code together or more loose the conditions in coding process are the higher the possibility of receiving nodes not being able to decode packets. Though, if coding conditions are hard to meet, than there are few coding opportunities and low bandwidth saving benefits.

#### 3.1 General coding conditions

BONCoding process is based on the local bearing of packets. We define packet bearing on a coding node as the unit vector showing direction between previous hop and next hop of the packet.

Let us consider the situation depicted in Figure 2. Packet P1 has already been transmitted from node 1 (N1) to node 2 (N2) and is destined to node 3 (N3). Packet Pn has already been transmitted from node 4 (N4) to N2. P1 and Pn are now waiting in the output queue on the coding node N2. When N2 gains channel access, it tries to encode P1 and Pn based on the local bearing process. On the coding node N2 local bearing of the packet P1 depends on the positions of its previous hop N1 and its next hop N3. Since the coding node is familiar with the node positions of the packet previous hop and the packet next hop, the coding node can calculate the direction vector  $\vec{a}$  which represents the direction in which the packet P1 is travelling. In Cartesian coordinate system we calculate the angle  $\alpha$  for P1 between the x-axis and the direction vector  $\vec{a}$  which is between 0 and  $2\pi$ .

After calculating the local bearing for the first packet P1, the algorithm goes through the remaining packets in the output queue searching for a matching packet to be encoded with P1. For each of the possibly matching packets Pn the angle  $\beta$  between the x-axis and its direction vector  $\vec{b}$  are calculated using the same principle as described above for the first native packet P1. Coding conditions are met for packets P1 and Pn, if (1) is true:

$$\pi - \epsilon \leq |\alpha - \beta| \leq \pi + \epsilon \quad (1)$$

where  $\epsilon$  is the tolerance angle. In a given situation N1 can decoded packet since it is the source of half of the information encoded in the  $P1 \oplus Pn$ . The question is whether N3 can decode this packet. The probability that N3 will successfully decode  $P1 \oplus Pn$  is the same as probability that N3 has overheard the Pn transmission from N4 to N2. We may assume that nodes on one side of the coding node have the knowledge of the content of packets of the first flow, while nodes

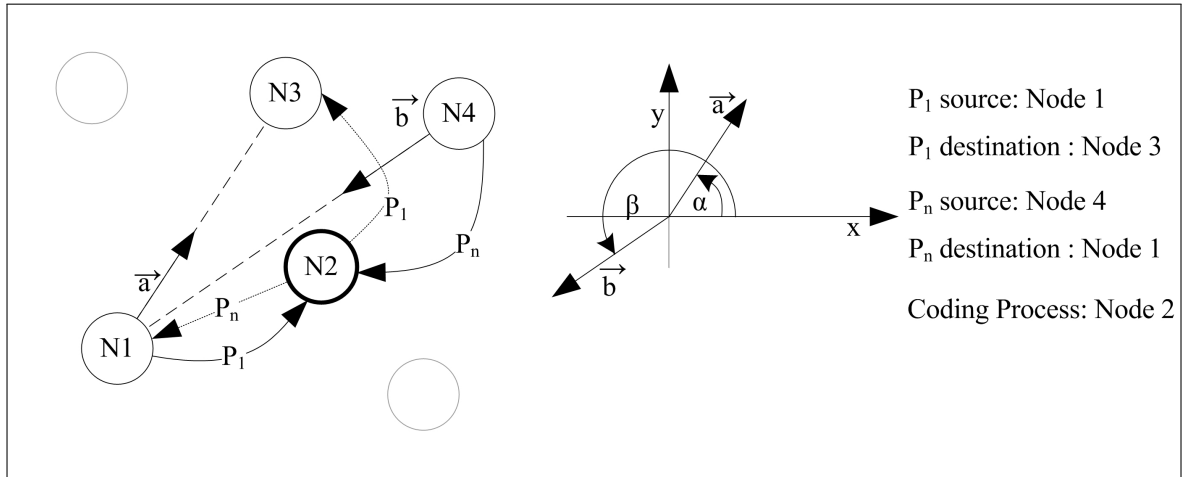


Figure 2: Coding process: the calculation of parameters for encoding evaluation for two packets.

on the other side of the coding node recognise the content of packets from the second flow, thus matching them up for encoding. Nodes that are placed in approximately the same bearing from the coding node, can also hear each other transmissions, thus being able to decode the packet. By increasing  $\epsilon$  the possibility of encoding multiple packets and that one of the receptionists will not be able to decode the packet is increased. By lowering  $\epsilon$  towards zero coding opportunities are reduced but the possibility of successful packet decoding on receiving nodes is increased. When two packets belonging to two traffic flows that are headed into opposite directions meet on the relay node, (1) is true even for  $\epsilon = 0$  and receiving nodes will always be able to decode the encoded packet.

BON allows coding of multiple packets. Multiple packets are encoded into one packet when the expression (1) is true for all packet pairs to be encoded. If  $\epsilon = 0$  two packets can be encoded at the most.

## 4 Network Coding Procedure

### 4.1 Packet coding

Each node maintains two packet queues. The highest priority queue is used for signalization packets and the lower priority queue is for packets carrying information through the network. If signalization queue is empty the coding algorithm takes the packet that is at the head of the queue and then searches the rest of the queue looking for possible coding matches based on the process described in Section 3. If a coding opportunity is found, the packets are encoded using XOR operation into a coded packet accompanied with the headers for the decoding process at the receiving nodes. If no coding opportunities are found the packet is transferred to the MAC layer as is and the native packet is transmitted.

### 4.2 Pseudo-broadcast

In the MAC layer the pseudo-broadcast mechanism, first presented in [13] is used. The link-layer destination field is set to the MAC address of one of the intended recipients. Since all 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 message to

the sender. Regardless of the address of the packet next hop the node pushes the packet to the NC module.

### 4.3 Packet reception

Upon the packet reception in the NC module further actions depend on whether the packet is coded or native (not coded). In the case of the coded packet the process checks the packet pool where all received and overheard packets are stored for decoding purposes to determine whether it has already received  $N-1$  of the packets coded in the coded packet. If not the coded packet cannot be decoded and it is simply dropped. If the node has at least the required  $N-1$  packets, i.e., enough information, it decodes the coded packet using these packets with the XOR operations, thus gaining a set of native packets. Each native packet is treated individually. From here on the process is the same as upon receiving a native packet. The process checks whether the node has already received the packet. If so it drops it. If the packet is new its copy is inserted into the packet pool for decoding purposes. It does so for every received native packet, as all received packets are potentially needed for further decoding purposes. The process checks whether the node is the next hop for the native packet. If so, and if the packet has been a part of the coded packet, ACK message is scheduled and the packet is sent to the upper layers for further processing.

### 4.4 Signalization

Since static nodes are under investigation, ACK messages are the only signalization packets required in BON. Depending on the application and expected gains, different ways of sending ACK messages can be used. Since we are optimising a network from the throughput perspective we adopt the structure proposed in [13]. ACK messages are periodically broadcasted as cumulative reports. If opportunity arises the ACK reports are attached to the regular outgoing packets, thus reducing the overhead.

## 5 Performance Evaluation

### 5.1 Simulation Setup

We evaluated BON using a simulation model built in OPNET Modeler [22] for analysis of network layer NC algorithms that has already been presented in [23].

We conducted a set of simulations using the following settings: 40 static nodes randomly placed on 4 km x 4 km area; 800, 900 and 1000 m of transmission range for three topologies T1, T2 and T3, respectively. The evaluated topologies are presented in Figure 3, where dashed lines indicate wireless links between nodes. All nodes receive and transmit on the same channel and each node has 2 Mbit/s of channel bandwidth.

BON and COPE use up to one retransmission in the wireless module and up to two retransmissions in the NC layer. Both algorithms store packets in the packet pool for 20 seconds after the reception, thus making sure packet decoding did not fail due to delays in the network. Output queues are limited to 50 packets.

BON and COPE send out cumulative ACKs every 0.5s. NC layer packet retransmissions are scheduled 0.8 s after the initial packet transmission. COPE update packets as described in [13] are sent out at least every 0.5 s. When possible, update packets are attached to regular outgoing packets, or cumulative ACKs thus introducing less overhead to the network.

$\epsilon$  has been set at 25 for BON, while  $\epsilon$  for evaluating coding condition in packet retransmission has been set to half of the initial value.

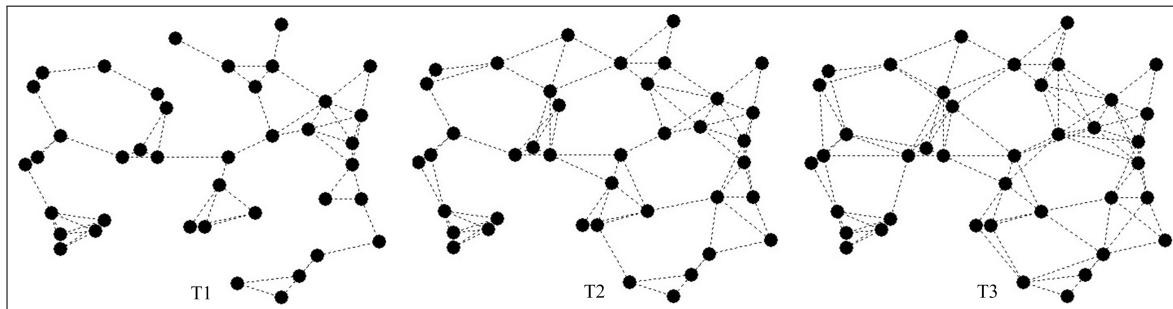


Figure 3: Wireless nodes connectivity for three evaluated topologies (namely T1, T2 and T3).

Simulation runs took 150 s. Results were collected between 60th and 120th second to observe steady-state conditions. Through each simulation run we observed network with one load. For each load we made three simulation runs using three different seeds. For the first 30 s traffic load was set at 10 % of the final load, between 30th and 60th second the load was set to 80 % of the final load (i.e. warm up time). Network was loaded with full load and results were collected between the 60th and 120th second. Routing tables are calculated at the beginning of each simulation run and are updated every 30 s. Dijkstra algorithm is used for routes calculation taking into account ETX metric, which is also required by COPE.

UDP like traffic has been generated. Packet sizes vary between 45 and 1500 bytes. Packet size distribution has two peaks and follows the measured Internet traffic as presented in [24]. Exponential distribution is used for calculating packet inter-arrival times.

Two traffic flow distributions have been used in the process of evaluation. The first, the Uniform distribution scenario, is where all nodes generate traffic flows and destine them to all nodes in the network (uniform distribution is used to generate packet destinations). In the second, the Gateways scenario, nodes communicate with its nearest gateway only. Nodes and gateways generate symmetric traffic flow. Three selected nodes from topologies have been given the base station functionality. Delivery probability (DP) between nodes depends on the amount of traffic between the neighbouring nodes and the distance between nodes (e.g. for T3, load 1.9 Mbit/s, uniform traffic distribution, BON algorithm average DP is 0.88, max DP 0.97, min DP 0.78 and median for DP is 0.87).

We have compared BON algorithm to the well-known COPE and to reference scenario (ref.sc.) where no coding was used.

## 5.2 Simulation results

For Uniform traffic flow distribution for all topologies we have observed average goodput (Figure 4) highlighting results important from the operator point-of-view while End-to-End (ETE) Delay (Figure 5) shows results important from the end-user perspective. Normalised Number of Wireless Transmissions (NNWT) which is the ratio between the number of packets received from NC layer by the wireless module and number of packets received in the application layer (Figure 6) shows the number transmissions needed in the wireless module for every goodput packet. In ideal link conditions NNWT is a sum of network average diameter and overhead in terms of standalone packets. We use this measure for overhead comparison reasons. Number of packet retransmissions in the NC layer indicates how well NC algorithms coded packets together (Figure 7). Maximal normalised gain which is the highest ratio between the number of goodput packets received using coding algorithm and the number of goodput packets received in ref. sc. (Table 1) shows highest measured gain when using NC. Each dot on the plot represents the average of three simulation runs with different seeds of pseudo random generator.

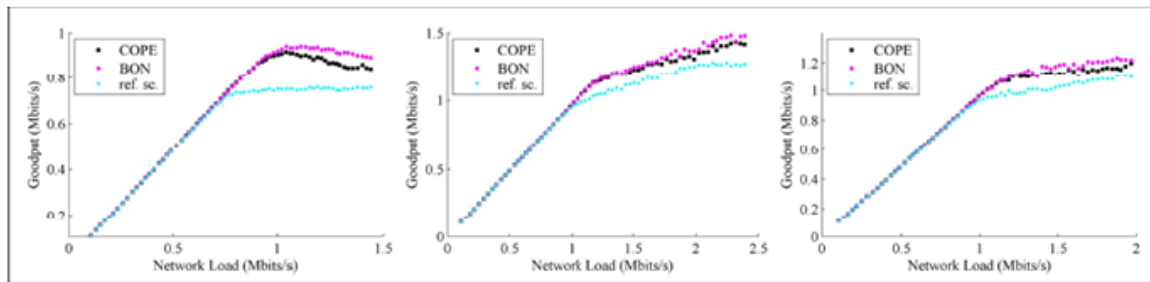


Figure 4: Goodput in dependency of network load for BON, COPE and ref.sc. for topologies T1, T2 and T3 in Uniform traffic distribution.

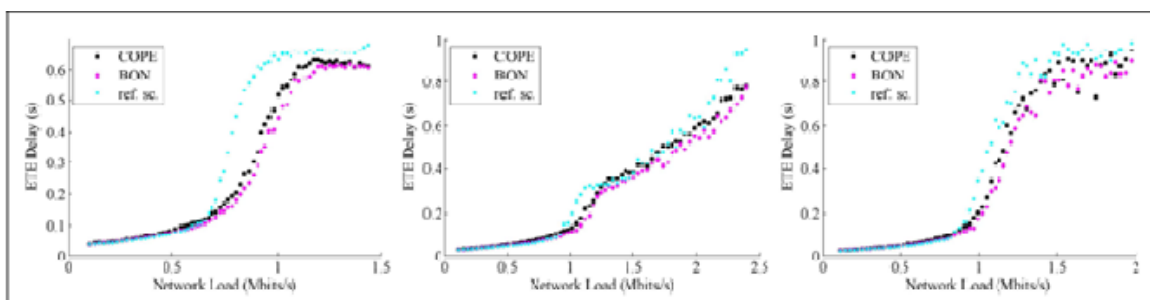


Figure 5: Delay in dependency of network load for BON, COPE and ref.sc. for topologies T1, T2 and T3 in Uniform traffic distribution.

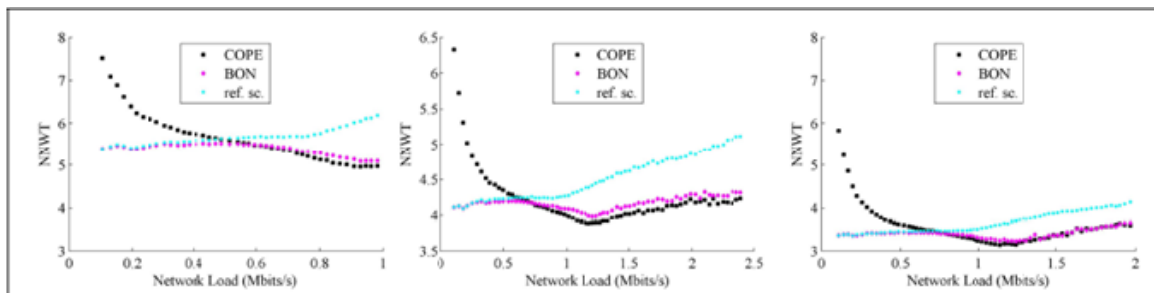


Figure 6: Normalised number of wireless transmissions in dependency of network load for BON, COPE and ref.sc. for topologies T1, T2 and T3 in Uniform traffic distribution.

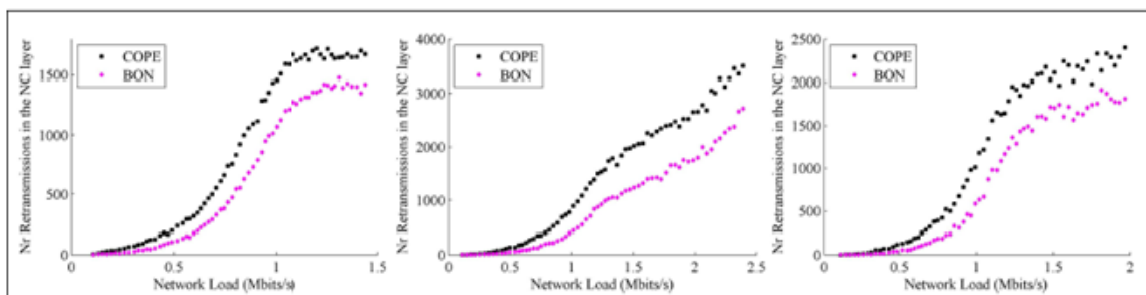


Figure 7: Nr Retransmissions in the NC layer in dependency of network load for BON and COPE for topologies T1, T2 and T3 in Uniform traffic distribution.

Table 1: Maximal normalised gain for BON and COPE for T1, T2 and T3 in Uniform traffic distribution.

	Maximal normalised gain		
	T1	T2	T3
COPE	1.22	1.12	1.14
BON	1.26	1.17	1.17

From the Goodput in dependency of network load plot (Figure 4) for all topologies we can observe that with low traffic loads goodput is approximately the same for both coding algorithms as well as with the reference scenario. However, by increasing the traffic load, we can see that both COPE and BON significantly improve the network performance as compared to the ref.sc Scenario, and that BON can provide the highest goodput. The gain itself also depends on the network load and observed topology. Table 1 shows that the Maximal normalised gain (ratio between number of packets received using coding algorithm and number of packets received in ref. sc) and that BON outperforms COPE for additional 4 to 5 percentage points.

From the End-to-End Delay in dependency of network load plot (Figure 5) we can observe that with low traffic loads the delay is low and approximately the same for BON and the reference scenario, while delay with COPE is slightly higher. By increasing the network load the delay increases for all three scenarios. The delay increases the fastest with ref.sc. COPE follows and BON keeps the lowest delay of the three for almost all loads and all three topologies. Higher delay with COPE in lower loads can be explained with the help of plots in Figure 6 (Normalised number of wireless transmissions). In low traffic loads COPE finds few opportunities to attach update packet to regular outgoing packets thus introducing higher overhead to the network. With the increased load COPE and BON need to transmit approximately the same number of packets in the wireless module in per application layer packet. COPE keeps the ratio slightly lower than BON with high loads. This is due to the fact that BON outperforms COPE in throughput gains and due to limited queue sizes. Packets travelling higher node distances are more likely to be dropped due to full queues. The ratio for both algorithms is much better in the congested network where NC benefits are at its best.

BON better performance in comparison to COPE can be explained also with the number of retransmissions in the NC layer. Retransmissions in the NC layer occur in case of reception node unsuccessful decoding or errors in the transmission (pseudo broadcast demands ACK message only from one of the recipient nodes). Hence, number of retransmitted packets indicates how successful the algorithm was in coding together packets that can be decoded on their next hop: BON needs fewer retransmissions than COPE and we may claim that it makes better coding decisions.

In Figure 8 and Figure 9 the results for the Gateways scenario are presented. Results were collected in the same way as for Uniform traffic case. We present only results for network goodput and End-to-End Delay as this are the most representative results; other results bring up the same conclusions as in the Uniform distribution scenario. In comparison to the Uniform traffic flow distribution coding gain benefits are lower for the flows directed from and to the gateways. This is due to the changed traffic conditions. Packets travel shorter hop distances between their source and destination, providing less coding opportunities and thus fewer possibilities for goodput improvements. Still, the BON algorithm can provide the highest goodput while keeping the delay lowest. Table 2 shows the Maximal normalised gain and that BON can bring additional 2 to 8 percentage points of goodput increase in comparison to COPE.



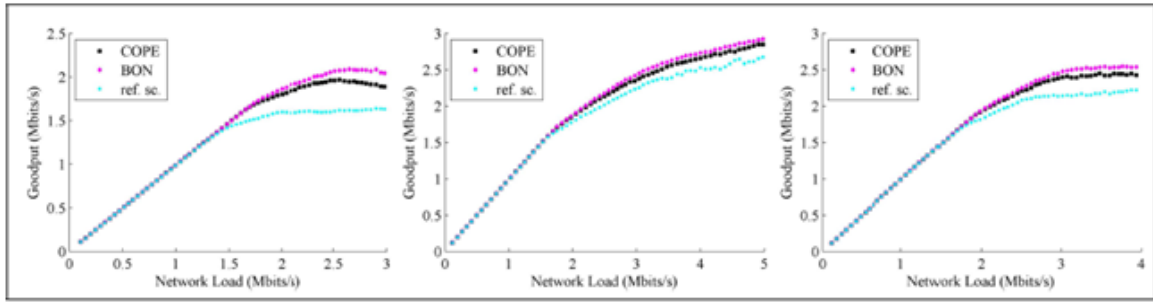


Figure 8: Goodput in dependency of network load for BON, COPE and ref.sc. for topologies T1, T2 and T3 in Gateways traffic distribution.

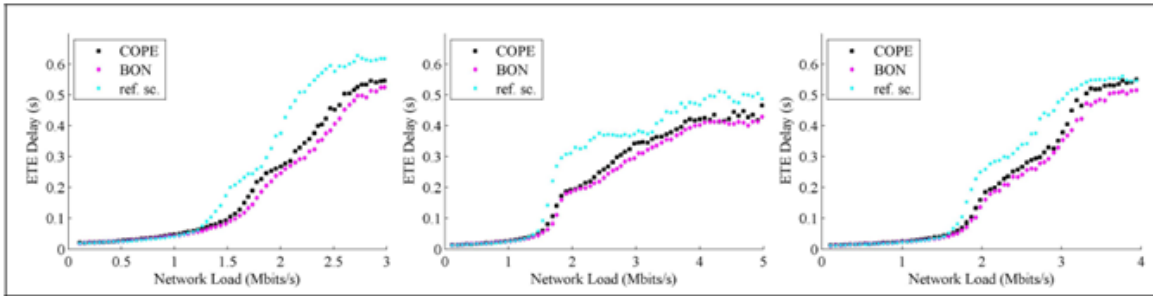


Figure 9: Delay in dependency of network load for BON, COPE and ref.sc. for topologies T1, T2 and T3 in Gateways traffic distribution.

Table 2: Maximal normalised gain for BON and COPE for T1, T2 and T3 in Gateways traffic distribution.

	Maximal normalised gain		
	T1	T2	T3
COPE	1.21	1.09	1.13
BON	1.29	1.11	1.18

## 6 Conclusions and further work

BON is a novel, general-purpose algorithm which works in the broadcast networks between the network and the MAC layer. It is applicable to the static networks such as WSN networks and metropolitan WiFi networks. Small modifications are made only to the MAC layer while other OSI layers remain unchanged (the same as in COPE). We assume that each node is familiar with positions of all of its neighbours which is easy to obtain as nodes are static.

BON is routing independent and as presented in the results it introduces very little overhead to the network, thus using less radio resources and consequently less energy. For nodes without any coding opportunities or generally in networks with low loads the overhead is the same as in the reference scenario. Since nodes do not need to save the information about packets received on neighbour nodes, the coding process is fairly simple. In addition, we have shown in the results that BON outperforms COPE in terms of increasing the network goodput while keeping the network delay lower for different topologies and traffic distributions.

In-depth analysis shows that COPE finds more coding opportunities, though incorrect coding decisions are more often made than in BON. Furthermore, lower overhead in BON also results

in higher good put.

Main drawback for using BON is its requirement of knowing the positions of the nodes; although the networks where nodes know their position are increasing rapidly e.g. in backhaul networks such as metropolitan WiFi where end-users use different part of spectrum for communication.

In addition, setting the tolerance angle might not appear straightforward. For static networks, such as in our case, this can be done through simulation or emulation, though through experience the approximate value of the optimal  $\hat{\Gamma}$  can be set by carefully observing the topology. By adapting the algorithm to automatically set the tolerance angle on individual nodes, planning as a future work, seems an important improvement towards additional practical value of BON.

BON low overhead makes BON suitable for studying energy efficiency in wireless sensor or similar networks, especially in load cases when network is not congested. Besides that, its routing independency shows potential for modifying routing metrics.

## Bibliography

- [1] R. Ahlswede, N. Cai, S.-y. R. Li, and R. W. Yeung (2000); Network Information Flow, *IEEE Transactions on Information Theory*, 46: 1204-1216.
- [2] C. Fragouli and E. Soljanin (2008); *Network Coding Applications*, Hanover, MA, USA Now Publishers.
- [3] T. Matsuda, T. Noguchi, T. Takine (2011); Survey of Network Coding and Its Applications, *IEICE Transactions on Communications*, vol. E94-B, 698-717.
- [4] C. Hausl, O. Ican, and F. Rossetto (2012); Resource allocation for asymmetric multi-way relay communication over orthogonal channels, *EURASIP Journal on Wireless Communications and Networking*, 1-20.
- [5] F. Vieira, S. Shintre, and J. A. Barros (2010); How feasible is network coding in current satellite systems?, *Advanced satellite multimedia systems conference (asma) and the 11th signal processing for space communications workshop (spsc)*, Cagliari, Italy, DOI: 10.1109/ASMS-SPSC.2010.5586880, 31-37.
- [6] D. Niu and B. Li (2011); Asymptotic Optimality of Randomized Peer-to-Peer Broadcast with Network Coding, *INFOCOM, Shanghai, China, 2011*, 1-9.
- [7] B. Li and D. Niu (2011); Random Network Coding in Peer-to-Peer Networks: From Theory to Practice, *Proceedings of the IEEE*, 99: 513-523.
- [8] J. Chen, L. Liu, X. Hu, and W. Tan (2011); Effective Retransmission in Network Coding for TCP, *International Journal of Computers Communications & Control*, 6(1):53-62.
- [9] S. Haoliang, L. Lixiang, and H. Xiaohui (2011); A Network Coding based DTN Convergence Layer Reliable Transport Mechanism over InterPlaNetary Networks, *International Journal of Computers Communications & Control*, 6(2):236-245.
- [10] K. Yasami, A. Razi, and A. Abedi (2012); Analysis of Channel Estimation Error in Physical Layer Network Coding, *IEEE Communications Letters*, 15: 1907-1910.
- [11] S. Chiochan and E. Hossain (2012); Network coding for unicast in a WiFi hotspot: Promises, challenges, and testbed implementation, *Computer Networks*, 56(2): 2963-2980.

- [12] L. You, L. Ding, P. Wu, Z. Pan, H. Hu, M. Song, et al. (2011); Cross-layer optimization of wireless multihop networks with one-hop two-way network coding, *Computer Networks*, 55(8): 1747-1769.
- [13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft (2008); XORs in the Air: Practical Wireless Network Coding, *IEEE/ACM Transactions on networking*, 16(3): 497-510.
- [14] B. Scheuermann, W. Hu, and J. Crowcroft (2007); Near-optimal coordinated coding in wireless multihop networks, *2007 ACM CoNEXT conference, 2007*, DOI: 10.1145/1364654.1364666.
- [15] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta (2008); Loss-aware network coding for unicast wireless sessions: design, implementation, and performance evaluation, *ACM SIGMETRICS Performance Evaluation Review - SIGMETRICS '08*, 36(1): 85-96.
- [16] K. Chi, X. Jiang, B. Ye, and Y. Li (2011); Flow-oriented network coding architecture for multihop wireless networks, *Computer Networks*, 55(10): 2425-2442.
- [17] C. Qin, Y. Xian, C. Gray, N. Santhapuri, and S. Nelakuditi (2008); I2MIX: Integration of Intra-flow and Inter-flow Wireless Network Coding, *5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2008. SECON Workshops '08*, DOI: 10.1109/SAHCNW.2008.29.
- [18] Y. Yan, B. Zhang, J. Zheng, and J. Ma (2010); CORE: a coding-aware opportunistic routing mechanism for wireless mesh networks, *IEEE Wireless Communications*, 17(3): 96-103.
- [19] Z. Zhou and L. Zhou (2010); Network Joint Coding-Aware Routing for Wireless Ad Hoc Networks, *2010 IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS), 2010*, DOI: 10.1109/WCINS.2010.5541877, 17-21.
- [20] L. Yifei, S. Cheng, X. Qin, and T. Jun (2009); ICM: a novel coding-aware metric for multi-hop wireless routing, *WiCOM'09 Proceedings of the 5th International Conference on Wireless communications, networking and mobile computing* NJ, USA: IEEE Press, DOI: 10.1109/WICOM.2009.5302189, 1-4.
- [21] J. Le, J. C. S. Lui, and D.-M. Chiu (2010); DCAR: Distributed Coding-Aware Routing in Wireless Networks, *IEEE Transactions on Mobile Computing*, 9(4): 596-608.
- [22] <http://www.riverbed.com/products/performance-management-control/opnet.html>.
- [23] K. Alic, E. Pertovt, and A. Svirgelj (2012); Network coding simulation model in OPNET modeler, in *OPNETWORK 2012*, Washington, USA.
- [24] A. Svirgelj, M. Mohorcic, G. Kandus, A. Kos, M. Pustisek, and J. Bester (2004); Routing in ISL Networks Considering Empirical IP Traffic, *IEEE Journal on Selected Areas in Communications*, 22(2): 261-272.