

CANCAR – Congestion-Avoidance Network Coding-Aware Routing for Wireless Mesh Networks

Erik Pertovt¹, Kemal Alić¹, Aleš Švigelj^{1,2}, Mihael Mohorčič^{1,2}

¹ *Jozef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia*
fige-mail: epertovt@gmail.com]

² *Jozef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[e-mail: kemalic@gmail.com, ales.svigelj@ijs.si, miha.mohorcic@ijs.si]

*Corresponding author: Aleš Švigelj

*Received August 29, 2017; revised December 4, 2017; accepted April 6, 2018;
published September 30, 2018*

Abstract

Network Coding (NC) is an approach recently investigated for increasing the network throughput and thus enhancing the performance of wireless mesh networks. The benefits of NC can further be improved when routing decisions are made with the awareness of coding capabilities and opportunities. Typically, the goal of such routing is to find and exploit routes with new coding opportunities and thus further increase the network throughput. As shown in this paper, in case of proactive routing the coding awareness along with the information of the measured traffic coding success can also be efficiently used to support the congestion avoidance and enable more encoded packets, thus indirectly further increasing the network throughput. To this end, a new proactive routing procedure called Congestion-Avoidance Network Coding-Aware Routing (CANCAR) is proposed. It detects the currently most highly-loaded node and prevents it from saturation by diverting some of the least coded traffic flows to alternative routes, thus achieving even higher coding gain by the remaining well-coded traffic flows on the node. The simulation results confirm that the proposed proactive routing procedure combined with the well-known COPE NC avoids network congestion and provides higher coding gains, thus achieving significantly higher throughput and enabling higher traffic loads both in a representative regular network topology as well as in two synthetically generated random network topologies.

Keywords: Network coding; Dynamic routing; Coding-aware routing; Wireless mesh networks

1. Introduction

In recent years, we are witnessing large scale deployments of WiFi access points and wireless sensor networks, both making use also of wireless mesh networking between the nodes. Moreover, wireless mesh networks [1] are nowadays also deployed to provide the last-mile Internet access in rural regions. Nodes in a mesh network communicate with each other through multi-hop wireless links, providing larger coverage area yet reducing the overall network throughput. In order to improve the wireless mesh network performance, several mechanisms have been proposed. Network coding (NC) [2] as one of them is experiencing an increasing attention since it can be applied on different layers of the protocol stack. Instead of using receive and forward mechanism for individual relaying of basic information blocks, e.g., packets, NC combines multiple received packets either from the same or from different traffic flows that fulfil certain criteria into a single encoded packet and forwards it, thus indirectly increasing the overall network capacity. NC among packets from the same traffic flow such as in MORE [3] is referred to as intra-session NC, whereas NC among packets from different traffic flows as in COPE [4] is called inter-session NC. In wireless mesh networks, opportunistic inter-session NC can exploit the broadcast nature of the wireless medium in order to create new coding opportunities [4]. Nodes are listening to all transmissions, therefore, they can overhear also packets, which are not destined to them, and use them to code together even more packets.

In this paper, we are concerned with the inter-session NC implemented between the medium access control (MAC) and network layers of the protocol stack, focusing on potential benefits provided by its tight collaboration with routing procedure. If the latter aims to discover coding opportunities in the network and take them into account during route selection, we refer to it as NC-aware routing [5]. By creating additional coding opportunities, NC-aware routing further increases network throughput compared to NC-unaware routing. When selecting paths with more coding opportunities, NC-aware routing tends to bundle multiple flows together. Therefore, the traffic intensity on those paths should also be considered to avoid network congestion. In wireless networks, high traffic does not only result in traffic congestion but also in the increased interference among the nodes located within each other's coverage area [5].

The existing distributed NC-aware routing schemes are mostly based on the reactive routing concept. They tend to find a path where more coding opportunities exist for a new traffic flow based on information of existing traffic and network topology. In this paper, however, we investigate the potential benefits of NC awareness in networks based on the proactive (i.e., table-driven) routing where the existing reactive NC-aware routing schemes cannot be used.

The procedure proposed in this paper is based on the observation that in a general network topology with non-homogeneous traffic flows part of the traffic is well-coded, while the other part is poorly-coded even when using NC-aware routing. With the term well-coded traffic, we refer to traffic flows in which packets are often coded together on a given node. We say that this traffic has high coding success. The poorly-coded or less-coded traffic, on the contrary, stands for traffic flows in which packets are not coded or are rarely coded on the node, thus considered as traffic flows with low coding success. The poorly-coded traffic on the node prevents the well-coded traffic to be coded even more, especially in highly-loaded (or even congested) areas of the network (i.e., bottlenecks), due to the limited node bandwidth and queue lengths. With poorly coded traffic flows kept on the node, overall less packets will be forwarded by this node (i.e. more packets will be dropped from the queue), whereas with only

well coded traffic flows left on the node, more incoming packets will be coded together and forwarded by the node at once. In this paper, we aim at exploiting this particular phenomenon by designing a proactive per-hop routing that combines NC awareness and congestion avoidance principles in order to increase the overall network throughput in wireless mesh networks. In particular, we propose a novel distributed proactive wireless link-state routing procedure referred to as Congestion-Avoidance Network Coding-Aware Routing (CANCAR). It considers the traffic intensity and the information of the measured traffic flow coding success to reroute part of the least coded traffic from the network bottlenecks in the case of high traffic load in the network.

The rest of the paper is structured as follows. In Section 2, the most representative related work on NC and NC-aware routing in wireless mesh networks is presented and compared to our work, emphasizing the novelty of our approach. In Section 3, we present the concept of the proposed proactive routing. In Section 4 we propose and describe the new routing procedure CANCAR and in Section 5 we provide simulation results for a representative regular network topology and for two synthetically generated random network topologies. Section 6 concludes the paper.

2. Related Work

In the following, we briefly present COPE network coding and conventional routing, used in this study as the baseline for the newly proposed routing procedure. This baseline has been selected as it represents a common reference for most NC-aware routing schemes, thus enabling their indirect qualitative comparison with our procedure. Furthermore, we provide an overview of some representative previous studies on NC-aware routing, which to some extent inspired the research presented in this paper, and provide the motivation for the development of a new procedure by emphasising its novelty compared to previous work.

2.1 COPE Network Coding procedure

COPE [4] is an opportunistic NC procedure purposely designed to make use of wireless communications characteristics and implemented as an independent sub-layer between the MAC and the network layers. It has shown a significant improvement in the network throughput in a real testbed environment and is commonly used as a reference procedure in the performance evaluation of newly proposed NC and NC-aware routing procedures. COPE is an inter-flow NC procedure. This means that based on opportunistic listening, packets of multiple unicast flows are opportunistically coded together along the path for one hop to the next node, where packets are decoded. This makes it well suited for the use with a proactive per-hop routing as considered in this study.

For coding, COPE uses also the information about delivery probabilities of wireless links that is calculated in all of ETX (Expected Transmission Count) based routing protocols. In the case when the exact information on the presence of a given packet at a particular neighbour's node is not available, COPE guesses on this based on packet delivery probability. In particular, a node estimates the probability that the node N has the packet P by looking at the packet delivery probability on the link between the packet's previous node and the node N. The ETX metric is also used by the shortest path routing algorithm (i.e. Dijkstra) to calculate routes in the network in COPE reference scenario.

In this paper, we consider COPE as a reference underlying NC procedure also used by CANCAR. However, the latter could also be used with other NC algorithms such as BON

(Bearing-Opportunistic Network) coding [6, 7]. Similarly as in [8], we consider the same channel capacity for all nodes, thus we use ETX metric. Similar as most previously proposed NC-aware routing procedures, we selected COPE procedure with conventional routing also as a benchmark for the performance evaluation of CANCAR, thus enabling indirect qualitative comparison of CANCAR with other NC-aware routing schemes.

2.2 Coding-Aware Routing

The prior work on coding-aware routing in wireless networks mainly consists of finding routes with more coding opportunities based on the existing traffic and network topology. Coding-aware routing tries to route the packets in such a way that more of them are being coded together, while taking into account also other parameters in the network, such as interference, the traffic intensity, the hop count of a route, etc. The existing coding-aware routing schemes are mostly based on on-demand (i.e., reactive) source routing paradigm, requiring firstly the acquisition of the routing information obtained by a route discovery protocol, calculation of paths and selection of the most suitable route, which all introduce certain delay, making these routing schemes suited for rather sparse communication patterns.

To illustrate the basic principle of coding-aware routing let us consider the example in Fig. 1(a), where A has to deliver packet P1 to B and B has to deliver packet P2 to A. With conventional (coding-unaware) routing, the disjoint routes may be selected, as shown on the right hand side. Consequently, four transmissions are required to deliver both packets. On the other hand, with coding-aware routing, coding opportunity arises on node X, which encodes both packets with, e.g., a single linear XOR (i.e., \oplus) operation into one encoded packet $P1 \oplus P2$. Subsequently, only the encoded packet is forwarded, as shown in Fig. 1(b), therefore, only three transmissions are used to deliver the packets.

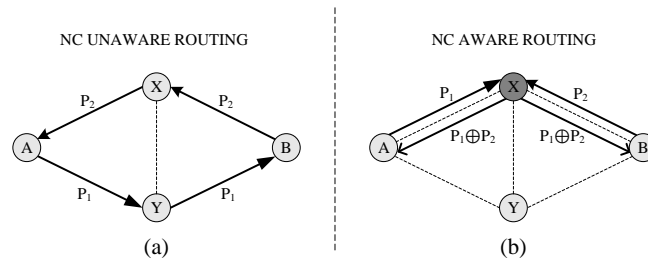


Fig. 1. Routing in NC: (a) conventional (coding-unaware), (b) coding-aware.

Several on-demand coding-aware routing procedures have been designed and presented in the literature in order to improve the NC benefits and further increase network throughput [5], considering various effects and limitations in wireless networks. For example, in [9] and [10], different metrics are proposed to balance between interference degradation and coding gain performance, i.e., to achieve a trade-off between routing multiple flows “close to each other” for more coding opportunities and routing multiple flows “away from each other” to reduce the interference. In [11], a theoretical framework for coding-aware routing is presented to reduce the number of transmissions and interference between nodes. The work provides an interesting insight into the design of protocols that integrate NC and routing selection techniques.

Cooperative network coding (CNC) proposed in [12] exploits spatial diversity for improving coding opportunities in multi-rate wireless networks. With theoretical analysis, the authors quantify the potential performance gain brought by exploiting CNC in route selection.

Routing with Opportunistically Coded Exchanges (ROCX) is proposed in [13] to reduce the

total number of coded transmissions with Exchange (ECX) metric based on a centralized coding-aware routing. The approach achieves the optimal trade-off between coding opportunities and the shortest path. ECX is an ETX-based metric and calculates the expected number of successful transmissions with coding to perform optimal routing under given topology and traffic.

In [14], an enhanced NC-aware routing algorithm is proposed named OQMCAR, which considers also the queue status in packet coding algorithm. An efficient “transmit or wait” decision for packets is made according to the queue-length based threshold policy at every transmission opportunity instead of the regular COPE opportunistic coding policy as used in existing coding-aware routing algorithms.

Centralized approaches, such as in [11] and [13], can achieve optimal solutions for a given problem but are prone to scalability problems, therefore, they are difficult to implement in real wireless networks. As an example of distributed approach, an active high-throughput optimizing routing protocol called ANCHOR is presented in [15]. It locally tries to optimize a route by creating coding opportunities with notifying related nodes to update the route.

Network Coding Aware Routing (NCAR) protocol [16] is based on a distributed Ad hoc On-demand Distance Vector (AODV)-like [17] source routing with three steps: coding condition with priority determination, route discovery process, and route selection process. It finds the best coding solutions at intermediate nodes considering link delivery ratio and coding opportunities. In the case of more than one unicast flow and more than one coding scheme it uses the Expected Transmission Count based on Network Coding (ETC-NC) metric, which is an improved ECX metric and allows differentiation between more and less reliable coding schemes.

Rate-adaptive coding-aware multiple path routing (RCR) protocol [18] is based on a distributed Dynamic Source Routing (DSR)-like [19] on-demand (i.e. reactive) routing. It uses three purposely developed metrics: Required Transmission Number (RTN), Matched Coding Rate (MCR) and Congestion Avoidance Rate (CAR). RTN represents the number of transmissions required for intermediate nodes to forward the packet belonging to a particular flow. MCR considers that different flows have different amounts of traffic, so it calculates the maximum possible rate of the sub-flow of a particular flow, which can be coded with other flows currently passing through that intermediate node. Similarly, a distributed on-demand DSR-based [19] routing protocol is proposed in [20] to use the proposed Free-ride-Oriented Routing metric (FORM). It enables a new flow to “free ride” on the existing flows, which means that it tries to code a new flow with the existing flows. DCAR [8] is also a distributed on-demand source routing protocol, which can encode packets for more than two hops. To facilitate the comparison of paths that enable coding with those that do not, it uses Coding-aware Routing Metric (CRM) metric, which is considering the expected number of transmissions for the successful transfer of existing and incoming packets for each new flow.

In [21], a distributed on-demand Coding and Capacity Aware Routing (C2AR) protocol exploits the benefits of coding-aware routing and load-balancing, making a trade-off between both. Besides exploring coding possibilities, it also takes into account the interference and capacity. An Estimated Path Throughput (EPT) metric is designed to jointly consider the effect of NC and residual capacity of the path. In order to solve the problem of CRM metric in DCAR, a distributed on-demand coding-aware routing protocol (PN-CAR) for unicast sessions is proposed in [22]. It is based on a novel routing metric, Expected Transmission Count with Coding Awareness (ETX-CA) that captures the characteristics of NC and the unicast sessions.

In [23], authors propose a principle called consistency of encoding and overhearing (CEO). It solves the problem found in most of the proposed coding-aware routing schemes [8, 20, 21, 22], which may misidentify the coding opportunities, especially when the packet is re-encoded. Avoiding coding of packets which subsequently cannot be decoded at the next hops, CEO principle ensures the successful decoding of all encoded packets. Two mechanisms are proposed in [23] and integrated in DCAR to achieve a distributed way for adherence of CEO.

There exists some recent work also on proactive routing, to some extent related to our work although focusing on different problems. In [24], a coding and interference aware proactive routing protocol is used to estimate the available bandwidth and identify high throughput paths with Coding-Aware Path Bandwidth Metric (CPBF). This is aimed at investigating QoS routing with NC as a particular subfield of research in routing with NC. The QoS guarantee is based on the differentiation of existing and new flows in the network.

In [25], an ETOX routing metric and a Hycare routing are proposed to route the packets when inter-session NC protocol is incorporated. The Hycare routing is then investigated in theoretical topologies with few flows in the network. With ETOX, the anticipated transmission time of a link is heuristically measured, considering the coding opportunities.

In [26], authors investigate multi-flow coding situation and introduce a Greedy Multi-flow-based Coding-Aware Routing (MuCAR) protocol. A decoding policy and the coding condition in the multi-flow environment are defined to achieve one of sub-optimal solutions to the presented problem.

2.3 The Motivation behind CANCAR Routing

The existing on-demand coding-aware routing procedures are typically tightly coupled with the NC (usually COPE-like) scheme and purposely defined metrics. In those procedures, coding opportunities are computed based on a given topology and traffic in the network to discover paths with more coding opportunities. These procedures are appropriate to be used as on-demand source routing, as they are designed to search for coding opportunities separately for each new traffic flow in the network. However, they cannot be used as a proactive per-hop routing, where routes are periodically and simultaneously calculated for all traffic flows in the network. In this case, coding opportunities should be simultaneously searched among all the traffic flows in the network.

The novelty of the proposed CANCAR routing is in using proactive routing, which explores the information of the actually measured coding success in various traffic flows. By discriminating between well-coded and poorly-coded traffic flows in the congested area of the network, it primarily facilitates congestion avoidance routing and indirectly affects coding opportunities in the network by releasing the most loaded parts of the network from poorly-coded traffic. Although these least coded or/and non-coded flows may not be better coded on their new paths, their rerouting to less-loaded parts of the network relieves the most loaded node from entering the congestion state. Thus, CANCAR increases the number of packets that get coded in remaining flows and, in some cases, it creates coding opportunities for previously least coded or/and non-coded flows along the new alternative routes, although the latter not being its primary goal. As it does not search for coding opportunities based on the topology and traffic in a given moment in time, it is appropriate to be used as a proactive per-hop routing, which enables immediate sending of packets of a new flow upon request, without any route determination delay. It is thus better suited for dense communication patterns, which also benefit more from NC. With this, CANCAR presents a completely new approach on how NC can benefit from coding based information when using proactive routing

in the network.

The contributions of this paper can be summarised as follows: (i) we introduce a new concept of a proactive routing that combines NC awareness and congestion avoidance based on the coding success information and we demonstrate its operation on a representative scenario; (ii) we propose a new robust distributed proactive link-state congestion avoidance routing CANCAR with low additional overhead that considers the traffic intensity and the measured information about traffic flow coding success, and demonstrate it on a representative scenario; (iii) we demonstrate that rerouting of poorly coded traffic flows away from the network bottlenecks may increase coding success on released nodes as well as it may create coding opportunities along alternative paths of rerouted flows; and (iv) we implement the proposed routing in a simulation model and evaluate its performance in a representative regular network topology as well as in two synthetically generated random network topologies.

3. The Proposed Concept

The proposed routing in this paper is concerned with traffic rerouting based on coding success differentiation. Some parts of the network can represent bottlenecks and the traffic in those parts can cause network congestion or even saturation. Thus, to avoid congestions, we propose that heavily loaded parts of the network are identified and traffic is sorted based on the coding success of individual flows. The less-coded traffic, which prevents more packets being coded due to the limited node bandwidth and queue length, is then rerouted away from the most loaded parts of the network. This allows that the remaining more-coded traffic on the node is better coded due to the duty cycle increase for this traffic, enabling higher number of encoded packets at the bottleneck. Moreover, due to the traffic rerouting, not only the traffic intensity but also the interference from the less-coded traffic in the heavily loaded parts of the network is reduced. On the other hand, the rerouted less-coded traffic can be potentially better coded on the diverted routes in other parts of the network.

The described concept is demonstrated on the representative section of a network depicted in [Fig. 2](#), where all the properties of the proposed NC-aware routing are presented. The depicted topology has been carefully selected to facilitate the explanation of the proposed concept. For easier explanation, we assume that four traffic flows between the source-destination node pairs 13-17, 17-13, 6-8 and 9-7 are generated, and each of the packets P_1 , P_2 , P_3 and P_4 corresponds to one of the flows, respectively ([Fig. 2\(a\)](#)). These flows intersect at node 1, assuming that the shortest path routing based on hop count is performed, thus paths $13 \rightarrow 12 \rightarrow 1 \rightarrow 15 \rightarrow 17$, $17 \rightarrow 15 \rightarrow 1 \rightarrow 12 \rightarrow 13$, $6 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 8$ and $9 \rightarrow 5 \rightarrow 1 \rightarrow 3 \rightarrow 7$ are being selected. In the given traffic situation, node 1 represents a bottleneck. Without NC, node 1 requires four transmissions to forward packets P_1 , P_2 , P_3 , and P_4 , while with NC, only three transmissions are required as node 1 can encode packets P_1 and P_2 into one encoded packet $P_1 \oplus P_2$ and broadcast it to nodes 12 and 15. In [Fig. 2\(b\)](#), packets are rerouted to the alternative paths around node 1 to prevent the congestion. In our approach, we propose that packets P_3 and P_4 , which cannot be coded on the bottleneck node 1, are rerouted to the alternative less loaded and only one hop longer paths $6 \rightarrow 18 \rightarrow 22 \rightarrow 20 \rightarrow 10 \rightarrow 8$ and $9 \rightarrow 11 \rightarrow 21 \rightarrow 23 \rightarrow 19 \rightarrow 7$. Packets P_1 and P_2 are still routed on the shortest paths through node 1, on which coding is still possible. Thus, node 1 forwards all the packets in a single transmission.

We now consider that more packets are in each flow. Theoretically, only half of all packets travelling through the bottleneck node 1 would be coded together in [Fig. 2\(a\)](#), i.e., packets of

flows between the source-destination node pairs 13-17 and 17-13, whereas in **Fig. 2(b)** all the packets travelling through node 1 would be coded. Considering that nodes have limited bandwidth and queue length, some of the packets travelling through the bottleneck node 1 would be dropped once the queue is full. By rerouting the packets of flows, which do not get coded, more packets of the encoded flows would be coded in the bottleneck in **Fig. 2(b)**, enabling less packets being dropped from the queue and thus increasing throughput of the network.

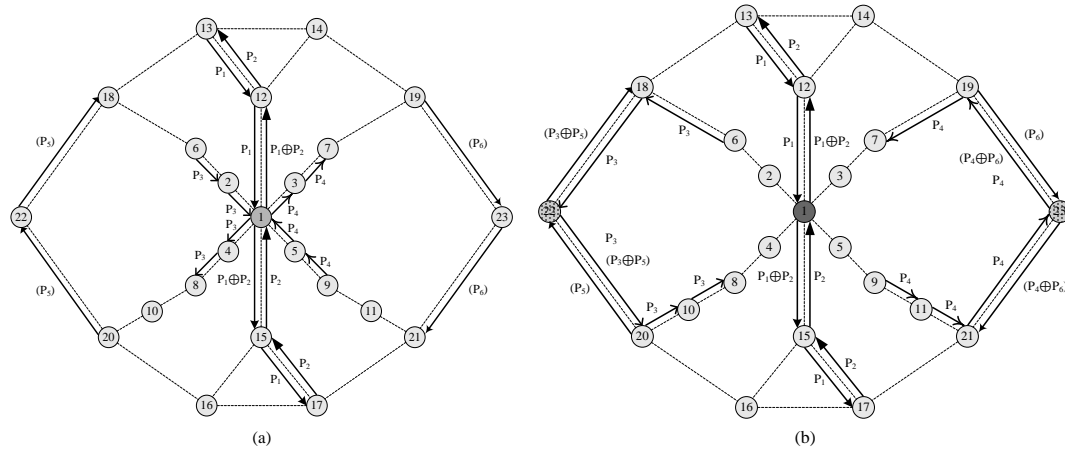


Fig. 2. Reference scenarios

Furthermore, let us assume two additional traffic flows between the source-destination node pairs 20-18 and 19-21, traversing paths 20→22→18 and 19→23→21, thus intersecting the alternative paths of packets P_3 and P_4 . Packets P_5 and P_6 , depicted in brackets in **Fig. 2**, correspond to these two flows. In such case, packets P_3 and P_4 routed through alternative paths also get coding opportunities. Packet P_3 can be potentially coded with packet P_5 ($P_3 \oplus P_5$) on node 22 and packet P_4 can be potentially coded with packet P_6 ($P_4 \oplus P_6$) on node 23, reducing the number of packets dropped from the queues of nodes 22 and 23 in the case of high traffic load.

4. CANCAR: Congestion-Avoidance Network Coding-Aware Routing

We implemented the proposed concept in the form of a link state routing procedure which performs a distributed dynamic proactive routing based on the existing conventional routing algorithms and in such a way that it can be used with different opportunistic NC algorithms. In this study, we describe its implementation on top of COPE [4], thus facilitating indirect qualitative comparison of CANCAR with other existing COPE-based procedures.

The overall CANCAR procedure consists of several steps that we grouped into three main phases, described in the following subsections and depicted in **Fig. 3**, the first and the second representing the conventional approach. In the first phase, the information required to perform routing is collected and distributed across the network. In the second phase, routes are calculated based on conventional routing. In the third phase, which is the core of CANCAR procedure and represents the main novelty, the alternative paths are discovered and the flows, which will be rerouted, are selected based on the algorithm for alternative paths selection. With CANCAR, each node has two routing tables; one for the shortest paths and one for the

alternative paths. Both are used with CANCAR. Which routing table is to be used is determined for each traffic flow on each node. By default, flows are routed on the shortest paths. Only flows that are identified by the currently most loaded node as poorly coded are routed along alternative paths. The source nodes of these flows have to know that these flows have been selected by the most loaded node to be routed on alternative paths. Therefore, the most loaded node sends a short report to the corresponding source nodes to set the flag in every packet of the selected flows. If the flag in an individual packet is set, nodes forward the packet based on the routing table for the alternative paths.

4.1 Collection and Distribution of the Information for Routing

First, the routing information needed to calculate routes has to be acquired. For proactive routing procedures this is done periodically before each update of a routing table. For CANCAR, on each wireless node in the network the information is collected on:

- the delivery probabilities of wireless links,
- the average number of packets in queues,
- the number of packets dropped from queues,
- the number of packets in each existing flow passing through a node,
- the percentage of encoded packets in each existing flow on a node.

The ETX value for all the links at the node and the average number of packets in the node queue are placed into the routing information messages, which are dispatched to all the nodes in the network.

4.2 Calculation of Shortest Paths

In this phase, the shortest paths are found with Dijkstra's shortest path routing algorithm using the ETX metric, which is calculated based on the link delivery probabilities. The calculation is performed at every routing table update interval. It is done in a distributed manner on every node in the network based on the information obtained from the routing information messages and the available local information on a particular node. Following this phase, every node has a routing table with the shortest paths, enabling conventional operation of the entire network without NC.

4.3 Calculation and Selection of Alternative Paths

Immediately after the calculation of shortest paths routing table CANCAR calculates also alternative paths and selects, through five sequential steps, a subset of alternative paths for insertion in the alternative routing table.

Step 1: Determining the Most Loaded Node. First, the currently most loaded node in the network is determined based on the average number of packets in the node queue. This information is distributed along with the ETX values of node links in the routing information messages, so it is readily available at the routing table update interval. A node is assumed to be highly-loaded (or congested) if the average number of packets in the queue is higher or equal to the value of the threshold parameter, P_Q . The most loaded node in the network is determined as the node with the highest average number of packets in the queue. If no highly-loaded nodes are detected, the remaining steps in a given routing table update interval are skipped and all traffic is routed according to existing routing tables.

Step 2: Calculation of Alternative Paths. In this step, every node in the network calculates alternative paths around the most loaded node and its neighbourhood and builds a routing table for alternative paths according to the following definition.

Definition 1. Let the node V_C be the currently most loaded node among all the nodes V in the network, where V_x is a node x ($x = 1, 2, \dots, i, j, \dots, N_V$). Let e_{ij} be the link between nodes V_i and V_j . Also, let V_N denote the set of neighbour nodes of the V_C and similarly V_I the set of neighbours of the V_N . Then, the alternative paths are calculated with Dijkstra's routing algorithm on the sub-network not taking into account the links e_{CN} , e_{NC} , e_{NI} and e_{IN} .

This means that the wireless links of the most loaded node and those of its neighbours (in both directions) will be avoided in calculation of the alternative paths. For example, in the representative scenarios in Fig. 2, the node V_C is node 1, nodes V_N are nodes 2, 3, 4 and 5, while nodes V_I are nodes 6, 7, 8 and 9. Thus, the avoided links at alternative paths calculation are e_{12} , e_{21} , e_{13} , e_{31} , e_{14} , e_{41} , e_{15} , e_{51} , e_{26} , e_{62} , e_{37} , e_{73} , e_{48} , e_{84} , e_{59} and e_{95} .

By disregarding all the wireless links of the neighbours of the most loaded node we also reduce the interference produced by the neighbours to the most loaded node, as they are all transmitting at the same frequency and thus interfering with each other. With this, link delivery probabilities are increased and the nodes in this region successfully transmit more coded and non-coded packets of the remaining flows. They also overhear more packets which are used to decode encoded packets by COPE algorithm, resulting in higher number of decoded packets and better performance of NC at the network bottleneck.

Step 3: Rerouting along Alternative Paths. In this step the most loaded node selects flows that will be rerouted along the alternative paths, which were calculated in the previous step. This step is performed locally only on the most loaded node, thus introducing no overhead to the network.

This step is the main part of CANCAR and is due to its importance separately presented in more details in Section 4.4.

Step 4: Informing Nodes about Alternative Paths Selection. In this step, the most loaded node informs the source nodes of the flows selected in Step 3 to reroute flows along the alternative paths. It simply sends a short report to the corresponding nodes. Until the information about the selected flows is spread across the network, the flows are routed on the shortest paths calculated in Section 4.2.

Step 5: Packets Routing. Packets are routed through the network based on the status of an additional flag in the packet header. The source nodes set the flag for each packet to denote whether the packet should be routed using alternative path (the flag is set) or if the shortest path is to be used (the flag is not set). Every intermediate node on the route checks the status of this flag and accordingly selects the next hop link for forwarding an individual packet from the appropriate routing table.

4.4 Algorithm for Alternative Paths Selection

After determining the currently most loaded node V_C among all the nodes V in the network and calculating the alternative paths, the algorithm selects which flows will be rerouted from the most loaded node. The flow chart of the algorithm is depicted in Fig. 3(a), while the annotations used in the flow chart are explained in Table 1.

Sorting Flows: First, all the existing flows F_{Ck} passing through the most loaded node V_C are sorted according to the increasing coding success starting from the flow with the least encoded packets on the V_C ($k = 1, 2, \dots, i, j, \dots, N_{F_C}$) using the information of the percentage of encoded

packets in the individual existing flow on the node V_C . This information is used as the information of the measured traffic coding success. The traffic flows with many encoded packets on the V_C represent the well-coded traffic while the traffic flows with few encoded packets on the V_C represent the poorly-coded traffic. It is assumed that a particular flow currently exists and traverses the most loaded node if the number of packets of the flow going through the most loaded node is larger than zero. Then, the first flow ($k = 1$) with the least encoded packets is selected by the algorithm and analysed.

Table 1. Annotations for the algorithm flow chart.

Annot.	Description
V_C	the most loaded node in the network
F_{Ck}	a flow k going through the V_C
k	sequential number of F_{Ck} according to coding success
$HC_{F_{Ck},Alt}$	hop count of the alternative path for the F_{Ck}
$HC_{F_{Ck},EtX}$	hop count of the shortest path calculated with ETX
H	the parameter for determining the maximum allowed increase of the $HC_{F_{Ck},Alt}$ compared to the $HC_{F_{Ck},EtX}$
$V(F_{Ck},Alt)$	the set of nodes on the alternative path of the F_{Ck}
$n_{C,Alt}$	the number of packets in all the already rerouted F_{Ck}
$n_{F_{Ck}}$	the number of packets in the rerouted F_{Ck} on the V_C
$n_{C,D}$	the number of packets dropped from the queue of V_C
N_{F_C}	the number of all F_{Ck}
U	the minimum number of flows always left on the V_C
$F_{Ck,Alt}$	the rerouted F_{Ck}
$V_{S,F_{Ck,Alt}}$	the source node of the $F_{Ck,Alt}$

Checking for an Alternative Path: For the analysed flow, the algorithm checks if the alternative path Alt around the currently most loaded node exists according to the Definition 1. The path exists when there exist alternative links in the network, which do not belong to any of the e_{CN} , e_{NC} , e_{NI} and e_{IN} links, to route the analysed flow. If the alternative path by the Definition 1 does not exist for this flow, the flow will be routed through the shortest path and the next flow is picked by the algorithm.

Checking the Hop Count of an Alternative Path: If the alternative path for the analysed flow exists, the algorithm checks if the path around the most loaded node is short enough in terms of ‘hop count’ according to Equation 1:

$$HC_{F_{Ck},Alt} \leq HC_{F_{Ck},EtX} + H \quad (1)$$

$HC_{F_{Ck},EtX}$ is the hop count of the shortest path, $HC_{F_{Ck},Alt}$ is the hop count of the alternative path and $H = 0,1,2, \dots$ is the maximum allowed increase of the number of hops on the alternative path as compared to the shortest path. According to Equation 1, if the increase of hop count on the alternative path is higher than H , the next flow is picked by the algorithm. With this, we prevent selecting too long paths, which would notably increase the network delay and cause higher overall traffic load and interference level in the network due to the same network traffic occupying more links along the route.

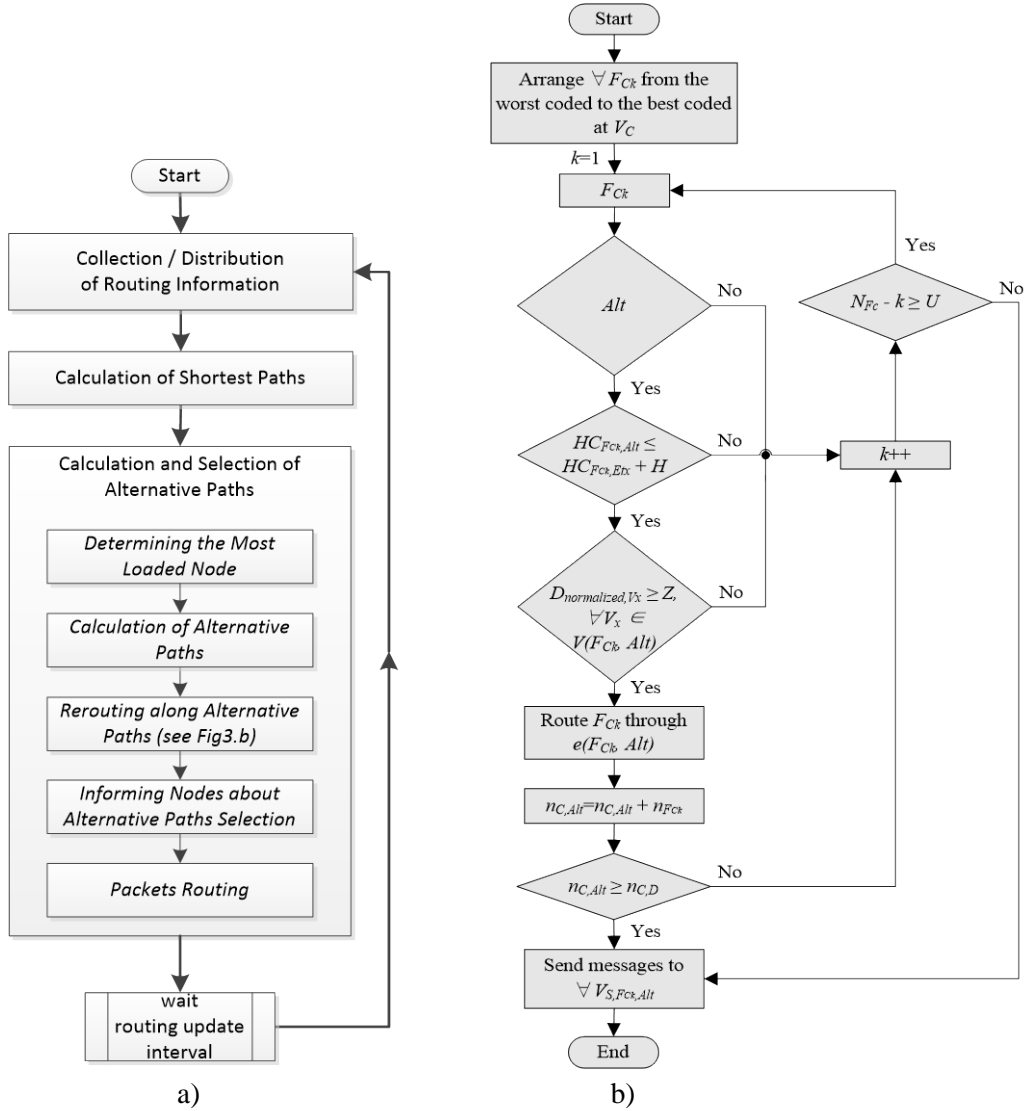


Fig. 3. Flow chart of (a) CANCAR and (b) the alternative path rerouting algorithm.

Checking the Traffic Level on an Alternative Path: Next, the algorithm checks if the alternative path is not similarly loaded as the shortest path. Let $V(F_{Ck}, Alt)$ be the set of nodes a flow F_{Ck} traverses on the alternative path. Then, if one of the nodes V_x from the set $V(F_{Ck}, Alt)$ is similarly loaded as the most loaded node V_C , the alternative path is assumed to be similarly loaded as the shortest path and the analysed flow cannot be rerouted. We determine if a node is similarly loaded by normalizing the difference between the average number of packets in the queues of the nodes V_C and V_x , i.e., \bar{q}_C and \bar{q}_x , to the length of the node's queue Q_V as given by Equation 2:

$$D_{normalized,Vx} = \frac{\bar{q}_C - \bar{q}_x}{Q_{V_x}} \quad (2)$$

V_x is determined to be similarly loaded as V_C if $D_{normalized,V_x} < Z$ is true, where Z is a load similarity threshold defined as $0 \leq Z \leq 1$. The lower the threshold Z , the more nodes remain as candidates for alternative paths and vice versa.

Selecting an Alternative Path as a Route: If the analysed flow meets all the discussed conditions, the alternative path around the most loaded node is selected as a route for the flow.

Checking the Amount of Traffic Selected to be Rerouted from the Most Loaded Node: Algorithm calculates the number of received packets in all the flows already selected to be rerouted from the node V_C on the alternative paths during the past observed time interval using the counter $n_{C,Alt}$. Until the condition $n_{C,Alt} \geq n_{C,D}$ is reached, where $n_{C,D}$ is the number of packets dropped from the queue of the node V_C , the entire process of determining the flows to be rerouted is repeated taking the next flow as per number of encoded packets. When true, we assume that sufficient amount of poorly-coded traffic has been rerouted from V_C . This condition determines that dropped packets during the past observed time interval represent the node's traffic surplus and the same amount of packets should be diverted from the node to prevent the node's congestion.

Selecting the Next Flow: In case that more traffic needs to be rerouted from the most loaded node, the next flow is picked by the algorithm provided there are still more than U flows available. The idea here is that U of the best coded flows going through the most loaded node V_C should remain on the node, to prevent leaving the node completely vacant and thus also avoid large traffic load oscillations. If any packets are coded on the node, the minimum reasonable value for U is 2 to maintain the already achieved inter-session coding opportunities, whereas if no packets are coded on the node, the algorithm tries to leave only one traffic flow on the most loaded node.

4.5 CANCAR Overhead

In this section, we evaluate the required routing overhead for the implementation of CANCAR in a real environment. As CANCAR uses the conventional routing as the supporting routing mechanism, it also adopts the same basic overhead. However, CANCAR generates additional overhead for congestion-avoidance coding-aware routing features. We first estimate the overhead of the conventional routing and then the additional overhead required by CANCAR.

For a distributed and dynamic ETX-based routing scheme, each node needs to send the information on the ETX for each link of a node to other nodes. If the average number of links per node is N_x and the ETX link information is coded with 8 bits, the information in a routing information message for ETX is of a size

$$I_{ETX} = N_x \cdot 8 \text{ bits} \quad (3)$$

For CANCAR-based routing, each node needs to send the information of ETX and the average number of packets in the node queue. Assuming that the average number of packets in the node queue is also coded with 8 bits, the size of the information required by routing results in:

$$I_{CANCAR} = I_{ETX} + 8 \text{ bits} = (N_x + 1) \cdot 8 \text{ bits} \quad (4)$$

In CANCAR, the most loaded node V_C is required to send a short report for every rerouted flow to the corresponding source of that flow, to notify it about the required rerouting on the

alternative path. If the number of rerouted flows is n_{FR} and the number of the wireless nodes in the network is N_V then the number of messages the most loaded node V_C is required to send is:

$$N_{Messages, V_C} = n_{FR} + N_V - 1 \quad (5)$$

The first term n_{FR} is for sending the short reports to source nodes of the rerouted flows and is required only for CANCAR, while the second term $N_V - 1$ is for sending the routing information messages to every node in the network but itself and is required for both CANCAR and ETX-based routing scheme. For ETX-based routing, every node has to send a routing information message to every node in the network, resulting in $N_{ETX} = N_V \cdot (N_V - 1)$ routing information messages. The same is done for CANCAR routing except for the most loaded node V_C that has to send also the n_{FR} short reports. The number of routing information messages that nodes have to send for CANCAR is thus:

$$N_{CANCAR} = N_V \cdot (N_V - 1) + n_{FR} = \left(1 + \frac{n_{FR}}{N_V \cdot (N_V - 1)}\right) \cdot N_{ETX} \quad (6)$$

where N_{ETX} is the number of messages that nodes have to send for ETX-based routing schemes. For CANCAR the percentage increase in terms of the origin packet count for routing information compared to ETX-based routing is thus $n_{FR} / (N_V \cdot (N_V - 1))$. We should consider also that the routing information messages have to be flooded throughout the network from every node to every node, while the short reports only need to be sent from the most loaded node V_C to a specific number (i.e., n_{FR}) of source nodes. However, if the $n_{FR} \approx N_V$ is in approximation true, we can also consider to slightly change the routing scheme and broadcast also the short reports. In such case, the scheme would broadcast only one origin packet from the most loaded node, in which the information would specify the rerouted flows resulting in only $1 / (N_V \cdot (N_V - 1))$ percentage increase of the number of origin packets for routing. If N_V is high, the CANCAR routing overhead increase can be neglected.

5. Performance Evaluation

We performed a comprehensive evaluation of the proposed CANCAR routing using NC simulation model for wireless mesh networks developed in the OPNET/Riverbed Modeler [27], extensively presented in [28, 29]. The overall goal of simulations was to quantify the benefits of CANCAR over the NC with the conventional coding-unaware routing as used in [4], following the approach taken by most of the related studies presented in Section 2.2.

For the performance evaluation of CANCAR, three types of procedures were used in simulation, for which we use the following notations:

- CANCAR: CANCAR was used on top of COPE and conventional ETX-based routing,
- COPE: only COPE was used with conventional ETX-based routing,
- ETX: conventional ETX-based routing without any NC was used in the network.

In all the three cases, Dijkstra's algorithm with ETX metric is used for conventional routing.

5.1 Simulation Parameters

In our analysis, we assumed homogeneous wireless networks, where all nodes had identical configuration and were static. All connected nodes had the channel capacity of 2 Mbit/s. The packet delivery probability of wireless links was changing dynamically. It depended on the distance between the nodes, considering the impact of path loss in a wireless environment, and the amount of traffic between the neighbouring nodes, modelling the link loss due to the interference in the wireless mesh network. During the simulation, the packet delivery probability was set between 100% and 90%, since COPE requires high delivery probabilities to perform well [30]. With this, we assumed that good quality links existed in the network and were thus used for routing.

In the simulation, UDP-like traffic was generated, in order to eliminate the impact of other mechanisms for congestion avoidance used in TCP. Packets were generated at the selected source nodes with the same intensity using exponential distribution of inter-arrival times and were sent to the selected destination nodes. The amount of generated packets within each flow was increased through the simulation runs representing different overall network traffic loads. We considered all packets having the same size of 10 kbits.

Every simulation run took 960 seconds. For the first 60 seconds, the traffic load was set at 80% of the final load for the system to become stable. Then, the network was loaded with the full load and the results were collected for 900 seconds. For each simulation case, we repeated the simulation eight times using different simulation seeds. The presented results are thus the average of results obtained with eight different seeds.

Routing tables were calculated at the beginning of each simulation run and were subsequently updated every 30 seconds. The observed time interval, in which the information required by the routing was taken, was the same for all the required information and equal to the routing tables update interval in all three considered procedures, i.e., CANCAR, COPE and ETX. Parameters used for CANCAR in subsequent performance analysis are: $Q_V=100$; $P_Q=1$; $Z=0.1$; $H=2$ and $U=2$. We defined the parameters' values heuristically, by extensive simulations, using different also more realistic network topologies.

For the evaluation of CANCAR, we also defined the measured coding gain G_x of a node V_x . It is given as the ratio between the number of outgoing source (i.e., before coding) packets S_x handled by the node and the number of packets R_x required to send those source packets after applying NC:

$$G_x = \frac{S_x}{R_x} \quad (7)$$

where $G_x \geq 1$. If no source packets are coded on the node V_x the measured coding gain G_x is equal to 1, as $S_x = R_x$. The parameter G_x indicates how good a node is in performing network coding. The higher the G_x , the more packets are coded on the node.

5.2 Results for reference scenarios

The performance of CANCAR was studied in the two reference scenarios explained in Section 3 with the network topology depicted in Fig. 2. The network topology consists of 23 wireless nodes placed in the area of 1.2 km square. The network diameter is 6, the network average hop count is 3.05 and the average number of wireless links per node is 2.61. The traffic flows in the network were generated as presented in Section 3.

5.2.1 Network Goodput

In Fig. 4, the network goodput in dependency of network load is shown for CANCAR, COPE, and ETX procedures for both theoretical scenarios analysed in Section 3. Four traffic flows between the source-destination node pairs 13-17, 17-13, 6-8 and 9-7 are used in Scenario I. In Scenario II, two additional traffic flows between the source-destination node pairs 20-18 and 19-21 were added to four traffic flows from Scenario I. The network goodput is the application level throughput and represents the amount of useful data delivered by the network per unit of time. The results are presented for the loads until the goodput increases in the case of using CANCAR in Scenario II.

When the network load is low, CANCAR, COPE and ETX perform similarly as there are only few coding opportunities among flows. When the network load increases, COPE that exploits NC delivers higher goodput than the ETX. This is mainly because packets are coded at the network bottleneck (i.e., on node 1) with COPE, which enables more packets to be forwarded by node 1 at high traffic load. The highest goodput, however, is achieved with CANCAR that can handle also higher traffic loads at which the network with COPE gets congested. CANCAR detects that at some point node 1 can no longer handle the increased traffic. This causes packets dropping from the queue of node 1. Therefore, CANCAR reroutes the non-coded traffic flows between node-pairs 6-8 and 9-7 passing node 1 to the alternative paths $6 \rightarrow 18 \rightarrow 22 \rightarrow 20 \rightarrow 10 \rightarrow 8$ and $9 \rightarrow 11 \rightarrow 21 \rightarrow 23 \rightarrow 19 \rightarrow 7$ away from node 1. With this, some resources on node 1 are released and can now be used by coded traffic flows between node-pairs 13-17 and 17-13. Packets of these two flows that would be dropped due to overflowing queues, if non-coded traffic remained on node 1, can now get more frequently coded together on the node. In the case of Scenario II, packets from rerouted flows can be potentially further coded on the alternative paths on nodes 22 and 23 saving bandwidth resources in those parts of the network, as presented in Section 3.

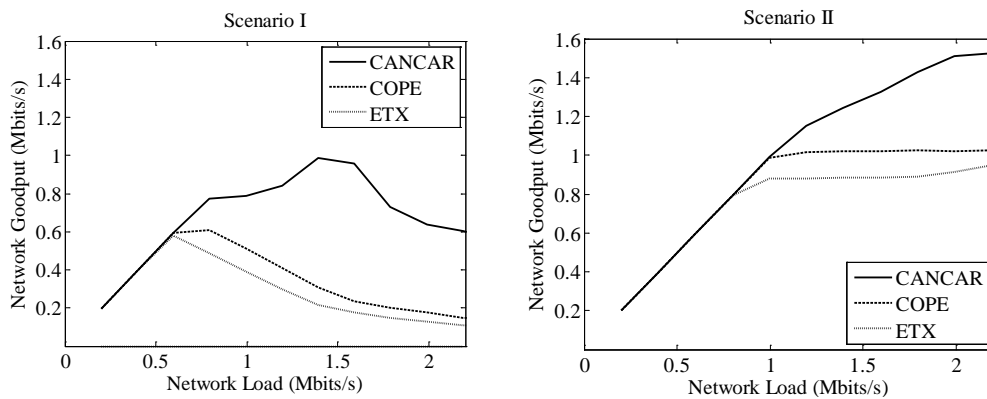


Fig. 4. Network goodput of reference scenarios using different routing procedures.

In Scenario I, the network goodput for ETX and COPE only increases with the network load up to approximately 0.6 Mbits/s and 0.8 Mbits/s, respectively, where the network gets congested and any further increase in network load only decreases the goodput. In this region, CANCAR provides the main performance improvement in terms of network goodput and handles increased network load until approximately 1.6 Mbit/s. This is because CANCAR detects the part of the network approaching the state of congestion (i.e., node 1) and reroutes the non-coded traffic flows, alleviating the traffic load around node 1 and enabling more coding of packets of the coded traffic flows.

5.2.2 Node Coding Gains

In Fig. 5, the node coding gains are depicted as obtained by COPE (a) and CANSAR (b) in Scenario I (left) and Scenario II (right), respectively. The presented node coding gains were obtained for the network traffic load of 1.4 Mbit/s in Scenario I and 1.8 Mbit/s in Scenario II; however, the same conclusions can be made for other similar loads, where CANSAR performs better than COPE in Fig. 4.

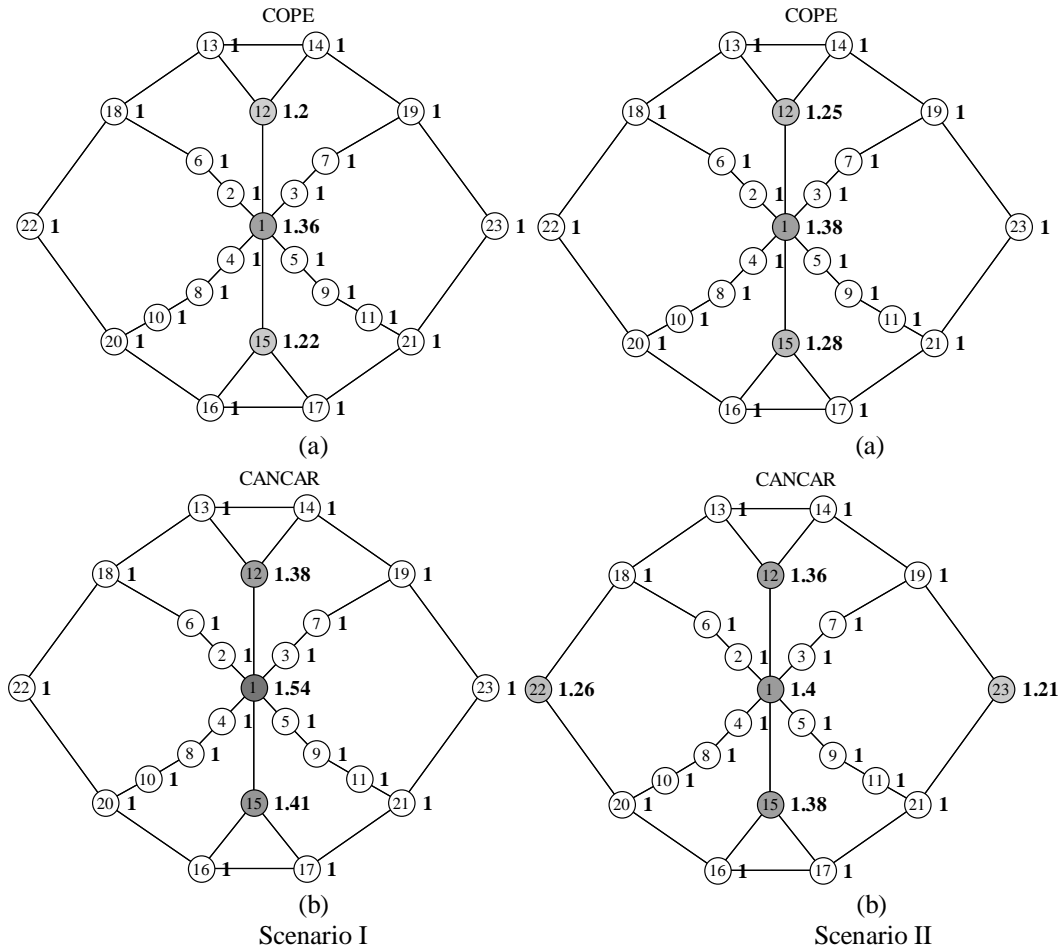


Fig. 5. The measured coding gain of nodes in Scenario I (left) and Scenario II (right) using different routing procedures: (a) COPE procedure, (b) CANSAR.

With CANSAR, coding takes place on nodes 1, 12, and 15 in Scenario I, while in Scenario II nodes 22 and 23 also contribute to coding. Notice that we have now flows of packets, while in Section 3 we analysed theoretically a few separate packets; one packet per flow. Therefore, in Section 3, coding appears only on node 1 in Scenario I and nodes 1, 22 and 23 in Scenario II, whereas, in practice, coding appears also on nodes 12 and 15. However, we can see that for these nodes coding gains are higher when using CANSAR. This is due to the fact that CANSAR reroutes the traffic flows between node-pairs 6-8 and 9-7, which are not coded on the node 1, thus allowing traffic flows 13-17 and 17-13 being coded even more on nodes 1, 12 and 15. This enables more packets being coded together (to save bandwidth resource) and less packets being dropped on these nodes, yielding higher goodput in the network.

Furthermore, nodes 22 and 23 do not encode any packets with COPE in Scenario II, while CANSAR enables that both nodes encode packets, which further saves some network resources. After being rerouted, flows between node-pairs 6-8 and 9-7 can be coded with flows 20-18 and 19-21 on nodes 22 and 23, respectively, where new coding opportunities are found.

5.2.3 The Number of Encoded Packets

We also investigated the number of encoded packets in dependency of network load for both CANSAR and COPE. Results are depicted in Fig. 6. We can see that more packets are coded with CANSAR in cases, when the benefits of CANSAR over COPE occur in Fig. 4. This is due to the higher node coding gain at the network bottleneck after rerouting the non-coded flows and the new coding opportunities experienced for the non-coded flows on new routes (in Scenario II). This results in better network performance in the sense of the network goodput, as presented in Fig. 4. We can also see that in a certain traffic load interval below 1 Mbit/s COPE codes more packets than CANSAR, which is more noticeable for Scenario I. This is a consequence of rerouting non-coded flows by CANSAR to prevent the congestion on node 1, which reduced traffic load on nodes where coding occurred below the level to create coding opportunities on nodes. When network load further increases, the traffic on nodes is again high and CANSAR encodes more packets than COPE.

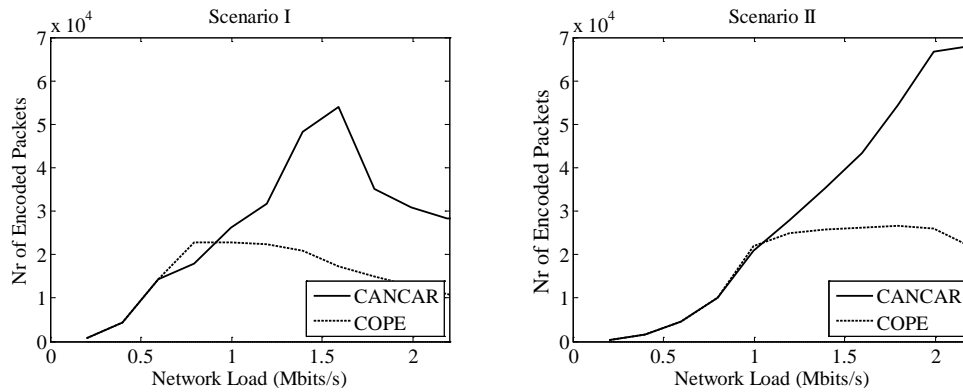


Fig. 6. Number of encoded packets in dependency of network load using different routing procedures.

5.2.4 Overhead Estimation

The routing overhead for CANSAR and the routing overhead for conventional routing with ETX metric are comparable. The increase of the number of source packets for routing with CANSAR can be neglected. In the reference Scenarios I and II, the currently most loaded node 1 has to reroute only 2 traffic flows. For rerouting these two flows, CANSAR only needs to send 2 additional unicast short report packets to the corresponding destination nodes (node 13 and 17). This results in 0.4 percentage increase in terms of the source packets count for routing information for CANSAR compared to ETX-based routing; the increase in terms of routing overhead for CANSAR is even lower as shown in Section 4.5. Similar observation applies to the increase of the number of bits in the routing information message, where only 8 bits are added to every routing information message.

5.3 Results for Realistic Wireless Network

In this section, we evaluate the performance of the proposed routing procedure CANSAR also in more realistic general topology wireless mesh networks. To this end, we generated two network topologies consisting of 40 wireless randomly distributed nodes over the area of 1 km

by 1 km. The two synthetic network topologies are depicted in **Fig. 8** and denoted by T1 (left) and T2 (right). The transmission range of a wireless node is assumed to be 350 m for T1, and 500 m for T2. The allowed number of wireless links per node is 8. We furthermore assumed that the wireless mesh networks represented WiFi networks which worked also as transport networks to transmit the traffic from one part to the other part of the network. Thus, we selected two groups of 10 nodes, one group representing network gateways, providing connectivity to external networks, and the other group representing access points. We generated 100 traffic flows between the two groups of nodes, where each node was selected exactly three times as a source and a destination of a flow, resulting in 50 combinations of bidirectional (UL/DL) flows, i.e., pairs of flows between selected nodes.

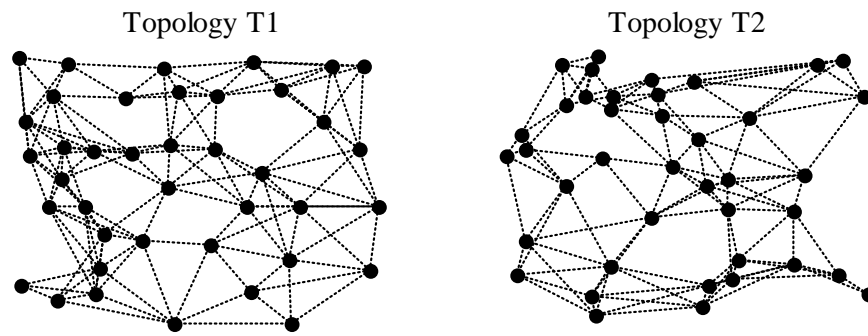


Fig. 7. Synthetically generated random topologies T1 and T2 of wireless mesh networks.

The network goodput in dependency of network load for CANSAR, COPE and ETX are depicted in **Fig. 8** for both topologies. The results show that COPE significantly increases the network goodput compared to ETX-based conventional routing without NC, while CANSAR provides further improvement of the network goodput also over COPE. Thus, CANSAR can handle the highest given traffic load, while the ETX routing without network coding can handle the lowest load in all network scenarios.

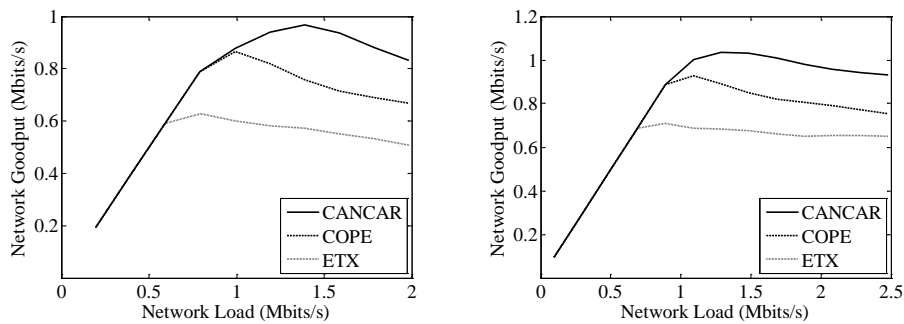


Fig. 8. Network goodput for topologies T1 (left), T2 (right).

For the same three procedures we also investigated changing of the most loaded node in the network at individual routing table update intervals throughout the simulation in order to estimate the occurrence of oscillations. The latter is characteristic for table-driven routing. In most cases, the most loaded node remains the same, which indicates that oscillations are similar for all the three cases and are not significant. In addition, not all traffic flows are diverted from the most loaded node and the second path is not longer than 2 nodes from the

shortest path in order to reduce the oscillation/stability problems.

In both topologies, the goodput gain of CANCAR becomes significant at high offered load. While COPE achieves significant goodput gain over ETX-based conventional routing without NC, CANCAR achieves up to 30% of additional gain over COPE.

The routing overhead for CANCAR and for conventional routing with ETX-based metric are also comparable. In fact, the difference is even smaller than in the reference scenarios because of the higher number of the nodes in the network (40 nodes in comparison to 23 nodes in the reference scenarios). The currently most loaded node has to reroute up to 10 traffic flows. For rerouting these flows, CANCAR has to send only 10 additional unicast short report packets to the corresponding destination nodes. This represents only 0.6 percentage increase in terms of the source packet count for routing information for CANCAR compared to ETX-based routing, which is negligible compared to the number of packets with the useful information. Similar as in reference scenarios, the percentage increase in terms of routing overhead for CANCAR is even lower as discussed in Section 4.5.

6. Conclusion

In this paper, we proposed CANCAR, a distributed dynamic congestion-avoidance routing procedure for wireless mesh networks that makes use of network coding awareness. We implemented CANCAR as a proactive link-state per-hop routing procedure that can be based on any existing conventional routing and NC procedures. It takes into account traffic intensity at the network bottlenecks and flow coding success with the algorithm for the selection of alternative paths. The algorithm is then used with the conventional routing mechanisms to effectively adapt the proposed routing to the existing routing paradigms. We implemented CANCAR in OPNET/Riverbed Modeler simulation tool to evaluate its performance compared to the reference COPE procedure and also to the ETX-based routing without NC. The benefits of using CANCAR routing are evident both in a reference topology with two traffic flow scenarios as well as in two synthetically generated random network topologies. In all cases it effectively relieves the most highly loaded part of the network from traffic flows which are least coded. With the rerouting of the non-coded flows to the alternative paths, the intensity of the traffic and the interference are reduced. This enables higher coding gains for the coded flows at the most loaded node and consequently higher goodput at and around the network bottleneck. Furthermore, if there are coding opportunities on the alternative paths, the previously non-coded flows can be also coded with the existing flows on these paths. With this, CANCAR with COPE encodes more packets than COPE alone and further increases the network performance in terms of the network goodput. This confirms the suitability of using the measured traffic flow coding success on the node as routing information.

6. References

- [1] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communication Magazine*, vol. 43, no. 9, pp. S23–S30, September 2005. [Article \(CrossRef Link\)](#).
- [2] R. Ahlswede, N. Cai, S.-Y. R. Li, and Raymond W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000. [Article \(CrossRef Link\)](#).
- [3] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "MORE: A Network Coding Approach to Opportunistic Routing," retrieved March, 2015. [Article \(CrossRef Link\)](#)

- [4] 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, no. 3, pp. 497–510, June 2008. [Article \(CrossRef Link\)](#).
- [5] M. A. Iqbal, B. Dai, B. Huang, A. Hassan, and S. Yu, "Survey of network coding-aware routing protocols in wireless networks," in *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1956–1970, August 2011. [Article \(CrossRef Link\)](#).
- [6] K. Alic, A. Svirgelj, "Self-Adaptive Practical Opportunistic Network-Coding Procedure for Static Wireless Mesh Networks", in *Ad Hoc & Sensor Wireless Networks* 36(1-4): 87-105, 2017.
- [7] K. Alic, E. Pertovt, and A. Svirgelj, "Bearing-Opportunistic Network Coding," *International Journal of Computers, Communications & Control (IJCCC)*, vol. 7, no. 2, pp. 42–53, April 2015.
- [8] J. Le, J. C. S. Lui, and D.-M. Chiu, "DCAR: distributed coding-aware routing in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 596–608, April 2010. [Article \(CrossRef Link\)](#).
- [9] L. Yifei, S. Cheng, X. Qin, and T. Jun, "ICM: a novel coding-aware metric for multi-hop wireless routing," in *Proc. of Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom '09)*, pp. 1–4, Beijing, China, September 2009.
- [10] Z. Wang, "Coding-aware and interference-avoid routing metric for wireless mesh networks," in *Proc. of Proceedings of the International Conference on Advanced Intelligence and Awareness Internet (AIAI 2010)*, pp. 424–427, Beijing, China, October 2010.
- [11] S. Sengupta, S. Rayanchu, and S. Banerjee, "Network coding-aware routing in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1158–1170, August 2010. [Article \(CrossRef Link\)](#).
- [12] J. Zhan, and Q. Zhang, "Cooperative Network Coding-Aware Routing for Multi-Rate Wireless Networks," in *Proc. of Proceedings of IEEE INFOCOM 2009*, pp. 181–189, Rio De Janeiro, Brazil, April 2009.
- [13] B. Ni, N. Santhapuri, Z. Zhong, and S. Nelakuditi, "Routing with opportunistically coded exchanges in wireless mesh networks," in *Proc. of Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh '06)*, p. 157–159, Reston, VA, USA, September 2006. [Article \(CrossRef Link\)](#).
- [14] C. Lu, S. Xiao and Y. Miao, "OQMCAR: An Enhanced Network Coding-aware Routing Algorithm Based on Queue State and Local Topology," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 8, pp. 2875-2893, 2015. [Article \(CrossRef Link\)](#).
- [15] X. Jiao, X. Wang, and X. Zhou, "Active network coding based high-throughput optimizing routing for wireless ad hoc networks," in *Proc. of Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, p. 1–5, Dalian, China, pp. 1–5. October 2008. [Article \(CrossRef Link\)](#).
- [16] X. Wei, L. Zhao, J. Xi, and Q. Wang, "Network coding aware routing protocol for lossy wireless networks," in *Proc. of Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom '09)*, pp. 1–4, Beijing, China, September 2009. [Article \(CrossRef Link\)](#).
- [17] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. of Proceedings of the 2th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, Washington, DC, USA, February 1999. [Article \(CrossRef Link\)](#).
- [18] Y. Yan, Z. Zhao, B. Zhang, H.T. Mouftah, and J. Ma, "Rate-adaptive coding-aware multiple path routing for wireless mesh networks," in *Proc. of Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2008)*, pp. 1–5, New Orleans, LA, USA, December 2008. [Article \(CrossRef Link\)](#).
- [19] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, Chapter 5, Kluwer Academic Publisher, pp. 153–181, 1996.
- [20] B. Guo, H. Li, C. Zhou, and Y. Cheng, "Analysis of general network coding conditions and design of a free-ride-oriented routing metric," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1714–1727, May 2011. [Article \(CrossRef Link\)](#).

- [21] M. Jhang, S. Lin, and W. Liao, “C2AR: coding and capacity aware routing for wireless ad hoc networks,” in *Proc. of Proceedings of the IEEE International Conference on Communications (ICC 2010)*, pp. 1–5, Cape Town, South Africa, May 2010. [Article \(CrossRef Link\)](#).
- [22] Y. Peng, Y. Yang, X. Lu, and X. Ding, “Coding-aware routing for unicast sessions in multi-hop wireless networks,” in *Proc. of Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2010)*, pp. 1–5, Miami, FL, USA, December 2010. [Article \(CrossRef Link\)](#).
- [23] L. F. Xie, P. H. J. Chong, S. C. Liew, and Y. L. Guan, “CEO: Consistency of Encoding and Overhearing in Network Coding-Aware Routing,” *IEEE Wireless Communications Letters*, vol. 2, no. 2, April 2013.
- [24] R. Hou, S. Qu, K.-S. Lui, and J. Li, “Coding- and interference-aware routing protocol in wireless networks,” *Computer Communications*, vol. 36, no. 17-18, pp. 1745–1753, November-December 2013. [Article \(CrossRef Link\)](#).
- [25] C. Yang, and D. P. Agrawal, “Hycare: Hybrid coding-aware routing with etox metric in multi-hop wireless networks,” in *Proc. of Proceedings of the 10th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2013)*, pp. 136–144, Hangzhou, China, October 2013.
- [26] J. Chen, Q. Yuan, R. Du, and J. Wu, “MuCAR: A Greedy Multi-flow-based Coding-Aware Routing in Wireless Networks,” in *Proc. of Proceedings of the IEEE International Conference on Sensing, Communication, and Networking (SECON 2015)*, pp. 20–27, Seattle, USA, June 2015. [Article \(CrossRef Link\)](#).
- [27] “Riverbed web page,” retrieved August, 2017. [Article \(CrossRef Link\)](#).
- [28] K. Alic, E. Pertovt, and A. Svingelj, “Simulation environment for network coding,” in *Proc. of Proceedings of the Mosharaka International Conference on Communications, Networking and Information Technology (MICCNIT 2011)*, pp. 25C–231, Dubai, UAE, December 2011.
- [29] K. Alic, E. Pertovt, and A. Svingelj, “Network coding simulation model in OPNET Modeler,” in *Proc. of OPNETWORK 2012*, Washington, USA, September 2012.
- [30] K. Chi, X. Jianf, S. Horiguchi, “Network Coding Opportunity Analysis of COPE in Multihop Wireless Networks,” in *Proc. of Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2008)*, pp. 2858–2863, Las Vegas, NV, USA, April 2008. [Article \(CrossRef Link\)](#).



Erik Pertovt was awarded his Ph.D. from the Jozef Stefan International Postgraduate School, Slovenia, in 2015. Now he is with BANKART, d.o.o. His recent research interests are in the fields of network protocols, architectures, and cross-layer optimization for wireless and wired networks.



Kemal Alič was awarded his Ph.D. from the Jozef Stefan International Postgraduate School, Slovenia, in 2016. His recent research interests are in the fields of network protocols, architectures, and cross-layer optimization for wireless and wired networks, and large networks optimization. He is currently employed at Metronik, d.o.o.



Aleš Švigelj was awarded a Ph.D. from the Faculty of Electrical Engineering, University of Ljubljana, in 2003. He is a research fellow in the Department of Communication Systems at the Jozef Stefan Institute and associate professor at the Jozef Stefan International Postgraduate School. He has extensive research in modelling, simulation, and design of advanced telecommunications elements, systems, and services. His current work focuses on advanced networking technologies for wireless systems and smart grids.



Mihael Mohorčič was awarded a Ph.D. from the Faculty of Electrical Engineering, University of Ljubljana, in 2002. He is a scientific advisor and head of the Department of Communication Systems at the Jozef Stefan Institute, and associate professor at the Jozef Stefan International Postgraduate School. His research and working experience include development and performance evaluation of network protocols and architectures for mobile and wireless communication systems, and resource management in terrestrial, stratospheric and satellite networks. His recent research interest is focused on cognitive radio networks, “smart” applications of wireless sensor networks, dynamic composition of communication services and wireless experimental testbeds.