https://doi.org/10.1016/j.comnet.2015.08.044

A Survey of Inter-Flow Network Coding in Wireless Mesh Networks with Unicast Traffic

[†]L. F. Xie, [††]Peter. H. J. Chong, [*]Ivan W. H. Ho, and [††]Y. L. Guan

[†]Faculty of Electrical Engineering and Computer Science, Ningbo University, China
[††]School of EEE, Nanyang Technological University, Singapore
[*]Department of EIE, The Hong Kong Polytechnic University, Hong Kong
Email: xiel0002@ntu.edu.sg
Email: ehjchong@ntu.edu.sg
Email: ivanwh.ho@polyu.edu.hk
Email: eylguan@ntu.edu.sg

[†]*Corresponding Author*: Peter Han Joo Chong, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

Email: ehjchong@ntu.edu.sg Tel: (+65) 6790 4437 Fax: (+65) 6792 0415

Abstract: Wireless network performance is much restricted by the unreliability of the wireless channel and the competition among different flows for the shared network resources such as the bandwidth. Network coding is a technique that exploits the broadcast of the wireless channel and can effectively address these two restrictions to improve network performance. For example, with network coding, an intermediate node of multiple flows can encode packets from these flows into one mixed packet and serve these flows using only one transmission instead of multiple transmissions in the traditional way, thus mitigating the competition among flows. Inter-flow network coding (XNC), as a form of network coding, considers encoding packets from different flows, and it can benefit wireless mesh networks (WMNs) with either reliable or lossy links. In this paper, we present a survey on XNC in WMNs for unicast traffic, with various design factors related to XNC being covered. Specifically, our survey considers two types of WMNs, one with reliable links and the other with lossy links, and we study how XNC can be effectively utilized in both two types of WMNs. In addition to the benefits of XNC, we also present in this survey some drawbacks of applying XNC in

WMNs. With this paper, we believe that readers will have a more thorough understanding of XNC on how it effectively mitigates the resource competition among flows and the channel unreliability problem in WMNs.

Keywords: inter-flow network coding, wireless mesh networks, unicast traffic, broadcast channel, channel unreliability

1. Introduction

The design of wireless networks is challenging, primarily due to the fading and broadcast nature of wireless channels, as well as the sharing of network resources [1]. First, similar to wired networks, resources such as the bandwidth in wireless networks are shared among different flows. Flows at one common intermediate node need to compete for the use of the common channel whenever a transmission opportunity arises. This competition imposes a major constraint on network performance. Second, as opposed to the stable and reliable wired channel, wireless channels generally experience drastic variation due to channel fading, causing each transmission prone to be a failure. In this situation, retransmissions generally serve as a necessary way to guarantee the packet delivery. However, retransmissions decrease the efficiency of exploiting the bandwidth. In addition, wireless channels are broadcast channels, which could cause interference between simultaneous transmissions. This interference problem further restricts the efficiency of using the bandwidth. Fortunately, the broadcast channel on the other hand can be exploited to mitigate the first two problems to improve network performance, as will be seen from network coding later.

Network coding [2] is a promising technique for increasing wireless network performance. Proposed in [2] for achieving multicast capacity in wired networks, network coding introduces a new paradigm of packet processing and forwarding. That is, network coding allows an intermediate node to *encode or mix* different received packets into one single *coded* packet and send the coded packet along an outgoing channel. Clearly, this new paradigm is different from traditional non-coding scheme, where packets are simply replicated and then separately forwarded at intermediate nodes. Importantly, this new paradigm of network coding can be successfully applied to wireless environments, e.g., wireless mesh (or ad-hoc) networks (WMNs), to improve network performance even under unicast traffic [3, 4]. Regarding the packets to be encoded together, network coding can be classified into inter-flow network coding (XNC) and intra-flow network coding (INC). XNC considers encoding packets from different flows whereas INC considers encoding packets from the same flow. It can be found in the literature that both XNC and INC [5, 6] can be used effectively for boosting wireless network performance, and some works even consider hybrid approaches that exploit both XNC and INC [7, 8, 53, 54]. This is because both XNC and INC can effectively address the first two challenges mentioned above. However, these hybrid approaches [7, 8, 53, 54] show that XNC and INC have distinctly different design principles. In this paper, we put emphasis on illustrating XNC in wireless networks.

So far, a body of XNC schemes have been proposed in the literature for WMNs. A WMN [55] consists of mesh clients, mesh routers, and mesh gateways, and it provides a cost-effective solution for the last-mile Internet access. In WMNs, the clients are connected by the routers and gateways in a multi-hop fashion, and one major challenge is how to achieve high network throughput [55]. Various studies [14-17] have shown that XNC is a powerful performance booster in WMNs under unicast traffic, and that the design of the XNC scheme generally involves several important design issues or factors, as will be discussed later in this paper.

Currently, there are some works on surveying network coding [10-13], but none of them is specifically on XNC for unicast traffic in WMNs. Thus, due to the usefulness of XNC in WMNs and those identified design factors, we are motivated to give a survey on this topic in this paper. Our survey consists of two major parts, i) XNC in WMNs under reliable links; and ii) XNC in WMNs under lossy links. Here, by a lossy (reliable) link, we mean a wireless link between two nodes does (not) suffer from the channel fading and the retransmission problem. In each of the two parts, we will first present a typical XNC example to show how an intermediate node in XNC forwards packets received from multiple flows and compare it to that in traditional non-coding scheme, and then we will present the survey in detail on how XNC could benefit WMNs more. In addition to the advantages of XNC, we also present in this paper some drawbacks of applying XNC in WMNs as well.

2. XNC in WMNs under Reliable Links

Before proceeding to the survey, we first show a typical example of XNC in WMNs under reliable links. Consider a simple scenario of only two flows in Fig. 1 where flow 1 (f_1) and flow 2 (f_2) intersect at nodes 1, 2 and 3. Suppose each link is reliable and each node is required to store all data packets it has already sent. We now illustrate how the intermediate node 2 handles the packet forwarding after it receives P_1 from f_1 and P_2 from f_2 in traditional non-coding scheme and XNC, respectively.

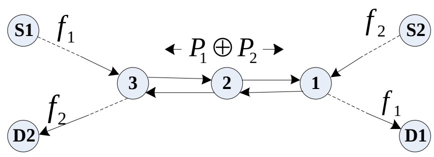


Fig. 1: An example of XNC under reliable links in WMNs.

Traditional non-coding scheme: Here, for any received packet, node 2 simply forwards it to its next hop, as mentioned earlier. Thus, P_1 and P_2 are sent individually by node 2 to nodes 1 and 3, respectively. This means f_1 and f_2 compete for the channel use at node 2 and two transmission times are needed to forward P_1 and P_2.

XNC: Here, we consider a simple packet encoding scheme, i.e., XOR. Node 2 XORs P_1 and P_2 into a coded packet, $P_1 \oplus P_2$, and then broadcasts $P_1 \oplus P_2$ to both nodes 1 and 3. Upon receiving $P_1 \oplus P_2$, node 1 (3) XORs it with the stored packet P_2

(P_1) to yield P_1 (P_2), which is what node 1 (3) desires to receive. Consequently, with XNC exploiting the broadcast channel, one transmission of node 2 serves f_1 and f_2 simultaneously, which mitigates or even eliminates the competition between f_1 and f_2 for the channel use at node 2, and thus improves network throughput as compared with traditional non-coding scheme.

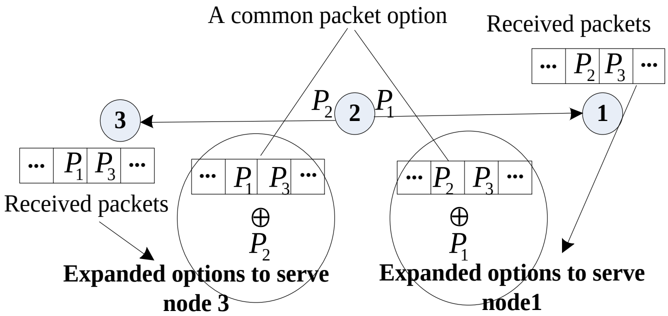


Fig. 2: Rationale behind XNC.

We expose the rationale behind the example of XNC in Fig. 1. Let us consider a scenario where one node, say node 1, demands a packet, P_1, from another node, say node 2. Traditionally, node 2 has only one option to satisfy the demand of node 1, i.e., node 2 only considers sending P_1 to node 1. However, the option at node 2 can actually be diversified by considering the packets previously received at node 1. For example, if node 1 has received a bundle of packets before, then, node 2 can encode any packet from that bundle with P_1 to form an optional encoded packet and send it to node 1. Thus, the packet options that node 2 can choose to satisfy node 1's demand are expanded. Further, in case of multiple different demands as shown in Fig. 2, node 2, with the expanded packet options for satisfying each demand, may find a "common packet option" to simultaneously serve them. If a common packet option is found, it means there is a network coding opportunity and thus, as shown in Fig. 1, the network throughput will be improved. Particularly, the improved performance is attained even with less energy consumption due to the reduced transmission times at the intermediate node. In fact, XNC is such a technique that exploits this commonness of packet options for different flows. The common packet option, however, only exists in some certain situations called *coding structures*, which will be introduced below.

Intuitively, the more coding opportunities (i.e., common packet options) we can find, the more improvement we can make to network performance. Hence, our survey here focuses on various ways to increase coding opportunities in WMNs. Fig. 3 shows various factors, including the determinant factor, the positive factors, and the counteractive factors, that will affect the coding opportunities to be found for two or more intersecting flows. In particular, the determinant factor and the positive factors can be considered for increasing coding opportunities whereas the counteractive factors are used to moderate the greediness of the network for coding opportunities. In what follows, we first explain how each factor in Fig. 3 affects the coding opportunities to be found. Then, a table is used to summarize how each proposed representative XNC scheme considers these factors. Finally, we end this Section with discussions.

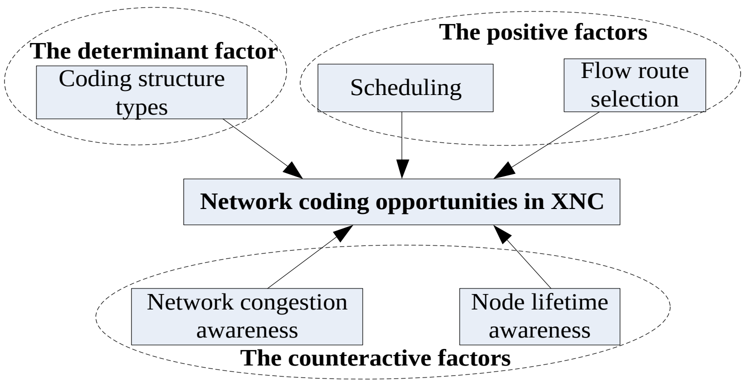


Fig. 3: Factors affecting coding opportunities in XNC.

A. *The Determinant Factor*

As seen in Fig. 3, the determinant factor refers to the coding structure type, which actually reflects the conditions of the coding opportunity in XNC. Hence, this factor is essential to XNC because it answers how different flows (e.g., f_1 and f_2 in Fig. 1) should intersect with each other such that the intersection node (e.g., node 2 in Fig. 1) can encode packets from them. In the literature, a number of coding structure types have been proposed, and they are enumerated as follows.

Two-hop coding structure [14]: This coding structure type considers encoding packets from N flows with $N \geq 2$. Suppose node C is the intersection node of these flows, and by $prev(i)$ and $next(i)$ we denote the previous hop and the next hop of f_i ($1 \leq i \leq N$) at node C, respectively. The conditions for node C to be an encoding node of the N flows are as follows: for any i and j ($i \neq j$), we must have $next(i) = prev(j)$ or $next(i) \in Ngb(prev(j))$, where $Ngb(prev(j))$ denotes all one-hop neighbors of the node, $prev(j)$. These conditions ensure the node $next(i)$ to obtain packet P_j from f_j and then decode P_i from $P_1 \oplus \ldots \oplus P_N$ mixed by node C.

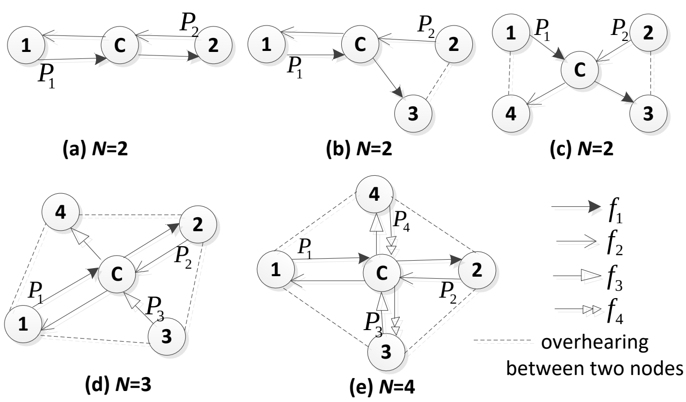
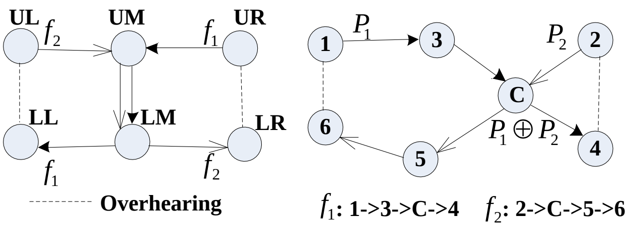


Fig. 4: Two-hop coding structures for different N.

Fig. 4 plots several topologies of the two-hop coding structure for different N. Fig. 4(a) is called reverse carpooling [15]. Let us explain why node C in Fig. 4(c) is an encoding node of f_1 and f_2. Suppose nodes 1 and 2 take turns to send P_1 and P_2 to node C, respectively, then their neighbors, node 4 on f_2 and node 3 on f_1, obtains P_1 and P_2, respectively. This process is termed 'overhearing', which allows one node to hear the packet transmission of its neighbors because of the broadcast channel. Thus, similar to Fig. 1, node C can broadcast the coded packet $P_1 \oplus P_2$ to nodes 3 and 4 to serve f_1 and f_2. Similarly, with overhearing in either Fig. 4(d) or (e), after node C receives P_i for all i, the next hop of node C on any flow already obtains the packets of all the other flows, thus allowing node C to mix all packets into $P_1 \oplus \ldots \oplus P_N$.



(a) Butterfly coding structure (b) One generalized coding structure

Fig. 5: Butterfly coding structure and the generalized coding structure.

Butterfly coding structure [16]: This is a specific structure and it only relaxes the two-hop coding structure in Fig. 4(c) by allowing the packet decoding at the node two hops away rather than one hop away from the encoding node. Fig. 5(a) shows the butterfly structure where nodes UM and LM are the two intersection nodes of f_1 and f_2 and node LL (LR) on f_1 (f_2) can overhear node UL (UR) on f_2 (f_1). Within this structure, node UM will encode packets P_1 and P_2 sent from nodes UR and UL, respectively, and send the mixed packet $P_1 \oplus P_2$ to node LM. Then, node LM will broadcast this packet to nodes LL and LR, where the packet decoding will take place. Here, we can see that the decoding nodes, LL and LR, are two hops away from the encoding node, UM. Actually, this butterfly structure can be easily extended to the generalized coding structure as shown below.

Generalized coding structure [17]: The authors in [17] propose the sufficient conditions for the intersection node of two flows to serve as the encoding node. The coding structure formed under these conditions is called the generalized coding structure [17]. Suppose the intersection node is node c, and we denote $Ngb(a)$,

$U(a,f)$, and $D(a,f)$ as node a's one-hop neighbors, all node a's upstream nodes on flow f, and all node a's downstream nodes on flow f, respectively. Then, node c will become an encoding node when the following two conditions are satisfied: (1) there exists a node $d_1 \in D(c,f_1)$, such that $d_1 \in U(c,f_2)$ or $d_1 \in Ngb(s_1)$, where $s_1 \in U(c,f_2)$; and (2) there exists a node $d_2 \in D(c,f_2)$, such that $d_2 \in U(c,f_1)$ or $d_2 \in Ngb(s_2)$, where $s_2 \in U(c,f_1)$. In Fig. 5(b), for example, node 6 on f_2 and node 4 on f_1 can overhear node 1 on f_1 and node 2 on f_2, respectively. Thus, we have $c = C$, $d_1 = 4$, $s_1 = 2$, $d_2 = 6$, and $s_2 = 1$, meaning that the two conditions above are met. Hence, node C can encode P_1 from f_1 and P_2 from f_2 into $P_1 \oplus P_2$, which will be decoded at nodes 4 and 6, respectively.

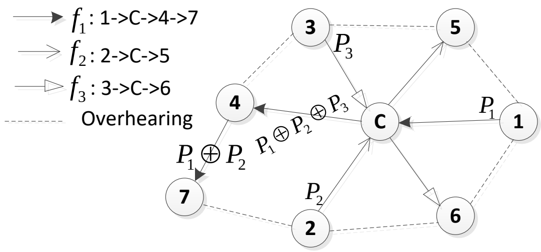


Fig. 6: A generalized coding structure for three flows.

Although the two conditions above are used for identifying the coding opportunity between two flows only, it can be easily extended to N flows with $N \geq 3$ [17]. When an intersection node of the N flows is an encoding node for any two flows among the N flows, it is allowed to encode N packets from these N flows into one coded packet. However, the coding structure for $N \geq 3$ could be very complicated. For example, one coded packet may need to be jointly decoded by multiple downstream nodes on one flow, whereas all abovementioned coding structures restrict the packet decoding at only one node. The authors in [18] propose such a coding structure for $N = 3$ as shown in Fig. 6. In Fig. 6, after node C receives P_i from node i, $1 \leq i \leq 3$, it mixes them into $P_1 \oplus P_2 \oplus P_3$ and broadcasts it. Then, this coded packet is directly decoded at node 5 on f_2 and node 6 on f_3. For f_1, however, the coded packet needs to be decoded at nodes 4 and 7 jointly, that is, node 4 first decodes $P_1 \oplus P_2$ from $P_1 \oplus P_2 \oplus P_3$, and node 7 then decodes the desired packet P_1 from $P_1 \oplus P_2$.

Grail coding structure [56]: In generalized coding structure, e.g., in Fig. 5(b), the decoding node 6 of f_2 (node 4 of f_1) always overhears P_1 (P_2) earlier than it receives the coded packet $P_1 \oplus P_2$, and thus $P_1 \oplus P_2$ is decoded immediately upon its arrival at the decoding nodes. By contrast, the grail coding structure [56] allows the earlier arrival of $P_1 \oplus P_2$, and it only modifies condition (2) in the generalized coding structure as follows: there exists a node $d_2 \in D(c, f_2)$, such that $d_2 \in Ngb(s_2)$, where $s_2 \in \{d_1\} \cup D(d_1, f_1)$ and $s_2 \neq D_1$ (the destination node of f_1). This new condition (2) states that the decoding node d_2 of f_2 should be able to overhear d_1 or any node between d_1 and the destination node of f_1. Fig. 7 illustrates an example of the grail coding structure, where we have $c = C$, $d_1 = s_2 = 3$, $d_2 = 4$, and $s_1 = 2$. Fig. 7 shows that node 4 on f_2 receives the coded packet $P_1 \oplus P_2$ earlier than it overhears P_1 from node 3 on f_1, causing the decoding of $P_1 \oplus P_2$ to be delayed for some time.

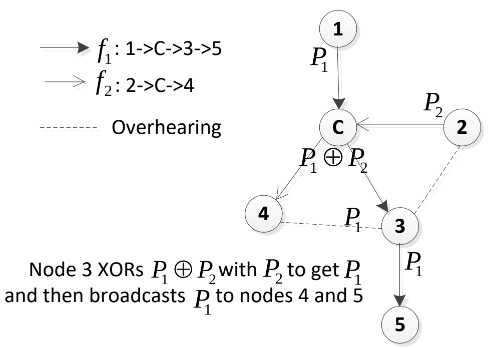
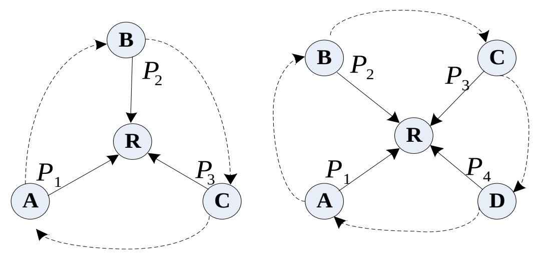


Fig. 7: A grail coding structure.

***k*-tuple coding structure** [19]: One common feature in the abovementioned coding structure types is that the encoding node needs to transmit only one encoded packet to serve all flows, even when $N \geq 3$. Actually, there is another kind of coding structure that requires an encoding node to transmit multiple encoded packets. This is the k-tuple coding structure [19]: in this structure, to serve k flows, $k-1$ encoded packets need to be transmitted, which saves one transmission as compared to traditional routing. Fig. 8(a) shows one example of the 3-tuple coding structure. Let us look at how XNC is performed here. In this simple structure, node A sends a packet, say P_1, to node B via node R; node B sends a packet, say P_2, to node C via node R; and node C sends a packet, say P_3, to node A via node R. After node R received all the three packets, it broadcasts the following two encoded packets to all the three neighbors: $P_1 \oplus P_2$ and $P_2 \oplus P_3$. Then, each receiver can get its desired packet by XORing the three packets, i.e., the two received encoded packets and the packet it sent to node R. For example, node A can get packet P_3 by doing $P_1 \oplus (P_1 \oplus P_2) \oplus (P_2 \oplus P_3)$. The similar happens at both nodes B and C.



(a) 3-tuple coding structure (b) 4-tuple coding structure

Fig. 8: Examples of k-tuple coding structure.

The k-tuple coding structure actually considers XNC in a scenario where k nodes demand traffic from each other via a common relay node R. However, the traffic cannot be random; it should follow a certain pattern that each packet source (or each demander) should be demanded by only one other node [19]. Fig. 8(b) shows an example of the 4-tuple coding structure, which presents the abovementioned traffic pattern.

In k-tuple coding structure, the relay R performs network coding as follows [19]. Suppose the demanded packets at the k nodes are P_1, …, and P_k, then, node R will generate the $k-1$ encoded packets as $P_1 \oplus P_2$, $P_2 \oplus P_3$, …, and $P_{k-1} \oplus P_k$. Finally, each demander can get the packet it desires by using the $k-1$ received encoded packets and the packet it sent to the relay.

Remarks on coding structure types: First, all the above coding structure types present different XNC conditions of encoding packets from multiple flows. It is obvious that the more the coding structures are considered, the more the coding opportunities can be found. However, we find in the literature that the most widely used coding structures are those simple structures for only two flows, e.g., the structures shown in Fig. 4 for $N = 2$. The reasons as follows. On one hand, to detect those complicated structures for three or more flows generally incurs more overhead. On the other hand, due to the geometrical constraints on the coding structure [20], the probability for any coding structure of N flows with $N \geq 4$ to appear in reality is very low [20]. Second, besides the XOR operation, random linear combination (RLC) [4, 54, 57] can be used equivalently in packet encoding in all the above coding structures. In Fig. 1, for example, node 2 can use RLC to mix P_1 and P_2 into $P = c_1P_1 + c_2P_2$, where c_1 and c_2 are random coefficients in finite fields [3]. With P and P_1 (P_2), node 3 (1) can extract P_2 (P_1). We find from the literature that RLC is generally used in INC (or INC combined with XNC) [5-8, 53, 54] to combat the lossy links, whereas XOR is used in most XNC schemes (e.g., those surveyed in this paper). We will also discuss the use of RLC in XNC under lossy links later.

B. The Scheduling Factor

The coding structures introduced above define the situations where packets from two or more flows can be encoded. However, it does not mean the encoding node can encode multiple packets from different flows in each of its transmissions, especially when *opportunistic scheduling* (or random access) such as the IEEE 802.11 standard is used in the network. For example, in Fig. 4(a), when node C is about to send a packet from f_1, it is highly possible that there happens to be no queued packet from f_2. Then, no network coding can be performed at node C, and thus network performance cannot be maximally enhanced. Hence, this raises the question of how to maximally utilize the coding opportunity in XNC. From the literature, we find that a major factor that affects the coding opportunities to be utilized is scheduling. In what follows, we present different scheduling schemes in XNC.

Centralized scheduling: Suppose in Fig. 4(a) node 1 on f_1 and node 2 on f_2 are sending traffic to each other at rates r_1 and r_2, respectively, then, the maximal coding opportunities that can be utilized in one unit time at node C will be $min(r_1, r_2)$. This upper bound of the amount of the coding opportunity actually can be achieved by the centralized scheduling. We find that this scheduling generally appears in the flow rate or energy optimization of XNC [15, 21-23]. However, the big challenges here involve (1) the finding of the maximal independent sets (MIS) [21] and (2) the distributed implementation of the centralized scheduling. A MIS consists of all links that can be scheduled simultaneously, but finding all MIS in a network is NP-hard [60]. One may maintain a small MIS pool to address (1) [23]. To ease challenge (2), a central coordinator could be introduced to the network [24], responsible for computing flow routes and allocating bandwidth to each flow.

Back-pressure based scheduling: Towards distributed optimal scheduling for flow rate optimization [56, 58] or energy optimization in XNC [59], back-pressure based scheduling is generally considered. This scheduling stems from the dual problem of the original optimization problem, and gives a new perspective for solving the optimization problem. Let $Q_i^f(t)$ denote the queue length at node i for flow f at time t, the back pressure along the link from node i to node j for flow f could be represented by $d_{i,j}^f(t) = Q_i^f(t) - Q_j^f(t)$. Depending on how the optimization problem is formulated, the back pressure over the link from node i to node j could be given by $d_{i,j}(t) = \sum_f (Q_i^f(t) - Q_j^f(t))$ [56] or $d_{i,j}(t) = \max_f (Q_i^f(t) - Q_j^f(t))$ [58, 59]. This scheduling opts for the links with higher product of the transmission rate and back pressure; the broadcast link in XNC usually has higher product [59]. Given nodes' queue lengths and link transmission rates, this scheduling aims to find the MIS that leads to the maximal sum of the above product. It is not a trivial task for this aim, even with a centralized coordinator/controller, but fortunately there are many works on the distributed algorithm of this scheduling (see the references in [58, 59]). We omit the details here.

noCoCo scheduling [25]: The authors in [25] propose a scheme called Near-Optimal Coordinated Coding (noCoCo) to maximally utilize coding opportunities for a special kind of traffic pattern, namely, the two-way traffic, where two nodes send packets to each other on the same path but in opposite directions. Fig. 9(a) shows one example of this traffic pattern, where node 0 and node 3 send packets to each other. The goal in [25] is to propose a proper distributed scheduling to maximally increase the chance for each intermediate node or relay node to send encoded packets. One important feature in noCoCo scheduling is the use of a backpressure rule, with which each intermediate node is not allowed to store in its buffer more than one packet of each flow. To follow this rule, each intermediate node only needs to know the packets stored in its neighbors' queues to determine which flow's packet to send. With the backpressure rule, it is stated in [25] that after injecting sufficient packets from the two flows, the network will eventually reach the desired state that some intermediate node, say node X, has stored two packets from two flows, and its left (right) neighbor stores one packet to the right (left) destination, as shown in Fig. 9(b). Then, node X can encode the two packets and broadcast the coded packet to its two neighbors. After that, node X will refrain from transmission until each of its two neighbors performs a transmission, which will ensure node X to perform network coding again in its next transmission. The similar happens at other intermediate nodes.

Although the simulation in [25] shows the benefit of XNC under noCoCo scheduling, this scheduling has the following limitations. First, it may not work well when one flow can be encoded with a *third* flow at some relay node. Consider three flows, say f_1, f_2, and f_3, intersecting at a relay node R where f_1 can be coded with either f_2 or f_3. Then, with the backpressure rule in noCoCo, node R is only allowed to store one packet from one flow. However, it will be preferable for node R to store two packets from f_1 for more coding opportunities. Thus, in this situation noCoCo will miss some coding opportunities [26]. Second, it may not be able to work effectively under random traffic patterns where flow paths may not be overlapped.

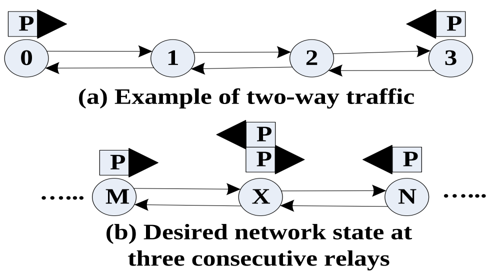


Fig. 9: Scheduling in two-way traffic scenario.

Random access with different priorities [20, 27-29]: The scheduling presented here is based on random access among nodes. In this scheduling, one key factor that affects coding opportunities is the node priorities in accessing the channel. All the works [20, 27-29] we survey here consider random access in the two-hop coding structure, and they generally employ Markov chain for analysis.

The authors in [20] have studied the performance of the two-hop coding structure with N flows under two types of scheduling (equal access among nodes and higher priority for the encoding node). They show that equal access leads to a higher network throughput in case of saturated sources. This is because each time when the encoding node occupies the channel, it generally can encode more packets into one coded packet with equal access rather than higher-priority access, thus resulting in higher network throughput. Similar findings are presented in [29] for the two-hop coding structure.

In addition, the authors in [27] and [28] analyze how the transmission probability of the encoding node, p_r, affects the coding opportunities and the network throughput in the coding structure of Fig. 4(a). Although their studies are based on Slotted ALOHA and Carrier Sense Multiple Access (CSMA), respectively, they draw a conclusion similar to [20] that a moderate p_r will result in the highest network throughput: a too high p_r will result in unsaturated queues at the encoding node and thus few coding opportunities at it; and a too low p_r actually turns the encoding node into the bottleneck of the network.

Random access with queue management: The authors in [60, 61] consider queue management for TCP traffic in random access to increase coding opportunities. The sensitivity of TCP traffic to the packet loss or collision in WMNs could cause low flow rate at the source and flow rate mismatch between flows at intermediate nodes, both reducing coding opportunities. To benefit more from XNC, two queue management strategies are proposed in the literature. First, [61] considers delaying the packet transmission for some time to enhance TCP throughput. It employs reverse carpooling to encode packets including ACK packets in TCP, and introduces a timer for each packet stored in the queue to postpone its transmission for a coding opportunity. [61] shows that only with a small delay, TCP throughput benefits from XNC. This is because an increasing delay yields a longer round-trip time (RTT), which then hurts TCP throughput. [61] also shows that an optimal delay varies with different network scenarios, e.g., different TCP flows. Obviously, we could also apply this delayed packet transmission to UDP traffic [62]. Second, [60] considers packet dropping at congested intermediate nodes to slow down a high-rate flow and solve the rate mismatch between flows to maximize coding opportunities among flows. The simulation results of [60] in the two-hop coding structure in Fig. 4 demonstrate improved TCP throughput with this packet dropping.

Remark on scheduling: Compared with the scheduling based on random access, the first several scheduling schemes such as the centralized scheduling and the back-pressure based scheduling generally yield more coding opportunities, but they have higher complexity for implementation. Considering the popularity of the 802.11 MAC in WMNs [55], the random access is more suitable for WMNs, and thus we could apply priorities and/or queue management for more coding opportunities in WMNs. However, there is still some challenges. For example, although it is stated in [20, 27-29] that we can adjust the transmission probability of the encoding node to achieve high network performance of only two-hop flows, it is still unknown whether the same adjustment above can be applied to the scenarios with multi-hop flows.

C. The Flow Route Selection Factor

As shown in the coding structures above, network flows, to be mixable with each other at one intersection node, need to properly intersect with each other. However, in WMNs there normally exist multiple available paths for each incoming flow. Thus, this begets the question of how to select the path for a flow so as to form some coding structures and benefit from XNC. In what follows, we survey the approaches that deal with the flow route establishment for more coding opportunities.

Network backbone construction [30, 31]: A multi-hop WMN is generally considered as a flat or infrastructureless network, where each node plays the same role in delivering packets. Hence, there normally exist multiple paths for each incoming flow in this kind of flat network. By contrast, an infrastructure based network always requires the traffic to be directed to the infrastructure nodes first. Due to the limited number of infrastructure nodes in the network, they can form a common path segment for different flows, with the traffic running on them in opposite directions, and thus a number of coding structures shown in Fig. 4(a) will appear on this segment. Motivated by this, the works in [30, 31] aim to maximally utilize network coding by selecting some nodes in the flat network to play as the infrastructure nodes or the dominators. By connecting these dominators, a network backbone can be formed, on which all traffic will be routed. The simulation in [30] shows the benefit of a backbone-based XNC over regular or traditional routing based XNC; the simulation in [31] shows that the backbone-based XNC also improves network performance even under broadcast traffic.

Local route switching [16, 32]: This approach is proposed under traditional routing such as Dynamic Source Routing (DSR) [33] to increase coding opportunities. Since traditional routing schemes do not consider coding opportunities to establish new flows during the path selection, the abovementioned coding structures may not appear for the flows. However, it is possible that some coding structure could appear by properly adjusting part of the established flow paths, thus making coding opportunities arise. We use the example in [32] for illustration. Two flows are initially established as shown in Fig. 10, and there is no coding opportunity at any node of these two flows. However, node B, with its neighbor information limited to two hops, can determine that by letting it forward the packet from node D to node E, the coding structure in Fig. 4(c) will appear. Thus, node B can ask node D to forward its packet to it instead of node H, and then it can send the packet using network coding.

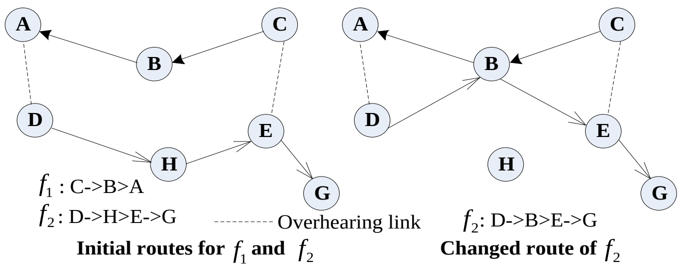


Fig. 10: Local route switching for coding opportunities.

Network coding-aware routing: Compared to the approach of local route switching, it is better to identify the coding opportunity during route establishment of a new incoming flow. This new routing is called network coding-aware routing [21]. By contrast, the routing that is oblivious to coding opportunities in the flow route setup and only identifies coding opportunities afterwards is called coding-oblivious routing [21]. The coding-aware routing aims to find a route or routes for a new incoming flow on which it can be mixed with existing flows. Generally, due to the benefit of XNC, a non-shortest route with coding opportunities on it may become an ideal option for a new flow. Now, we present some representative network coding-aware routing schemes as follows.

The works in [15, 21-23] formulate the joint scheduling and network coding-aware routing as a linear program. The formulation considers any traffic pattern of concurrent unicast flows. More specifically, [21, 23] aim to find the maximal network throughput for those flows, whereas [15, 22] aim to find the minimal energy consumption for delivering certain traffic amount of each flow. By solving the linear program, the upper bound of the throughput in [21, 23] and the lower bound of the energy consumption in [15, 22] can be obtained. For example, [21] shows that in many scenarios, network coding-aware routing achieves higher network throughput than coding-oblivious routing, both under *optimal scheduling*. Note that this higher throughput is obtained under optimal centralized scheduling. Moreover, [15] proposes a distributed way for achieving the lower bound of the energy consumption, of which we omit the details here.

Besides, various distributed network coding-aware routing schemes have been proposed, most aiming to find the best path for a new flow [17, 34-37]. Basically, we find that all these proposed schemes differ from each other in the coding structure types used for coding opportunity identification and the metrics used for path evaluation. As mentioned earlier, here the widely used coding structure types include the two-hop coding structure for two flows and the generalized coding structure for two flows. The path metrics used in these schemes mainly include (i) the path length in terms of the hop number, (ii) the amount of coding opportunities on the path, and (iii) the congestion state of the path. In general, more coding opportunities could be found on a longer route or on a more congested route. Thus, we find in these schemes that the quality of a path is not only measured by the coding opportunity amount: [34, 37] consider (i) and (ii) as the path metrics and [17, 36] consider (ii) and (iii) as the path metrics. As opposed to looking for the best path for a new flow, [35] considers using all possible paths and determines the flow rate on each path according to the coding opportunity amount on the path, as well as its congestion state.

D. The Counteractive Factors

Here, we present two counteractive factors, the congestion awareness and the node or network lifetime awareness, that are against the greediness of the network for coding opportunities. In other words, by considering either one of the two factors the coding opportunities will be decreased to some extent.

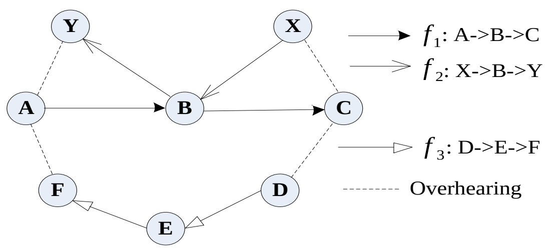


Fig. 11: Congestion awareness in coding-aware routing.

Even though coding opportunities can be effectively increased by applying the backbone-based routing or network coding-aware routing, it does not necessarily mean that network performance will always be improved. Indeed, the greediness for coding opportunities sometimes might harm network performance. As mentioned above, it is possible that a network with more coding opportunities could become more congested. Fig. 11 shows such an example where f_3, from node D to node F, is searching for a route with coding opportunities after f_1 and f_2 are established. It can be seen that f_3 will find coding opportunities at node B by selecting node B as the next hop. However, as observed in the figure, due to the traffic from f_1 and f_2 via node B, node B becomes much more congested than node E where no coding opportunity exists for f_3. Hence, a better choice for f_3 in this situation is to choose node E instead of node B as the next hop for higher throughput. From this example, we can see why the aforementioned coding-aware routing schemes [17, 35, 36] consider both the coding opportunity amount and the path congestion state when searching for a route for a new flow.

Besides, the authors in [38] point out another disadvantage of the greediness for coding opportunities. That is, although routing traffic for more coding opportunities can save the total energy consumption of the network, it causes the encoding node to deplete its energy at a much faster rate than other non-encoding nodes. For example, in the backbone-based routing, the nodes in the backbone will drain their energy faster than others because most traffic needs to be forwarded by them. In fact, the faster the energy depletion of some nodes, the shorter the network lifetime. Hence, to extend the lifetime of the network, it is necessary to be aware of the energy consumption of each relay node when routing traffic: the authors in [38] propose to adopt multi-path routing together with flow traffic splitting to distribute the traffic more evenly in the network and thus ensure the network lifetime.

E. Summary of the Proposed XNC Schemes

Now we summarize in Table 1 how each representative XNC scheme proposed under reliable links addresses the factors presented above. We see from the table that different schemes address these factors differently. In fact, for a better XNC design, we need to consider all these factors as a whole. In particular, we see that most centralized scheduling (or back-pressure scheduling) based schemes (e.g., [19, 21, 38, 56, 59]) usually employ coding-aware and congestion-aware routing, hence they could serve as the performance bound or benchmark for our future XNC design.

Table 1: Summary of the proposed XNC schemes.

Schemes	Coding structure (CS)	Scheduling	Flow route selection	Congestion awareness	Node lifetime awareness
COPE [14]	Two-hop CS	Opportunistic	Coding-oblivious routing	No	No

BFLY [16]	Butterfly CS	Opportunistic	Local route switching	No	No
DCAR [17]	Generalized CS	Opportunistic	Coding-aware routing	Yes	No
Jones [19]	*k*-tuple CS	Centralized	Coding-aware routing	Yes	No
Le [20]	Two-hop CS	Random access with different priorities	Coding-oblivious routing	No	No
Umehara [27, 28]					
Sengupta [21]	Two-hop CS ($N = 2$)	Centralized	Coding-aware routing	Yes	No
Gheibi [15]	Reverse carpooling			No	
Shabdanov [23]	Reverse carpooling			Yes	
Huang [22]	Two-hop CS ($N = 2$)			No	
Khreishah [56]	Generalized CS ($N = 2$) and grail CS	Back-pressure based	Coding-aware routing	Yes	No
Chaporkar [58]	Reverse carpooling	Back-pressure based	Coding-oblivious routing	No	No
Cui [59]	Two-hop CS	Back-pressure based	Coding-aware routing	Yes	No
noCoCo [25]	Reverse carpooling	noCoCo	Coding-oblivious routing	No	No
NCAQM [60]	Two-hop CS	Random access with queue management	Coding-oblivious routing	Yes	No
Huang [61]	Reverse carpooling				
BRONC [30]	Two-hop CS ($N = 2$)	Opportunistic	Backbone Construction	No	No
NCDS [31]	Two-hop CS	Opportunistic	Backbone Construction	No	No
MMSR [34]	Reverse carpooling	Opportunistic	Coding-aware routing	No	No
FORM [37]	Generalized CS ($N = 2$)	Opportunistic	Coding-aware routing	No	No
C^2AR [36]	Two-hop CS ($N = 2$)	Opportunistic	Coding-aware routing	Yes	No
RCR [35]	Reverse carpooling	Opportunistic	Coding-aware routing	Yes	No
Gaddam [38]	Reverse carpooling	Centralized	Coding-aware routing	Yes	Yes

F. Discussions

We have surveyed a number of factors which can affect the coding opportunities to be found in XNC under reliable links. Particularly, it can be seen that the coding opportunity identification highly depends on the coding structure types. However, it is quite possible that in some scenarios, two or more flows may not form any of the aforementioned coding structures. This begets the question of whether an intermediate node can still perform XNC when none of those coding structures is formed. This actually aims for the relaxation of the XNC conditions. We believe that an attempt for this relaxation should be worthwhile, because XNC can be exploited further in that manner. Here, we show one possible attempt using an example borrowed from [9, 39] where wired networks are considered.

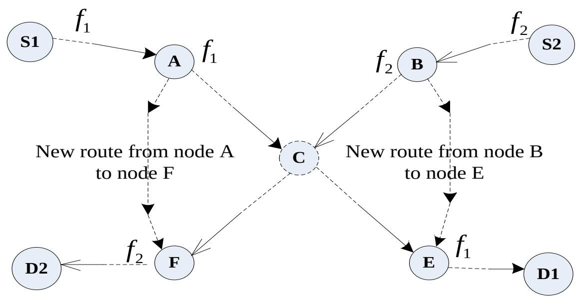


Fig. 12: An example of XNC with relaxed coding conditions.

Fig. 12 shows that network coding is still enabled at the intersection node, node C, of two flows (i.e., f_1 and f_2) even when no aforementioned coding structure is formed by the two flows. As compared to the generalized coding structure in Fig. 5(b), this example does not require any downstream node on f_1 (f_2) of node C to overhear some upstream node on f_2 (f_1) of node C. However, it guarantees the decoding of the packet encoded at node C by requiring some upstream node on f_1 (f_2) to deliberately send packets to some downstream node on f_2 (f_1) along some new route. This example shows one way to relax the XNC conditions, but several issues need to be further addressed for this relaxation. For example, we may need to determine 1) which upstream node on f_1 (f_2) should be selected to send packets; 2) which downstream node on f_2 (f_1) should be selected to receive packets; and 3) which route should be selected between the two nodes.

3. XNC in WMNs under Lossy Links

Similar to the survey in Section 2, we first show a typical XNC example in WMNs under lossy links. Still consider the scenario in Fig. 1, but assume the packet delivery probability from node i to node j is given by $p_{i\to j}$ $(0 < p_{i\to j} \leq 1)$. Now, let us compare the average total transmission time at node 2 for the packet forwarding between traditional non-coding scheme and XNC.

Traditional non-coding scheme: Node 2 needs to separately transmit P_1 and P_2 to node 1 and node 3, respectively, so the average total transmission time will be the sum of the expected transmission time from node 2 to node 1 and that from node 2 to node 3. Since the probability for P_1 to be correctly received at node 1 is $p_{2\to 1}$, and the probability for the acknowledgement of node 1 to be correctly received at node 2 is $p_{1\to 2}$, the expected transmission time from node 2 to node 1 is $1/(p_{1\to 2}\, p_{2\to 1})$ [40]. Likewise, the expected transmission time from node 2 to node 3 is $1/(p_{3\to 2}\, p_{2\to 3})$. Thus, the average total transmission time at node 2, $T_{routing}$, is given by

$$T_{routing} = 1/(p_{1\to 2}\, p_{2\to 1}) + 1/(p_{3\to 2}\, p_{2\to 3})$$

XNC: Instead of sending P_1 and P_2 separately, node 2 keeps broadcasting the coded packet, $P_1 \oplus P_2$, to nodes 1 and 3 at a given transmission rate until it receives the acknowledgements from both nodes. Then, we can derive the following expression for T_{nc}, the average total transmission time at node 2. Details of the derivation of T_{nc} can be found in [40].

$$T_{nc} = T_{routing} - \frac{1}{p_{1\to 2}p_{2\to 1} + p_{3\to 2}p_{2\to 3} - (p_{1\to 2}p_{2\to 1}p_{3\to 2}p_{2\to 3})}$$

Obviously, XNC here can save some retransmission time for node 2 to complete its tasks as compared to traditional non-coding scheme. The reason for the benefit of T_{nc} is that during the retransmissions of the mixed packet $P_1 \oplus P_2$ at node 2, any next hop node (i.e., node 1 or 3) that receives or overhears the coded packet can extract its desired packet; whereas during a packet retransmission in traditional routing, it is useless for a neighboring node (e.g., node 3) to overhear the retransmission destined to some other node (e.g., node 1). Hence, with XNC exploiting the wireless broadcast channel, an intersection node of two flows can reduce the total retransmission time. Similarly, it can be seen that XNC also works for other coding structures surveyed above *as long as there exists a coding opportunity or a common packet option which is useful for any next hop node*.

A. Classification of Coding Structures

With lossy links in WMNs, we need to classify the aforementioned coding structures into the following two types.

- The coding structure without overhearing links: This type requires no overhearing link in the coding structure. Obviously, the reverse carpooling in Fig. 4(a) and the *k*-tuple structure in Fig. 8 fall into this type.
- The coding structure with overhearing links: This type makes use of the overhearing link for packet decoding purpose. It can be seen that the two-hop coding structures from Fig. 4(b) to (e), the butterfly coding structure in Fig. 5, and the grail coding structure in Fig. 7 fall into this type.

The reason behind this classification is that under lossy link condition, the encoding node in the first type can always encode multiple packets from multiple flows, whereas it may be not allowed in the second type due to the unreliable overhearing links. In Fig. 4(c), for example, if node E fails to overhear P_1 from node A due to the unreliable link between them, then, node C cannot encode P_1 and P_2. Due to this uncertainty of the coding opportunity under lossy link condition, the design of XNC schemes for the second type will be different from that for the first type. We will illustrate below that the rate adaptation is differently considered in these two types of coding structures for maximizing the benefit from XNC.

B. XNC with Fixed Transmission Rate

Here, we consider XNC schemes with fixed transmission rate under lossy links. The XNC schemes for the first and second coding structure types will be surveyed.

***XNC for the First Coding Structure Type*:** We find that most factors surveyed in Section 2 are still applicable to XNC schemes for the first coding structure type, due to the certainty of the coding opportunity in this type and the benefit of XNC shown in this Section. Particularly, because of the simplicity of the reverse carpooling structure, we find that it has been studied more often than the *k*-tuple structure. In the following, three XNC schemes [40-42] are surveyed to show how the scheduling factor and flow route selection factor can be considered to increase coding opportunities based on the reverse carpooling structure.

The authors in [40] consider the minimal cost to deliver a certain amount of traffic of each flow. This minimization problem is formulated as a linear program in [40], where the routing, scheduling, and the reverse carpooling based XNC are jointly considered. The basic idea behind the formulation is that the traffic of each flow is allowed to be split and distributed onto multiple paths, which then determines the amount of coding opportunities at each intermediate node. Given a packet delivery probability for each link, the total required number of packet transmissions (i.e., the cost) can then be computed by summing up the required number of transmission at each node. Note that an encoding node in the reverse carpooling structure only needs T_{nc} rather than $T_{routing}$ transmission time to deliver two packets from two flows. Through the performance evaluation in [40], it was shown that the shortest path routing with network coding can save the cost as compared to that without network coding, and that network coding-aware routing can further reduce the cost.

The authors in [41] continue with the work in [40] by addressing the problem of the greediness for coding opportunities. In maximizing coding opportunities, [40] ignores the constraints of link capacity and the interference around each link. Hence, [41] adds these constraints into the linear program for a more reasonable modeling and formulation. Here, we omit the details of the formulation in [41].

Other distributed reverse carpooling based XNC schemes, e.g., [42], can also be found in the literature, but we omit details of them because most of them deal with the same design factors as those schemes we surveyed under reliable links.

XNC for the Second Coding Structure Type**:** We find that the uncertainty of the coding opportunity under lossy link condition in the second coding structure type makes those aforementioned factors difficult to be applied. For example, the work in [20] states that using the equal channel access in the two-hop coding structure shown in Fig. 4(c), plenty of coding opportunities will arise at node C and the highest network throughput could be attained under reliable links. However, this may not be true if the overhearing link from node A to node E (or from node B to node D) has low delivery probability, because many packets queued in the buffer of node C cannot be encoded in this situation. If node C is forced to encode the queued packets, the achieved performance will be severely hurt or degraded compared to traditional routing [43]. We find in the literature that an encoding node in this second type normally chooses to perform XNC only when it knows the next hop of each flow has already overheard or obtained the packet for decoding [7, 8, 44], and that XNC is widely used in combination with INC under unreliable links [7, 8], which is out of the scope of this survey.

C. XNC with Rate Adaptation

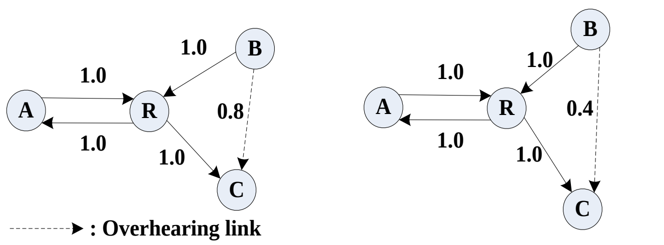
Now, we present a new factor, i.e., the transmission rate, which can affect the performance of XNC schemes for both coding structure types classified above. The reason to consider this factor is that the packet delivery probability of each link can be improved by reducing the transmission rate [45]. We can see that T_{nc} in the first coding structure type varies with different packet delivery probabilities, and that the amount of coding opportunity in the second structure type varies with different reliabilities of the overhearing links. Since both T_{nc} and the amount of coding opportunity are closely related to network performance, the transmission rate can affect the performance of an XNC scheme for each of the two coding structure types.

***XNC for the First Coding Structure Type*:** The authors in [46] consider the optimal transmission rate for the encoding node under the reverse carpooling structure. For node 2 in Fig. 1 to deliver P_1 and P_2 to nodes 1 and 3, respectively, node 2 adopts a variation of the transmission scheme shown in the typical example in this Section [46]. First, node 2 repeatedly broadcasts $P_1 \oplus P_2$ at a rate of R_{nc} until the packet is received by at least one receiver (i.e., node 1 or node 3). Next, if node 1 (node 3) does not received its desired packet, the desired packet will be transmitted by node 2 at an optimal rate R_1 (R_3). Note that the rate R_1 (R_3) is the rate at which node 2 can transmit a unit of data to node 1 (node 3) in the shortest time. To determine an optimal R_{nc}, the authors point out by simulation that in most cases the two packets, P_1 and P_2, are delivered to their destinations in the shortest time by setting $R_{nc} = min(R_1, R_2)$. However, they fail to propose a solution to determine an optimal R_{nc}.

***XNC for the Second Coding Structure Type*:** The authors in [47] consider the rate adaptation *only* for the encoding node in the two-hop coding structure of N ($N \geq 2$) flows to send a stack of stored packets. This is actually the case very similar to [46] above, because what packets in the stack can be encoded has already been determined given the received packets at the encoding node. One critical assumption in [47] is that the encoding node is allowed to transmit an encoded or non-encoded packet only *once* rather than *multiple times* (i.e., T_{nc} shown in this Section) to ensure the packet reception at each intended receiver. To ensure the packet reception at a set of M ($M \leq N$) nodes, the encoding node needs to choose a transmission rate R such that $R = \min(R_1, \dots, R_M)$, where R_i ($1 \leq i \leq M$) is the maximal rate at which the encoding node can reliably deliver a packet to the ith receiver. Thus, encoding more packets in one transmission generally reduces the transmission rate, and the encoding node, before sending a packet, is instructed to properly choose a set of M receivers such that the link capacity to any receiver (which is equal to $M \times R$) is maximized [47].

In addition, the authors in [48] consider the same setting as [47], but assume a deadline for each packet in the stack. The encoding node is then instructed to schedule the transmission of these packets possibly by XNC to minimize the number of packets missing their deadlines. Obviously, encoding too many packets in one transmission normally requires a lower transmission rate and a longer transmission time, which may cause more packets to miss their deadlines in the next transmission. The authors propose a heuristic algorithm to determine what packets to encode in each transmission by balancing the two packet numbers above.

Furthermore, the authors in [49] employ the rate adaptation to address the unreliability of the overhearing link in the second coding structure type. One assumption in [49] is that for one node to transmit a packet to another node, if the transmission rate is higher (no larger) than the link capacity, the receiver will erroneously (correctly) receive the packet. For simplicity, we only present two examples in [49] here, as shown in Fig. 13(a) and (b), to illustrate how the rate adaptation can be combined with XNC to affect network performance. In both figures, the number associated with each link denotes the maximal capacity of that link, and nodes A and B intend to send 1 bit to nodes C and A, respectively. In Fig. 13(a), if node B chooses the rate of 1, then, node C cannot overhear node B and there is no coding opportunity at node R. As a result, the throughput, defined as the ratio of the number of bits delivered in total over the needed transmission time, is $2/4 = 0.5$. If node B chooses the rate of 0.8, then, node R can encode the two bits from nodes A and B, which results in a larger throughput of $\frac{2}{2+1.25} = 8/13$. However, the situation will turn around in the example of Fig. 13(b). Obviously, without network coding at node R in Fig. 13(b), the throughput is still 0.5. In contrast, when network coding is enabled at node R, we get a throughput of $4/9$, which is less than 0.5. Hence, from these two examples, we can see that lowering the transmission rate to increase the reliability of the overhearing links and the coding opportunities only helps improve network performance in some situations. The rate adaptation can also be applied to a more complicated two-hop relay-based scenario to increase coding opportunities, of which details can be found in [49].



(a) Improved performance with XNC (b) Degraded performance with XNC

Fig. 13: Examples of XNC with rate adaptation.

For the two-hop coding structures (e.g., those in Fig. 4(b), (c), and (d)) where overhearing links are involved, the authors in [45] propose a framework for *each node* in the coding structures to choose an optimal rate among a set of available rates. It is assumed in [45] that the packet delivery probability of each link, including the overhearing link, continuously varies with the transmission rate. Nodes in this framework choose the transmission rate as follows. First, *each previous-hop node* of the encoding node is only allowed to choose the rate such that the packet delivery probability of each overhearing link is higher than a certain threshold. Note that this threshold is used to ensure the reliability of the overhearing link or ensure XNC to be performed at the encoding node. Second, the *encoding node* will choose a rate such that the expected network throughput is maximized. For more details, readers are referred to [45]. Simulation results in [45] indicate that XNC with rate selection scheme for each node usually leads to higher network performance than that with a fixed rate for each node.

Remark on XNC with rate adaptation: Although the works surveyed here target those simple coding structures, mainly two-hop coding structures, we can also apply the rate adaptation to a more general multi-hop network by viewing the multi-hop network as a combination of many two-hop coding structures and applying the technique to each formed two-hop coding structure. The work in [45] has shown the applicability of the rate adaptation to a multi-hop network.

4. Drawbacks of Applying XNC

Although XNC benefits network performance in most situations, in some scenarios applying XNC could be detrimental to network performance, even with no greediness for coding opportunities. The following presents two major drawbacks of applying XNC in WMNs.

Spatial reuse reduction in XNC: XNC exploits the broadcast channel in WMNs to improve network performance, but the coded packet broadcast could reduce the spatial reuse in WMNs. Specifically, a packet broadcast in XNC could restrict other transmission for the following two reasons.

Reason 1: higher transmission power in XNC. Due to signal attenuation, a higher power threshold is generally required for a sender to transmit a packet to a receiver which is further away. Thus, when an encoding node in XNC broadcasts a packet to a set of receivers, it needs to select a higher power level to ensure that the farthest receiver can correctly receive the broadcast packet [50]. As a result, the use of a larger transmission power will bring stronger interference to other transmissions, which may harm network performance. One illustrative example used in [50] is shown in Fig. 14(a), where node B is an encoding node for f_1 and f_2. It is assumed in [50] that the transmission power threshold from node B to node A is larger than that from node B to node C, i.e., $P_{BA} > P_{BC}$, and that the use of P_{BA} at node B will impose strong interference on the transmission from node G to node F. Thus, when node B is using P_{BA} to broadcast a coded packet, node G needs to refrain from transmitting. Other conflicting transmission pairs are connected in the figure by the dashed lines. Suppose one packet transmission takes one time slot, then, the optimal scheduling for XNC and the non-coding scheme are constructed and shown in Fig. 14(b) and (c), respectively. Obviously, we see that the non-coding scheme here can deliver the same amount of traffic within a shorter period.

Although the authors in [50] point out in one particular scenario the disadvantage of applying XNC shown above, they fail to specify other general situations in which performing XNC at an encoding node is indeed detrimental to network performance.

Reason 2: vulnerability to other transmissions in XNC. In XNC, a coded packet is intended to multiple receivers, and thus its broadcast is more vulnerable to other simultaneous transmissions due to interference. In Fig. 14, for example, in non-coding scheme the transmissions from node B to node A and from node H to node I can be scheduled simultaneously, whereas in XNC, node H as a neighbor of node C is not allowed to transmit when node B is broadcasting coded packet to nodes A and C. This means that the gain of XNC is achieved at the cost of preventing other transmissions. A detailed example showing how the non-coding scheme outperforms XNC can be found in [58]. We omit the details here.

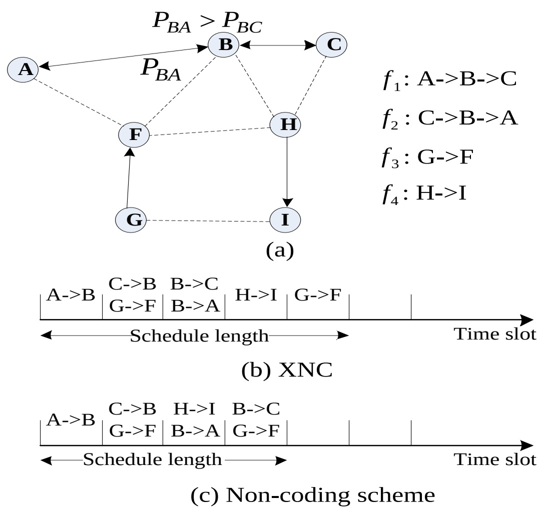


Fig. 14: The negative impact of the higher transmission power required in XNC.

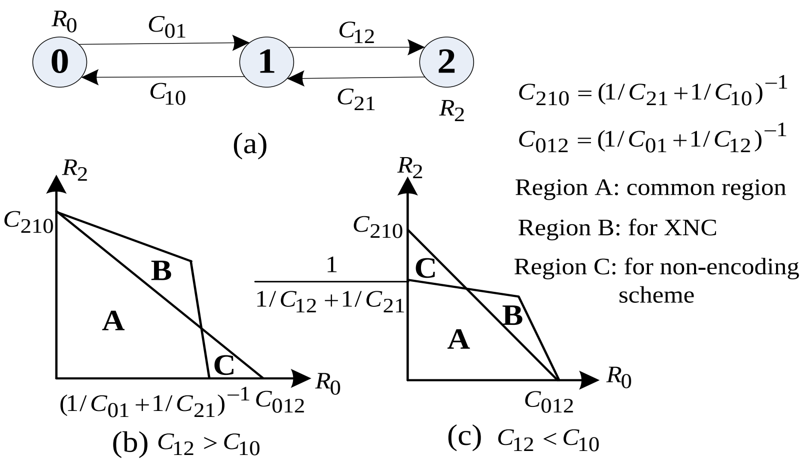


Fig. 15: The negative impact of the lower link rate selection in XNC.

Lower link rate selection in XNC: Suppose that an encoding node needs to broadcast one encoded packet once to two or more receivers, and that the links to the receivers have different maximal link rates. Then, for correct reception at all receivers, the encoding node needs to transmit at the lowest link rate among these rates [51, 52]. This requirement actually has been mentioned when we introduce the works [47, 48] above. Using a lower rate to broadcast the packet is very likely to harm network performance. Here, we employ the work in [51] for a more complete understanding of this drawback. In Fig. 15(a), each link is labeled with a maximal capacity, and we are interested in the regions of (R_0, R_2) under XNC and the non-encoding scheme, where R_0 (R_2) is the bit rate for the traffic from node 0 (node 2) to node 2 (node 0). This rate region problem can be formulated as a simple linear program [51], of which the details are omitted here. Fig. 15(b) and (c) show the regions of (R_0, R_2) under the two schemes when $C_{12} \neq C_{10}$. From the figures, we can find that when $C_{12} > C_{10}$, it is not possible for XNC to support a (R_0, R_2) tuple with $R_0 > 1/(1/C_{01} + 1/C_{21})$, and that when $C_{12} < C_{10}$, it is not possible for XNC to support a (R_0, R_2) tuple with $R_2 > 1/(1/C_{12} + 1/C_{21})$. Hence, blindly applying XNC to a network where links differ from each other in the capacity is risky, and we may require some strategies as proposed in [47, 48] to properly encode the packets to avoid the low link rate in XNC.

5. Conclusion and Future Work

In this paper, we have surveyed various inter-flow network coding (XNC) schemes in wireless mesh networks (WMNs) with reliable links and with lossy links. For the survey in each of the two scenarios, we first present a typical XNC example and expose the rationale behind it. Then, we present and illustrate some relevant factors which can be considered for more benefits from XNC. Moreover, some drawbacks, e.g., the lower broadcast rate at the encoding node, of applying XNC are explained in this paper. With the exposed rationales, the enumerated relevant factors, and the listed drawbacks, we believe that readers can get a more thorough understanding of applying XNC in WMNs (that is, for a better XNC design, all the surveyed factors and drawbacks need to be considered and properly addressed).

Based on the survey in this paper, the potential future works on XNC in WMNs could be as follows.

- *Relaxation of XNC conditions*: It should be meaningful to investigate how the conditions of XNC under reliable links can be further relaxed. The work in this direction could be built based on the example shown in Fig. 12 in this paper to seek for more coding opportunities.
- *Decision on coded packet retransmission under lossy links*: As introduced earlier in [47, 48], an encoding node needs to choose a lower transmission rate to broadcast coded packets only once to ensure packet reception at all receivers. However, it will be interesting to see if a shorter transmission time can be achieved with coded packet retransmissions at a higher transmission rate. Particularly, the retransmission scheme can be the one in [46].
- *Random linear combination (RLC) for XNC under lossy links*: We find that RLC could further improve XNC under lossy links. Refer to XNC in Fig. 1 under lossy links, and now consider node 2 has to send packets P_1^1 and P_1^2 (P_2^1 and P_2^2) from f_1 (f_2). Using XOR, node 2 will keep broadcasting $P_1^1 \oplus P_2^1$ and $P_1^2 \oplus P_2^2$ until both are received by nodes 1 and 3. By contrast, using RLC, node 2 can mix the four packets into many linearly independent packets, $P_c^i = c_1^i P_1^1 + c_2^i P_1^2 + c_3^i P_2^1 + c_4^i P_2^2$, $i = 1,2 \cdots$, where c_1^i, c_2^i, c_3^i, and c_4^i are random coefficients in some large-size fields [3], and then keep broadcasting P_c^i until nodes 1 and 3 respectively receive any two P_c^i to decode their desired packets. XNC with RLC will outperform that with XOR, because with RLC, any coded

packet received by node 1 or 3 is useful for decoding, whereas with XOR, node 1 or 3 may receive a coded packet repeatedly, e.g., until node 1 receives $P_1^1 \oplus P_2^1$, node 3 may have received two or more $P_1^1 \oplus P_2^1$. Hence, with RLC, XNC may opt for more packets of different flows accumulated at the encoding node, which then calls for new scheduling schemes.

- *Rate adaptation for the generalized coding structure*: Although the rate adaptation is widely studied in the two-hop coding structures [45-49], it should be also studied in the generalized coding structure. This is because any upstream node (e.g., node 1 in Fig. 5(b)) of the encoding node on one flow in the generalized coding structure can be a candidate to adapt its transmission rate for coding opportunities, and there may exist the best one for performance improvement. By contrast, only the previous hop (e.g., node A in Fig. 4(c)) of the encoding node on one flow can be chosen in the two-hop coding structures.
- *Design of XNC schemes with awareness of the transmission power*: Based on the first shown drawback of applying XNC, i.e., the higher transmission power required at the encoding node, it should be interesting to design a distributed mechanism to let the encoding node detect when the packet encoding should be or should not be performed.
- *Mobility issues in XNC:* In WMNs, the mesh routers are assumed with minimal mobility, but the mesh clients could be mobile [55]. The node mobility can be both a challenge and a feature to exploit for the XNC design. As a challenge, node mobility could disturb a formed coding structure, e.g., if node 1 in Fig. 4(c) moves out of the overhearing range of node 4, then no coding opportunity exists at node C, and furthermore, the path used by f_2 may be no longer the best. This begets the question of whether the existing flows

like f_2 should be rerouted for better paths [37]. As an advantage, node mobility could also be exploited for XNC. For example, if node C in Fig. 13(b) moves toward node B, then XNC will outperform the non-coding scheme. How to properly deal with or even control node mobility in XNC for optimal performance can be a focus for future research.

- *Security issues in XNC:* XNC is vulnerable to a variety of attacks, ranging from packet pollution to packet drop [63]. Driven by the benefit of XNC, a new incoming flow usually opts for a path with more coding opportunities. A malicious node could misreport the coding opportunities on it, e.g., through misreporting the neighboring nodes in the generalized coding structure [17], to lure the new flow to choose a path via it, thus increasing the chance of an attack. Hence, to design secure XNC schemes that incorporate effective defense mechanisms against attacks is critical. This will be a focus for future research, despite some existing works in the literature on secure XNC design, e.g., [64].

6. Acknowledgement

This research is supported by the research grant DSOCL04020 from the Defence Research and Technology Office, Singapore.

References

[1] B. Lee, D. Park, and H. Seo, *Wireless Communications Resource Management*, John Wiley & Sons, 2009.

[2] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network Information Flow," *IEEE Trans. Info. Theory*, vol. 46, 2000, pp. 1204-16.

[3] C. Fragouli, J. L. Boudec, and J. Widmer, "Network Coding: an Instant Primer, " in *Proc. ACM SIGCOMM'06*, 2006, pp. 63-68.

[4] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, "Wireless Network Coding: Opportunities & Challenges," in *Proc. IEEE MILCOM'07*, 2007, pp. 1-8.

[5] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," in *Proc. ACM SIGCOMM’07*, 2007, pp. 169-80.

[6] D. Koutsonikolas, C. C. Wang, and Y. C. Hu, "CCACK: Efficient Network Coding Based Opportunistic Routing Through Cumulative Coded Acknowledgments," in *Proc. IEEE INFOCOM'10*, 2010, pp. 1-9.
[7] C. Qin, Y. Xian, C. Gray, N. Santhapuri, and S. Nelakuditi, "I2MIX: Integration of Intra-Flow and Inter-Flow Wireless Network Coding," in *Proc. IEEE SECON'08*, 2008, pp. 1-6.
[8] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "CORE: A Coding-Aware Opportunistic Routing Mechanism for Wireless Mesh Networks," *IEEE Wireless Commun.*, vol. 17, 2010, pp. 96-103.
[9] A. Eryilmaz and D. S. Lun, "Control for Inter-Session Network Coding," in *Proc. NetCod'07*, 2007.
[10] M. A. Iqbal, B. Dai, B. Huang, A. Hassan, and S. Yu, "Survey of Network Coding-Aware Routing Protocols in Wireless Networks," *J. Network and Computer Applications*, vol. 34, no. 6, 2011, pp. 1956-70.
[11] T. Matsuda, T. Noguchi, and T. Takine, "Survey of Network Coding and Its Applications," *IEICE Trans. Commun.*, vol. E94-B, no. 3, 2011, pp. 698-717.
[12] R. Bassoli, H. Marques, J. Rodriguez, K. W. Shum, and R. Tafazolli, "Network Coding Theory: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, 2013, pp. 1950-78.
[13] B. Li and D. Niu, "Random Network Coding in Peer-to-Peer Networks: from Theory to Practice," *Proceedings of the IEEE*, vol. 99, no. 3, 2011, pp. 513-23.
[14] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the Air: Practical Wireless Nework Coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, 2008, pp. 497-510.
[15] A. P. Gheibi, A. Ozdaglar, M. Effros, and M. Medard, "Optimal Reverse Carpooling Over Wireless Networks - A Distributed Optimization Approach," in *Proc. CISS'10*, 2010.
[16] S. Omiwade, R. Zheng, and C. Hua, "Practical Localized Network Coding in Wireless Mesh Networks," in *Proc. IEEE SECON'08*, 2008, pp. 332-40.
[17] J. Le, C. S. Lui, and D. Chiu, "DCAR: Distributed Coding-Aware Routing in Wireless Networks," *IEEE Trans. Mobile Comput.*, vol. 9, 2010, pp. 596-608.
[18] Z. Zhou and L. Zhou, "Network Joint Coding-Aware Routing for Wireless Ad Hoc Networks," in *Proc. Int'l. Conf. Wireless Communications, Networking and Information Security*, 2010, pp. 17-21.
[19] N. M. Jones, B. Shrader, and E. Modiano, "Optimal Routing and Scheduling for a Simple Network Coding Scheme," in *Proc. IEEE INFOCOM'12*, 2012, pp. 352-60.
[20] J. Le, J. C. S. Lui, and D. M. Chiu, "On the Performance Bounds of Practical Wireless Network Coding," *IEEE Trans. Mobile Comput.*, vol. 9, no. 8, 2010, pp. 1134-46.
[21] S. Sengupta, S. Rayanchu, and S. Banerjee, "Network Coding-Aware Routing in Wreless Networks," *IEEE/ACM Trans. Netw.*, vol. 18, 2010, pp. 1158-70.
[22] L. Huang and C. W. Sung, "An Iterative Routing Algorithm for Energy Minimization in Coded Wireless Networks," in *Proc. IEEE PIMRC'11*, 2011, pp. 1124-28.
[23] S. Shabdanov, C. Rosenberg, and P. Mitran, "Joint Routing, Scheduling, and Network Coding for Wireless Multihop Networks," in *Proc. IEEE WiOpt'11*, 2011, pp. 33-40.

[24] P. S. Mogre, N. Heureuse, M. Hollick, and R. Steinmetz, "A Case for Joint Routing, Scheduling, and Network Coding in TDMA-based Wireless Mesh Networks: A Cross-layer Approach," in *Proc. IEEE MASS'07*, 2007, pp. 658-60.
[25] B. Scheuermann, W. Hu, and J. Crowcroft, "Near-Optimal Co-ordinated Coding in Wireless Multihop Networks," in *Proc. ACM CoNEXT'07*, 2007.
[26] D. Koutsonikolas, Y. C. Hu, and C. Wang, "An Empirical Study of Performance Benefits of Network Coding in Multihop Wireless Networks," in *Proc. IEEE INFOCOM'09*, 2009, pp. 2981-85.
[27] D. Umehara, T. Hirano, S. Denno, M. Morikura, and T. Sugiyama, "Wireless Network Coding in Slotted ALOHA with Two-Hop Unbalanced Traffic," *IEEE JSAC*, vol. 27, no. 5, 2009, pp. 647-61.
[28] D. Umehara, S. Denno, M. Morikura, and T. Sugiyama, "Throughput Analysis of Two-Hop Wireless CSMA Network Coding," in *Proc. IEEE ICC'10*, 2010, pp. 1-6.
[29] F. Zhao and M. Medard, "On Analyzing and Improving COPE Performance," in *Proc. IEEE Workshop on Information Theory and Applications*, 2010, pp. 317-22.
[30] H. Guo, Y. Qian, K. Lu, and N. Moayeri, "Backbone Routing over Multihop Wireless Networks: Increased Network Coding Opportunity," in *Proc. IEEE ICC'10*, 2010, pp. 1-5.
[31] S. Wang, G. Tan, Y. Liu, H. Jiang, and T. He, "Coding Opportunity Aware Backbone Metrics for Broadcast in Wireless Networks," in *Proc. IEEE INFOCOM'13*, 2013, pp. 275-79.
[32] X. Jiao, X. Wang, and X. Zhou, "Active Network Coding based High-Throughput Optimizing Routing for Wireless Ad Hoc Networks," in *Proc. IEEE WiCOM'08*, 2008, pp. 1-5.
[33] D. B. Johnson, D. A. Maltz, and Y. C. Hu, "The Dynamic Source Routing for Mobile Ad Hoc Networks," *Internet-Draft*, 2004.
[34] Y. Wu, S. M. Das, and R. Chandra, "Routing with a Markovian Metric to Promote Local Mixing," in *Proc. INFOCOM'07*, 2007, pp. 2381-85.
[35] 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. IEEE GLOBECOM'08*, 2008, pp. 1-5.
[36] M. Jhang, S. Lin, and W. Liao, "C^2AR: Coding and Capacity Aware Routing for Wireless Ad Hoc Networks," in *Proc. IEEE ICC'10*, 2010, pp. 1-5.
[37] 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 Trans. Veh. Technol.*, vol. 60, no. 4, 2011. pp. 1714-27.
[38] N. Gaddam, S. A. Gathala, D. Lastine, and A. Somani, "Energy Minimization through Network Coding for Lifetime Constrained Wireless Networks," in *Proc. ICCCN'08*, 2008, pp. 1-6.
[39] J. Price and T. Javidi, "Network Coding Games with Unicast Flows," *IEEE JSAC*, vol. 26, no. 7, 2008, pp. 1302-16.
[40] B. Ni, N. Santhapuri, Z. Zhong, and S. Nelakuditi, "Routing with Opportunistically Coded Exchanges in Wireless Mesh Networks," in *Proc. IEEE WiMesh'06*, 2006, pp. 157-59.
[41] Y. Benfattoum, S. Martin, and K. A. Agha, "IROCX: Interference-Aware Routing with Opportunistically Coded Exchanges in Wireless Mesh Networks," in *Proc. IEEE WCNC'11*, 2011, pp. 1113-18.

[42] Y. Peng, Y. Yang, X. Lu, and X. Ding, "Coding-Aware Routing for Unicast Sessions in Multi-hop Wireless Networks," in *Proc. IEEE GLOBECOM'10*, 2010, pp. 1-5.
[43] A. O. F. Atya, I. Broustis, S. Singh, D. Syrivelis, S. V. Krishnamurthy, and T. Porta, "Wireless Network Coding: Deciding When to Flip the Switch," in *Proc. INFOCOM'13*, 2013, pp. 260-64.
[44] A. Khreishah, I. M. Khalil, P. Ostovari, and J. Wu, "Flow-based XOR Network Coding for Lossy Wireless Networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, 2012, pp. 2321-29.
[45] T. Kim, I. Broustis, S. Vural, D. Syrivelis, S. Singh, S. V. Krishnamurthy, and T. Porta, "Realizing the Benefits of Wireless Network Coding in Multirate Settings," *IEEE/ACM Trans. Netw.*, vol. 21, no. 3, 2013, pp. 950-62.
[46] B. Ni, N. Santhapuri, C. Gray, and S. Nelakuditi, "Selection of Bit-rate for Wireless Network Coding," in *Proc. SECON'08*, 2008, pp. 1-6.
[47] H. Yomo and P. Popovski, "Opportunistic Scheduling for Wireless Network Coding," in *Proc. IEEE ICC'07*, 2007, pp. 5610-15.
[48] X. Wang, C. Yuen, and Y. Xu, "Joint Rate Selection and Wireless Network Coding for Time Critical Applications," in *Proc. IEEE WCNC'12*, 2012, pp. 1964-69.
[49] Y. Kim and G. Veciana, "Is Rate Adaptation Beneficial for Inter-session Network Coding?," *IEEE JSAC*, vol. 27, no. 5, 2009, pp. 635-46.
[50] J. El-Najjar, H. AlAzemi, and C. Assi, "On the Interplay Between Spatial Reuse and Network Coding in Wireless Networks," *IEEE Trans. Wireless Commun.*, 2011, pp. 560-69.
[51] C. Liu and F. Xue, "Network Coding for Two-Way Relaying: Rate Region, Sum Rate and Opportunistic Scheduling," in *Proc. IEEE ICC'08*, 2008, pp. 1960-65.
[52] J. Zhang and Q. Zhang, "Cooperative Network Coding-Aware Routing for Multi-Rate Wireless Networks," in *Proc. IEEE INFOCOM'09*, 2009, pp. 181-89.
[53] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan, "I^2NC: Intra- and Inter-Session Network Coding for Unicast Flows in Wireless Networks," in *Proc. IEEE INFOCOM'11*, 2011, pp. 1035-1043.
[54] A. Khreishah, I. Khalil, and J. Wu, "Low Complexity and Provably Efficient Algorithm for Joint Inter and Intrasession Network Coding in Wireless Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 10, 2013, pp. 2015-2024.
[55] I. Akyildiz, X. Wang, and W. Wang, "Wireless Mesh Networks: A Survey," *Computer Networks*, vol. 47, 2005, pp. 445-487.
[56] A. Khreishah, C. Wang, and N. Shroff, "Cross-Layer Optimization for Wireless Multihop Networks with Pairwise Intersession Network Coding," *IEEE JSAC*, vol. 27, no. 5, 2009, pp. 606-621.
[57] C. Wang, A. Khreishah, and N. Shroff, "Cross-Layer Optimizations for Intersession Network Coding on Practical 2-Hop Relay Networks," in *Proc. ACSSC'09*, 2009, pp. 771-775.
[58] P. Chaporkar, and A. Proutiere, "Adaptive Network Coding and Scheduling for Maximizing Throughput in Wireless Networks," in *Proc. ACM MOBICOM'07*, 2007, pp. 135-146.
[59] T. Cui, L. Chen, and T. Ho, "Energy Efficient Opportunistic Network Coding for Unicast Flows in Wireless Networks," in *Proc. IEEE INFOCOM'08*, 2008.

[60] H. Seferoglu, and A. Markopoulou, "Network Coding-Aware Queue Management for Unicast Flows over Coded Wireless Networks," in *Proc. IEEE NETCOD'10*, 2010, pp. 1-6.
[61] Y. Huang, M. Ghaderi, D. Towsley, and W. Gong, "TCP Performance in Coded Wireless Mesh Networks," in *Proc. IEEE SECON'08*, 2008, pp. 179-187.
[62] P. Ostovari, J. Wu, and A. Khreishah, "Deadline-Aware Broadcasting in Wireless Networks with Local Network Coding," in *Proc. IEEE ICNC'12*, 2012, pp. 1123-1127.
[63] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Secure Network Coding for Wireless Mesh Networks: Threats, Challenges, and Directions," *Computer Communications*, vol. 32, no. 17, 2009, pp. 1790-1801.
[64] J. Dong, R. Curtmola, C. Nita-Rotaru, and D. Yau, "Pollution Attacks and Defenses in Wireless Interflow Network Coding Systems," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 5, 2012, pp. 741-755.