Chapter 1
Network Coding Techniques for Wireless and Sensor Networks

[1]Pouya Ostovari,[1]Jie Wu, and[2]Abdallah Khreishah

Abstract

Network coding is a technique where relay nodes mix packets using mathematical operations, which reduces the number of transmitted packets. Network coding was first proposed for wired networks to solve the bottleneck problem and to increase the throughput. However, the broadcast nature of wireless networks and the diversity of the links make network coding more attractive in wireless networks. Network coding can be classified as either inter or intra-session. Inter-session network coding allows the packets from different sessions (sources) to be mixed to solve the bottleneck problem. In contrast, intra-session network coding, which can be used to address the packet loss problem, uses the diversity of the wireless links and mixes packets from the same sessions. In this chapter, we survey the recent works on network coding in both general wireless networks and wireless sensor networks. We present various network coding techniques, their assumptions, applications, as well as an overview of the proposed methods.

Key words: Network coding, wireless networks, wireless sensor networks, inter-session, intra-session, broadcast, multicast, unicast.

1.1 Introduction

As the demand for communication services is growing, wireless solutions becomes more and more important. Due to their ease of deployment, wireless networks play a major role in our lives. They are also ideal to provide a convenient solution to the last mile problem [1][2]. Wireless networks can be cellular networks that are used for mobile phones, or Wi-Fi networks that provide an Internet connection. Different multihop wireless network settings are used. Mesh networks can be used to provide Internet access and file sharing [3]. Wireless Sensor Networks [4] (WSN) can be

[1]Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122 ·
[2]Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102

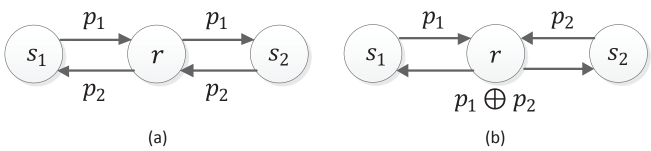

Fig. 1.1 Binary network coding.

used for military applications, such as enemy detection in battlefields. They can also be used for disaster detection and monitoring applications.

Despite the diverse types of wireless networks and their applications, the common features of wireless networks create opportunities to be exploited and challenges to be addressed. These common features include the broadcast nature of wireless links, the interference among the links, the diversity of the links, and the lossy behavior of the links [5]. Also, the correlation between the links may affect the performance of the wireless communication protocols.

Inter-Session Network Coding. The broadcast nature of wireless networks is considered a challenge, as it creates interference between the links and produces unnecessary multiple copies of the same packet. However, if we allow the intermediate wireless nodes to code the packets, the broadcast nature becomes an opportunity. Consider the example in Figure 1.1(a), where nodes s_1 and s_2 want to exchange their own packets, p_1 and p_2, respectively. Assuming that these nodes are out of range of each other, this communication incurs four transmissions; two transmissions for sending the packets to the relay node, and two transmissions for relaying the packets. However, the relay node can simply *XOR* the packets and send the coded packet $p_1 \oplus p_2$ [6], which is shown in Figure 1.1(b). The nodes s_1 and s_2 can retrieve each others' packets by XOR-ing $p_1 \oplus p_2$ with their own packets, p_1 and p_2, respectively. As a result, the number of transmissions has been reduced to three by using binary network coding. *Inter-session* network coding solves the bottleneck problem and reduces the number of transmissions, by allowing packets from different sessions (sources) to be coded together. By reducing the number of required transmissions, network coding increases the throughput and decreases the interference between the links in wireless networks.

Intra-Session Network Coding. Another important application of network coding is to provide reliability in wireless networks. The traditional way to provide reliability for both wired and wireless networks is to use feedback messages to report the received (or lost) packets. By using feedback messages, the sender node will know which packets need to be sent again. However, these feedback messages consume bandwidth. Consider the example in Figure 1.2; the source node wants to deliver packets p_1 and p_2 to the node d. The reliability of the link $s_1 \rightarrow d$ is equal to $\frac{2}{3}$. In the case that the source node sends three coded packets, $p_1 + p_2$, $p_1 + 2p_2$, and $2p_1 + p_2$, on average, the destination node will receive two of the three coded packets. Therefore, the destination nodes will be able to retrieve the packets p_1 and

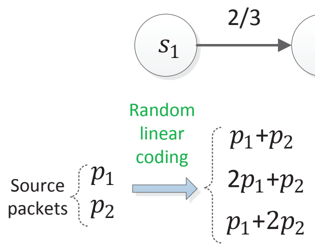

Fig. 1.2 Application of network coding to provide reliability.

p_2. However, without network coding, we need to use a feedback mechanism or else the source node needs to transmit each packet twice. As a result, communication schemes with network coding can provide reliability with a fewer transmissions than schemes without network coding.

Coding the packets from the same session (source) is called *intra-session* network coding, which exploits the diversity of the links. In intra-session network coding, the packets from the same source are coded together (usually linearly), which makes the importance of the packets the same. Therefore, when k packets are coded together, a relay node does not need to know exactly which packets are received by the destination node; it is thereby enough to successfully deliver k coded packets out of the transmitted coded packets.

Opportunistic Routing. An efficient way to address packet loss in wireless networks without network coding is to use *opportunistic routing* approaches [7]. When a node broadcasts a packet, it is probable that the next-hop does not receive the packet. However, because of the broadcast nature of the wireless medium, and the diversity among the links, a neighbor of the sender can receive and forward the packet as the next-hop with high probability. In opportunistic routing, there is no specific path from the source to the destination, and any node that overhears the packet can relay it. Take Figure 1.3 as an example, in which node s wants to send 4 packets to the destination d. The delivery rate of the links are shown beside the links. Assume that each relay node received the packets shown beside the nodes. If we use traditional shortest path routing, the route from s to d will be fixed. Assuming that the chosen route is $s \rightarrow r_1 \rightarrow d$, the source node needs to retransmit the packets p_3 and p_4. On the other hand, if we allow the other nodes that received the packets p_3 and p_4 to forward them, the source node will not need to retransmit any packet.

The main challenge in opportunistic routing is coordinating the intermediate nodes. To prevent redundant transmissions, the intermediate nodes need to send feedback or listen to the other nodes' transmissions to find out if there is a neighbor that has received the transmitted packet. For this purpose, the intermediate nodes need to be able to overhear each other, which might not be possible, as shown in Figure 1.3. Network coding can solve this problem [8]. To this purpose, the source

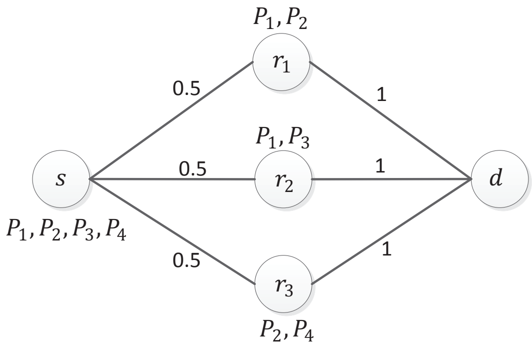

Fig. 1.3 Opportunistic routing.

node divides the packets to be sent in batches of k packets. The source keeps sending coded packets of the form $\sum_{i=1}^{k} \alpha_i p_i$, where α_i is a random coefficient chosen over a finite field. When an intermediate node receives a coded packet, the node checks if the coded packet is linearly independent to the previously received packets. If so, the node will add the packet to its buffer. Each intermediate node generates linear combinations of the packets in its buffer and sends the coded packets. The destination node can decode all of the packets of the batch when it receives k linearly independent packets. In this case, the destination node sends feedback to the source to stop sending the packets.

Cross-Layer Design. Using network coding methods in wireless protocols incurs new challenges. For example, previous routing protocols are unaware of network coding. However, the routing protocol affects the coding opportunity. If two flows pass through relay nodes that are far from each other, there will be no coding opportunity. On the other hand, flows that are close to each other result in more interference. Therefore, to increase the efficiency of the proposed protocols for wireless networks, cross-layer approaches are needed. In cross-layer approaches, the protocols of different layers are independent. However, they communicate with each other to make decisions and perform more efficiently.

Wireless Sensor Networks. Sensor networks differ from the general wireless networks in performance metrics, traffic patterns, and their amount of available memory and processing resources [9]. These differences make some of the network coding approaches proposed for general wireless networks inappropriate for WSNs. For example, in some of the network coding methods, the nodes should listen to their neighbors and store the overheard messages in their buffers. However, in sensor networks, because of the memory limitation, sensor nodes cannot cache overheard packets that might not be useful [10]. WSNs' protocols must be simple and easily implemented. Moreover, the links' quality between the sensor nodes vary over the time, and nodes can fail or disconnect. Therefore, the dynamic environment

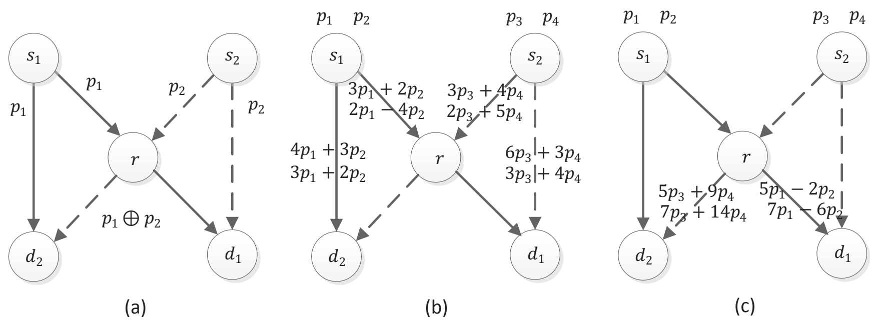

Fig. 1.4 XOR and random linear coding.

should be considered, and the algorithms should be adaptive to reflect this dynamic nature [10].

The rest of this chapter is organized as follows. We provide our classification methodology in Section 1.2. In Section 1.3, we describe some of the well-known proposed methods for unicast application, and we categorize them. A discussion about multicast and broadcast network coding approaches is provided in Sections 1.4 and 1.5, respectively. Section 1.6 concludes the chapter. Note that fountain codes (also known as rateless erasure codes), such as online codes [11], LT codes [12], and raptor codes [13], are beyond the scope of this chapter.

1.2 Classification of Network Coding Approaches

From one perspective, network coding can be classified into *XOR* (binary) coding and *Random Linear* (RL) coding. In binary coding, XOR operations are performed between the packets. Take Figure 1.4(a) as an example; we have two flows: one of them between nodes s_1 and d_1 and the other between nodes s_2 and d_2. Without network coding, the relay node needs two transmissions to send the packets, one for each flow. However, the relay node r can exploit the broadcast nature of its output links and reduce the number of transmissions to one by XORing the two packets. The nodes d_1 and d_2 decode the coded packet by XOR-ing $p_1 \oplus p_2$ with the overheard packets, p_2 and p_1, respectively.

In random linear coding, the relay nodes create coded packets of the form $\sum_{i=1}^{k} \alpha_i p_i$, where α_i is a random coefficient chosen over a finite field, and p_i's can be coded or uncoded packets. Assume that the delivery rate of all of the links in 1.4(b) is 0.5. Each source node generates four random linearly coded packets and sends them. Two linearly independent packets from each session are received by the relay node r. Then, the relay node generates four random coded packets for each session

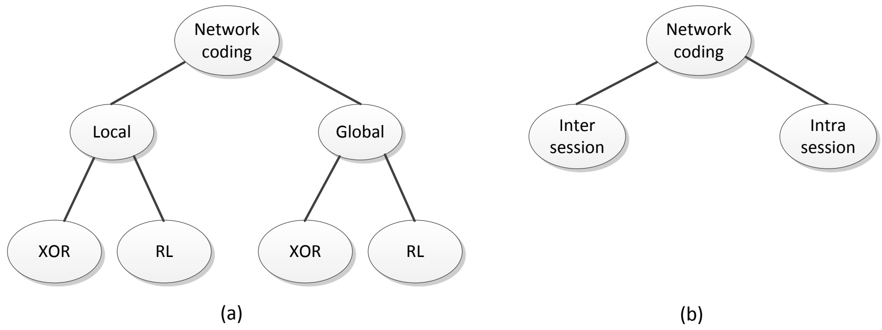

Fig. 1.5 Classification of network coding approaches.

(1.4(c)). Each destination receives two linearly independent packets. The decoding process is similar to solving a system of linear equations.

From another view, we can classify network coding as *local* or *global* coding. In local network coding, a relay node sends the coded packets such that the next hop nodes are able to decode the coded packets. Then, the next hop nodes decode the coded packets and use the same policy to code the packets. Therefore, in a multi-hop transmission, hop-by-hop coding and decoding is performed. In contrast, in global network coding, the intermediate nodes do not perform decoding; they just code the coded packets again. At the end, when the destination nodes receive enough packets, they will be able to decode them. Usually, local network coding protocols use XOR coding, and global protocols perform random linear coding.

As described in the introduction, network coding can be *inter-session* or *intra-session*. Inter-session network coding allows the relay nodes to code packets from the same session (source) to solve the bottleneck problem, and to reduce the number of transmissions (1.4(a)). On the other hand, in intra-session network coding, the relay nodes code packets from the same session to make the importance of the packets the same. Intra-session network coding is a natural way to address the packet loss problem in wireless networks (1.4(b) and (c)). Figure 1.5 shows our classification of network coding methods.

1.3 Network Coding Methods for Unicast Applications

In this section, we describe some of the proposed network coding approaches for unicast application. We categorize the methods based on their methodologies, which are inter or intra-session network coding. Then, we compare the methods and summarize their advantages and drawbacks in the following sections.

1.3.1 Inter-Session Network Coding

COPE. A practical forwarding architecture, called COPE, is proposed in [6] which increases the throughput of wireless networks. This paper addresses the case of unicast traffic: dynamic and potentially bursty flows. COPE incorporates three main techniques, opportunistic listening, opportunistic coding, and learning neighbors' states. In COPE, the nodes snoop on all communications and store the overheard packets for a limited period of time. The nodes broadcast reception reports to tell their neighbors which packets they have in their buffers. On the other hand, in the network coding phase, a node may have multiple choices for coding. However, the goal is to maximize the number of packets delivered in a single transmission, while making sure that all next-hops are able to decode the coded packet, so that they retrieve their respective packets. When it comes to learning a neighbor's state, COPE does not rely solely on the reception reports, since they may get lost or arrive late. For this reason, the delivery rate of the links are computed and broadcasted periodically.

A forwarder node in COPE works as follows. First, it selects a packet at the head of the forwarding queue. Then, it sequentially selects another packet in the queue, and computes the decodability probability of the packets at the next-hops when the packets are coded together. If the decodability probabilities at all of the next-hop nodes are greater than a given threshold, the relay node will code the packets together. Assume that in Figure 1.6, the next-hops for the packets from nodes s_1, s_2, and s_3 are nodes d_1, d_2, and d_3, respectively. Also, assume that the delivery rate of the shown links is 1, but the overhearing probability between the s nodes and d nodes is 0.8. The node r has received packets p_1, p_2, and p_3 from nodes s_1, s_2, and s_3, respectively. First, the relay node selects packet p_1. Then, it computes the decodability probability of the coded packet $p_1 \oplus p_2$ at the respective next-hops of packets p_1 and p_2, d_1 and d_2. This probability is equal to 0.8. Assuming that the coding threshold is equal to 0.8 (this is the default value in [6]), COPE allows these packets to be coded together. Then, the relay node checks the decodability probability when p_3 is coded with $p_1 \oplus p_2$. For all next-hops, this probability is equal to 0.64, which is less than the threshold. Thus, p_3 cannot be coded with the packet $p_1 \oplus p_2$.

With the current sensor nodes' technology, COPE might not much appropriate for sensor networks. First, the nodes in COPE should snoop on all communications and store the overheard packets in their buffer, which is not practical in WSNs, because of the power and memory limitations. Second, the reception reports of the packets and the delivery rate of the links should be broadcasted in COPE periodically; this results in large amounts of power consumption in WSNs.

Centralized Approach. A network coding-aware routing method is proposed in [14] to achieve optimal throughput. In contrast with COPE, in which the routing and coding algorithms are separate, the proposed mechanism in [14] is a cross-layer approach. The authors argue that when the paths of two flows are far apart (the flows pass through nodes that are far from each other), the interference between them is minimized. On the other hand, choosing close flows paths increases the

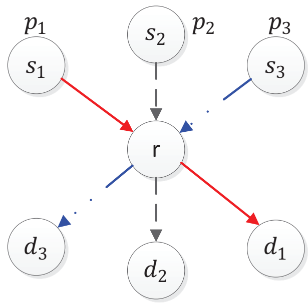

Fig. 1.6 COPE approach.

coding opportunities. Therefore, a trade-off between coding opportunity and conflict should be performed. A conflict graph is used in this work to model the interference between the links, and linear programming is used to find the optimal solution for the joint routing and network coding problem. The main drawback of this work is that the authors do not consider all of the possible overhearing cases between the nodes. In the same way as the COPE approach, this approach is not suitable for sensor networks.

Distributed Approach. The problem of energy efficient opportunistic network coding for multiple unicast flows is addressed in [15]. The proposed inter-sessions network coding method, which is referred to as COPR, decomposes multiple unicast sessions into a superposition of multicast and unicast sessions in wired networks, with coding within each session (note that these sessions are artificial, and they differ from the original sessions). The network is modeled as a directed hypergraph, and the achievable rate region of one-hop XOR network coding is determined under a primary interference model. To simplify the network operation, the authors propose a back pressure algorithm for dynamic scheduling that does not optimize overheard flows.

Network coding opportunities are not fully exploited in TCP flows over wireless network coding, due to the bursty behavior of the flows. Rate mismatches between the flows reduce the coding opportunities since the intermediate nodes may not have enough packets from different flows to code together. [16] addresses this problem by proposing coding-aware queue management for unicast flows. The authors formulate congestion control as a network utility maximization problem and solve it via a distributed scheme. Using the optimal solution, a network coding-aware queue management scheme at intermediate nodes (NCAQM) is proposed, which stores coded packets and drops packets based on network coding and congestion information. NCAQM does not change the TCP or MAC protocols, which makes the approach practical. The bursty flows are not usual in WSNs, so this method might not be very useful for their current applications.

Analysis. A formal analysis on the performance of COPE is provided in [17]. The authors use the encoding number as the performance measure. The encoding number is defined as the number of packets that can be coded together at a relay node in each transmission, and an upper bound on the encoding number at a single relay node is proposed. It is shown that, in the case of overhearing, the upper bounds of 2D and 3D networks are equal to 5 and 6, respectively. The authors also propose a methodology for computing the average coding number under a general class of a random access link-scheduling mechanism. They extend their analysis to general multi-hop wireless networks, and they formally prove the upper bound of the throughput gain for the practical XOR coding scheme.

Lossy Links. The CLONE approach, which is a loss-aware network coding method, is proposed for unicast sessions in [18]. The relay nodes use local XOR coding to code the packets from different sessions. However, in contrast with the relay nodes in COPE, which try to send the minimum number of transmissions, in order to achieve higher throughput, the relay nodes in CLONE use redundancy to increase the probability of delivering the packets. The idea can be motivated by the example in Figure 1.1(b). Assume that the links from nodes s_1 and s_2 to the relay node are loss free, and the loss probability of the links from the relay node to the source nodes is P'. Using COPE, the number of transmissions is equal to 3, and the number of received packets is equal to $(1-P')$. Therefore, the throughput is equal to $(1-P')/3$. On the other hand, if the relay node transmits the coded packet twice, the throughput will be equal to $(1-P'^2)/4$; thus, for $P' = 0.5$, the throughput of the first and the second schemes are equal to 0.167 and 0.1875, respectively. In CLONE, the relay nodes construct redundant coded packets such that the delivery probabilities of the original packets to their next-hop achieve a given threshold. CLONE is not deployable in practice (especially in WSNs) due to its computational complexity. In addition, intra-session network coding provides a more efficient way to address the lossy behavior of the links, which is discussed in the following section.

Flow-Based Approach. The authors in [19, 20] use inter-session network coding to increase the throughput of multi unicast flows, while maintaining fairness between the flows. The optimal solution for lossy 2-hop relay networks is #P-complete when the packets are considered separately. For this reason, in this work, the authors consider flows instead of individual packets. Using this policy, they optimize the overhearing and characterize the capacity region in the form of linear equations when XOR network coding is used. Linear programming is used in this work to compute the capacity region.

SenseCode. The authors in [9] use network coding to provide a reliable and energy-efficient data gathering approach in WSNs. It is assumed that the sensing task is periodic, and during each round all of the nodes should send their sensed data to a sink node. They argue that the traditional tree-based methods, in which each intermediate node transmits the received packets from its children nodes to its parent, cannot provide reliability. The reason is that in the case of node or link failures, the data will not be able to reach the sink node. In order to solve this problem, in SenseCode, the sensed data from each sensor is transmitted through different paths. In this method, each node stores all of the messages it has generated by itself, and the packets it has received from its children nodes during the current round, in a queue. The node also stores the overheard packets to a separate queue. When a node has a new message to send to the sink node, the node creates a packet and marks it as uncodabe. Then, the message will be transmitted to the parent node. Moreover, the node sends $R-1$ linear combinations of the packets from its queues and marks them as codable. Here, R is a configurable redundancy factor. Also, when a node receives a packet from its child node, and it is marked as uncodable, the node will relay the packet. If the packet is marked as codable, the node sends a linear combination of the received packet and the packets in its queues. In this way, some of the packets will be received by the sink node as uncoded packets, which provides reliability even in the case of high link loss rates.

Physical Layer Coding. A physical layer network coding scheme (PNC) is proposed in [21] for linear networks. In contrast to traditional network coding schemes, where coding is performed by the relay nodes on digital bits, PNC makes use of the additive nature of simultaneously-arriving electromagnetic waves for network coding. Take the example in Figure 1.1, in which nodes s_1 and s_2 want to exchange their packets through relay node r. In binary network coding, the source nodes send their packets in different time slots, and the relay node XORs the packets after receiving them. In contrast, in PNC, the source nodes transmit their packets simultaneously, so the relay node receives a combined signal. Assume that the signals sent by nodes s_1 and s_2 are $a_1\cos(\omega t)+b_1\sin(\omega t)$ and $a_2\cos(\omega t)+b_2\sin(\omega t)$, respectively. Then, the signal received by the relay node will be in the form of $(a_1+a_2)\cos(\omega t)+(b_1+b_2)\sin(\omega t)$. The relay node maps the received signal, such that when the nodes s_1 and s_2 receive the mapped signal, they will be able to extract the signal sent by the other source node. Physical layer coding can be very useful and efficient for WSNs [22]. In these networks, the sensor nodes are placed in a line to monitor linear structures like roads, or long pipelines carrying oil, gas and water resources, etc.

1.3.2 Intra-Session Network Coding

In the previous section, we reviewed some of the inter-session network coding approaches that have been proposed for lossy environments. However, the natural way to address the loss problem is to use intra-session network coding, which makes the importance of the transmitted packets the same. In this section, we will review some of these approaches.

MORE. An opportunistic routing method, called MORE, is proposed in [8]. This approach, which uses random linear network coding, can be used for unicast and multicast applications. In contrast with traditional routing methods, in which the path from the source to the destination node is predetermined, opportunistic routing allows any node that overhears the transmission and is closer to the destination node to participate in forwarding the packet [7]. However, opportunistic routing faces two challenges. Multiple nodes may overhear a packet and forward the packet. Also, the

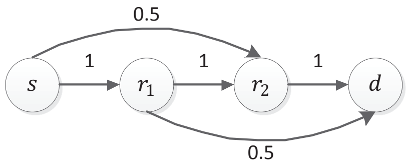

Fig. 1.7 Opportunistic routing.

MAC protocol needs to be modified. MORE uses random linear network coding to address these problems.

Consider Figure 1.7. Traditional routing sends the packet along path $s \to r_1 \to r_2 \to d$. However, there is a chance that node r_2, which is closer to the destination node, will receive some of the packets. For example, assume that the source (s) sends two packets, p_1 and p_2. Both of them are received by the node r_1, and the node r_2 received the packet p_1. Therefore, node r_1 does not need to forward packet p_1 since a node closer to the destination node (r_2) can forward the packet. In order to prevent this unnecessary transmission, the nodes need to be coordinated, which is hard for large networks. To solve this problem, node r_1 can forward a random linear combination of the packet, $c_1p_1 + c_2p_2$. The node r_2 can easily retrieve the missed packet p_2 by subtracting c_1p_1 from the received coded packet.

MORE works as follows. The source node breaks up the file into batches of k uncoded packets, called *native packets*. The source creates a random linear combination of the native packets in the current batch, and broadcasts the coded packet. A coded packet $\sum_{i=1}^{k} \alpha_i p_i$, where α_i is a random coefficient and p_i is the native packet of the current batch. The source node attaches a header to each packet, which contains the coefficients and the list of forwarder nodes. MORE uses ETX (expected number of transmissions) to compute the forwarder list. The source node includes the nodes which are closer to the destination node (in term of the ETX metric) in the forwarder list. When a forwarder node receives a packet, the node checks if the packet contains new information. In other words, the node checks whether the new received packet is linearly independent from the received packet in the node's buffer, in which case it is called an *innovative packet*. Non-innovative packets will be ignored. Otherwise, the node generates a linear combination of the received coded packets from the current batch and broadcasts it. When the destination node receives k linearly independent packets, it can decode the whole batch.

The remaining question in MORE is: how many packets does each forwarding node need to send when the node receives an innovative packet from an upstream node? The authors use the ETX metric to calculate the number of transmissions that should be done at a forwarder node upon receiving an innovative packet from an upstream node. They call this expected value TX_credit (transmission credit).

MORE is not suitable for WSNs. The reason is that in MORE, every node can be a potential forwarder to transmit the packets from the source node to the destination.

Therefore, the nodes should remain in active mode to participate in opportunistic routing, which increases the energy consumption of the sensor nodes.

Extensions over MORE. MORE does not consider the possible congestion caused by multiple forwarders that have new packets to transmit. The problem arises when a large number of intermediate forwarders are involved in the unicast. A distributed optimization framework, called OMNC, is proposed in [23]; OMNC jointly optimizes rate control and multi-path routing. OMNC avoids network congestion through its rate control mechanism. Instead of determining the number of packets, OMNC assigns the encoding and broadcast rate to each node in a decentralized manner, and tries to optimize the bandwidth usage and congestion avoidance. OMNC is designed for long-lived unicast sessions in lossy wireless networks.

The authors in [24] address the problem of resolving conflicts of interest among multiple competing flows with wireless multi-path network coding. They use game theory to optimize resource allocation for network coding-based unicast protocols. In the proposed framework, called Dice, the problem is modeled as a network game, in which players share the bandwidth resource through negotiation or competition. For both cases, the players, which are the end users (destinations), perform a localized optimization of two subproblems: multi-path opportunistic routing, and the broadcast and coding rate allocation among competing players.

Dividing the packets into different batches (segments) and performing coding between the packets from the same segments is referred to as *segmented network coding*, which reduces the complexity of network coding. In MORE, the source node transmits only one segment at any time while waiting for acknowledgment from the destination node. This stop and wait policy degrades performance, as it leads to wasted wireless bandwidth. Also, the existence of just a single segment in the network may not be sufficient to saturate its delay-bandwidth product. This problem is addressed in [25] by allowing the coexistence of different segments. In the proposed method, called CodeOR, the source node transmits W (window size) concurrent segments. When the source node receives end-to-end feedback from the destination node, the node adds a new segment to the current window. In addition, each downstream node sends one-hop feedback after receiving a sufficient number of coded packets. The authors propose a heuristic to calculate the threshold for the sufficient packets at a given node. When a relay node (including the source node) receives an acknowledgment from all of its downstream nodes, it starts sending the packets of the next segment. The authors also adopt a similar algorithm to TCP Vegas [26], which uses increased queueing delays as congestion signals.

The authors in [27] propose an optimization framework for opportunistic routing based on network utility maximization (NUM), and they derive optimal scheduling, routing, flow control, and rate adaptation schemes. In this work, the links' rate constraints are defined per broadcast region instead of unicast links. The authors prove the optimality of their approach, and derive a primal-dual algorithm that is the basis of their practical protocol.

CCACK. The performance of MORE depends on the accuracy of the estimated loss rates. Loss rates change over time, but to reduce the overhead of calculating and collecting loss rates in MORE, the loss rates are collected only before the source

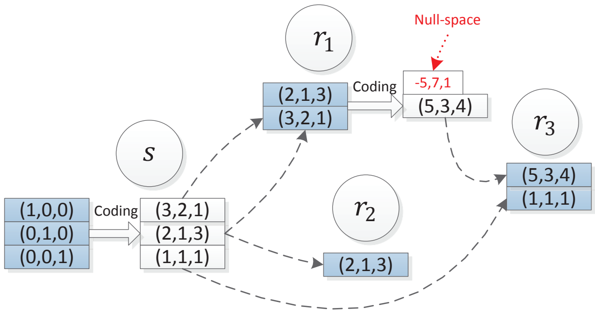

Fig. 1.8 CCACK approach.

node starts the transmission of the packets. MORE also assumes that the links are independent, and it does not consider the correlation between the links. The CCACK approach [28] solves these problems. In CCACK, nodes use cumulative coded acknowledgments, which allow nodes to acknowledge the coded received packets to their upstream nodes, using a single compressed feedback message, with almost zero cost. For this purpose, each node calculates the coefficients' null-space of the received coded packets, and the node adds the null-space to the forwarding messages. The null-space of a set of vectors V is a vector z, such that the inner products between z and each vector in V is zero. When an upstream node overhears a packet from a downstream node, the upstream node multiplies the coefficient of packets in its buffer with the received null-space. A non-zero result means that the packet in the buffer is innovative to the packet in the downstream node's buffer. CCACK is not applicable to WSNs, for the same reason that MORE is not applicable.

Let's consider Figure 1.8. The source node s has three packets in its buffer. The node constructs three coded packets and broadcasts them. Assume that all nodes need to decode all of the packets. Node r_1 has only two packets, so it is not able to decode the received packets. The node r_1 will send a null-space of the received coefficient vectors, which can be any vector of form $(-5y, 7y, y)$. Suppose that z is chosen as $(-5,7,1)$. Since $(-5,7,1).(1,1,1) = 3 \neq 0$, node r_3 must transmit the packet $(1,1,1)$ to node r_1. On the other hand, $(-5,7,1).(2,1,3) = 0$, so node r_2 does not have any innovative packet for node r_1.

The use of null-space in opportunistic routing suffers from a problem called the collective space problem [28]. Suppose that nodes r_3, r_2, and r_1 are sorted in increasing order of their distance from the destination node. Nodes r_2 and r_3 collectively cover all of the three sent packets from the source node. As a result, the node r_1, which is farther from the destination node, does not need to transmit any more packets. However, the inner product of the packet $(3,2,1)$ in the buffer of r_1 and the null-space of r_2 and r_3 are not zero. The reason is that r_1 does not consider the col-

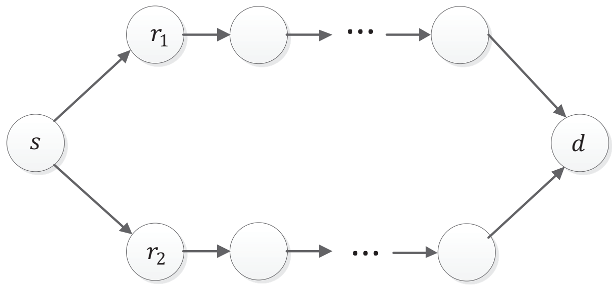

Fig. 1.9 The drawback of the CCACK approach when the links are highly correlated.

lective covered space by nodes r_2 and r_3. In order to solve this problem, the authors in [28] use separate buffers at each node i for the coefficient of the received packets from upstream nodes (B_u), the coefficient of sent coded packets (B_w), and the received innovative packets (B_v). When the node i overhears a coded packet from the downstream nodes, the node marks the coefficients in B_u and B_v, if their inner product with the recited null-space is equal to zero. When the rank of the marked coefficient vectors in $B_u \cup B_w$ becomes equal to the rank of the packets in buffer (B_v), the downstream nodes (which are closer to the destination node) collectively cover all of the packets in the node i's buffer. Therefore, node i does not need to transmit more packets.

Most of the proposed methods for the networks with lossy links assume that the links are independent, and they do not consider the effect of the correlation between the links [29] on the performance. Take Figure 1.9 for example. Assume that each node stops to transmit more packets when its next-hop nodes have collectively received the same number of linearly independent packets to what it has in its buffer. Assume that the delivery rate of the links between the source node s and the nodes r_1 and r_2 is 0.5, and the batch size is 6. In the case that the links $s \to d_1$ and $s \to d_2$ are independent, the source node needs to try 8 transmissions, as the probability of receiving a transmission by at least one of the nodes d_1 and d_2 is 0.75. In the case of highly correlated links, either both of the nodes will receive a transmission or none of them will. Therefore, the source node needs to transmit 12 packets. When the links are negatively correlated, exactly one of the nodes d_1 and d_2 will receive a transmitted packet. As a result, the number of required transmissions by the source node will be 6. It can be inferred that correlation between the links has a huge effect on the throughput of the methods. Now assume that the links are highly correlated, so the nodes r_1 and r_2 will receive all of the six packets. Since they are not aware of each other's received packets, both of them will send all of the packets, which results in unnecessary redundant transmissions. This problem can be solved by giving a credit, equal to 3, to nodes r_1 and r_2. The work in [30] considers correlation between the links and improves the performance of CCACK.

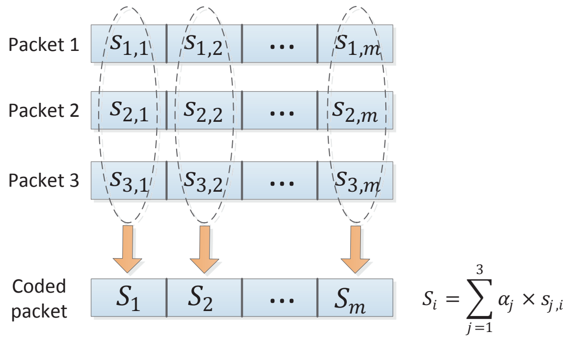

Fig. 1.10 Symbol-level network coding.

MIXIT. Symbol-level network coding for wireless mesh networks in introduced in [31]. The main idea behind the MIXIT approach is that even when no node receives a packet correctly, any given bit might be received by some node correctly. As a result, instead of insisting on forwarding only correct packets, the intermediate nodes can forward the correct received bits to the destination. For this purpose, the intermediate nodes in MIXIT use physical layer hints to guess which bits in a corrupted packet are likely correct. Unlike the previous work on network coding, the network code in MIXIT operates at the granularity of symbols, which is defined as a small sequence of bits, rather than packets. Take figure 1.10, in which the original symbols and the coded symbols are noted as s and S, respectively. In contrast with the packet-level network coding, each coded packet in MIXIT consists of multiple coded symbols. As a result, if some parts of a packet encounters with an error, the other symbols are still useful. In MIXIT, each router forwards random linear combinations of the high-confidence symbols belonging to different packets, and the destination node is able to decoded the symbols once it receives enough number of coded packets.

The first problem that MIXIT addresses is using an scalable coordination among the nodes in order to prevent duplicate transmissions of the same symbol. In contrast with node coordination-based approaches like ExOR, MIXIT uses the randomness from the network code and a dynamic programming algorithm to solve the cordination problem. The second issue is error recovery. The destination node needs to correct the errors that might exist in the received symbols. MIXIT uses symbol-level network coding along with an end-to-end Maximum Rank Distance (MRD) codes [32] for this purpose. The routers in MIXIT only forward random linear combinations of high-confidence symbols, and they do not perform any error correcting. MIXIT protocol benefits from a congestion-aware forwarding. It forwards coded symbols through paths that have small queues and high delivery probabilities. MIXIT may be applicable in WSNs to deliver data to sink nodes. In WSNs most

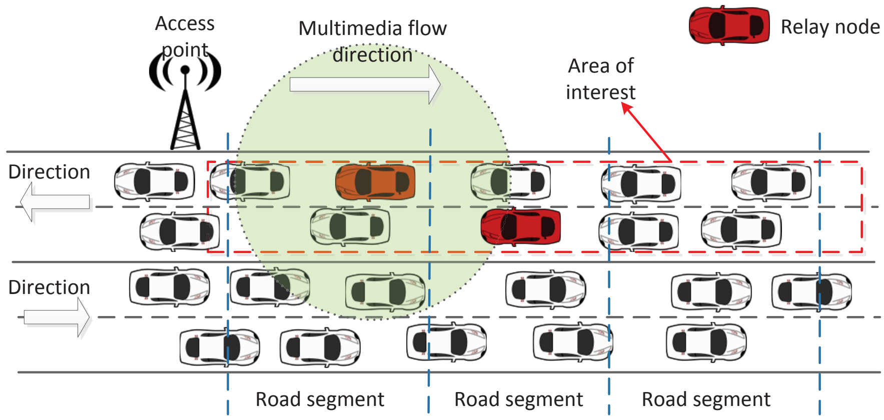

Fig. 1.11 Multimedia Streaming in VANETs.

traffic is from the sensors to the sink node, so data from different sensor nodes can be coded together to improve throughput. The MIXIT protocol can also be used for multicast applications in mesh networks. For this purpose, routers can keep transmitting coded packets until all destination nodes can decode them.

The authors in [33] show that the symbol-level network coding outperforms the packet level network coding for content distribution in Vehicular Ad-Hoc Networks (VANET). In [34], they later study the advantage of symbol level network coding for live media streaming in VANETs. As shown in Figure 1.11, the goal in [34] is to designate live streaming multimedia to all of the nodes in a specific region of a road, called area of interest. The core part of the proposed method, called CodePlay, is a coordinated local push mechanism. In order to disseminate the content from sources to all the receivers smoothly and timely, a set of spatially separated relay nodes are selected in CodePlay distributively. The relay nodes are selected in such a way that their transmissions can bring most useful information to their nearby vehicles. For this purpose, CodePlay uses an objective function to calculate the contribution of each potential relay node. The proposed method segments the road during initialization so that the relay selection could be made locally within each segment. Each selected relay node actively pushes coded data to cover its neighborhood. Using symbol-level network coding CodePlay can better tolerance transmission interference, and concurrent transmissions of all relays could be optimally coordinated locally. In CodePlay, adjacent segments share the wireless channel resource in a round-robin fashion to reduce interference.

1.3.3 Joint Inter and Intra-Session Network Coding

It is not desirable to use inter-session network coding alone in a lossy link environment, since intra-session network coding is an efficient way to deal with the lossy links. Thus, it is critical to have joint inter and intra-session network coding for wireless networks with lossy links.

The work in [35] proposes a heuristic to combine inter and intra-session network coding in lossy multi-hop wireless networks. This approach limits network coding to be within a hop and provides a limited performance gain in the range of 20% to 30%. Also, the proposed approach in [35] lacks theoretical analysis.

A joint inter and intra-session network coding scheme, called I^2NC, is proposed in [36]. This work is grounded in network utility maximization formulation of the problem. Assuming that the number of packets in each segment is k, each relay node constructs $k+k'$ linear combination of the packets instead of k coded packets. It is sufficient for the receiver nodes to receive k out of $k+k'$ packets. In other words, the k' additional packets work as parity packets. After adding redundancy to the packets, and coding them together, I^2NC uses inter-session network coding to mix the coded packets of different sessions. The authors propose two schemes: I^2NC-state and I^2NC-stateless. In the former scheme, each node listens to all transmissions in its neighborhood, stores the overheard packets in its buffer, and periodically informs its neighbors about the content of the buffer. In contrast, the I^2NC-stateless scheme only relies on the local loss-rates of the links.

A cross-layer optimization scheme for lossy 2-hop relay networks is proposed in [37] that uses joint inter and intra-session network coding. The work optimizes overhearing, considers flows instead of packets, and assumes limited feedback. Linear equations are used to characterize the capacity region for the problem of when the number of sessions is less than three. Also, a near-optimal coding scheme is proposed for the case with more than two sessions, and its performance is characterized using linear equations. However, the complexity of the near-optimal scheme is hyper-exponential.

A polynomial time coding method for the 2-hop relay network problem is proposed in [38, 39]. This scheme, which uses random linear network coding, makes a linear number of decisions. The authors characterize the performance of their scheme by using linear constraints in terms of link delivery rates. They use the proposed 2-hop relay scheme as a building block to extend the proposed scheme to multi-hop wireless networks. Based on this policy, a linear programming formulation of the achievable rate region is proposed.

1.3.4 Summary and Discussion

COPE is the first proposed practical inter-session network coding method. Its complexity is not high, and it works in networks with perfect links. However, COPE is not appropriate when the links have a moderate loss probability of 20%, as it turns off coding in this case. CLONE solves this problem by sending different redundant coded packets such that a given level of reliability is provided. However, CLONE does not optimize the overhearing, and it limits the operation to XOR. As the optimal solution is #p-complete, approximation heuristics are proposed. In [19], the authors tackle the problem of optimal inter-session network coding from a different angle, as they consider flows instead of packets. They optimize overhearing and characterize the capacity region. The authors in [14] propose a cross-layer method that combines the routing and inter-session network coding. They model interference between the nodes as a conflict graph, and they find the optimal solution by using optimization techniques. The drawback of this work is that some overhearing cases are not considered during the formulation of the problem.

MORE is a practical opportunistic routing approach that uses intra-session network coding to provide reliability in lossy link environments. In MORE, there is no need to send feedback messages from the intermediate nodes, and only when the destination nodes receive all of the packets is a feedback message sent to stop the source node from sending more packets. The Dice method addresses the problem of a conflict of interests among multiple flows. The CodeOR protocol solves the stop and wait problem of MORE, which degrades performance. The CCACK method is proposed to solve vulnerability of MORE to links' quality changes. Instead of estimating the number of required transmissions, in CCACK, the intermediate nodes send the null-space of the received coded packets to help their neighbors discover when they should transmit more packets.

MIXIT proposes the idea of performing network coding in the granularity of symbols instead of packets to increase the transmission efficiency in lossy environments. CopePlay uses the idea of symbol-level network coding for live multimedia streaming in VANETs. Using symbol-level network coding for the highly mobile nodes in VANETs decreases the interference problem by enabeling using the received correct symbol even in the case that a packet is not received correctly.

Table 1.1 classifies the discussed methods for unicast application based on the used methodology. This table also shows the objective of the approaches, and whether they assume the existence of lossy links or perfect links.

1.4 Network Coding Methods for Multicast Applications

In this section, we look at network coding approaches that can be used for multicast applications. With simple modifications, some of the proposed approaches for unicast application can be applied for multicasting. To the best of our knowledge there is no inter-session network coding for multicasting in wireless networks. It should be noted that there are some works on inter-session network coding for multicasting in wired networks, which are beyond the scope of this chapter.

Approach	**Methodology**	**Topology**	**Objective**	**XOR or RL**	**Local or Global**	**Links**
COPE [6]	Inter-session	Multi-hop	Throughput	XOR	Local	Lossy
[14]	Inter-session	Multi-hop	Throughput	XOR	Local	Perfect
[15]	Inter-session	Multi-hop	Energy efficiency	XOR	Local	Lossy
NCAQM [16]	Inter-session	Multi-hop	Throughput	XOR	Local	Perfect
CLONE [18]	Inter-session	Multi-hop	Throughput	XOR	Local	Lossy
[19]	Inter-session	Multi-hop	Throughput and fairness	XOR	Local	Lossy
SensCode [21]	Inter-session	Multi-hop linear network	Throughput	Physical	Local	Perfect
MORE [8]	Intra-session	Multi-hop	Throughput	RL	Global	Lossy
OMNC [23]	Intra-session	Multi-hop	Throughput	RL	Global	Lossy
Dice [24]	Intra-session	Multi-hop	Throughput and fairness	RL	Global	Lossy
Dice [27]	Intra-session	Multi-hop	Throughput	RL	Global	Lossy
CCACK [28]	Intra-session	Multi-hop	Throughput	RL	Global	Lossy
[30]	Intra-session	Multi-hop	Throughput	RL	Global	Lossy
MIXIT [31]	Intra-session	Multi-hop	Throughput	RL (symbol-level)	Global	Lossy
CopePlay [31]	Intra-session	Multi-hop	Throughput	RL (symbol-level)	Global	Lossy
[35]	Joint Inter and Intra-session	Multi-hop	Throughput	XOR	Local	Lossy
[36]	Joint Inter and Intra-session	Multi-hop	Throughput	RL	Local	Lossy
[37]	Joint Inter and Intra-session	2-hop	Throughput and fairness	RL	Local	Lossy
[38]	Joint Inter and Intra-session	2-hop Multi-hop	Throughput and fairness	RL	Local	Lossy

Table 1.1 Classification of the network coding methods for unicasting.

1.4.1 Intra-Session Network Coding

In addition to unicast applications, MORE [8] can be used for multicasting. For this purpose, the authors make simple modifications to their unicast algorithm. The first reconciliation is that the source node does not proceed to the next batch until all of the destinations receive the packets in the current batch. Also, the list of forwarder nodes for multicast applications differs from the unicast. The source node computes the list of forwarder nodes for each unicast flow from itself to the destinations in the multicast group. The forwarder list of the multicast flow is the union of the forwarders of the unicast flows. Moreover, the TX_credit (transmissions credit) at each forwarder node is the maximum of the required transmissions for different unicast flows in the multicast group. The last modification is that when the source node receives a feedback message from a destination node, the source node recomputes the list of the forwarder nodes and their TX_credits for the remaining destinations. As discussed before, MORE is not applicable in WSNs.

The modified MORE for multicast applications suffers from two problems. First, it can lead to congestion since too many nodes may act as forwarder nodes, even for a single destination. This situation is worsened as the number of flows increases. Next, if one of the receivers has a poor connection, then trying to satisfy reliability for this receiver may result in throughput degradation for the other receivers. This problem is called the *crying baby* [40] problem and is unique to multicast. The Pacifier approach [41] proposes a multicast tree-based opportunistic routing design to solve these problems. Pacifier creates a multicast tree to connect the source node to the multicast receivers. The source node builds this shortest-ETX tree by taking the union of all of the shortest-ETX paths from the source to the receivers. The source node reconstructs this tree when a receiver node receives the complete batch. In contrast with MORE, in which every node with a greater ETX value can be the next-hop, Pacifier limits the forwarder nodes to the nodes in the multicast tree. The source node assigns a TX_credit to each forwarder node. TX_credit specifies how many packets that a forwarder node should transmit upon the reception of a packet from an upstream node (a node with a greater ETX).

To solve the crying baby problem, Pacifier changes the sending pattern of the batches. In MORE, the source node does not start transmitting the next batch until all of the destination nodes acknowledge reception of the current batch. However, the source node in the Pacifier approach transmits the packets in a Round-Robin pattern. In details, when one of the receivers sends the acknowledgment of receiving the current batch, the source node moves to the next batch. Forwarder nodes only buffer the packets belonging to the current batch, and the nodes delete their buffer upon reception of a packet from a new batch. The source node will continue sending the packets from the first batch when the other batches are received by at least one of the destination nodes. Figure 1.12 describes the order of transmitting the packets by the source node in the MORE and Pacifier approaches. In contrast to MORE, Pacifier is suitable for WSNs. The reason is that Pacifier limits the forwarder nodes to the nodes in the multicast tree. This method can be used in WSNs to send code updates or other data from the sink node to a group of sensor nodes.

In [42], the authors address the network coding-based opportunistic routing problem for multicast. They argue that the important factors that affect the performance of the multicast protocols are loss rate, the correlation among the links, and the reachability of the node. They formulate the optimal network coding-based opportunistic routing for multicast as an optimization optimization, and develop a distributed algorithm for the problem in which each node only requires local information. The proposed distributed algorithm consists of two phases. In the first phase, the proposed method uses ETX metric to construct the most reliable broadcasting tree. In the second phase, each node runs a credit assignment algorithm to calculate the number of coded packets that it has to send. The authors show that the proposed distributed algorithm adapts to the changes in the channel conditions, and converges

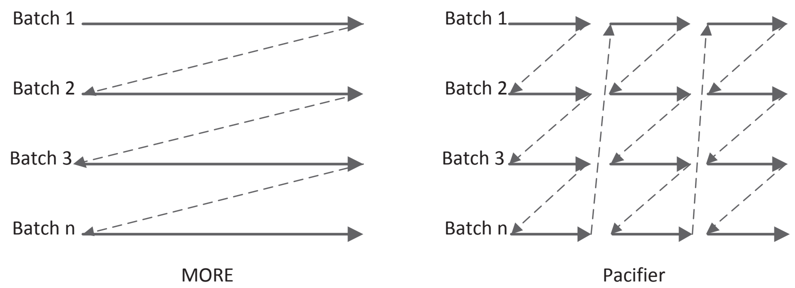

Fig. 1.12 The order of transmitting the batches of packets in the MORE and Pacifier.

Approach	**Methodology**	**Topology**	**Objective**	**XOR or RL**	**Local or Global**	**Links**
MORE [8]	Intra-session	Multi-hop	Throughput	RL	Global	Lossy
Pacifier [41]	Intra-session	Multi-hop	Throughput	RL	Global	Lossy
CCACK [28]	Intra-session	Multi-hop	Throughput	RL	Global	Lossy
[42]	Intra-session	Multi-hop	Throughput	RL	Global	Lossy

Table 1.2 Classification of the network coding methods for multicasting.

to the optimal solution. Moreover, this approach does not need any explicit knowledge about the correlation among the links or the channel conditions. In addition to using coded packet, the authors use coded feedback messages to reduce the number of feedback messages, and to resolve the problem of delayed feedback. The simulation result in the paper show the effectiveness of the proposed method over the MORE approach. The distributed approach can be applied on WSNs in multicast applications such as software update of the sensor nodes.

Table 1.2 classifies the discussed methods for multicast application, based on their methodology, objective, and whether they assume the existence of lossy links or perfect links.

1.5 Network Coding Method for Broadcast Applications

In this section, we survey some of the works that address inter-session and intra-session network coding for broadcast applications. At the end, we compare the proposed methods and summarize the results.

1.5.1 Inter-Session Network Coding

CODEB The problem of minimizing the number of transmissions in all-to-all broadcasting is addressed in [43]. All-to-all broadcasting is a special case of broadcasting, in which all node broadcast their respective packets to all other nodes. In this paper, the authors combine network coding with a deterministic forwarding approach, and they show that using network coding results in a significant reduction in the number of transmissions. They apply coding to the Partial Dominant Pruning (PDP) [44] forwarding approach, which is a local forwarding method, but their coding algorithm can be applied to other localized deterministic approaches. The PDP approach is used to select a subset of the nodes as the relay nodes. In PDP, each relay node uses two-hop local information to select a subset of its neighbors such that they can cover all two-hop neighbors of the relay node. In the CODEB approach, each relay node maintains a neighbor reception table that shows the received packets by each neighbor. CODEB is inappropriate for WSNs, because of its overhead.

In the proposed XOR-based coding method, the relay nodes code the packets such that their neighbors can decode the received coded packets using the packets in their buffer. This means that the receiver nodes can decode the received XOR-ed coded packet without waiting for more packets to arrive. In more detail, each relay node with a set of packets in its output queue tries to find a subset of the packets to XOR, such that the number of native packets in the coded packet are maximized. In [43], it is shown that this problem is NP-hard. Therefore, a greedy algorithm is proposed. This greedy algorithm selects the first packet in the output queue and sequentially checks if other packets can be XOR-ed with this packet, such that all of the neighbors can decode the coded packet. In the case of delay-tolerant networks, when there is no coding opportunity at a relay node, the node will postpone the transmission of the packets for a random amount of time.

Directional Antenna. The problem of efficient broadcasting using network coding and directional antennas is studied in [45]. A node with directional antenna capabilities can divide the omnidirectional area into different sectors and turn a subset of them on for transmission. Therefore, in the proposed method, the forwarder nodes transmit the coded or uncoded broadcast messages to restricted sectors, which decreases the energy consumption. The authors assume that the links are perfect, and they use a directional connected dominating set (DCDS) [46] to construct a directional network backbone. A connected dominating set is a subset of the nodes such that all of the nodes in the set are connected together. In addition, each node of the network is either a member of this set or is connected to a member of this set. In the proposed method, each forwarder node performs the coding between the packets that should be sent in the same section. The proposed approach can be applied to WSNs that their sensor node are equipped with directional antenna; however, the method might not be realistic, as the links are assumed to be perfect.

Deadline-Aware. The problem of deadline-aware broadcast scheduling using network coding is considered in [47]. It is assumed that each packet has a deadline, by which it must be sent by a relay node. To solve the problem of minimizing the number of transmissions at a relay node subject to deadline constraints, the au-

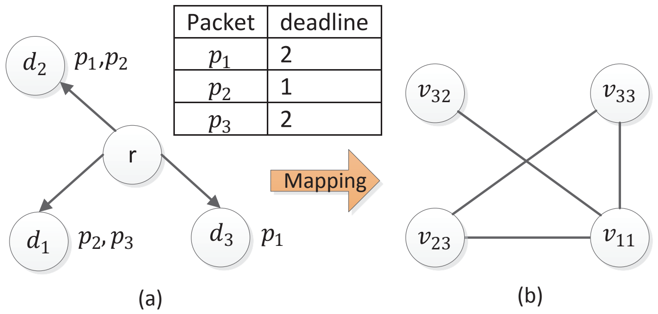

Fig. 1.13 Mapping delay-aware network coding problem to maximal clique problem.

thors map the problem to a maximal clique problem. In the mapped problem, each vertex represents a packet needed by a node in the network. There is a link between two vertices if the vertices correspond to the same packet, or if the correspondent packet of each of the vertices is received by the correspondent node of the other vertex. This means that these two nodes have received the packet that the other node has missed. Therefore, if we code these packets together, the two destination nodes will be able to decode it.

Take Figure 1.13(a) for example. The received packets of each node are shown beside it. Node v_{23} in the mapped problem (Figure 1.13(b)) shows that node d_2 has not received packet p_3. Since both of the nodes d_2 and d_3 need packet p_3, there is a link between vertices v_{23} and v_{33}. Also, since node d_1 has packet p_2 and node d_3 has packet p_1, the vertices v_{11} and v_{32} are connected. After mapping the problem, a weight is assigned to each vertex. These weights are proportional to the deadline of the packets. As finding the maximal clique in a weighted graph is NP-complete, a greedy algorithm is used in [47] to find the maximal clique. After finding the maximal clique, the relay node codes the correspondent packets of the vertices in the clique together. The drawback of this work is that the deadline of the packets is considered for one-hop transmissions, but it is not clear how to calculate the one-hop delays to meet the global deadline. The authors study the effects of different weight functions in [48].

The problem of deadline-aware broadcasting using binary network coding is addressed in [49]. It is assumed that a subset of nodes are the source nodes, and each packet has a deadline to be received by all nodes, and the nodes have multi-channel multi-radio capability capability. Similar to [43], this work combines PDP, which is a deterministic forwarding approach with binary network coding. In [43], if there is a deadline constraint, the relay nodes will send the received packets immediately, which may decrease coding opportunities. In order to increase the coding opportunities, the authors in [49] propose three methods to compute the waiting time of the packets at the relay nodes, such that the packets meet the deadline constraints. The authors define the extra time as the remaining time to the deadline of the packet minus the maximum remaining hops to the farthest destination nodes (it is assumed

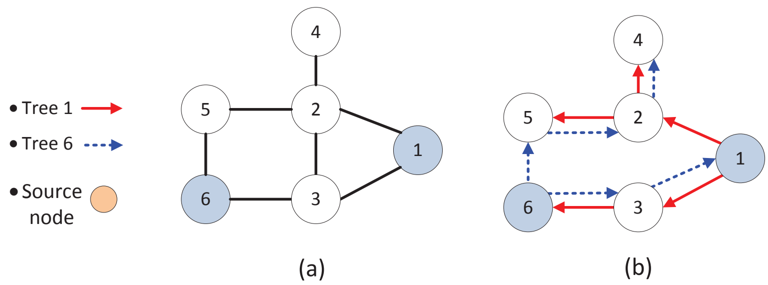

Fig. 1.14 (a) A given topology. (b) Two broadcasting trees routed at the source nodes 1 and 6.

that each transmission takes a unit of time to be received by the next hop). In the first method, which is velocity-based distribution of waiting time, the assigned waiting time to each relay node is equal to the extra time divided by the maximum remaining hops. Because of more coding opportunities at the nodes with more crossing flows, the second proposed method distributes the remaining time proportional to the number of crossing flows to the nodes, which increases coding opportunities. The last proposed method is a random distribution method, which randomly selects each node's waiting time from a specific range. All of the proposed methods in this work are very simple, and their computation complexity are low due to using XOR coding; thus, they can be applied in WSNs.

Authors in [50] study the problem of periodic broadcasting in wireless networks. In this work, a subset of the nodes are the source nodes and their packets should be broadcasted to all the nodes in the network. The authors use random linear network coding to reduce the number of required transmissions. In this work, a broadcasting tree is defined as a spanning tree routed at a source node. The authors propose the idea of using one broadcasting tree for disseminating each source packet, and preforming random linear network coding at the intermediate nodes that are relay nodes in more than one tree. The main idea behind using broadcasting trees is that it ensures decodability of the coded packets at every node, as every node receives enough linearly coded packets. Figure 1.14 (a) show a given topology with two source nodes. The two broadcasting trees routed at the source nodes are shown in Figure 1.14 (b). In this figure, nodes 2 and 3 are relay nodes in both of the trees. As a result, they can encode the received packets. Node 3 linearly combines the packets and transmits one coded packet. In contrast, node 2 needs to transmit two coded packets. This is because that there are two parallel edges from node 2 to node 4, which means node 2 should provide two packets to node 4. In order to minimize the number of parallel edges, which results in less number of transmissions, the broadcasting trees are constructed using a heuristic algorithm. In the next phase, in order to guarantee meeting the packets' deadlines, the authors propose a heuristic to partition the trees such that coding the packets of each partition does not result in any deadline misses. The proposed method can be used in WSNs for periodic broadcasting tasks. However, the drawback of this scheme for WSNs is that the unreliability of the links is not considered in this work.

Analysis. The problem of energy-efficient all-to-all broadcasting is studied in [51]. The presented theoretical analysis shows that network coding improves performance by a constant factor in fixed networks. The authors calculate this factor for some canonical networks, such as circular networks and square grid networks. They also propose a simple algorithm in which each node in the network sends a random linear combination of the received packets with a given probability, called the forwarding factor. To calculate the forwarding factors, two heuristics are proposed that use local two-hop local information. The first heuristic assigns forwarding factors to the nodes inversely proportional to the number of their 1-hop neighbors. The second heuristic sets the forwarding factors of the nodes inversely proportional to the minimum of the number of neighbors of the nodes, 1-hop neighbors. The authors extend their work in [52]. They show that, in networks where the topology dynamically changes and operations are restricted to a simple distributed algorithm, network coding offers improvements of factor $\log n$, where n represents the number of nodes in the network.

1.5.2 Intra-Session Network Coding

One-hop. In [53], network coding is used to decrease the number of required retransmissions due to packet loss in one-hop broadcasting over packet-erasure channels. Firstly, the authors propose two NAK-based (negative acknowledgment) schemes without network coding to provide reliability for broadcasting. In the first proposed network coding-based broadcasting method, the source node receives a NAK message immediately after each message transmission. However, the source does not retransmit the lost packet immediately when it receives the NAK, and it maintains a list of lost packets and the receivers that lost each packet. The retransmission phase starts at a fixed interval of time. Then, the source node tries to code the maximum number of packets in a single coded packet. The source node retransmits the coded packet until all of the destination nodes that have a lost packet in the coded packet receive the packet. In an effort to improve the efficiency of the method, another method is proposed, in which the source node dynamically changes the coded packet based on the received feedback after each retransmission. This approach can be applied for applications such as driver or software updates of sensor nodes in single-hop WSNs. Base station sends the updates in one-hop transmissions to the sensor nodes, and receives NAK from the sensor nodes in the case of transmission errors.

The setting in [54] is the same as in [53], but here the base station (BS) broadcasts a fixed batch of packets. The proposed approach consists of two phases: information transmission phase and retransmission phase. In the first phase, the BS transmits the batch of N packets and receives a feedback message from each destination node. The BS uses the benefit of network coding in the retransmission phase to send the

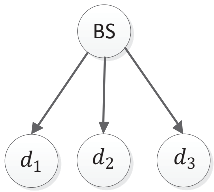

Fig. 1.15 One-hop broadcasting.

lost packets. The authors use binary network coding, and the constraint on coding packets together is that each destination should not have more than one lost packet in the coded packet. Firstly, the proposed algorithm finds the destination with the maximum number of lost packets, and adds each of its lost packets to a different coding set. Then, the algorithm sorts the remaining erased packets in increasing order, according to the number of coding sets, the lost packet can be allocated such that the coding constraint is satisfied. Starting from the packet with the minimum number of choices, the remaining lost packets are allocated to an eligible encoding set. If there is no eligible coding set for a packet, a new coding set will be generated. At the end, the BS node codes the packets of each coding set together and transmits them. This process is repeated until all of the destination nodes receive all of the packets. Similar to [53], this method can be used for driver updates of sensor nodes in WSNs.

Assume that in Figure 1.15, node d_1 missed packets p_1, p_2, and p_3. Also, node d_2 missed packets p_1 and p_4, and node d_3 missed packet p_5. The node d_1 has the maximum number of lost packets. Therefore, the algorithm adds each of the lost packets by node d_1 as a separate coding set. Let $S_1 = \{p_1\}$, $S_2 = \{p_2\}$, and $S_3 = \{p_3\}$. Now, packets p_4 and p_5 are remaining. Packet p_4 can be added to sets S_2 and S_3, but packet p_5 can be added to S_1, S_2, or S_3. The packet p_4 has the smallest number of choices, so the algorithm will add it to one of the sets, S_2 or S_3. Assume that packet p_4 is added to set S_2. Packet p_5 can be added to all of the sets. The final result will be $S_1 = \{p_1, p_5\}$, $S_2 = \{p_2, p_4\}$, and $S_3 = \{p_3\}$.

In [55], the same problem as in [53] is addressed using a different approach. The authors map the problem to a graph coloring problem and introduce a greedy heuristic to solve it. The mapping process is as follows. For each lost packet, a vertex is added to the graph. If two packets are missed by the same destination node, a link will be added between the corresponding vertices. A coding constraint implies that missed packets by the same destination nodes cannot be coded together since the destination node will not be able to decode the coded packet. This constraint is exactly the same as the coloring constraint, in which two neighbor nodes cannot be colored with the same color. Therefore, the vertices (packets) with the same color can be coded together, and the minimum number of required colors for coloring the correspondent graph is equal to the number of transmissions. The graph coloring

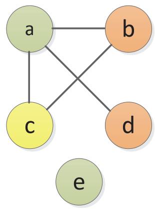

Fig. 1.16 Graph coloring in reliable broadcasting using network coding.

problem is also NP-complete; the authors use the proposed greedy algorithm in [56] to address the mapped problem. This greedy algorithm sorts the vertices in descending order, according to their degree. Then, starting from the first node, the algorithm colors this node and all of the nodes that are not connected to this node with the same color. This process is repeated for the uncolored nodes. Figure 1.16 shows the mapping from the example in the previous paragraph to a graph coloring. The proposed approach is useful for broadcasting data from the base station to the sensor nodes in single-hop WSNs.

The problem of efficient one-hop broadcasting of layered-video is studied in [57]. In this problem, a server node broadcasts a layered-video to a set of users. Because of different channel conditions, the clients receive different number of transmissions from the server. A promising approach to overcome this problem is using Multi-Resolution Coding (MRC) [58, 59, 60]. MRC is originally introduced for wired networks, and it divides a video into a base layer and multiple enhancement layers. In this scheme, the clients can independently decide how many layers to receive from the server according to their available bandwidth from the server. In contrast with the wired networks, in a wireless network all transmitted layers share the medium. As a result, sending higher layers reduces the available bandwidth for sending lower layers. The authors in [57] show that we can overcome the user diversity problem in broadcasting video over Wi-Fi by combining MRC with inter-layer network coding to increase the number of useful layers that can be retrieved by the users (It should be noted that inter-layer coding is different from inter-session coding. This work is in the category of intra-session network coding as the coding is done between the packets of the same session).

It is shown in [57] that inter-layer coding improves the number of decoded layers even for a single receiver. The reason is that it allows retrieving useful layers from more combinations of received transmissions. The authors show that even for a single receiver, the previously proposed even canonical triangular scheme [61, 62] for inter-layer network coding can perform poorly, and they propose two simple heuristics to enhance the gain. In triangular network coding, the encoded layers are in the form of $\sum_{j=1}^{k} \alpha_j l_j$, where $1 \leq k \leq h$ and α_j are random coefficients. In other words, each coded layer is a combination of the first k original layers. The advantage of triangular network coding is that it reduces the number of possible coding strategies.

$$\begin{cases} l_1 \\ l_2 \\ l_3 \end{cases} \quad \begin{cases} l_1,\ l_2,\ l_3 \\ l_1+l_2,\ l_1+l_3,\ l_2+l_3 \\ l_1+l_2+l_3 \end{cases} \quad \begin{cases} l_1 \\ l_1+l_2 \\ l_1+l_2+l_3 \end{cases}$$

(a) (b) (c)

Fig. 1.17 (a) Original packets. (b) General form of random linear network coding. (c) Triangular network coding.

Considering a video with n layers, the possible ways for coding the layers using inter-layer triangular coding and the general form of linear coding are equal to n and $2^n - 1$, respectively. Figures 1.17 (b) and (c) show the possible coded layers of the original layers in Figure 1.17 (a) using the general form of network coding and triangular coding, respectively. We do not show the coefficients in the figures for simplicity. For example, $l_1 + l_2$ means $\alpha_1 l_1 + \alpha_2 l_2$, where α_1 and α_2 are two random coefficients. For the case of multiple receivers, the proposed method calculates the gain of all possible canonical triangular coding, and selects the best one. The authors propose three optimization technics that drastically reduce the complexity of scanning the gain of all the possible canonical triangular schemes.

Relay-Aided. The problem of efficient relay-aided one-hop broadcasting is addressed in [63]. This paper is an extension of [54], in an effort to use a relay node (Figure 1.18). The authors assume that the links are lossy, and the base station (BS) to relay channel and the relay to users' (destinations) channels are better than the BS to user channels. The proposed method has 3 phases. In the fist phase, the BS transmits N packets to the relay and user nodes. Then, the user nodes send feedback messages to notify the BS and the relay node about the received packets by the user nodes. Also, the relay node sends a feedback message to the BS node. In the second phase, the BS node retransmits the set of lost packets by the user nodes. The BS node uses network coding to increase the efficiency of this phase. After transmitting all of the lost packets, the relay and user nodes send new feedback messages. The BS node repeats this process until the user nodes receive all of the packets or the relay node receives all of the lost packets by the user nodes. The third phase is similar to the second phase, but the relay node performs retransmissions instead of the BS node. This is because the relay node has all of the lost packets by the user nodes, and the relay to user links are better than BS to user links. Much like as [53], this method is appropriate for data transmissions from the base station to the sensor nodes.

Multi-hop. A reliable data dissemination protocol using adaptive network coding, called AdapCode, is proposed in [10]. The authors use linear network coding to reduce the traffic in WSNs, which results in increasing the battery life of the sensors.

They show that when nodes have more neighbors, we can increase the segment size (the number of packets that will be coded together). This allows us to encode more packets together without losing reliability, since they can get enough coded packets from their neighbors. Based on this observation and the fact that the network topology may change, an adaptive network coding protocol is proposed, where nodes dynamically change the segment size.

An extension over AdapCode method is proposed in [64]. In the AdapCode approach, throughout the code dissemination process, each sensor node dynamically decides how many packets should be coded together (decides about the segment size). The performance of the AdapCode method highly depends on the density of the sensor nodes. Therefore, it is important to calculate the number of the sensor nodes' neighbors correctly. However, in AdapCode the nodes can only find their full active neighbors, and in the case that their neighbors do not send any message, the number of neighbors cannot be calculated correctly. To solve this problem, the authors in [64] propose an energy-efficient neighbors discovery method. To make the discovery process efficient, they use network beacons. After running the discovery phase, a similar code dissemination phase to the AdapCode approach will be run to deliver the packets to the sensor nodes.

The R-Code approach, proposed in [65], uses network coding to provide reliable broadcast in wireless mesh networks with unreliable links. R-code uses ETX value of the links as their weights, and constructs a minimum spanning tree. In contrast with the AdapCode approach, in R-code only the non-leaf nodes in the spanning tree are the relay nodes. Each parent node is responsible for delivering a sufficient number of linearly coded packets to its children nodes. The parent node stops sending more packets after receiving acknowledgment messages from all of its children nodes. Similar to AdapCode, R-code can be used in WSNs to send code updates from sink node to the sensor nodes.

The DutyCode approach, which combines network coding with duty-cycling is proposed in [66]. Duty-cycling is a technique for saving energy in WSNs. In this scheme, the nodes turn off part or all of their systems for periods of time. Network coding and duty-cycling achieve energy efficiency through conflicting means. Network coding saves energy by exploiting the broadcast nature of the medium and overhearing, whereas duty-cycling saves energy by reducing idle listening, which reduces overhearing. The authors in [66] address the combination of these technics in flooding-based WSNs applications, such as code dissemination applications that require a non-negligible amount of time, possibly tens of minutes in large scale WSNs.

The main idea in DutyCode is that due to the redundancy of coding, in some periods of time a sensor node does not benefit from overhearing coded packets. The goal of the authors is to determine these periods of time, and let sensor nodes that do not benefit from these useless packets, to go to the sleep mode. DutyCode is a cross layer method. In this approach, the MAC layer provides streaming, random sleeping and synchronization facilities. On the other hand, the proposed network coding-aware application layer uses information from the stream being transmitted to determine the time to sleep and its duration. In DytyCode approach, the network

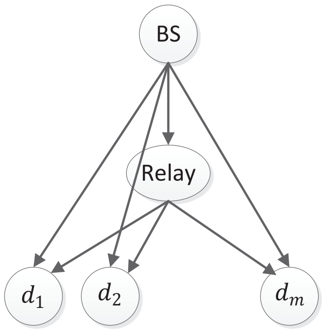

Fig. 1.18 Relay-aided broadcasting.

coding application specifies the sleep duration when it requests the node to go to the sleep mode. Then, the MAC protocol turns off the sensors radio for the requested duration if there is no pending transmission. The MAC protocol does not put the sensor node in the sleeping mode periodically. When requested, and if feasible, it shuts down the sensor's radio for the requested period.

1.5.3 Summary and Discussion

The proposed inter-sessions network coding approach in [43] is a distributed method that relies only on local 2-hop information. It is an appropriate scheme for the case of a delay-tolerant network. However, in the case of applications with deadline constraints, the relay nodes forward the received packets immediately, and the relay nodes do not postpone the transmission of the received packets to receive more packets. In order to increase the coding opportunities at the relay nodes, the authors in [49] proposed three methods to compute waiting time at the relay nodes; this is to assure that these waiting times do not result in any deadline misses. The main drawback of this work is that it is assumed that the nodes have multi-channel multi-radio capabilities. The work in [50] uses broadcasting trees to periodically disseminate the source packets to all the other nodes in the network, and performs random linear network coding at the intermediate nodes. In [48], relay nodes transmit the packets in an order which decreases the packet delay. It is assumed that there is a deadline to transmit a packet at each relay node. However, it is not clear how the deadlines can be calculated.

The work in [45] uses the advantage of directional antennas to reduce the energy consumption of relay nodes. The authors use a directional connected dominant set to find relay nodes. Inter-session network coding is used at relay nodes to reduce the number of required transmissions. In [52], random linear network coding is combined with a probabilistic forwarding approach. The performance of this approach is highly dependant on the computed forwarding factor of the relay nodes. In this scheme, overestimating the forwarding factor results in unnecessary redundant transmissions. On the other hand, by underestimating the forwarding factor, the nodes will not be able to decode the packets due to receiving insufficient number of coded packets.

The approaches in [53, 54, 55] are proposed for one-hop reliable broadcasting applications. In these methods, the source node uses intra-session network coding to retransmit the missed packets by different destination nodes. As the problem of efficient reliable broadcasting is NP-complete, all of the proposed approaches are heuristic algorithms. The work in [55] is extended in [63] to use the advantage of relay node for one-hop broadcasting applications. Triangular network coding is uses in [57] in order to increase the efficiency of multi-layer video broadcasting to a set of client nodes over single-hop error-prone wireless networks.

The work in [10, 64, 65, 66] are proposed specifically for WSNs. In AdapCode, the sensor nodes use their neighbors density to adopt the number of packets coded together (segment size). In order to increase the efficiency of Adapcode, the authors in [64] use network beacons to propose an energy-efficient neighbors discovery method. The R-Code approach [65] uses ETX metric to construct a minimum spanning tree, and the non-leaf nodes in the tree keep sending linear coded packets until they receive acknowledgment messages from all of their children nodes. Duty-Code [66] combines network coding with duty-cycling to increase the battery conservation.

Table 1.3 classifies the discussed methods for broadcast application, based on the used methodology. This table also shows the objective of the approaches and whether they assume the existence of lossy links or perfect links.

1.6 Conclusion

In this chapter, we surveyed recently proposed network coding approaches for wireless networks and WSNs. In general, network coding methods can be classified as inter-session or intra-session network coding approaches. We surveyed some of the proposed inter-session network coding approaches, which allow mixing the packets from different sessions to solve the bottleneck problem. We also reviewed intra-session network coding methods, which use the diversity of the links and mix the packets from the same sessions to solve the packet loss problem. The network coding methods can be applied in unicast, multicast, or broadcast applications. Moreover, some of the network coding approaches have been proposed just for one-hop, two-hop, or multi-hop networks. Therefore, we classified the methods based on their objective, application, and network topology assumption. Some of the proposed approaches specific to WSNs are surveyed, and we argued which of the proposed

Approach	**Methodology**	**Topology**	**Objective**	**XOR or RL**	**Local or Global**	**Links**
CODEB	Inter-session	Multi-hop	Throughput	XOR	Local	Perfect
[45]	Inter-session	Multi-hop	Transmissions	XOR	Local	Perfect
[48]	Inter-session	Multi-hop	Throughput, Deadline	XOR	Local	Perfect
[49]	Inter-session	Multi-hop	Throughput, Deadline	XOR	Local	Perfect
[50]	Inter-session	Multi-hop	Throughput, Deadline	RL	Global	Perfect
[52]	Inter-session	Multi-hop	Energy efficiency	RL	Global	Lossy
[53]	Intra-session	One-hop	Transmissions	XOR	Local	Lossy
[54]	Intra-session	One-hop	Transmissions	XOR	Local	Lossy
[55]	Intra-session	One-hop	Transmissions	XOR	Local	Lossy
[57]	Intra-session	One-hop	Transmissions	RL (Triangular)	Local	Lossy
[63]	Intra-session	One-hop	Transmissions	XOR	Local	Lossy
[67]	Intra-session	One-hop	Transmissions, Delay	XOR	Local	Lossy
AdapCode [10]	Intra-session	Multi-hop	Transmissions, Reliability	RL	Global	Lossy
R-code [65]	Intra-session	Multi-hop	Transmissions, Reliability	RL	Global	Lossy
DutyCode [66]	Intra-session	Multi-hop	Transmissions, Reliability, and duty-cycle	RL	Global	Lossy

Table 1.3 Classification of the network coding methods for broadcasting.

network coding methods for general wireless networks are applicable and suitable for WSNs.

Acknowledgment

This research was supported in part by NSF grants ECCS 1128209, CNS 1065444, CCF 1028167, CNS 0948184, and CCF 0830289.

References

1. C. Cordeiro, H. Gossain, R. Ashok, and D. Agrawal, "The last mile: Wireless technologies for broadband and home networks," in *Tutorial Presented in the 21th Brazilian Symposium on Computer Networks*, 2003.
2. S. Cherry, "The wireless last mile," *IEEE Spectrum*, vol. 40, no. 9, pp. 18–22, 2003.

3. I. Akyildiz and X. Wang, "A survey on wireless mesh networks," *Communications Magazine, IEEE*, vol. 43, no. 9, pp. S23–S30, 2005.
4. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Raptor codes," *IEEE Communications magazine*, vol. 40, no. 8, pp. 102–114, 2002.
5. D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11 b mesh network," in *ACM SIGCOMM*, 2004, pp. 121–132.
6. S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 243–25, 2006.
7. S. Biswas and R. Morris, "Exor: opportunistic multi-hop routing for wireless networks," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 133–144, 2005.
8. S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *ACM SIGCOMM*, 2007.
9. L. Keller, E. Atsan, K. Argyraki, and C. Fragouli, "SenseCode: Network coding for reliable sensor networks," in *EPFL Technical Report*, 2009.
10. I. Hou, Y. Tsai, T. Abdelzaher, and I. Gupta, "AdapCode: Adaptive network coding for code updates in wireless sensor networks," in *IEEE INFOCOM*, 2008.
11. P. Maymounkov, "Online codes," in *Technical Report TR2002-833, New York University*, 2002.
12. M. Luby, "Lt codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280.
13. A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
14. S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing," in *IEEE INFOCOM*, 2007, pp. 1028–1036.
15. T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *IEEE INFOCOM*, 2008, pp. 361–365.
16. H. Seferoglu and A. Markopoulou, "Network coding-aware queue management for unicast flows over coded wireless networks," in *NetCod*, 2010, pp. 1–6.
17. J. Le, J. Lui, and D. Chiu, "How many packets can we encode?-an analysis of practical wireless network coding," in *IEEE INFOCOM*, 2008, pp. 371–375.
18. S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, "Loss-aware network coding for unicast wireless sessions: Design, implementation, and performance evaluation," *SACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 1, pp. 85–96, 2008.
19. A. Khreishah, J. Wu, P. Ostovari, and I. Khalil, "Flow based xor network coding for lossy wireless networks," in *IEEE GLOBECOM*, 2011.
20. A. Khreishah, I. Khalil, P. Ostovari, and J. Wu, "Flow-based xor network coding for lossy wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 2321–2329, 2012.
21. S. Zhang, S. Liew, and P. Lam, "Hot topic: physical-layer network coding," in *MobiCom*, 2006, pp. 358–365.
22. I. Jawhar, N. Mohamed, and D. P. Agrawal, "Linear wireless sensor networks: Classification and applications," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1671–1682, 2011.
23. X. Zhang and B. Li, "Optimized multipath network coding in lossy wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 622–634, 2009.
24. ——, "Dice: a game theoretic framework for wireless multipath network coding," in *ACM MobiHoc*, 2008.
25. Y. Lin, B. Li, and B. Liang, "Codeor: Opportunistic routing in wireless mesh networks with segmented network coding," in *IEEE ICNP*, 2008.
26. L. Brakmo and L. Peterson, "TCP vegas: End to end congestion avoidance on a global internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, 1995.

27. B. Radunović, C. Gkantsidis, P. Key, and P. Rodriguez, "Toward practical opportunistic routing with intra-session network coding for mesh networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 2, pp. 420–433, 2010.
28. D. Koutsonikolas, C. Wang, and Y. Hu, "CCACK: Efficient network coding based opportunistic routing through cumulative coded acknowledgments," in *IEEE INFOCOM*, 2010, pp. 1–9.
29. K. Srinivasan, M. Jain, J. Choi, T. Azim, E. Kim, P. Levis, and B. Krishnamachari, "The κ factor: inferring protocol performance using inter-link reception correlation," in *ACM MobiCom*, 2010, pp. 7317–328.
30. A. Khreishah, I. Khalil, and J. Wu, "Universal opportunistic routing scheme using network coding," in *IEEE SECON*, 2012.
31. S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 401–412, 2008.
32. È. Gabidulin, "Theory of codes with maximum rank distance," *Problemy Peredachi Informatsii*, vol. 21, no. 1, pp. 3–16, 1985.
33. M. Li, Z. Yang, and W. Lou, "Codeon: Cooperative popular content distribution for vehicular networks using symbol level network coding," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 223–235, 2011.
34. Z. Yang, M. Li, and W. Lou, "CodePlay: Live multimedia streaming in vanets using symbol-level network coding," *IEEE Transactions on Wireless Communications*, vol. 11, no. 8, pp. 3006–3013, 2012.
35. C. Qin, Y. Xian, C. Gray, N. Santhapuri, and S. Nelakuditi, "I^2MIX: Integration of intra-flow and inter-flow wireless network coding," in *IEEE SECON Workshops*, 2008, pp. 1–6.
36. H. Seferoglu, A. Markopoulou, and K. Ramakrishnan, "I2NC: Intra-and inter-session network coding for unicast flows in wireless networks," in *IEEE INFOCOM*, 2011, pp. 1035–1043.
37. C. Wang, A. Khreishah, and N. Shroff, "Cross-layer optimizations for intersession network coding on practical 2-hop relay networks," in *Asilomar*, vol. 41, 2009, pp. 771–775.
38. A. Khreishah, I. Khalil, and J. Wu, "Polynomial time and provably efficient network coding scheme for lossy wireless networks," in *IEEE MASS*, 2011.
39. A. Khreishah, I. Khalil, , and J. Wu, "Low complexity and provably efficient algorithm for joint inter and intrasession network coding in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, 2012.
40. H. Holbrook, S. Singhal, and D. Cheriton, "P. fan and c. zhi and c. wei and k. ben letaief," *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 4, pp. 328–341, 2005.
41. D. Koutsonikolas, Y. Hu, and C. Wang, "Pacifier: High-throughput, reliable multicast without"crying babies"in wireless mesh networks," in *IEEE INFOCOM*, 2009, pp. 2473–2481.
42. A. Khreishah, I. Khalil, and J. Wu, "Distributed network coding-based opportunistic routing for multicast," in *MobiHoc*, 2012, pp. 115–124.
43. L. Li, R. Ramjee, M. Buddhikot, and S. Miller, "Network coding-based broadcast in mobile ad-hoc networks," in *IEEE INFOCOM*, May 2007, pp. 1739–1747.
44. W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," *IEEE Transactions on Mobile Computing*, vol. 1, no. 2, pp. 111– 122, Apr-Jun 2002.
45. S. Yang and J. Wu, "Efficient broadcasting using network coding and directional antennas in MANETs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 148–161, Feb 2010.
46. S. Yang, J. Wu, and F. Dai, "Efficient backbone construction methods in manets using directional antennas," in *ICDCS*, 2007.
47. Z. Dong, C. Zhan, and Y. Xu, "Delay aware broadcast scheduling in wireless networks using network coding," in *IEEE NSWCTC*, 2010, pp. 214–217.
48. C. Zhan and Y. Xu, "Broadcast scheduling based on network coding in time critical wireless networks," in *IEEE International Symposium on Network Coding*, June 2010.
49. P. Ostovari, J. Wu, and A. Khreishah, "Deadline-aware broadcasting in wireless networks with local network coding," in *IEEE ICNC*, Jan 2012.

50. P. Ostovari, A. Khreishahand, and J. Wu, “Deadline-aware broadcasting in wireless networks with network coding,” in *IEEE GLOBECOM*, Dec 2012.
51. C. Fragouli, J. Widmer, and J. L. Boudec, “A network coding approach to energy efficient broadcasting: From theory to practice,” in *IEEE INFOCOM*, 2006, pp. 1–11.
52. ——, “Efficient broadcasting using network coding,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 450–463, Apr 2008.
53. D. Nguyen, T. Tran, T. Nguyen, and B. Bose, “Wireless broadcast using network coding,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 914–925, 2009.
54. L. Lu, M. Xiao, M. Skoglund, L. Rasmussen, G. Wu, and S. Li, “Efficient network coding for wireless broadcasting,” in *IEEE WCNC*, 2010, pp. 1–6.
55. W. Fang, F. Liu, Z. Liu, L. Shu, and S. Nishio, “Reliable broadcast transmission in wireless networks based on network coding,” in *IEEE INFOCOM Workshops (INFOCOM WKSHPS)*, 2011, pp. 555–559.
56. D. Welsh and M. Powell, “An upper bound for the chromatic number of a graph and its application to timetabling problems,” *The Computer Journal*, vol. 10, no. 1, pp. 85–86, 1967.
57. D. Koutsonikolas, Y. Hu, C. Wang, M. Comer, and A. Mohamed, “Efficient online WiFi delivery of layered-coding media using inter-layer network coding,” in *ICDCS*, 2011, pp. 237–247.
58. U. Horn, K. Stuhlmüller, M. Link, and B. Girod, “Robust internet video transmission based on scalable coding and unequal error protection,” *Signal Processing: Image Communication*, vol. 15, no. 1, pp. 77–94, 1999.
59. D. Wu, Y. Hou, and Y. Zhang, “Scalable video coding and transport over broadband wireless networks,” *Proceedings of the IEEE*, vol. 89, no. 1, pp. 6–20, 2001.
60. A. Majumda, D. Sachs, I. Kozintsev, K. Ramchandran, and M. Yeung, “Multicast and unicast real-time video streaming over wireless lans,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 524–534, 2002.
61. D. Koutsonikolas, Y. Hu, C. Wang, M. Comer, and A. Mohamed, “On the performance of network coding in multi-resolution wireless video streaming,” in *NetCod*, 2010, pp. 1–6.
62. M. Halloush and H. Radha, “Practical network coding for scalable video in error prone networks,” in *PCS*, 2009, pp. 1–4.
63. L. Lu, M. Xiao, and L. Rasmussen, “Relay-aided broadcasting with instantaneously decodable binary network codes,” in *ICCCN*, 2011, pp. 1–5.
64. H. Shwe and F. Adachi, “Power efficient adaptive network coding in wireless sensor networks,” in *IEEE ICC*, 2011, pp. 1–5.
65. Z. Yang, M. Li, and W. Lou, “R-code: Network coding based reliable broadcast in wireless mesh networks with unreliable links,” in *IEEE GLOBECOM*, 2009.
66. R. Chandanala and R. Stoleru, “Network coding in duty-cycled sensor networks,” in *INSS*, 2010, pp. 203–210.
67. S. Sorour and S. Valaee, “Minimum broadcast decoding delay for generalized instantly decodable network coding,” in *IEEE GLOBECOM*, 2010, pp. 1–5.