

## Research Article

# CARTA: Coding-Aware Routing via Tree-Based Address

Xingyu He<sup>1</sup> and Guisong Yang<sup>2,3</sup> 

<sup>1</sup>College of Communication and Art Design, University of Shanghai for Science and Technology, Shanghai 200093, China

<sup>2</sup>Department of Computer Science and Engineering, Shanghai Key Lab of Modern Optical System, University of Shanghai for Science and Technology, Shanghai 200093, China

<sup>3</sup>Department of Electrical and Computer Engineering, Michigan State University, East Lansing 48823, USA

Correspondence should be addressed to Guisong Yang; [gsyang@usst.edu.cn](mailto:gsyang@usst.edu.cn)

Received 31 August 2019; Revised 9 June 2020; Accepted 6 July 2020; Published 20 July 2020

Academic Editor: Rüdiger C. Pryss

Copyright © 2020 Xingyu He and Guisong Yang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network coding-aware routing has become an effective paradigm to improve network throughput and relieve network congestion. However, to detect coding opportunities and make routing decision for a data flow, most existing XOR coding-aware routing methods need to consume much overhead to collect overhearing information on its possible routing paths. In view of this, we propose low-overhead and dynamic Coding-Aware Routing via Tree-based Address (CARTA) for wireless sensor networks (WSNs). In CARTA, a Multi-Root Multi-Tree Topology (MRMTT) with a tree-based address allocation mechanism is firstly constructed to provide transmission paths for data flows. Then, a low-overhead coding condition judgment method is provided to detect real-time coding opportunities via tree address calculation in the MRMTT. Further, CARTA defines routing address adjustments caused by encoding and decoding to ensure the flows' routing paths can be adjusted flexibly according to their real-time coding opportunities. It also makes additional constraints on congestion and hop count in the coding condition judgment to relieve network congestion and control the hop counts of routing paths. The simulation results verify that CARTA can utilize more coding opportunities with less overhead on coding, and this is ultimately beneficial for promoting network throughput and balancing energy consumption in WSNs.

## 1. Introduction

Network coding has attracted much interest in wireless network applications, because it can improve network throughput and relieve network congestion [1, 2]. It can also be used to lower network energy consumption in IOT networks [3]. It allows that flows are encoded into an encoded flow in their intersecting node as long as their destination nodes can retrieve them from the encoded flow separately. To take the advantages of network coding, network coding-aware routing has been proposed in wireless networks [4]. XOR coding is one of the simplest but effective interflow network coding schemes and thus widely used in the existing network coding-aware routing methods [5–7].

Figure 1 shows a XOR network coding scenario in a Cluster-Tree Topology (CTT) in WSNs. In Figure 1,  $C_1$  is

the Cluster Head (CH) of  $D_2$ , and  $C_2$  is the CH of  $S_1$ . Similarly,  $C_5$  is the CH of  $D_1$ ,  $C_4$  is the CH of  $S_2$ , and there is a backbone tree to connect all CHs and a root  $C_3$ . To find the coding opportunity between flow  $a$  from  $S_1$  to  $D_1$  and flow  $b$  from  $S_2$  to  $D_2$  at  $C_3$ , most existing XOR coding-aware routing methods require that, before the two flows are routed, their source nodes  $S_1$  and  $S_2$  send routing requests along all of their possible routing paths to collect the overhearing information of the nodes on these paths, and their destination nodes  $D_1$  and  $D_2$  send routing replies to inform the nodes on these paths of the collected overhearing information. Obviously, the collection and transmission of the overhearing information cost significant overhead.

Furthermore, in most existing XOR coding-aware routing methods, before a flow is routed, its routing path is decided according to the overhearing information collected before-

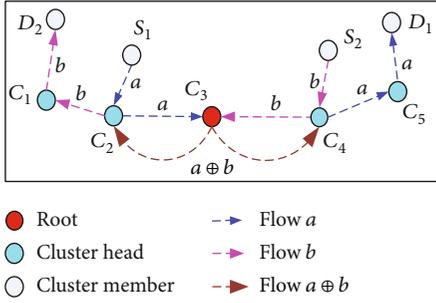


FIGURE 1: A scenario of XOR network coding in CTT.

hand and cannot be changed in its transmission process according to its real-time coding opportunities, which incurs it may miss some real-time coding opportunities.

Thus, two important questions emerge: whether does there exist a low-overhead method to detect real-time XOR coding opportunities and how to flexibly adjust the routing paths of flows according to their real-time XOR coding opportunities? To solve these questions, we introduce the tree-based address allocation mechanism defined in the ZigBee standards [8] and combine it with the XOR network coding. For instance, in Figure 1, if each node is assigned a tree address and  $C_3$  has the tree addresses of  $C_2$ ,  $C_4$ ,  $S_1$ ,  $S_2$ ,  $D_1$ , and  $D_2$  on the backbone tree, via tree address calculation, it can know  $C_2$  is a descendant of  $C_3$  and the parent of  $S_1$  but an ancestor of  $D_2$  and  $C_4$  is a descendant of  $C_3$  and the parent of  $S_2$  but an ancestor of  $D_1$  on the backbone tree, then it can infer two facts: (1) flow  $a \oplus b$  transited long the backbone tree will certainly pass  $C_2$  (the CH of  $S_1$ ) before reaching  $D_2$  and will certainly pass  $C_4$  (the CH of  $S_2$ ) before reaching  $D_1$  and (2)  $C_2$  and  $C_4$  can decode flow  $a \oplus b$  by overhearing flow  $a$  and flow  $b$  from  $S_1$  and  $S_2$ , respectively. Therefore, it can detect the coding opportunity between flow  $a$  and flow  $b$  easily via the tree address calculation instead of via the collection of overhearing information.

In this paper, we propose low-overhead and dynamic Coding-Aware Routing via Tree-based Address (CARTA) for WSNs. In CARTA, a Multi-Root Multi-Tree Topology (MRMTT) with the tree-based address allocation mechanism is firstly constructed to provide transmission paths for data flows; then, real-time coding opportunities between the flows in the MRMTT are detected via tree address calculation, and the flows' routing paths are adjusted flexibly according to their real-time coding opportunities via routing address adjustments. The main contributions of our work are summarized as follows:

- (1) Firstly, a Multi-Root Multi-Tree Topology (MRMTT) is constructed, which uses the tree-based address allocation mechanism and is capable of providing transmission paths and coding opportunities for data flows
- (2) Secondly, a general coding condition is defined for both original flows and encoded flows, and accordingly, a low-overhead coding condition judgment method is provided to detect real-time coding opportunities between flows via tree address calculation

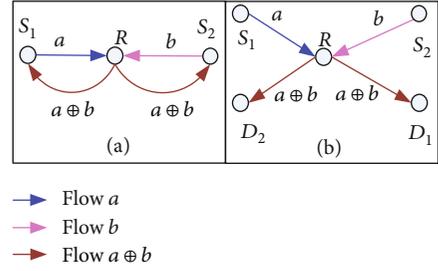


FIGURE 2: XOR network coding in the chain structure (a) and X structure (b).

- (3) Thirdly, a coding-aware routing algorithm is designed. In this algorithm, routing address adjustments caused by encoding and decoding are defined to ensure flows' routing paths can be adjusted flexibly, and additional constraints on congestion and hop count are defined in the coding condition judgment to relieve network congestion and control the hop counts of routing paths

The rest of this paper is organized as follows: Section 2 gives the related work. Section 3 defines the network topology of CARTA. Details of encoding and decoding in CARTA are discussed in Section 4. Routing algorithm execution in CARTA is presented in Section 5. Performance of CARTA is evaluated in Section 6, and this work is concluded in Section 7.

## 2. Related Work

As an effective method to improve network throughput and relieve network congestion, network coding technology has attracted much concern in wireless network applications [9, 10], and it can be classified into interflow network coding and intraflow network coding. Interflow network coding is performed between different flows, and XOR coding [5] is one of the simplest but effective interflow network coding schemes, which is utilized in this paper to improve the network performance of WSNs.

**2.1. XOR Coding Structure.** The basic idea of XOR network coding is illustrated in Figure 2: the chain structure scenario and the X structure scenario are shown in Figure 2(a) and Figure 2(b), respectively. In both scenarios, flow  $a$  from  $S_1$  and flow  $b$  from  $S_2$  are encoded into flow  $a \oplus b$  at  $R$ , and then flow  $a \oplus b$  is transmitted to destination nodes  $D_1$  and  $D_2$  simultaneously, so the number of transmissions is reduced from 2 to 1. The difference is that the destination nodes in the X structure decode flow  $a \oplus b$  by overhearing flow  $a$  and flow  $b$  from  $S_1$  and  $S_2$ , respectively. As shown in Figure 2, the network performance is improved with less energy consumption and congestion due to the reduced number of transmissions.

The above XOR network coding structures are in a two-hop region. To apply the XOR coding structures in a two-hop region, authors of [11, 12] decompose a wireless network into a superposition of simple relay networks called two-hop

relay networks. In [11], the capacity region of the two-hop relay network is characterized to achieve an upper bound when the coding operations are limited to XOR. In [12], a polynomial time coding scheme is proposed for the two-hop relay network, in which the random network coding is used to carefully mix intra and intersession network coding and make a linear, not exponential, number of decisions. Authors of [13] provide a  $k$ -tuple XOR network coding structure in a two-hop region and a generalization of pairwise coding with next-hop decoding ability.

To break through the coding structures in a two-hop region, DCAR [6] proposes a  $k$ -hop ( $k > 2$ ) model (shown in Figure 3), in which, the overhearing and decoding for flow  $a \oplus b$  occur multiple hops away from its encoding node  $R$  and the coding structure formed in this model is called the generalized coding structure. FORM [7] and NCRT [14] are also typical network coding methods that support the generalized coding structure. FORM [7] makes adjustments to the coding conditions of DCAR, so that not only the original flows but also the encoded flows can be involved in the coding opportunity detection, which means FORM supports multilayer coding. NCRT [14] improves the weakness of the generalized coding conditions of FORM to further increase the coding opportunities. In this paper, CARTA also supports the generalized coding structure and multilayer coding.

**2.2. Coding-Aware Routing.** In addition to relieving network congestion, network coding technology also has a great effect on reducing network energy consumption [15] and prolonging network life [16]. Thus, it has been used in routing protocols. The routing protocols based on network coding are called coding-aware routing protocols.

A basic work in coding-aware routing is to create more reliable coding opportunities. In [17, 18], the virtual overhearing (VOH) technique is proposed to increase coding opportunities and enable network coding when the side information cannot be obtained from the traditional overhearing. CORE [19] integrates interflow network coding with opportunistic forwarding to increase the coding opportunities in the network. Moreover, to enable more encoded packets and achieve higher coding gain, CANCAR [20] diverts some of the least coded traffic flows to alternative routes and releases the most loaded parts of the network from poorly coded traffic. Authors of [21] provide a simple XOR-assisted cooperative diversity scheme called XOR-CD to exploit coding opportunities on bidirectional traffic on the uplink and downlink of a mobile station.

To reduce decoding failures is also very important in coding-aware routing. FlexONC [22] utilizes a switch rule and an ACK strategy to limit decoding failures and allows nonintended forwarders to help in decoding, encoding, and forwarding. To promote the reliability of coding opportunities and deal with the coding collision problem effectively, CFCR [23] also introduces the information confirmation process to decrease the failure ratio of decoding. In [24], a principle called consistency of encoding and overhearing is proposed and adhered with DCAR to help the encoding node to avoid misidentifying coding opportunities and ensure the

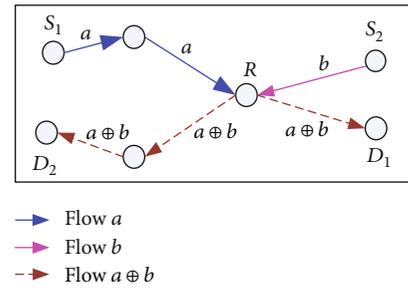


FIGURE 3: XOR network coding in the generalized structure.

successful decoding of all encoded packets. In [25], a network coding framework called encoded packet-assisted retransmission is proposed, in which receivers store all heard packets that cannot be decoded and report the reception status to the sender periodically.

Another important work in coding-aware routing is using routing metrics to evaluate the benefits brought by coding opportunities on paths and make routing decisions. In DCAR [6], the Coding-aware Routing Metric (CRM) is defined based on queue length, which is modified by coding opportunities, to compare coding-possible and coding-impossible paths. The routing metrics in FORM [7] consist of the modified benefit and the degree of free ride. Specifically, the modified benefit is computed according to the difference between the gain and the loss in hop count for sending a packet on the considered path, and the degree of free ride reflects the abundance of coding opportunities in the considered path. NCRT [14] proposes a new routing metric that considers both coding opportunities and network workload. Authors of [26] present and define the coding conditions to identify a coding host, then estimate the bandwidth consumption of a coding host under the contention-based wireless networks with a random access mechanism and propose a bandwidth-satisfied and coding-aware multicast routing protocol.

In static coding-aware routing protocols, routing metrics and decisions are attained according to the overhearing information collected beforehand and remain unchanged. Thus, they consume much overhead on collection of overhearing information and have low adaptability and flexibility in dynamic networks. To cope with the dynamics of the network, CAR [27] selects routes based on real-time coding opportunities. In NCAR [28], to promote the throughput of time-varying networks, each source requires some of its sent packets to piggyback their overhearing information periodically to detect better coding opportunities and make better routing decisions for its packets to be sent. However, the dynamic updating and periodical collection of overhearing information in [27, 28] both bring significant overhead increase.

To detect real-time coding opportunities without the collection of overhearing information and dynamically adjust routing paths of flows according to their real-time coding opportunities, we aim to provide a dynamic and low-overhead coding-aware routing protocol via the tree-based address for WSNs in this paper.

### 3. Network Topology of CARTA

To bring more transmission paths and coding opportunities, we construct a special CTT: the Multi-Root Multi-Tree Topology (MRMTT) for WSNs. In the MRMTT, CHs are elected periodically, and multiple decentralized nodes are predetermined as roots that are responsible for starting the formation process of the corresponding multiple backbone trees. Each of the backbone trees is required to include and connect all CHs and roots. For example, Figure 4 shows an MRMTT that has three backbone trees formed from roots  $N_1$ ,  $N_{11}$ , and  $N_{15}$ , respectively. There are two phases to form an MRMTT: the cluster formation phase and backbone tree formation phase.

**3.1. Cluster Formation.** In each round of CH election in the MRMTT, each node generates a random number in the range of 0 to 1 and calculates an election threshold by the same method used in LEACH [29]. The nodes whose random numbers are under the threshold are elected as CHs automatically. Each of the remaining nodes needs to join the cluster of its closest CH and then becomes a Cluster Member (CM) of the cluster.

After the CH election, once a CH finds that it has no adjacent CH in its communication range, it will broadcast a bridge node request to all CMs in its communication range no matter whether they are in or out of its cluster. Then, it will choose the one among them which has the highest residual energy and can communicate with another CH, as a bridge node. In the following phase, the chosen bridge nodes are treated like CHs so that all the elected CHs can be connected.

**3.2. Backbone Tree Formation.** In the formation process of backbone trees in the MRMTT, multiple decentralized and predetermined roots start forming multiple backbone trees successively in a predefined order (e.g., clockwise or counter-clockwise, we use clockwise in Section 6). In the formation of each backbone tree, a root is predefined as the starting root. It broadcasts a signal to inform each of the other nodes to find a parent and then connect with it, and the other roots are treated like CHs. In order to create more transmission paths, while forming a new backbone tree, each CH preferentially finds its neighbor CHs that have not been connected with it directly in the already-formed backbone trees and chooses the one among them which is closest to the starting root of the new backbone tree, as its parent node. In case all of its neighbor CHs have already been connected with it directly, it will choose the neighbor CH which is closest to the starting root of the new backbone tree, as its parent node. Meanwhile, each CM chooses its own CH as its parent to make itself a leaf of the new backbone tree. For example, in Figure 4, Tree 1 and Tree 2 have already been constructed with  $N_1$  and  $N_{11}$  as their starting roots, respectively. In the construction process of Tree 3,  $N_2$  chooses  $N_3$  as its parent, because  $N_3$  has not been connected with it directly in the previously formed trees.  $N_7$  chooses  $N_8$  as its parent, because all of its neighbor CHs have already been connected with it directly, and  $N_8$  is the neighbor CH which is closest to the current starting root  $N_{15}$ .

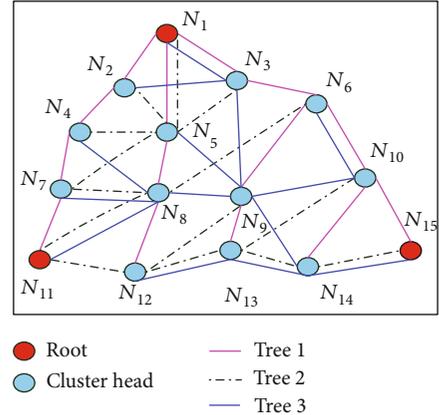


FIGURE 4: Illustration of the MRMTT.

Backbone trees in the MRMTT use the tree-based address allocation mechanism defined in the ZigBee standards [8], but their starting roots have different starting addresses. This means the backbone trees have different address spaces, so each node in the MRMTT has multiple addresses assigned by different backbone trees and in different address spaces. For instance, in Figure 4, there are three backbone trees, so each node has three different addresses that are assigned by Tree 1, Tree 2, and Tree 3, respectively. Utilizing the tree address, coding opportunities brought by the MRMTT can be detected conveniently on each backbone tree, which will be discussed in the following sections.

### 4. Coding and Decoding in CARTA

**4.1. Coding Condition in the MRMTT.** We call the flows that have not been encoded as original flows and the other flows as encoded flows. We define the coding between two original flows as one-layer coding and define the coding between an encoded flow and another flow (original or encoded one) as multilayer coding.

**4.1.1. One-Layer Coding.** According to Figure 3, two original flows  $a$  and  $b$  can be encoded into an encoded flow  $a \oplus b$  at node  $R$ , if there exists a node on the downstream routing path of flow  $a \oplus b$  from  $R$  to  $D_1$  (the dotted line from  $R$  to  $D_1$ ) which can retrieve flow  $a$  from flow  $a \oplus b$  and if there exists a node on the downstream routing path of flow  $a \oplus b$  from  $R$  to  $D_2$  (the dotted line from  $R$  to  $D_2$ ) which can extract flow  $b$  from flow  $a \oplus b$ . In fact, the neighbor nodes of flow  $a$ 's source node  $S_1$  are capable of extracting flow  $b$  from the encoded flow  $a \oplus b$  because they can overhear flow  $a$  from  $S_1$ . The nodes on the upstream routing path of flow  $a$  from  $S_1$  to  $R$  (the blue solid line) and their neighbors can also retrieve flow  $b$  from the encoded flow  $a \oplus b$  as long as they still keep flow  $a$  in their storage. For a flow, the nodes which can retrieve the other flow involved in its encoded flow are called as its decoding-capable nodes; these nodes are also decoding node candidates for the other flow. This means, for a flow, its source node, nodes on its upstream routing path, and the neighbor nodes of these two kinds of nodes are its decoding-capable nodes.

Therefore, in the MRMTT, each node is required to store the flows it has forwarded for a predefined period of time. It must also maintain a decoding-capable node list that records its neighbor CHs in its communication range with their corresponding addresses and itself with its own addresses. For instance, Table 1 shows a decoding-capable node list of a node in an MRMTT having three backbone trees (Tree 1, Tree 2, and Tree 3). In this list, each decoding-capable node is recorded with all of its addresses assigned by the three backbone trees, respectively.

We also require that each flow is transmitted with its source node's decoding-capable node list as its initial decoding-capable node list. Then, once it arrives at a new relay node, this relay node's decoding-capable node list will be added into its own list, because these addresses are crucial for detecting its coding opportunities. With the help of the decoding-capable node list, our one-layer coding condition in the MRMTT is defined as follows.

*Coding condition in the MRMTT:* the coding condition for two original flows  $F_1$  and  $F_2$  which encounter each other at node  $X$  in the MRMTT is that there exists a tree path from  $X$  to the destination node of  $F_1$  that passes through a node in the decoding-capable node list of  $F_2$ , and there exists a tree path from  $X$  to the destination node of  $F_2$  that passes through a node in the decoding-capable node list of  $F_1$ .

How to judge the coding condition between two flows in the MRMTT via tree address calculation will be discussed in Section 4.2. For an encoded flow, the node which has performed encoding to it is called its encoding node, and the node that is designated to decode it is called its decoding node. In the MRMTT, an encoded flow's encoding node is responsible to determine its routing paths and designate its decoding nodes, which will be discussed in Section 4.3.

**4.1.2. Multilayer Coding.** Here, a scenario is shown in Figure 5 to illustrate multilayer coding in the MRMTT. In Figure 5, flow  $a$  and flow  $b$  have been encoded into flow  $a \oplus b$  by  $C_1$ , because the defined one-layer coding condition has been satisfied between flow  $a$  from  $S_1$  to  $D_1$  and flow  $b$  from  $S_2$  to  $D_2$  at  $C_1$ . Specifically,  $C_4$  included in  $DC(a)$  is on a tree path from  $C_1$  to  $D_2$  and  $C_3$  included in  $DC(b)$  is on a tree path from  $C_1$  to  $D_1$  ( $DC(a)$ ,  $DC(b)$ , and  $DC(c)$  denote the decoding-capable node lists of flow  $a$ , flow  $b$ , and flow  $c$ , respectively). And  $C_1$  is the encoding node of flow  $a \oplus b$ , and  $C_3$  and  $C_4$  have been designated as the two decoding nodes of flow  $a \oplus b$ .

We notice that, for the encoded flow  $a \oplus b$ , if its encoding node  $C_1$  has been taken as its temporary source node, its decoding nodes  $C_3$  and  $C_4$  have been taken as its temporary destination nodes, and the decoding-capable node list of  $C_1$  has been taken as its temporary initial decoding-capable node list (denoted by  $DC(a \oplus b)$ ) and been updated on the routing path sections from  $C_1$  to  $C_3$  and  $C_4$ ; it can be treated like a new original flow in its following coding opportunity detection on the routing path sections from  $C_1$  to  $C_3$  and  $C_4$  because of these temporary information. For instance, in Figure 5, while the encoded flow  $a \oplus b$  faces its multilayer coding opportunity detection at  $C_2$ , it can be treated like a new original flow whose temporary source and destination

TABLE 1: Addresses in a decoding-capable node list.

Decoding-capable nodes	$N_1$	$N_2$	$N_3$
Addresses assigned by tree 1	$A_{1,T1}$	$A_{2,T1}$	$A_{3,T1}$
Addresses assigned by tree 2	$A_{1,T2}$	$A_{2,T2}$	$A_{3,T2}$
Addresses assigned by tree 3	$A_{1,T3}$	$A_{2,T3}$	$A_{3,T3}$

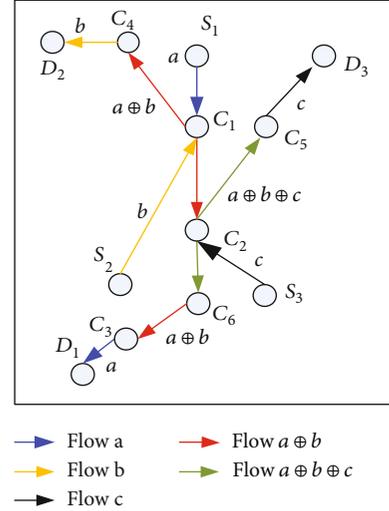


FIGURE 5: Illustration of multilayer coding.

are  $C_1$  and  $C_3$ , respectively. And if the defined one-layer coding condition is satisfied between the encoded flow  $a \oplus b$  from  $C_1$  to  $C_3$  and flow  $c$  from  $S_3$  to  $D_3$ , that is, if  $C_6$  included in  $DC(c)$  is on a tree path from  $C_2$  to  $C_3$  and  $C_5$  included in  $DC(a \oplus b)$  is on a tree path from  $C_2$  to  $D_3$ , the encoded flow  $a \oplus b$  and flow  $c$  can be encoded into flow  $a \oplus b \oplus c$  by  $C_2$ .

According to the discovery in Figure 5, in the MRMTT, each encoded flow is required to carry a coding-decoding node list to record all of its encoding nodes and corresponding decoding nodes with their addresses. For each encoded flow, the latest recorded encoding and decoding nodes in its coding-decoding node list are used as its temporary source and destination, respectively. Further, the decoding-capable node list of its temporary source node is used as its temporary initial decoding-capable node list and updated on its following relay nodes. Once it arrives at one of its decoding nodes, the decoding node and the corresponding encoding node will be cancelled from its coding-decoding node list.

By utilizing the temporary source and destination, the defined coding condition can be used as a general coding condition to detect both one-layer and multilayer coding opportunities in the MRMTT. In the MRMTT, the usage of temporary source and destination can also ensure that a flow's next time of multilayer coding and corresponding decoding happen before it reaches the decoding node of its last time of coding; thus, a flow's next time of multilayer coding is certainly decoded earlier than its last time of coding. For instance, in Figure 6, the sequence of a flow's three times of decoding is exactly opposite to that of its three times of coding.

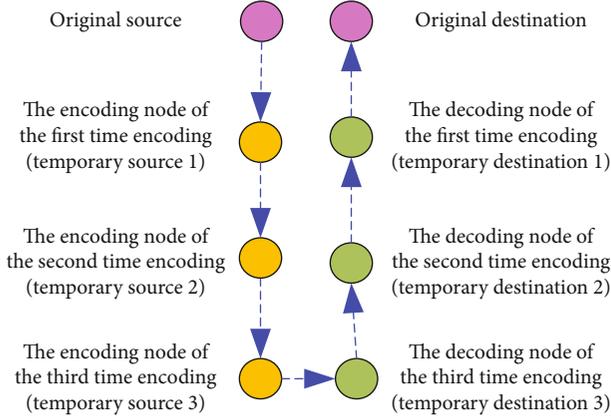


FIGURE 6: The whole routing path of a flow in the MRMTT.

**4.2. Coding Condition Judgement in the MRMTT.** On the surface, to judge the defined coding condition in the MRMTT is a complex process, because it needs to be determined whether or not there exists a tree path from a flow's encoding node to its destination node that passes through one of its decoding node candidates. Due to the tree-based address allocation mechanism in the MRMTT, the judgment can be converted into determining the address relationship of the three nodes (the flow's encoding, decoding and destination nodes) on backbone trees via simple tree address calculation. For instance, in Figure 7, if we know the address relationship of nodes  $N_1$ ,  $N_2$ , and  $N_3$  on backbone trees, we can determine whether or not there exists a tree path from  $N_1$  to  $N_2$  that passes through  $N_3$ .

As shown in Figure 7, if there exists a tree path from  $N_1$  to  $N_2$  that passes through  $N_3$  on a backbone tree, there are only five cases of address relationships for  $N_1$ ,  $N_2$ , and  $N_3$ . Conversely, if it is found that the address relationship of nodes  $N_1$ ,  $N_2$ , and  $N_3$  belongs to one of the five cases on a backbone tree, there must exist a tree path from  $N_1$  to  $N_2$  that can pass through  $N_3$  on the backbone tree. The five cases are (a)  $N_3$  is a descendant of  $N_2$ , and  $N_1$  is a descendant of  $N_3$ ; (b)  $N_3$  is a descendant of  $N_1$ , and  $N_2$  is a descendant of  $N_3$ ; (c)  $N_1$  is a descendant of  $N_3$ , and  $N_2$  is neither a descendant nor an ancestor of  $N_3$ ; (d)  $N_2$  is a descendant of  $N_3$ , and  $N_1$  is neither a descendant nor an ancestor of  $N_3$ ; and (e) both  $N_1$  and  $N_2$  are descendants of  $N_3$ , but  $N_1$  is neither a descendant nor an ancestor of  $N_2$ .

According to the tree-based address allocation mechanism, on a backbone tree, each node can calculate the address space of its descendants. For example, on a backbone tree, if node  $N_1$ 's tree address is denoted by  $A_1$  and the address space of its descendants is denoted by  $Cskip(d(N_1) - 1)$ , the address of any of  $N_1$ 's descendants must be greater than  $A_1$  and less than  $A_1 + Cskip(d(N_1) - 1)$ ,  $d(N_1)$  is  $N_1$ 's network depth and  $Cskip(d(N_1) - 1)$  can be calculated according to

$$Cskip(d(N_1) - 1) = \frac{1 + C_m - R_m - C_m \times R_m^{L_m - (d(N_1) - 1) - 1}}{1 - R_m}, \quad (1)$$

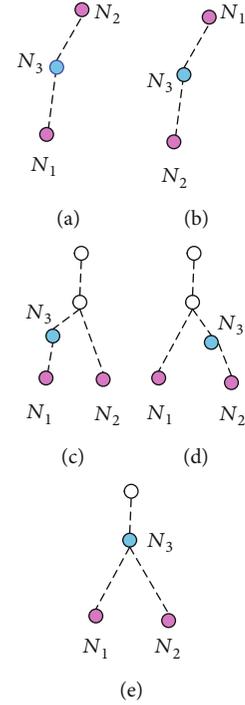


FIGURE 7: The address relationship of three nodes on a backbone tree.

where  $C_m$  is the defined maximum number of a node's child nodes on the backbone tree,  $R_m$  is the defined maximum number of router nodes among a node's child nodes on the backbone tree (the node having a parent node and also allowed to have at least one child node is defined as the router node), and  $L_m$  is the maximum depth of the backbone tree. In other words, on the backbone tree, for any other node  $N_2$  which has tree address  $A_2$ , if  $A_1 < A_2 < A_1 + Cskip(d(N_1) - 1)$ ,  $N_2$  must be a descendant of  $N_1$  and  $N_1$  must be an ancestor of  $N_2$ .

Based on the above analysis, we can determine the address relationship of any three different nodes on a backbone tree according to their tree addresses. Thus, in the coding condition judgment in the MRMTT, there is no need to collect the overhearing information of all relay nodes of the involved flows beforehand, but only the simple tree address calculation is performed in real time. Specifically, while judging the defined coding condition in the MRMTT for two flows, we need to check the address relationship on each backbone tree among their intersecting node, the nodes in their decoding-capable node lists, and their destination nodes.

**4.3. Selection of Routing Path and Decoding Node.** In the coding condition judgment between two flows, it may happen that they have multiple coding opportunities that are brought by tree paths belonging to different backbone trees or brought by the same tree path but involving multiple decoding-capable nodes. Therefore, we require that, in the coding condition judgment of two flows at a relay node, if one of the two flows has multiple decoding-capable nodes involved in their multiple coding opportunities, no matter

whether the coding opportunities are on the same tree path or different tree paths, the decoding-capable node that takes the least hops to the relay node will be chosen as the decoding node which will extract the other flow from their corresponding encoded flow, and the shortest tree path from the relay node to the decoding node will be chosen as one of the next routing path sections of their corresponding encoded flow.

In Figure 8, in the coding condition judgment between flow  $a$  and flow  $b$  at  $X$ , flow  $a$  has two decoding-capable nodes  $C_1$  and  $C_2$  which are involved in two coding opportunities brought by two different tree paths (the tree paths bringing coding opportunities are called routing path candidates). If  $C_2$  can take less hops to  $X$  than  $C_1$ ,  $C_2$  will be chosen as the decoding node that will extract flow  $b$  from flow  $a \oplus b$ , and the shortest tree path from  $C_2$  to  $X$  will be chosen as one of the next routing path sections of flow  $a \oplus b$ . Similarly, flow  $b$  has two decoding-capable nodes  $C_3$  and  $C_4$  which are involved in two coding opportunities brought by the same tree path, and if  $C_3$  can take less hops to  $X$  than  $C_4$ ,  $C_3$  will be chosen as the decoding node that will extract flow  $a$  from flow  $a \oplus b$ , and the shortest tree path from  $C_3$  to  $X$  will be chosen as the other next routing path section of flow  $a \oplus b$ . The hop count between two nodes on a backbone tree can be calculated according to their tree addresses [30].

## 5. Routing Algorithm of CARTA

In this section, we discuss how to execute CARTA in the MRMTT. The tree address is not only the basis of the coding condition judgment but also the crucial information for making routing decisions. To ensure coding and routing are compatible with each other, we firstly define the routing address adjustments caused by encoding and decoding, then present the details of performing CARTA.

**5.1. Routing Address Adjustments.** As shown in Figure 6, the whole routing path of a flow consists of routing path sections which take the flow's (original or/and temporary) sources and destinations as beginnings and ends. On a routing path section of a flow, only the addresses assigned by the backbone tree that provides this path section are used as the routing-related addresses. Each flow is transmitted with a routing address list that records the routing-related addresses on each of its routing path sections. In our routing algorithm, for a flow, in its routing address list, the latest recorded pair of source and destination addresses is defined as its routing addresses and used to make the real-time routing decision.

An original flow chooses the shortest tree path from its original source to its original destination as its initial routing path, and it only records the addresses of its original source and destination that are assigned by the backbone tree that provides the initial routing path in its routing address list, as its routing addresses. For example, in Figure 9, at  $N_1$ , flow  $F_1$  is an original flow, so the routing-related addresses of its original source and destination (denoted by  $S_1^0$  and  $D_1^0$ ) are taken as its routing addresses.

For an encoded flow, once it updates its temporary source and destination and has a new routing path section, it is required to record the addresses of its new temporary source

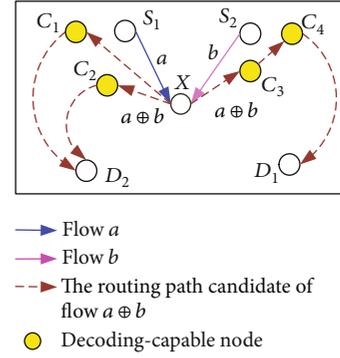


FIGURE 8: Decoding node selection in the MRMTT.

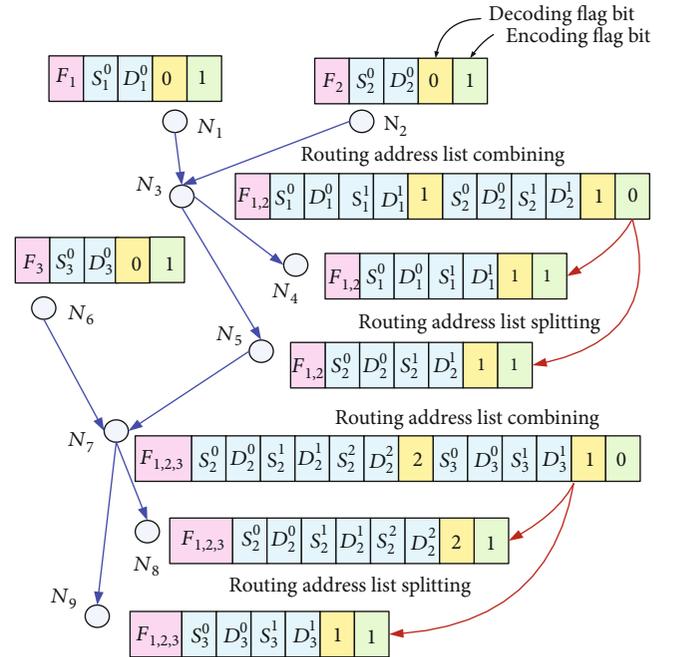


FIGURE 9: Routing address adjustment caused by encoding.

and destination that are assigned by the backbone tree that provides the new routing path section in its routing address list, as its new routing addresses. For example, in Figure 9, at  $N_3$ , flow  $F_1$  and flow  $F_2$  have a coding opportunity. Before they are encoded together by  $N_3$ ,  $N_3$  and their corresponding decoding nodes of the coding opportunity are taken as their temporary sources and destinations. And the routing-related addresses of their temporary sources and destinations (denoted by  $S_1^1$  and  $D_1^1$ ,  $S_2^1$ , and  $D_2^1$ ) are put into their routing address lists separately, as their routing addresses.

A flow's routing addresses can help it to find its next relay node on its routing path section. For instance, for a flow on its current routing path section, the backbone tree providing the current routing path section is defined as its target backbone tree, the node where the flow is currently staying is node  $C$ , and its destination node on the current routing path section is node  $D$ . If the addresses of nodes  $C$  and  $D$  assigned by the target backbone tree are  $A_C$  and  $A_D$ , respectively, and if  $A_C < A_D < A_C + \text{Cskip}(d_C - 1)$  ( $d_C$  is node  $C$ 's

network depth on the target backbone tree), node  $D$  must be a descendant of node  $C$  on the target backbone tree. Therefore, a child of node  $C$  that is also an ancestor of node  $D$  on the target backbone tree will be chosen as the next relay node of the flow, and the address of this next relay node  $A_N$  can be calculated as follows ( $Cskip(d_C)$  can also be calculated according to (1)):

$$A_N = A_C + 1 + \left\lceil \frac{A_D - (A_C + 1)}{Cskip(d_C)} \right\rceil \times Cskip(d_C). \quad (2)$$

Otherwise, the parent of node  $C$  on the target backbone tree will be chosen as the flow's next relay node.

After two flows are encoded together at an intersecting node, two next relay nodes of their encoded flow are on different next routing path sections leading to corresponding decoding nodes (two temporary destinations), and the two next relay nodes can be found by using the routing addresses of the two flows. In order to reduce energy consumption and relieve congestion, the encoded flow is required to be sent from the intersecting node to the two next relay nodes simultaneously with a routing address list that combines the routing address lists of the two flows together, so that the two next relay nodes can extract their needed routing address lists from the combined address list separately. For instance, in Figure 9, at  $N_3$ , after flow  $F_1$  and flow  $F_2$  are encoded into flow  $F_{1,2}$ , the routing addresses of flow  $F_1$  and flow  $F_2$  ( $S_1^1$  and  $D_1^1$ ,  $S_2^1$ , and  $D_2^1$ ) can be used to find the two next relay nodes of  $F_{1,2}$  ( $N_4$  and  $N_5$ ), and flow  $F_{1,2}$  is sent from  $N_3$  to  $N_4$  and  $N_5$  simultaneously with a routing address list which combines the routing address lists of both flow  $F_1$  and flow  $F_2$ . After receiving  $F_{1,2}$ , according to the routing addresses in the combined address list,  $N_4$  can find itself as a relay node on the routing path section of  $F_{1,2}$  to  $D_1^1$  and then split the routing address list related to flow  $F_1$  out of the combined address list. Similarly,  $N_5$  can also extract the routing address list it needs.

After a flow is decoded, its current routing addresses need to be canceled from its routing address list, and the routing-related addresses of its previous temporary source and destination are taken as the routing addresses of the decoded flow. For example, in Figure 10, after  $F_{1,2,3}$  is decoded to  $F_{1,2}$  at  $N_{10}$ ,  $S_2^2$  and  $D_2^2$  are canceled from the routing address list of  $F_{1,2,3}$ , and  $S_2^1$  and  $D_2^1$  are taken as the routing addresses of flow  $F_{1,2}$  ( $S_i^j$  and  $D_i^j$  denote the routing-related addresses of flow  $F_i$  temporary source and destination after its  $j$ th time encoding, respectively).

We define an encoding flag bit (the green blocks in Figures 9 and 10) for each flow to show whether or not it is allowed to be encoded. If a flow's encoding flag bit is 1, it will be allowed to be encoded, and otherwise, forbidden. Specifically, an encoded flow with a combined address list is not allowed to be encoded again until it goes through the routing address list splitting. For instance, in Figure 9,  $F_{1,2}$  at  $N_3$  and  $F_{1,2,3}$  at  $N_7$  are not allowed to be encoded again, and their encoding flag bit is 0. We also define a decoding flag bit (the yellow blocks in Figures 9 and 10) at the end of each

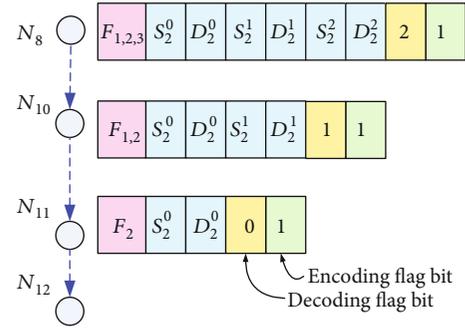


FIGURE 10: Routing address adjustment caused by decoding.

flow's routing address list. A flow's decoding flag bit shows how many times of decoding it needs to become an original flow, and this bit needs to be updated after each encoding and decoding. For instance, in Figure 9, at  $N_8$ ,  $F_{1,2,3}$ 's decoding flag bit shows that it needs twice of decoding to become an original flow. After each time of encoding, this bit is increased by 1; after each time of decoding, it is reduced by 1.

**5.2. Routing Compatible with Encoding and Decoding.** The flow chart of CARTA is shown in Figure 11. In this algorithm, each node has a receiving flow queue and a coding flow queue to store the received flows and the flows to be coded, respectively. The node that is making the routing decision is called the current node. According to Figure 11, after a flow is input to the algorithm from the current node's receiving flow queue, Check1 is executed immediately. In Check1, if the flow's encoding flag bit is 1, it will be allowed to be encoded and go to Check2; otherwise, it will need to go through the routing address list splitting and then go to Check2. In Check2, if the address of the current node is found in the flow's routing address list, it will go to Check3; otherwise, it will be put into the current node's coding flow queue and will wait for Check4. In Check3, if the flow's decoding flag bit is 0, the current node is its original destination node; otherwise, the current node is its temporary destination node as well as the decoding node, and this flow will be decoded with the corresponding routing address list adjustment and then go back to Check2. In Check4, if there is a coding opportunity meeting the constraints on congestion and hop count (these constraints will be discussed in the next section) for the first flow and another flow in the coding flow queue, the first flow and the other flow will be encoded together, and their routing address lists will be combined. Then, the encoded flow will be sent to its two corresponding next relay nodes with the combined address list; otherwise, the first flow in the coding flow queue will be sent to its next relay node directly.

**5.3. Constraints on Congestion and Hop Count.** On a flow's routing path, coding opportunities can bring positive effects such as congestion relief and throughput promotion, but they may also incur detours. For example, in Figure 12, the routing-related addresses of a flow's original source and destination are  $S_x^0$  and  $D_x^0$ , its routing addresses after the first time encoding are  $S_x^1$  and  $D_x^1$ , and its routing addresses after

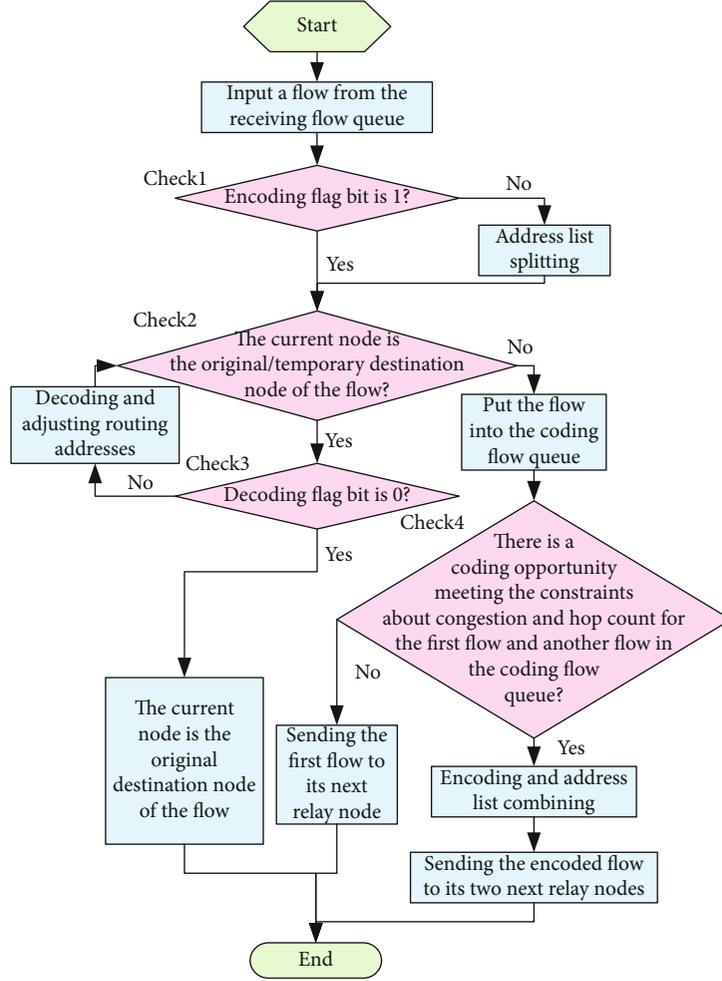


FIGURE 11: Flow chart of CARTA.

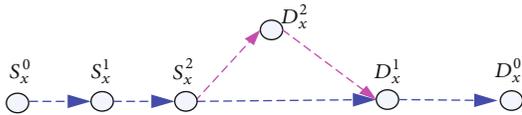


FIGURE 12: Detour incurred by encoding.

the second time encoding are  $S_x^2$  and  $D_x^2$ . Let  $H(S_x^2 \rightarrow D_x^1)$ ,  $H(S_x^2 \rightarrow D_x^2)$ , and  $H(D_x^2 \rightarrow D_x^1)$  denote the hop counts of the routing path sections from  $S_x^2$  to  $D_x^1$ , from  $S_x^2$  to  $D_x^2$ , and from  $D_x^2$  to  $D_x^1$ , respectively. Let  $H_x^2$  denote the increased hop count that is incurred by the second time encoding,  $H_x^2 = H(S_x^2 \rightarrow D_x^2) + H(D_x^2 \rightarrow D_x^1) - H(S_x^2 \rightarrow D_x^1)$ , if  $H_x^2 > 0$ , which means that the second time encoding incurs a detour increasing the hop count.

To solve the above dilemma, in Check4, constraints on congestion and hop count are defined in our coding condition judgment. Specifically, the congestion threshold for each node is denoted by  $L(\text{CF})_{\text{threshold}}$ , the increased hop count threshold for each coding opportunity is denoted by  $H_{\text{threshold}}$ , which can be adjusted according to the degree of congestion, the minimum increased hop count threshold for each coding opportunity is denoted by  $H_{\text{threshold}}^{\min}$ , and

the total increased hop count threshold caused by coding for each flow is denoted by  $H_{\text{threshold}}^{\text{total}}$ . In Check4, if there is a coding opportunity for two flows in the coding flow queue, let  $H$  denote the increased hop count caused by the coding opportunity,  $H^{\text{total}}$  denote the maximum of the total increased hop counts of the two flows caused by coding, and  $L(\text{CF})$  denote the number of flows in the coding flow queue. The two flows can be encoded together only when  $H \leq H_{\text{threshold}}$ ,  $H^{\text{total}} \leq H_{\text{threshold}}^{\text{total}}$ , and  $H_{\text{threshold}}$  follows the constraints as follows:

$$H_{\text{threshold}} = \begin{cases} H_{\text{threshold}}^{\min}, & \text{if } (L(\text{CF}) \leq L(\text{CF})_{\text{threshold}}) \\ H_{\text{threshold}}^{\min} \times (L(\text{CF})/L(\text{CF})_{\text{threshold}}), & \text{otherwise} \end{cases} \quad (3)$$

## 6. Performance Evaluation

In this section, to evaluate the performance of CARTA, simulations are executed in NS-2. In the simulations, 100 nodes are randomly deployed in a 100 m  $\times$  100 m network, each node has 0.5 J initial energy and follows the energy consumption model defined in [29], the communication radius of each

node is 15m, and the size of each packet is 1000 B. The MRMTT is formed with the root number denoted by  $R_n$ , and other parameters are set as follows:  $H_{\text{threshold}}^{\text{total}} = 10$ ,  $H_{\text{threshold}}^{\text{min}} = 5$ , and  $L(\text{CF})_{\text{threshold}} = 10$ .

Via rounds of random simulations, the network performance related to coding, data transmission, and energy consumption are collected and compared. In each round, 20 pairs of source and destination nodes are chosen randomly, and the source nodes are required to send packets to their corresponding destination nodes at predetermined flow rates. The duration for each round is 5 s, the CHs in the MRMTT are elected every 20 rounds, and the duration for the election of CHs is also 5 s.

The simulations mainly have two purposes: (1) to verify the potential of CARTA to surpass the typical XOR network coding methods FORM [7] and NCRT [14] (they also support the generalized coding structure and multilayer coding), in terms of throughput, overhead on coding, ratio of encoded flows, packet delivery ratio, transmission delay, and energy consumption and (2) to prove the effect of the MRMTT on the performance of coding, data transmission, and energy consumption in CARTA.

### 6.1. Comparison between CARTA and Typical Methods

**6.1.1. Performance Related to Coding.** To avoid the influence of network congestion, we set a low flow rate (30 kbps) in the simulations to estimate the ratios of encoded flows and overhead on coding.

In Figure 13, it can be seen that, in the same MRMTT, the ratio of encoded flows in CARTA is higher than that in FORM and NCRT, and as the number of roots increases, the ratio of encoded flows in CARTA rises more obviously than that in FORM and NCRT. The reason is that, in FORM and NCRT, a flow's routing path is invariable; thus, it may miss some real-time coding opportunities. But in CARTA, each flow's routing path is allowed to be adjusted dynamically to utilize more real-time coding opportunities, and in the MRMTT with more roots, each flow has more optional transmission paths and can reach the defined coding condition more easily.

Figure 14 shows that, in the same MRMTT, the overhead on coding in CARTA is much lower than that in FORM and NCRT. This is because, to detect coding opportunities, both of FORM and NCRT need to spend much overhead on collecting and sharing overhearing information. Differently, CARTA detects coding opportunities via simple tree address calculation. In CARTA, the overhead on coding is mostly spent on carrying tree addresses, and it will be relatively low if there are few roots. Further, theoretically, in CARTA, as the root number rises, each node has more addresses, and the overhead for each flow to carry these addresses gets higher; relatively, the overhead on coding increases. This result is also proved by the results in Figure 14.

**6.1.2. Performance Related to Data Transmission.** The average end-to-end throughput of CARTA, FORM, and NCRT are shown in Figure 15, from which it can be found that CARTA has higher average end-to-end throughput than

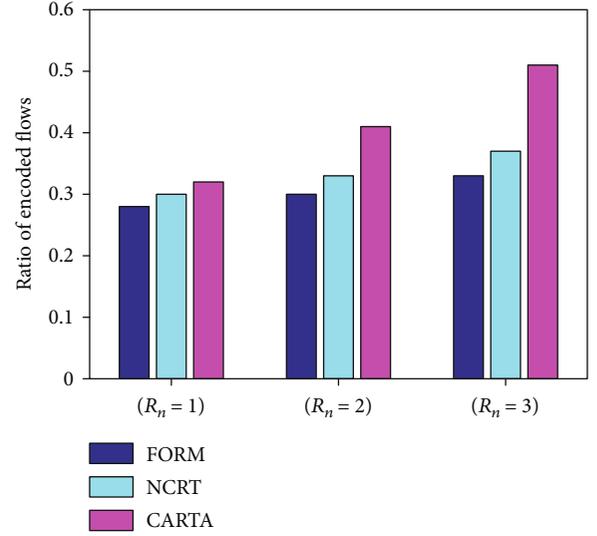


FIGURE 13: Ratio of encoded flows.

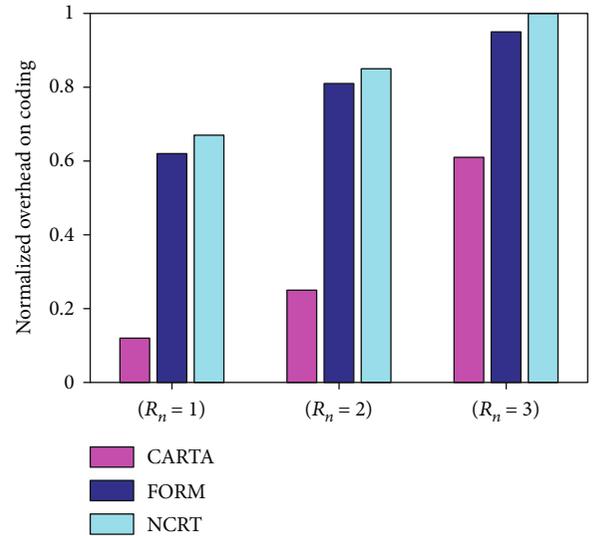


FIGURE 14: Overhead on coding.

FORM and NCRT in the same MRMTT ( $R_n = 3$ ), especially when the flow rate is high. This is because CARTA utilizes more real-time coding opportunities to promote network throughput and relieve network congestion.

Figure 16 shows us that the packet delivery ratio in CARTA is also higher than that in FORM and NCRT in the same MRMTT ( $R_n = 3$ ), especially when the flow rate is high. This result stems from three factors: (1) CARTA utilizes more coding opportunities and thus has advantages of throughput promotion and network congestion relief; (2) in CARTA, the dynamic routing and decoding based on the MRMTT and the tree address are more beneficial for reducing decoding failures; and (3) in FORM and NCRT, the static method to detect coding opportunities is not adapt to the dynamic network changes and thus incurs decoding failures.

Figure 17 shows that, in the same MRMTT ( $R_n = 3$ ), CARTA has lower average transmission delay than FORM

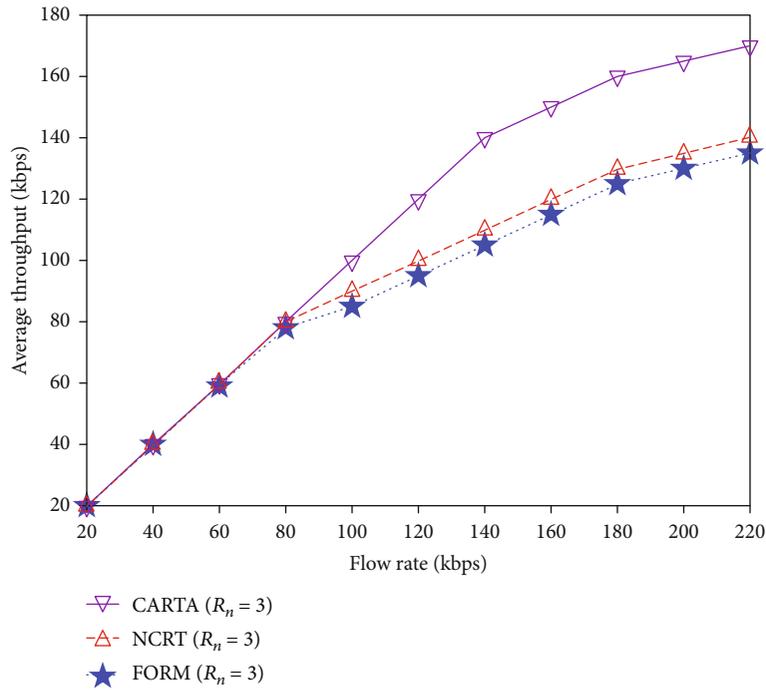


FIGURE 15: Average throughput.

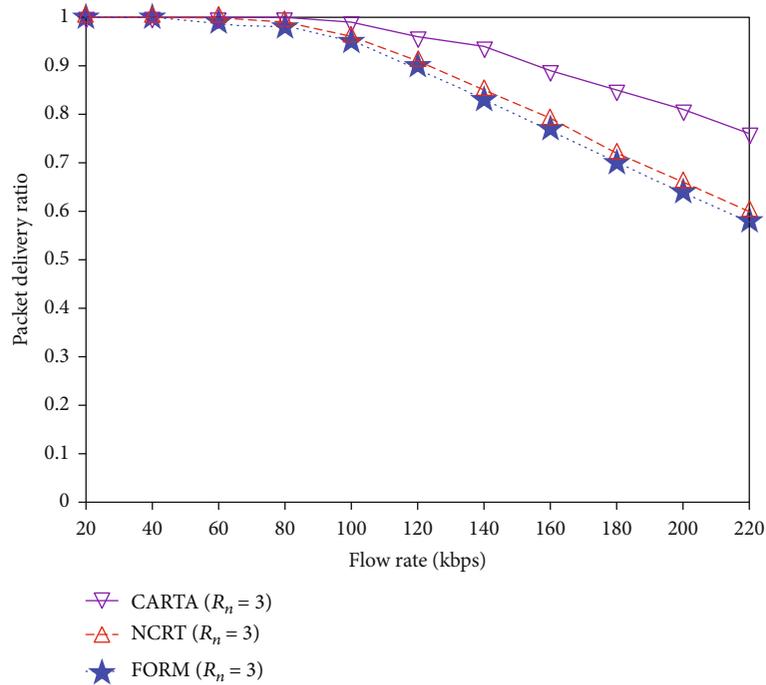


FIGURE 16: Packet delivery ratio.

and NCRT, especially when the flow rate is high; this is also due to utilization of more coding opportunities, higher average end-to-end throughput, and fewer decoding failures in CARTA.

6.1.3. *Performance Related to Energy Consumption.* To evaluate network energy consumption, the flow rate in each round

is set as 80 kbps, and each node is required to record its residual energy. The standard deviation of energy consumption for nodes is calculated to reflect the energy consumption balance degree, and the number of alive nodes is counted to reflect the network lifetime.

Figures 18 and 19 show that CARTA has a lower standard deviation of energy consumption and longer network

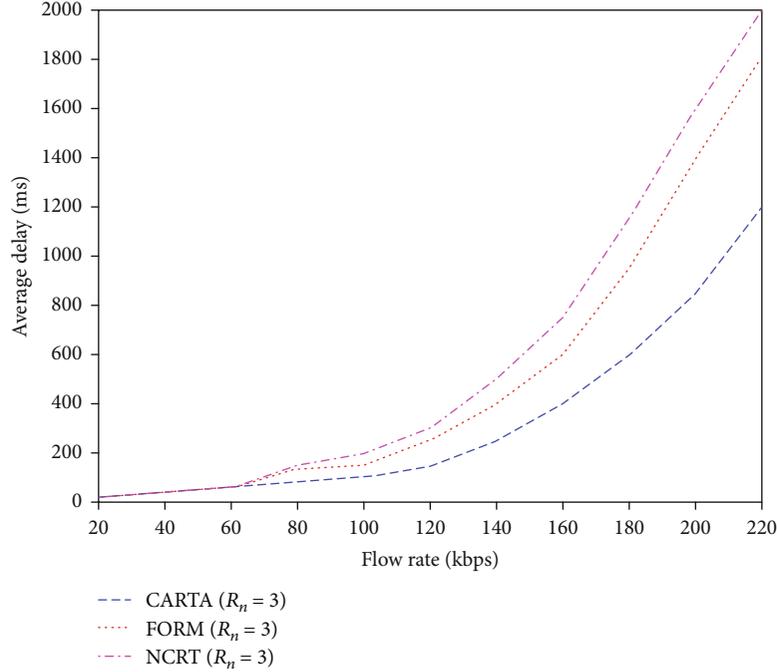


FIGURE 17: Average transmission delay.

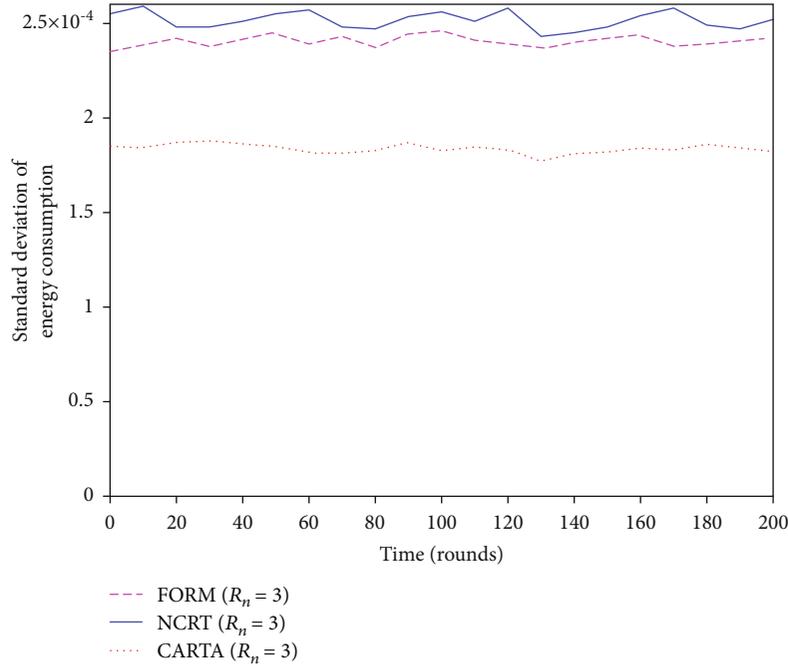


FIGURE 18: Energy consumption balance degree.

lifetime than FORM and NCRT in the same MRMTT ( $R_n = 3$ ). The reason is that CARTA utilizes more coding opportunities that can reduce data transmission times and save energy for each node, especially for the nodes closer to the roots.

6.2. Effect of the MRMTT on CARTA. The effect of the MRMTT on the coding performance of CARTA is shown in Figures 13 and 14. To further evaluate the effect of the

MRMTT on the performance related to data transmission and energy consumption in CARTA, simulations are performed by varying the number of roots in CARTA.

6.2.1. Effect of the MRMTT on Data Transmission. Figure 20 shows the effect of the MRMTT on the average end-to-end throughput in CARTA. It confirms that CARTA in the MRMTT with more roots can provide more optional

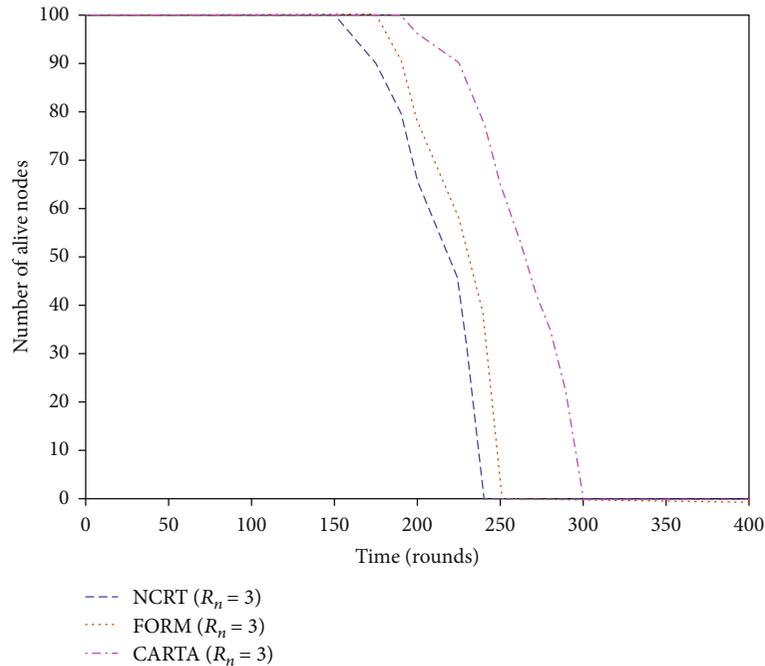


FIGURE 19: Network lifetime.

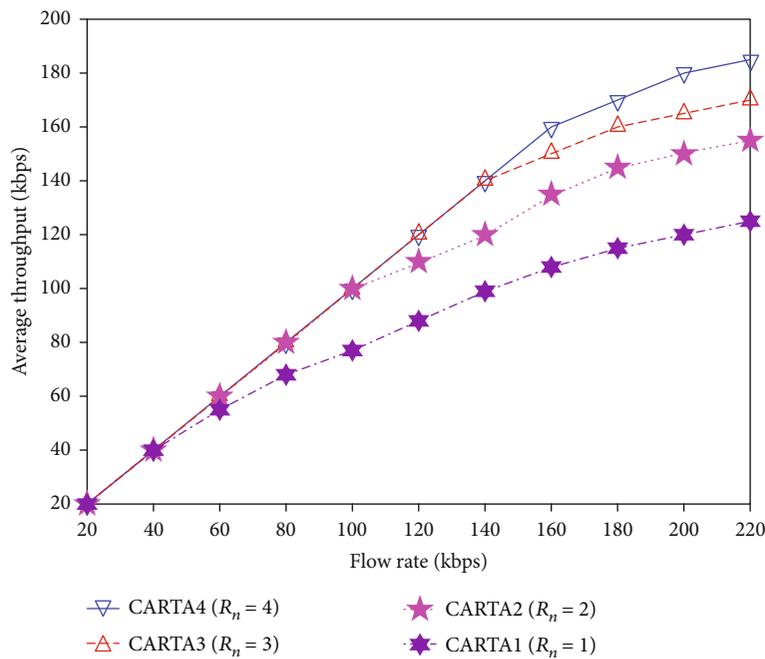


FIGURE 20: The effect of the MRMTT on average throughput.

transmission paths and bring more coding opportunities for each flow and thus has greater average end-to-end throughput.

In Figure 21, with the increase of root number, the packet delivery ratio of CARTA rises. This is because, more roots bring more backbone trees to provide more transmission paths and more coding opportunities, thereby promoting the throughput, relieving the flow congestion, and reducing transmission failures greatly.

In Figure 22, as the root number increases, the average transmission delay of CARTA decreases, which is also due to the benefits of the MRMTT on network throughput promotion and packet delivery ratio improvement.

6.2.2. *Effect of the MRMTT on Energy Consumption.* From Figures 23 and 24, it can be seen that, with the increase of the root number, the energy consumption balance degree in CARTA is improved, and the network lifetime

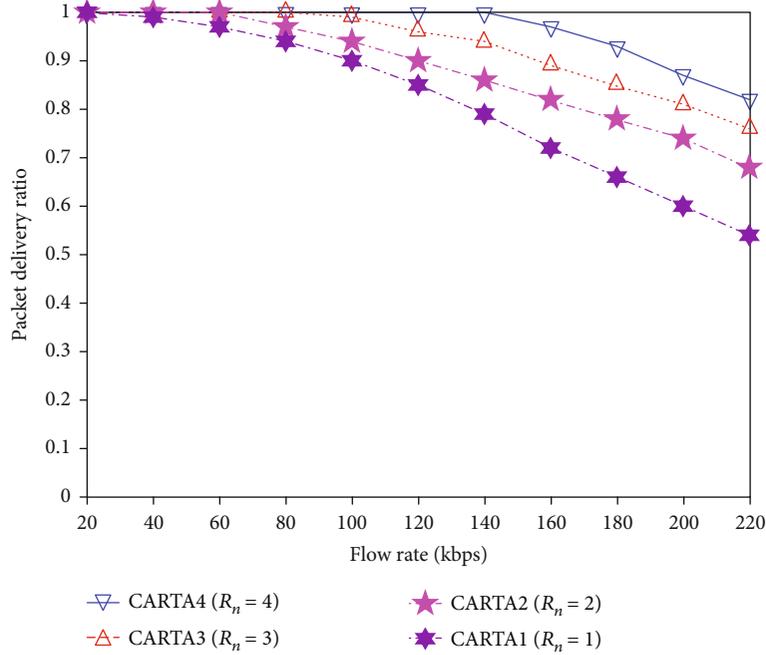


FIGURE 21: The effect of the MRMTT on packet delivery ratio.

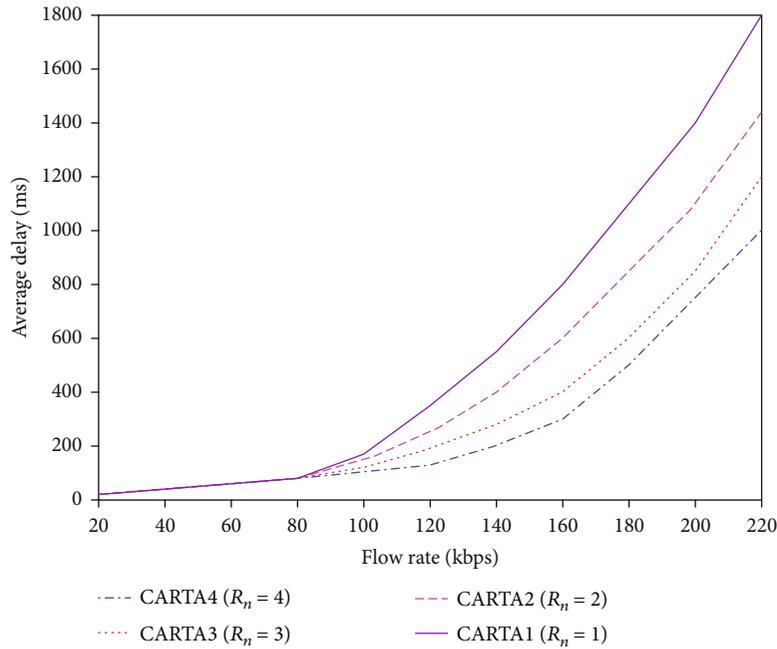


FIGURE 22: The effect of the MRMTT on average transmission delay.

is prolonged. This is because more roots bring more optional transmission paths and coding opportunities, which can reduce and balance the energy consumption in CARTA.

### 7. Conclusion

In this paper, low-overhead and dynamic Coding-Aware Routing via Tree-based Address (CARTA) is proposed for

WSNs. CARTA firstly constructs a Multi-Root Multi-Tree Topology (MRMTT) with the tree-based address allocation mechanism to provide transmission paths for data flows. It then defines a general coding condition for both original flows and encoded flows and provides a low-overhead coding condition judgment method to detect real-time coding opportunities via simple tree address calculation. In CARTA, the routing paths of flows can be adjusted dynamically and are compatible with their encoding and decoding. The

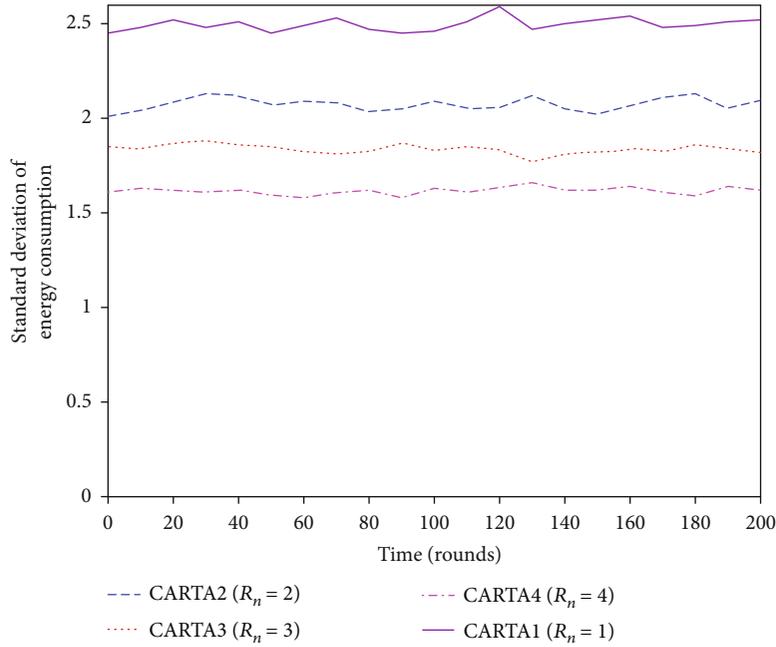


FIGURE 23: The effect of the MRMTT on energy consumption balance degree.

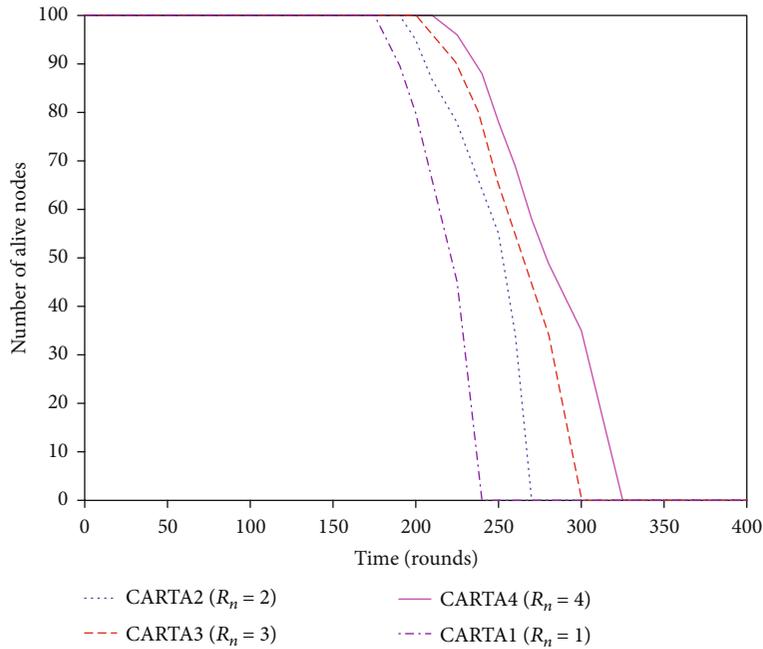


FIGURE 24: The effect of the MRMTT on network lifetime.

simulation results reveal that CARTA can utilize more coding opportunities and achieve greater throughput with less overhead on coding than FORM and NCRT, and the MRMTT has a great effect on data transmission performance in CARTA.

**Data Availability**

The data (figures) used to support the findings of this study are included within the article.

**Conflicts of Interest**

The authors declare that there is no conflict of interest regarding the publication of this paper.

**Acknowledgments**

This work was supported in part by the National Natural Science Foundation of China under grants (61802257) and (61602305) and by the Natural Science Foundation of Shanghai under grants (18ZR1426000) and (19ZR1477600).

## References

- [1] S. Kafaie, Y. Chen, O. A. Dobre, and M. H. Ahmed, "Joint inter-flow network coding and opportunistic routing in multi-hop wireless mesh networks: a comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1014–1035, 2018.
- [2] L. F. Xie, P. H. J. Chong, I. W. H. Ho, and Y. L. Guan, "A survey of inter-flow network coding in wireless mesh networks with unicast traffic," *Computer Networks*, vol. 91, pp. 738–751, 2015.
- [3] K. Lei, S. Zhong, F. Zhu, K. Xu, and H. Zhang, "An NDN IoT content distribution model with network coding enhanced forwarding strategy for 5G," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2725–2735, 2018.
- [4] M. A. Iqbal, B. Dai, B. Huang, A. Hassan, and S. Yu, "Survey of network coding-aware routing protocols in wireless networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1956–1970, 2011.
- [5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.
- [6] J. Le, J. C. S. Lui, and D.-M. Chiu, "DCAR: distributed coding-aware routing in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 596–608, 2010.
- [7] B. Guo, H. Li, C. Zhou, and Y. Cheng, "Analysis of general network coding conditions and design of a free-ride-oriented routing metric," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1714–1727, 2011.
- [8] G. Ding, Z. Sahinoglu, P. Orlik, J. Zhang, and B. Bhargava, "Tree-Based data broadcast in IEEE 802.15.4 and ZigBee networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 11, pp. 1561–1574, 2006.
- [9] R. Hou, K.-S. Lui, and J. Li, "Joint congestion control and scheduling in wireless networks with network coding," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 7, pp. 3304–3317, 2014.
- [10] M. Z. Farooqi, S. M. Tabassum, M. H. Rehmani, and Y. Saleem, "A survey on network coding: from traditional wireless networks to emerging cognitive radio networks," *Journal of Network and Computer Applications*, vol. 46, pp. 166–181, 2014.
- [11] A. Khreishah, I. M. 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.
- [12] 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*, vol. 24, no. 10, pp. 2015–2024, 2013.
- [13] N. M. Jones, B. Shrader, and E. Modiano, "Optimal routing and scheduling for a simple network coding scheme," in *2012 Proceedings IEEE INFOCOM*, pp. 352–360, Orlando, FL, USA, March 2012.
- [14] G. X. Kok, C. O. Chow, and H. Ishii, "Improving network coding in wireless ad hoc networks," *Ad Hoc Networks*, vol. 33, pp. 16–34, 2015.
- [15] H. Alshaheen and H. T. Rizk, "Improving the energy efficiency for the WBSN bottleneck zone based on random linear network coding," *IET Wireless Sensor Systems*, vol. 8, no. 1, pp. 17–25, 2018.
- [16] R. R. Rout and S. K. Ghosh, "Enhancement of lifetime using duty cycle and network coding in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 656–667, 2013.
- [17] L. F. Xie, P. H. J. Chong, I. W. H. Ho, and H. C. B. Chan, "Virtual overhearing: an effective way to increase network coding opportunities in wireless ad-hoc networks," *Computer Networks*, vol. 105, pp. 111–123, 2016.
- [18] L. F. Xie, P. H. J. Chong, and Y. L. Guan, "Performance analysis of network coding with virtual overhearing in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 2051–2061, 2015.
- [19] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "CORE: a coding-aware opportunistic routing mechanism for wireless mesh networks [Accepted from Open Call]," *IEEE Wireless Communications*, vol. 17, no. 3, pp. 96–103, 2010.
- [20] P. Erik, A. Kemal, Š. Aleš, and M. Mohorčič, "CANCAR – congestion-avoidance network coding-aware routing for wireless mesh networks," *KSI Transactions on Internet and Information Systems*, vol. 12, no. 9, pp. 4205–4226, 2018.
- [21] H. Xu and B. Li, "An optimization framework for XOR-assisted cooperative relaying in cellular networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 979–991, 2014.
- [22] S. Kafaie, Y. Chen, M. H. Ahmed, and O. A. Dobre, "FlexONC: joint cooperative forwarding and network coding with precise encoding conditions," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7262–7277, 2017.
- [23] J. Chen, K. He, R. du, M. Zheng, Y. Xiang, and Q. Yuan, "Dominating set and network coding-based routing in wireless mesh networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 423–433, 2015.
- [24] L. F. Xie, P. H. J. Chong, S. C. Liew, and Y. L. Guan, "CEO: consistency of encoding and overhearing in network coding-aware routing," *IEEE Wireless Communications Letters*, vol. 2, no. 2, pp. 187–190, 2013.
- [25] Z. Zhou and L. Zhou, "Encoded packet-assisted retransmission for multiple unicast flows in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3409–3422, 2013.
- [26] Y.-H. Chen, E. H. K. Wu, C. H. Lin, and G. H. Chen, "Bandwidth-satisfied and coding-aware multicast protocol in MAN-ETs," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1778–1790, 2018.
- [27] H. Liu, H. Yang, Y. Wang, B. Wang, and Y. Gu, "CAR: coding-aware opportunistic routing for unicast traffic in wireless mesh networks," *Journal of Network and Systems Management*, vol. 23, no. 4, pp. 1104–1124, 2015.
- [28] L. Hai, J. Wang, P. Wang, H. Wang, and T. Yang, "High throughput network coding aware routing in time-varying multi-hop networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6299–6309, 2017.
- [29] Z. Peng, H. Yin, H. T. Dong, and H. Li, "LEACH protocol based two-level clustering algorithm," *International Journal of Hybrid Information Technology*, vol. 8, no. 10, pp. 15–26, 2015.
- [30] W. Qiu, E. Skafidas, and P. Hao, "Enhanced tree routing for wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 3, pp. 638–650, 2009.